Bayesian Inference for
Dirichlet-Multinomials and Dirichlet Processes

Mark Johnson
Macquarie University
Sydney, Australia

MLSS “Summer School”
Random variables and “distributed according to” notation

- A *probability distribution* $F$ is a non-negative function from some set $\mathcal{X}$ whose values sum (integrate) to 1.
- A random variable $X$ is *distributed according* to a distribution $F$, or more simply, $X$ has distribution $F$, written $X \sim F$, iff:

$$P(X = x) = F(x) \text{ for all } x$$

(This is for discrete RVs).

- You’ll sometimes see the notion

$$X \mid Y \sim F$$

which means “$X$ is generated conditional on $Y$ with distribution $F$” (where $F$ usually depends on $Y$), i.e.,

$$P(X \mid Y) = F(X \mid Y)$$
Outline

Introduction to Bayesian Inference

Mixture models

Sampling with Markov Chains

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomial mixtures

Topic modeling with Dirichlet multinomial mixtures

Chinese Restaurant Processes
Bayes’ rule

$$P(\text{Hypothesis} \mid \text{Data}) = \frac{P(\text{Data} \mid \text{Hypothesis}) P(\text{Hypothesis})}{P(\text{Data})}$$

- Bayesian’s use Bayes’ Rule to *update beliefs in hypotheses in response to data*
- $P(\text{Hypothesis} \mid \text{Data})$ is the *posterior distribution*,
- $P(\text{Hypothesis})$ is the *prior distribution*,
- $P(\text{Data} \mid \text{Hypothesis})$ is the *likelihood*, and
- $P(\text{Data})$ is a normalising constant sometimes called the *evidence*
Computing the normalising constant

\[ P(\text{Data}) = \sum_{\text{Hypothesis}' \in \mathcal{H}} P(\text{Data}, \text{Hypothesis}') \]

\[ = \sum_{\text{Hypothesis}' \in \mathcal{H}} P(\text{Data} | \text{Hypothesis}') P(\text{Hypothesis}') \]

- If set of hypotheses \( \mathcal{H} \) is small, can calculate \( P(\text{Data}) \) by enumeration
- But often these sums are intractable
Bayesian belief updating

- Idea: treat posterior from last observation as the prior for next
- Consistency follows because likelihood factors
  - Suppose \( d = (d_1, d_2) \). Then the posterior of a hypothesis \( h \) is:

\[
P(h \mid d_1, d_2) \propto P(h) P(d_1, d_2 \mid h) \\
= P(h) P(d_1 \mid h) P(d_2 \mid h, d_1) \\
\propto P(h \mid d_1) P(d_2 \mid h, d_1)
\]

updated prior likelihood
Discrete distributions

- A *discrete distribution* has a finite set of outcomes $1, \ldots, m$
- A discrete distribution is parameterized by a vector $\theta = (\theta_1, \ldots, \theta_m)$, where $P(X = j|\theta) = \theta_j$ (so $\sum_{j=1}^{m} \theta_j = 1$)
  - Example: An $m$-sided die, where $\theta_j =$ prob. of face $j$
- Suppose $X = (X_1, \ldots, X_n)$ and each $X_i|\theta \sim \text{DISCRETE}(\theta)$.
  Then:

  $$P(X|\theta) = \prod_{i=1}^{n} \text{DISCRETE}(X_i; \theta) = \prod_{j=1}^{m} \theta_j^{N_j}$$

  where $N_j$ is the number of times $j$ occurs in $X$.
- Goal of next few slides: compute $P(\theta|X)$
Multinomial distributions

- Suppose $X_i \sim \text{DISCRETE}(\theta)$ for $i = 1, \ldots, n$, and $N_j$ is the number of times $j$ occurs in $X$.
- Then $N|n, \theta \sim \text{MULTI}(\theta, n)$, and

$$P(N|n, \theta) = \frac{n!}{\prod_{j=1}^{m} N_j!} \prod_{j=1}^{m} \theta_j^{N_j}$$

where $n! / \prod_{j=1}^{m} N_j!$ is the number of sequences of values with occurrence counts $N$.

- The vector $N$ is known as a sufficient statistic for $\theta$ because it supplies as much information about $\theta$ as the original sequence $X$ does.
Dirichlet distributions

- **Dirichlet distributions** are probability distributions over multinomial parameter vectors
  - called **Beta distributions** when \( m = 2 \)
- Parameterized by a vector \( \alpha = (\alpha_1, \ldots, \alpha_m) \) where \( \alpha_j > 0 \) that determines the shape of the distribution

\[
\text{DIR}(\theta \mid \alpha) = \frac{1}{C(\alpha)} \prod_{j=1}^{m} \theta_j^{\alpha_j-1}
\]

\[
C(\alpha) = \int_{\Delta} \prod_{j=1}^{m} \theta_j^{\alpha_j-1} d\theta = \frac{\prod_{j=1}^{m} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{m} \alpha_j)}
\]

- \( \Gamma \) is a generalization of the factorial function
- \( \Gamma(k) = (k - 1)! \) for positive integer \( k \)
- \( \Gamma(x) = (x - 1)\Gamma(x - 1) \) for all \( x \)
Plots of the Dirichlet distribution

\[
P(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{j=1}^{m} \alpha_j\right)}{\prod_{j=1}^{m} \Gamma(\alpha_j)} \prod_{k=1}^{m} \theta_k^{\alpha_k - 1}
\]

\[
\alpha = (1,1)
\]
\[
\alpha = (5,2)
\]
\[
\alpha = (0.1,0.1)
\]
Dirichlet distributions as priors for $\theta$

- Generative model:
  
  $\theta \mid \alpha \sim \text{DIR}(\alpha)$
  
  $X_i \mid \theta \sim \text{DISCRETE}(\theta), \quad i = 1, \ldots, n$

- We can depict this as a Bayes net using *plates*, which indicate *replication*
Inference for $\theta$ with Dirichlet priors

- Data $X = (X_1, \ldots, X_n)$ generated i.i.d. from $\text{DISCRETE}(\theta)$
- Prior is $\text{DIR}(\alpha)$. By Bayes Rule, posterior is:

$$P(\theta | X) \propto P(X | \theta) P(\theta)$$

$$\propto \left( \prod_{j=1}^{m} \theta_j^{N_j} \right) \left( \prod_{j=1}^{m} \theta_j^{\alpha_j - 1} \right)$$

$$= \prod_{j=1}^{m} \theta_j^{N_j + \alpha_j - 1}, \text{ so}$$

$$P(\theta | X) = \text{DIR}(N + \alpha)$$

- So if prior is Dirichlet with parameters $\alpha$, posterior is Dirichlet with parameters $N + \alpha$

