10-708: Probabilistic Graphical Models 10-708, Spring 2017

Bayesian nonparametrics: Dirichlet Process, Indian Buffet Process

Lecturer: Avinava Dubey *Scribes: Jilong Yu, Yang Yang*

1 Nonparametric Bayesian Graphical Model

1.1 Bayesian Finite Mixture Model

A Bayesian approach of inferring an unknown underlying distribution $\mathcal{D}$ of data $x_1, x_2, ..., x_N$ is by placing a prior over $\mathcal{D}$ then computing the posterior over $\mathcal{D}$ given data. Traditionally, the prior over distributions is given by a parametric family with finite number of parameters and parameters are then integrated out in inference.

For example, in modeling following clustered data, mixture of K Guassians is usually adopted for likelihood representation.

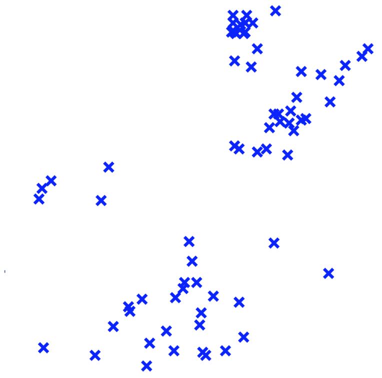


Figure 1: unlabeled 2-D data

$$p(x_1, ..., x_N|\pi, \{\mu_k\}, \{\Sigma_k\}) = \prod_{n=1}^{\infty} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_k|\mu_k, \Sigma_k)$$

A Bayesian approach put priors $p(\pi), p(\Sigma_{1:K}), p(\mu_{1:K})$ on them and integrate out.

$$p(x_1, ..., x_N) = \int\int\int \left(\prod_{n=1}^{\infty} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_k|\mu_k, \Sigma_k) \right) p(\pi)p(\Sigma_{1:K})p(\mu_{1:K}) d\pi d\Sigma_{1:K} d\mu_{1:K}$$

Usually, the conjugate priors are chosen where possible, so here we use Guassian for mean distribution and inverse Wishart for distribution of covariance matrix.

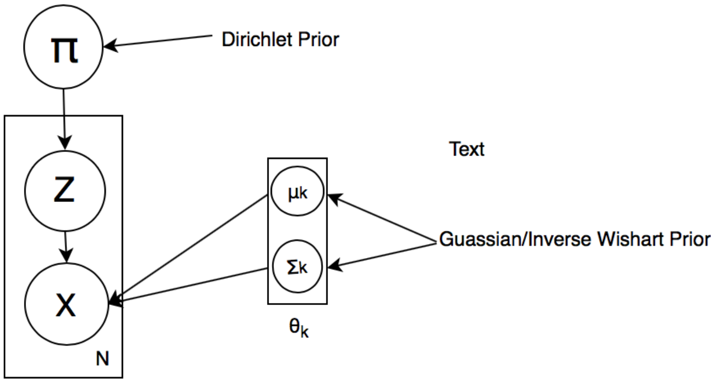


Figure 2: finite guassian mixture model

However,this traditional ways of using a fixed and finite number of parameters can suffer from over or under fitting of data when there is a misfit between the complexity of the model (often expressed in terms of the number of parameters, and in this case, number of clusters) and the amount of data available. As a result, model selection, or the choice of a model with the right complexity, is often an important issue in parametric modeling. Unfortunately, model selection is an operation that is fraught with difficulties, whether we use cross validation or marginal probabilities as the basis for selection.

1.2 Bayesian nonparametric approach

The Bayesian nonparametric approach is an alternative to parametric modeling and selection. Instead of having finite and fixed number of parameters that are independent of the dataset, Bayesian nonparametric approach use a model of unbounded complexity, the number of parameters can grow as size of dataset increases. It can be understood that the model has infinite or random number of parameters.

2 Dirichlet and Hierarchical Dirichlet Process

2.1 The Dirichlet distribution and its property

The Dirichlet distribution is a distribution over $K-1$ dimensional simplex ($\pi_{1:K} \in \{\pi_{1:K} | \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0\}$) with K dimensional parameter vector $\alpha_{1:K} \in \{\alpha_{1:K} | \alpha_k \geq 0\}$.

$$Dirichlet(\pi|\alpha_1, ..., \alpha_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

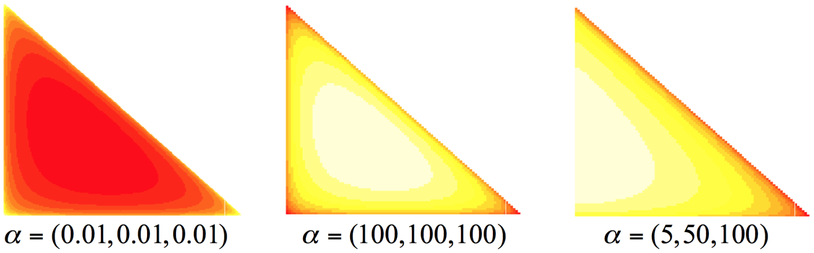


Figure 3: Density of the 3-component Dirichlet distribution for different parameters. Red indicates higher density.

