The Indian Buffet Process: An Introduction and Review

T. L. Griffiths, and Z. Ghahramani.

Andrea Teruzzi

November 25, 2022

Table of Contents

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment
  - Number of features

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment
  - Number of features
- The alternative is to assume that the amount of latent structure is **unbounded**

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment
  - Number of features
- The alternative is to assume that the amount of latent structure is **unbounded**
  - **Bayesian non-parametric** (BNP) methods are extremely suited for this scope

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment
  - Number of features
- The alternative is to assume that the amount of latent structure is **unbounded**
  - **Bayesian non-parametric** (BNP) methods are extremely suited for this scope
  - The celebrated **Dirichlet process mixture** (DPM) , is a good example of unbounded number of latent components.

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment
  - Number of features
- The alternative is to assume that the amount of latent structure is **unbounded**
  - **Bayesian non-parametric** (BNP) methods are extremely suited for this scope
  - The celebrated **Dirichlet process mixture** (DPM) , is a good example of unbounded number of latent components.
- In DPM **each datapoint is assigned to latent class** and each class is associated with a distribution. The particular feature of the model is that the prior responsible of assigning observations to latent class is bounded only by the number of objects, making DPM models **"infinite" mixture models**.

Introduction - Latent Structure Problem

- One key goal of unsupervised learning is to determine the amount of **latent structure** associated to each data object.
  - Cluster assignment
  - Number of features
- The alternative is to assume that the amount of latent structure is **unbounded**
  - **Bayesian non-parametric** (BNP) methods are extremely suited for this scope
  - The celebrated **Dirichlet process mixture** (DPM) , is a good example of unbounded number of latent components.
- In DPM **each datapoint is assigned to latent class** and each class is associated with a distribution. The particular feature of the model is that the prior responsible of assigning observations to latent class is bounded only by the number of objects, making DPM models **"infinite" mixture models**.
  - Generative process: **Chinese restaurant process**

Introduction – Beyond the DPM

- DPM has been subject to several extensions, but all of this models associate each object with **one latent variable** that assigns the object to one class or parameter determining its probability law.

Introduction – Beyond the DPM

- DPM has been subject to several extensions, but all of this models associate each object with **one latent variable** that assigns the object to one class or parameter determining its probability law.
  - That is not always the case! As each object can be produced by **multiple (unknown) number of causes** and presenting multiple feature.

Introduction – Beyond the DPM

- DPM has been subject to several extensions, but all of this models associate each object with **one latent variable** that assigns the object to one class or parameter determining its probability law.
  - That is not always the case! As each object can be produced by **multiple (unknown) number of causes** and presenting multiple feature.
- We can **represent each object with a binary vector**, with entries indicating the presence or absence of each feature.

Introduction – Beyond the DPM

- DPM has been subject to several extensions, but all of this models associate each object with **one latent variable** that assigns the object to one class or parameter determining its probability law.
  - That is not always the case! As each object can be produced by **multiple (unknown) number of causes** and presenting multiple feature.
- We can **represent each object with a binary vector**, with entries indicating the presence or absence of each feature.
  - We would like to not put an upper bound to the number of features. Dirichlet process are not suited for this goal.

Introduction – Beyond the DPM

- DPM has been subject to several extensions, but all of this models associate each object with **one latent variable** that assigns the object to one class or parameter determining its probability law.
  - That is not always the case! As each object can be produced by **multiple (unknown) number of causes** and presenting multiple feature.
- We can **represent each object with a binary vector**, with entries indicating the presence or absence of each feature.
  - We would like to not put an upper bound to the number of features. Dirichlet process are not suited for this goal.
- The objective of the paper is presenting a non-parametric approach to models in which objects are represented using an unknown number of latent feature.

Introduction – Beyond the DPM

- DPM has been subject to several extensions, but all of this models associate each object with **one latent variable** that assigns the object to one class or parameter determining its probability law.
  - That is not always the case! As each object can be produced by **multiple (unknown) number of causes** and presenting multiple feature.
- We can **represent each object with a binary vector**, with entries indicating the presence or absence of each feature.
  - We would like to not put an upper bound to the number of features. Dirichlet process are not suited for this goal.
- The objective of the paper is presenting a non-parametric approach to models in which objects are represented using an unknown number of latent feature.
  - Generative process: **Indian buffet process**

Latent Class Models

- Assume we have N row vectors (objects) $\mathbf{x}_i$ each having D observable properties and the matrix $\mathbf{X} = [\mathbf{x}_1^\top \mathbf{x}_2^\top \dots \mathbf{x}_N^\top]$ to indicate the properties of all the objects.

Latent Class Models

- Assume we have N row vectors (objects) $\mathbf{x}_i$ each having D observable properties and the matrix $\mathbf{X} = [\mathbf{x}_1^\top \mathbf{x}_2^\top \dots \mathbf{x}_N^\top]$ to indicate the properties of all the objects.
- We assign each $\mathbf{x}_i$ to a single class c_i and we indicate with $\mathbf{c}$ the class assignment of each object.