$\Rightarrow$ can regard Dirichlet parameters $\alpha$ as “pseudo-counts” from “pseudo-data”
Conjugate priors

- If prior is \( \text{DIR}(\alpha) \) and likelihood is i.i.d. \( \text{DISCRETE}(\theta) \), then posterior is \( \text{DIR}(N + \alpha) \)
  \[ \implies \text{prior parameters } \alpha \text{ specify “pseudo-observations”} \]
- A class \( \mathcal{C} \) of prior distributions \( P(H) \) is \text{conjugate} to a class of likelihood functions \( P(D|H) \) iff the posterior \( P(H|D) \) is also a member of \( \mathcal{C} \)
- In general, conjugate priors encode “pseudo-observations”
  - the difference between prior \( P(H) \) and posterior \( P(H|D) \) are the observations in \( D \)
  - but \( P(H|D) \) belongs to same family as \( P(H) \), and can serve as prior for inferences about more data \( D' \)
    \[ \implies \text{must be possible to encode observations } D \text{ using parameters of prior} \]
- In general, the likelihood functions that have conjugate priors belong to the \text{exponential family}
Point estimates from Bayesian posteriors

- A “true” Bayesian prefers to use the full \( P(H|D) \), but sometimes we have to choose a “best” hypothesis
- The *Maximum a posteriori* (MAP) or *posterior mode* is

\[
\hat{H} = \arg\max_H P(H|D) = \arg\max_H P(D|H)P(H)
\]

- The *expected value* \( \mathbb{E}_P[X] \) of \( X \) under distribution \( P \) is:

\[
\mathbb{E}_P[X] = \int x P(X = x) \, dx
\]

The expected value is a kind of average, weighted by \( P(X) \). The *expected value* \( \mathbb{E}[\theta] \) of \( \theta \) is an estimate of \( \theta \).
The posterior mode of a Dirichlet

- The *Maximum a posteriori* (MAP) or *posterior mode* is

\[
\hat{H} = \arg\max_H P(H|D) = \arg\max_H P(D|H) P(H)
\]

- For Dirichlets with parameters \( \alpha \), the MAP estimate is:

\[
\hat{\theta}_j = \frac{\alpha_j - 1}{\sum_{j'=1}^{m}(\alpha_{j'} - 1)}
\]

so if the posterior is \( \text{DIR}(N + \alpha) \), the MAP estimate for \( \theta \) is:

\[
\hat{\theta}_j = \frac{N_j + \alpha_j - 1}{n + \sum_{j'=1}^{m}(\alpha_{j'} - 1)}
\]

- If \( \alpha = 1 \) then \( \hat{\theta}_j = N_j / n \), which is also the *maximum likelihood estimate* (MLE) for \( \theta \)
The expected value of $\theta$ for a Dirichlet

- The *expected value* $E_P[X]$ of $X$ under distribution $P$ is:

$$E_P[X] = \int x P(X = x) \, dx$$

- For Dirichlets with parameters $\alpha$, the expected value of $\theta_j$ is:

$$E_{\text{DIR}}(\alpha)[\theta_j] = \frac{\alpha_j}{\sum_{j'=1}^{m} \alpha_{j'}}$$

- Thus if the posterior is $\text{DIR}(N + \alpha)$, the expected value of $\theta_j$ is:

$$E_{\text{DIR}}(N+\alpha)[\theta_j] = \frac{N_j + \alpha_j}{n + \sum_{j'=1}^{m} \alpha_{j'}}$$

- $E[\theta]$ smooths or regularizes the MLE by adding pseudo-counts $\alpha$ to $N$
Sampling from a Dirichlet

\[ \theta | \alpha \sim \text{DIR}(\alpha) \iff P(\theta | \alpha) = \frac{1}{C(\alpha)} \prod_{j=1}^{m} \theta_{j}^{\alpha_j - 1}, \text{ where:} \]

\[ C(\alpha) = \frac{\prod_{j=1}^{m} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{m} \alpha_j)} \]

- There are several algorithms for producing samples from \( \text{DIR}(\alpha) \). A simple one relies on the following result:
- If \( V_k \sim \text{GAMMA}(\alpha_k) \) and \( \theta_k = \frac{V_k}{\sum_{k'=1}^{m} V_{k'}} \), then \( \theta \sim \text{DIR}(\alpha) \)
- This leads to the following algorithm for producing a sample \( \theta \) from \( \text{DIR}(\alpha) \)
  - Sample \( v_k \) from \( \text{GAMMA}(\alpha_k) \) for \( k = 1, \ldots, m \)
  - Set \( \theta_k = \frac{v_k}{\sum_{k'=1}^{m} v_{k'}} \)
Posterior with Dirichlet priors

\[ \theta \mid \alpha \sim \text{DIR}(\alpha) \]

\[ X_i \mid \theta \sim \text{DISCRETE}(\theta), \quad i = 1, \ldots, n \]

- **Integrate out** \( \theta \) **to calculate posterior probability of** \( X \)

\[
P(X|\alpha) = \int P(X, \theta|\alpha) \, d\theta = \int_\Delta P(X|\theta) P(\theta|\alpha) \, d\theta
\]

\[
= \int_\Delta \left( \prod_{j=1}^m \theta_j^{N_j} \right) \left( \frac{1}{C(\alpha)} \prod_{j=1}^m \theta_j^{\alpha_j-1} \right) \, d\theta
\]

\[
= \frac{1}{C(\alpha)} \int_\Delta \prod_{j=1}^m \theta_j^{N_j+\alpha_j-1} \, d\theta
\]

\[
= \frac{C(N + \alpha)}{C(\alpha)}, \quad \text{where} \quad C(\alpha) = \frac{\prod_{j=1}^m \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^m \alpha_j)}
\]

- **Collapsed Gibbs samplers** and the **Chinese Restaurant Process** rely on this result
Predictive distribution for Dirichlet-Multinomial