It is conjugacy to the multinomial distribution. Therefore, the posterior of the multinomial likelihood is also a Dirichlet distribution:

$$\begin{aligned}
p(\pi|x_1, ..., x_N) &\propto p(x_1, ..., x_N|\pi)p(\pi) \\
&= \left(\frac{n!}{m_1!m_2!...m_K!}\pi_1^{m_1}...\pi_K^{m_K}\right)\left(\frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)}\prod_{k=1}^K \pi_k^{\alpha_k - 1}\right) \\
&\propto \frac{\Gamma(\sum_{k=1}^K \alpha_k + m_k)}{\prod_{k=1}^K \Gamma(\alpha_k + m_k)}\prod_{k=1}^K \pi_k^{\alpha_k + m_k - 1} \\
&= Dirichlet(\pi|m_1 + \alpha_1, ..., m_K + \alpha_K)
\end{aligned}$$

Which suggests, the more data we see for a particular component, the more concentrate the posterior will become on that particular component.

Properties:

- $\mathbf{E}\left[(\pi_1, ...\pi_N)\right] = \frac{(\alpha_1,...,\alpha_N)}{\sum_{k=1}^N \alpha_k}$
- Relationship to Gamma Distribution: If $\eta_k \sim Gamma(\alpha_k, 1)$, $\frac{(\eta_1,...\eta_K)}{\sum_{k=1}^K \eta_k} \sim Dirichlet(\alpha_1, ..., \alpha_K)$
- If $(\pi_1, ...\pi_K) \sim Dirichlet(\alpha_1, ..., \alpha_K)$, $(\pi_1 + \pi_2, \pi_3, ...\pi_K) \sim Dirichlet(\alpha_1 + \alpha_2, \alpha_3, ..., \alpha_K)$
- Beta Distribution is a Dirichlet distribution on a 1D simplex.
- If $(\pi_1, ...\pi_K) \sim Dirichlet(\alpha_1, ..., \alpha_K)$ and $\theta \sim Beta(\alpha_1 b, \alpha_1(1 - b))$, then $(\pi_1\theta, \pi_1(1 - \theta), \pi_2, ...\pi_K) \sim Dirichlet(\alpha_1 b, \alpha_1(1 - b), \alpha_2, ..., \alpha_K)$
- If $(\pi_1, ...\pi_K) \sim Dirichlet(\alpha_1, ..., \alpha_K)$ and $\theta \sim Beta(\alpha_1 b_1, \alpha_1 b_2, ..., \alpha_1 b_N)$ with $\sum_{n=1}^N b_n = 1$, then $(\pi_1\theta_1, \pi_1\theta_2, ..., \pi_1\theta_N, \pi_2, ...\pi_K) \sim Dirichlet(\alpha_1 b_1, \alpha_1 b_2, ..., \alpha_1 b_N, \alpha_2, ..., \alpha_K)$
- Renormalization: If $(\pi_1, ...\pi_K) \sim Dirichlet(\alpha_1, ..., \alpha_K)$, then $\frac{(\pi_2,...,\pi_N)}{\sum_{k=2}^N \pi_k} \sim Dirichlet(\alpha_2, ..., \alpha_K)$

2.2 Dirichlet prior in finite mixture model

Dirichlet distribution is a distribution over positive vectors that sum to one. In finite mixture model, if we use Dirichlet prior for mixture weights, then each entry in the drawn vector is associated with a parameter value (mean, covariance matrix) drawn from the corresponding prior distribution.

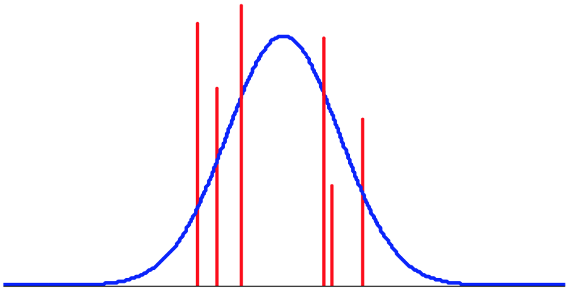
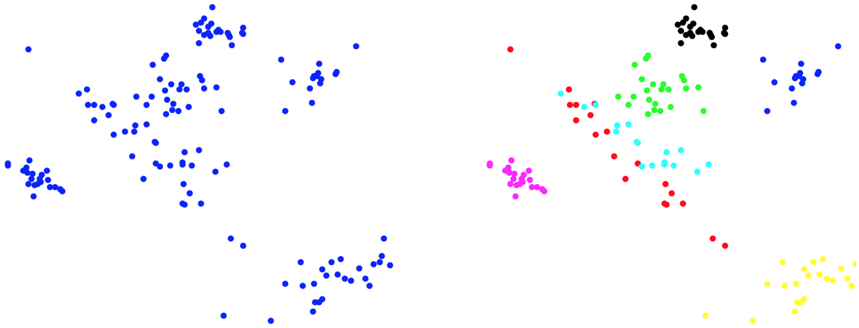


Figure 4: Sample from Dirichlet distribution is a distribution over parameters

2.3 Bayesian nonparametric mixture model and Dirichlet process

In Bayesian finite mixture model, a problem is defining the cluster number. Theoretically, our prior should be independent of the dataset, but as we see more data points, our assumption on the cluster number might be changed and our prior should also be changed accordingly. But in Bayesian nonparametric, we assume an infinite, that is, finite but random number of clusters. Therefore, we need something like Dirichlet Prior with infinite number of components.



$$p(x_n|\pi, \{\mu_k\}, \{\Sigma_k\}) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

The Dirichlet process (DP) is a stochastic process whose sample paths are probability measures with probability one. Draws from a DP can be interpreted as random distributions. It is an infinite dimensional generalization of Dirichlet distribution.

