

# Chinese Restaurant Processes

Mark Johnson

CG168 notes

# Non-parametric models

- *Parametric* models are characterized by a *fixed-sized vector* of real-valued parameters
  - ▶ topic model characterized by number of topics  $\ell$ , topic probability vector  $\phi$  and topic  $\rightarrow$  word probability vectors  $\theta$
- *Non-parametric* models are models that aren't parametric models
  - ▶ doesn't mean they don't have parameters
  - ▶ but the number of parameters can *grow unboundedly* depending on the data
  - ▶ *let the data choose* the appropriate complexity of the model

# Dirichlet processes and Chinese restaurant processes

- *Dirichlet processes* (DP) extend Dirichlet-multinomials to an *unbounded number of outcomes*
  - ▶ Example: generalize topic model to an unbounded number of words and/or unbounded number of topics
  - ▶ The math is complicated
- *Chinese restaurant processes* are the *conditional distributions*  $P(X_{n+1} | \mathbf{X}_{1:n})$  when  $P(\mathbf{X})$  is generated by a DP
  - ▶ They are what you need to build a Gibbs sampler for DPs
  - ▶ They are only a little more complicated than the sampler for the collapsed Dirichlet-multinomial
- We'll start by looking at models in which each outcome has equal prior probability, but CRPs generalize to situations in which different outcomes have different priors

# Review of Dirichlet-multinomials

$$\begin{aligned}P(\mathbf{X}|\boldsymbol{\alpha}) &= \int P(\mathbf{X}|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\boldsymbol{\alpha}) d\boldsymbol{\theta} \\&= \int \left( \prod_{j=1}^m \theta_j^{N_j(\mathbf{X})} \right) \left( \frac{1}{C(\boldsymbol{\alpha})} \prod_{j=1}^m \theta_j^{\alpha_j-1} \right) d\boldsymbol{\theta} \\&= \frac{1}{C(\boldsymbol{\alpha})} \int \prod_{j=1}^m \theta_j^{N_j(\mathbf{X})+\alpha_j-1} d\boldsymbol{\theta} \\&= \frac{C(\mathbf{N}(\mathbf{X}) + \boldsymbol{\alpha})}{C(\boldsymbol{\alpha})}, \text{ where:}\end{aligned}$$

$$C(\boldsymbol{\beta}) = \int \prod_{j=1}^m \theta_j^{\beta_j-1} d\boldsymbol{\theta} = \frac{\prod_{j=1}^m \Gamma(\beta_j)}{\Gamma(\boldsymbol{\beta}_{\bullet})}, \text{ and } \boldsymbol{\beta}_{\bullet} = \sum_{j=1}^m \beta_j$$

$$P(X_{n+1} = k | \mathbf{X}_{1:n}, \boldsymbol{\alpha}) = \frac{N_k(\mathbf{X}_{1:n}) + \alpha_k}{n + \alpha_{\bullet}}$$

# Exchangability of mixture components

- In D-M mixture models, we usually have uniform priors over mixture components,
  - ⇒ the mixture components are *exchangable*, i.e., all permutations of the mixture components are equally likely
  - ⇒ we can choose the order in which the mixture components get used
- The Chinese restaurant process assigns mixture components to data items  $X_i$  in *in a fixed order*
  - ▶  $X_{n+1}$  can only be assigned to mixture component  $k$  if mixture component  $k - 1$  has already been occupied in  $X_1, \dots, X_n$ , i.e., if  $Y_i = k$  for some  $i \in 1, \dots, n$ .

# Towards the Chinese restaurant process

- D-M sampling distribution with a *uniform prior*  $\alpha_k = \alpha_{\bullet}/m$

$$P(X_{n+1} = k | \mathbf{X}_{1:n}, \alpha_{\bullet}) = \frac{N_k(\mathbf{X}_{1:n}) + \alpha_{\bullet}/m}{n + \alpha_{\bullet}}$$

- Hold  $\alpha_{\bullet}$  and  $n$  constant and let *number of outcomes*  $m \rightarrow \infty$ 
  - ▶ If  $m \gg n$ , then almost all outcomes *have no samples*
- Suppose  $N_k(\mathbf{X}_{1:n}) > 0$ , i.e., we saw outcome  $k$  in  $\mathbf{X}_{1:n}$ . Then:

$$P(X_{n+1} = k | \mathbf{X}_{1:n}, \alpha_{\bullet}) = \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha_{\bullet}}$$