Latent Class Models

- Assume we have N row vectors (objects) $\mathbf{x}_i$ each having D observable properties and the matrix $\mathbf{X} = [\mathbf{x}_1^\top \mathbf{x}_2^\top \dots \mathbf{x}_N^\top]$ to indicate the properties of all the objects.
- We assign each $\mathbf{x}_i$ to a single class c_i and we indicate with $\mathbf{c}$ the class assignment of each object.
- The statistical model for a latent class model consists in specifying

$$P(\mathbf{c})$$
$$p(\mathbf{X}|\mathbf{c})$$

Finite Mixture Model

- Mixture models assume that **the assignment of an object to a class is independent of the assignments of all other objects**. In finite mixture model we have K classes.

$$P(\mathbf{c}) = \prod_{i=1}^{N} P(c_i | \theta) = \prod_{i=1}^{N} \theta_{c_i},$$

with θ multinomial distribution and θ_k the probability of class k under that distribution (i.e. $\sum_{k=1}^{K} \theta_k = 1$)

Finite Mixture Model

- Mixture models assume that **the assignment of an object to a class is independent of the assignments of all other objects**. In finite mixture model we have K classes.

$$P(\mathbf{c}) = \prod_{i=1}^{N} P(c_i | \, \theta) = \prod_{i=1}^{N} \theta_{c_i},$$

  with θ multinomial distribution and θ_k the probability of class k under that distribution (i.e. $\sum_{k=1}^{K} \theta_k = 1$)
- Under this assumption:

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} \sum_{k=1}^{K} p(\mathbf{x}_i | \, c_i = k)\theta_k,$$

  that is a mixture of the K class distributions $p(\mathbf{x}_i | \, c_i = k)$, with θ_k determining the weight of class k.

Finite Mixture Model

- In Bayesian modeling, θ is assumed to follow a prior distribution $p(\theta)$, with conjugate choice the **Dirichlet distribution over K classes** with parameters $\alpha_1, \alpha_2, \ldots, \alpha_K$.

$$p(\theta) = \frac{\prod_{k=1}^{K} \theta_k^{\alpha_k - 1}}{D(\alpha_1, \alpha_2, \ldots, \alpha_K)},$$

with normalizing constant $D(\alpha_1, \alpha_2, \ldots, \alpha_K)$ define as

$$\begin{aligned} D(\alpha_1, \alpha_2, \ldots, \alpha_K) &= \int_{\Delta_K} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \, \mathrm{d}\theta \\ &= \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}, \end{aligned}$$

where Δ_K is the simplex of multinomials over K classes and $\Gamma(m) = (m-1)!$ is the gamma function.

Finite Mixture Model

- Using a symmetric Dirichlet (i.e. $\alpha_k = \frac{\alpha}{K}$ for all k):

$$\theta \,|\, \alpha \sim \text{Dirichlet}\left(\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$$
$$c_i|\theta \sim \text{Discrete}(\theta),$$

where Discrete(θ) is the multiple-outcome analogue of a Bernoulli event, where the probabilities of the outcomes are specified by θ.

Finite Mixture Model

- Using a symmetric Dirichlet (i.e. $\alpha_k = \frac{\alpha}{K}$ for all k):

$$\theta | \, \alpha \sim \text{Dirichlet}\left(\frac{\alpha}{K}, \frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$$
$$c_i | \theta \sim \text{Discrete}(\theta),$$

where Discrete(θ) is the multiple-outcome analogue of a Bernoulli event, where the probabilities of the outcomes are specified by θ.

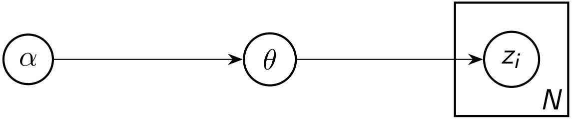

Figure 1: Graphical model for the Dirichlet-multinomial model

Finite Mixture Model

- Integrating over all values of θ the probability of an assignment vector $\mathbf{c}$ is:

$$P(\mathbf{c}) = \int_{\Delta_K} \prod_{i=1}^{n} P(c_i|\theta)p(\theta)\,\mathrm{d}\theta = \int_{\Delta_K} \frac{\prod_{k=1}^{K} \theta_k^{m_k+\alpha/K-1}}{D(\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K})}\,\mathrm{d}\theta$$
$$= \frac{D(m_1 + \frac{\alpha}{K}, m_2 + \frac{\alpha}{K}, \dots, m_k + \frac{\alpha}{K})}{D(\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K})}$$
$$= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}, \tag{1}$$

where $m_k = \sum_{i=1}^{N} \delta(c_i = k)$ is the number of objects assigned to class k.

Finite Mixture Model

- Integrating over all values of θ the probability of an assignment vector $\mathbf{c}$ is:

$$P(\mathbf{c}) = \int_{\Delta_K} \prod_{i=1}^{n} P(c_i|\theta)p(\theta)\,\mathrm{d}\theta = \int_{\Delta_K} \frac{\prod_{k=1}^{K} \theta_k^{m_k+\alpha/K-1}}{D(\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K})}\,\mathrm{d}\theta$$
$$= \frac{D(m_1 + \frac{\alpha}{K}, m_2 + \frac{\alpha}{K}, \dots, m_k + \frac{\alpha}{K})}{D(\frac{\alpha}{K}, \frac{\alpha}{K}, \dots, \frac{\alpha}{K})}$$
$$= \frac{\prod_{k=1}^{K} \Gamma(m_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}, \tag{1}$$

where $m_k = \sum_{i=1}^{N} \delta(c_i = k)$ is the number of objects assigned to class k.

- Class assignments $\mathbf{c}$ are not independent, rather they are **exchangeable**.

Finite Mixture Model

- Integrating over all values of θ the probability of an assignment vector $\mathbf{c}$ is:

$$
\begin{aligned}
P(\mathbf{c}) = \int_{\Delta_K} \prod_{i=1}^{n} P(c_i|\theta)p(\theta)\,\mathrm{d}\theta &= \int_{\Delta_K} \frac{\prod_{k=1}^{K} \theta_k^{m_k+\alpha/K-1}}{D(\frac{\alpha}{K},\frac{\alpha}{K},\ldots,\frac{\alpha}{K})}\,\mathrm{d}\theta \\
&= \frac{D(m_1+\frac{\alpha}{K}, m_2+\frac{\alpha}{K},\ldots,m_k+\frac{\alpha}{K})}{D(\frac{\alpha}{K},\frac{\alpha}{K},\ldots,\frac{\alpha}{K})} \\
&= \frac{\prod_{k=1}^{K}\Gamma(m_k+\frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K}\frac{\Gamma(\alpha)}{\Gamma(N+\alpha)},
\end{aligned} \tag{1}
$$

  where $m_k = \sum_{i=1}^{N} \delta(c_i = k)$ is the number of objects assigned to class k.
- Class assignments $\mathbf{c}$ are not independent, rather they are **exchangeable**.
- Equation 1 **assumes an upper bound on the number of classes** of objects

