Nonparametric Bayesian Models (Wrap-up)

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

April 1, 2019

Recap: Nonparametric Bayesian Mixture Models

- Also known as infinite mixture models. Can be mathematically represented as

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

Recap: Nonparametric Bayesian Mixture Models

- Also known as infinite mixture models. Can be mathematically represented as

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

where π_k and ϕ_k are the mixing prop. and params of the k-th component, and for $n = 1, \ldots, N$

Recap: Nonparametric Bayesian Mixture Models

- Also known as infinite mixture models. Can be mathematically represented as

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

where π_k and ϕ_k are the mixing prop. and params of the k-th component, and for $n = 1, \ldots, N$

$$\begin{aligned} \theta_n &\sim G \qquad (\theta_n \text{ will be equal to } \phi_k \text{ with prob. } \pi_k) \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\theta_n) \end{aligned}$$

Recap: Nonparametric Bayesian Mixture Models

- Also known as infinite mixture models. Can be mathematically represented as

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

where π_k and ϕ_k are the mixing prop. and params of the k-th component, and for $n = 1, \ldots, N$

$$\begin{aligned} \theta_n &\sim G & (\theta_n \text{ will be equal to } \phi_k \text{ with prob. } \pi_k) \\ \boldsymbol{x}_n &\sim p(\boldsymbol{x}|\theta_n) \end{aligned}$$

- Can view/define such infinite mixture models using various equivalent ways
  - Stick-breaking Process
  - Dirichlet Process
  - Chinese Restaurant Process
  - Pólya-Urn Scheme

Recap: Stick-Breaking Process

- Sethuraman's stick-breaking construction provides a sequential way to generate π_k's

Recap: Stick-Breaking Process

- Sethuraman's stick-breaking construction provides a sequential way to generate π_k's

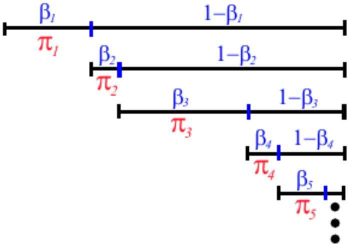
$$\beta_1 \sim \text{Beta}(1, \alpha), \quad \pi_1 = \beta_1$$

Recap: Stick-Breaking Process

- Sethuraman's stick-breaking construction provides a sequential way to generate π_k's

$$\beta_1 \sim \text{Beta}(1,\alpha), \quad \pi_1 = \beta_1$$

$$\beta_k \sim \text{Beta}(1,\alpha), \quad \pi_k = \beta_k \prod_{\ell=1}^{k-1}(1-\beta_{\ell-1}), \quad k = 2,\ldots,\infty$$



Recap: Dirichlet Process (DP)

- A Dirichlet Process DP(α, G_0) defines a distribution over distributions

Recap: Dirichlet Process (DP)

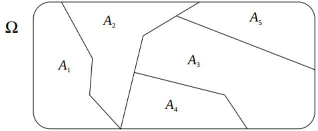
- A Dirichlet Process $\text{DP}(\alpha, G_0)$ defines a distribution over distributions
- If $G \sim \text{DP}(\alpha, G_0)$ then any finite dim. marginal of G is Dirichlet distributed

Recap: Dirichlet Process (DP)

- A Dirichlet Process $\text{DP}(\alpha, G_0)$ defines a distribution over distributions
- If $G \sim \text{DP}(\alpha, G_0)$ then any finite dim. marginal of G is Dirichlet distributed

$$[G(A_1), \ldots, G(A_K)] \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K))$$

for any finite partition $A_1, \ldots, A_K$ of the space Ω (Ferguson, 1973)

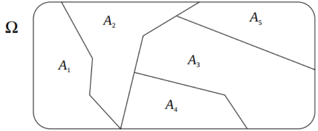


Recap: Dirichlet Process (DP)

- A Dirichlet Process $\text{DP}(\alpha, G_0)$ defines a distribution over distributions
- If $G \sim \text{DP}(\alpha, G_0)$ then any finite dim. marginal of G is Dirichlet distributed

$$[G(A_1), \ldots, G(A_K)] \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K))$$

for any finite partition $A_1, \ldots, A_K$ of the space Ω (Ferguson, 1973)



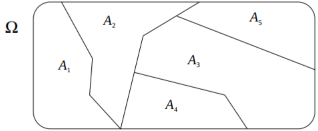
- G is a discrete distribution of the form $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$

Recap: Dirichlet Process (DP)

- A Dirichlet Process $\text{DP}(\alpha, G_0)$ defines a distribution over distributions
- If $G \sim \text{DP}(\alpha, G_0)$ then any finite dim. marginal of G is Dirichlet distributed

$$[G(A_1), \ldots, G(A_K)] \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K))$$

for any finite partition $A_1, \ldots, A_K$ of the space Ω (Ferguson, 1973)



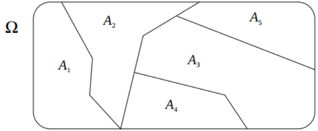
- G is a discrete distribution of the form $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$
- α is concentration parameter, G_0 is the base distribution of $\text{DP}(\alpha, G_0)$

Recap: Dirichlet Process (DP)

- A Dirichlet Process $\text{DP}(\alpha, G_0)$ defines a distribution over distributions
- If $G \sim \text{DP}(\alpha, G_0)$ then any finite dim. marginal of G is Dirichlet distributed

$$[G(A_1), \ldots, G(A_K)] \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K))$$

for any finite partition $A_1, \ldots, A_K$ of the space Ω (Ferguson, 1973)



- G is a discrete distribution of the form $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$
- α is concentration parameter, G_0 is the base distribution of $\text{DP}(\alpha, G_0)$
- $\mathbb{E}[G] = G_0$ and as $\alpha \to \infty$, $G \to G_0$

Recap: DP Posterior and Posterior Predictive

- Assume N i.i.d. draws $\theta_1, \ldots, \theta_N$ from the discrete distribution $G \sim \text{DP}(\alpha, G_0)$

Recap: DP Posterior and Posterior Predictive

- Assume N i.i.d. draws $\theta_1, \ldots, \theta_N$ from the discrete distribution $G \sim \text{DP}(\alpha, G_0)$
- The posterior of G will also be a DP (due to discrete-Dirichlet conjugacy)

$$G|\theta_1, \ldots, \theta_N \quad \sim \quad \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i})$$

Recap: DP Posterior and Posterior Predictive

- Assume N i.i.d. draws $\theta_1, \ldots, \theta_N$ from the discrete distribution $G \sim \text{DP}(\alpha, G_0)$
- The posterior of G will also be a DP (due to discrete-Dirichlet conjugacy)

$$G|\theta_1, \ldots, \theta_N \sim \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i})$$

$$\text{(equivalent to)} \quad G|\theta_1, \ldots, \theta_N \sim \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\phi_k})$$