- The **predictive distribution** is the distribution of observation $X_{n+1}$ given observations $X = (X_1, \ldots, X_n)$ and prior $\text{DIR}(\alpha)$

$$P(X_{n+1} = k \mid X, \alpha) = \int_{\Delta} P(X_{n+1} = k \mid \theta) P(\theta \mid X, \alpha) \, d\theta$$

$$= \int_{\Delta} \theta_k \text{DIR} (\theta \mid N + \alpha) \, d\theta$$

$$= \frac{N_k + \alpha_k}{\sum_{j=1}^{m} N_j + \alpha_j}$$
Example: rolling a die

- Data \( d = (2, 5, 4, 2, 6) \)

\[
\begin{align*}
\alpha & = (1,1,1,1,1,1) \\
\alpha & = (1,2,1,1,1,1) \\
\alpha & = (1,2,1,1,2,1) \\
\alpha & = (1,2,1,2,2,1) \\
\alpha & = (1,3,1,2,2,1) \\
\alpha & = (1,3,1,2,2,2)
\end{align*}
\]
Inference in complex models

- If the model is simple enough we can calculate the posterior exactly (conjugate priors)
- When the model is more complicated, we can only approximate the posterior
- **Variational Bayes** calculate the function closest to the posterior within a class of functions
- **Sampling algorithms** produce samples from the posterior distribution
  - *Markov chain Monte Carlo algorithms* (MCMC) use a Markov chain to produce samples
    - A *Gibbs sampler* is a particular MCMC algorithm
- **Particle filters** are a kind of *on-line* sampling algorithm (on-line algorithms only make one pass through the data)
Outline

Introduction to Bayesian Inference

Mixture models

Sampling with Markov Chains

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomial mixtures

Topic modeling with Dirichlet multinomial mixtures

Chinese Restaurant Processes
Mixture models

- Observations $X_i$ are a *mixture* of $\ell$ source distributions $F(\theta_k), k = 1, \ldots, \ell$
- The value of $Z_i$ specifies which source distribution is used to generate $X_i$ ($Z$ is like a switch)
- If $Z_i = k$, then $X_i \sim F(\theta_k)$
- Here we assume the $Z_i$ are not observed, i.e., *hidden*

\[
X_i \mid Z_i, \theta \sim F(\theta_{Z_i}) \quad i = 1, \ldots, n
\]
Applications of mixture models

- **Blind source separation**: data $X_i$ come from $\ell$ different sources
  - Which $X_i$ come from which source?
    ($Z_i$ specifies the source of $X_i$)
  - What are the sources?
    ($\theta_k$ specifies properties of source $k$)
- $X_i$ could be a document and $Z_i$ the topic of $X_i$
- $X_i$ could be an image and $Z_i$ the object(s) in $X_i$
- $X_i$ could be a person’s actions and $Z_i$ the “cause” of $X_i$
- These are *unsupervised learning problems*, which are kinds of *clustering problems*
- In a Bayesian setting, compute posterior $P(Z, \theta|X)$
  *But how can we compute this?*
Dirichlet Multinomial mixtures

\[ \begin{align*}
\phi & \mid \beta \sim \text{DIR}(\beta) \\
Z_i & \mid \phi \sim \text{DISCRETE}(\phi) \quad i = 1, \ldots, n \\
\theta_k & \mid \alpha \sim \text{DIR}(\alpha) \quad k = 1, \ldots, \ell \\
X_{i,j} \mid Z_i, \theta & \sim \text{DISCRETE}(\theta_{Z_i}) \quad i = 1, \ldots, n; j = 1, \ldots, d_i
\end{align*} \]

- \( Z_i \) is generated from a multinomial \( \phi \)
- Dirichlet priors on \( \phi \) and \( \theta_k \)
- Easy to modify this framework for other applications
- Why does each observation \( X_i \) consist of \( d_i \) elements?
- What effect do the priors \( \alpha \) and \( \beta \) have?
Outline

Introduction to Bayesian Inference

Mixture models

**Sampling with Markov Chains**

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomomial mixtures

Topic modeling with Dirichlet multinomomial mixtures

Chinese Restaurant Processes
Why sample?

• Setup: Bayes net has variables $X$, whose value $x$ we observe, and variables $Y$, whose value we don’t know
  ▶ $Y$ includes any parameters we want to estimate, such as $\theta$
• Goal: compute the expected value of some function $f$:

$$E[f|X = x] = \sum_y f(x, y) P(Y = y|X = x)$$

  ▶ E.g., $f(x, y) = 1$ if $x_1$ and $x_2$ are both generated from same hidden state, and 0 otherwise
• In what follows, everything is conditioned on $X = x$, so take $P(Y)$ to mean $P(Y|X = x)$
• Suppose we can produce $n$ samples $y^{(t)}$, where $Y^{(t)} \sim P(Y)$. Then we can estimate:

$$E[f|X = x] = \frac{1}{n} \sum_{t=1}^{n} f(x, y^{(t)})$$
Markov chains

- A (first-order) **Markov chain** is a distribution over random variables $S^{(0)}, \ldots, S^{(n)}$ all ranging over the same *state space* $S$, where:

\[
P(S^{(0)}, \ldots, S^{(n)}) = P(S^{(0)}) \prod_{t=0}^{n-1} P(S^{(t+1)}|S^{(t)})
\]

$S^{(t+1)}$ is *conditionally independent* of $S^{(0)}, \ldots, S^{(t-1)}$ given $S^{(t)}$

- A Markov chain in **homogeneous** or **time-invariant** iff:

\[
P(S^{(t+1)} = s'|S^{(t)} = s) = P_{s',s} \quad \text{for all } t, s, s'
\]

The matrix $P$ is called the *transition probability matrix* of the Markov chain

- If $P(S^{(t)} = s) = \pi^{(t)}_s$ (i.e., $\pi^{(t)}$ is a vector of state probabilities at time $t$) then:
  - $\pi^{(t+1)} = P \pi^{(t)}$
  - $\pi^{(t)} = P^t \pi^{(0)}$
Ergodicity

• A Markov chain with tpm $P$ is ergodic iff there is a positive integer $m$ s.t. all elements of $P^m$ are positive (i.e., there is an $m$-step path between any two states)

• Informally, an ergodic Markov chain “forgets” its past states

• Theorem: For each homogeneous ergodic Markov chain with tpm $P$ there is a unique limiting distribution $D_P$, i.e., as $n$ approaches infinity, the distribution of $S_n$ converges on $D_P$

