

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

October 25, 2011

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 a distance function defined between data points, biasing nearby data to share more features. The choice of distance function allows for many kinds of dependencies, including temporal or spatial. Further, the original IBP is a special case of the dd-IBP. In this paper, we develop the dd-IBP and theoretically characterize the distribution of how features are shared between data. We derive a Markov chain Monte Carlo sampler for a linear Gaussian model with a dd-IBP prior and study its performance on several data sets for which exchangeability is not a reasonable assumption.

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. Statistical methods for inferring the latent features are dimensionality reduction methods, such as principal component analysis, factor analysis, and probabilistic matrix factorization (Bishop, 2006). Dimensionality reduction methods characterize a small set of dimensions and model each data point with a weighted average of them. (These weights are the latent features.) Dimensionality reduction can help form predictions about future data and provide an exploratory tool for discovering hidden structures in observed data.

Dimensionality reduction methods typically require that the number of latent features (i.e., the number of dimensions) is 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; Griffiths and Ghahramani, 2005, 2011), 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, or “infinite” latent feature models, have been applied to several statistical problems (e.g., Knowles and Ghahramani, 2007; Meeds et al., 2007; Miller et al., 2009; Navarro and Griffiths, 2008).

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 will tend to be autocorrelated over time; the latent features describing environmental risk factors will be 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 have been devised to allow arbitrary dependencies between datapoints (e.g., Rasmussen and Ghahramani, 2002; Griffin and Steel, 2006; Caron et al., 2007; Duan et al., 2007). Among these methods is the *distance dependent Chinese restaurant process* (dd-CRP; Blei and Frazier, 2010). The dd-CRP is a non-exchangeable generalization of the Chinese restaurant process, the prior over partitions of data that emerges in Bayesian nonparametric mixture modeling (Escobar and West, 1995; Rasmussen, 2000).

The dd-CRP models non-exchangeability by using using distances between data that are external to the observations themselves—nearby data are more likely to be assigned to the same partition, i.e., the same mixture component. The dd-IBP extends these ideas to infinite latent feature models, where distances between data induce sharing of latent features between data. Intuitively, nearby data (e.g., in time or space) should be more likely to share latent features than distant data.

We review the Indian buffet process in Section 2.1 and develop the distance dependent IBP in Section 2.2. Several other infinite latent feature models have been developed to capture dependencies between data in different ways, for example using phylogenetic trees (Miller et al., 2008) or latent Gaussian processes (Williamson et al., 2010). Of particular relevance to this work is the model of Zhou et al. (2011), which uses a hierarchical beta process construction 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. (2011). 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; Andrieu et al., 2003) 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 complex data.

2 The distance dependent Indian buffet process

We first review the definition of the Indian buffet process (IBP) and its role in defining infinite latent feature models. We then introduce the distance dependent IBP.

2.1 The Indian buffet process

The Indian buffet process is a prior over binary matrices \mathbf{Z} with an infinite number of columns (Griffiths and Ghahramani, 2005, 2011). 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. 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 $\text{Poisson}(\alpha)$ 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 of random length and, beyond the last dish sampled, the remaining columns equal to zero.

Subsequent customers $i = 2, \dots, N$ enter and sample some dishes. Each samples the previously sampled dishes according to their popularity,

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

where m_k is the number of previous customers that sampled dish k . (We emphasize that k is restricted to dishes that were previously sampled.) Then, each samples a $\text{Poisson}(\alpha/i)$ number of new dishes. Again these are represented as a contiguous block of ones in the columns beyond the last dish that at least one previous customer has sampled.

Though described sequentially, Griffiths and Ghahramani (2005) showed that 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 distance dependent Indian Buffet Process (dd-IBP) is a distribution over binary latent feature matrices with a finite number of rows and an infinite number of columns. The idea is that the customers have a set of distances between them, 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, such that each dish is “owned” by a single customer. (These are akin to the new dishes in the IBP process.)

Then, for each dish (that is, feature column), customers connect to one another. The probability that one customer connects to another decreases in the distance between them. Note that customers are not sampling each dish, as in the IBP, but rather connecting to another customer.

The last step is to compute dish inheritance: A customer deterministically inherits a dish if its owner (from the first step) is reachable in the connectivity graph for that dish. (If you insist on a complete gastronomical metaphor, customer connectivity can be thought of as “I’ll have what he’s having.”) The dishes that each customer samples are those that he inherits or owns. Thus, the distance function induces similarity of sampled dishes between nearby customers.

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

- There is an infinite set of *dishes*. The set of dishes owned by customer i is \mathcal{K}_i ; the cardinality of this set is $\lambda_i = |\mathcal{K}_i|$; 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} . The set of unowned dishes is \mathcal{K}_\emptyset .

- The dd-IBP requires a *distance matrix*. The $N \times N$ distance matrix between customers is \mathbf{D} , where the distance between customers i and j is d_{ij} . We require that $d_{ii} = 0$.

The *decay function* $f(d)$ maps distance to proximity. 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 $h_i = \sum_{j=1}^N f(d_{ij})$.

- The hidden variables connect each customer to another, within each dish. The *connectivity matrix* is \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}_k^* , where $c_k^* \in \{1, \dots, N\}$ indicates the customer who owns dish k .