Infinite Mixture Models

- We specify the probability of $\mathbf{X}$ for infinitely many classes

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(\mathbf{x}_i|\, c_i = k)\theta_k$$

Infinite Mixture Models

- We specify the probability of $\mathbf{X}$ for infinitely many classes

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(\mathbf{x}_i | \, c_i = k)\theta_k$$

1. Define a prior $p(\theta)$ on infinite-dimensional multinomials and compute $p(c)$ by integrating over θ

Infinite Mixture Models

- We specify the probability of $\mathbf{X}$ for infinitely many classes

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(\mathbf{x}_i | \, c_i = k)\theta_k$$

1. Define a prior $p(\theta)$ on infinite-dimensional multinomials and compute $p(c)$ by integrating over θ
2. Consider Equation 1 and take the limit for $K \to \infty$

Infinite Mixture Models

- We specify the probability of $\mathbf{X}$ for infinitely many classes

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(\mathbf{x}_i|\, c_i = k)\theta_k$$

  1. Define a prior $p(\theta)$ on infinite-dimensional multinomials and compute $p(c)$ by integrating over θ
  2. Consider Equation 1 and take the limit for $K \to \infty$

- We rearrange Equation 1 considering the recursion property of $\Gamma(x)$:

$$P(\mathbf{c}) = \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K_+} \prod_{j=1}^{m_k - 1} \left(j + \frac{\alpha}{K}\right)\right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}, \tag{2}$$

with K_+ is the number of classes for which $m_k > 0$ for the ordered sequence of indices such that $m_k > 0$ for all $k < K_+$.

Infinite Mixture Models

- We specify the probability of $\mathbf{X}$ for infinitely many classes

$$p(\mathbf{X}|\mathbf{c}) = \prod_{i=1}^{N} \sum_{k=1}^{\infty} p(\mathbf{x}_i|\, c_i = k)\theta_k$$

  1. Define a prior $p(\theta)$ on infinite-dimensional multinomials and compute $p(c)$ by integrating over θ
  2. Consider Equation 1 and take the limit for $K \to \infty$

- We rearrange Equation 1 considering the recursion property of $\Gamma(x)$:

$$P(\mathbf{c}) = \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K_+} \prod_{j=1}^{m_k - 1} \left(j + \frac{\alpha}{K}\right)\right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}, \tag{2}$$

  with K_+ is the number of classes for which $m_k > 0$ for the ordered sequence of indices such that $m_k > 0$ for all $k < K_+$.

- Since there are K^N possible configurations for $\mathbf{c}$, $P(\mathbf{c}) \to 0$ as $K \to \infty$.

Distribution over partitions

Definition

A **partition** is a division of the set of N objects into subsets, where each object belongs to a single subset and the ordering of the subsets does not matter.

Distribution over partitions

Definition

A **partition** is a division of the set of N objects into subsets, where each object belongs to a single subset and the ordering of the subsets does not matter.

$$e.g. \quad \{c_1, c_2, c_3\} = \{1, 1, 2\} \text{ is equivalent to } \{2, 2, 1\}$$

Distribution over partitions

Definition

A **partition** is a division of the set of N objects into subsets, where each object belongs to a single subset and the ordering of the subsets does not matter.

$$e.g. \quad \{c_1, c_2, c_3\} = \{1, 1, 2\} \text{ is equivalent to } \{2, 2, 1\}$$

- A partition defines an **equivalence class** of assignment vectors, which we denote $[\mathbf{c}]$

Distribution over partitions

Definition

A **partition** is a division of the set of N objects into subsets, where each object belongs to a single subset and the ordering of the subsets does not matter.

$$e.g. \quad \{c_1, c_2, c_3\} = \{1, 1, 2\} \text{ is equivalent to } \{2, 2, 1\}$$

- A partition defines an **equivalence class** of assignment vectors, which we denote $[\mathbf{c}]$
- $p(\mathbf{X}|\,\mathbf{c})$ is the same for all vectors $\mathbf{c}$ corresponding to the same partition $[\mathbf{c}]$

Distribution over partitions

Definition

A **partition** is a division of the set of N objects into subsets, where each object belongs to a single subset and the ordering of the subsets does not matter.

$$e.g. \quad \{c_1, c_2, c_3\} = \{1, 1, 2\} \text{ is equivalent to } \{2, 2, 1\}$$

- A partition defines an **equivalence class** of assignment vectors, which we denote $[\mathbf{c}]$
- $p(\mathbf{X}|\,\mathbf{c})$ is the same for all vectors $\mathbf{c}$ corresponding to the same partition $[\mathbf{c}]$
- Assume we have a partition of N objects into K_+ subsets with $K = K_0 + K_+$ classes. There are $\frac{K!}{K_0}$ assignments of vector $\mathbf{c}$ that belong to the equivalence class defined by that partition $[\mathbf{c}]$

Distribution over partitions

- The limiting probability of those class assignments is:

$$
\begin{aligned}
P([\mathbf{c}]) = \sum_{\mathbf{c}\in[\mathbf{c}]} P(\mathbf{c}) &= \lim_{K\to\infty} \frac{K!}{K_0!} P(\mathbf{c}) \\
&= \lim_{k\to\infty} \frac{K!}{K_0!} \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K_+} \prod_{j=1}^{m_k-1} \left(j + \frac{\alpha}{K}\right)\right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \\
&= \alpha^{K^+} \left(\prod_{k=1}^{K_+} \left(m_k - 1\right)!\right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)},
\end{aligned} \tag{3}
$$

that defines a **distribution over partitions** that is the prior over class assignments for an infinite mixture model.

