An introduction to Bayesian nonparametrics
Lecture 1: The Dirichlet process

Sinead Williamson

MLSS Madrid 2018

THE UNIVERSITY OF TEXAS AT AUSTIN

Lecture Overview

- Lecture 1: The Dirichlet process
  - What is Bayesian nonparametrics?
  - From Dirichlet distribution to Dirichlet Process
  - Representations
  - Inference
- Lecture 2: The Indian buffet process
- Lecture 3: Hierarchical nonparametric models

What is Bayesian nonparametrics?

- In this summer school, you've seen various examples of Bayesian modeling.

- General framework:
  - Come up with a class of models, parametrized by some set of parameters Θ.
  - Place a prior distribution over the parameters.
  - Update our *posterior* distribution as we see observations.

What is Bayesian nonparametrics?

- In this summer school, you've seen various examples of Bayesian modeling.

- General framework:
  - Come up with a class of models, parametrized by some set of parameters Θ.
  - Place a prior distribution over the parameters.
  - Update our *posterior* distribution as we see observations.

- Challenge... how to choose a prior?

- We want to capture intuitions about the data, while minimizing erroneous assumptions... this can be hard!
  - How to choose the number of topics to model the New York Times?
  - What if our test set contains features not present in our training set?

What is Bayesian nonparametrics?

- A *parametric* Bayesian model is one with a fixed, pre-specified number of global parameters:
    - Bayesian linear regression: $y_i \sim \text{Normal}(\mathbf{x}_i^T \boldsymbol{\beta}, \sigma^2)$, $\boldsymbol{\beta}$ is of fixed size.
    - Mixture of K Gaussians: K means, K covariances, one probability vector.

What is Bayesian nonparametrics?

- A *parametric* Bayesian model is one with a fixed, pre-specified number of global parameters:
  - Bayesian linear regression: $y_i \sim \text{Normal}(\mathbf{x}_i^T \boldsymbol{\beta}, \sigma^2)$, $\boldsymbol{\beta}$ is of fixed size.
  - Mixture of K Gaussians: K means, K covariances, one probability vector.

- A *nonparametric* Bayesian model is *not* a model with no parameters...
- It is a model where the number of parameters can grow with dataset size.

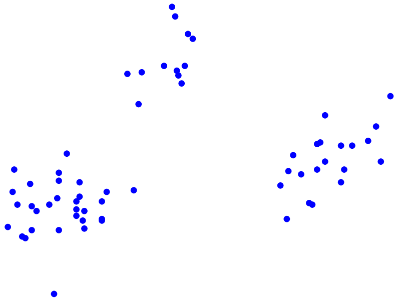
What is Bayesian nonparametrics?

- A *parametric* Bayesian model is one with a fixed, pre-specified number of global parameters:
  - Bayesian linear regression: $y_i \sim \text{Normal}(\mathbf{x}_i^T \boldsymbol{\beta}, \sigma^2)$, $\boldsymbol{\beta}$ is of fixed size.
  - Mixture of K Gaussians: K means, K covariances, one probability vector.

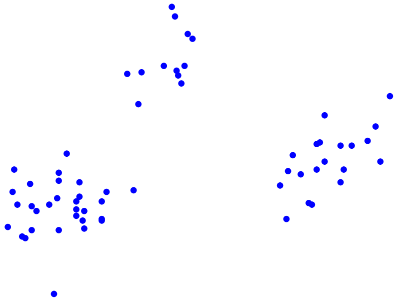
- A *nonparametric* Bayesian model is *not* a model with no parameters...
- It is a model where the number of parameters can grow with dataset size.

- We achieve this by allowing an infinite number of parameters *a priori*.
- However, a finite data set will only ever use a finite number of data points.
  - ~~Bayesian linear regression~~ Gaussian processes – we need infinitely many values to pin down the function.
  - ~~Mixture of K Gaussians~~ Dirichlet process mixture model – infinitely many mixture components *a priori*.

Bayesian parametric models for clustering data

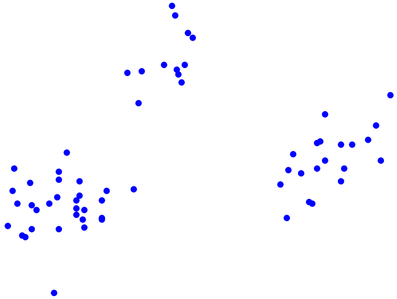


Bayesian parametric models for clustering data



- One obvious model: Mixture of three Gaussians, parametrized by a probability vector $\pi = (\pi_1, \pi_2, \pi_3)$, three means μ_1, μ_2, μ_3, three covariances $\Sigma_1, \Sigma_2, \Sigma_3$.
- For each data point,
  - Sample cluster indicator $z_i \sim \boldsymbol{\pi}$
  - Sample $x_i \sim \text{Normal}(\mu_{z_i}, \Sigma_{z_i})$

Bayesian parametric models for clustering data



- One obvious model: Mixture of three Gaussians, parametrized by a probability vector $\pi = (\pi_1, \pi_2, \pi_3)$, three means μ_1, μ_2, μ_3, three covariances $\Sigma_1, \Sigma_2, \Sigma_3$.
- For each data point,
  - Sample cluster indicator $z_i \sim \boldsymbol{\pi}$
  - Sample $x_i \sim \text{Normal}(\mu_{z_i}, \Sigma_{z_i})$
- This gives us a likelihood

$$p(x_1, \ldots, x_N | \boldsymbol{\pi}, \{\mu_k\}, \{\Sigma_k\}) = \prod_{i=1}^{N} \sum_{k=1}^{3} \pi_k \text{Normal}(x_n | \mu_k, \Sigma_k)$$

Bayesian mixture models

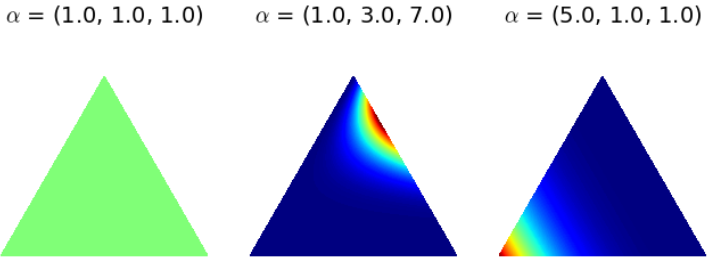
- How to choose the mixing weights $\boldsymbol{\pi}$ and the mixture parameters $\{\mu_k, \Sigma_k\}$?
- Bayesian choice: Put a prior on them and integrate out:

$$p(x_1, \dots, x_N) = \int\int\int \underbrace{p(x_1, \dots, x_N | \boldsymbol{\pi}, \{\mu_k\}, \{\Sigma_k\})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\pi}) \prod_{k=1}^{3} p(\mu_k, \Sigma_k)}_{\text{prior}} \, d\boldsymbol{\pi} d\mu_k d\Sigma_k$$

- Where possible, use conjugate priors:
  - Gaussian-inverse Wishart for mixture parameters
  - Dirichlet distribution for mixing weights
- Let's think about the Dirichlet distribution for a bit...

The Dirichlet distribution: A distribution over probability vectors

- The Dirichlet distribution is a distribution over the $(K-1)$-dimensional simplex – in other words, the space of all K-dimensional probability vectors.
- Parametrized by $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$ such that each $\alpha_k \geq 0$ and $\sum_k \alpha_k > 0$.
- The expected value of a Dirichlet random variable π is given by $\mathbb{E}[\pi] = \frac{(\alpha_1, \dots, \alpha_K)}{\sum_k \alpha_k}$



α = (1.0, 1.0, 1.0)

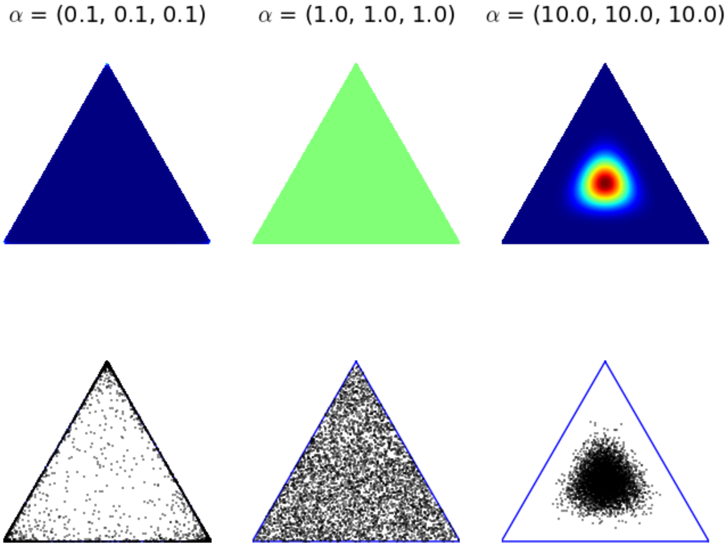
α = (1.0, 3.0, 7.0)

α = (5.0, 1.0, 1.0)

- The Dirichlet(1,1,1) distribution is the uniform distribution on the 2-simplex.
- The Dirichlet(α, β) distribution is the Beta(α, β) distribution.

The Dirichlet distribution: A distribution over probability vectors

- The magnitude $\sum_k \alpha_k$ of the parameters acts as an inverse variance.
- Larger magnitude → more similar samples.
- Smaller magnitude → sparser samples.



The Dirichlet distribution: Conjugacy to the multinomial

- There are a number of distributions over probability vectors... but the Dirichlet is nice because it's conjugate to the multinomial.
- If $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$, then

$$p(\pi) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1}$$

The Dirichlet distribution: Conjugacy to the multinomial

- There are a number of distributions over probability vectors... but the Dirichlet is nice because it's conjugate to the multinomial.
- If $\pi \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K)$, then

$$p(\pi) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1}$$

- If $x_i \overset{iid}{\sim} \pi$ for $i = 1, \dots, N$, then

$$p(x_1, \dots, x_N | \pi) = \frac{N!}{m_1! \cdots m_K!} \prod_{k=1}^K \pi_k^{m_k}$$

where $m_k = \sum_i \mathbb{I}(x_i = k)$

The Dirichlet distribution: Conjugacy to the multinomial

- There are a number of distributions over probability vectors... but the Dirichlet is nice because it's conjugate to the multinomial.
- If $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$, then

$$p(\pi) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \pi_k^{\alpha_k - 1}$$

- If $x_i \overset{iid}{\sim} \pi$ for $i = 1, \ldots, N$, then

$$p(x_1, \ldots, x_N | \pi) = \frac{N!}{m_1! \cdots m_K!} \prod_{k=1}^K \pi_k^{m_k}$$

where $m_k = \sum_i \mathbb{I}(x_i = k)$

- So, the posterior takes the form

$$p(\pi | x_1, \ldots, x_N) \propto p(x_1 \ldots x_N | \pi) p(\pi) \propto \prod_{k=1}^K \pi_k^{\alpha_k + m_k - 1}$$

so, $p(\pi | x_1, \ldots, x_N) = \text{Dirichlet}(\pi | \alpha_1 + m_1, \ldots, \alpha_K + m_K)$

Combining and splitting elements

The Dirichlet distribution has a number of nice properties...

[Agglomeration property]

- We can get a $K-1$-dimensional Dirichlet distribution from a K-dimensional distribution.
- If

$$(\pi_1, \dots, \pi_K) \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K)$$

then

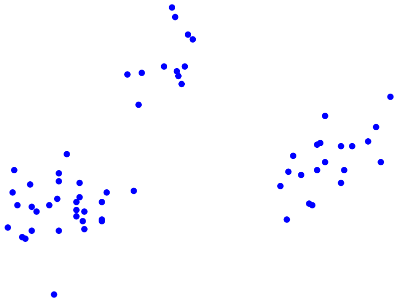
$$(\pi_1 + \pi_2, \pi_3, \dots, \pi_K) \sim \text{Dirichlet}(\alpha_1 + \alpha_2, \alpha_3, \dots, \alpha_K)$$

Combining and splitting elements

[Decimation property]

- We can get a $K+1$-dimensional Dirichlet$(\alpha_1 b, \alpha_1(1-b), \alpha_2, \dots, \alpha_K)$ distribution from a K-dimensional Dirichlet$(\alpha_1, \alpha_2, \dots, \alpha_K)$ distribution.
- If
$$(\pi_1, \dots, \pi_K) \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K)$$
and
$$\theta \sim \text{Beta}(\alpha_1 b, \alpha_1(1-b)),\ 0 < b < 1$$
then
$$(\pi_1\theta_1, \pi_1(1-\theta_1), \pi_2, \dots, \pi_K) \sim \text{Dirichlet}(\alpha_1 b, \alpha_1(1-b) \dots, \alpha_K)$$

Returning to the Bayesian mixture of Gaussians



- Now we know about the Dirichlet distribution... we will return to our mixture of Gaussians.
  - Sample $\pi \sim$ Dirichlet$(\alpha_1, \alpha_2, \alpha_3)$
  - For each cluster, sample $\mu_k, \Sigma_k \sim$ Normal-inverse Wishart$(\mu_0, \lambda, \Psi, \nu)$
  - For the ith data point...
    - Sample a cluster indicator $z_i \sim \pi$.
    - Sample a location $x_i \sim$ Normal$(\mu_{z_i}, \Sigma_{z_i})$

Bayesian mixture of Gaussians: Posterior inference

- We can make use of conjugacy to sample from the posteriors of π, μ and Σ.
- Conditioned on the indicators z_i, then

$$\pi | z_1, \dots, z_N \sim \text{Dirichlet}(\alpha_1 + m_1, \alpha_2 + m_2, \alpha_3 + m_3)$$

where $m_k = \sum_{i=1}^N \mathbb{I}(x_i = k)$.

Bayesian mixture of Gaussians: Posterior inference

- We can make use of conjugacy to sample from the posteriors of π, μ and Σ.
- Conditioned on the indicators z_i, then

$$\pi|z_1, \ldots, z_N \sim \text{Dirichlet}(\alpha_1 + m_1, \alpha_2 + m_2, \alpha_3 + m_3)$$

  where $m_k = \sum_{i=1}^{N} \mathbb{I}(x_i = k)$.
- Conditioned on π and the μ_k and Σ_k, we can sample from the posteriors of z:

$$P(z_i = k|x_i, \pi, \mu_k, \Sigma_k) \propto \pi_k \text{Normal}(x_i; \mu_k, \Sigma_k)$$

Bayesian mixture of Gaussians: Posterior inference

- We can make use of conjugacy to sample from the posteriors of π, μ and Σ.
- Conditioned on the indicators z_i, then

$$\pi | z_1, \dots, z_N \sim \text{Dirichlet}(\alpha_1 + m_1, \alpha_2 + m_2, \alpha_3 + m_3)$$

  where $m_k = \sum_{i=1}^N \mathbb{I}(x_i = k)$.
- Conditioned on π and the μ_k and Σ_k, we can sample from the posteriors of z:

$$P(z_i = k | x_i, \pi, \mu_k, \Sigma_k) \propto \pi_k \text{Normal}(x_i; \mu_k, \Sigma_k)$$

- Conditioned on all the observations x_i s.t. $z_i = k$, we can sample from the posteriors for μ and Σ as in a standard normal model.

Bayesian mixture of Gaussians: Posterior inference

- We can make use of conjugacy to sample from the posteriors of π, μ and Σ.
- Conditioned on the indicators z_i, then

$$\pi | z_1, \ldots, z_N \sim \text{Dirichlet}(\alpha_1 + m_1, \alpha_2 + m_2, \alpha_3 + m_3)$$

  where $m_k = \sum_{i=1}^N \mathbb{I}(x_i = k)$.
- Conditioned on π and the μ_k and Σ_k, we can sample from the posteriors of z:

$$P(z_i = k | x_i, \pi, \mu_k, \Sigma_k) \propto \pi_k \text{Normal}(x_i; \mu_k, \Sigma_k)$$

- Conditioned on all the observations x_i s.t. $z_i = k$, we can sample from the posteriors for μ and Σ as in a standard normal model.
- Alternatively, instead of explicitly sampling π, we can integrate it out.

An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

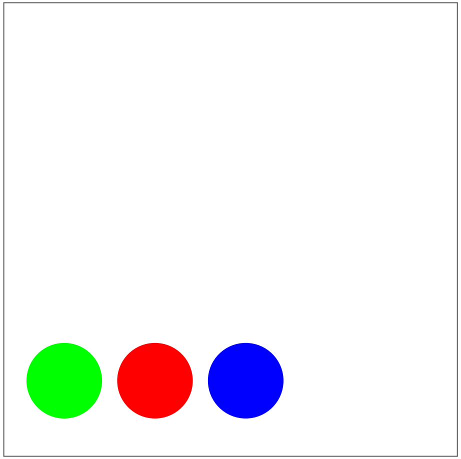
$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.

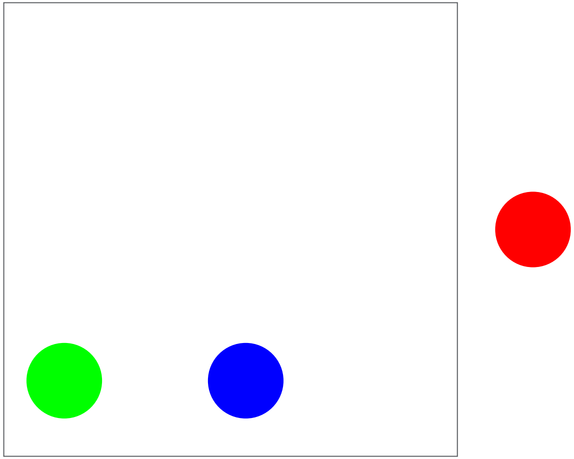


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.

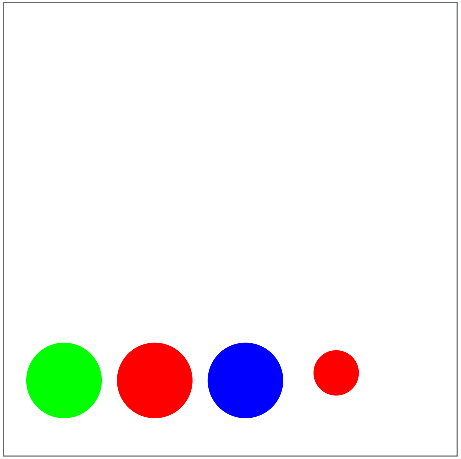


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.