.. where n_k = number of θ_i's that are equal to ϕ_k

Recap: DP Posterior and Posterior Predictive

- Assume N i.i.d. draws $\theta_1, \ldots, \theta_N$ from the discrete distribution $G \sim \text{DP}(\alpha, G_0)$
- The posterior of G will also be a DP (due to discrete-Dirichlet conjugacy)

$$G|\theta_1, \ldots, \theta_N \sim \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i})$$

$$\text{(equivalent to)} \quad G|\theta_1, \ldots, \theta_N \sim \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\phi_k})$$

.. where n_k = number of θ_i's that are equal to ϕ_k

- The posterior predictive for the next draw θ_{N+1} from G will be

$$\theta_{N+1}|\theta_1, \ldots, \theta_N \sim \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}$$

Recap: DP Posterior and Posterior Predictive

- Assume N i.i.d. draws $\theta_1, \ldots, \theta_N$ from the discrete distribution $G \sim \text{DP}(\alpha, G_0)$
- The posterior of G will also be a DP (due to discrete-Dirichlet conjugacy)

$$G|\theta_1, \ldots, \theta_N \sim \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i})$$

$$\text{(equivalent to)} \quad G|\theta_1, \ldots, \theta_N \sim \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\phi_k})$$

.. where n_k = number of θ_i's that are equal to ϕ_k

- The posterior predictive for the next draw θ_{N+1} from G will be

$$\theta_{N+1}|\theta_1, \ldots, \theta_N \sim \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}$$

$$\text{(equivalent to)} \quad \theta_{N+1}|\theta_1, \ldots, \theta_N \sim \frac{\alpha}{\alpha + N} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\phi_k} \quad \text{(mixture of } K + 1 \text{ distributions)}$$

Recap: DP Posterior and Posterior Predictive

- Assume N i.i.d. draws $\theta_1, \ldots, \theta_N$ from the discrete distribution $G \sim \text{DP}(\alpha, G_0)$
- The posterior of G will also be a DP (due to discrete-Dirichlet conjugacy)

$$G|\theta_1, \ldots, \theta_N \quad \sim \quad \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i})$$

$$\text{(equivalent to)} \quad G|\theta_1, \ldots, \theta_N \quad \sim \quad \text{DP}(\alpha + N, \frac{\alpha}{\alpha + N} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\phi_k})$$

.. where n_k = number of θ_i's that are equal to ϕ_k

- The posterior predictive for the next draw θ_{N+1} from G will be

$$\theta_{N+1}|\theta_1, \ldots, \theta_N \quad \sim \quad \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}$$

$$\text{(equivalent to)} \quad \theta_{N+1}|\theta_1, \ldots, \theta_N \quad \sim \quad \frac{\alpha}{\alpha + N} G_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\phi_k} \quad \text{(mixture of } K+1 \text{ distributions)}$$

i.e., $\theta_{N+1} = \phi_k$ with prob. $\frac{n_k}{\alpha+N}$ or a new value drawn from G_0 with prob. $\frac{\alpha}{\alpha+N} G_0$

A Sequential Generative Scheme

- The form of the DP predictive distribution

$$\theta_{N+1}|\theta_1, \ldots, \theta_N \sim \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}$$

suggests the following scheme to generate a sequence of parameters $\theta_1, \ldots, \theta_N, \theta_{N+1}, \ldots$

A Sequential Generative Scheme

- The form of the DP predictive distribution

$$\theta_{N+1}|\theta_1, \ldots, \theta_N \sim \frac{\alpha}{\alpha + N} G_0 + \frac{1}{\alpha + N} \sum_{i=1}^{N} \delta_{\theta_i}$$

suggests the following scheme to generate a sequence of parameters $\theta_1, \ldots, \theta_N, \theta_{N+1}, \ldots$

$$\theta_1 \quad \sim \quad G_0$$

A Sequential Generative Scheme

- The form of the DP predictive distribution

$$\theta_{N+1}|\theta_1,\ldots,\theta_N \sim \frac{\alpha}{\alpha+N}G_0 + \frac{1}{\alpha+N}\sum_{i=1}^{N}\delta_{\theta_i}$$

suggests the following scheme to generate a sequence of parameters $\theta_1,\ldots,\theta_N,\theta_{N+1},\ldots$

$$\begin{aligned}\theta_1 &\sim G_0\\ \theta_2|\theta_1 &\sim \frac{\alpha}{\alpha+1}G_0 + \frac{1}{\alpha+1}\delta_{\theta_1}\\ &\vdots\end{aligned}$$

A Sequential Generative Scheme

- The form of the DP predictive distribution

$$\theta_{N+1}|\theta_1,\ldots,\theta_N \sim \frac{\alpha}{\alpha+N}G_0 + \frac{1}{\alpha+N}\sum_{i=1}^{N}\delta_{\theta_i}$$

suggests the following scheme to generate a sequence of parameters $\theta_1,\ldots,\theta_N,\theta_{N+1},\ldots$

$$\begin{aligned}\theta_1 &\sim G_0\\ \theta_2|\theta_1 &\sim \frac{\alpha}{\alpha+1}G_0 + \frac{1}{\alpha+1}\delta_{\theta_1}\\ &\vdots\\ \theta_n|\theta_1,\ldots,\theta_{n-1} &\sim \frac{\alpha G_0 + \sum_{i=1}^{n-1}\delta_{\theta_i}}{\alpha+n-1}\end{aligned}$$

A Sequential Generative Scheme

- The form of the DP predictive distribution

$$\theta_{N+1}|\theta_1,\ldots,\theta_N \sim \frac{\alpha}{\alpha+N}G_0 + \frac{1}{\alpha+N}\sum_{i=1}^{N}\delta_{\theta_i}$$

suggests the following scheme to generate a sequence of parameters $\theta_1,\ldots,\theta_N,\theta_{N+1},\ldots$

$$\begin{aligned}
\theta_1 &\sim G_0 \\
\theta_2|\theta_1 &\sim \frac{\alpha}{\alpha+1}G_0 + \frac{1}{\alpha+1}\delta_{\theta_1} \\
&\vdots \\
\theta_n|\theta_1,\ldots,\theta_{n-1} &\sim \frac{\alpha G_0 + \sum_{i=1}^{n-1}\delta_{\theta_i}}{\alpha+n-1}
\end{aligned}$$

- The joint distribution $p(\theta_1,\theta_2,\ldots,\theta_n) = p(\theta_1)p(\theta_2|\theta_1)\ldots p(\theta_n|\theta_1,\ldots,\theta_{n-1})$