Let:

$$\begin{aligned}
\pi|\alpha &\sim Dirichlet(\frac{\alpha}{K}, ..., \frac{\alpha}{K}) \\
z_i|\pi &\sim Multinomial(\pi) \\
\theta_k^*|H &\sim H \\
x_i|z_i, \{\theta_k^*\} &\sim F(\theta_{z_i}^*)
\end{aligned}$$

Where H is the distribution over component parameters θ_i^*. And $F(\theta)$ is the component distribution parameterized by θ. For large K, the number of components typically used to model n data items becomes independent of K and is approximately $O(\alpha log(n))$. This implies that the mixture model stays well-defined as $K \to \infty$, leading to what is known as an infinite mixture model.

2.3.1 Dirichlet Process

- Let $\pi \sim \lim_{K\to\infty} Dirichlet(\frac{\alpha}{K}, ..., \frac{\alpha}{K})$
- For $k = 1, ..., \infty$, let $\theta_k \sim H$
- Then $G := \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ is a sample from infinite distribution over H.
- We say G is a Dirichlet process with base H and concentration parameter α.

$$G \sim DP(\alpha, H)$$

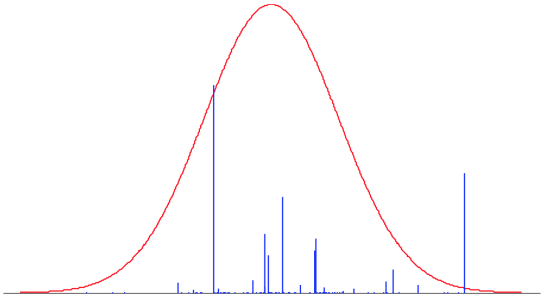


Figure 5: Sample from Dirichlet process.We call the point mass in the resulting distribution atoms, and base measure H determine the location of atoms

The concentration parameter α determines the size of the atoms. With increasing α, the size should be increasing.

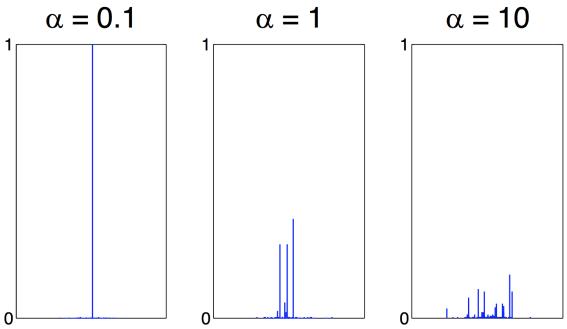


Figure 6: Small value of α give sparse distributions

For any finite measurable partition $A_1, ..., A_r$ of prameter probability spcae, Θ, $(P(A_1), ..., P(A_r))$ is random because G is random. And

$$(P(A_1), ..., P(A_r)) \sim Dirichlet(\alpha H(A_1), ..., \alpha H(A_r))$$

2.3.2 Conjugacy of Dirichlet Process

If we are given one observation x The posterior of the dirichlet process G should be:

$$G|x \sim DP(+1, \frac{\alpha H + \delta_x}{\alpha + 1})$$

Then if we see more observation in any partition A_j:

$$((P(A_1), ..., P(A_r))|x \in A_j) \sim Dirichlet(\alpha H(A_1), ..., (\alpha + 1)H(A_j), ..., \alpha H(A_r))$$

2.3.3 Predictive Distribution

The Dirichlet distribution cluster observations. A new data point can either join a cluster or start a new cluster. The first point always start a new cluster and we sample a parameter for that cluster. If we split our parameter space in two: singleton θ_1 and the rest in the begining. Then:
A priori: $(\pi_1, \pi^*) \sim Dirichlet(0, \alpha)$.
After we see a point of θ_1, our posterior will change: $(\pi_1, \pi^*) \sim Dirichlet(1, \alpha)$

$$\begin{aligned} P(X_2 = \theta_k|X_1 = \theta_1) &= \int P(X_2 = \theta_k|(\pi_1, \pi^*))P((\pi_1, \pi_*)|X_1 = \theta_1)d\pi_1 \\ &= \int \pi_k Dirichlet((\pi_1, 1 - \pi_1)|1, \alpha)d\pi_1 \\ &= E_{Dirichlet(1,\alpha)}\left[\pi_k\right] \\ &= \begin{cases} \frac{1}{1+\alpha} & \text{if} \quad k = 1 \\ \frac{\alpha}{1+\alpha} & \text{for} \quad \text{new k} \end{cases} \end{aligned}$$

$$\begin{aligned} P(X_3 = \theta_k|X_1 = \theta_1, X_2 = \theta_2) &= \int P(X_3 = \theta_k|\pi)P(\pi|X_1 = \theta_1, X_2 = \theta_2)d\pi \\ &= \int \pi_k Dirichlet(\pi|1, 1, \alpha)d\pi \\ &= E_{Dirichlet(1,1,\alpha)}\left[\pi_k\right] \\ &= \begin{cases} \frac{1}{2+\alpha} & \text{if} \quad k = 1 \\ \frac{1}{2+\alpha} & \text{if} \quad k = 2 \\ \frac{\alpha}{2+\alpha} & \text{for} \quad \text{new k} \end{cases} \end{aligned}$$