• $D_P$ is called the stationary distribution of the Markov chain

• Let $\pi$ be the vector representation of $D_P$, i.e., $D_P(y) = \pi_y$. Then:

\[
\pi = P \pi, \quad \text{and} \\
\pi = \lim_{n \to \infty} P^n \pi^{(0)} \quad \text{for every initial distribution } \pi^{(0)}
\]
Using a Markov chain for inference of $P(Y)$

- Set the state space $S$ of the Markov chain to the range of $Y$ ($S$ may be *astronomically large*)
- Find a tpm $P$ such that $P(Y) \sim D_P$
- “Run” the Markov chain, i.e.,
  - Pick $y^{(0)}$ somehow
  - For $t = 0, \ldots, n - 1$:
    - sample $y^{(t+1)}$ from $P(\mathbf{Y}^{(t+1)}|\mathbf{Y}^{(t)} = y^{(t)})$, i.e., from $P_{\cdot|y^{(t)}}$
  - After discarding the first *burn-in* samples, use remaining samples to calculate statistics
- **WARNING:** in general the samples $y^{(t)}$ are *not independent*
Outline

Introduction to Bayesian Inference

Mixture models

Sampling with Markov Chains

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomial mixtures

Topic modeling with Dirichlet multinomial mixtures

Chinese Restaurant Processes
The Gibbs sampler

- The Gibbs sampler is useful when:
  - \( \mathbf{Y} \) is multivariate, i.e., \( \mathbf{Y} = (Y_1, \ldots, Y_m) \), and
  - easy to sample from \( P(Y_j|\mathbf{Y}_{-j}) \)

- The **Gibbs sampler** for \( P(\mathbf{Y}) \) is the tpm \( P = \prod_{j=1}^{m} P^{(j)} \), where:

\[
    P^{(j)}_{y',y} = \begin{cases} 
      0 & \text{if } y'_{-j} \neq y_{-j} \\
      P(Y_j = y'_j|\mathbf{Y}_{-j} = y_{-j}) & \text{if } y'_{-j} = y_{-j}
    \end{cases}
\]

- Informally, the Gibbs sampler cycles through each of the variables \( Y_j \), replacing the current value \( y_j \) with a sample from \( P(Y_j|\mathbf{Y}_{-j} = y_{-j}) \)

- There are *sequential scan* and *random scan* variants of Gibbs sampling
A simple example of Gibbs sampling

\[
P(Y_1, Y_2) = \begin{cases} 
  c & \text{if } |Y_1| < 5, |Y_2| < 5 \text{ and } |Y_1 - Y_2| < 1 \\
  0 & \text{otherwise}
\end{cases}
\]

- The Gibbs sampler for \( P(Y_1, Y_2) \) samples repeatedly from:
  \[
P(Y_2 | Y_1) = \text{UNIFORM}(\max(-5, Y_1 - 1), \min(5, Y_1 + 1))
  \]
  \[
P(Y_1 | Y_2) = \text{UNIFORM}(\max(-5, Y_2 - 1), \min(5, Y_2 + 1))
  \]

**Sample run**

<table>
<thead>
<tr>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>-0.119</td>
</tr>
<tr>
<td>0.363</td>
<td>-0.119</td>
</tr>
<tr>
<td>0.363</td>
<td>0.146</td>
</tr>
<tr>
<td>-0.681</td>
<td>0.146</td>
</tr>
<tr>
<td>-0.681</td>
<td>-1.551</td>
</tr>
</tbody>
</table>
A non-ergodic Gibbs sampler

\[ P(Y_1, Y_2) = \begin{cases} 
  c & \text{if } 1 < Y_1, Y_2 < 5 \text{ or } -5 < Y_1, Y_2 < -1 \\
  0 & \text{otherwise}
\end{cases} \]

- The Gibbs sampler for \( P(Y_1, Y_2) \), initialized at (2,2), samples repeatedly from:

\[
P(Y_2|Y_1) = \text{UNIFORM}(1, 5) \\
P(Y_1|Y_2) = \text{UNIFORM}(1, 5)
\]

I.e., *never visits the negative values of \( Y_1, Y_2 \)*

**Sample run**

<table>
<thead>
<tr>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2.72</td>
</tr>
<tr>
<td>2.84</td>
<td>2.72</td>
</tr>
<tr>
<td>2.84</td>
<td>4.71</td>
</tr>
<tr>
<td>2.63</td>
<td>4.71</td>
</tr>
<tr>
<td>2.63</td>
<td>4.52</td>
</tr>
<tr>
<td>1.11</td>
<td>4.52</td>
</tr>
</tbody>
</table>
Why does the Gibbs sampler work?

- The Gibbs sampler tpm is $P = \prod_{j=1}^{m} P^{(j)}$, where $P^{(j)}$ replaces $y_j$ with a sample from $P(Y_j|Y_{-j} = y_{-j})$ to produce $y'$
- But if $y$ is a sample from $P(Y)$, then so is $y'$, since $y'$ differs from $y$ only by replacing $y_j$ with a sample from $P(Y_j|Y_{-j} = y_{-j})$
- Since $P^{(j)}$ maps samples from $P(Y)$ to samples from $P(Y)$, so does $P$
  $\Rightarrow$ $P(Y)$ is a stationary distribution for $P$
- If $P$ is ergodic, then $P(Y)$ is the unique stationary distribution for $P$, i.e., the sampler converges to $P(Y)$
Gibbs sampling with Bayes nets

- Gibbs sampler: update $y_j$ with sample from $P(Y_j|Y_{-j}) \propto P(Y_j, Y_{-j})$
- Only need to evaluate terms that depend on $Y_j$ in Bayes net factorization
  - $Y_j$ appears once in a term $P(Y_j|Y_{Pa_j})$
  - $Y_j$ can appear multiple times in terms $P(Y_k|\ldots, Y_j, \ldots)$
- In graphical terms, need to know value of:
  - $Y_j$'s parents
  - $Y_j$'s children, and their other parents
Outline