Distribution over partitions

- The limiting probability of those class assignments is:

$$
\begin{aligned}
P([\mathbf{c}]) = \sum_{\mathbf{c}\in[\mathbf{c}]} P(\mathbf{c}) &= \lim_{K\to\infty} \frac{K!}{K_0!} P(\mathbf{c}) \\
&= \lim_{k\to\infty} \frac{K!}{K_0!} \left(\frac{\alpha}{K}\right)^{K_+} \left(\prod_{k=1}^{K_+} \prod_{j=1}^{m_k-1} \left(j + \frac{\alpha}{K}\right)\right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \\
&= \alpha^{K^+} \left(\prod_{k=1}^{K_+} \left(m_k - 1\right)! \right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}, \qquad (3)
\end{aligned}
$$

that defines a **distribution over partitions** that is the prior over class assignments for an infinite mixture model.

- Equation 3 is consistent with the other derivation, as Dirichlet Process (Blackwell and MacQueen, 1973) or stick breaking priors (Sethuraman, 1994).

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

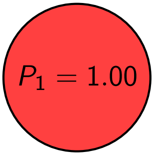

Figure 2: Example of Chinese Restaurant Process with $\alpha = 2$

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

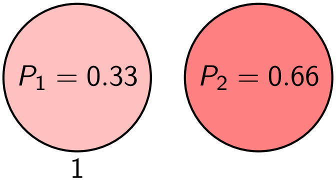

Figure 2: Example of Chinese Restaurant Process with $\alpha = 2$

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K + 1 \end{cases}$$

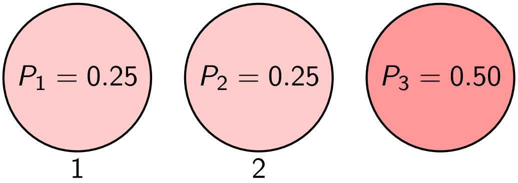

Figure 2: Example of Chinese Restaurant Process with $\alpha = 2$

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

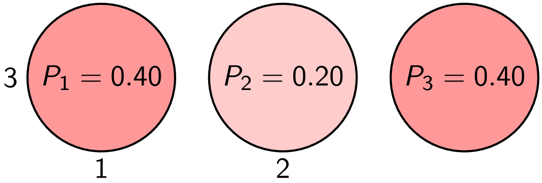

Figure 2: Example of Chinese Restaurant Process with $\alpha = 2$

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \dots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

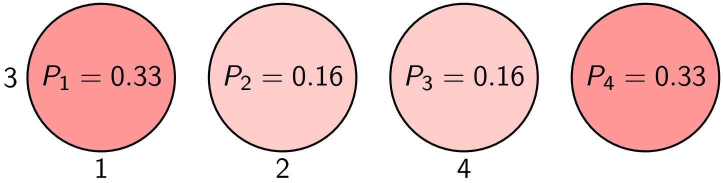

Figure 2: Example of Chinese Restaurant Process with $\alpha = 2$

The Chinese Restaurant Process

Imagine a restaurant with an infinite number of tables, each with an infinite number of seats. The ith client will seat to the kth table with probability:

$$P(c_i = k | c_1, c_2, \ldots, c_{i-1}) = \begin{cases} \frac{m_k}{i-1+\alpha} & k \leq K_+ \\ \frac{\alpha}{i-1+\alpha} & k = K+1 \end{cases}$$

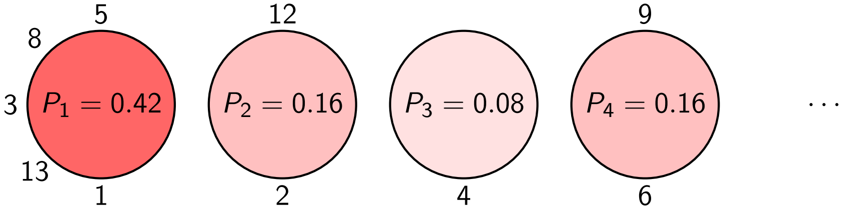

Figure 2: Example of Chinese Restaurant Process with $\alpha = 2$

Inference by Gibbs Sampling

- In a mixture model the variables to be sampled are the class assignments $\mathbf{c}$, applying Bayes' rule

$$P(c_i = k | \mathbf{c}_{-i}, \mathbf{X}) \propto p(\mathbf{X} | \mathbf{c}) P(c_i = k | \mathbf{c}_{-i})$$

Inference by Gibbs Sampling

- In a mixture model the variables to be sampled are the class assignments **c**, applying Bayes' rule

$$P(c_i = k|\,\mathbf{c}_{-i}, \mathbf{X}) \propto p(\mathbf{X}|\,\mathbf{c})P(c_i = k|\,\mathbf{c}_{-i})$$

- For a finite model with conjugate prior as define in Equation 1 we integrate over θ

$$\begin{aligned} P(c_i = k|\,\mathbf{c}_{-i}) &= \int P(c_i = k|\theta)p(\theta|\,\mathbf{c}_{-i})\,\mathrm{d}\theta \\ &= \frac{m_{-i,k} + \frac{\alpha}{K}}{i - 1 + \alpha} \quad k \leq K_+ \end{aligned}$$

  where $m_{-i,k}$ is the number of objects assigned to class k, excluding i.