In general, if m_k is the number of times we have seen $X_i = k$, and n is the total number of observed values,

$$\begin{aligned} P(X_{n+1} = \theta_k|X_1, ..., X_n) &= \int P(X_{n+1} = \theta_k|\pi)P(\pi|X_1, ..., X_n)d\pi \\ &= E_{Dirichlet(m_1,...,m_k,\alpha)}\left[\pi_k\right] \\ &= \begin{cases} \frac{m_k}{n+\alpha} & \text{if} \quad k \leq K \\ \frac{\alpha}{n+\alpha} & \text{for} \quad \text{new k} \end{cases} \end{aligned}$$

This equation show two properties. Firstly, if we have a big cluster, the probability of sampling from that cluster becomes bigger, which is call rich-get-richer property. But we still have small probability of picking a new cluster, which means we can always add new feature to our model.

2.3.4 Metaphor: Polya urn schem & Chinese Restaurant process

The resulting distribution over data points can be thought of using follwing Polya urn scheme:

- An urn initially contains a black ball of mass α.
- For $n = 1, 2, ...$, sample a ball from the urn with probability proportional to its mass.
- If the ball is black, choose a previously unseen color, record that color, and return the black ball plus a unit mass ball of the new color to the urn.
- If the ball is not black, record it's color and return it, plus another unit-mass ball of the same color to the urn.

Chinese Restaurant Process:

- The first customer enters a restaurant, and picks a table.
- The n^{th} customer enters the restaurant. He sits at an existing table with probability $m_k/(n-1+\alpha)$, where m_k is the number of people sat at table k. He starts a new table with probability $\alpha/(n-1+\alpha)$.

In CRP, the distribution over the clustering of the first N customers does not depend on the order in which they arrived. However, the customers are not independent, they tend to sit at popular tables. We say distribution like this are exchangeable. If a sequence of observations are exchangeable, there must exist a distribution given which they are iid.

2.3.5 Stick breaking construction

This analogy will give the same posterior distribution with Dirichlet process, and works well with the DP mixture Model.
Imagine a stick of length 1, representing total probability.
For $k = 1, 2, ...$

- Sample a $Beta(1, \alpha)$ random variable b_k.
- Break off a fraction b_k of the stick. This is the k^th atom size.
- Sample a random location for this atom.
- Recurse on the remaining stick:

$$G := \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$$

$$\pi_k := b_k \prod_{j=1}^{k-1} (1 - b_k)$$

$$b_k \sim Beta(1, \alpha)$$

2.4 Inference in DP mixture Model

$$\begin{aligned} G := \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} &\sim DP(\alpha, H) \\ \phi_n &\sim G \\ x_n &\sim f(\phi_n) \end{aligned}$$

2.4.1 Collapsed Sampler

When sampling any data point, we can always rearrange the ordering so that it is the last data point. So, let z_n be the cluster allocation of the nth data point. Let K be the total number of instantiated clusters. Then:

$$p(z_n = k | x_n, z_{-n}, \phi_{1:K}) \propto \left\{ \begin{array}{cc} m_k f(x_n|\phi_k) & k \leq K \\ \alpha \int_\Omega f(x_n|\phi) H(d\phi) & k = K+1 \end{array} \right.$$

If we use a conjugate prior for the likelihood, we can often integrate out the cluster parameters. Because we are integrating out the parameters, this method is called collapsed. After integrating out the parameters, the clusters become dependent according to figure2. We have to sample each of the data in a sequential fashion. In addition, imagine two "true" clusters are merged into a single cluster - a single data point is unlikely to "break away". Furthermore, getting to the true distribution involves going through low probability states, so mixing can be slow. Even worse, if the likelihood is not conjugate, integrating out parameter values for new features can be difficult.

2.4.2 Blocked Gibbs Sampler

Rather than integrate out G, we can instantiate it.

- Approximate G with a truncated stick-breaking process:

$$\begin{aligned} G &:= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k} \\ \pi_k &:= b_k \prod_{j=1}^{k-1} (1 - b_k) \\ b_k &\sim Beta(1, \alpha), k = 1, ..., K-1 \\ b_K &= 1 \end{aligned}$$

- Sampling the cluster indicators:

$$p(z_n = k | rest) \propto \pi_k f(x_n|\theta)$$

- Sampling the stick breaking variables:
  We can think of the stick breaking process as a sequence of binary decisions.

  1. Choose $z_n = 1$ with probability b_1.
  2. If $z_n \neq 1$, choose $z_n = 2$ with probability b_2.
  3. etc...

$$b_k | rest \sim Beta(1 + m_k, \alpha + \sum_{j=k+1}^{K} m_j)$$

However, the problem with batch sampler is that fixed truncation introduces error.