Introduction to Bayesian Inference

Mixture models

Sampling with Markov Chains

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomial mixtures

Topic modeling with Dirichlet multinomial mixtures

Chinese Restaurant Processes
Dirichlet-Multinomial mixtures

\[ \phi \mid \beta \sim \text{DIR}(\beta) \]
\[ Z_i \mid \phi \sim \text{DISCRETE}(\phi) \quad i = 1, \ldots, n \]
\[ \theta_k \mid \alpha \sim \text{DIR}(\alpha) \quad k = 1, \ldots, \ell \]
\[ X_{i,j} \mid Z_i, \theta \sim \text{DISCRETE}(\theta_{Z_i}) \quad i = 1, \ldots, n; j = 1, \ldots, d_i \]

\[
P(\phi, Z, \theta, X \mid \alpha, \beta) = \frac{1}{C(\beta)} \prod_{k=1}^{\ell} \left( \phi_k^{\beta_k-1+N_k(Z)} \right) \]
\[
\quad \cdot \frac{1}{C(\alpha)} \prod_{j=1}^{m} \theta_{k,j}^{\alpha_j-1+\sum_{i:Z_i=k} N_j(X_i)}
\]

where

\[ C(\alpha) = \frac{\prod_{j=1}^{m} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{m} \alpha_j)} \]
Gibbs sampling for D-M mixtures

\[
\begin{align*}
\phi & \mid \beta \sim \text{Dir}(\beta) \\
Z_i & \mid \phi \sim \text{Discrete}(\phi) \quad i = 1, \ldots, n \\
\theta_k & \mid \alpha \sim \text{Dir}(\alpha) \quad k = 1, \ldots, \ell \\
X_{i,j} & \mid Z_i, \theta \sim \text{Discrete}(\theta_{Z_i}) \quad i = 1, \ldots, n; j = 1, \ldots, d_i
\end{align*}
\]

\[
P(\phi \mid Z, \beta) = \text{Dir}(\phi; \beta + N(Z))
\]

\[
P(Z_i = k \mid \phi, \theta, X_i) \propto \phi_k \prod_{j=1}^{m} \theta_{k,j}^{N_j(X_i)}
\]

\[
P(\theta_k \mid \alpha, X, Z) = \text{Dir}(\theta_k; \alpha + \sum_{i:Z_i=k} N(X_i))
\]
Collapsed Dirichlet Multinomial mixtures

\[
P(Z|\beta) = \frac{C(N(Z) + \beta)}{C(\beta)}
\]

\[
P(X|\alpha, Z) = \prod_{k=1}^{\ell} \frac{C(\alpha + \sum_{i:Z_i=k} N(X_i))}{C(\alpha)}, \text{ so}
\]

\[
P(Z_i = k|Z_{-i}, \alpha, \beta) \propto \frac{N_k(Z_{-i}) + \beta_k}{n-1 + \beta} \cdot \frac{C(\alpha + \sum_{i' \neq i:Z_{i'}=k} N(X_{i'}))}{C(\alpha + \sum_{i' \neq i:Z_{i'}=k} N(X_{i'}))}
\]

- \(P(Z_i = k|Z_{-i}, \alpha, \beta)\) is proportional to the prob. of generating:
  - \(Z_i = k\), given the other \(Z_{-i}\), and
  - \(X_i\) in cluster \(k\), given \(X_{-i}\) and \(Z_{-i}\)
Gibbs sampling for Dirichlet multinomial mixtures

- Each $X_i$ could be generated from one of several Dirichlet multinomials
- The variable $Z_i$ indicates the source for $X_i$
- The *uncollapsed sampler* samples $Z, \theta$ and $\phi$
- The *collapsed sampler* integrates out $\theta$ and $\phi$ and just samples $Z$
- Collapsed samplers often (but not always) converge faster than uncollapsed samplers
- Collapsed samplers are usually easier to implement
Outline

Introduction to Bayesian Inference

Mixture models

Sampling with Markov Chains

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomial mixtures

Topic modeling with Dirichlet multinomial mixtures

Chinese Restaurant Processes
Topic modeling of child-directed speech

- Data: Adam, Eve and Sarah’s mothers’ child-directed utterances
  
  I like it.
  why don’t you read Shadow yourself?
  that’s a terribly small horse for you to ride.
  why don’t you look at some of the toys in the basket.
  want to?
  do you want to see what I have?
  what is that?
  not in your mouth.

- 59,959 utterances, composed of 337,751 words
Uncollapsed Gibbs sampler for topic model

- Data consists of “documents” $X_i$
- Each $X_i$ is a sequence of “words” $X_{i,j}$
- Initialize by *randomly* assign each document $X_i$ to a topic $Z_i$
- Repeat the following:
  - Replace $\phi$ with a sample from a Dirichlet with parameters $\beta + N(Z)$
  - For each topic $k$, replace $\theta_k$ with a sample from a Dirichlet with parameters $\alpha + \sum_{i:Z_i=k} N(X_i)$
  - For each document $i$, replace $Z_i$ with a sample from

$$P(Z_i = k | \phi, \theta, X_i) \propto \phi_k \prod_{j=1}^m \theta_{k,j}^{N_j(X_i)}$$
Collapsed Gibbs sampler for topic model

- Initialize by *randomly* assign each document $X_i$ to a topic $Z_i$
- Repeat the following:
  - For each document $i$ in $1, \ldots, n$ (in random order):
    - Replace $Z_i$ with a random sample from $P(Z_i|Z_{-i}, \alpha, \beta)$

$$
P(Z_i = k|Z_{-i}, \alpha, \beta) \propto \frac{N_k(Z_{-i}) + \beta_k}{n - 1 + \beta} \cdot \frac{C(\alpha + \sum_{i' \neq i:Z_{i'} = k} N(X_{i'}) + N(X_i))}{C(\alpha + \sum_{i' \neq i:Z_{i'} = k} N(X_{i'}))}
$$
Topics assigned after 100 iterations

1  big drum?
3  horse.
8  who is that?
9  those are checkers.
3  two checkers # yes.
1  play checkers?
1  big horn?
2  get over # Mommy.
1  shadow?
9  I like it.
1  why don’t you read Shadow yourself?
9  that’s a terribly small horse for you to ride.
2  why don’t you look at some of the toys in the basket.
1  want to?
1  do you want to see what I have?
8  what is that?
2  not in your mouth.
2  let me put them together.
2  no # put floor.
3  no # that’s his pencil.
3  that’s not Daddy # that’s Colin.
### Most probable words in each cluster