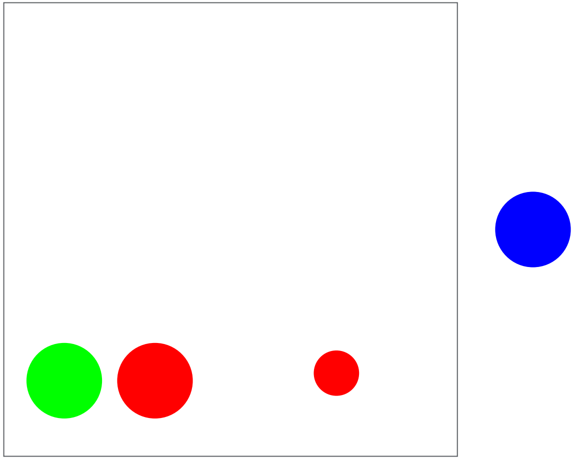


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

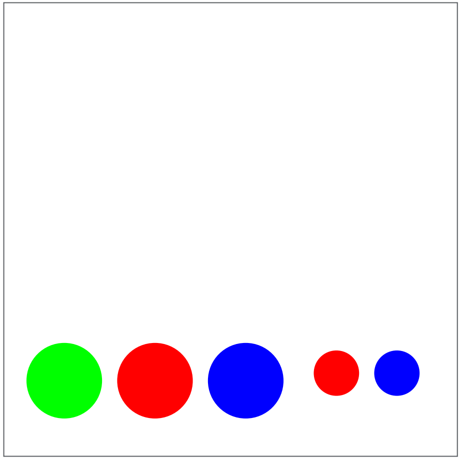


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

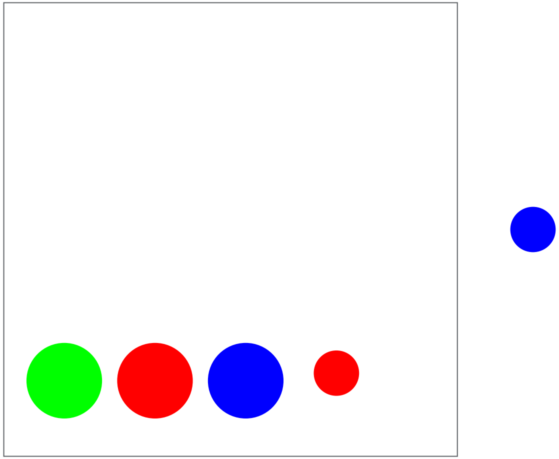


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

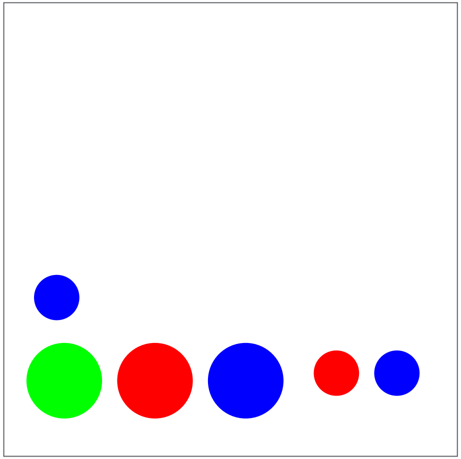


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

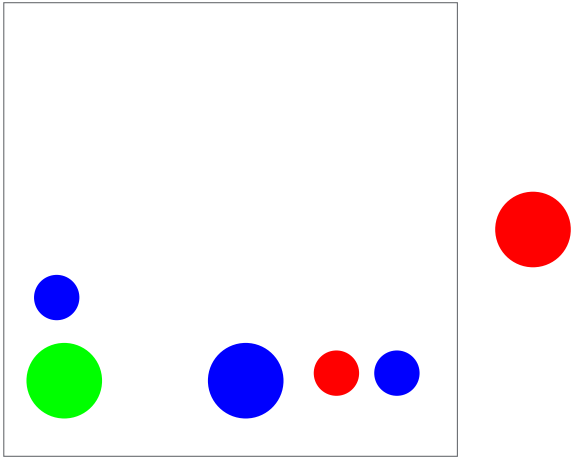


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

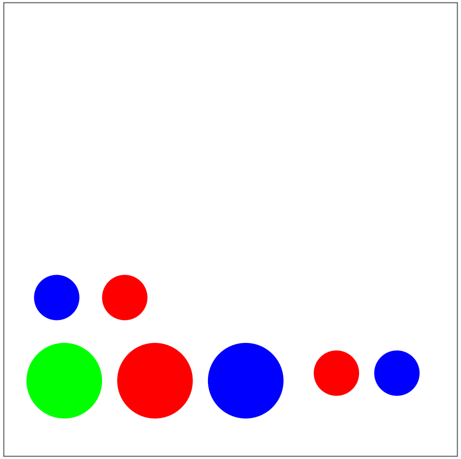


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

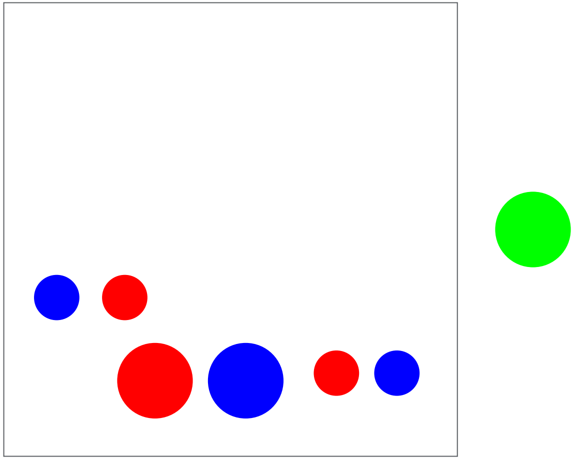


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.

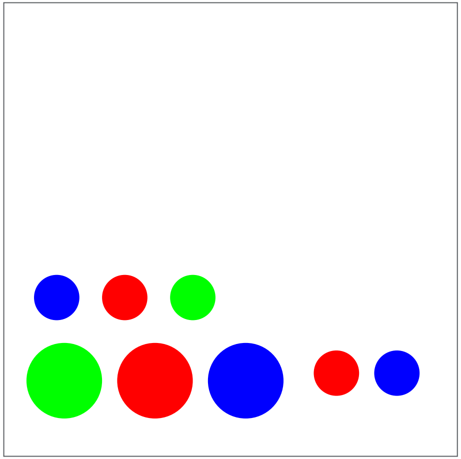


An urn representation

- Conditioned on π, the cluster indicators are independent: $p(z_i = k|\pi) = \pi_k$.
- When we integrate out π, they are no longer independent, and we have

$$p(z_i = k|z_{1:i-1}) = \int p(z_i = k|\pi)p(\pi|z_{1:i-1})d\pi = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- We can describe this using an urn analogy.
- Start with K different colored balls, each of size α_k.
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.
- Repeat to build up dataset.



Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?

Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?
- Of course! More red balls $\rightarrow$ more likely to pick a red ball.
- The balls are *not* i.i.d.

Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?
- Of course! More red balls $\rightarrow$ more likely to pick a red ball.
- The balls are *not* i.i.d.
- Does changing the order of the sequence matter? Does $p(r,r,r,b,g) = p(g,r,b,r,r)$?

Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?
- Of course! More red balls → more likely to pick a red ball.
- The balls are *not* i.i.d.
- Does changing the order of the sequence matter? Does $p(r,r,r,b,g) = p(g,r,b,r,r)$?
- No! But this might not be as obvious... so we can double check

Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?
- Of course! More red balls → more likely to pick a red ball.
- The balls are *not* i.i.d.
- Does changing the order of the sequence matter? Does $p(r,r,r,b,g) = p(g,r,b,r,r)$?
- No! But this might not be as obvious... so we can double check

$$p(r,r,r,b,g) = \frac{\alpha_r}{\sum_k \alpha_k}\frac{\alpha_r+1}{\sum_k \alpha_k + 1}\frac{\alpha_r+2}{\sum_k \alpha_k+2}\frac{\alpha_b}{\sum_k \alpha_k+3}\frac{\alpha_g}{\sum_k \alpha_k+3}$$

Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?
- Of course! More red balls $\rightarrow$ more likely to pick a red ball.
- The balls are *not* i.i.d.
- Does changing the order of the sequence matter? Does $p(r,r,r,b,g) = p(g,r,b,r,r)$?
- No! But this might not be as obvious... so we can double check

$$p(r,r,r,b,g) = \frac{\alpha_r}{\sum_k \alpha_k} \frac{\alpha_r + 1}{\sum_k \alpha_k + 1} \frac{\alpha_r + 2}{\sum_k \alpha_k + 2} \frac{\alpha_b}{\sum_k \alpha_k + 3} \frac{\alpha_g}{\sum_k \alpha_k + 3}$$

$$p(g,r,b,r,r) = \frac{\alpha_g}{\sum_k \alpha_k} \frac{\alpha_r}{\sum_k \alpha_k + 1} \frac{\alpha_b}{\sum_k \alpha_k + 2} \frac{\alpha_r + 1}{\sum_k \alpha_k + 3} \frac{\alpha_r + 2}{\sum_k \alpha_k + 3}$$

Exchangeability

- Does the probability of the ith ball being red depend on how many of the first $i-1$ balls are red?
- Of course! More red balls → more likely to pick a red ball.
- The balls are *not* i.i.d.
- Does changing the order of the sequence matter? Does $p(r,r,r,b,g) = p(g,r,b,r,r)$?
- No! But this might not be as obvious... so we can double check

$$p(r,r,r,b,g) = \frac{\alpha_r}{\sum_k \alpha_k}\frac{\alpha_r+1}{\sum_k \alpha_k+1}\frac{\alpha_r+2}{\sum_k \alpha_k+2}\frac{\alpha_b}{\sum_k \alpha_k+3}\frac{\alpha_g}{\sum_k \alpha_k+3}$$

$$p(g,r,b,r,r) = \frac{\alpha_g}{\sum_k \alpha_k}\frac{\alpha_r}{\sum_k \alpha_k+1}\frac{\alpha_b}{\sum_k \alpha_k+2}\frac{\alpha_r+1}{\sum_k \alpha_k+3}\frac{\alpha_r+2}{\sum_k \alpha_k+3}$$

- This property is known as exchangeability – the probability of a sequence is invariant to permutations

Why does exchangeability matter?

- Exchangeability allows us treat every data point as if it were the last one that we've seen.
- We know that

$$p(z_i = k|z_{1:i-1}) = \frac{\sum_{j=1}^{i-1} \mathbb{I}(z_j = k) + \alpha_k}{i - 1 + \sum_k \alpha_k}$$

- Instead of just conditioning on the first $i-1$ data points, we can pretend the ith data point is actually the last one we saw, so that

$$p(z_i = k|z_{-i}) = \frac{\sum_{j \neq i} \mathbb{I}(z_j = k) + \alpha_k}{N - 1 + \sum_k \alpha_k}$$

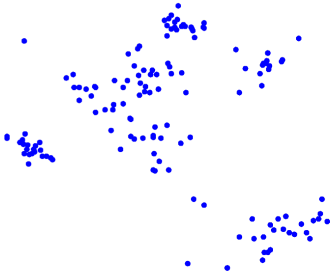
- We can combine this with the cluster likelihood to get the posterior distribution

$$p(z_i = k|x_i z_{-i}, \{\mu_k\}, \{\Sigma_k\}) \propto \frac{\sum_{j \neq i} \mathbb{I}(z_j = k) + \alpha_k}{N - 1 + \sum_k \alpha_k} \text{Normal}(x_i|\mu_k, \Sigma_k)$$

- This makes it easy to construct a Gibbs sampler!

Choosing the number of clusters

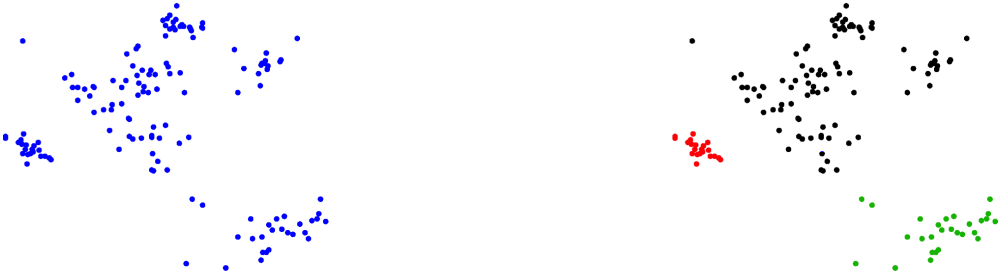
- The Dirichlet distribution is a great choice when there is a clear, fixed number of clusters... but sometimes that's not the case.
- Sometimes it's hard to tell what the right number of clusters is...



- Even if we have a good idea of how many clusters we have today... what if we see new clusters tomorrow?
- We should make sure we have more clusters than we actually need.

Choosing the number of clusters

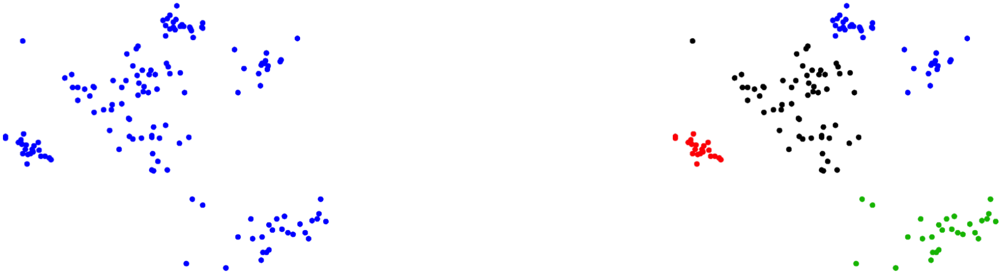
- The Dirichlet distribution is a great choice when there is a clear, fixed number of clusters... but sometimes that's not the case.
- Sometimes it's hard to tell what the right number of clusters is...



- Even if we have a good idea of how many clusters we have today... what if we see new clusters tomorrow?
- We should make sure we have more clusters than we actually need.

Choosing the number of clusters

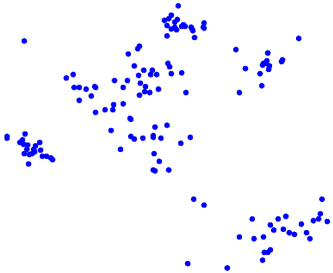
- The Dirichlet distribution is a great choice when there is a clear, fixed number of clusters... but sometimes that's not the case.
- Sometimes it's hard to tell what the right number of clusters is...

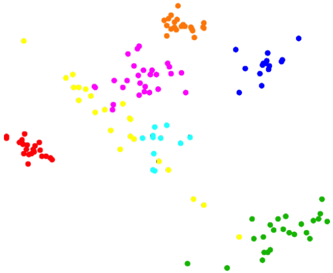


- Even if we have a good idea of how many clusters we have today... what if we see new clusters tomorrow?
- We should make sure we have more clusters than we actually need.

Choosing the number of clusters

- The Dirichlet distribution is a great choice when there is a clear, fixed number of clusters... but sometimes that's not the case.
- Sometimes it's hard to tell what the right number of clusters is...





- Even if we have a good idea of how many clusters we have today... what if we see new clusters tomorrow?
- We should make sure we have more clusters than we actually need.

Bayesian nonparametric mixture models

- The finite mixture model had K mixture components:

$$p(x_n|\pi, \{\mu_k\}, \{\Sigma_k\}) = \sum_{k=1}^{K} \pi_k \text{Normal}(x_n|\mu_k, \Sigma_k)$$

Bayesian nonparametric mixture models

- The finite mixture model had K mixture components:

$$p(x_n|\pi, \{\mu_k\}, \{\Sigma_k\}) = \sum_{k=1}^{K} \pi_k \text{Normal}(x_n|\mu_k, \Sigma_k)$$

- To make sure we never run out of clusters, no matter how many data points we see, we need (countably) infinite clusters!

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

- N data points will use at most N clusters.
- However, if some of the π_k are bigger than others, there will probably be fewer than N.
- So, a finite data set will always use a finite—but random—number of clusters.

Bayesian nonparametric mixture models

- The finite mixture model had K mixture components:

$$p(x_n|\pi, \{\mu_k\}, \{\Sigma_k\}) = \sum_{k=1}^{K} \pi_k \mathsf{Normal}(x_n|\mu_k, \Sigma_k)$$

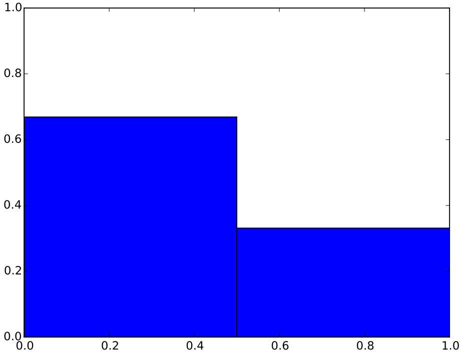
- To make sure we never run out of clusters, no matter how many data points we see, we need (countably) infinite clusters!

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

- N data points will use at most N clusters.
- However, if some of the π_k are bigger than others, there will probably be fewer than N.
- So, a finite data set will always use a finite—but random—number of clusters.
- How to choose an appropriate prior?
- We want something *like* a Dirichlet prior... but with an infinite number of components.

Constructing an appropriate prior

- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$



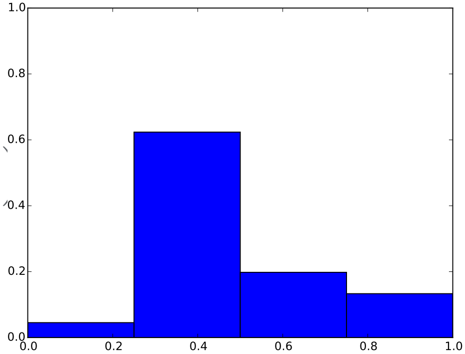
Constructing an appropriate prior

- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\boldsymbol{\pi}^{(4)} = \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)$$

$$\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

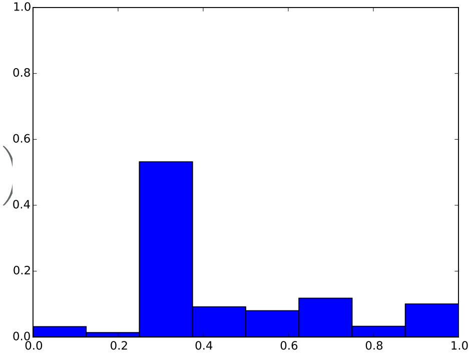


Constructing an appropriate prior

- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\begin{aligned}\theta_1, \theta_2 &\overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)\\ \boldsymbol{\pi}^{(4)} &= \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)\\ &\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)\end{aligned}$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.

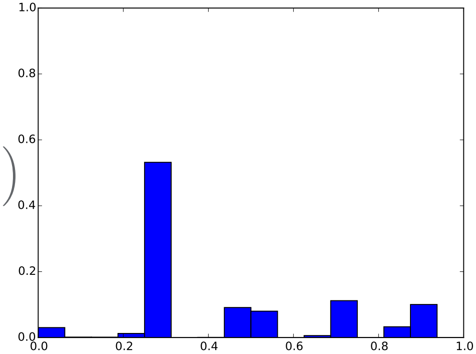


Constructing an appropriate prior

- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\begin{aligned}\theta_1, \theta_2 &\overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)\\ \boldsymbol{\pi}^{(4)} &= \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)\\ &\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)\end{aligned}$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.



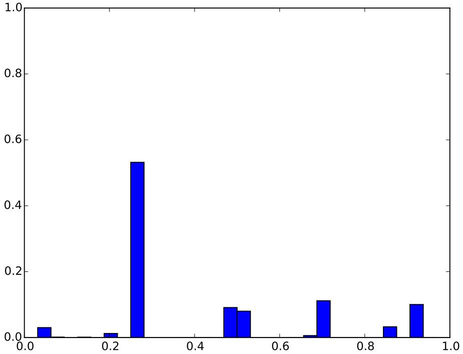
Constructing an appropriate prior

- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\begin{aligned}\boldsymbol{\pi}^{(4)} &= \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right) \\ &\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)\end{aligned}$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.



Constructing an appropriate prior

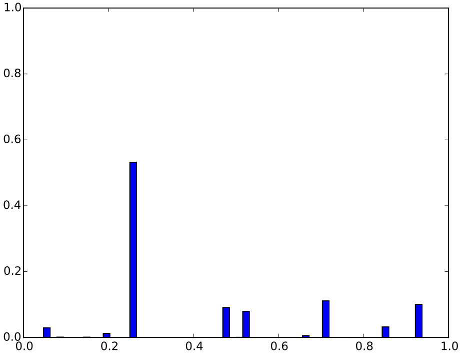
- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\boldsymbol{\pi}^{(4)} = \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)$$

$$\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.



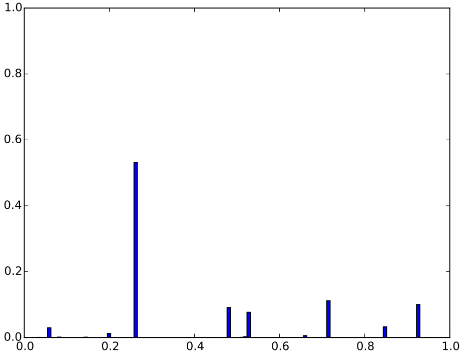
Constructing an appropriate prior

- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\begin{aligned}\boldsymbol{\pi}^{(4)} &= \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)\\ &\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)\end{aligned}$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \dots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.