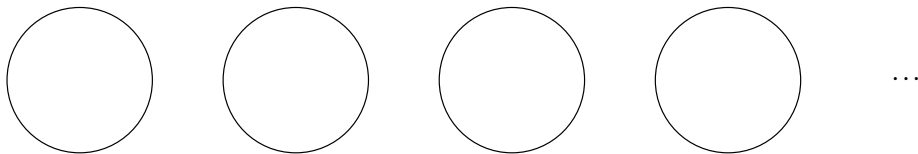
- The probability of picking *any previously unseen outcome* (i.e.,  $N_k(\mathbf{X}_{1:n}) = 0$ ) is:

$$P(X_{n+1} \notin \mathbf{X}_{1:n} | \mathbf{X}_{1:n}, \alpha_{\bullet}) = \frac{\alpha_{\bullet}}{n + \alpha_{\bullet}}$$

# The Chinese restaurant metaphor

- A *Chinese restaurant* has infinitely many *tables* (outcomes)
- Each table can seat infinitely many *customers* (samples)
- Tables are ordered and occupied *in order*. A customer can sit at an occupied table or *the next unoccupied table*.
- When customer  $X_{n+1}$  comes into the restaurant:
  - ▶ sits at an *already occupied table*  $k$  with probability  $N_k / (n + \alpha)$ , where  $N_k$  is number of customers at table  $k$
  - ▶ sits at the *next unoccupied table* with probability  $\alpha / (n + \alpha)$

# The Chinese restaurant process (0)

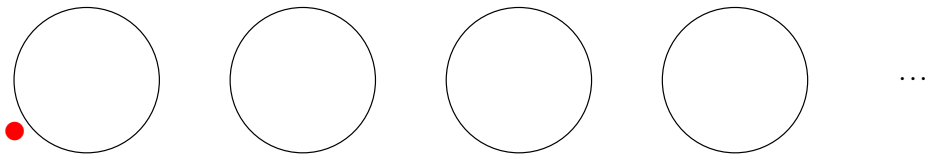


1

- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

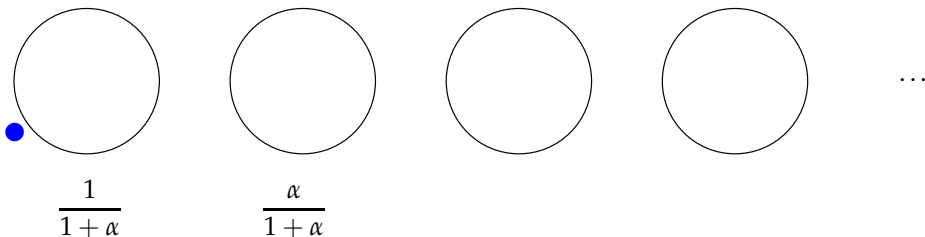
# The Chinese restaurant process (1a)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

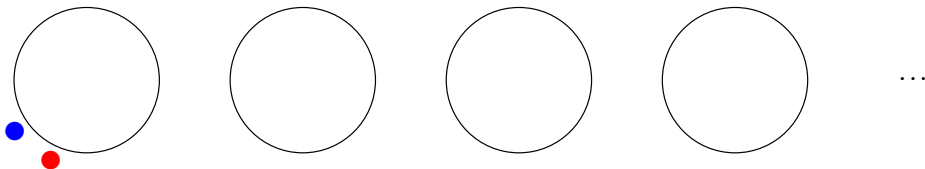
# The Chinese restaurant process (1b)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

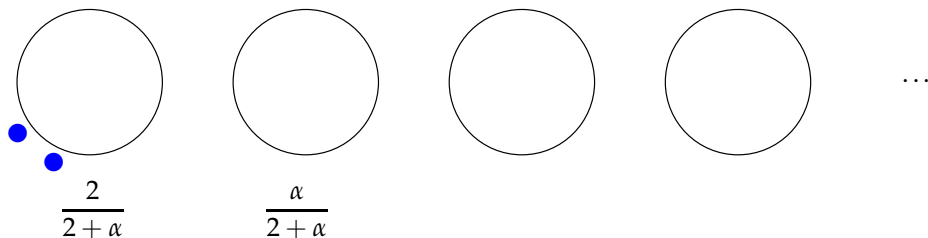
## The Chinese restaurant process (2a)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