<table>
<thead>
<tr>
<th>P(Z=4) = 0.4334</th>
<th>P(Z=9) = 0.3111</th>
<th>P(Z=7) = 0.2555</th>
<th>P(Z=3) = 5.003e-05</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>P(X</td>
<td>Z)</td>
<td>X</td>
</tr>
<tr>
<td>.</td>
<td>0.12526</td>
<td>?</td>
<td>0.19147</td>
</tr>
<tr>
<td>#</td>
<td>0.045402</td>
<td>you</td>
<td>0.062577</td>
</tr>
<tr>
<td>you</td>
<td>0.040475</td>
<td>what</td>
<td>0.061256</td>
</tr>
<tr>
<td>the</td>
<td>0.030259</td>
<td>that</td>
<td>0.022295</td>
</tr>
<tr>
<td>it</td>
<td>0.024154</td>
<td>the</td>
<td>0.022126</td>
</tr>
<tr>
<td>I</td>
<td>0.021848</td>
<td>#</td>
<td>0.021809</td>
</tr>
<tr>
<td>to</td>
<td>0.018473</td>
<td>is</td>
<td>0.021683</td>
</tr>
<tr>
<td>don’t</td>
<td>0.015473</td>
<td>do</td>
<td>0.016127</td>
</tr>
<tr>
<td>a</td>
<td>0.013662</td>
<td>it</td>
<td>0.015927</td>
</tr>
<tr>
<td>?</td>
<td>0.013459</td>
<td>a</td>
<td>0.015092</td>
</tr>
<tr>
<td>in</td>
<td>0.011708</td>
<td>to</td>
<td>0.013783</td>
</tr>
<tr>
<td>on</td>
<td>0.011064</td>
<td>did</td>
<td>0.012631</td>
</tr>
<tr>
<td>your</td>
<td>0.010145</td>
<td>are</td>
<td>0.011427</td>
</tr>
<tr>
<td>and</td>
<td>0.009578</td>
<td>what’s</td>
<td>0.011195</td>
</tr>
<tr>
<td>that</td>
<td>0.0093303</td>
<td>your</td>
<td>0.0098961</td>
</tr>
<tr>
<td>have</td>
<td>0.0088019</td>
<td>huh</td>
<td>0.0082591</td>
</tr>
<tr>
<td>no</td>
<td>0.0082514</td>
<td>want</td>
<td>0.0076782</td>
</tr>
<tr>
<td>put</td>
<td>0.0067486</td>
<td>where</td>
<td>0.0072346</td>
</tr>
<tr>
<td>know</td>
<td>0.0064239</td>
<td>why</td>
<td>0.0070656</td>
</tr>
</tbody>
</table>

47 / 73
Remarks on cluster results

- The samplers cluster words by clustering the documents they appear in, and cluster documents by clustering the words that appear in them.
- Even though there were $\ell = 10$ clusters and $\alpha = 1, \beta = 1$, typically only 4 clusters were occupied after convergence.
- Words $x$ with high marginal probability $P(X = x)$ are typically so frequent that they occur in all clusters.
- Listing the most probable words in each cluster may not be a good way of characterizing the clusters.
- Instead, we can Bayes invert and find the words that are most strongly associated with each class.

$$P(Z = k \mid X = x) = \frac{N_{k,x}(Z, X) + \epsilon}{N_x(X) + \epsilon \ell}$$
## Purest words of each cluster

| X             | P(Z|X) | X             | P(Z|X) | X             | P(Z|X) | X             | P(Z|X) |
|---------------|-------|---------------|-------|---------------|-------|---------------|-------|
| I’ll          | 0.97168 | d(o)          | 0.97138 | 0             | 0.94715 | quack         | 0.6434 |
| we’ll         | 0.96486 | what’s        | 0.95242 | mmhm          | 0.944  | .             | 0.0001 |
| c(o)me       | 0.95319 | what’re       | 0.94348 | www           | 0.90244 |               |       |
| you’ll        | 0.95238 | happened      | 0.93722 | m:hm          | 0.83019 |               |       |
| may          | 0.94845 | hmm           | 0.93343 | uhhuh         | 0.81667 |               |       |
| let’s         | 0.947  | whose         | 0.92437 | uh(uh)        | 0.78571 |               |       |
| thought       | 0.94382 | what          | 0.9227  | uhhuh         | 0.77551 |               |       |
| won’t         | 0.93645 | where’s       | 0.92241 | that’s        | 0.7755  |               |       |
| come          | 0.93588 | doing         | 0.90196 | yep           | 0.76531 |               |       |
| let           | 0.93255 | where’d       | 0.9009  | um            | 0.76282 |               |       |
| I             | 0.93192 | don’t]        | 0.89157 | oh+boy        | 0.73529 |               |       |
| (h)ere        | 0.93082 | whyn’t        | 0.89157 | d@l           | 0.72603 |               |       |
| stay          | 0.92073 | who           | 0.88527 | goodness      | 0.7234  |               |       |
| later         | 0.91964 | how’s         | 0.875   | s@l           | 0.72    |               |       |
| thank         | 0.91667 | who’s         | 0.85068 | sorry         | 0.70588 |               |       |
| them          | 0.9124  | [:            | 0.85047 | thank+you     | 0.6875  |               |       |
| can’t         | 0.90762 | ?             | 0.84783 | o:h           | 0.68    |               |       |
| never         | 0.9058  | matter        | 0.82963 | nope          | 0.67857 |               |       |
| em            | 0.89922 | what’d        | 0.8125  | hi            | 0.67213 |               |       |
Summary

- Complex models often don’t have analytic solutions
- Approximate inference can be used on many such models
- Monte Carlo Markov chain methods produce samples from (an approximation to) the posterior distribution
- Gibbs sampling is an MCMC procedure that resamples each variable conditioned on the values of the other variables
- If you can sample from the conditional distribution of each hidden variable in a Bayes net, you can use Gibbs sampling to sample from the joint posterior distribution
- We applied Gibbs sampling to Dirichlet-multinomial mixtures to cluster sentences
Outline

Introduction to Bayesian Inference

Mixture models

Sampling with Markov Chains

The Gibbs sampler

Gibbs sampling for Dirichlet-Multinomial mixtures

Topic modeling with Dirichlet multinomial mixtures

Chinese Restaurant Processes
Bayesian inference for Dirichlet-multinomials

- Probability of next event with *uniform Dirichlet prior* with mass $\alpha$ over $m$ outcomes and observed data $\mathbf{Z}_{1:n} = (Z_1, \ldots, Z_n)$

\[
P(Z_{n+1} = k \mid \mathbf{Z}_{1:n}, \alpha) \propto n_k(\mathbf{Z}_{1:n}) + \alpha / m
\]

where $n_k(\mathbf{Z}_{1:n})$ is number of times $k$ appears in $\mathbf{Z}_{1:n}$