2.4.3 Slice Sampler

the problem with batch sampler is that fixed truncation introduces error. So we got the idea of introducing random truncation. If we marginalize over the random truncation, we recover the full model.

- Introduce a uniform random variable u_n for each data point.
- Sample indicator z_n according to $p(z_n = k|rest) = I(\pi_k > u_n)f(x_n|\theta_k)$
- Only a finite number of possible values.
- The conditional distribution for un is just:

$$u_n|rest \sim Uniform(0, \pi_{z_n})$$

- Conditioned on the u_n and the z_n, the $_k$ can be sampled according to the block Gibbs sampler.
- Only need to represent a finite number K of components such that:

$$1 - \sum_{k=1}^{K} \pi_k < min(u_n)$$

3 Indian Buffet Process

3.1 Infinite Topic Models

[This part is partially related to Hierarchical Dirichlet Process. I'll first cover it here and we can adjust it later] Topic models describe documents using a distribution over features, and feature is a distribution over words. Each document is represented as a collection of words, usually under the unordered "bag of words" assumption. The words within a document are distributed according to a document-specific mixture model, such that each word in a document is associated with a feature. The features are shared between documents, and the features tend to give high probability to semantically related words, which are called the "topics".

Latent Dirichlet Allocation (LDA) is a widely used method for topic modeling [1]. The process is as followed and the model is shown in Figure 7.

- For each topic $k = 1, \cdots, K$, sample a distribution over words, $\beta \sim \text{Dir}(\eta_1, \cdots, \eta_V)$.
- For each document $m = 1, \cdots, M$
  - sample a distribution over topics, $\theta_m \sim \text{Dir}(\alpha_1, \cdots, \alpha_K)$,
  - for each word $n = 1, \cdots, N_m$, sample a topic $z_{mn} \sim \text{Discrete}(\theta_m)$, and sample a word $w_{mk} \sim \text{Discrete}(\beta_z)$.

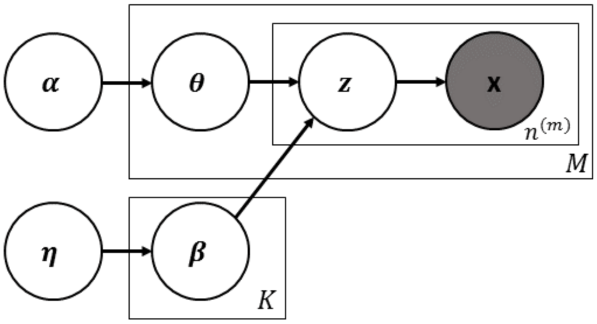


Figure 7: The method of LDA

In LDA each distribution is associated with a distribution over K topics. The first problem is how to choose the number of topics. One solution is to choose infinitely many topics and replace the Dirichlet distribution over topics with a Dirichlet process. The second problem is to how to ensure that the topics are shared between documents.

In LDA we have M independent samples from a Dirichlet distribution. The weights are different, but the topics are fixed to be the same. If we replace the Dirichlet distributions with Dirichlet processes, each atom of each Dirichlet Process will pick a topic independently of the other topics. Because the base measure is continuous, the probability of picking the same topic twice is zero.

To have shared topics, we need to use discrete base measure, e.g., $H = \sum_{k=1}^{K} \alpha_k \delta_{\beta_k}$. We want to have an infinite number of topics and the location of the topics is random, so we need an infinite, discrete, random base measure.

The solution is to sample the base measure from a Dirichlet Process (Teh et al, 2006), which is known as the Hierarchical Dirichlet Process [2] and shown in Figure 8.

$$G_0 \sim \mathrm{DP}(\gamma, H)$$
$$G_m \sim \mathrm{DP}(\alpha, G_0)$$

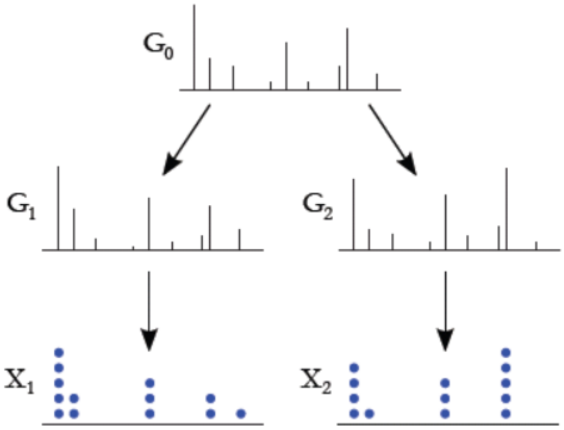


Figure 8: Hierarchical Dirichlet Process to sample discrete base measure.

This model can be described by a stochastic process named Chinese restaurant franchise. Suppose there is a franchise of restaurants. We have the following assumptions: i.the franchise of Chinese restaurants serve an infinitely large, global menu, ii. each table in each restaurant orders a single dish. Let n_{rt} be the number of customers in restaurant r sitting at table t. Let m_{rd} be the number of tables in restaurant r serving dish d. Let $m_{\cdot d}$ be the number of tables serving dish d across all the restaurants. The rules of the Chinese restaurant franchise are as following:

- Customers enter the restaurant, and sit at tables according to the Chinese Restaurant Process
- Each table in each restaurant picks a dish, with probability proportional to the number of times it has been served across all restaurant.