Constructing an appropriate prior

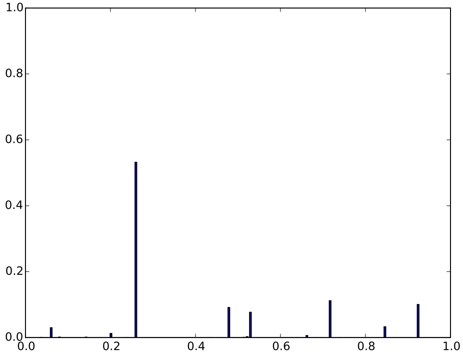
- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\boldsymbol{\pi}^{(4)} = \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)$$

$$\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.



Constructing an appropriate prior

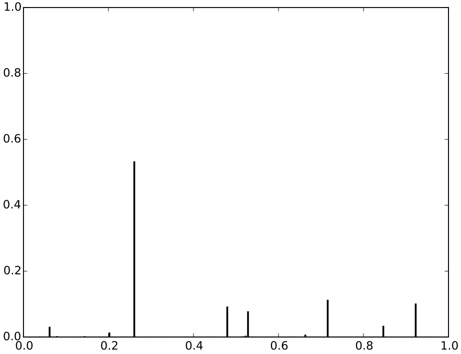
- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\boldsymbol{\pi}^{(4)} = \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)$$

$$\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \dots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.



Constructing an appropriate prior

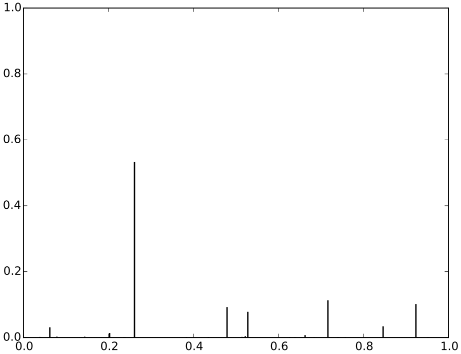
- Start off with $\boldsymbol{\pi}^{(2)} = (\pi_1^{(2)}, \pi_2^{(2)}) \sim \text{Dirichlet}\left(\frac{\alpha}{2}, \frac{\alpha}{2}\right)$
- Split each component according to our beta splitting rule:

$$\theta_1, \theta_2 \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

$$\boldsymbol{\pi}^{(4)} = \left(\theta_1\pi_1^{(2)}, (1-\theta_1)\pi_1^{(2)}, \theta_2\pi_2^{(2)}, (1-\theta_2)\pi_2^{(2)}\right)$$

$$\sim \text{Dirichlet}\left(\frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}\right)$$

- Repeat to get $\boldsymbol{\pi}^{(K)} \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$
- As $K \to \infty$, we get a vector with infinitely many components.

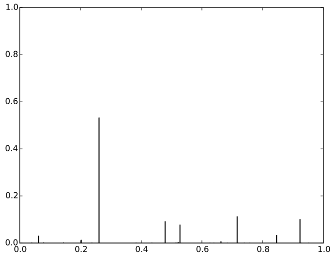


The Dirichlet process [Ferguson, 1973]

- We can combine this with a mechanism for generating parameter values.

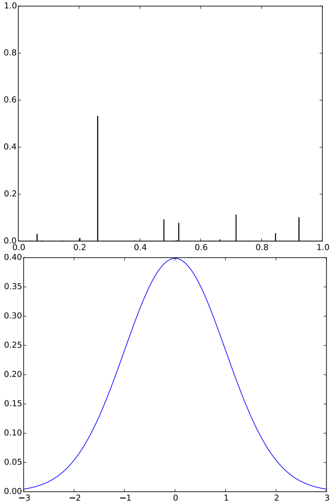
The Dirichlet process [Ferguson, 1973]

- We can combine this with a mechanism for generating parameter values.
- Let $\pi \sim \lim_{K\to\infty} \text{Dirichlet}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$



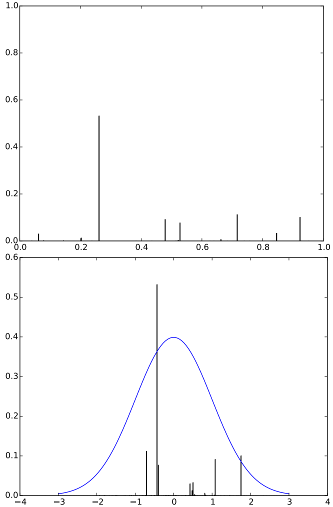
The Dirichlet process [Ferguson, 1973]

- We can combine this with a mechanism for generating parameter values.
- Let $\pi \sim \lim_{K\to\infty} \text{Dirichlet}\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$
- Let H be a distribution on some space Ω... e.g. a Gaussian distribution on the real line.



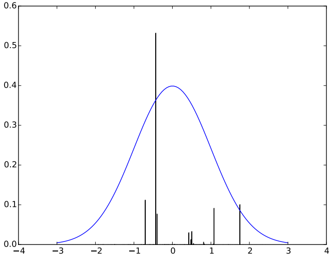
The Dirichlet process [Ferguson, 1973]

- We can combine this with a mechanism for generating parameter values.
- Let $\pi \sim \lim_{K\to\infty} \text{Dirichlet}\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$
- Let H be a distribution on some space Ω... e.g. a Gaussian distribution on the real line.
- For $k = 1, 2, \dots,$ sample $\theta_k \sim H$
- Then $G := \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$ is a probability distribution over Ω.
- Samples from the Dirichlet process are *discrete*. We call the point masses, *atoms*



Samples from the Dirichlet process

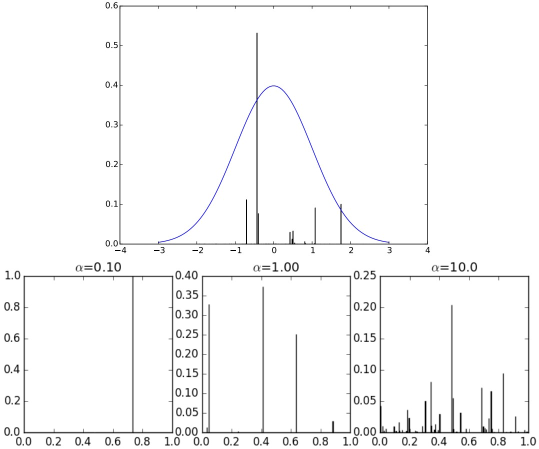
- We write $G \sim DP(\alpha, H)$
- The **base measure** H determines the *locations* of the atoms.



Samples from the Dirichlet process

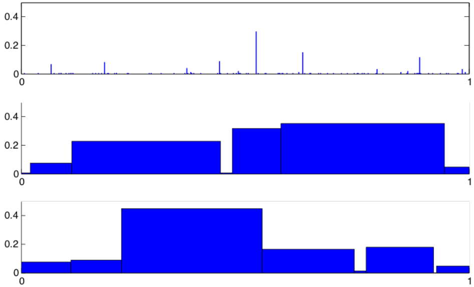
- We write $G \sim DP(\alpha, H)$
- The **base measure** H determines the *locations* of the atoms.

- The **concentration parameter** α determines the distribution over atom sizes.
- Small values of α give sparser distributions



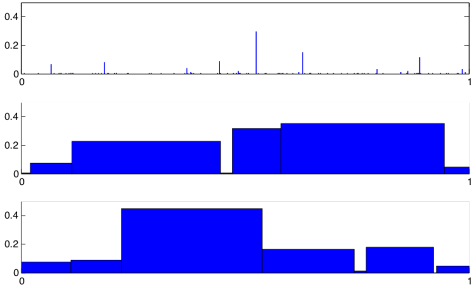
Dirichlet process and Dirichlet marginals

- Consider a Dirichlet process on $(0,1)$ with uniform base measure H.
- Pick any partition $A_1, \ldots, A_K$ of $(0,1)$, and sum up the atoms in each partition.



Dirichlet process and Dirichlet marginals

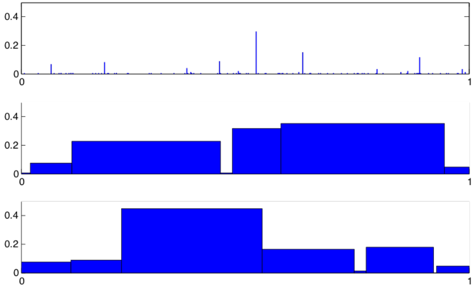
- Consider a Dirichlet process on $(0, 1)$ with uniform base measure H.
- Pick any partition $A_1, \ldots, A_K$ of $(0, 1)$, and sum up the atoms in each partition.



- Remember: If $(\pi_1, \pi_2, \ldots, \pi_K) \sim \text{Dirichlet}(\alpha_1, \alpha_2, \ldots, \alpha_K)$, then $(\pi_1 + \pi_2, \pi_3 \ldots, \pi_K) \sim \text{Dirichlet}(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_K)$
- So, the weights assigned to the partition are $\text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_K))$

Dirichlet process and Dirichlet marginals

- Consider a Dirichlet process on $(0, 1)$ with uniform base measure H.
- Pick any partition $A_1, \ldots, A_K$ of $(0, 1)$, and sum up the atoms in each partition.



- Remember: If $(\pi_1, \pi_2, \ldots, \pi_K) \sim \text{Dirichlet}(\alpha_1, \alpha_2, \ldots, \alpha_K)$, then $(\pi_1 + \pi_2, \pi_3 \ldots, \pi_K) \sim \text{Dirichlet}(\alpha_1 + \alpha_2, \alpha_3, \ldots, \alpha_K)$
- So, the weights assigned to the partition are $\text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_K))$
- This gives an alternative definition of the Dirichlet process: The (unique) distribution over Ω such that, for a partition $A_1, \ldots, A_K$ of Θ,

$$(P(A_1), \ldots, P(A_K)) \sim \text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_K))$$

The Dirichlet process mixture model [Antoniak, 1974]

- We can use the Dirichlet process in place of the Dirichlet distribution to construct a mixture model with infinitely many clusters.
  - Sample a probability distribution $G \sim \text{DP}(\alpha, H)$ where H is a normal-inverse Wishart distribution (i.e. a distribution over means and covariances).
  - This gives us $G = \sum_k \pi_k \delta_{\theta_k}$.

The Dirichlet process mixture model [Antoniak, 1974]

- We can use the Dirichlet process in place of the Dirichlet distribution to construct a mixture model with infinitely many clusters.
    - Sample a probability distribution $G \sim \text{DP}(\alpha, H)$ where H is a normal-inverse Wishart distribution (i.e. a distribution over means and covariances).
    - This gives us $G = \sum_k \pi_k \delta_{\theta_k}$.
    - For each observation, sample a cluster indicator $z_i \sim \pi$, and set $\phi_i := (\mu_i, \Sigma_i) = \theta_{z_i}$.

The Dirichlet process mixture model [Antoniak, 1974]

- We can use the Dirichlet process in place of the Dirichlet distribution to construct a mixture model with infinitely many clusters.
  - Sample a probability distribution $G \sim \text{DP}(\alpha, H)$ where H is a normal-inverse Wishart distribution (i.e. a distribution over means and covariances).
  - This gives us $G = \sum_k \pi_k \delta_{\theta_k}$.
  - For each observation, sample a cluster indicator $z_i \sim \pi$, and set $\phi_i := (\mu_i, \Sigma_i) = \theta_{z_i}$.
  - Then, sample the observation $x_i \sim \text{Normal}(\mu_i, \Sigma_i)$

The Dirichlet process mixture model [Antoniak, 1974]

- We can use the Dirichlet process in place of the Dirichlet distribution to construct a mixture model with infinitely many clusters.
  - Sample a probability distribution $G \sim \text{DP}(\alpha, H)$ where H is a normal-inverse Wishart distribution (i.e. a distribution over means and covariances).
  - This gives us $G = \sum_k \pi_k \delta_{\theta_k}$.
  - For each observation, sample a cluster indicator $z_i \sim \pi$, and set $\phi_i := (\mu_i, \Sigma_i) = \theta_{z_i}$.
  - Then, sample the observation $x_i \sim \text{Normal}(\mu_i, \Sigma_i)$
- The Dirichlet process has some similar properties to the Dirichlet distribution, that make inference feasible.

Conjugacy to the multinomial

- We saw that the Dirichlet distribution was conjugate to the multinomial.
- This is also true of the Dirichlet process!

- Pick a partition $A_1, \ldots, A_K$ of Ω, and let $P(A_k)$ be the mass assigned to A_k by $G \sim \text{Dirichlet}(\alpha, H)$.
- Then $(P(A_1), \ldots, P(A_k)) \sim \text{Dirichlet}\left(\alpha H(A_1), \ldots, \alpha H(A_K)\right)$.

Conjugacy to the multinomial

- We saw that the Dirichlet distribution was conjugate to the multinomial.
- This is also true of the Dirichlet process!
- Pick a partition $A_1, \ldots, A_K$ of Ω, and let $P(A_k)$ be the mass assigned to A_k by $G \sim \text{Dirichlet}(\alpha, H)$.
- Then $(P(A_1), \ldots, P(A_k)) \sim \text{Dirichlet}\left(\alpha H(A_1), \ldots, \alpha H(A_K)\right)$.
- If we see an observation in the jth segment, then we must have

$$(P(A_1), \ldots, P(A_j), \ldots, P(A_k)) \sim \text{Dirichlet}\left(\alpha H(A_1), \ldots, \alpha H(A_j) + 1, \ldots, \alpha H(A_K)\right)$$

Conjugacy to the multinomial

- We saw that the Dirichlet distribution was conjugate to the multinomial.
- This is also true of the Dirichlet process!
- Pick a partition $A_1, \dots, A_K$ of Ω, and let $P(A_k)$ be the mass assigned to A_k by $G \sim \text{Dirichlet}(\alpha, H)$.
- Then $(P(A_1), \dots, P(A_k)) \sim \text{Dirichlet}\,(\alpha H(A_1), \dots, \alpha H(A_K))$.
- If we see an observation in the jth segment, then we must have

$$(P(A_1), \dots, P(A_j), \dots, P(A_k)) \sim \text{Dirichlet}\,(\alpha H(A_1), \dots, \alpha H(A_j) + 1, \dots, \alpha H(A_K))$$

- This must be true for *all possible partitions* of Ω.
- This is only possible if the posterior of G is given by

$$G|X_1 = x \sim \text{DP}\left(\alpha + 1, \frac{\alpha H + \delta_x}{\alpha + 1}\right)$$

Predictive distribution

- Remember, for the Dirichlet distribution we could integrate out π to get $P(z_k = k | z_{-n}) \propto \sum_{i \neq j} \mathbb{I}(z_j = k) + \alpha_k$
- We can do something similar for the Dirichlet process!

- Let m_k be the number of times we have seen $X_i = \theta_k$ in the first n observations – or equivalently the number of times that $Z_i = k$ – and let K_+ be the number of values we've seen so far.
- The posterior distribution over G given n observations is

$$\text{DP}\left(\alpha + n, \frac{\alpha H + \sum_{k=1}^{K_+} m_k \delta_{\theta_k}}{\alpha + n}\right)$$

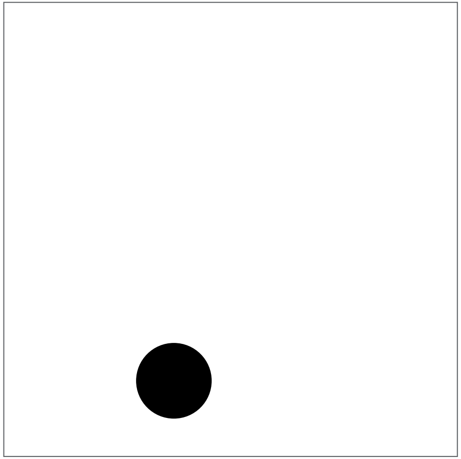
- So, we have

$$P(Z_{n+1} = k | Z_{1:n}) = \begin{cases} \frac{m_k}{n+\alpha} & \text{if } k \leq K_+ \\ \frac{\alpha}{n+\alpha} & \text{for new cluster} \end{cases}$$

- If we pick a new cluster, we sample it's parameter from H.

Polya urn scheme [Blackwell and MacQueen, 1973]

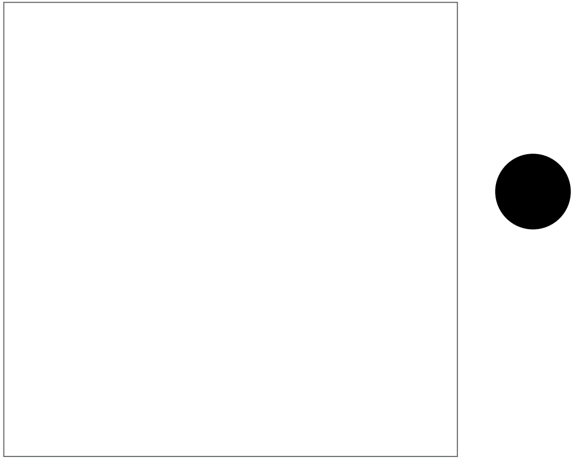
- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.



Polya urn scheme [Blackwell and MacQueen, 1973]

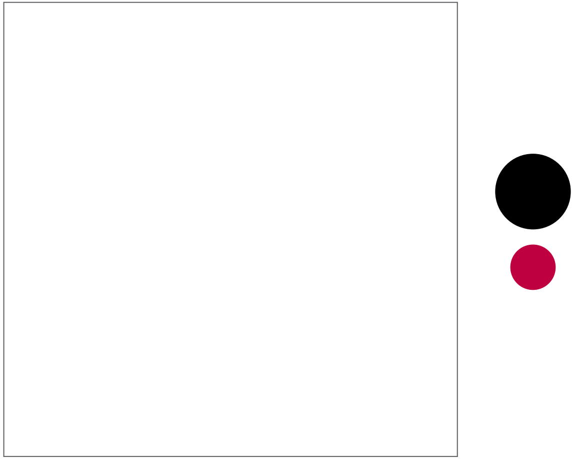
- Again, we can describe this using an urn analogy.

- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.



Polya urn scheme [Blackwell and MacQueen, 1973]

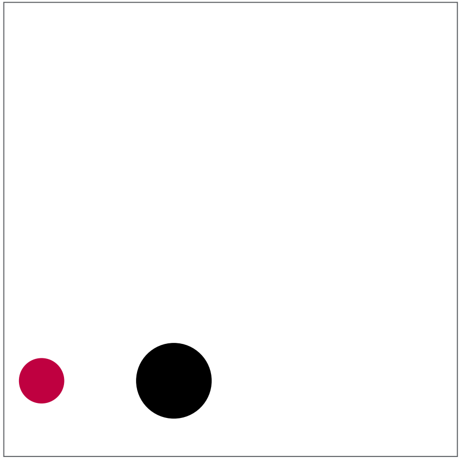
- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.



Polya urn scheme [Blackwell and MacQueen, 1973]

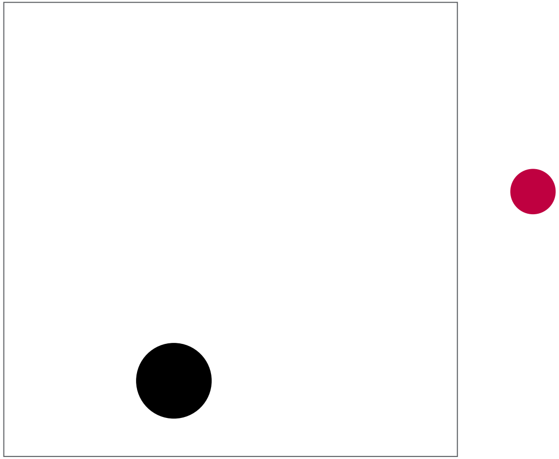
- Again, we can describe this using an urn analogy.

- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.



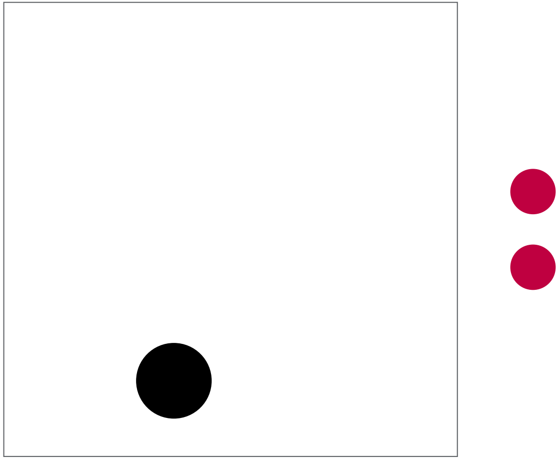
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



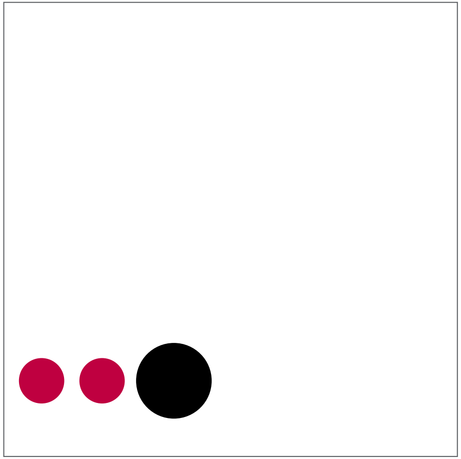
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



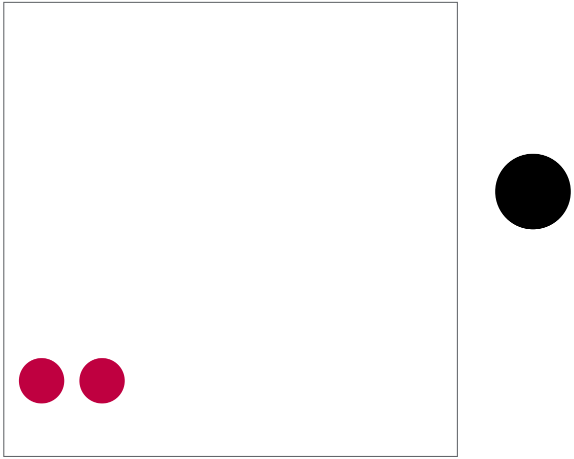
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



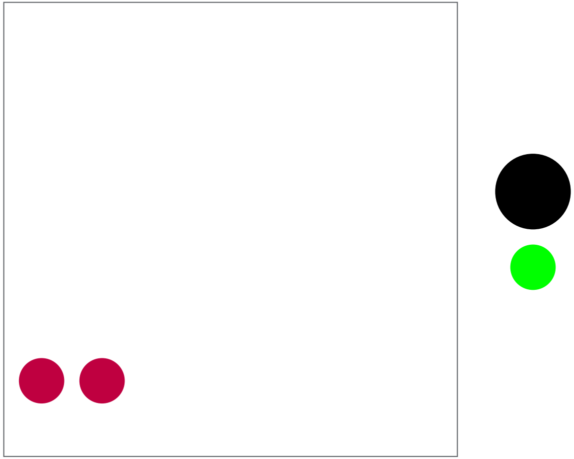
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

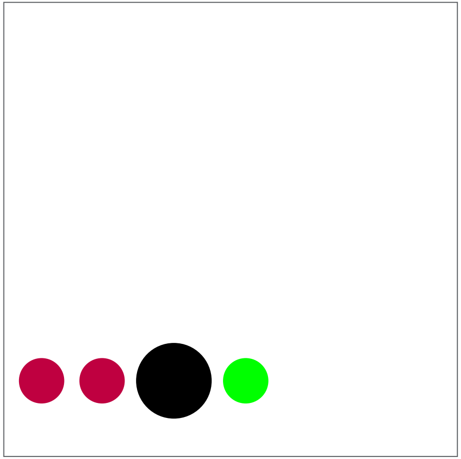
- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

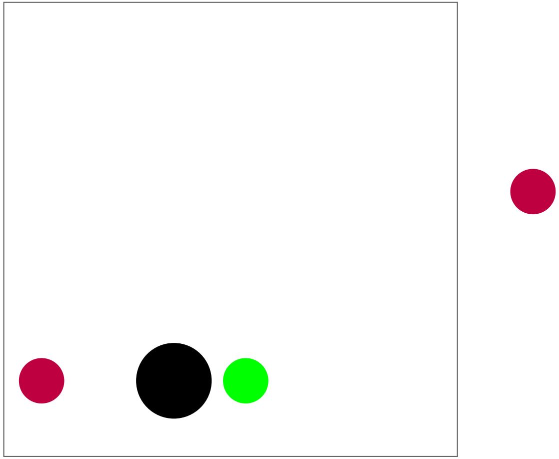
- Again, we can describe this using an urn analogy.

- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



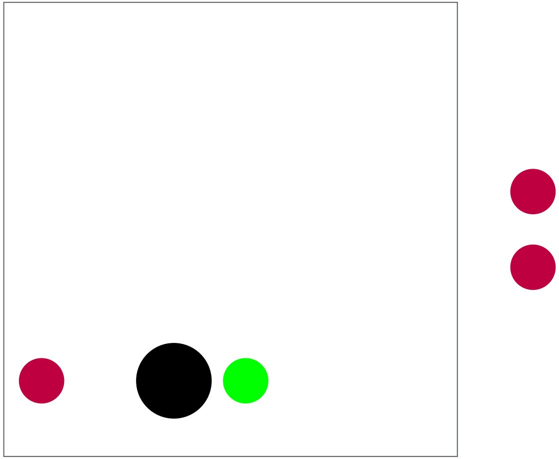
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



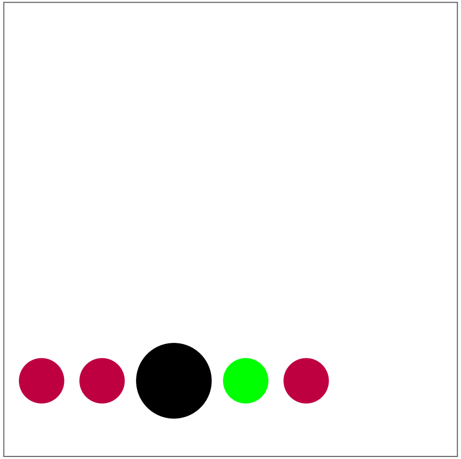
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



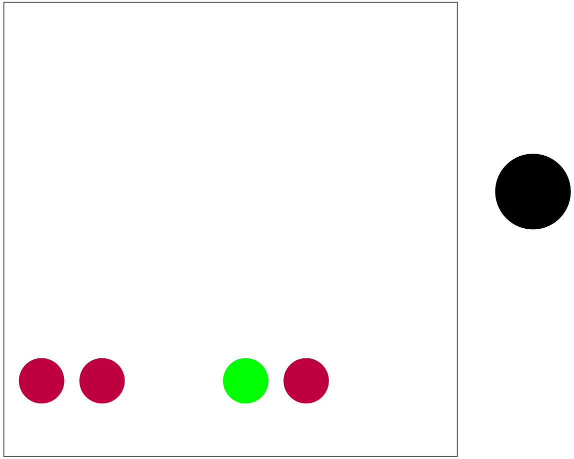
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



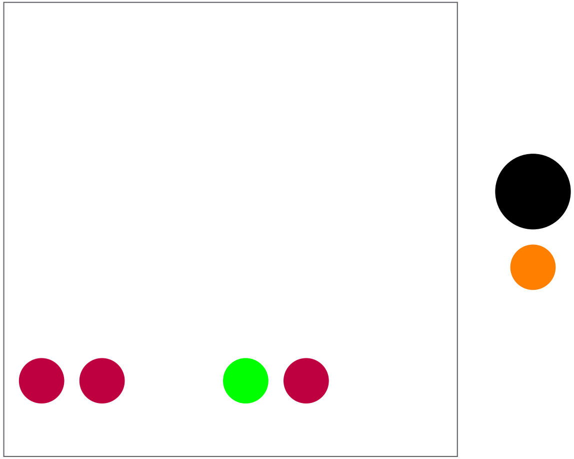
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

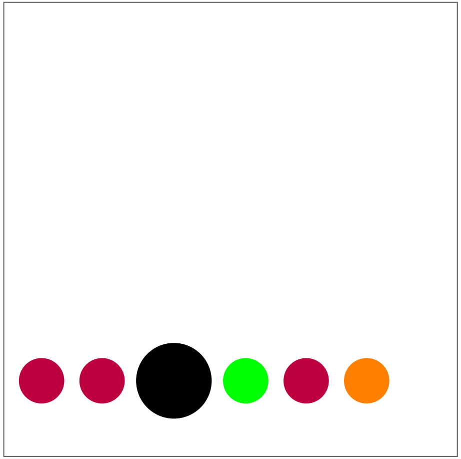
- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.

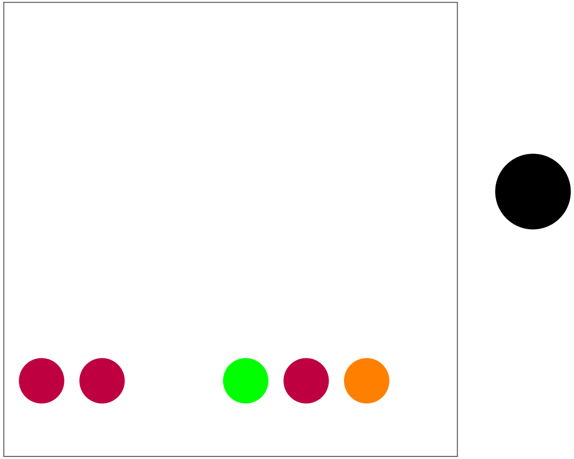
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

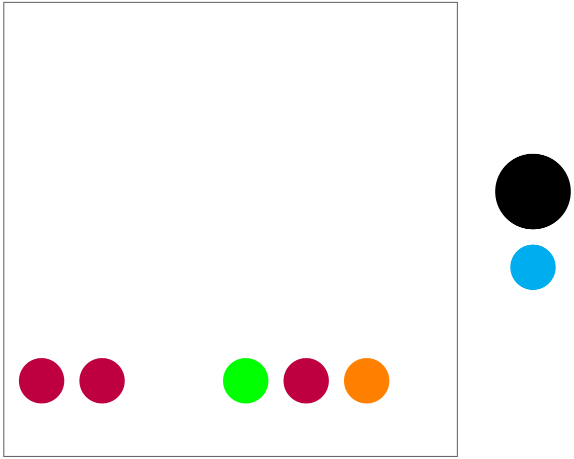
- Again, we can describe this using an urn analogy.

- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



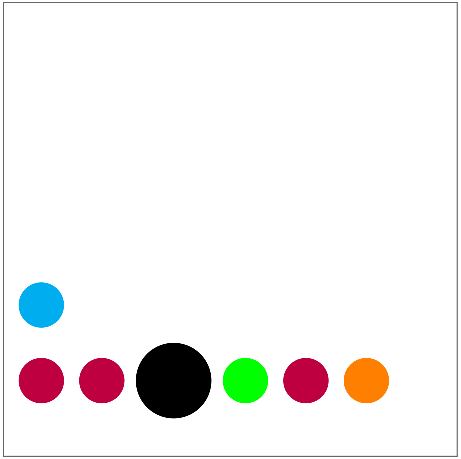
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

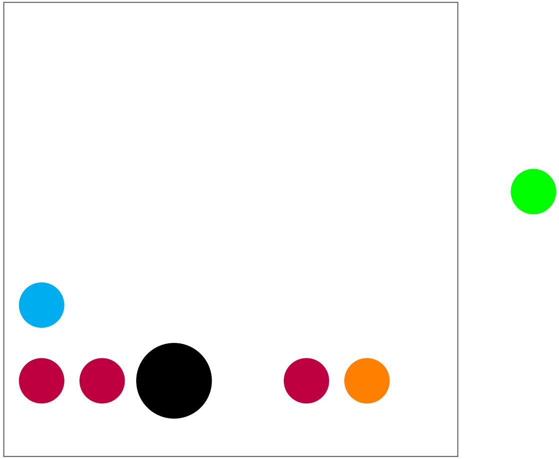
- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

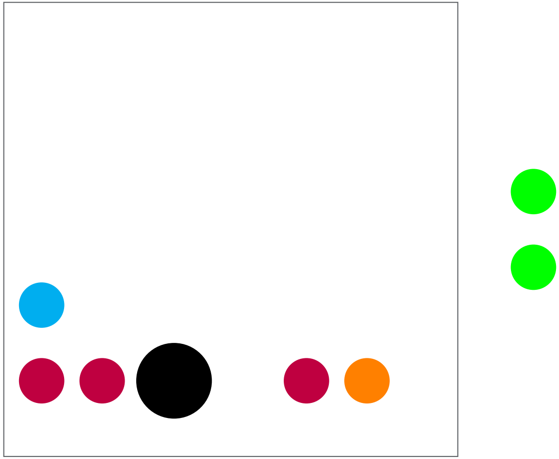
- Again, we can describe this using an urn analogy.

- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

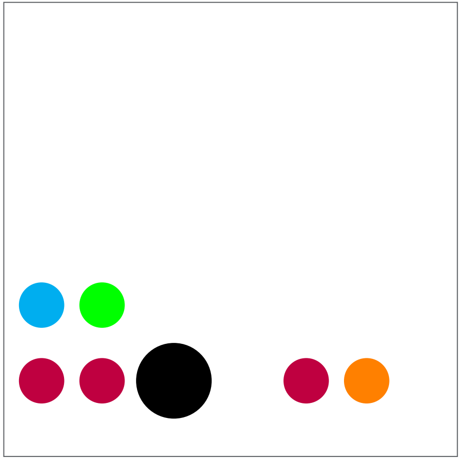
- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



Polya urn scheme [Blackwell and MacQueen, 1973]

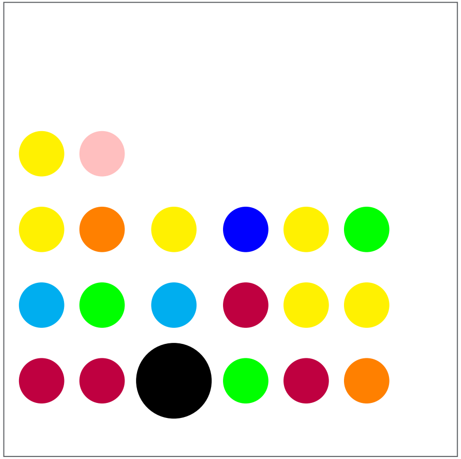
- Again, we can describe this using an urn analogy.

- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.



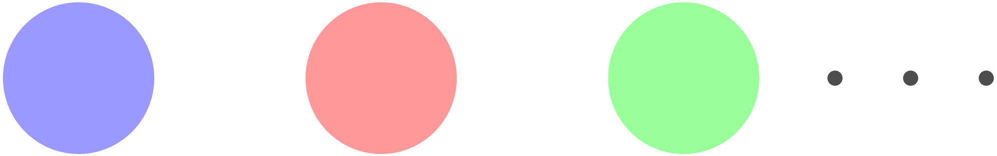
Polya urn scheme [Blackwell and MacQueen, 1973]

- Again, we can describe this using an urn analogy.
- Start with one black ball, of size α.
- Pick a ball with probability proportional to its size.
- If it's the black ball, sample a new color from H. Return the black ball plus a unit-size ball of the new color.
- If it's a colored ball, return that ball, plus a unit-size ball of the same color.
- Note, we can always sample a new color (the black ball is always there), but it gets less likely as N grows.



The Chinese restaurant process

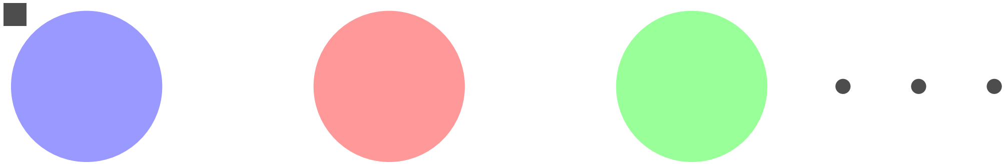
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.

The Chinese restaurant process

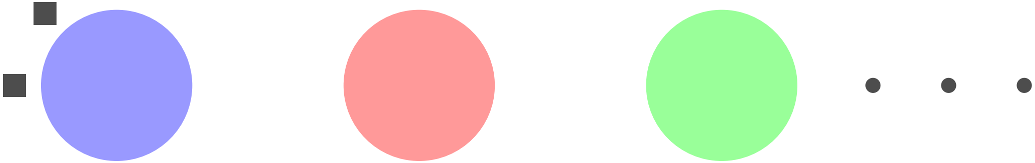
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.

The Chinese restaurant process

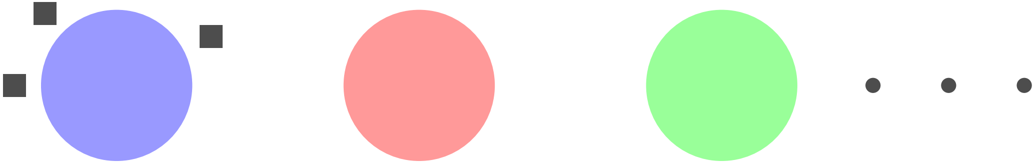
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.
- The second customer enters the restaurant. He sits at the first table with probability $\frac{1}{1+\alpha}$, or sits at a new table with probability $\frac{\alpha}{1+\alpha}$.

The Chinese restaurant process

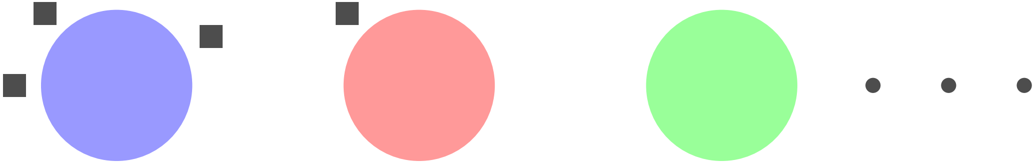
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.
- The second customer enters the restaurant. He sits at the first table with probability $\frac{1}{1+\alpha}$, or sits at a new table with probability $\frac{\alpha}{1+\alpha}$.
- Let m_k be the number of people sat at the kth table. The nth customer sits at the kth table with probability $\frac{m_k}{n-1+\alpha}$, or at a new table with probability $\frac{1}{n-1+\alpha}$.

The Chinese restaurant process

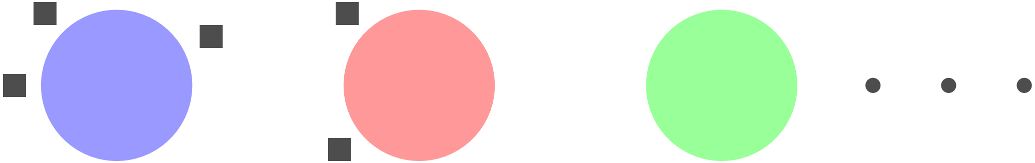
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.
- The second customer enters the restaurant. He sits at the first table with probability $\frac{1}{1+\alpha}$, or sits at a new table with probability $\frac{\alpha}{1+\alpha}$.
- Let m_k be the number of people sat at the kth table. The nth customer sits at the kth table with probability $\frac{m_k}{n-1+\alpha}$, or at a new table with probability $\frac{1}{n-1+\alpha}$.