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

1. **Assign dish ownership.** Initialize all dishes to be unowned, $\mathcal{K}_\emptyset = \{1, \dots, \infty\}$. For each customer i , allocate a Poisson(α/h_i) number of unowned dishes in \mathcal{K}_\emptyset to \mathcal{K}_i and remove these dishes from \mathcal{K}_\emptyset . For each $k \in \mathcal{K}_i$, set the ownership vector $c_k^* = i$.
2. **Assign customer connections.** For each customer i and dish $k \in \mathcal{K}_{-i}$, draw a customer assignment according to $P(c_{ik} = j | \mathbf{D}, f) = a_{ij}$. Note that customers can connect to themselves. (In that case, they do not inherit a dish unless they are its owner. 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^* . We encode reachability with \mathcal{L} . If j is reachable from i then $\mathcal{L}_{ijk} = 1$. Otherwise $\mathcal{L}_{ijk} = 0$.

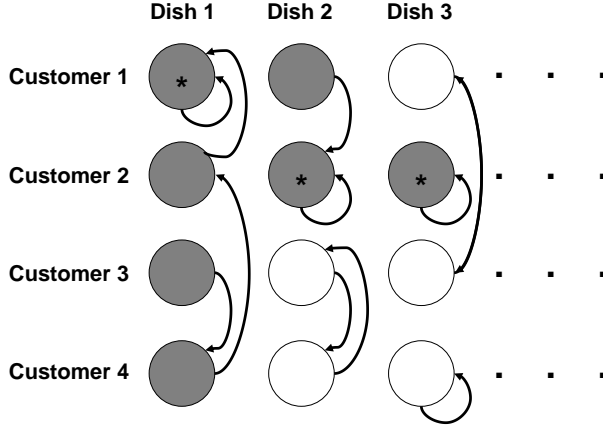


Figure 1: **Schematic of the dd-IBP.** An example of a latent feature matrix generated by the dd-IBP. Row correspond to customers (datapoints) and columns correspond to dishes (features). Customers connect to each other, as indicated by arrows. Customers inherit a dish if the owner of that dish (c_k^* , indicated by stars) is reachable by a sequence of connections. Gray shading indicates that a feature is active for a given datapoint.

4. **Compute the feature indicator matrix.** For each customer i and dish k we set $z_{ik} = 1$ if i owns k or reaches the owner of k .

An example of customer assignments sampled from the dd-IBP is shown in Figure 1. In this example, customer 1 owns dish 1; customers 2-4 are all linked to 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 is linked to customer 2, and hence feature 2 is active for customers 1 and 2. Dish 3 is owned by customer 2, but no other customers link to 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,

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

Consider the first term. Recall that the number 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

$$p(\mathbf{c}^* | D, \alpha, f) = \prod_{k=1}^K p(\mathcal{K}_i | D, \alpha, f), \quad (3)$$

where each \mathcal{K}_i is a Poisson random variable with mean α/h_i , and $h_i = \sum_{j=1}^N f(d_{ij})$.

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 A (derived from the

distances and decay function),

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

We use this expression when monitoring convergence of our inference algorithm (Section 5). More precisely, we monitor the log of the joint distribution of the data, α , \mathbf{C} and the number of dishes that each customer owns.

Note that the only random variables in this process are the customer connections within each dish \mathbf{C} , and the ownership vector \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 many-to-one function of the dd-IBP variables, which we denote by ϕ . We compute the probability of a binary matrix by marginalizing out the appropriate configurations of the dd-IBP variables

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

We want to highlight two special cases. First, we call the distance function *sequential* when $d_{ij} = \infty$ for $j > i$. In this case, customers can only connect to previous customers. Second, when $f(d) = 1$ for all $d < \infty$ and the distance function is sequential, the dd-IBP reduces to the standard IBP. To see this, consider the probability that the k th dish is sampled by the i th customer (that is, $z_{ik} = 1$). This will be equal to the proportion of previous customers that already reach c_k^* (because the probability of connecting to each customer is proportional to one). Thus, this is equivalent to m_k/i , which is the same probability in the IBP.

Many different decay functions are possible within this framework. Figure 2 shows samples of \mathbf{Z} using four decay functions and a sequential distance defined by absolute temporal distance.

- 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 different ways.

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 ddCRP, which is not marginally invariant either.) In some

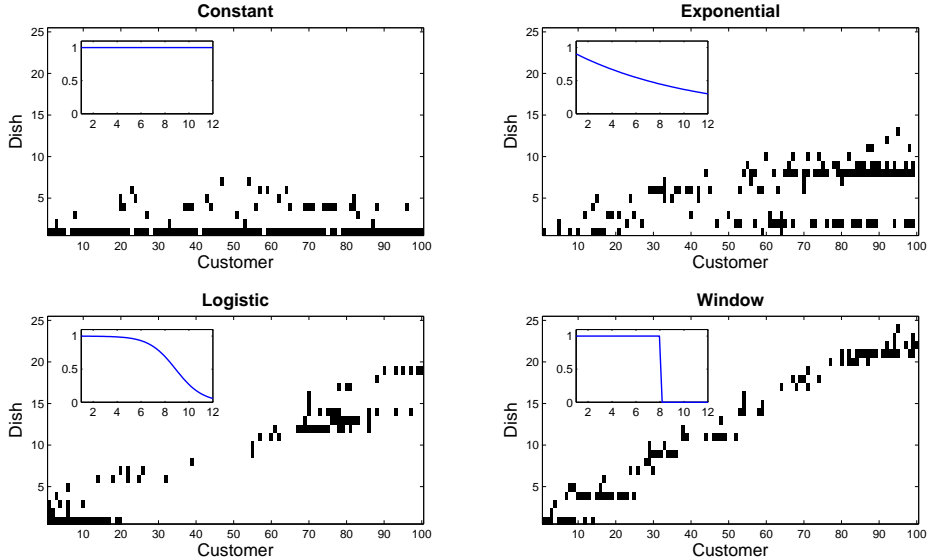


Figure 2: **Decay functions.** Each panel presents a different latent feature matrix, sampled from the dd-IBP with sequential distances and the decay functions shown in the insets.