A Sequential Generative Scheme

- The form of the DP predictive distribution

$$\theta_{N+1}|\theta_1,\ldots,\theta_N \sim \frac{\alpha}{\alpha+N}G_0 + \frac{1}{\alpha+N}\sum_{i=1}^{N}\delta_{\theta_i}$$

suggests the following scheme to generate a sequence of parameters $\theta_1,\ldots,\theta_N,\theta_{N+1},\ldots$

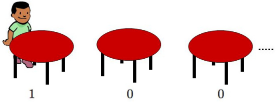
$$\begin{aligned}
\theta_1 &\sim G_0 \\
\theta_2|\theta_1 &\sim \frac{\alpha}{\alpha+1}G_0 + \frac{1}{\alpha+1}\delta_{\theta_1} \\
&\vdots \\
\theta_n|\theta_1,\ldots,\theta_{n-1} &\sim \frac{\alpha G_0 + \sum_{i=1}^{n-1}\delta_{\theta_i}}{\alpha+n-1}
\end{aligned}$$

- The joint distribution $p(\theta_1,\theta_2,\ldots,\theta_n) = p(\theta_1)p(\theta_2|\theta_1)\ldots p(\theta_n|\theta_1,\ldots,\theta_{n-1})$
- Note that $\theta_1,\ldots,\theta_{n-1},\theta_n$ is an "exchangeable sequence" (joint probability invariant to ordering)

$$p(\theta_1,\theta_2,\ldots,\theta_n) = p(\theta_{\sigma(1)},\theta_{\sigma(2)},\ldots,\theta_{\sigma(n)}) \qquad \text{(for any permutation } \sigma\text{)}$$

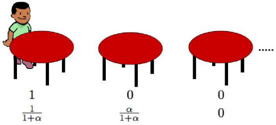
Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



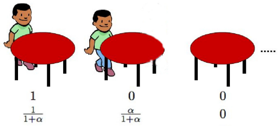
Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



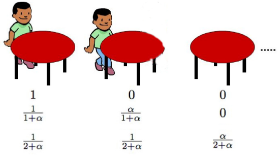
Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



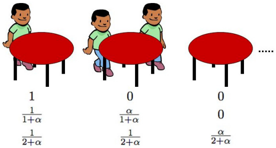
Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



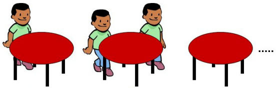
Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \dots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



Chinese Restaurant Process (CRP)

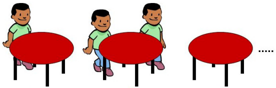
- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



1	0	0
$\frac{1}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	0
$\frac{1}{2+\alpha}$	$\frac{1}{2+\alpha}$	$\frac{\alpha}{2+\alpha}$
$\frac{1}{3+\alpha}$	$\frac{2}{3+\alpha}$	$\frac{\alpha}{3+\alpha}$

Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.

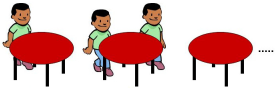


1	0	0
$\frac{1}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	0
$\frac{1}{2+\alpha}$	$\frac{1}{2+\alpha}$	$\frac{\alpha}{2+\alpha}$
$\frac{1}{3+\alpha}$	$\frac{2}{3+\alpha}$	$\frac{\alpha}{3+\alpha}$

- Probability of sitting at an already occupied table $k \propto n_k$ (n_k: # of people sitting at table k)

Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.

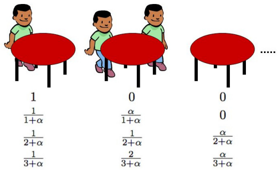


1	0	0
$\frac{1}{1+\alpha}$	$\frac{\alpha}{1+\alpha}$	0
$\frac{1}{2+\alpha}$	$\frac{1}{2+\alpha}$	$\frac{\alpha}{2+\alpha}$
$\frac{1}{3+\alpha}$	$\frac{2}{3+\alpha}$	$\frac{\alpha}{3+\alpha}$

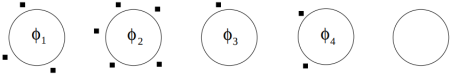
- Probability of sitting at an already occupied table $k \propto n_k$ (n_k: # of people sitting at table k)
- Probability of sitting at an unoccupied table $\propto \alpha$ (where α is a novelty hyperparameter)

Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.

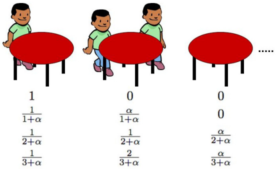


- Probability of sitting at an already occupied table $k \propto n_k$ (n_k: # of people sitting at table k)
- Probability of sitting at an unoccupied table $\propto \alpha$ (where α is a novelty hyperparameter)
- Imagine table k is associated with a unique ϕ_k. Then the arragement would look like..

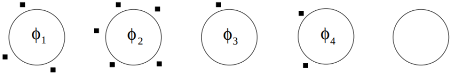


Chinese Restaurant Process (CRP)

- A metaphor to describe the way $\theta_1, \ldots, \theta_n$ (equivalently, the cluster assignments) are generated
- Think of the θ_i's as customers who sequentially enter a restaurant (need not be Chinese!) and decide which table to sit at. All θ_i's sitting at the same table will be identical.



- Probability of sitting at an already occupied table $k \propto n_k$ (n_k: # of people sitting at table k)
- Probability of sitting at an unoccupied table $\propto \alpha$ (where α is a novelty hyperparameter)
- Imagine table k is associated with a unique ϕ_k. Then the arragement would look like..



- The table assignment distribution is the same as the DP predictive distribution

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors
- Take a ball. Color it using some color. Put it in an urn.