The Chinese restaurant process

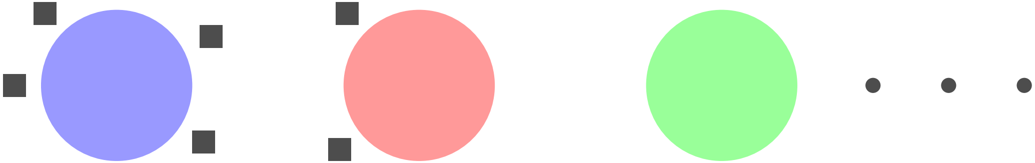
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.
- The second customer enters the restaurant. He sits at the first table with probability $\frac{1}{1+\alpha}$, or sits at a new table with probability $\frac{\alpha}{1+\alpha}$.
- Let m_k be the number of people sat at the kth table. The nth customer sits at the kth table with probability $\frac{m_k}{n-1+\alpha}$, or at a new table with probability $\frac{1}{n-1+\alpha}$.

The Chinese restaurant process

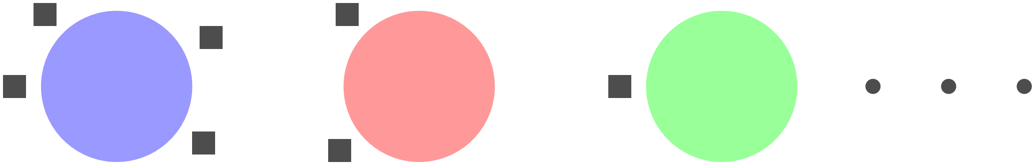
We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.
- The second customer enters the restaurant. He sits at the first table with probability $\frac{1}{1+\alpha}$, or sits at a new table with probability $\frac{\alpha}{1+\alpha}$.
- Let m_k be the number of people sat at the kth table. The nth customer sits at the kth table with probability $\frac{m_k}{n-1+\alpha}$, or at a new table with probability $\frac{1}{n-1+\alpha}$.

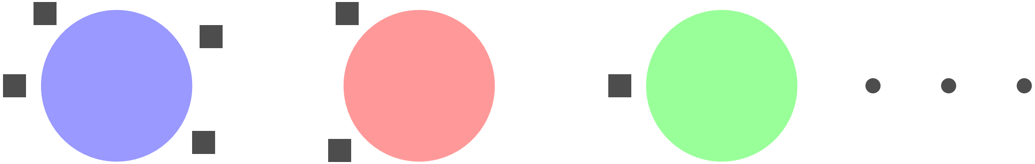
The Chinese restaurant process

We can also describe a sample from a DP-distributed probability distribution in terms of the following restaurant metaphor.



- Imagine a restaurant with infinitely many tables, each serving a different dish.
- The first customer comes to the restaurant, and sits at the first table.
- The second customer enters the restaurant. He sits at the first table with probability $\frac{1}{1+\alpha}$, or sits at a new table with probability $\frac{\alpha}{1+\alpha}$.
- Let m_k be the number of people sat at the kth table. The nth customer sits at the kth table with probability $\frac{m_k}{n-1+\alpha}$, or at a new table with probability $\frac{1}{n-1+\alpha}$.

The Chinese restaurant process



- We tend to sit at popular tables! This is known as the “rich-get-richer” property.
- We can always add new tables – *nonparametric*.
- For a given number of customers, the number of clusters is random.
- We can show that the probability of an assignment of people to tables is exchangeable (if we ignore the table ordering).

Inference using the Chinese restaurant process

- Since the cluster assignments are exchangeable, we can treat each customer as if he is the last.
- So, conditioned on the other $N-1$ observations, we know that the prior predictive probability of the ith data point (customer) being in cluster (table) k is

$$P(z_i = k | z_{-i}) \propto \begin{cases} m_k^{-i} & k \leq K_+ \\ \alpha & \text{new cluster} \end{cases}$$

Inference using the Chinese restaurant process

- Since the cluster assignments are exchangeable, we can treat each customer as if he is the last.
- So, conditioned on the other $N-1$ observations, we know that the prior predictive probability of the ith data point (customer) being in cluster (table) k is

$$P(z_i = k | z_{-i}) \propto \begin{cases} m_k^{-i} & k \leq K_+ \\ \alpha & \text{new cluster} \end{cases}$$

- Each cluster (table) k is associated with a parameter (dish) θ_k – e.g. the mean and covariance of a Gaussian.
- So, the conditional probability of the ith data point x_i being in cluster k is:

$$P(z_i = k | x_i, z_{-i}) \propto \begin{cases} m_k^{-i} f(x_i; \theta_k) & k \leq K_+ \\ \alpha \int f(x_i; \theta) dH(\theta) \text{new cluster} \end{cases}$$

where $f(x;\theta)$ is the appropriate likelihood model.

Collapsed inference using the Chinese restaurant process

This suggests a Gibbs sampler of the form:

- For $i = 1, \dots, N$:
  - Sample the cluster allocation of the ith data point, given the conditional distribution

$$P(z_i = k | x_i, z_{-i}) \propto \begin{cases} m_k^{-i} f(x_i; \theta_k) & k \leq K_+ \\ \alpha \int f(x_i; \theta) dH(\theta) & \text{new cluster} \end{cases}$$

  - If the number of clusters grow or shrink, adjust our representation accordingly (add/delete clusters)
- For $k = 1 : K_+$:
  - Sample the cluster parameters from their conditional distribution (unless they are integrated out)

Problems with the collapsed sampler

The collapsed sampler is easy to implement – but can have some problems

- We are only updating one data point at a time.
- Imagine two “true” clusters are merged into a single cluster – a single data point is unlikely to “break away”.
- Getting to the true distribution involves going through low probability states, so mixing can be slow.
- If the likelihood is not conjugate, integrating out parameter values for new features can be difficult.

- An alternative approach is to instantiate G, so we can update multiple data points at once.
- Problem: G is infinite-dimensional!
- Luckily, there is a nice representation that can help us...

Stick breaking construction [Sethuraman, 1994]

- Imagine a stick of unit length, representing the total probability.
- For $k = 1, 2, \ldots$
  - Sample a Beta$(1, \alpha)$ random variable b_k
  - Break off a fraction b_k of the stick. This is the first atom.
  - Sample a random location for this atom.
  - Recurse on the remaining stick to get:

$$b_k \sim \text{Beta}(1, \alpha) \qquad \pi_k = b_k \prod_{j=1}^{k-1} (1 - b_k) \qquad \theta_k \sim H \qquad G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$$

Stick breaking construction [Sethuraman, 1994]

- Imagine a stick of unit length, representing the total probability.
- For $k = 1, 2, \ldots$
  - Sample a Beta$(1, \alpha)$ random variable b_k
  - Break off a fraction b_k of the stick. This is the first atom.
  - Sample a random location for this atom.
  - Recurse on the remaining stick to get:

$$b_k \sim \text{Beta}(1, \alpha) \qquad \pi_k = b_k \prod_{j=1}^{k-1} (1 - b_k) \qquad \theta_k \sim H \qquad G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$$

- We can use the b_k directly to obtain the cluster assignment.
- Starting at the first cluster, choose cluster k with probability b_k, else move on to the next cluster.

Blocked Gibbs sampling using the stick breaking construction

This gives us an alternative inference approach.

- Pick a truncation K, so we are working with an approximation to the DP,

$$b_k \sim \text{Beta}(1, \alpha) \qquad \pi_k = b_k \prod_{j=1}^{k-1}(1 - b_k) \qquad \theta_k \sim H \qquad G^K = \sum_{k=1}^{K} \pi_k \delta_{\theta_k}$$

- Conditioned on G_k, we can sample a cluster assignment using

$$P(z_i = k | G^K, x_i) \propto \pi_k f(x_i|\theta_k)$$

- Contitioned on the cluster allocations, we can update each b_k.
- Remember, b_k is the probability of belonging to the kth cluster, conditioned on not belonging to any previous clusters.
- So,

$$b_k|\mathbf{z} \sim \text{Beta}\left(1 + m_k, \alpha + \sum_{j=k+1}^{K} m_j\right)$$

Summary

- We've introduced the Dirichlet process, which we can think of as an infinitely large Dirichlet distribution.
- We've explored different ways of representing the DP:
  - Dirichlet marginals
  - Urns
  - Chinese restaurant process
  - Stick-breaking construction
- We've explored the main ways of doing inference... if you can code up a collapsed Gibbs sampler for a Dirichlet mixture of Gaussians, you should be able to code up a sampler for a DP mixture of Gaussians.
- Now let's look at a new class of models...

Lecture Overview

- Lecture 1: The Dirichlet process
- Lecture 2: The Indian buffet process
  - From clustering to latent feature modeling
  - The beta-Bernoulli process
  - The Indian buffet process
  - Modeling
  - Inference
- Lecture 3: Hierarchical nonparametric models

Beyond clustering

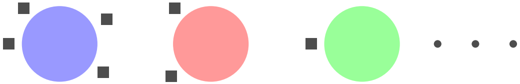
- The Dirichlet distribution and the Dirichlet process are great if we want to cluster data into non-overlapping clusters.
- However, DP/Dirichlet mixture models cannot share features between clusters.
- In many applications, data points exhibit properties of multiple latent features
  - Images contain multiple objects.
  - Actors in social networks belong to multiple social groups.
  - Movies contain aspects of multiple genres.

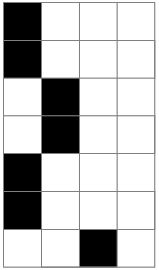
Latent variable models

- *Latent variable models* allow each data point to exhibit multiple features, to varying degrees.
- Example: Factor analysis: $X = WA^T + \epsilon$, where
  - K rows of A = latent features
  - N rows of W = datapoint-specific weights for these features
  - ϵ = Gaussian noise.
- Question: Can we make the number of features unbounded a posteriori, as we did with the DP?
- Like the DP, we want to allow infinitely many features *a priori* – ie let W have infinitely many columns.
- Problem: In factor analysis, the matrices W and A are dense... if we make them infinitely large, we'd have to represent infinitely many features!
- Solution: make our matrix W of datapoint-specific weights sparse.
- Intuition: We allow a data point to exhibit infinitely many features a priori... but in practice, most of them are zeroed out.

The CRP: A distribution over binary matrices

- Recall that the CRP gives us a distribution over partitions of our data.
- We can represent this as a distribution over binary matrices, where each row corresponds to a data point, and each column to a cluster.





■	□	□	□
■	□	□	□
□	■	□	□
□	■	□	□
■	□	□	□
■	□	□	□
□	□	■	□

- This gives us a sparse matrix... but only one feature per data point.

A sparse, finite latent variable model

- We're going to have to come up with a different model.
- Let's think about the finite case first.
- Simplest idea: make W a binary matrix Z – so a feature is either "on" or "off".
- For K features, we can do this using a *beta-Bernoulli* prior on Z.

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

A sparse, finite latent variable model

- We're going to have to come up with a different model.
- Let's think about the finite case first.
- Simplest idea: make W a binary matrix Z – so a feature is either "on" or "off".
- For K features, we can do this using a *beta-Bernoulli* prior on Z.

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

- π_k is the global probability of a data point exhibiting feature k.
- z_{nk} tells us whether the nth data point exhibits a given feature.

The beta process

- However, we don't want K features... we want infinitely many features!
- The **beta process** [Hjort, 1990] is a distribution over discrete measures $B = \sum_k \mu_k \delta_{\theta_k}$ with atoms in $[0, 1]$.
  - If you don't know measure theory, just think of a measure as an unnormalized probability distribution.
  - For our purposes, we can think of it as a distribution over infinitely long sequences of probabilities μ_k.
- This sequence corresponds to the limit, as $K \to \infty$, of K Beta$(\alpha/K, 1)$ random variables.
- Most of these will be really small... but some of them will be significant.

The beta-Bernoulli process

- The **Bernoulli process** is a distribution over infinite-dimensional binary sequences $z_1, z_2, \dots$
- It is parametrized by an infinite sequence of probabilities $\mu_1, \mu_2, \dots$
- Each element z_k is sampled according to Bernoulli(μ_k)

The beta-Bernoulli process

- The **Bernoulli process** is a distribution over infinite-dimensional binary sequences $z_1, z_2, \dots$
- It is parametrized by an infinite sequence of probabilities $\mu_1, \mu_2, \dots$
- Each element z_k is sampled according to Bernoulli(μ_k)
- So, we can write the limit of our latent feature model, as $K \to \infty$, as

$$\begin{aligned} B = (\mu_1, \mu_2, \dots) &\sim \text{BetaProc}(\alpha) \\ z_i &\sim \text{BernoulliProc}(B) \end{aligned}$$

- This is known as a beta-Bernoulli process [Thibaux and Jordan, 2007]

The beta process and the Dirichlet process – an aside

- The beta process and the Dirichlet process are fairly similar... they are both infinite-dimensional measures of the form $\sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$.
- They both have atoms between zero and one.
- The main difference is that the Dirichlet process sums to one, and the beta process doesn't.

- These are both examples of a much wider class of nonparametric processes:
  - The gamma process and the stable process are distributions over infinite-dimensional measures with positive real-valued atoms.
  - The Poisson process is a distribution over infinite-dimensional measures with unit-valued atoms (counting measures).
  - The Pitman-Yor process is a distribution over probability distributions similar to the Dirichlet process.

A stick-breaking representation for the beta process

- When working with a beta process, we have infinitely many almost-zero atoms.
- When generating samples from a beta process, we want to generate the largest atoms first.
- We can use a stick-breaking process to do this.

- Begin with a stick of unit length.
- For $k = 1, 2, \ldots$
  - Sample a Beta$(\alpha, 1)$ random variable μ_k.
  - Break off a fraction μ_k of the stick. This is the kth atom size.
  - Throw away what's left of the stick.
  - Recurse on the part of the stick that you broke off

- Note that, unlike the DP stick breaking construction, the atoms will not sum to one.

[Teh et al., 2007]

Integrating out the beta process

- In practice, we often don't want to work directly with an infinite-dimensional vector!

- How did we deal with this in the Dirichlet process?
  - Integrate out the infinite-dimensional vector to get a collapsed representation, with a restaurant analogy.
  - Because a finite number of data points must belong to a finite number of clusters, we are left with a finite-dimensional vector of cluster assignments.
  - The restaurant scheme directly suggests a way to do Gibbs sampling.

- We can do exactly the same thing with the beta process!

Predictive distribution of a beta-Bernoulli distribution

- Before we get into the full predictive distribution, let's just think about the beta distribution.
- Let $p \sim \text{Beta}(\alpha, \beta)$ be the bias of a coin.
- Let $x_i \sim \text{Bernoulli}(p)$ be the outcome of a coin toss.
- The posterior distribution after n coin tosses is

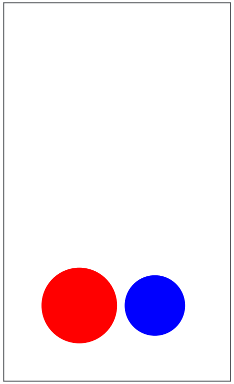
$$p|x_1, \dots, x_n \sim \text{Beta}\left(\alpha + \sum_{i=1}^{n} x_i, \beta + n - \sum_{i=1}^{n} x_i\right)$$

- If we integrate out p, the predictive distribution for x_{n+1} is

$$P(x_{n+1} = 1|x_1, \dots, x_n) = \frac{\alpha + \sum_{i=1}^{n} x_i}{\alpha + \beta + n}$$

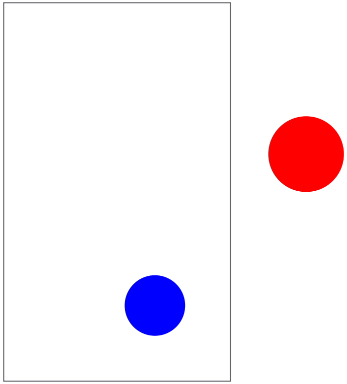
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β



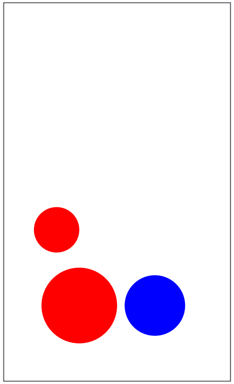
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.



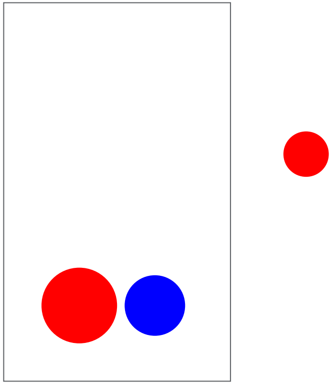
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.



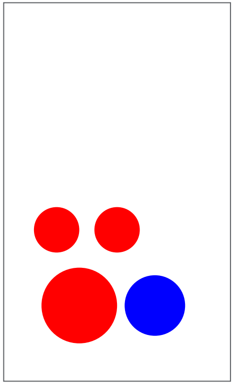
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.



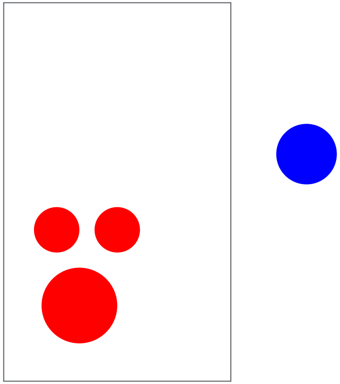
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.



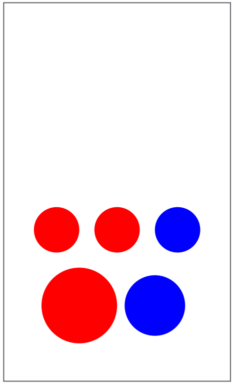
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.



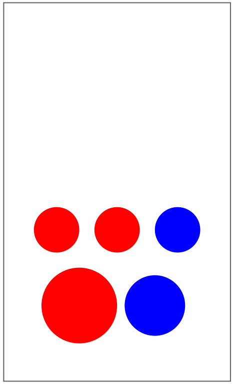
Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.



Predictive distribution of a beta-Bernoulli distribution

- Again, we can think of this in terms of urns!
- Start off with a red ball of size α, and a blue ball of size β
- Pick a ball with probability proportional to its size.
- Return that ball, plus a unit-size ball of the same color.



- Note that the order we sample the balls doesn't matter...

$$P(r,r,b) = \frac{\alpha}{\alpha+\beta}\frac{\alpha+1}{\alpha+\beta+1}\frac{\beta}{\alpha+\beta+2}$$

$$P(r,b,r) = \frac{\alpha}{\alpha+\beta}\frac{\beta}{\alpha+\beta+1}\frac{\alpha+1}{\alpha+\beta+2}$$

- In other words, it is exchangeable!

Predictive distribution of a beta-Bernoulli distribution

- Our model is the limit, as $K \to \infty$, of K Beta$(\alpha/K, 1)$ random variables.
- Let's consider sampling the nth row.
- If we have already seen $m_k > 0$ non-zero elements in column k, that corresponds to m_k unit-sized red balls (plus an infinitesimally small initial ball), plus $n - m_k$ unit-sized blue balls (including the inital ball).
- So, the probability of a non-zero entry is the proportion of red: $\frac{m_k}{n}$.

Predictive distribution of a beta-Bernoulli distribution

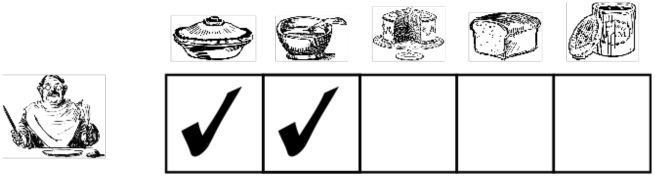
- Our model is the limit, as $K \to \infty$, of K Beta$(\alpha/K, 1)$ random variables.
- Let's consider sampling the nth row.
- If we have already seen $m_k > 0$ non-zero elements in column k, that corresponds to m_k unit-sized red balls (plus an infinitesimally small initial ball), plus $n - m_k$ unit-sized blue balls (including the inital ball).
- So, the probability of a non-zero entry is the proportion of red: $\frac{m_k}{n}$.
- Problem: What if we haven't seen any non-zero entries yet for the kth feature? We have a bunch of blue balls, and our initial red ball is infinitely small!