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 do. This is less of an issue for exploratory analysis of fully observed datasets. Furthermore, marginal invariance may not be an appropriate modelling assumption for some datasets.

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 distance 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,

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

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 will focus on the most closely related model, which is the *dependent hierarchical beta process* (dHBP; Zhou et al., 2011). 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, the 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, B is a random measure on the feature space Ω and $P(B)$ is the de Finetti mixing distribution (see Bernardo and Smith, 1994). Thibaux and Jordan (2007) 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,

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

If B_0 is non-atomic, then p_k is a draw from a degenerate beta distribution; if it is atomic and of the form $B_0 = \sum_k q_k \delta_{\omega_k}$, then $p_k \sim \text{Beta}(cq_k, c(1 - q_k))$. Following Thibaux and Jordan (2007), we define the *mass parameter* as $\gamma = B_0(\Omega)$. Note that B_0 is not a probability measure, and hence γ can take on values greater than 1.

Conditional on a draw from the beta process, the feature representation of datapoint i is generated by drawing from the *Bernoulli process* (BeP) with base measure B : $X_i|B \sim \text{BeP}(B)$. This corresponds to activating each feature with probability p_k . Sampling \mathbf{Z} from the compound beta-Bernoulli process is equivalent to sampling \mathbf{Z} directly from the IBP when $c = 1$ and $\gamma = \alpha$ (Thibaux and Jordan, 2007).

3.2 Dependent hierarchical beta processes

The *dependent hierarchical beta process* (dHBP) builds external dependence between data points into the Beta process model Zhou et al. (2011). The dependencies are induced by mixing independent BP random measures, weighted by their proximities \mathbf{A} .

The dHBP is based on the following generative model process,

$$\begin{aligned} X_i &\sim \text{BeP}(B_{g_i}^*), & g_i &\sim \text{Multinomial}(\mathbf{a}_i), \\ B_j^* &\sim \text{BP}(c_1, B), & B &\sim \text{BP}(c_0, B_0). \end{aligned} \quad (8)$$

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

$$X_i \sim \text{BeP}(B_i), \quad B_i = \sum_{j=1}^N a_{ij} B_j^*. \quad (9)$$

Dependencies between datapoints are captured in the dHBP by the proximity matrix \mathbf{A} , as in the dd-IBP.¹ Intuitively, proximal datapoints (e.g., in time or space) should 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 datapoints, 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 et al. (2010) used a hierarchical Gaussian process to couple the latent features of data in a covariate-dependent manner. Their framework is elegant and 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. In contrast, our inference algorithm directly samples the latent features without requiring auxiliary variables.

Miller et al. (2008) proposed a “phylogenetic IBP” that encodes tree-structured dependencies between data. Doshi-Velez and Ghahramani (2009b) proposed a “correlated IBP” that couples datapoints 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. (2008) 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. 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 more natural and richer specification of the dependency structure between observations than either the phylogenetic or correlated IBP.

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. We consider an asymptotic regime in which

¹Zhou et al. (2011) formalize dependencies in an equivalent manner using a normalized kernel function defined over pairs of covariates associated with the datapoints.

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) 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 .

Proposition 1. *Let R_i denote the number of features held by X_i and R_{ij} denote the number of features shared by X_i and X_j , where $i \neq j$. Then under the dd-IBP,*

$$R_i \sim \text{Poisson} \left(\alpha \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1) \right) \quad (10)$$

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

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 $\alpha \rightarrow \infty$:*

$$\frac{R_i}{\alpha} \xrightarrow{d} \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1) \quad (12)$$

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

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

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 datapoints 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 datapoints 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. Let R_i denote the number of features held by X_i and R_{ij} denote the number of features shared by X_i and X_j , where $i \neq j$. If B_0 is continuous, then under the dHBP,

$$R_i | \mathbf{g}_{1:N} \sim \text{Poisson}(\gamma) \quad (15)$$

$$R_{ij} | \mathbf{g}_{1:N} \sim \begin{cases} \text{Poisson}\left(\gamma \frac{c_0+c_1+1}{(c_0+1)(c_1+1)}\right) & \text{if } g_i = g_j \\ \text{Poisson}\left(\gamma \frac{1}{c_0+1}\right) & \text{if } g_i \neq g_j. \end{cases} \quad (16)$$

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 $\gamma \rightarrow \infty$:

$$\frac{R_i}{\gamma} \xrightarrow{d} 1 \quad (17)$$

$$\frac{R_{ij}}{\gamma} \xrightarrow{d} \begin{cases} \frac{c_0+c_1+1}{(c_0+1)(c_1+1)} & \text{if } g_i = g_j \\ \frac{1}{c_0+1} & \text{if } g_i \neq g_j. \end{cases} \quad (18)$$

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

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

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

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 $i \neq j$,

$$\frac{R_i}{\alpha} \xrightarrow{d} 1, \quad \frac{R_{ij}}{\alpha} \xrightarrow{d} \frac{1}{2}, \quad \frac{R_{ij}}{R_i} \xrightarrow{d} \frac{1}{2}. \quad (21)$$

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 datapoints. 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 simplified behaviour 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 datapoints. Given a matrix specifying the proportion of features that are believed to be shared by pairs of datapoints, 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 relatively 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 datapoints 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 datapoints 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 datapoints 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 3 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 $\gamma = \alpha =$