Inference by Gibbs Sampling

- In a mixture model the variables to be sampled are the class assignments $\mathbf{c}$, applying Bayes' rule

$$P(c_i = k \mid \mathbf{c}_{-i}, \mathbf{X}) \propto p(\mathbf{X} \mid \mathbf{c}) P(c_i = k \mid \mathbf{c}_{-i})$$

- For a finite model with conjugate prior as define in Equation 1 we integrate over θ

$$\begin{aligned} P(c_i = k \mid \mathbf{c}_{-i}) &= \int P(c_i = k \mid \theta) p(\theta \mid \mathbf{c}_{-i}) \, \mathrm{d}\theta \\ &= \frac{m_{-i,k} + \frac{\alpha}{K}}{i - 1 + \alpha} \quad k \leq K_+ \end{aligned}$$

  where $m_{-i,k}$ is the number of objects assigned to class k, excluding i.
- For the infinite case we can use **exchangeability** and choose an ordering in which the ith object is the last to be assigned to a class. **We sample directly from the CRP**

$$P(c_i = k \mid \mathbf{c}_{-i}) = \begin{cases} \frac{m_{-i,k} + \frac{\alpha}{K}}{i-1+\alpha} & m_{-i,k} > 0 \\ \frac{\alpha}{i-1+\alpha} & k = K_{-i,+} + 1 \\ 0 & \text{otherwise} \end{cases}$$

Latent Feature Model

- Assume we have N objects and K features and the possession of feature k by object i is indicated by a binary variable z_{ik} which forms a binary $N \times K$ feature matrix $\mathbf{Z}$

Latent Feature Model

- Assume we have N objects and K features and the possession of feature k by object i is indicated by a binary variable z_{ik} which forms a binary $N \times K$ feature matrix $\mathbf{Z}$
- We assume that each object possesses feature k a probability π_k and that the features are **generated independently**, forming $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_K\}$,.

$$\underline{\text{Latent Feature Model}}\text{:}\ \theta \in [0,1] \quad \text{with} \ \sum_{k=1}^{K} \theta_k = 1$$

$$\underline{\text{Latent Class Model}}\text{:}\ \pi_k \in [0,1]$$

Latent Feature Model

- Assume we have N objects and K features and the possession of feature k by object i is indicated by a binary variable z_{ik} which forms a binary $N \times K$ feature matrix $\mathbf{Z}$
- We assume that each object possesses feature k a probability π_k and that the features are **generated independently**, forming $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots, \pi_K\},$.

$$\underline{\text{Latent Feature Model}}: \ \theta \in [0,1] \quad \text{with} \ \sum_{k=1}^{K} \theta_k = 1$$

$$\underline{\text{Latent Class Model}}: \ \pi_k \in [0,1]$$

- The statistical model for a latent class model consists in

$$P(\boldsymbol{\pi})$$
$$p(\mathbf{Z}|\boldsymbol{\pi})$$

Finite Feature Model

- The probability of a matrix $\mathbf{Z}$ given $\boldsymbol{\pi}$ is:

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

where $m_k = \sum_{i=1}^{N} z_{ik}$

Finite Feature Model

- The probability of a matrix $\mathbf{Z}$ given $\boldsymbol{\pi}$ is:

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

  where $m_k = \sum_{i=1}^{N} z_{ik}$
- We assume each π_k follows a Beta(r,s), which is conjugate to the binomial

$$p(\pi_k) = \frac{\pi_k^{r-1}(1-\pi_k)^{s-1}}{B(r,s)},$$

  where $B(r,s)$ is the beta function

$$\begin{aligned} B(r,s) &= \int_0^1 \pi_k^{r-1}(1-\pi_k)^{s-1}\,\mathrm{d}\pi_k \\ &= \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)} \end{aligned}$$

Finite Feature Model

- We take $r = \frac{\alpha}{K}$ and $s = 1$:

$$\pi_k | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right)$$
$$z_{ik} | \pi_k \sim \text{Bernoulli}(\pi_k)$$

Each z_{ik} is **independent of all other assignments conditioned on** π_k, and the π_k are **generated independently**.

Finite Feature Model

- We take $r = \frac{\alpha}{K}$ and $s = 1$:

$$\pi_k | \alpha \sim \text{Beta}\left(\frac{\alpha}{K}, 1\right)$$
$$z_{ik} | \pi_k \sim \text{Bernoulli}(\pi_k)$$

Each z_{ik} is **independent of all other assignments conditioned on** π_k, and the π_k are **generated independently**.

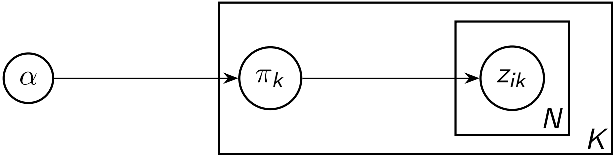

Figure 3: Graphical model for the beta-binomial model

Finite Feature Model

- Integrating over all values of $\boldsymbol{\pi}$, the probability of a binary matrix $\mathbf{Z}$ is

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

Finite Feature Model

- Integrating over all values of $\boldsymbol{\pi}$, the probability of a binary matrix $\mathbf{Z}$ is

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

the result follows from the beta-binomial conjugacy and the distribution is **exchangeable**, depending only on the counts m_k.

Equivalence Classes: Left-Ordered Matrices

- In order to use the same approach as before and letting $\rightarrow \infty$ we need to define **equivalence classes of binary matrices**.

Equivalence Classes: Left-Ordered Matrices

- In order to use the same approach as before and letting $\rightarrow \infty$ we need to define **equivalence classes of binary matrices**.
- Let $lof(\cdot)$ a **many-to-one function ordering the columns of the binary matrix Z** from left to right by the magnitude of the binary number expressed by that column.