Therefore,

$$p(\text{table } t \text{ chooses dish } d|\text{previous tables}) = \begin{cases} \frac{m_d}{T+\gamma}, & \text{for an existing table} \\ \frac{\gamma}{T+\gamma}, & \text{for a new table} \end{cases}$$

If we image the restaurants represent the documents, and the dishes represent the topics, then this is an infinite topic model. Let H be a V-dimensional Dirichlet distribution, so a sample from H is a distribution over a vocabulary of V words. Firstly, we sample a global distribution over topics,

$$G_0 := \sum_{k=1}^{\infty} \pi_k \delta_{\beta_k} \sim \mathrm{DP}(\alpha, H).$$

Next we take the following sampling procedure:

- Sample a distribution over topics, $G_m \sim \mathrm{DP}(\gamma, G_0)$.

- For each word $n = 1, \cdots, N_m$
  - Sample a topic $\phi_{mn} \sim \text{Discrete}(G_0)$
  - Sample a word $w_{mk} \sim \text{Discrete}(\phi_{mn})$

The comparison of topic numbers in the typical LDA model and the Hierarchical Dirichlet Process (HDP) Mixture based model are shown as follows. Lower perplexity represents better performance. The results show that the HDP Mixture based model gains performance improvement over the typical LDA model.

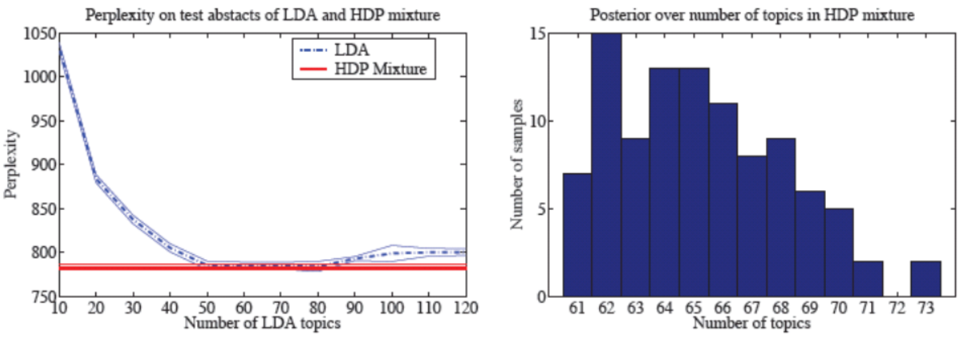


Figure 9: Topic number evaluation of the HDP Mixture based model and comparison with the LDA model.

3.2 Infinite Latent Variable Models

The Dirichlet distribution and the Dirichlet Process are very useful if we want to cluster data into non-overlapping clusters. However, the Dirichlet Process and Dirichlet mixture models cannot share features between clusters. In many cases, data points exhibit properties of multiple latent features. For example, images may contain multiple objects. Movies contain aspects of multiple genres. Actors in social networks may belong to multiple social groups.

Latent variable models allow each data point to exhibit multiple features to varying degrees. For example, factor analysis employs a latent variable model, where the data points are presented as $\mathbf{X} = \mathbf{W}\mathbf{A}^T + \epsilon$. Each row of $\mathbf{A}$ corresponds to a feature. Each row of $\mathbf{W}$ represents data point specific weights for the features. ϵ is the Gaussian noise. Another example is the Latent Dirichlet Allocation (LDA) model, where each document is represented by a mixture of features.

For the latent variable models, an important problem is how to choose the number of features. We ask the question: can we make the number of features unbounded a posterior, as we did with the Dirichlet Process? A solution is to allow infinitely many features a priori. We still take factor analysis for example. The solution is to let $\mathbf{A}$ has infinitely many rows. But the problem is we cannot represent infinitely many features. Griffits, et al.[3] proposed the solution of designing a distribution to make the infinitely large matrix sparse and derived the distribution from a stochastic process, known as the Indian Buffet Process (IBP).

Recall that the Chinese Restaurant Process gives us a distribution over partitions of the data. We can represent it as a distribution over binary matrices, where each row corresponds to a data point, each column corresponds to a cluster. For the latent variable model, suppose we have N objects and K features. A binary matrix $\mathbf{Z}$ is used to indicate the possession of features by each object. A binary variable z_{nk} indicates whether the nth object has the feature k. The z_{nk} then form the $N \times K$ binary matrix $\mathbf{Z}$.

Taking the factor analysis as an example, to obtain a sparse latent variable model, let

$$\begin{aligned} \mathbf{X} &= \mathbf{W}\mathbf{A}^T + \epsilon, \\ \mathbf{W} &= \mathbf{Z} \odot \mathbf{V} \end{aligned}$$

for some sparse matrix $\mathbf{Z}$.