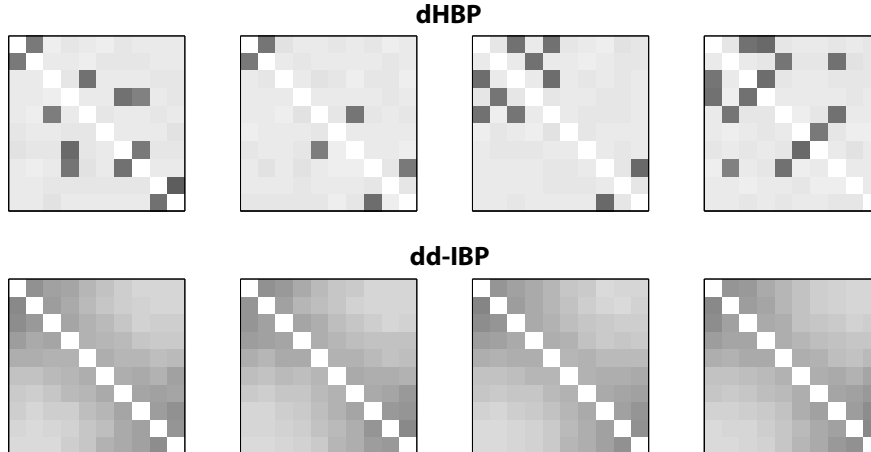


Figure 3: **Feature-sharing in the dHBP and dd-IBP, limiting case.** Along the horizontal axis, we show 4 independent draws from the dHBP (*Top*) and dd-IBP (*Bottom*). Within each subfigure, the shade of a cell (i, j) shows the fraction R_{ij}/R_i , where R_i is the number of features held by datapoint 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 datapoints, but does not model uncertainty about these proportions when mass parameters are large.

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 wide variety of different values, but remain unchanged across independent draws.

Figure 4 illustrates non-asymptotic feature-sharing behavior in a simple setting with only two datapoints. The figure shows the dd-IBP (left column) and the dHBP (right column) 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 $p(R_{ij} = r)$, with r along the vertical axis, as a function of the distance d_{ij} (for the dd-IBP) or the proximity a_{ij} (for the dHBP). For comparability, we set $d_{ij} = 1/a_{ij}$. Because there are only two datapoints, and $a_{ii} = 1$, $a_{ij} = a_{ji}$, 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 4 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, the probability mass function $p(R_{ij} = r)$ concentrates on larger values of r as a_{ij} increases. However, the way in which the probability mass functions change with a_{ij} is very different 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 near $a_{ij} = 0.7$. As one varies a_{ij} across its full range, the set of most likely values for R_{ij} under

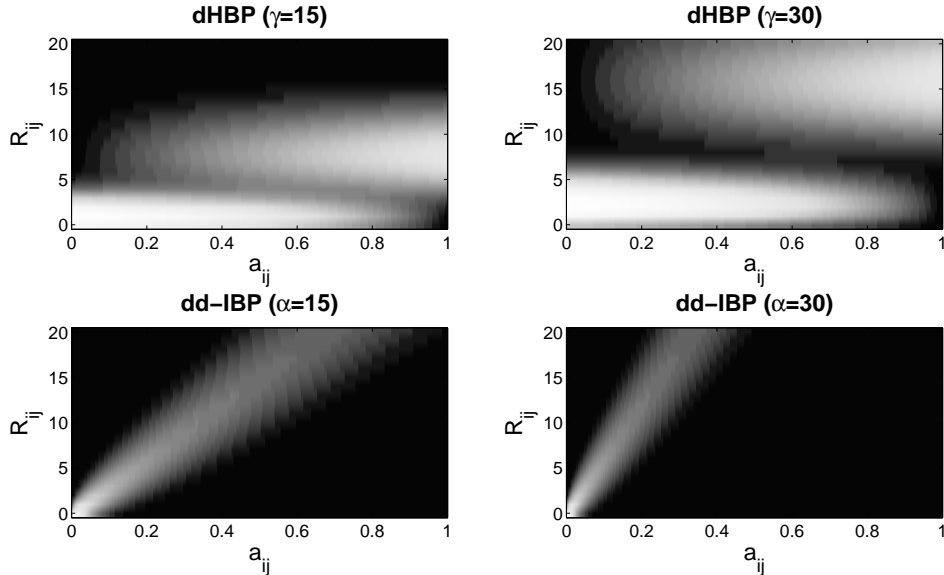


Figure 4: **Feature-sharing in the dHBP and dd-IBP.** Heatmaps of the probability mass function over the number of shared features R_{ij} (y-axis) as a function of proximity a_{ij} (x-axis) in a data set consisting of two datapoints. For the dHBP, we set $c_0 = 10$ and $c_1 = 1$. Note that $\mathbb{E}[R_i]$ is the same for both the dHBP and dd-IBP in these examples (when $\alpha = \gamma$).

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, $\gamma/(c_0 + 1) = (0.91)\gamma$ and $\gamma(c_0 + c_1 + 1)/(c_0 + 1)(c_1 + 1) = (0.55)\gamma$.

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 $\mathbf{X} = \{\mathbf{x}_i\}_{i=1:N}$, 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:

$$P(\mathbf{C}, \mathbf{c}^*, \theta, \alpha | \mathbf{X}, \mathbf{D}, f) \propto P(\mathbf{X} | \mathbf{C}, \mathbf{c}^*, \theta) P(\theta) P(\mathbf{C} | \mathbf{D}, f) P(\mathbf{c}^* | \alpha) P(\alpha), \quad (22)$$

where the first term is the likelihood, 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 . Exact inference in this model is computationally

intractable. In Section 6, we examine a specific instantiation of this model under linear-Gaussian assumptions.