Equivalence Classes: Left-Ordered Matrices

- In order to use the same approach as before and letting $\to \infty$ we need to define **equivalence classes of binary matrices**.
- Let $lof(\cdot)$ a **many-to-one function ordering the columns of the binary matrix $\mathbf{Z}$** from left to right by the magnitude of the binary number expressed by that column.
- We denote by $[\mathbf{Z}]$ the lof-equivalence class of a binary matrix $\mathbf{Z}$, i.e.

$$[\mathbf{Z}] = \{\mathbf{Y} : lof(\mathbf{Y}) = lof(\mathbf{Z})\}.$$

Equivalence Classes: Left-Ordered Matrices

- In order to use the same approach as before and letting $\to \infty$ we need to define **equivalence classes of binary matrices**.
- Let $lof(\cdot)$ a **many-to-one function ordering the columns of the binary matrix $\mathbf{Z}$** from left to right by the magnitude of the binary number expressed by that column.
- We denote by $[\mathbf{Z}]$ the *lof*-equivalence class of a binary matrix $\mathbf{Z}$, i.e.

$$[\mathbf{Z}] = \{\mathbf{Y} : lof(\mathbf{Y}) = lof(\mathbf{Z})\}.$$

Equivalence Classes: Left-Ordered Matrices

- In order to use the same approach as before and letting $\to \infty$ we need to define **equivalence classes of binary matrices**.
- Let $lof(\cdot)$ a **many-to-one function ordering the columns of the binary matrix $\mathbf{Z}$** from left to right by the magnitude of the binary number expressed by that column.
- We denote by $[\mathbf{Z}]$ the lof-equivalence class of a binary matrix $\mathbf{Z}$, i.e.

$$[\mathbf{Z}] = \{\mathbf{Y} : lof(\mathbf{Y}) = lof(\mathbf{Z})\}.$$

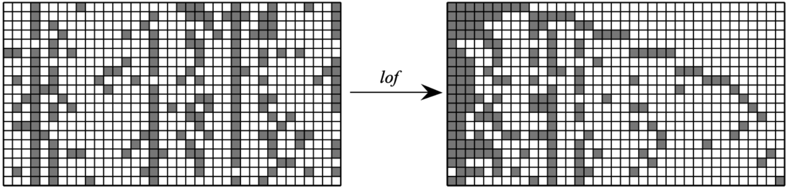

Figure 4: $\mathbf{Z}$ and $lof(\mathbf{Z})$

Infinite Feature Model

- From Equation 3, the probability of a particular *lof*-equivalence class is

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

Infinite Feature Model

- From Equation 3, the probability of a particular *lof*-equivalence class is

$$P([\mathbf{Z}]) = \sum_{\mathbf{Z}\in[\mathbf{Z}]} P(\mathbf{Z})$$
$$= \frac{K!}{\prod_{h=0} 2^N - 1 K_h!} P(\mathbf{Z}), \quad (5)$$

- Substituting and rearranging Equation 4 in Equation 5 and letting $K \to \infty$

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

where $H_N = \sum_{j=1}^{N} \frac{1}{j}$.

Infinite Feature Model

- From Equation 3, the probability of a particular *lof*-equivalence class is

$$P([\mathbf{Z}]) = \sum_{\mathbf{Z}\in[\mathbf{Z}]} P(\mathbf{Z})$$
$$= \frac{K!}{\prod_{h=0} 2^N - 1 K_h!} P(\mathbf{Z}), \tag{5}$$

- Substituting and rearranging Equation 4 in Equation 5 and letting $K \to \infty$

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

where $H_N = \sum_{j=1}^{N} \frac{1}{j}$.

- The distribution is **exchangeable** and it is coherent with the prior distribution defined by Hjort (1990) and the stick breaking construction suggested in Teh et al. (2007).

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

$\alpha = 4.0$

													$\cdots$
.50	.50	.50											

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

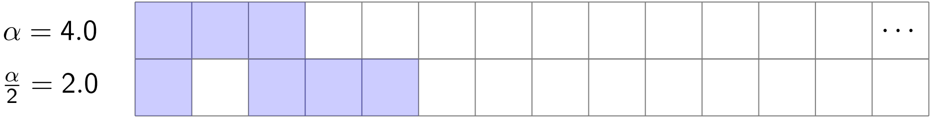

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

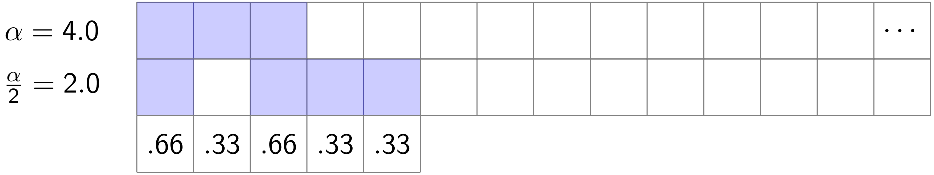

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

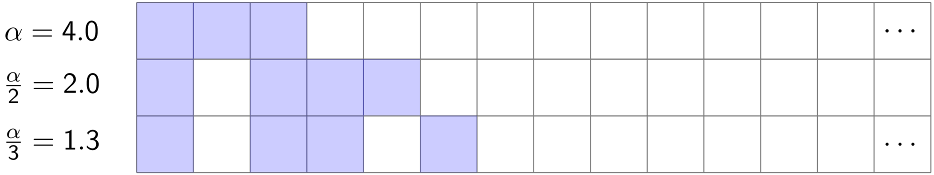

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

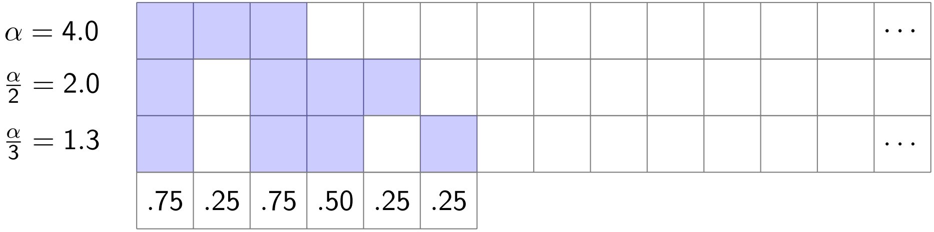

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

The Indian Buffet Process

- The first customer starts at the left of the buffet and takes Poisson(α) number of dishes.
- Customer ith moves along the buffet and takes dish k with probability $\frac{m_k}{i}$ and then tries Poisson($\frac{\alpha}{i}$) number of new dishes.