We place a beta-Bernoulli prior on Z, such that

$$\begin{aligned} \pi_k &\sim Beta(\frac{\alpha}{K}, 1),\ k = 1, \cdots, K \\ z_{nk} &\sim Bernoulli(\pi_k),\ n = 1, \cdots, N. \end{aligned}$$

The marginal probability of the matrix $\mathbf{Z}$ is

$$\begin{aligned} p(\mathbf{Z}) &= \prod_{k=1}^{K} \int \Big(\prod_{i=1}^{N} p(z_{nk}|\pi_k)\Big) p(\pi_k) d\pi_k \\ &= \prod_{k=1}^{K} \frac{B(m_k + \alpha/K, N - m_k + 1)}{B(\alpha/K, 1)} \\ &= \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \alpha/K)\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \alpha/K)}, \end{aligned}$$

where $m_k = \sum_{n=1}^{N} z_{nk}$. $\mathbf{Z}$ is exchangeable as $P(\mathbf{Z})$ is not dependent on the order of the rows or columns. To observe the sparsity of the matrix for a infinite model, we take the K to infinity. Since all the columns are equal in expectation, there are more and more empty columns as K grows. However, we do not want to represent infinitely many empty columns.

Let $[\mathbf{Z}]$ be an equivalence class of matrices where the nonzero columns are all to the left of the empty columns. Let $lof(\cdot)$ be a function that maps binary matrices to left-ordered binary matrices. The matrix transformation by $log(\cdot)$ is shown in Figure 10 [4]. Since all the matrices in the equivalence set $[\mathbf{Z}]$ have equal probabilities by exchangeability of the columns, we know its probability if we know the size of the $[\mathbf{Z}]$.

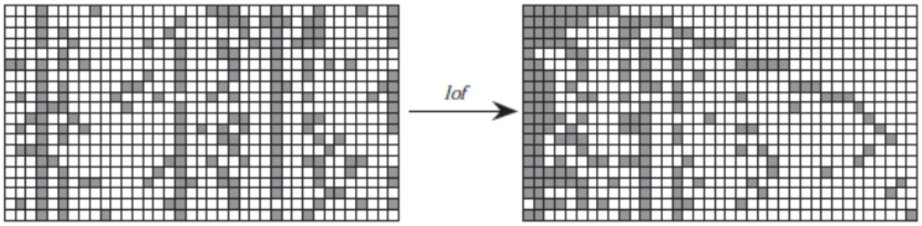


Figure 10: Binary matrices and the left-ordered form. The binary matrix on the left is transformed into the left-ordered binary matrix on the right by the function $lof(\cdot)$. This left-ordered matrix was generated from the exchangeable Indian buffet process with $\alpha = 10$. Empty columns are omitted from both matrices.

Let the vector $(z_{1k}, z_{2k}, \cdots, z_{(n-1)k})$ be the history of feature k at data point n. Let K_h and K_+ denote the number of features possessing history h, and the total number of features, respectively. The total number of lof-equivalent matrices in $[\mathbf{Z}]$ is $\frac{K!}{\prod_{n=0}^{2^N-1} K_n!}$. The probability of $[\mathbf{Z}]$ is

$$\begin{aligned} p([\mathbf{Z}]) &= \sum_{\mathbf{Z}\in[\mathbf{Z}]} p(\mathbf{Z}) \\ &= \frac{K!}{\prod_{n=0}^{2^N-1} K_n!} \prod_{k=1}^{K} \frac{\alpha}{K} \frac{\Gamma(m_k + \alpha/K)\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \alpha/K)} \\ &= \frac{\alpha^{K_+}}{\prod_{n=1}^{2^N-1}} \frac{K!}{K_0! K^{K_+}} \Big(\frac{N!}{\prod_{j=1}^{N}(j + \alpha/K)}\Big)^K \prod_{k=1}^{K_+} \frac{(N - m_k)! \prod_{j=1}^{m_k-1}(j + \alpha/K)}{N!}. \end{aligned}$$

We can take the limit of the finite model as K approaches infinity. As

$$\lim_{K\infty} \frac{K!}{K_0! K^{K_+}} = 1, \quad \lim_{K\infty} \Big(\frac{N!}{\prod_{j=1}^{N} j + \alpha/K}\Big)^K = \exp\{-\alpha H_N\}$$

$$\lim_{K\infty} \prod_{k=1}^{K_+} \frac{(N - m_k)! \prod_{j=1}^{m_k-1}(j + \alpha/K)}{N!} = \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!},$$

we have

$$\begin{aligned} \lim_{K\to\infty} p([\mathbf{Z}]) &= \lim_{K\to\infty} \frac{\alpha^{K_+}}{\prod_{n=1}^{2^N-1}} \cdot \frac{K!}{K_0! K^{K_+}} \cdot \Big(\frac{N!}{\prod_{j=1}^{N} j + \alpha/K}\Big)^K \cdot \prod_{k=1}^{K_+} \frac{(N - m_k)! \prod_{j=1}^{m_k-1}(j + \alpha/K)}{N!} \\ &= \frac{\alpha^{K_+}}{\prod_{n=1}^{2^N-1} K_n!} \cdot \exp\{-\alpha H_N\} \cdot \prod_{k=1}^{K_+} \frac{(N - m_k)!(m_k - 1)!}{N!}, \end{aligned} \tag{1}$$

where H_N is the Nth harmonic number, $H_N = \sum_{j=1}^{N} \frac{1}{j}$.