- Example: Coin ($m = 2$), $\alpha = 1$, $\mathbf{Z}_{1:2} = (\text{heads}, \text{heads})$
  - $P(Z_3 = \text{heads} \mid \mathbf{Z}_{1:2}, \alpha) \propto 2.5$
  - $P(Z_3 = \text{tails} \mid \mathbf{Z}_{1:2}, \alpha) \propto 0.5$
Dirichlet-multinomials with many outcomes

• Predictive probability:

\[ P(Z_{n+1} = k \mid Z_{1:n}, \alpha) \propto n_k(Z_{1:n}) + \alpha / m \]

• Suppose the number of outcomes \( m \gg n \). Then:

\[
P(Z_{n+1} = k \mid Z_{1:n}, \alpha) \propto \begin{cases} 
n_k(Z_{1:n}) & \text{if } n_k(Z_{1:n}) > 0 \\
\alpha / m & \text{if } n_k(Z_{1:n}) = 0
\end{cases}
\]

• But most outcomes will be unobserved, so:

\[ P(Z_{n+1} \notin Z_{1:n} \mid Z_{1:n}, \alpha) \propto \alpha \]
From Dirichlet-multinomials to Chinese Restaurant Processes

• Suppose number of outcomes is unbounded but we pick the event labels
• If we number event types in order of occurrence ⇒ Chinese Restaurant Process

\[ Z_1 = 1 \]

\[ P(Z_{n+1} = k \mid Z_{1:n}, \alpha) \propto \begin{cases} 
  n_k(Z_{1:n}) & \text{if } k \leq m = \max(Z_{1:n}) \\
  \alpha & \text{if } k = m + 1
\end{cases} \]
Chinese Restaurant Process (0)

- Customer → table mapping $Z = \{ \}
- P(z) = 1

- Next customer chooses a table according to:

$$P(Z_{n+1} = k \mid Z_{1:n}) \propto \begin{cases} n_k(Z_{1:n}) & \text{if } k \leq m = \max(Z_{1:n}) \\ \alpha & \text{if } k = m + 1 \end{cases}$$
Chinese Restaurant Process (1)

- Customer → table mapping $Z = 1$
- $P(z) = \frac{\alpha}{\alpha}$

Next customer chooses a table according to:

$$P(Z_{n+1} = k \mid Z_{1:n}) \propto \begin{cases} n_k(Z_{1:n}) & \text{if } k \leq m = \max(Z_{1:n}) \\ \alpha & \text{if } k = m + 1 \end{cases}$$
Chinese Restaurant Process (2)

- Customer → table mapping $Z = 1, 1$
- $P(z) = \frac{\alpha}{\alpha} \times \frac{1}{1 + \alpha}$

- Next customer chooses a table according to:

$$P(Z_{n+1} = k \mid Z_{1:n}) \propto \begin{cases} 
  n_k(Z_{1:n}) & \text{if } k \leq m = \max(Z_{1:n}) \\
  \alpha & \text{if } k = m + 1
\end{cases}$$
Chinese Restaurant Process (3)

- Customer $\rightarrow$ table mapping $Z = 1, 1, 2$
- $P(z) = \frac{\alpha}{\alpha} \times \frac{1}{(1 + \alpha)} \times \frac{\alpha}{(2 + \alpha)}$

- Next customer chooses a table according to:

$$P(Z_{n+1} = k \mid Z_{1:n}) \propto \begin{cases} n_k(Z_{1:n}) & \text{if } k \leq m = \max(Z_{1:n}) \\ \alpha & \text{if } k = m + 1 \end{cases}$$
Chinese Restaurant Process (4)

- Customer $\rightarrow$ table mapping $Z = 1, 1, 2, 1$
- $P(z) = \alpha/\alpha \times 1/(1 + \alpha) \times \alpha/(2 + \alpha) \times 2/(3 + \alpha)$

- Next customer chooses a table according to:

$$P(Z_{n+1} = k \mid Z_{1:n}) \propto \begin{cases} 
  n_k(Z_{1:n}) & \text{if } k \leq m = \max(Z_{1:n}) \\
  \alpha & \text{if } k = m + 1
\end{cases}$$
Pitman-Yor Process (0)

- Customer $\rightarrow$ table mapping $z = $ 
- $P(z) = 1$

- In CRPs, probability of choosing a table $\propto$ number of customers $\Rightarrow$ strong *rich get richer* effect
- Pitman-Yor processes take mass $a$ from each occupied table and give it to the new table

$$P(Z_{n+1} = k \mid z) \propto \begin{cases} n_k(z) - a & \text{if } k \leq m = \max(z) \\ ma + b & \text{if } k = m + 1 \end{cases}$$
Customer → table mapping $z = 1$

$P(z) = \frac{b}{b}$

In CRPs, probability of choosing a table $\propto$ number of customers $\Rightarrow$ strong *rich get richer* effect

Pitman-Yor processes take mass $a$ from each occupied table and give it to the new table

$$P(Z_{n+1} = k \mid z) \propto \begin{cases} n_k(z) - a & \text{if } k \leq m = \max(z) \\ ma + b & \text{if } k = m + 1 \end{cases}$$
Pitman-Yor Process (2)

- Customer → table mapping $z = 1, 1$
- $P(z) = b/b \times (1 - a)/(1 + b)$

- In CRPs, probability of choosing a table $\propto$ number of customers $\Rightarrow$ strong *rich get richer* effect
- Pitman-Yor processes take mass $a$ from each occupied table and give it to the new table

$$P(Z_{n+1} = k \mid z) \propto \begin{cases} n_k(z) - a & \text{if } k \leq m = \max(z) \\ ma + b & \text{if } k = m + 1 \end{cases}$$
Pitman-Yor Process (3)

- Customer → table mapping $z = 1, 1, 2$
- $P(z) = b/b \times (1-a)/(1+b) \times (a+b)/(2+b)$

- In CRPs, probability of choosing a table $\propto$ number of customers $\Rightarrow$ strong *rich get richer* effect
- Pitman-Yor processes take mass $a$ from each occupied table and give it to the new table

$$P(Z_{n+1} = k \mid z) \propto \begin{cases} n_k(z) - a & \text{if } k \leq m = \max(z) \\ ma + b & \text{if } k = m + 1 \end{cases}$$
Pitman-Yor Process (4)