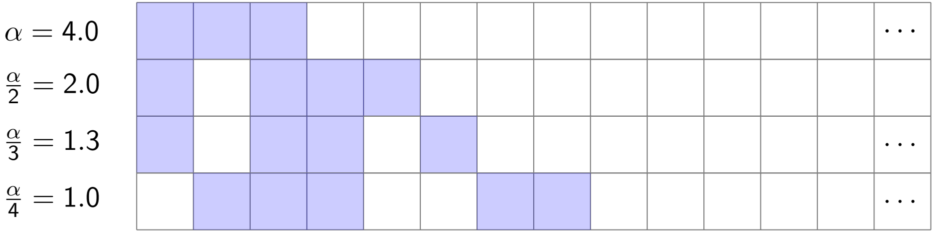

Figure 5: Example of Indian Buffet Process with $\alpha = 4$

IBP Properties

- The previous IBP is not exchangeable! The number of new dishes depends on i.

IBP Properties

- The previous IBP is not exchangeable! The number of new dishes depends on i.
- **Exchangeable IBP** is a generative process for distributions over collections of histories equivalent to $P([\mathbf{Z}])$ in Equation 6.

IBP Properties

- The previous IBP is not exchangeable! The number of new dishes depends on i.
- **Exchangeable IBP** is a generative process for distributions over collections of histories equivalent to $P([\mathbf{Z}])$ in Equation 6.
- History of feature k at object i is defined to be $(z_{1k}, \ldots, z_{(i-1)k})$.

$$
\begin{aligned}
e.g. \quad (z_{1k}, z_{1k}) &= (0, 0) \\
(z_{1k}, z_{2k}) &= (1, 0) \\
(z_{1k}, z_{2k}) &= (0, 1) \\
(z_{1k}, z_{2k}) &= (1, 1)
\end{aligned}
$$

IBP Properties

- The previous IBP is not exchangeable! The number of new dishes depends on i.
- **Exchangeable IBP** is a generative process for distributions over collections of histories equivalent to $P([\mathbf{Z}])$ in Equation 6.
- History of feature k at object i is defined to be $(z_{1k}, \ldots, z_{(i-1)k})$.

$$
\begin{aligned}
e.g. \quad (z_{1k}, z_{1k}) &= (0,0) \\
(z_{1k}, z_{2k}) &= (1,0) \\
(z_{1k}, z_{2k}) &= (0,1) \\
(z_{1k}, z_{2k}) &= (1,1)
\end{aligned}
$$

1. The effective dimension, K_+, of the model follows a Poisson(αH_N) distribution.

IBP Properties

- The previous IBP is not exchangeable! The number of new dishes depends on i.
- **Exchangeable IBP** is a generative process for distributions over collections of histories equivalent to $P([\mathbf{Z}])$ in Equation 6.
- History of feature k at object i is defined to be $(z_{1k}, \dots, z_{(i-1)k})$.

$$\begin{aligned} e.g. \quad (z_{1k}, z_{1k}) &= (0,0) \\ (z_{1k}, z_{2k}) &= (1,0) \\ (z_{1k}, z_{2k}) &= (0,1) \\ (z_{1k}, z_{2k}) &= (1,1) \end{aligned}$$

1. The effective dimension, K_+, of the model follows a Poisson(αH_N) distribution.
2. The number of features possessed by each object follows a Poisson(α) distribution.

IBP Properties

- The previous IBP is not exchangeable! The number of new dishes depends on i.
- **Exchangeable IBP** is a generative process for distributions over collections of histories equivalent to $P([\mathbf{Z}])$ in Equation 6.
- History of feature k at object i is defined to be $(z_{1k}, \ldots, z_{(i-1)k})$.

$$
\begin{aligned}
e.g. \quad (z_{1k}, z_{1k}) &= (0,0) \\
(z_{1k}, z_{2k}) &= (1,0) \\
(z_{1k}, z_{2k}) &= (0,1) \\
(z_{1k}, z_{2k}) &= (1,1)
\end{aligned}
$$

1. The effective dimension, K_+, of the model follows a Poisson(αH_N) distribution.
2. The number of features possessed by each object follows a Poisson(α) distribution.
3. The expected number of entries in $\mathbf{Z}$ is $N\alpha$, so $\mathbf{Z}$ remains sparse as $K \to \infty$.

Inference by Gibbs Sampling

- To sample from the distribution defined by the IBP we need to compute the full conditional $P(z_{ik} = 1 | \mathbf{Z}_{-(ik)})$, where $\mathbf{Z}_{-(ik)}$ denotes the entries of $\mathbf{Z}$ other than z_{ik}.

Inference by Gibbs Sampling

- To sample from the distribution defined by the IBP we need to compute the full conditional $P(z_{ik} = 1 | \mathbf{Z}_{-(ik)})$, where $\mathbf{Z}_{-(ik)}$ denotes the entries of $\mathbf{Z}$ other than z_{ik}.
- In the finite model, we use Equation 4 for $P(\mathbf{Z})$ to obtain the conditional distribution for any z_{ik}. Integrating out π_k

$$
\begin{aligned}
P(z_{ik} = 1 | \mathbf{z}_{-i,k}) &= \int_0^1 P(z_{ik}|\pi_k) p(\pi_k|\mathbf{z}_{-i,k}) \, \mathrm{d}\pi_k \\
&= \frac{m_{-i,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}
\end{aligned}
$$

where $m_{-i,k}$ is the number of object with features k, not including i.