3.3 Indian Buffet Process

The sparse, infinite latent variable model can be described by the Indian Buffet Process. In an Indian Buffet Process, we have the following assumptions and rules:

- N customers enter a restaurant one after another.
- The restaurant provides the customers with a buffet consisting of infinitely many dishes arranged in a line.
- The first customer starts from the left of the buffet, taking a serving from each dish, and stops after picking Poisson(α) dishes.
- The nth customer moves along the buffet and picks dishes in proportional to their popularity. He helps himself to each dish with probability m_k/n, where m_k is the number of previous customers who have sampled the dish.
- The nth customer then tries a Poisson(α/n) number of new dishes.

We use a binary matrix $\mathbf{Z}$ with N rows and infinitely many columns to represent which customers chose which dishes, where $z_{nk} = 1$ if the nth customer chose the kth dish. Let $K_1^{(n)}$ be the number of new chosen dishes (features) in the n^{th} row. We will show the IBP is lof-equivalent to the infinite beta-Bernoulli model. The probability of a particular matrix produced by this process is

$$\begin{aligned}
P(\mathbf{Z}) &= \prod_{n=1}^{N} p(\mathbf{z}_n|\mathbf{z}_{1:(n-1)}) \\
&= \prod_{n=1}^{N} Poisson\Big(K_1^{(n)}|\frac{\alpha}{n}\Big) \prod_{k=1}^{K_+} \Big(\frac{\sum_{i=1}^{n-1} z_{ik}}{n}\Big)^{z_{nk}} \Big(\frac{n-\sum_{i=1}^{n-1} z_{ik}}{n}\Big)^{1-z_{nk}} \\
&= \prod_{n=1}^{N} \Big(\frac{\alpha}{n}\Big)^{K_1^(n)} \frac{1}{K_1^{(n)}!} e^{-\alpha/n} \prod_{k=1}^{K_+} \Big(\frac{\sum_{i=1}^{n-1} z_{ik}}{n}\Big)^{z_{nk}} \Big(\frac{n-\sum_{i=1}^{n-1} z_{ik}}{n}\Big)^{1-z_{nk}} \\
&= \frac{\alpha^{K_+}}{\prod_{n=1}^{N} K_1^{(n)}!} \exp\{-\alpha H_N\} \prod_{k=1}^{K_+} \frac{(N-m_k)!(m_k-1)!}{N!}.
\end{aligned}$$

The matrices produced by this process are generally not in left-ordered form. However, if we only consider the lof-equivalence classes of the matrices produced by this process, we can obtain the exchangeable distribution $P([\mathbf{Z}])$ from which the generated matrices have the left-ordered form. $P([\mathbf{Z}])$ can be obtained by multiplying $P(\mathbf{Z})$ with the cardinality $[\mathbf{Z}]$. It is shown that the Indian buffet process is lof-equivalent to the infinite beta-Bernoulli model expressed in Eq.(1).

IBP have some properties.

- The "rich get richer" property, as "popular" dishes become more popular.
- The number of nonzero entries for each row in the matrix $\mathbf{Z}$ is distributed according to Poisson(α) due to exchangeability.
- The number of nonzero entries for the whole matrix is distributed according to Poisson($N\alpha$).

  If x_1 ~Poisson(α_1) and x_2 ~Poisson(α_2), then $(x_1 + x_2)$ ~Poisson($\alpha_1 + \alpha_2$).

- The number of non-empty columns is distributed according to Poisson(αH_N).

The IBP can be used to build latent feature models with an unbounded number of features. Let each column of the IBP correspond to one of an infinite number of features. Each row of the IBP selects a finite subset of these features. The rich-get-richer property for the IBP ensures that features are shared between data points. To perform the latent feature model building from the IBP, we need to pick a likelihood model that determines what the featrues look like and how they are combined.

An example of the binary feature matrix generated from IBP is shown in Figure 11.

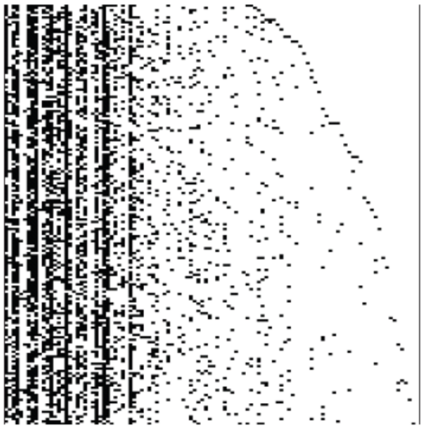


Figure 11: Example of a binary feature matrix sampled from the Indian buffet process.

4 Conclusions

Bayesian nonparametrics in general involve ways of defining infinite-dimensional priors over parameters. The Dirichlet Process is a stochastic process widely used in Bayesian nonparametric models of data, particularly in Dirichlet Process mixture models. The DP is motivated by moving from finite to infinite mixture models. It is the canonical distribution over probability measures with a wide range of generalizations. More complex constructions, such as Hierarchical Dirichlet Process are also developed. The Indian buffet process is a stochastic process defining a probability distribution over equivalence classes of sparse binary matrices with a finite number of rows (objects) and an unbounded number of columns (features). This distribution is suitable to use as a prior in probabilistic models that represent objects using a potentially infinite array of features.

References

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[2] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[3] Thomas L Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. In *NIPS*, volume 18, pages 475–482, 2005.

[4] Thomas L Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12(Apr):1185–1224, 2011.