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors
- Take a ball. Color it using some color. Put it in an urn.
- For each subsequent ball (say number $n+1$), color it using following scheme

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors
- Take a ball. Color it using some color. Put it in an urn.
- For each subsequent ball (say number $n+1$), color it using following scheme
  - Use a new color with probability $\frac{\alpha}{\alpha+n}$

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors
- Take a ball. Color it using some color. Put it in an urn.
- For each subsequent ball (say number $n+1$), color it using following scheme
  - Use a new color with probability $\frac{\alpha}{\alpha+n}$
  - With probability $\frac{n}{\alpha+n}$, pull out a ball randomly from the urn and copy its color

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors
- Take a ball. Color it using some color. Put it in an urn.
- For each subsequent ball (say number $n+1$), color it using following scheme
  - Use a new color with probability $\frac{\alpha}{\alpha+n}$
  - With probability $\frac{n}{\alpha+n}$, pull out a ball randomly from the urn and copy its color
  - Place both balls (chosen and the new one) back to the urn

Pólya-Urn Scheme

- Another metaphor to describe the way $\theta_1, \ldots, \theta_n$ are sequentially generated
- Suppose we have a collection of uncolored ball. We'd like to color them using a set of colors
- Take a ball. Color it using some color. Put it in an urn.
- For each subsequent ball (say number $n+1$), color it using following scheme
  - Use a new color with probability $\frac{\alpha}{\alpha+n}$
  - With probability $\frac{n}{\alpha+n}$, pull out a ball randomly from the urn and copy its color
  - Place both balls (chosen and the new one) back to the urn
- The color assignment scheme has the same distribution as the DP predictive distribution

de Finetti's Theorem and Infinite Exchangeability

- de Finetti's Theorem is one of the most fundamental results in Bayesian statistics

de Finetti's Theorem and Infinite Exchangeability

- de Finetti's Theorem is one of the most fundamental results in Bayesian statistics
- Infinitely Exchangeable Sequence: One for which any finite collection $\theta_1, \ldots, \theta_N$ is exchangeable

de Finetti's Theorem and Infinite Exchangeability

- de Finetti's Theorem is one of the most fundamental results in Bayesian statistics
- Infinitely Exchangeable Sequence: One for which any finite collection $\theta_1, \ldots, \theta_N$ is exchangeable
- Exchangeable: A finite sequence of random variables $\theta_1, \ldots, \theta_N$ is called exchangeable if its joint distribution is invariant under permutations

$$p(\theta_1, \ldots, \theta_N) = p(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(N)})$$

.. for any permutation $\sigma(1), \ldots, \sigma(N)$ of $1, \ldots, N$

de Finetti's Theorem and Infinite Exchangeability

- de Finetti's Theorem is one of the most fundamental results in Bayesian statistics
- Infinitely Exchangeable Sequence: One for which any finite collection $\theta_1, \ldots, \theta_N$ is exchangeable
- Exchangeable: A finite sequence of random variables $\theta_1, \ldots, \theta_N$ is called exchangeable if its joint distribution is invariant under permutations

$$p(\theta_1, \ldots, \theta_N) = p(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(N)})$$

.. for any permutation $\sigma(1), \ldots, \sigma(N)$ of $1, \ldots, N$

- de Finetti's Theorem: For an inf. exchangeable sequence, there exists a random distribution G s.t.

$$p(\theta_1, \ldots, \theta_N) = \int \prod_{i=1}^{N} p(\theta_i | G) dp(G)$$

.. that is, $\theta_1, \ldots, \theta_N$ are i.i.d. given G

de Finetti's Theorem and Infinite Exchangeability

- de Finetti's Theorem is one of the most fundamental results in Bayesian statistics
- Infinitely Exchangeable Sequence: One for which any finite collection $\theta_1, \ldots, \theta_N$ is exchangeable
- Exchangeable: A finite sequence of random variables $\theta_1, \ldots, \theta_N$ is called exchangeable if its joint distribution is invariant under permutations

$$p(\theta_1, \ldots, \theta_N) = p(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(N)})$$

.. for any permutation $\sigma(1), \ldots, \sigma(N)$ of $1, \ldots, N$

- de Finetti's Theorem: For an inf. exchangeable sequence, there exists a random distribution G s.t.

$$p(\theta_1, \ldots, \theta_N) = \int \prod_{i=1}^{N} p(\theta_i | G) dp(G)$$

.. that is, $\theta_1, \ldots, \theta_N$ are i.i.d. given G

- Note that the sequence $\theta_1, \ldots, \theta_N$ generated by the Pólya-Urn/CRP schemes is also exchangeable

de Finetti's Theorem and Infinite Exchangeability

- de Finetti's Theorem is one of the most fundamental results in Bayesian statistics
- Infinitely Exchangeable Sequence: One for which any finite collection $\theta_1, \ldots, \theta_N$ is exchangeable
- Exchangeable: A finite sequence of random variables $\theta_1, \ldots, \theta_N$ is called exchangeable if its joint distribution is invariant under permutations

$$p(\theta_1, \ldots, \theta_N) = p(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(N)})$$

.. for any permutation $\sigma(1), \ldots, \sigma(N)$ of $1, \ldots, N$

- de Finetti's Theorem: For an inf. exchangeable sequence, there exists a random distribution G s.t.

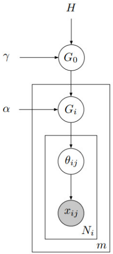
$$p(\theta_1, \ldots, \theta_N) = \int \prod_{i=1}^{N} p(\theta_i | G) dp(G)$$

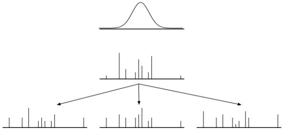
.. that is, $\theta_1, \ldots, \theta_N$ are i.i.d. given G

- Note that the sequence $\theta_1, \ldots, \theta_N$ generated by the Pólya-Urn/CRP schemes is also exchangeable
  - It implies that there must exist such a distribution G (and that is $G \sim \text{DP}(\alpha, G_0)$)

Hierarchical Dirichlet Process (HDP)

- Defines a DP whose base distribution G_0 itself is drawn from another DP



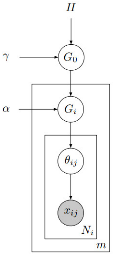


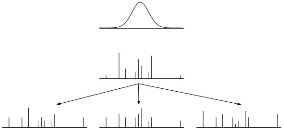
$$\begin{aligned} G_0 &\sim \text{DP}(\gamma, H) \\ G_i &\sim \text{DP}(\alpha, G_0) \\ \theta_{ij}|G_i &\sim G_i \\ x_{ij}|\theta_{ij} &\sim p(x_{ij}|\theta_{ij}) \end{aligned}$$

- Can be used if we would like to cluster m data sets, each using a DP mixture model

Hierarchical Dirichlet Process (HDP)

- Defines a DP whose base distribution G_0 itself is drawn from another DP



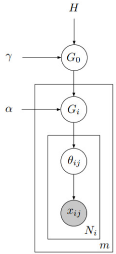


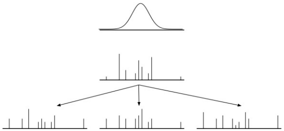
$$\begin{aligned} G_0 &\sim \text{DP}(\gamma, H) \\ G_i &\sim \text{DP}(\alpha, G_0) \\ \theta_{ij}|G_i &\sim G_i \\ x_{ij}|\theta_{ij} &\sim p(x_{ij}|\theta_{ij}) \end{aligned}$$