We will use Monte Carlo Markov chain (MCMC) sampling 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.

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:

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

where the prior and likelihood will vary from problem to problem. 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 (Andrieu et al., 2003). Generally, updates for θ will be further broken down 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:

$$P(\alpha|\mathbf{c}^*, \mathbf{D}, f) \propto P(\alpha) \prod_{i=1}^N \text{Poisson}(\lambda_i; \alpha/h_i), \quad (24)$$

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:

$$\alpha|\mathbf{c}^*, \mathbf{D}, f \sim \text{Gamma} \left(\nu_\alpha + \sum_{i=1}^N \lambda_i, \eta_\alpha + \sum_{i=1}^N h_i^{-1} \right). \quad (25)$$

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

$$P(c_{ik}|\mathbf{c}_{-i}, \mathbf{x}_i, \mathbf{c}^*, \theta, \mathbf{D}, f) \propto P(\mathbf{x}_i|\mathbf{C}, \mathbf{c}^*, \theta)P(c_{ik}|\mathbf{D}, f), \quad (26)$$

where \mathbf{x}_i denotes the i th row of \mathbf{X} , \mathbf{c}_i denotes the i th row of \mathbf{C} , and \mathbf{c}_{-i} denotes \mathbf{C} excluding row i .² The first factor in Eq. 26 is the likelihood,³ and the second factor is the prior, given by

²We 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 .

³In calculating the likelihood, we only include the active columns of \mathbf{Z} (i.e., those for which $\sum_{n=1}^N z_{nk} > 0$).

$P(c_{ik} = j | \mathbf{D}, f) = a_{ij}$. In considering possible assignments of c_{ik} , one of two scenarios will occur: Either datapoint i reaches the owner of k (in which case feature k becomes active for i as well as for all other datapoints that reach i), or it doesn't (in which case feature k becomes inactive for i as well as for all other datapoints 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. A new connectivity matrix, \mathbf{C}' , and ownership vector, $\mathbf{c}^{*'}$, 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 datapoint $i = 1, \dots, N$.
2. Set $\mathbf{C}' \leftarrow \mathbf{C}$. Then populate or depopulate it by performing, for each $i = 1, \dots, N$,
 - (a) If $\lambda'_i > \lambda_i$, reallocate $\lambda'_i - \lambda_i$ dishes from in \mathcal{K}_\emptyset to \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 $\lambda'_i < \lambda_i$, reallocate $\lambda_i - \lambda'_i$ randomly selected dishes in \mathcal{K}_i to \mathcal{K}_\emptyset .

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 canceling out):

$$\zeta = \frac{P(\mathbf{X} | \mathbf{C}', \mathbf{c}^{*'}, \theta)}{P(\mathbf{X} | \mathbf{C}, \mathbf{c}^*, \theta)}. \quad (27)$$

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

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 (see Figure 5). This model was originally studied for the IBP by Griffiths and Ghahramani (2005, 2011). 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:

$$\mathbf{X} = \mathbf{Z}\mathbf{W} + \epsilon, \quad (28)$$

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

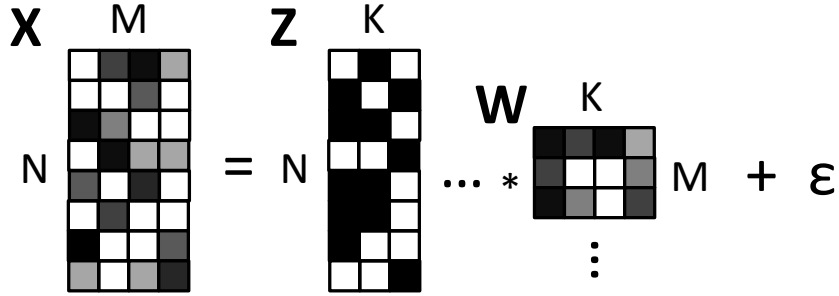


Figure 5: **Linear-Gaussian model**. Matrix multiplication view of how latent features (\mathbf{Z}) combine with a weight matrix (\mathbf{W}) and white noise (ϵ) to produce observed data.

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:

$$\begin{aligned}
 P(\mathbf{X}|\mathbf{Z}) &= \int_{\mathbf{W}} P(\mathbf{X}|\mathbf{Z}, \mathbf{W})P(\mathbf{W})d\mathbf{W} \\
 &= \frac{\exp\left\{-\frac{1}{2\sigma_x^2}\text{tr}(\mathbf{X}^T(\mathbf{I} - \mathbf{Z}\mathbf{H}^{-1}\mathbf{Z}^T)\mathbf{X})\right\}}{(2\pi)^{NM/2}\sigma_x^{(N-K)M}\sigma_w^{KM}|\mathbf{H}|^{M/2}}, \tag{29}
 \end{aligned}$$

where $\text{tr}(\cdot)$ denotes 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 and regression tasks when the datapoints are non-exchangeable. We then show how the dd-IBP can be applied to missing data problems.