Predictive distribution of a beta-Bernoulli distribution

- Our model is the limit, as $K \to \infty$, of K Beta$(\alpha/K, 1)$ random variables.
- Let's consider sampling the nth row.
- If we have already seen $m_k > 0$ non-zero elements in column k, that corresponds to m_k unit-sized red balls (plus an infinitesimally small initial ball), plus $n - m_k$ unit-sized blue balls (including the inital ball).
- So, the probability of a non-zero entry is the proportion of red: $\frac{m_k}{n}$.
- Problem: What if we haven't seen any non-zero entries yet for the kth feature? We have a bunch of blue balls, and our initial red ball is infinitely small!
- Well, we have infinitely many features with no non-zero entries.
- Because of the relationship between the Bernoulli and the Poisson, we know that, out of these infinitely many urns, we will get a Poisson-distributed number of balls in total.

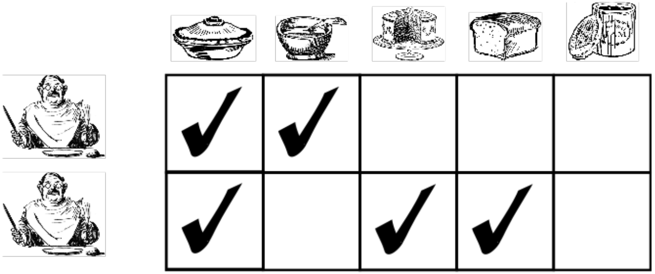
The Indian buffet process (IBP)

- We can combine these urns to get a predictive distribution over the binary matrix Z.
- We can describe this in terms of the following restaurant analogy.
  - A customer enters a restaurant with an infinitely large buffet.
  - He helps himself to Poisson(α) dishes.



The Indian buffet process (IBP)

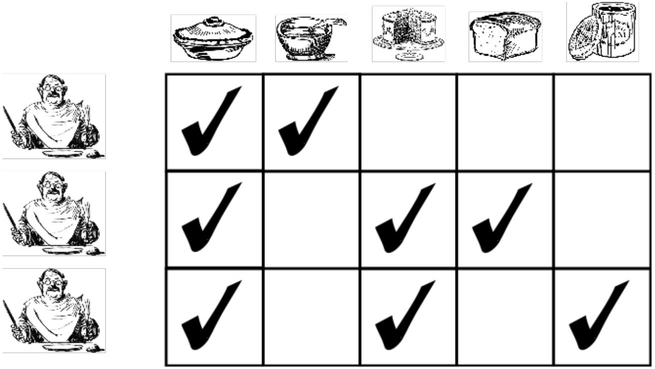
- We can combine these urns to get a predictive distribution over the binary matrix Z.
- We can describe this in terms of the following restaurant analogy.
  - A customer enters a restaurant with an infinitely large buffet.
  - He helps himself to Poisson(α) dishes.
  - The ith customer enters the restaurant
  - She helps herself to each dish with probability m_k/i, where m_k is the number of people who've tried dish k.
  - She then tries Poisson(α/i) new dishes



✓	✓			
✓		✓	✓	

The Indian buffet process (IBP)

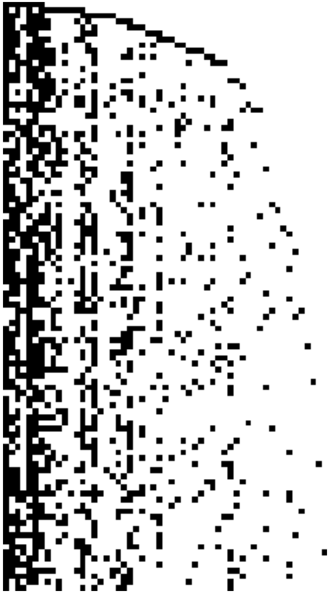
- We can combine these urns to get a predictive distribution over the binary matrix Z.
- We can describe this in terms of the following restaurant analogy.
  - A customer enters a restaurant with an infinitely large buffet.
  - He helps himself to Poisson(α) dishes.
  - The ith customer enters the restaurant
  - She helps herself to each dish with probability m_k/i, where m_k is the number of people who've tried dish k.
  - She then tries Poisson(α/i) new dishes



✓	✓			
✓		✓	✓	
✓		✓		✓

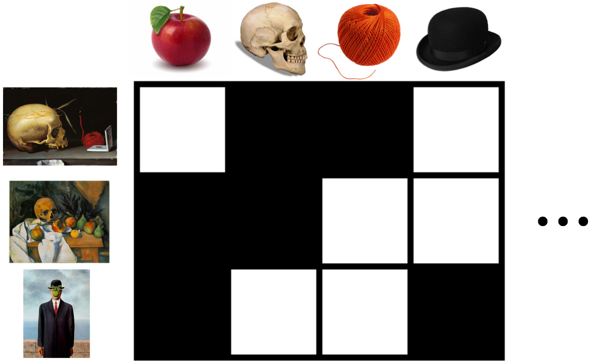
The Indian buffet process (IBP)

- Each row has a Poisson(α) number of features – due to exchangeability.
- "Rich get richer" property – popular dishes become more popular.
- Total number of non-empty columns unbounded and grows with N.
- Concretely, total number of dishes is Poisson(αH_N), where $H_N = \sum_{i=1}^N \frac{1}{i}$.



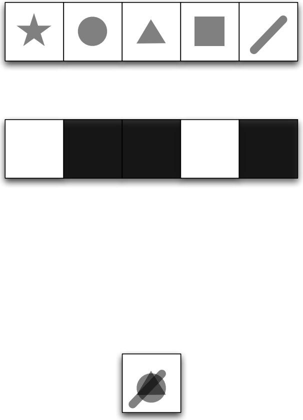
Building latent feature models using the IBP

- We can use the IBP 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 of the IBP ensures features are shared between data points.
- We must pick a likelihood model that determines what the features look like and how they are combined.



A linear Gaussian model

- The simplest likelihood model is to assume the features are normally distributed, and we superimpose features selected by the IBP.
- Sample $Z \sim \text{IBP}(\alpha)$
- For each feature k, sample $A_k \sim \text{Normal}(0, \sigma_A^2 \mathbf{I})$
- Sample nth observation $x_n \sim \text{Normal}(z_n A^T, \sigma_X^2 \mathbf{I})$



[Griffiths and Ghahramani, 2005]

Infinite factor analysis

- Problem with linear Gaussian model: Features are "all or nothing"
- Not always reasonable... maybe a movie is mostly a horror, but with some elements of comedy and romance?
- Factor analysis allows for weighted features: $X = WA^T + \epsilon$ where
  - Rows of A = latent features (Gaussian)
  - Rows of W = datapoint-specific weights for these features (Gaussian)
  - ϵ = Gaussian noise.

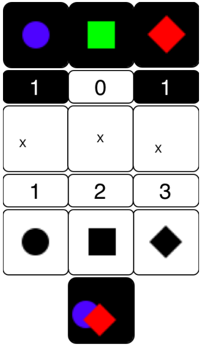
Infinite factor analysis

- Problem with linear Gaussian model: Features are "all or nothing"
- Not always reasonable... maybe a movie is mostly a horror, but with some elements of comedy and romance?
- Factor analysis allows for weighted features: $X = WA^T + \epsilon$ where
  - Rows of A = latent features (Gaussian)
  - Rows of W = datapoint-specific weights for these features (Gaussian)
  - ϵ = Gaussian noise.
- We can obtain similar properties in an infinite model, by associating each non-zero entry of our IBP with a Gaussian weight
- Let $X = (Z \odot V)A^T + \epsilon$, where
  - $Z \sim \text{IBP}(\alpha)$
  - $v_{nk} \sim \text{Normal}(0, \sigma_V^2)$
  - $A_k \sim \text{Normal}(0, \sigma_A^2 \mathbf{I})$

[Knowles and Ghahramani, 2007]

More complicated models

- We can come up with more complicated models to fit our data.
- For example, assume we are trying to model images in terms of latent features.
- Here, we need to deal with occlusion, translation and rotation of features.
- We can model each feature using a combination of a Gaussian-distributed image vector, a location between foreground and background, and a binary alpha-channel.
- We can associate each non-zero entry of the IBP with a transformation.
- We generate the image by transforming the selected features using the corresponding transformation, and superimposing them in the given order.



[Zhai et al., 2012]

Inference in the IBP

- When looking at the IBP, we considered two main forms of inference
  - Collapsed inference, using a restaurant analogy.
  - Uncollapsed inference, using a stick-breaking representation.
- We can construct analogous samplers for the IBP!
- We'll work with the linear Gaussian model:

$$\begin{aligned} Z &\sim \text{IBP}(\alpha) \\ A_k &\sim \text{Normal}(0, \sigma_A^2 \mathbf{I}) \\ x_n &\sim \text{Normal}(z_n A^T, \sigma_X^2 \mathbf{I}) \end{aligned}$$

Collapsed inference using the restaurant analogy

- We can sample a matrix from an IBP using the restaurant analogy
  - A customer enters a restaurant with an infinitely large buffet.
  - He helps himself to Poisson(α) dishes.
  - The ith customer enters the restaurant
  - She helps herself to each dish with probability m_k/i, where m_k is the number of people who've tried dish k.
  - She then tries Poisson(α/i) new dishes
- However, the sequence of customers is exchangeable... it doesn't matter what order we enter in.
- Rather than condition on the first $i-1$ data points, we can always "pretend" the ith data point is actually the last.

Collapsed inference using the restaurant analogy

- We can re-work the Indian buffet process to generate samples from the IBP.
- Lets assume we have N customers in our restaurant, all eating different dishes.
- The ith customer drops her plate, so she has to go back to the buffet.
- She considers all the dishes that her fellow diners have on their plates. She takes each dish with probability m_k^{-i}/N, where m_k^{-i} is the number of customers currently with dish k (except herself).
- She then takes a Poisson(α/N) number of new dishes.

Collapsed inference using the restaurant analogy

- We can re-work the Indian buffet process to generate samples from the IBP.
- Lets assume we have N customers in our restaurant, all eating different dishes.
- The ith customer drops her plate, so she has to go back to the buffet.
- She considers all the dishes that her fellow diners have on their plates. She takes each dish with probability m_k^{-i}/N, where m_k^{-i} is the number of customers currently with dish k (except herself).
- She then takes a Poisson(α/N) number of new dishes.

- In other words:
  - For columns where $m_k^{-i} > 0$, we have $P(z_{ik} = 1|z_{-i,k}) = \frac{m_k^{-i}}{N}$
  - In addition, we have a Poisson(α/N) number of features that appear only in the ith row.

Collapsed inference using the restaurant analogy

- We can construct a Gibbs sampler that iterates through each element of each row.
- To resample the ith row Z_i, we first resample the elements where $m_k^{-i} > 0$
- Combining our prior predictive with our likelihood, we have

$$P(z_{ik}=1|x_i, Z_{-ik}, A) \propto m_k f(x_i; z_{ik}=1, Z_{-ik}, A)$$
$$P(z_{ik}=1|x_i, Z_{-ik}, A) \propto (N-m_k) f(x_i; z_{ik}=0, Z_{-ik}, A)$$

  where f is our linear Gaussian likelihood.
- In the linear Gaussian case we can integrate out A if desired [Griffiths and Ghahramani, 2005]

Collapsed inference using the restaurant analogy

- Next, we must propose adding/removing singleton features using a Metropolis-Hastings distribution.
  - Let K^*_{old} be the number of features appearing only in the ith data point.
  - Propose $K^*_{new} \sim \text{Poisson}(\alpha/N)$, and let Z^* be the matrix with K^*_{new} features appearing only in the ith data point.
  - If $K^*_{new} > K^*_{old}$ and you are not integrating out A, sample $K^*_{new} - K^*_{old}$ new features to create a proposal feature matrix A^*
  - Accept the proposal with probability $\min\left(1, \frac{f(x_i|Z^*,A^*)}{f(x_i|Z,A)}\right)$

Collapsed inference using the restaurant analogy

- Next, we must propose adding/removing singleton features using a Metropolis-Hastings distribution.
  - Let K^*_{old} be the number of features appearing only in the ith data point.
  - Propose $K^*_{new} \sim \text{Poisson}(\alpha/N)$, and let Z^* be the matrix with K^*_{new} features appearing only in the ith data point.
  - If $K^*_{new} > K^*_{old}$ and you are not integrating out A, sample $K^*_{new} - K^*_{old}$ new features to create a proposal feature matrix A^*
  - Accept the proposal with probability $\min\left(1, \frac{f(x_i|Z^*,A^*)}{f(x_i|Z,A)}\right)$
- If we're instantiating A, we then sample new values conditioned on X and Z (see [Doshi-Velez and Ghahramani, 2009])

Summary

We've looked at two of the main "building blocks" in nonparametric Bayesian modeling

- The Dirichlet process is an infinite-dimensional analogue of the Dirichlet distribution.
- We use the Dirichlet distribution for clustering data into K clusters (among other things).
- Similarly, we can use the Dirichlet process to cluster data into an unbounded (and growing) number of clusters.

- The Indian buffet process is an infinite-dimensional model for feature subset selection.
- We can use it to construct latent feature models with infinitely many features.
- We can customize the latent feature model to match our data.

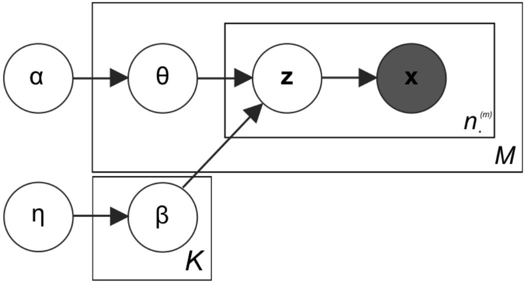
- Many more building blocks – gamma process, Poisson process, Pitman-Yor process, Kingman's coalescent... but these are the two most popular.
- But for now, we're going to take a quick look at some hierarchical models that use the DP and IBP as building blocks.

Latent Dirichlet allocation

- Dirichlet distributions are commonly used in topic models.
- Topic models describe documents using a distribution over “topics”.
- Each “topic” is a distribution over words

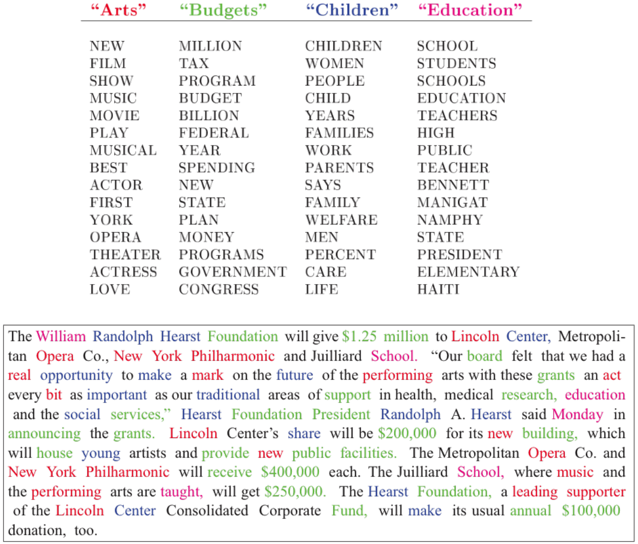
Latent Dirichlet allocation

- Dirichlet distributions are commonly used in topic models.
- Topic models describe documents using a distribution over "topics".
- Each "topic" is a distribution over words
- Example: Latent Dirichlet allocation [Blei et al., 2003]



- For each topic $k = 1, \dots, K$
  - Sample a distribution over words, $\beta_k \sim \text{Dirichlet}(\eta_1, \dots, \eta_V)$
- For each document $m = 1, \dots, M$:
  - Sample a distribution over topics, $\theta_m \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_k)$
  - For each word $n = 1, \dots, N_m$ in the mth document:
    - Sample a topic $z_{mn} \sim \text{Discrete}(\theta_m)$
    - Sample a word $w_{mn} \sim \text{Discrete}(\beta_{z_{mn}})$

Latent Dirichlet allocation



“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Image from [Blei et al., 2003]

Latent Dirichlet allocation

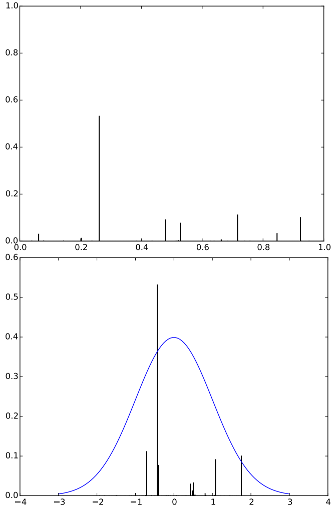
- For each topic $k = 1, \ldots, K$
  - Sample a distribution over words, $\beta \sim \text{Dirichlet}(\eta_1, \ldots, \eta_V)$
- For each document $m = 1, \ldots, M$:
  - Sample a distribution over topics, $\theta_m \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k)$
  - For each word $n = 1, \ldots, N_m$ in the mth document:
    - Sample a topic $z_{mn} \sim \text{Discrete}(\theta_m)$
    - Sample a word $w_{mn} \sim \text{Discrete}(\beta_{z_{mn}})$

- We have two Dirichlet distributions... one over words, one over topics.
- It's probably OK to say we have a fixed, known number of words... the dictionary is fairly constant.
- However, it's hard to pick a number of topics.
- Solution: Let's replace the distribution over topics with a Dirichlet process!

The Dirichlet process... a reminder

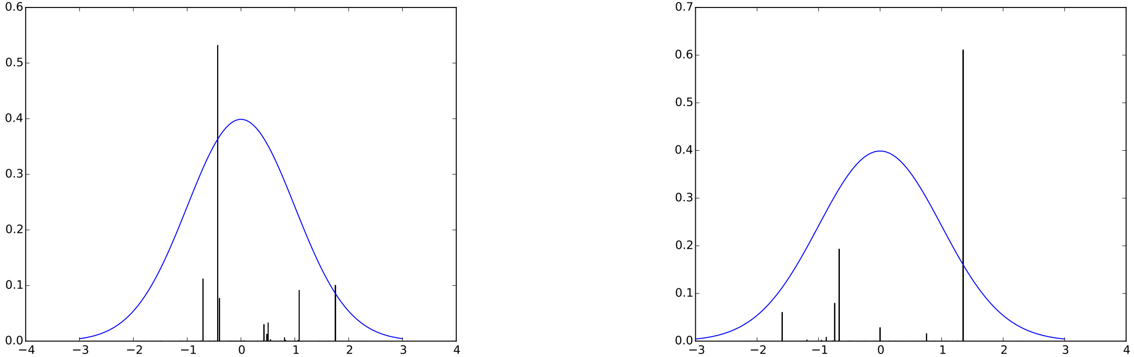
Let's remind ourselves how we draw samples from the DP

- We can combine this with a mechanism for generating parameter values.
- Let $\pi \sim \lim_{K\to\infty} \text{Dirichlet}\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$
- Let H be a distribution on some space Ω.
- For $k = 1, 2, \dots,$ sample $\theta_k \sim H$



Multiple samples from the Dirichlet process

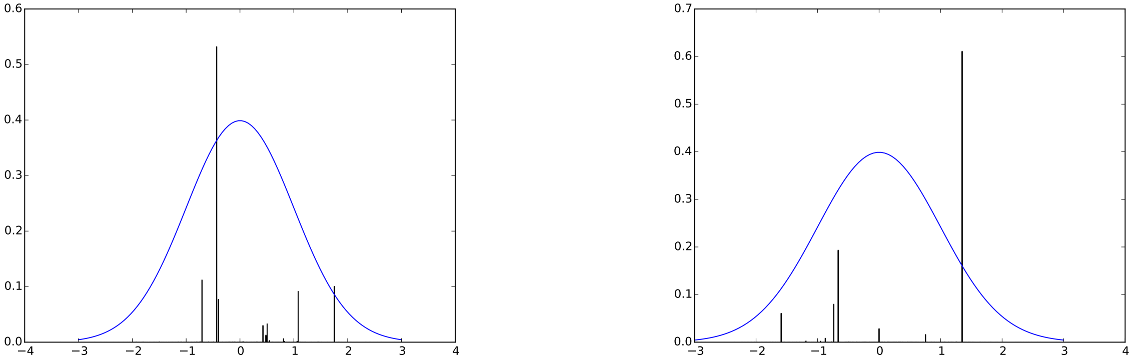
Let's consider two independent samples from the Dirichlet process.