- Can be used if we would like to cluster m data sets, each using a DP mixture model
- The discreteness of the shared base distribution G_0 enables sharing information across the m clustering problems (reason: because the discreteness allows sharing clusters/atoms)

Hierarchical Dirichlet Process (HDP)

- Defines a DP whose base distribution G_0 itself is drawn from another DP



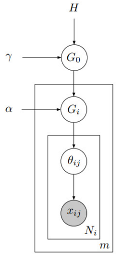


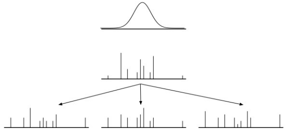
$$
\begin{aligned}
G_0 &\sim \mathrm{DP}(\gamma, H)\\
G_i &\sim \mathrm{DP}(\alpha, G_0)\\
\theta_{ij}|G_i &\sim G_i\\
x_{ij}|\theta_{ij} &\sim p(x_{ij}|\theta_{ij})
\end{aligned}
$$

- Can be used if we would like to cluster m data sets, each using a DP mixture model
- The discreteness of the shared base distribution G_0 enables sharing information across the m clustering problems (reason: because the discreteness allows sharing clusters/atoms)
- Important: If G_0 were a continuous distribution, we won't be able to share atoms (probability of G_i and G_j sharing any atoms will be zero if G_0 is a continuous distribution)

Hierarchical Dirichlet Process (HDP)

- Defines a DP whose base distribution G_0 itself is drawn from another DP





$$
\begin{aligned}
G_0 &\sim \mathrm{DP}(\gamma, H) \\
G_i &\sim \mathrm{DP}(\alpha, G_0) \\
\theta_{ij}|G_i &\sim G_i \\
x_{ij}|\theta_{ij} &\sim p(x_{ij}|\theta_{ij})
\end{aligned}
$$

- Can be used if we would like to cluster m data sets, each using a DP mixture model
- The discreteness of the shared base distribution G_0 enables sharing information across the m clustering problems (reason: because the discreteness allows sharing clusters/atoms)
- Important: If G_0 were a continuous distribution, we won't be able to share atoms (probability of G_i and G_j sharing any atoms will be zero if G_0 is a continuous distribution)
- HDP used in nonparametric Bayesian version of LDA topic model

Some Other Properties/Extensions of DP

- *a priori* expected number of clusters (as per the DP prior) $K = \mathcal{O}(\alpha \log N)$

Some Other Properties/Extensions of DP

- *a priori* expected number of clusters (as per the DP prior) $K = \mathcal{O}(\alpha \log N)$
- Pitman-Yor Process: A variant of DP for which K has a power-law growth $\mathcal{O}(N^d)$, where $0 \leq d < 1$ is an additional "discount" parameter and $\alpha > -d$

Some Other Properties/Extensions of DP

- *a priori* expected number of clusters (as per the DP prior) $K = \mathcal{O}(\alpha \log N)$
- Pitman-Yor Process: A variant of DP for which K has a power-law growth $\mathcal{O}(N^d)$, where $0 \leq d < 1$ is an additional "discount" parameter and $\alpha > -d$
- For the n-th customer, the probabilities are

$$p(\text{table} = k) \quad \propto \quad \frac{n_k - d}{n - 1 + \alpha} \qquad k = 1, \ldots, K$$

Some Other Properties/Extensions of DP

- *a priori* expected number of clusters (as per the DP prior) $K = \mathcal{O}(\alpha \log N)$
- Pitman-Yor Process: A variant of DP for which K has a power-law growth $\mathcal{O}(N^d)$, where $0 \leq d < 1$ is an additional "discount" parameter and $\alpha > -d$
- For the n-th customer, the probabilities are

$$
\begin{aligned}
p(\text{table} = k) &\propto \frac{n_k - d}{n - 1 + \alpha} \qquad k = 1, \ldots, K \\
p(\text{new table}) &\propto \frac{\alpha + dK}{n - 1 + \alpha}
\end{aligned}
$$

Some Other Properties/Extensions of DP

- *a priori* expected number of clusters (as per the DP prior) $K = \mathcal{O}(\alpha \log N)$
- Pitman-Yor Process: A variant of DP for which K has a power-law growth $\mathcal{O}(N^d)$, where $0 \leq d < 1$ is an additional "discount" parameter and $\alpha > -d$
- For the n-th customer, the probabilities are

$$
\begin{aligned}
p(\text{table} = k) &\propto \frac{n_k - d}{n - 1 + \alpha} \qquad k = 1, \ldots, K \\
p(\text{new table}) &\propto \frac{\alpha + dK}{n - 1 + \alpha}
\end{aligned}
$$

- For PY process, probability of occupying existing tables with discounted by d

Some Other Properties/Extensions of DP

- *a priori* expected number of clusters (as per the DP prior) $K = \mathcal{O}(\alpha \log N)$
- Pitman-Yor Process: A variant of DP for which K has a power-law growth $\mathcal{O}(N^d)$, where $0 \leq d < 1$ is an additional "discount" parameter and $\alpha > -d$
- For the n-th customer, the probabilities are

$$
\begin{aligned}
p(\text{table} = k) &\propto \frac{n_k - d}{n - 1 + \alpha} \qquad k = 1, \ldots, K \\
p(\text{new table}) &\propto \frac{\alpha + dK}{n - 1 + \alpha}
\end{aligned}
$$

- For PY process, probability of occupying existing tables with discounted by d
- Creation of new tables is encouraged more and more and K grows

Modeling Binary Matrices with Unbounded Number of Columns

- Assume each observation $\boldsymbol{x}_n \in \mathbb{R}^D$ to be a subset combination of K vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$

$$\boldsymbol{x}_n = \sum_{k=1}^{K} z_{nk} \boldsymbol{a}_k + \epsilon_n$$

where $\boldsymbol{z}_n = [\boldsymbol{z}_{n1}, \ldots, \boldsymbol{z}_{nK}]$ is a binary vector

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Modeling Binary Matrices with Unbounded Number of Columns

- Assume each observation $\boldsymbol{x}_n \in \mathbb{R}^D$ to be a subset combination of K vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$

$$\boldsymbol{x}_n = \sum_{k=1}^{K} z_{nk} \boldsymbol{a}_k + \epsilon_n$$

  where $\boldsymbol{z}_n = [\boldsymbol{z}_{n1}, \ldots, \boldsymbol{z}_{nK}]$ is a binary vector

- For N observations $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, the model can be written as $\mathbf{X} = \mathbf{ZA} + \mathbf{E}$

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Modeling Binary Matrices with Unbounded Number of Columns