7.1 Dimensionality reduction for classification and regression

The performance of supervised learning algorithms is often enhanced by pre-processing the data to reduce its dimensionality (Bishop, 2006). Classical techniques for dimensionality reduction, such as principal components analysis and factor analysis, assume exchangeability, as does the infinite latent feature model. For this reason, these techniques may not work as well for pre-processing

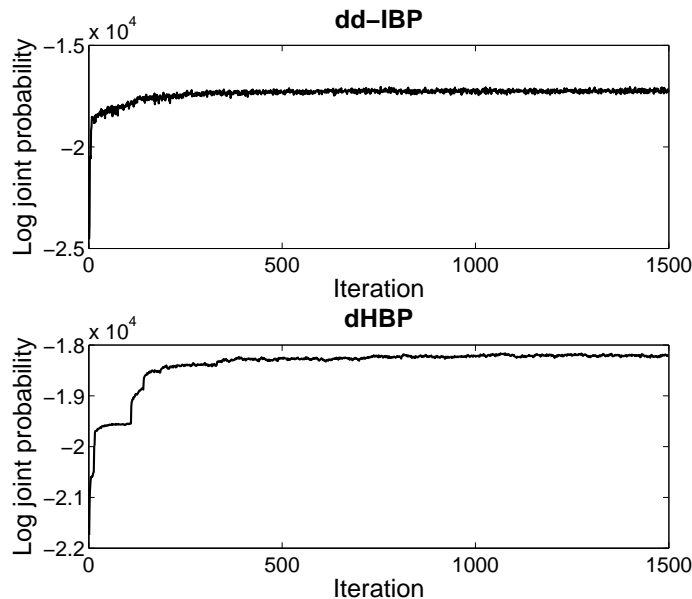


Figure 6: **Trace plots.** Representative traces of the log joint probability of the Alzheimer’s data and latent variables for the dd-IBP (*top*) and dHBP (*bottom*). Each iteration corresponds to a Gibbs sweep over all the latent variables.

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 (Christou and Dinov, 2011).⁴ 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. The regression task is to predict individual clinical dementia rating scores, a standard numerical scale used to quantify dementia severity (Morris, 1997).

Age-related changes in brain structure produce natural declines in cognitive function that make diagnosis of Alzheimer’s disease difficult (e.g., Erkinjuntti et al., 1986). Thus, it is important to take age into account 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 . 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 Gibbs 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 (L2-regularized logistic regression for classification, L2-

⁴Available at: http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_July2009_ID_NI.

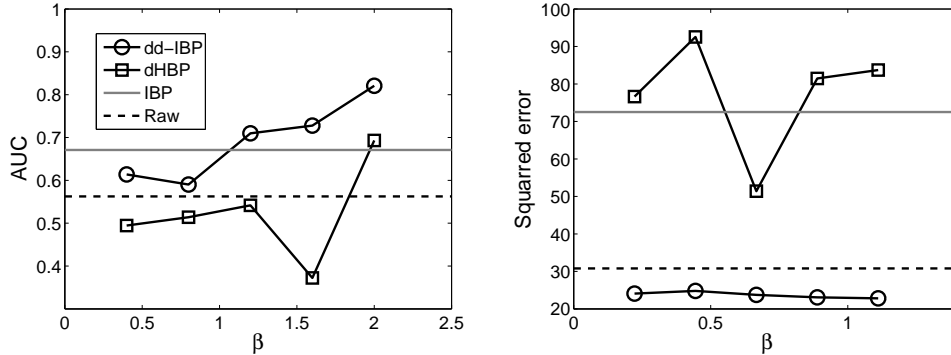


Figure 7: **Classification and regression results for Alzheimer’s data set.** (*Left*) Area under the curve (AUC) for binary classification (Alzheimer’s vs. normal control) using regularized logistic regression. Each curve represents a different choice of predictor variables for logistic regression. 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 and dHBP results were based on using the latent features of the *maximum a posteriori* sample following 1500 iterations of Gibbs sampling. (*Right*) Squared error for prediction of clinical dementia ratings using regularized linear regression.

regularized linear regression for prediction of clinical dementia ratings, with the regularization constant set to 10^{-6} in both cases). Training was performed on half of the data, and testing on the other half.⁵ The noise hyperparameters of the dd-IBP and dHBP (σ_x and σ_y) were updated using Metropolis-Hastings proposals. Visual inspection of the log joint probability traces suggested that the sampler reaches a local maximum within 400-500 iterations (Figure 6). 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, regression, and reconstruction measures for each restart, averaging the resulting measures to reduce sampling variability. For comparison, we also made predictions using the standard IBP and the raw observed features (i.e., no pre-processing).

Classification results are shown in Figure 7 (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.

Regression results are shown in Figure 7 (right), where performance is measured by the squared error between predicted and observed clinical dementia. In this case, lower numbers indicate better performance. As with the classification task, pre-processing with the dd-IBP produces superior predictive performance compared to the alternative models. Unlike with the classification task, the dd-IBP results do not appear to be sensitive to the value of β .

⁵A few individuals were removed from the test set to make it balanced.

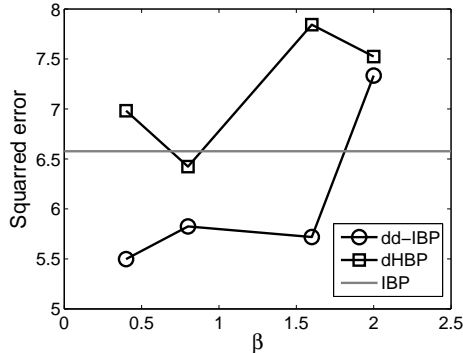


Figure 8: **Reconstruction of missing EEG data.** Reconstruction error for latent feature models as a function of the exponential decay function parameter, β . Results were based on the *maximum a posteriori* sample following 1500 iterations of Gibbs sampling.