- Customer $\rightarrow$ table mapping $z = 1, 1, 2, 1$
- $P(z) = \frac{b}{b} \times \frac{1-a}{1+b} \times \frac{a+b}{2+b} \times \frac{2-a}{3+b}$
- In CRPs, probability of choosing a table $\propto$ number of customers $\Rightarrow$ strong *rich get richer* effect
- Pitman-Yor processes take mass $a$ from each occupied table and give it to the new table

$$P(Z_{n+1} = k \mid z) \propto \begin{cases} n_k(z) - a & \text{if } k \leq m = \max(z) \\ ma + b & \text{if } k = m + 1 \end{cases}$$
Labeled Chinese Restaurant Process (0)

- Table $\rightarrow$ label mapping $Y = $
- Customer $\rightarrow$ table mapping $Z = $
- Output sequence $X = $
- $P(X) = 1$

- **Base distribution** $P_0(Y)$ generates a label $Y_k$ for each table $k$
- All customers sitting at table $k$ (i.e., $Z_i = k$) share label $Y_k$
- Customer $i$ sitting at table $Z_i$ has label $X_i = Y_{Z_i}$
Labeled Chinese Restaurant Process (1)

- Table $\rightarrow$ label mapping $Y = \text{fish}$
- Customer $\rightarrow$ table mapping $Z = 1$
- Output sequence $X = \text{fish}$
- $P(X) = \alpha / \alpha \times P_0(\text{fish})$

- **Base distribution** $P_0(Y)$ generates a *label* $Y_k$ for each table $k$
- All customers sitting at table $k$ (i.e., $Z_i = k$) share label $Y_k$
- Customer $i$ sitting at table $Z_i$ has label $X_i = Y_{Z_i}$
Labeled Chinese Restaurant Process (2)

- Table $\rightarrow$ label mapping $Y = \text{fish}$
- Customer $\rightarrow$ table mapping $Z = 1, 1$
- Output sequence $X = \text{fish, fish}$
- $P(X) = P_0(\text{fish}) \times 1/(1 + \alpha)$

- **Base distribution** $P_0(Y)$ generates a *label* $Y_k$ for each table $k$
- All customers sitting at table $k$ (i.e., $Z_i = k$) share label $Y_k$
- Customer $i$ sitting at table $Z_i$ has label $X_i = Y_{Z_i}$
Labeled Chinese Restaurant Process (3)

- Table $\rightarrow$ label mapping $Y = \text{fish, apple}$
- Customer $\rightarrow$ table mapping $Z = 1, 1, 2$
- Output sequence $X = \text{fish, fish, apple}$
- $P(X) = P_0(\text{fish}) \times \frac{1}{1 + \alpha} \times \frac{\alpha}{2 + \alpha} P_0(\text{apple})$

- **Base distribution** $P_0(Y)$ generates a *label* $Y_k$ for each table $k$
- All customers sitting at table $k$ (i.e., $Z_i = k$) share label $Y_k$
- Customer $i$ sitting at table $Z_i$ has label $X_i = Y_{Z_i}$
Labeled Chinese Restaurant Process (4)

- Table → label mapping $Y = \text{fish, apple}$
- Customer → table mapping $Z = 1, 1, 2$
- Output sequence $X = \text{fish, fish, apple, fish}$
- $P(X) = P_0(\text{fish}) \times \frac{1}{1 + \alpha} \times \frac{\alpha}{2 + \alpha} P_0(\text{apple}) \times \frac{2}{3 + \alpha}$

- **Base distribution** $P_0(Y)$ generates a label $Y_k$ for each table $k$
- All customers sitting at table $k$ (i.e., $Z_i = k$) share label $Y_k$
- Customer $i$ sitting at table $Z_i$ has label $X_i = Y_{Z_i}$
From Chinese restaurants to Dirichlet processes

- Labeled Chinese restaurant processes take a distribution \( P_0 \) and return a stream of samples from a different distribution with the same support.
- The Chinese restaurant process is a sequential process, generating the next item conditioned on the previous ones.
- We can get a different distribution each time we run a CRP (allocation of customers to tables and labeling of tables are randomized).
- Abstracting away from the sequential generation of the CRP, we can view it as a mapping from a base distribution \( P_0 \) to a distribution over distributions \( DP(\alpha, P_0) \).
- \( DP(\alpha, P_0) \) is called a Dirichlet process with concentration parameter \( \alpha \) and base distribution \( P_0 \).
- Distributions in \( DP(\alpha, P_0) \) are discrete (w.p. 1) even if the base distribution \( P_0 \) is continuous.
Gibbs sampling with Chinese restaurants

- Idea: resample $z_i$ as if $z_{-i}$ were “real” data
- The CRP is *exchangelable*: all ways of generating an assignment of customers to labeled tables have the same probability
- This means $P(z_i|z_{-i})$ is the same as if $z_i$ were generated after $s_{-i}$
  - Exchangability means “treat every customer as if they were your last”
- Tables are generated and garbage-collected during sampling
- The probability of generating a new table includes the probability of generating its label
- When retracting $z_i$ reduces the number of customers at a table to 0, garbage-collect the table
- CRPs not only estimate model parameters, they also *estimate the number of components (tables)*
A DP clustering model

- Idea: replace multinomials with Chinese restaurants
- \( P(z) \) is a distribution over integers (clusters), generated by a CRP
- For each cluster \( z \), run separate Chinese restaurants for \( P(x|c) \)
- \( P(x|c) \) are distributions over words, so they need generator distributions
  - generators could be uniform over the named entities/contexts in training data, or
  - \((n\text{-gram})\) language models generating possible named entities/contexts (unbounded vocabulary)
- In a \textit{hierarchical Dirichlet process}, these generators could themselves be Dirichlet processes that possibly share a common vocabulary
Summary: Chinese Restaurant Processes

- Chinese Restaurant Processes (CRPs) generalize Dirichlet-Multinomials to an unbounded number of outcomes
  - concentration parameter $\alpha$ controls how likely a new outcome is
  - CRPs exhibit a rich get richer power-law behaviour
- Labeled CRPs use a base distribution to label each table
  - base distribution can have infinite support
  - concentrates mass on a countable subset
  - power-law behaviour $\Rightarrow$ Zipfian distributions