- Assume each observation $\boldsymbol{x}_n \in \mathbb{R}^D$ to be a subset combination of K vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$

$$\boldsymbol{x}_n = \sum_{k=1}^{K} z_{nk}\boldsymbol{a}_k + \epsilon_n$$

  where $\boldsymbol{z}_n = [\boldsymbol{z}_{n1}, \ldots, \boldsymbol{z}_{nK}]$ is a binary vector
- For N observations $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, the model can be written as $\mathbf{X} = \mathbf{ZA} + \mathbf{E}$
- Here $\mathbf{Z}$ is $N \times K$ binary matrix (row n is $\boldsymbol{z}_n$), and $\mathbf{A}$ is $K \times D$ matrix (row k is $\boldsymbol{a}_k$)

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

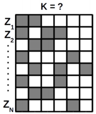
Modeling Binary Matrices with Unbounded Number of Columns

- Assume each observation $\boldsymbol{x}_n \in \mathbb{R}^D$ to be a subset combination of K vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$

$$\boldsymbol{x}_n = \sum_{k=1}^{K} z_{nk} \boldsymbol{a}_k + \epsilon_n$$

where $\boldsymbol{z}_n = [\boldsymbol{z}_{n1}, \ldots, \boldsymbol{z}_{nK}]$ is a binary vector

- For N observations $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, the model can be written as $\mathbf{X} = \mathbf{ZA} + \mathbf{E}$
- Here $\mathbf{Z}$ is $N \times K$ binary matrix (row n is $\boldsymbol{z}_n$), and $\mathbf{A}$ is $K \times D$ matrix (row k is $\boldsymbol{a}_k$)
- How do we learn K? Can do it if we can learn the number of columns in the binary matrix $\mathbf{Z}$



"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

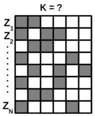
Modeling Binary Matrices with Unbounded Number of Columns

- Assume each observation $\boldsymbol{x}_n \in \mathbb{R}^D$ to be a subset combination of K vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$

$$\boldsymbol{x}_n = \sum_{k=1}^{K} z_{nk}\boldsymbol{a}_k + \epsilon_n$$

  where $\boldsymbol{z}_n = [\boldsymbol{z}_{n1}, \ldots, \boldsymbol{z}_{nK}]$ is a binary vector

- For N observations $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, the model can be written as $\mathbf{X} = \mathbf{ZA} + \mathbf{E}$
- Here $\mathbf{Z}$ is $N \times K$ binary matrix (row n is $\boldsymbol{z}_n$), and $\mathbf{A}$ is $K \times D$ matrix (row k is $\boldsymbol{a}_k$)
- How do we learn K? Can do it if we can learn the number of columns in the binary matrix $\mathbf{Z}$



- A nonparam. Bayesian model called "Indian Buffet Process" (IBP) defines a prior for such matrices

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

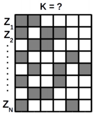
Modeling Binary Matrices with Unbounded Number of Columns

- Assume each observation $\boldsymbol{x}_n \in \mathbb{R}^D$ to be a subset combination of K vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K$

$$\boldsymbol{x}_n = \sum_{k=1}^{K} z_{nk}\boldsymbol{a}_k + \epsilon_n$$

where $\boldsymbol{z}_n = [\boldsymbol{z}_{n1}, \ldots, \boldsymbol{z}_{nK}]$ is a binary vector

- For N observations $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, the model can be written as $\mathbf{X} = \mathbf{ZA} + \mathbf{E}$
- Here $\mathbf{Z}$ is $N \times K$ binary matrix (row n is $\boldsymbol{z}_n$), and $\mathbf{A}$ is $K \times D$ matrix (row k is $\boldsymbol{a}_k$)
- How do we learn K? Can do it if we can learn the number of columns in the binary matrix $\mathbf{Z}$

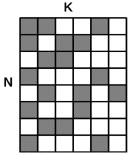


- A nonparam. Bayesian model called "Indian Buffet Process" (IBP) defines a prior for such matrices
- Just like CRP, the IBP is a metaphor to describe the process that generates such matrices

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Modeling Binary Matrices with Finite Many Columns

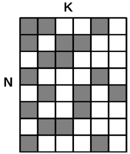
- Consider the generative process of an $N \times K$ binary matrix Z



- Rows denote the N examples, columns denote the K latent features

Modeling Binary Matrices with Finite Many Columns

- Consider the generative process of an $N \times K$ binary matrix Z

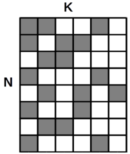


- Rows denote the N examples, columns denote the K latent features
- Assume $\pi_k \in (0,1)$ to be probabiliy of latent feature k being 1

$$z_{nk} \sim \text{Bernoulli}(\pi_k), \qquad \pi_k \sim \text{Beta}(\alpha/K, 1)$$

Modeling Binary Matrices with Finite Many Columns

- Consider the generative process of an $N \times K$ binary matrix Z



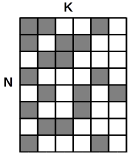
- Rows denote the N examples, columns denote the K latent features
- Assume $\pi_k \in (0,1)$ to be probabiliy of latent feature k being 1

$$z_{nk} \sim \text{Bernoulli}(\pi_k), \qquad \pi_k \sim \text{Beta}(\alpha/K, 1)$$

- Note: All z_{nk}'s are i.i.d. given π_k

Modeling Binary Matrices with Finite Many Columns

- Consider the generative process of an $N \times K$ binary matrix Z



- Rows denote the N examples, columns denote the K latent features
- Assume $\pi_k \in (0, 1)$ to be probabiliy of latent feature k being 1

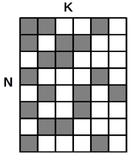
$$z_{nk} \sim \text{Bernoulli}(\pi_k), \qquad \pi_k \sim \text{Beta}(\alpha/K, 1)$$

- Note: All z_{nk}'s are i.i.d. given π_k
- For this model, the conditional probability of $z_{nk} = 1$, given other entries in column k of **Z**

$$p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \int p(z_{nk} = 1 | \pi_k) p(\pi_k | \boldsymbol{z}_{-n,k})$$

Modeling Binary Matrices with Finite Many Columns

- Consider the generative process of an $N \times K$ binary matrix Z



- Rows denote the N examples, columns denote the K latent features
- Assume $\pi_k \in (0,1)$ to be probabiliy of latent feature k being 1

$$z_{nk} \sim \text{Bernoulli}(\pi_k), \qquad \pi_k \sim \text{Beta}(\alpha/K, 1)$$

- Note: All z_{nk}'s are i.i.d. given π_k
- For this model, the conditional probability of $z_{nk} = 1$, given other entries in column k of $\mathbf{Z}$

$$p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \int p(z_{nk} = 1|\pi_k)p(\pi_k|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}} \quad \text{(verify)}$$

where $m_{-n,k} = \sum_{i \neq n} z_{ik}$ denotes how many other entries in column k are equal to 1