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 this and the standard IBP sampler with respect to the resulting performance on the Alzheimer’s classification and the EEG reconstruction (see next section); there was a significant difference in the Alzheimer’s regression results, with the dd-IBP sampler giving better performance. This is likely due to a tendency of the two samplers to get stuck in different local modes in this particular problem, with the dd-IBP sampler delivering a better local mode. In Alzheimer’s regression, the average sampler of the dd-IBP with $\beta = 0$ had performance essentially indistinguishable from its performance with $\beta > 0$.

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 data⁶ are from a visual detection experiment in which human subjects were asked to count how many times a particular image appeared on the screen (Hoffmann et al., 2008). The data were collected as part of a larger effort to design brain-computer interfaces to assist physically disabled subjects.

Distance between datapoints was defined using the absolute time-difference. Data were z-scored prior to analysis. For 10 of the datapoints, 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. 29) conditional on the current values of the latent features and hyperparameters. We then used the MAP sample for reconstruction. Performance was measured by the squared reconstruction error on the missing data. Figure 8 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.

⁶<http://mmspl.epfl.ch/page33712.html>

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 proper 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 (Thibaux and Jordan, 2007), but this result does not generalize to the dd-IBP due to its non-exchangeability (a consequence of de Finetti’s theorem). Nonetheless, this lacuna 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 (2009a) 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 (Meeds et al., 2007; Miller et al., 2009) or text data (Thibaux and Jordan, 2007). As pointed out by Miller et al. (2009), covariates like age or location often play an important role in link prediction. Whereas Miller et al. (2009) incorporated covariates into the likelihood function, one could instead incorporate them into the prior by defining covariate-based distances between datapoints (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 fruitfully using hierarchical models (Beckmann et al., 2003; Woolrich et al., 2004). One way to extend the dd-IBP hierarchically would be to allow the parameters of the distance function to vary across individuals while being coupled together by higher-level variables.

Acknowledgements

SJG was supported by a NSF graduate research fellowship. We thank Matt Hoffman, Chong Wang, Gungor Polatkan, Sean Gerrish and John Paisley for helpful discussions.

References

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43.

- Beckmann, C., Jenkinson, M., and Smith, S. (2003). General multilevel linear modeling for group analysis in fMRI. *NeuroImage*, 20(2):1052–1063.
- Bernardo, J. and Smith, A. (1994). *Bayesian Theory*. John Wiley & Son Ltd.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Blei, D. and Frazier, P. (2010). Distance dependent Chinese restaurant processes. *International Conference on Machine Learning*.
- Caron, F., Davy, M., and Doucet, A. (2007). Generalized Polya urn for time-varying Dirichlet process mixtures. In *Uncertainty in Artificial Intelligence*.
- Christou, N. and Dinov, I. (2011). Confidence interval based parameter estimation a new socr applet and activity. *PloS One*, 6(5):e19178.
- Doshi-Velez, F. and Ghahramani, Z. (2009a). Accelerated sampling for the Indian buffet process. In *International Conference on Machine Learning*, pages 273–280.
- Doshi-Velez, F. and Ghahramani, Z. (2009b). Correlated non-parametric latent feature models. *Uncertainty in Artificial Intelligence*.
- Duan, J., Guindani, M., and Gelfand, A. (2007). Generalized spatial Dirichlet process models. *Biometrika*, 94(4):809–825.
- Erkinjuntti, T., Laaksonen, R., Sulkava, R., Syrjäläinen, R., and Palo, J. (1986). Neuropsychological differentiation between normal aging, alzheimer’s disease and vascular dementia. *Acta Neurologica Scandinavica*, 74(5):393–403.
- Escobar, M. and West, M. (1995). Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430):577–588.
- Griffin, J. and Steel, M. (2006). Order-based dependent Dirichlet processes. *Journal of the American statistical Association*, 101(473):179–194.
- Griffiths, T. and Ghahramani, Z. (2005). Infinite latent feature models and the Indian buffet process. *Advances in Neural Information Processing Systems*, 18.
- Griffiths, T. and Ghahramani, Z. (2011). The Indian Buffet Process: An Introduction and Review. *Journal of Machine Learning Research*, 12:1185–1224.
- Hoffmann, U., Vesin, J., Ebrahimi, T., and Diserens, K. (2008). An efficient P300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods*, 167(1):115–125.
- Knowles, D. and Ghahramani, Z. (2007). Infinite sparse factor analysis and infinite independent components analysis. *Independent Component Analysis and Signal Separation*, pages 381–388.
- Meeds, E., Ghahramani, Z., Neal, R., and Roweis, S. (2007). Modeling dyadic data with binary latent factors. *Advances in Neural Information Processing Systems*, 19.
- Miller, K., Griffiths, T., and Jordan, M. (2008). The phylogenetic indian buffet process: A non-exchangeable nonparametric prior for latent features.

- Miller, K., Griffiths, T., and Jordan, M. (2009). Nonparametric latent feature models for link prediction. *Advances in Neural Information Processing Systems*.
- Morris, J. (1997). Clinical dementia rating: a reliable and valid diagnostic and staging measure for dementia of the alzheimer type. *International Psychogeriatrics*, 9(S1):173–176.
- Navarro, D. and Griffiths, T. (2008). Latent features in similarity judgments: A nonparametric Bayesian approach. *Neural computation*, 20(11):2597–2628.
- Rasmussen, C. (2000). The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 12:554–560.
- Rasmussen, C. and Ghahramani, Z. (2002). Infinite mixtures of gaussian process experts. In *Advances in Neural Information Processing Systems 14*, pages 881–888.
- Thibaux, R. and Jordan, M. (2007). Hierarchical beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, volume 11, pages 564–571. Citeseer.
- Williamson, S., Orbanz, P., and Ghahramani, Z. (2010). Dependent Indian buffet processes. *International Conference on Artificial Intelligence and Statistics*.
- Woolrich, M., Behrens, T., Beckmann, C., Jenkinson, M., and Smith, S. (2004). Multilevel linear modelling for fMRI group analysis using Bayesian inference. *NeuroImage*, 21(4):1732–1747.
- Zhou, M., Yang, H., Sapiro, G., Dunson, D., and Carin, L. (2011). Dependent hierarchical beta process for image interpolation and denoising. *International Conference on Artificial Intelligence and Statistics*.