- The atom locations for each sample are iid samples from H.
- If H is continuous, we will never get repeats.
- So, the support of the two distributions are always different.

Multiple samples from the Dirichlet process

Let's consider two independent samples from the Dirichlet process.



- The atom locations for each sample are iid samples from H.
- If H is continuous, we will never get repeats.
- So, the support of the two distributions are always different.

- If we replace our Dirichlet distributions with Dirichlet processes, each atom corresponds to a topic.
- This means that a topic in a given document will never appear in any other documents.
- We can't draw statistical strength across documents.

Sharing topics

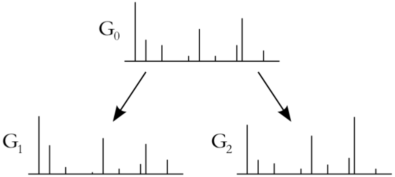
- The reason we won't share topics is because the base measure is continuous, so we have zero probability of picking the same topic twice.
- If we want to pick the same topic twice, we need to use a discrete base measure.
- We want there to be an infinite number of topics, so we want to use an *infinite, discrete* probability distribution as our base measure.

Sharing topics

- The reason we won't share topics is because the base measure is continuous, so we have zero probability of picking the same topic twice.
- If we want to pick the same topic twice, we need to use a discrete base measure.
- We want there to be an infinite number of topics, so we want to use an *infinite, discrete* probability distribution as our base measure.
- Luckily we know how to construct an infinite, discrete probability measure... use a Dirichlet process!

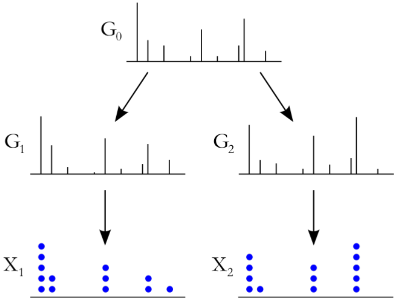
Hierarchical Dirichlet process [Teh et al., 2006]

$$
\begin{aligned}
G_0 \sim & \text{DP}(\gamma, H) \\
G_m \sim & \text{DP}(\alpha, G_0)
\end{aligned}
$$



- We sample a shared distribution over topics $G_0 \sim \text{DP}(\gamma, H)$.
- The concentration parameter γ controls how many high-probability topics we get – small γ leads to a sparser distribution.
- For each document, we then sample a document-specific distribution over topics $G_m \sim \text{DP}(\alpha, G_0)$.
- We generate atom sizes according to a stick breaking process.
- When we pick our atom locations, we will tend to pick high-probability locations in G_0 – multiple sticks can go to the same location.
- G_0 acts as a mean distribution, and α controls how much variation there is between the G_m.

Hierarchical Dirichlet process



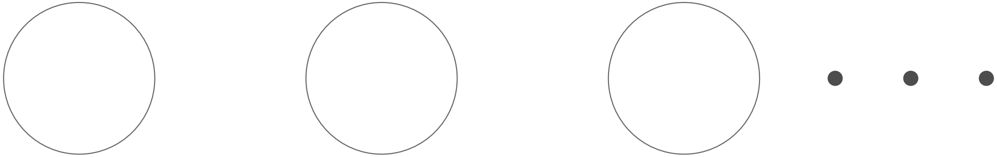
- Once we have our document-specific distributions over topics, we can generate our documents.
- For each topic, sample a distribution over words, $\beta_k \sim \text{Dirichlet}(\eta_1, \ldots, \eta_V)$.
- For each word in the mth document
  - Sample a topic according to $z_{mn} \sim \text{Discrete}(G_m)$
  - Sample a word according to $w_{mn} \sim \text{Discrete}(\beta_{z_{mn}})$

The Chinese restaurant franchise

- As with our previous models, we often want to integrate out the infinite random variables.
- This gives us an extension of the Chinese restaurant process.
- Imagine we have a restaurant franchise with a common menu.
- A single document is represented by a restaurant.
- Words are clustered into tables... remember each table corresponds to a stick in our stick-breaking construction.
- The discrete base measure means that multiple sticks can go to the same location!
- This is equivalent to having multiple tables serving the same dish.

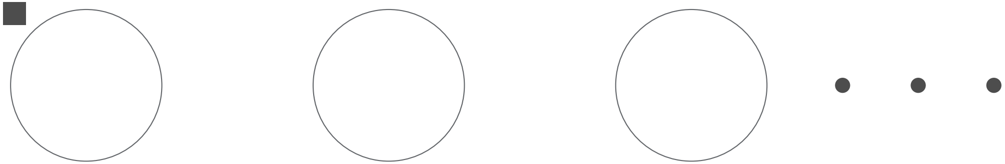
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



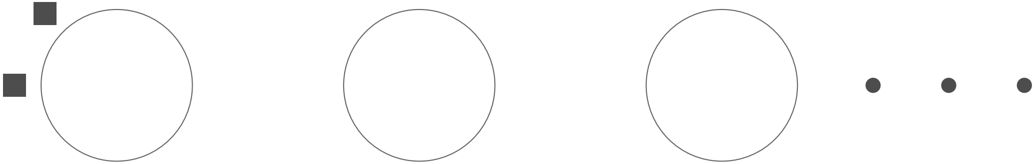
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



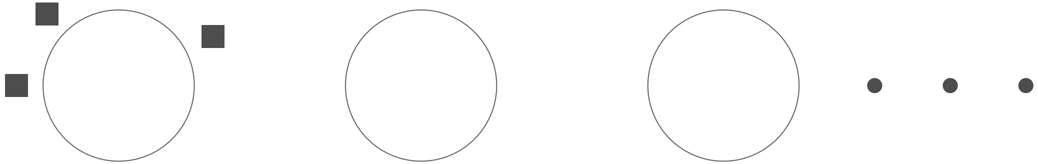
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



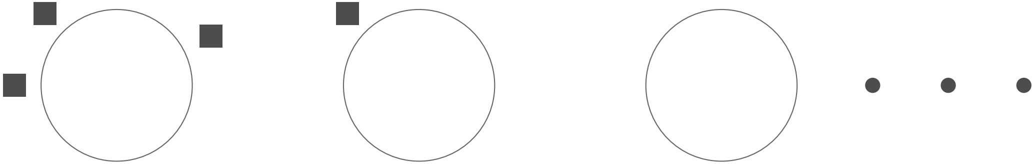
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



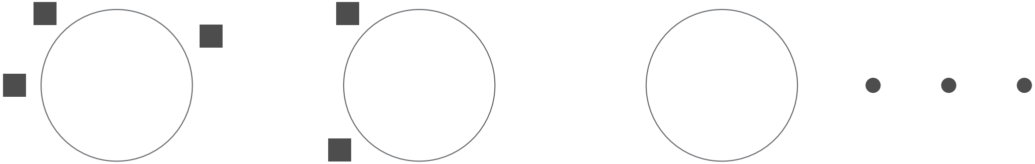
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



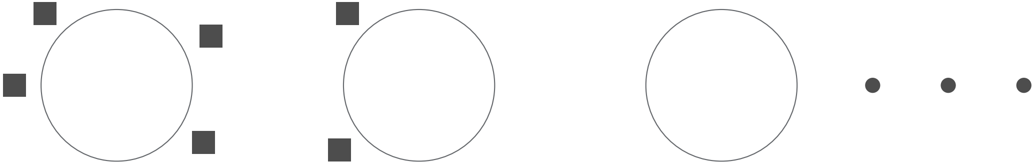
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



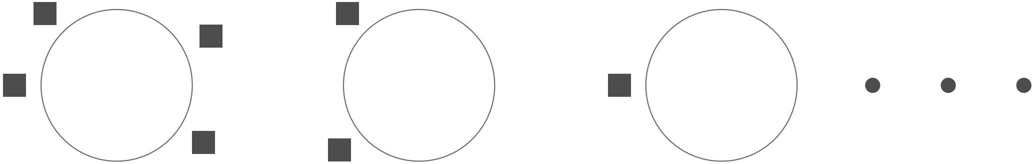
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



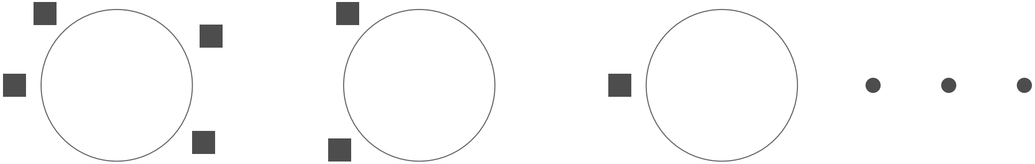
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



The Chinese restaurant franchise

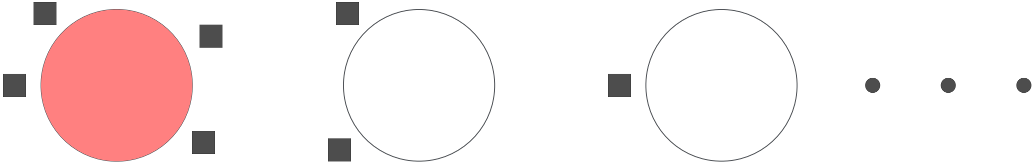
- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



- Each table asks their waiter to pick a dish.
- The waiter considers all dishes that have been served previously in the franchise. He picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the entire franchise.

The Chinese restaurant franchise

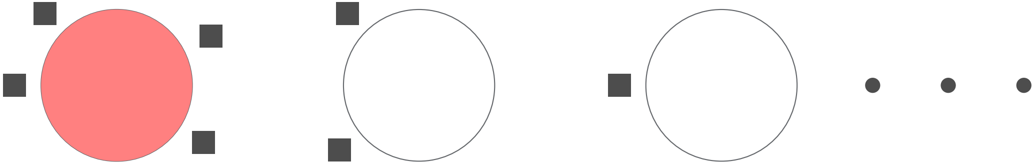
- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



- Each table asks their waiter to pick a dish.
- The waiter considers all dishes that have been served previously in the franchise. He picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the entire franchise.
    - Since this is the first restaurant, the first table gets a new dish.

The Chinese restaurant franchise

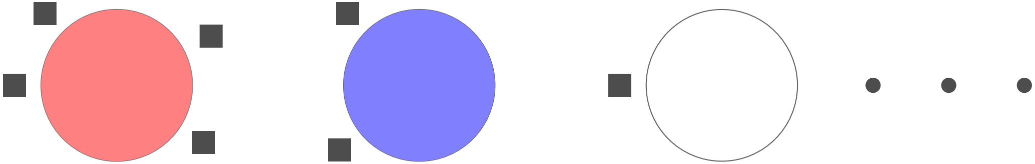
- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



- Each table asks their waiter to pick a dish.
- The waiter considers all dishes that have been served previously in the franchise. He picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the entire franchise.
  - Since this is the first restaurant, the first table gets a new dish.
  - The second table gets the red dish with probability $\frac{1}{1+\gamma}$, or a new dish with probability $\frac{\gamma}{1+\gamma}$.

The Chinese restaurant franchise

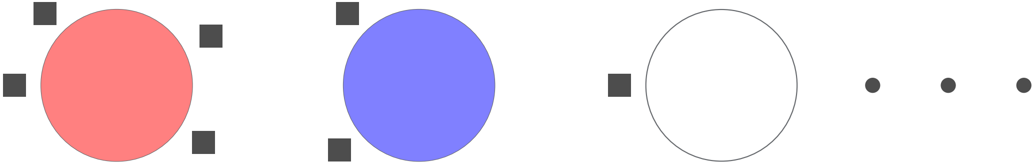
- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



- Each table asks their waiter to pick a dish.
- The waiter considers all dishes that have been served previously in the franchise. He picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the entire franchise.
  - Since this is the first restaurant, the first table gets a new dish.
  - The second table gets the red dish with probability $\frac{1}{1+\gamma}$, or a new dish with probability $\frac{\gamma}{1+\gamma}$.

The Chinese restaurant franchise

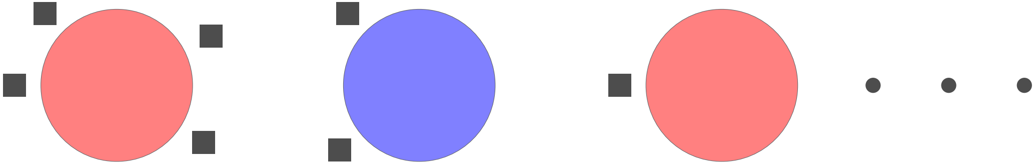
- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α



- Each table asks their waiter to pick a dish.
- The waiter considers all dishes that have been served previously in the franchise. He picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the entire franchise.
  - Since this is the first restaurant, the first table gets a new dish.
  - The second table gets the red dish with probability $\frac{1}{1+\gamma}$, or a new dish with probability $\frac{\gamma}{1+\gamma}$.
  - The third table gets the red dish with probability $\frac{1}{2+\gamma}$, the blue dish with probability $\frac{1}{2+\gamma}$, or a new dish with probability $\frac{\gamma}{2+\gamma}$

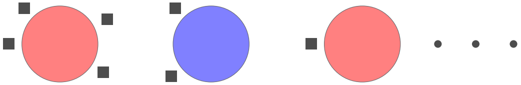
The Chinese restaurant franchise

- Consider the first restaurant (document)
- Customers pick tables according to a Chinese restaurant process with parameter α

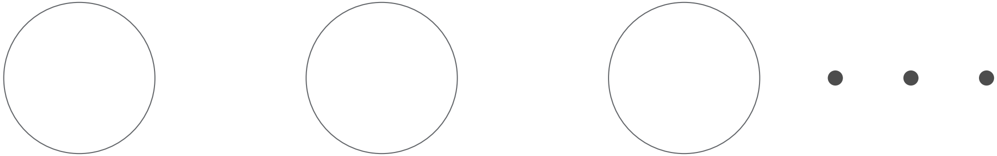


- Each table asks their waiter to pick a dish.
- The waiter considers all dishes that have been served previously in the franchise. He picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the entire franchise.
  - Since this is the first restaurant, the first table gets a new dish.
  - The second table gets the red dish with probability $\frac{1}{1+\gamma}$, or a new dish with probability $\frac{\gamma}{1+\gamma}$.
  - The third table gets the red dish with probability $\frac{1}{2+\gamma}$, the blue dish with probability $\frac{1}{2+\gamma}$, or a new dish with probability $\frac{\gamma}{2+\gamma}$

The Chinese restaurant franchise

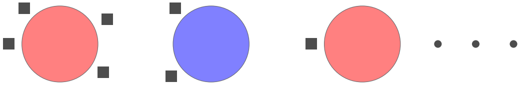


- Let's move on to the second restaurant.

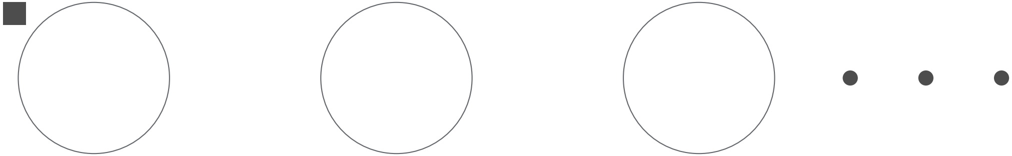


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.

The Chinese restaurant franchise

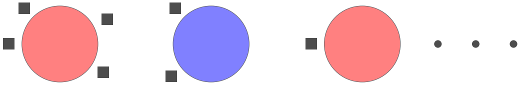


- Let's move on to the second restaurant.

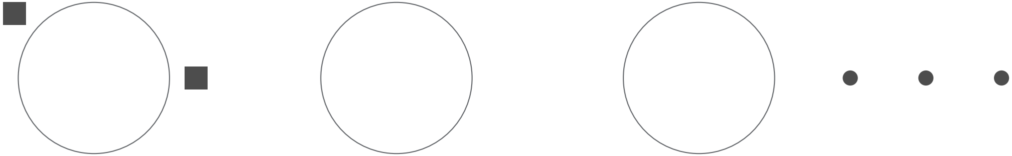


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.

The Chinese restaurant franchise

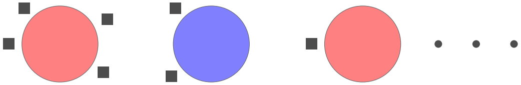


- Let's move on to the second restaurant.

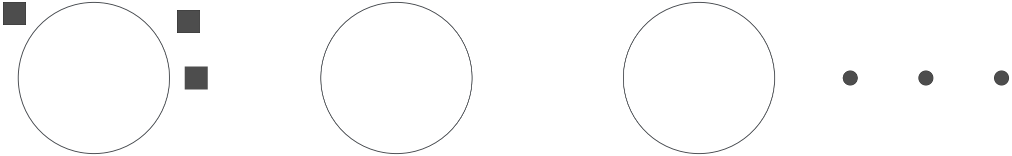


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.

The Chinese restaurant franchise

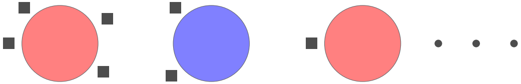


- Let's move on to the second restaurant.

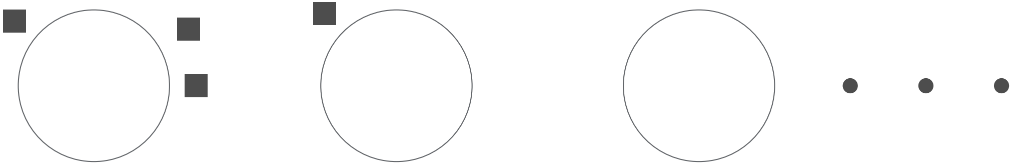


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.

The Chinese restaurant franchise

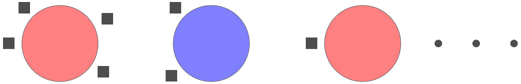


- Let's move on to the second restaurant.

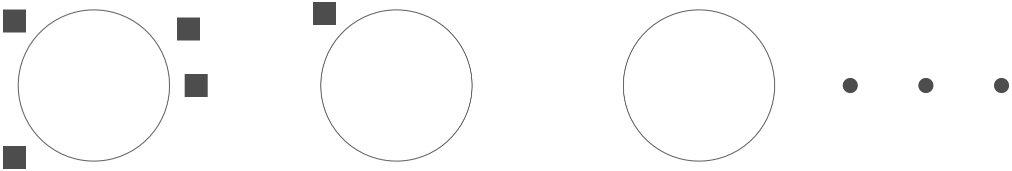


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.

The Chinese restaurant franchise

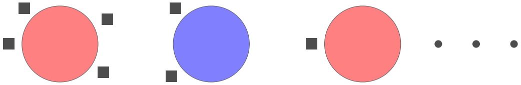


- Let's move on to the second restaurant.