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\boldsymbol{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\boldsymbol{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \rightarrow \infty$, we will have $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\boldsymbol{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

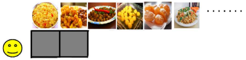
Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\boldsymbol{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

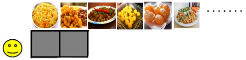
- For the finite K case, we saw that $p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \frac{m_{-n,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0 | \boldsymbol{z}_{-n,k}) = \frac{N - m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
  - Customer 1 selects Poisson(α) dishes



"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

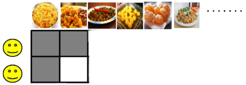
- For the finite K case, we saw that $p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \frac{m_{-n,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0 | \boldsymbol{z}_{-n,k}) = \frac{N - m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
    - Customer 1 selects Poisson(α) dishes
    - The n-th customer selects:



"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\boldsymbol{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
  - Customer 1 selects Poisson(α) dishes
  - The n-th customer selects:
    - Each already selected dish k with probability $m_{-n,k}/n$ (m_k : how many previous customers before n selected dish k)

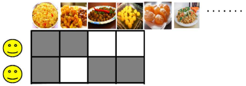


  - Customers = objects; dishes = latent features

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \frac{m_{-n,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1 | \boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0 | \boldsymbol{z}_{-n,k}) = \frac{N - m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
  - Customer 1 selects Poisson(α) dishes
  - The n-th customer selects:
    - Each already selected dish k with probability $m_{-n,k}/n$ (m_k : how many previous customers before n selected dish k)
    - Poisson(α/n) new dishes (this can create new columns in $\mathbf{Z}$)

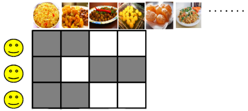


  - Customers = objects; dishes = latent features

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\mathbf{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\mathbf{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\mathbf{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
  - Customer 1 selects Poisson(α) dishes
  - The n-th customer selects:
    - Each already selected dish k with probability $m_{-n,k}/n$ (m_k : how many previous customers before n selected dish k)
    - Poisson(α/n) new dishes (this can create new columns in $\mathbf{Z}$)

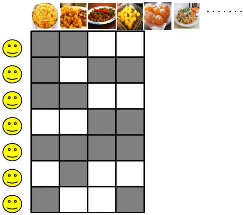


  - Note that as n grows, number of new dishes goes to zero (and the number of columns K converges to some finite number)
  - Customers = objects; dishes = latent features

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

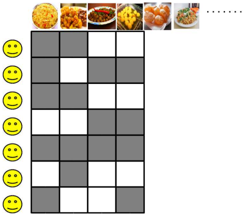
- For the finite K case, we saw that $p(z_{nk} = 1|\mathbf{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\mathbf{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\mathbf{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
  - Customer 1 selects Poisson(α) dishes
  - The n-th customer selects:
    - Each already selected dish k with probability $m_{-n,k}/n$ (m_k : how many previous customers before n selected dish k)
    - Poisson(α/n) new dishes (this can create new columns in $\mathbf{Z}$)
  - Note that as n grows, number of new dishes goes to zero (and the number of columns K converges to some finite number)
  - Customers = objects; dishes = latent features



"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Towards Unbounded Number of Columns

- For the finite K case, we saw that $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}+\frac{\alpha}{K}}{N+\frac{\alpha}{K}}$
- As $K \to \infty$, we will have $p(z_{nk} = 1|\boldsymbol{z}_{-n,k}) = \frac{m_{-n,k}}{N}$ and $p(z_{nk} = 0|\boldsymbol{z}_{-n,k}) = \frac{N-m_{-n,k}}{N}$
- Note that this too exhibits a "rich-gets-richer" phenomenon (just like CRP)
- The Indian Buffet Process is a metaphor for this model. Assume a buffet with infinite dishes
  - Customer 1 selects Poisson(α) dishes
  - The n-th customer selects:
    - Each already selected dish k with probability $m_{-n,k}/n$ (m_k : how many previous customers before n selected dish k)
    - Poisson(α/n) new dishes (this can create new columns in $\mathbf{Z}$)
  - Note that as n grows, number of new dishes goes to zero (and the number of columns K converges to some finite number)
  - Customers = objects; dishes = latent features



- The above can be used as a prior for $\mathbf{Z}$. Refer to (Griffiths and Ghahramani, 2011) for examples and other theoretical details of the model. Also has connections to Beta Processes

"Indian Buffet Process: An Introduction and Review (Griffiths and Ghahramani, 2011)

Another Example: Multiplicative Gamma Process

- Consider the following probabilistic version of SVD

$$\mathbf{X} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top + \mathbf{E}$$

[a]Sparse Bayesian infinite factor models (Bhattacharya and Dunson, 2011)

Another Example: Multiplicative Gamma Process

- Consider the following probabilistic version of SVD

$$\mathbf{X} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top + \mathbf{E}$$

- Consider the following prior on the "singular values" λ_k

"Sparse Bayesian infinite factor models (Bhattacharya and Dunson, 2011)

Another Example: Multiplicative Gamma Process

- Consider the following probabilistic version of SVD

$$\mathbf{X} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top + \mathbf{E}$$

- Consider the following prior on the "singular values" λ_k

$$\lambda_k \sim \mathcal{N}(0, \tau_k^{-1})$$

"Sparse Bayesian infinite factor models (Bhattacharya and Dunson, 2011)

Another Example: Multiplicative Gamma Process

- Consider the following probabilistic version of SVD

$$\mathbf{X} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top + \mathbf{E}$$

- Consider the following prior on the "singular values" λ_k

$$\begin{aligned} \lambda_k &\sim \mathcal{N}(0, \tau_k^{-1}) \\ \tau_k &= \prod_{\ell=1}^{k} \delta_\ell \end{aligned}$$

"Sparse Bayesian infinite factor models (Bhattacharya and Dunson, 2011)

Another Example: Multiplicative Gamma Process

- Consider the following probabilistic version of SVD

$$\mathbf{X} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top + \mathbf{E}$$

- Consider the following prior on the "singular values" λ_k

$$\begin{aligned} \lambda_k &\sim \mathcal{N}(0, \tau_k^{-1}) \\ \tau_k &= \prod_{\ell=1}^{k} \delta_\ell \\ \delta_\ell &\sim \text{Gamma}(\alpha, 1) \quad \text{where } \alpha > 1 \end{aligned}$$

"Sparse Bayesian infinite factor models (Bhattacharya and Dunson, 2011)