Appendix: proofs

Proposition 1

Let R_i denote the number of features held by X_i and R_{ij} denote the number of features shared by X_i and X_j , with $i \neq j$. Then for the dd-IBP,

$$R_i \sim \text{Poisson} \left(\alpha \sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1) \right) \quad (30)$$

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

Proof. For each feature, there is some probability π_i that it is turned on for datapoint 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 datapoints 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 datapoint i is distributed according to $R_i \sim \text{Poisson}(\lambda\pi_i)$. Similarly, the total number of features shared by datapoints i and j is $R_{ij} \sim \text{Poisson}(\lambda\pi_{ij})$. The activation and co-activation probabilities are given by:

$$\pi_i = \sum_{n=1}^N P(c_k^* = n) P(\mathcal{L}_{in} = 1) = \frac{\sum_{n=1}^N h_n^{-1} P(\mathcal{L}_{in} = 1)}{\sum_{j=1}^N h_j^{-1}}, \quad (32)$$

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

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

Proposition 2

Let R_i denote the number of features held by X_i and R_{ij} denote the number of features shared by X_i and X_j , with $i \neq j$. If B_0 is continuous, then for the dHBP,

$$R_i | \mathbf{g}_{1:N} \sim \text{Poisson}(\gamma) \quad (34)$$

$$R_{ij} | \mathbf{g}_{1:N} \sim \begin{cases} \text{Poisson} \left(\gamma \frac{c_0 + c_1 + 1}{(c_0 + 1)(c_1 + 1)} \right) & \text{if } g_i = g_j \\ \text{Poisson} \left(\gamma \frac{1}{c_0 + 1} \right) & \text{if } g_i \neq g_j. \end{cases} \quad (35)$$

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.

$$B = \sum_{k=1}^{\infty} p_k \delta_{\omega_k}, \quad p_k \sim \text{Beta}(0, c_0), \quad \omega_k \sim B_0. \quad (36)$$

$$B_j^* = \sum_{k=1}^{\infty} p_{jk}^* \delta_{\omega_k}, \quad p_{jk}^* \sim \text{Beta}(c_1 p_k, c_1(1 - p_k)). \quad (37)$$

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

$$R_i = \sum_{k=1}^K z_{ik} \quad \text{and} \quad R_{ij} = \sum_{k=1}^K z_{ik} z_{jk}. \quad (38)$$

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:

$$P(p_k \in dp | p_k > \epsilon) = \frac{c_0 p^{-1} (1-p)^{c_0-1}}{\int_{\epsilon}^1 c_0 u^{-1} (1-u)^{c_0-1} du} dp, \quad p \in (\epsilon, 1). \quad (39)$$

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

$$q_i(\epsilon) = \mathbb{E}[z_{ik} | p_k > \epsilon] = \mathbb{E}[p_{g_i k}^* | p_k > \epsilon] = \mathbb{E}[p_k | p_k > \epsilon] = \frac{\int_{\epsilon}^1 p c_0 p^{-1} (1-p)^{c_0-1} dp}{\int_{\epsilon}^1 c_0 p^{-1} (1-p)^{c_0-1} dp}, \quad (40)$$

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

$$R_i(\epsilon) \sim \text{Poisson}(\lambda_{\epsilon} q_i(\epsilon)). \quad (41)$$

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,

$$q_{ij}(\epsilon) = P(z_{ik} = z_{ij} = 1 | g_i, g_j, p_k > \epsilon). \quad (42)$$

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

$$R_{ij}(\epsilon) | g_i, g_j \sim \text{Poisson}(\lambda_\epsilon q_{ij}(\epsilon)). \quad (43)$$

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:** $g_i = g_j$

$$\begin{aligned} q_{ij}(\epsilon) &= \mathbb{E}[z_{ik} z_{jk} | p_k > \epsilon, g_i, g_j] = \mathbb{E}[(p_{g_i k}^*)^2 | p_k > \epsilon, g_i, g_j] \\ &= \mathbb{E}[\mathbb{E}[(p_{g_i k}^*)^2 | p_k, g_i, g_j] | p_k > \epsilon, g_i, g_j] = \mathbb{E}[p_k(c_1 p_k + 1)/(c_1 + 1) | p_k > \epsilon, g_i, g_j] \\ &= \frac{\int_\epsilon^1 \frac{c_1 p + 1}{c_1 + 1} p c_0 p^{-1} (1-p)^{c_0-1} dp}{\int_\epsilon^1 c_0 p^{-1} (1-p)^{c_0-1} dp} = \gamma \frac{c_0}{c_1 + 1} \frac{\int_\epsilon^1 (c_1 p + 1)(1-p)^{c_0-1} dp}{\lambda_\epsilon}. \end{aligned} \quad (44)$$

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 $\int_0^1 p(1-p)^{c_0-1} dp = \frac{1}{c_0(c_0+1)}$.