Inference by Gibbs Sampling

- To sample from the distribution defined by the IBP we need to compute the full conditional $P(z_{ik} = 1 | \mathbf{Z}_{-(ik)})$, where $\mathbf{Z}_{-(ik)}$ denotes the entries of $\mathbf{Z}$ other than z_{ik}.
- In the finite model, we use Equation 4 for $P(\mathbf{Z})$ to obtain the conditional distribution for any z_{ik}. Integrating out π_k

$$
\begin{aligned}
P(z_{ik} = 1 | \mathbf{z}_{-i,k}) &= \int_0^1 P(z_{ik}|\pi_k) p(\pi_k|\mathbf{z}_{-i,k}) \, \mathrm{d}\pi_k \\
&= \frac{m_{-i,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}
\end{aligned}
$$

  where $m_{-i,k}$ is the number of object with features k, not including i.
- For the infinite case, **we arbitrary choose object ith to be the last one to visit the buffet**

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

  that is also the limit of the full conditional for the finite case.

IBP sampling algorithm

Algorithm 1 Gibbs sampler for IBP

Initialize binary matrix $\mathbf{Z}$
for $i = 1$ to N **do**
    for $k = i$ to K **do**
        if $m_{-i,k} > 0$ **then**
            $z_{ik} \sim P(z_{ik} = 1 | \mathbf{z}_{-i,k})$
        else
            Delete column k
        end if
        Add Poisson($\frac{\alpha}{N}$) new columns
    end for
end for

Two-Parameter IBP

- IBP has only one parameter α which controls both the **sparsity** of $\mathbf{Z}$ and its **dimensionality**.

Two-Parameter IBP

- IBP has only one parameter α which controls both the **sparsity** of $\mathbf{Z}$ and its **dimensionality**.
- Ghahramani et al. (2007) introduced a a **two-parameter generalization of the IBP, with**

$$\pi_z | \, \alpha, \beta \sim \text{Beta}\left(\frac{\alpha\beta}{K}, \beta\right)$$

Two-Parameter IBP

- IBP has only one parameter α which controls both the **sparsity** of **Z** and its **dimensionality**.
- Ghahramani et al. (2007) introduced a a **two-parameter generalization of the IBP, with**

$$\pi_z | \alpha, \beta \sim \text{Beta}\left(\frac{\alpha\beta}{K}, \beta\right)$$

- The generative process

Two-Parameter IBP

- IBP has only one parameter α which controls both the **sparsity** of $\mathbf{Z}$ and its **dimensionality**.
- Ghahramani et al. (2007) introduced a a **two-parameter generalization of the IBP, with**

$$\pi_z | \alpha, \beta \sim \mathrm{Beta}\left(\frac{\alpha\beta}{K}, \beta\right)$$

- The generative process

1. The first customer starts at the left of the buffet and samples Poisson(α) dishes.

Two-Parameter IBP

- IBP has only one parameter α which controls both the **sparsity** of $\mathbf{Z}$ and its **dimensionality**.
- Ghahramani et al. (2007) introduced a a **two-parameter generalization of the IBP, with**

$$\pi_z | \alpha, \beta \sim \mathrm{Beta}\left(\frac{\alpha\beta}{K}, \beta\right)$$

- The generative process

1. The first customer starts at the left of the buffet and samples Poisson(α) dishes.
2. The ith customer takes any dish previously sampled with probability $m_k/(\beta + i - 1)$, then he takes additional Poisson($\alpha\beta/(\beta + i - 1)$) dishes.

Two-Parameter IBP

- IBP has only one parameter α which controls both the **sparsity** of $\mathbf{Z}$ and its **dimensionality**.
- Ghahramani et al. (2007) introduced a a **two-parameter generalization of the IBP, with**

$$\pi_z | \alpha, \beta \sim \text{Beta}\left(\frac{\alpha\beta}{K}, \beta\right)$$

- The generative process
  1. The first customer starts at the left of the buffet and samples Poisson(α) dishes.
  2. The ith customer takes any dish previously sampled with probability $m_k/(\beta + i - 1)$, then he takes additional Poisson($\alpha\beta/(\beta + i - 1)$ dishes.
- Parameter β **controls the number of shared features** between objects.

Two-Parameter IBP

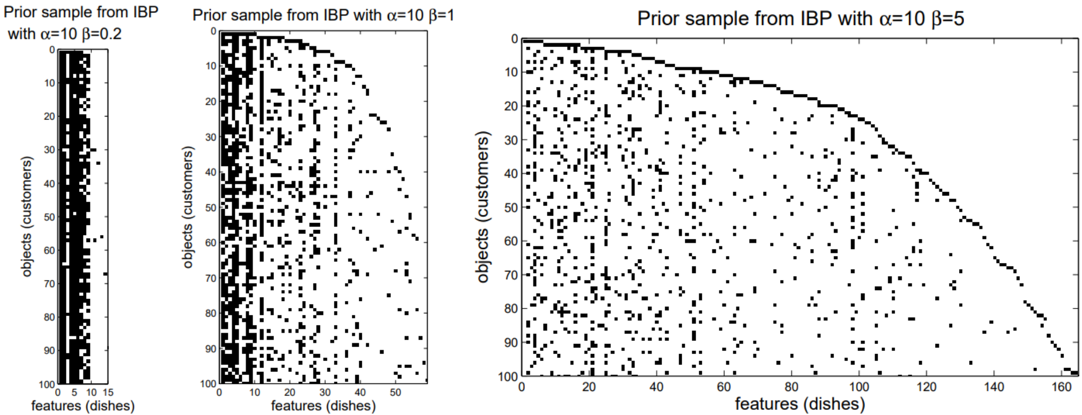

Figure 6: Three samples from the two-parameter IBP with $\alpha = 10$ and $\beta = 0.2$ (left), $\beta = 1$ (middle), and $\beta = 5$ (right).

Bibliography I

D. Blackwell and J. B. MacQueen. Ferguson Distributions Via Polya Urn Schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

Z. Ghahramani, T. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. *Bayesian Statistics*, 8, 2007.

T. L. Griffiths and Z. Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12(32):1185–1224, 2011.

N. L. Hjort. Nonparametric bayes estimators based on beta processes in models for life history data. *Annals of Statistics*, 18:1259–1294, 1990.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

Y. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the indian buffet process. pages 556–563. MIT Press, 2007.