Another Example: Multiplicative Gamma Process

- Consider the following probabilistic version of SVD

$$\mathbf{X} = \sum_{k=1}^{K} \lambda_k \boldsymbol{u}_k \boldsymbol{v}_k^\top + \mathbf{E}$$

- Consider the following prior on the "singular values" λ_k

$$\begin{aligned} \lambda_k &\sim \mathcal{N}(0, \tau_k^{-1}) \\ \tau_k &= \prod_{\ell=1}^{k} \delta_\ell \\ \delta_\ell &\sim \text{Gamma}(\alpha, 1) \quad \text{where } \alpha > 1 \end{aligned}$$

- Note that as k becomes large, τ_k gets larger and larger and λ_k shrinks to zero

"Sparse Bayesian infinite factor models (Bhattacharya and Dunson, 2011)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to “DP-means” (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero

“Revisiting k-means: New Algorithms via Bayesian Nonparametrics” (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero
  - E.g., in DP mixture model with Gaussian clusters, take $\sigma^2 \to 0$

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero
  - E.g., in DP mixture model with Gaussian clusters, take $\sigma^2 \to 0$
- The data to cluster assignments in the DP-means algorithm look like

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero
  - E.g., in DP mixture model with Gaussian clusters, take $\sigma^2 \to 0$
- The data to cluster assignments in the DP-means algorithm look like
  - Assign $\boldsymbol{x}_n$ to the closest existing cluster k_* if $||\boldsymbol{x}_n - \mu_{k_*}|| \leq \rho$

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero
  - E.g., in DP mixture model with Gaussian clusters, take $\sigma^2 \rightarrow 0$
- The data to cluster assignments in the DP-means algorithm look like
  - Assign $\boldsymbol{x}_n$ to the closest existing cluster k_* if $||\boldsymbol{x}_n - \mu_{k_*}|| \leq \rho$
  - Otherwise, assign $\boldsymbol{x}_*$ to a new cluster and set $\mu_{K+1} = \boldsymbol{x}_n$

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero
  - E.g., in DP mixture model with Gaussian clusters, take $\sigma^2 \rightarrow 0$
- The data to cluster assignments in the DP-means algorithm look like
  - Assign $\boldsymbol{x}_n$ to the closest existing cluster k_* if $||\boldsymbol{x}_n - \mu_{k_*}|| \leq \rho$
  - Otherwise, assign $\boldsymbol{x}_*$ to a new cluster and set $\mu_{K+1} = \boldsymbol{x}_n$
- For more details, please refer to Kulis and Jordan (2012) and Broderick (2013)

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

NPBayes-inspired Simpler Non-probabilistic Models

- Many NPBayes models can be reduced to simpler non-probabilistic models with NPBayes flavor
- Example: DP Mixture Models reduced to "DP-means" (akin to K-means with unbounded clusters)
- Such simplications are based on small-variance asymptotics (SVA)
  - Basically, take the noise variance of observation model to zero
  - E.g., in DP mixture model with Gaussian clusters, take $\sigma^2 \rightarrow 0$
- The data to cluster assignments in the DP-means algorithm look like
  - Assign $\boldsymbol{x}_n$ to the closest existing cluster k_* if $||\boldsymbol{x}_n - \mu_{k_*}|| \leq \rho$
  - Otherwise, assign $\boldsymbol{x}_*$ to a new cluster and set $\mu_{K+1} = \boldsymbol{x}_n$
- For more details, please refer to Kulis and Jordan (2012) and Broderick (2013)
- Many complex NPBayes models have been simplified using small-variance asymptotics idea

"Revisiting k-means: New Algorithms via Bayesian Nonparametrics" (Kulis and Jordan, 2012), MAD-Bayes: MAP-based Asymptotic Derivations from Bayes (Broderick et al, 2013)

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^{L}$

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^L$
- Other criteria such as Akaike or Bayesian Information Criteria are also commonly used

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^L$
- Other criteria such as Akaike or Bayesian Information Criteria are also commonly used
  - Usually defined as a sum of negative log-lik. and model size (models with smaller values preferred)

$$
\begin{aligned}
AIC &= 2k - 2 \times \text{log-lik} \\
BIC &= k \log N - 2 \times \text{log-lik}
\end{aligned}
$$

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^{L}$
- Other criteria such as Akaike or Bayesian Information Criteria are also commonly used
  - Usually defined as a sum of negative log-lik. and model size (models with smaller values preferred)

$$
\begin{aligned}
AIC &= 2k - 2 \times \text{log-lik} \\
BIC &= k \log N - 2 \times \text{log-lik}
\end{aligned}
$$

  where k denotes the number of parameters of the model, N denotes number of data points

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^{L}$
- Other criteria such as Akaike or Bayesian Information Criteria are also commonly used
  - Usually defined as a sum of negative log-lik. and model size (models with smaller values preferred)

$$AIC = 2k - 2 \times \text{log-lik}$$
$$BIC = k \log N - 2 \times \text{log-lik}$$

  where k denotes the number of parameters of the model, N denotes number of data points
- However, marginal likelihood, AIC/BIC, etc. try multiple models and then choose the best

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^L$
- Other criteria such as Akaike or Bayesian Information Criteria are also commonly used
  - Usually defined as a sum of negative log-lik. and model size (models with smaller values preferred)

$$
\begin{aligned}
AIC &= 2k - 2 \times \text{log-lik} \\
BIC &= k \log N - 2 \times \text{log-lik}
\end{aligned}
$$

  where k denotes the number of parameters of the model, N denotes number of data points
- However, marginal likelihood, AIC/BIC, etc. try multiple models and then choose the best
- In contrast, NPBayes models learn a single model having an unbounded complexity

Some Comments

- Nonparametric Bayesian models have been widely used in several applications
  - Clustering, dim-red, regression/classification, time-series models such as HMM, and many others
- Nonparametric Bayesian models are not the only way to learn the right model size
- Marginal likelihood $p(\mathcal{D}|\mathcal{M})$ can be used for model selection from a set of models $\{\mathcal{M}_i\}_{i=1}^{L}$
- Other criteria such as Akaike or Bayesian Information Criteria are also commonly used
  - Usually defined as a sum of negative log-lik. and model size (models with smaller values preferred)

$$\begin{aligned} AIC &= 2k - 2 \times \text{log-lik} \\ BIC &= k \log N - 2 \times \text{log-lik} \end{aligned}$$

  where k denotes the number of parameters of the model, N denotes number of data points
- However, marginal likelihood, AIC/BIC, etc. try multiple models and then choose the best
- In contrast, NPBayes models learn a single model having an unbounded complexity
  - Also natural for streaming data where model selection is difficult/impractical to perform