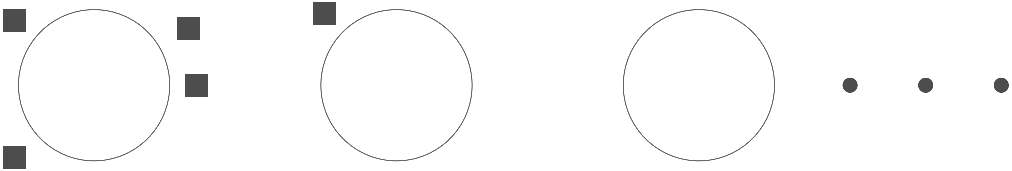


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.

The Chinese restaurant franchise

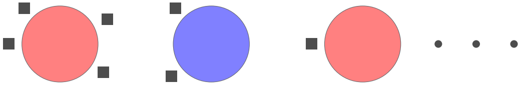


- Let's move on to the second restaurant.

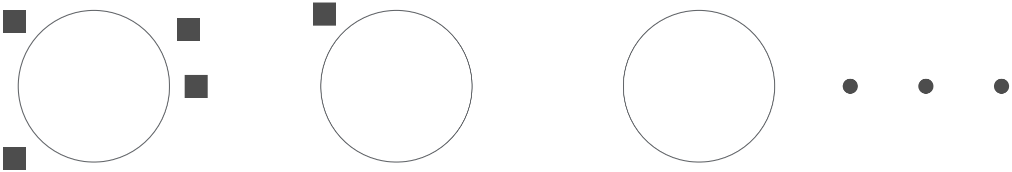


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.
- Each table asks their waiter to pick a dish. The waiter picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the *entire* franchise.

The Chinese restaurant franchise

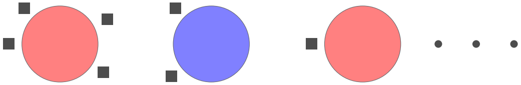


- Let's move on to the second restaurant.

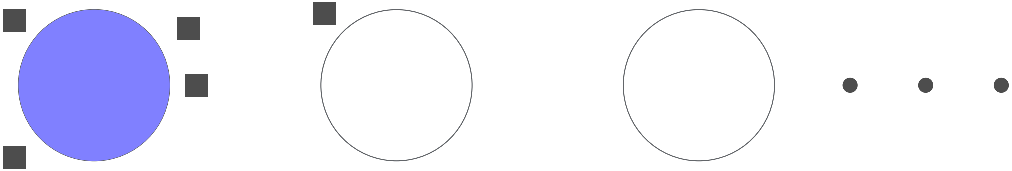


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.
- Each table asks their waiter to pick a dish. The waiter picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the *entire* franchise.
  - The first table gets the red dish with probability $\frac{2}{3+\gamma}$, the blue dish with probability $\frac{1}{3+\gamma}$, or a new dish with probability $\frac{\gamma}{3+\gamma}$.

The Chinese restaurant franchise

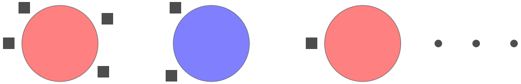


- Let's move on to the second restaurant.

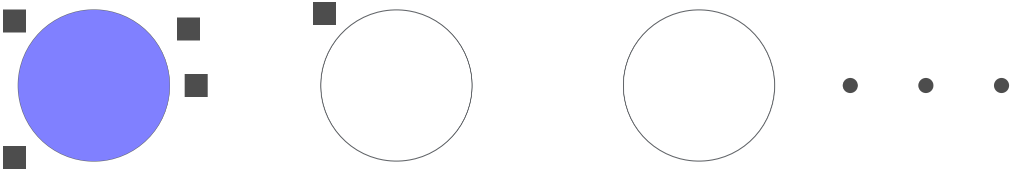


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.
- Each table asks their waiter to pick a dish. The waiter picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the *entire* franchise.
  - The first table gets the red dish with probability $\frac{2}{3+\gamma}$, the blue dish with probability $\frac{1}{3+\gamma}$, or a new dish with probability $\frac{\gamma}{3+\gamma}$.

The Chinese restaurant franchise

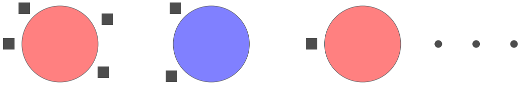


- Let's move on to the second restaurant.

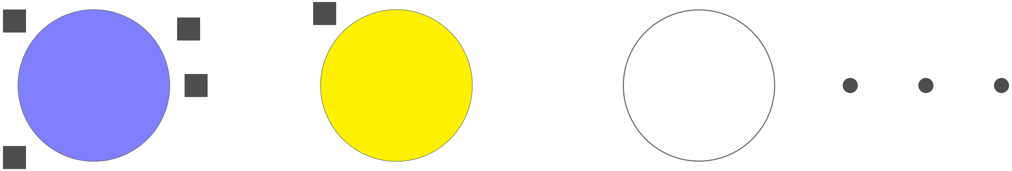


- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.
- Each table asks their waiter to pick a dish. The waiter picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the *entire* franchise.
  - The first table gets the red dish with probability $\frac{2}{3+\gamma}$, the blue dish with probability $\frac{1}{3+\gamma}$, or a new dish with probability $\frac{\gamma}{3+\gamma}$.
  - The second table gets the red dish with probability $\frac{2}{4+\gamma}$, the blue dish with probability $\frac{2}{4+\gamma}$, or a new dish with probability $\frac{\gamma}{4+\gamma}$

The Chinese restaurant franchise



- Let's move on to the second restaurant.



- Customers pick tables according to a Chinese restaurant process with parameter α, *independently* of the first restaurant.
- Each table asks their waiter to pick a dish. The waiter picks an existing dish d with probability proportional to the number of tables n_d that have chosen that dish, across the *entire* franchise.
  - The first table gets the red dish with probability $\frac{2}{3+\gamma}$, the blue dish with probability $\frac{1}{3+\gamma}$, or a new dish with probability $\frac{\gamma}{3+\gamma}$.
  - The second table gets the red dish with probability $\frac{2}{4+\gamma}$, the blue dish with probability $\frac{2}{4+\gamma}$, or a new dish with probability $\frac{\gamma}{4+\gamma}$

An infinite dimensional topic model

- We can use this Chinese restaurant franchise to model a corpus of documents.
    - Each document is a restaurant.
    - Each customer is a word.
    - Each dish is a topic.
- The restaurant process gives us the predictive distributions we need to resample the assignments of customers to tables, and tables to dishes.

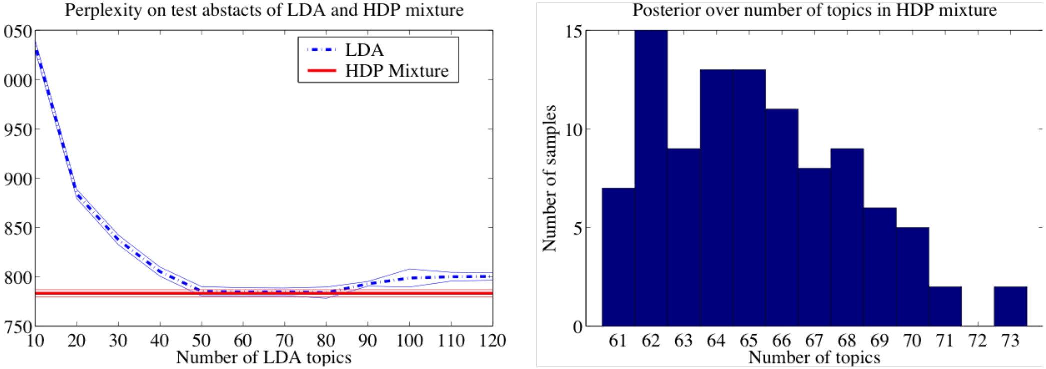


Image from [Teh et al., 2006]

BNP for networks

Why model networks?

- **Prediction**: we have an observed network, and want to predict missing/future interactions
  - e.g. predicting interactions in social or communication networks.
- **Understanding**: we have an observed network, and want to understand its latent structure
  - e.g. community detection, anomaly detection.
- **Network elucidation**: we want to infer a latent network that generated our data
  - e.g. noisy network observations (Twitter), biological interactions

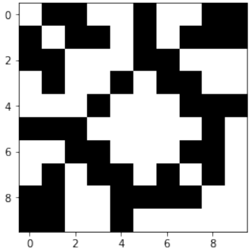
Basic network models: Erdös-Renyi models

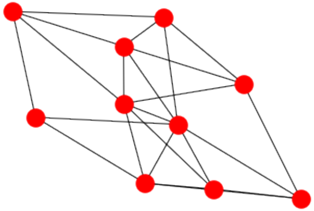
$G(n, p)$ model:

- n nodes, each edge included with probability p.
- As n grows, number of edges grows as n^2.

$G(n, M)$ model:

- n nodes, M edges sampled uniformly without replacement.
- Equivalently, sample uniformly from all networks with n nodes and M edges



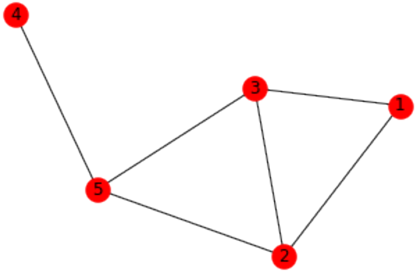


- Neither really look like social networks... they lack structure.
- They also can't grow in a reasonable manner with more observations.

$G(n, M)$ model

The $G(n, M)$ model treats networks as a sequence of links.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \leftrightarrow \begin{matrix} (1,2) \\ (1,3) \\ (2,3) \\ (2,5) \\ (3,5) \\ (4,5) \end{matrix} \leftrightarrow$$

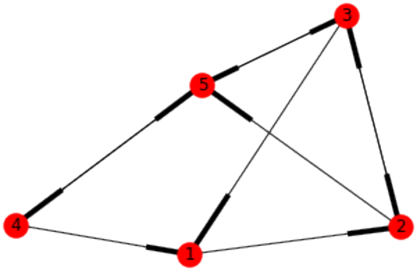


$G(n, M)$ model

The $G(n, M)$ model treats networks as a sequence of links.

- I'm going to modify this slightly to give a directed network

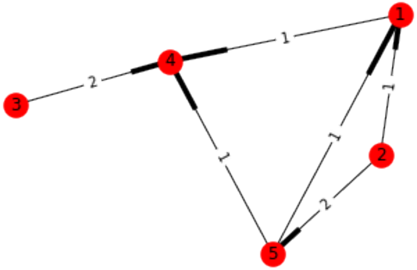
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \leftrightarrow \begin{array}{c} (1,2) \\ (2,3) \\ (2,5) \\ (3,1) \\ (3,2) \\ (3,5) \\ (4,1) \\ (4,5) \\ (5,3) \\ (5,4) \end{array} \leftrightarrow$$



Slight modification

Sampling without replacement is hard If we are OK with **integer-valued** networks, we can get a conditionally iid model by sampling with replacement

$$\begin{array}{c}(5,1)\\(3,4)\\(2,5)\\(3,4)\\(3,3)\\(2,5)\\(1,4)\\(5,4)\\(2,1)\end{array} \quad \leftrightarrow \quad \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \quad \leftrightarrow$$



But, the $G(n, M)$ model assumes n and M are fixed.

- For prediction, we want to be able to grow the number of links M.
- We also want to allow the number of nodes to grow.

A nonparametric version: Dirichlet Network Distributions

Rather than sample pairs (s, r) of nodes from a uniform distribution, we can sample them from a nonparametric distribution f

- Easiest way:

$$\pi \sim \mathrm{DP}(\alpha, H)$$
$$s_i, r_i \overset{iid}{\sim} \pi$$

A nonparametric version: Dirichlet Network Distributions

Rather than sample pairs (s, r) of nodes from a uniform distribution, we can sample them from a nonparametric distribution f

- Easiest way:

$$\pi \sim \mathrm{DP}(\alpha, H)$$
$$s_i, r_i \overset{iid}{\sim} \pi$$

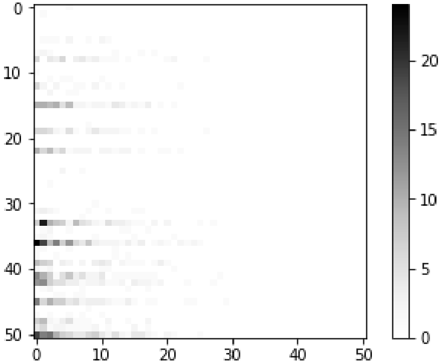
- Assumes “symmetry”... a node is equally likely to be chosen as a “sender” or a “recipient”.
- To break symmetry, we can use two different distributions

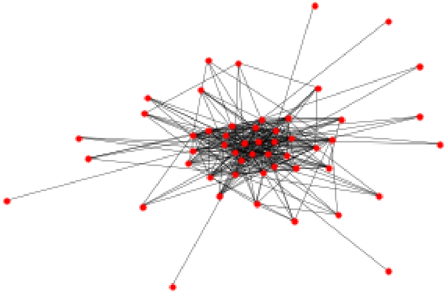
$$s_i \sim \pi_s \qquad\qquad r_i \sim \pi_r$$

- To ensure shared support (i.e. avoid bipartite graph), hierarchically couple π_s and π_r

$$\pi_s \sim \mathrm{DP}(\tau, H) \qquad \pi_r \sim \mathrm{DP}(\tau, H) \qquad H \sim \mathrm{DP}(\gamma, \Theta)$$

Structure of the Dirichlet Network Distribution





- Unbounded number of nodes due to infinite-dimensional support.
- Sparsity and degree distribution are similar to “real” graphs.
- No real structure beyond a preferential attachment-like behavior.

- Real networks have more complex structure than this.
- We see clustering and the formation of cliques... here we only have one cluster.

Adding structure

- We can capture this using a mixture of Dirichlet Network Distributions (MDND)!
  - Each component is a Dirichlet Network Distribution.
  - Different components put high probability on different sets of nodes.
  - A globally shared base measure ensures the component networks share nodes.
- Intuition: Emails clustered by type of person they are to/from.
  - An *email* might belong to a faculty-to-student cluster.
  - This cluster assigns high probability to senders being faculty and receivers being students.
  - An *individual* might have high probability under several clusters.

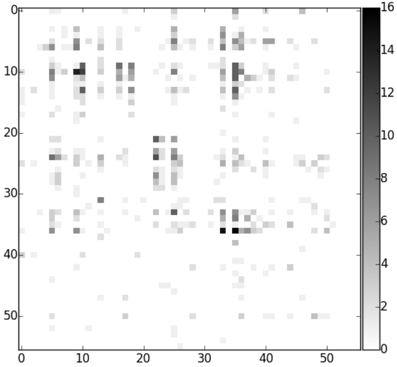
Adding structure

More concretely,

$$
\begin{aligned}
D &\sim \mathrm{DP}(\alpha, \Omega) && \text{distribution over clusters} \\
H &\sim \mathrm{DP}(\gamma, \Theta) && \text{shared distribution over nodes} \\
A_k &\sim \mathrm{DP}(\tau, H) && \text{per-cluster distribution over sender nodes} \\
B_k &\sim \mathrm{DP}(\tau, H) && \text{per-cluster distribution over receiver nodes} \\
c_n &\sim D && \text{pick a cluster for the } n\text{th link} \\
s_n &\sim A_{c_n} && \text{sample a sender...} \\
r_n &\sim B_{c_n} && \text{and a receiver}
\end{aligned}
$$

Adding structure

Now we have some block structure!



- Clustering concentration parameter α controls the number of groups.
- Bottom level concentration parameter τ controls the degree of similarity/degree of overlap between the groups.
- Bottom and top level concentration parameters γ and τ control the total number of nodes and the sparsity of the resulting network.

Many more applications and extensions!

- Incorporating spatio-temporal dynamics – Dependent Dirichlet processes [MacEachern, 1999], Dependent IBPs [Williamson et al., 2010]
- Infinite-context n-gram models [Teh, 2006]
- Hierarchical clustering models [Adams et al., 2010]
- Hidden Markov models with infinitely many states [Teh et al., 2006]
- ...
- If you're coming to NIPS, we'll have a workshop on Bayesian nonparametrics.

Further resources

There are some excellent tutorials on Bayesian nonparametrics, from a Machine Learning perspective:

- *A tutorial on Bayesian nonparametric models*, S.J. Gershman and D.M. Blei, Journal of Mathematical Psychology (56):1-12, 2012.
- The introduction of Erik Sudderth's PhD thesis is a very well-written introduction to Bayesian nonparametrics, particularly the Dirichlet process.
- Any of Tamara Broderick's lectures.
- Yee Whye Teh's lectures from past MLSS's on VideoLectures.

Some a little more Stats-y...

- *Bayesian Nonparametric Models*, P. Orbanz and Y.W. Teh. In Encyclopedia of Machine Learning (Springer), 2010.
- Peter Orbanz's lectures from past MLSS's on VideoLectures.

Further resources

Want to play with code? (Caveat: I've not used all of these...)

- DPs and HDPs:
  - Python: bnpy-dev `https://bitbucket.org/michaelchughes/bnpy-dev/`
  - Julia: BNP.jl `https://github.com/trappmartin/BNP.jl`
  - Matlab: Yee Whye Teh `http://www.stats.ox.ac.uk/ teh/software.html`
  - Several R packages
- IBP:
  - Python: PyIBP `https://github.com/davidandrzej/PyIBP`
  - Matlab: Finale Doshi-Velez `http://people.csail.mit.edu/finale/`

- Great paper on inference in the DP: *Markov chain sampling methods for Dirichlet process mixture models*, RM Neal, Journal of Computational and Graphical Statistics, 9:249-265, 2000.

References I

Adams, R. P., Jordan, M. I., and Ghahramani, Z. (2010).
Tree-structured stick breaking for hierarchical data.
In *Advances in Neural Information Processing Systems*, pages 19–27.

Antoniak, C. (1974).
Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems.
The Annals of Statistics, pages 1152–1174.

Blackwell, D. and MacQueen, J. B. (1973).
Ferguson distributions via Pólya urn schemes.
The annals of statistics, pages 353–355.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003).
Latent dirichlet allocation.
Journal of machine Learning research, 3(Jan):993–1022.

Doshi-Velez, F. and Ghahramani, Z. (2009).
Accelerated sampling for the Indian buffet process.
In *International Conference on Machine Learning*.

Ferguson, T. S. (1973).
A Bayesian analysis of some nonparametric problems.
The Annals of Statistics, pages 209–230.

References II

Griffiths, T. L. and Ghahramani, Z. (2005).
Infinite latent feature models and the Indian buffet process.
In *Advances in Neural Information Processing Systems*.

Hjort, N. L. (1990).
Nonparametric Bayes estimators based on beta processes in models for life history data.
The Annals of Statistics, pages 1259–1294.

Knowles, D. and Ghahramani, Z. (2007).
Infinite sparse factor analysis and infinite independent component analysis.
In *Independent Component Analysis*.

MacEachern, S. N. (1999).
Dependent nonparametric processes.
In *Bayesian Statistical Science*.

Sethuraman, J. (1994).
A constructive definition of Dirichlet priors.
Statistica sinica, pages 639–650.

Teh, T., Jordan, M., Beal, M., and Blei, D. (2006).
Hierarchical Dirichlet processes.
Journal of the American Statistical Association, 101(476):1566–1581.

References III

Teh, Y., Görür, D., and Ghahramani, Z. (2007).
Stick-breaking construction for the Indian buffet process.
In *International Conference on Artificial Intelligence and Statistics*.

Teh, Y. W. (2006).
A hierarchical bayesian language model based on pitman-yor processes.
In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics.

Thibaux, R. and Jordan, M. I. (2007).
Hierarchical beta processes and the Indian buffet process.
In *Artificial Intelligence and Statistics*.

Williamson, S., Orbanz, P., and Ghahramani, Z. (2010).
Dependent Indian buffet processes.
In *Artificial Intelligence and Statistics*.

Zhai, K., Hu, Y., Williamson, S., and Boyd-Graber, J. (2012).
Modeling images using transformed indian buffet processes.
In *International Conference of Machine Learning*.