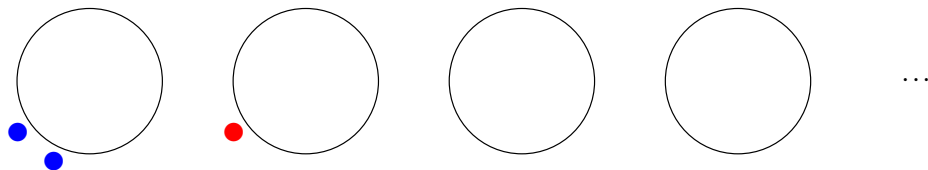
## The Chinese restaurant process (2b)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

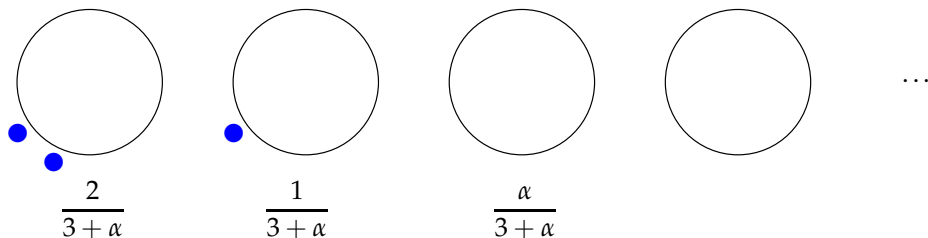
## The Chinese restaurant process (3a)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

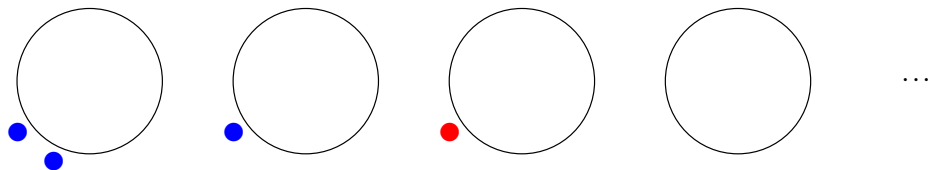
# The Chinese restaurant process (3b)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

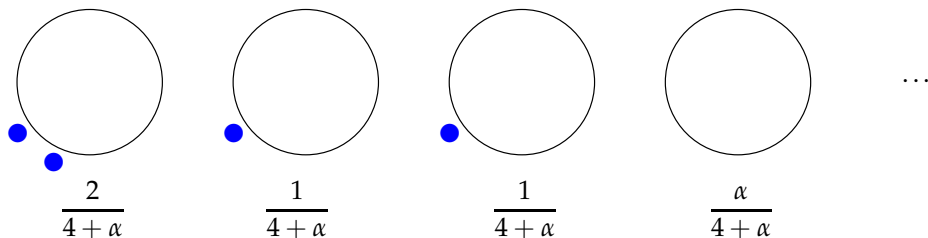
## The Chinese restaurant process (4a)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

## The Chinese restaurant process (4b)



- Each  $X_i$  ranges over outcomes, i.e., positive integers or *tables*
- If generating  $\mathbf{X}_{1:n} = (X_1, \dots, X_n)$  has occupied tables  $1, \dots, m$ , then:

$$P(X_{n+1} = k \mid \mathbf{X}_{1:n}, \alpha) = \begin{cases} \frac{N_k(\mathbf{X}_{1:n})}{n + \alpha} & \text{if } k \in 1, \dots, m \\ \frac{\alpha}{n + \alpha} & \text{if } k = m + 1 \end{cases}$$

# The rich get richer

- CRP rule: next customer sits at a table with prob. proportional to number of customers already sitting at it (and sits at new table with prob. proportional to  $\alpha$ )
- ⇒ customers tend to sit at most popular tables
- ⇒ most popular tables attract the most new customers, and become even more popular
- CRPs exhibit “power law” behavior, where a few tables attract the bulk of the customers
- The *concentration parameter*  $\alpha$  determines how likely a customer is to sit at a fresh table

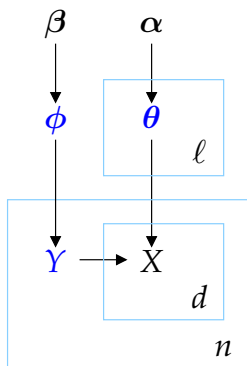
# Labeled Chinese Restaurant processes

- CRPs define distributions over tables (positive integers)
- They can be used to define probability distributions over arbitrary objects as follows:
  - ▶ Label each table  $j$  with an object  $Y_j$  drawn from a *base distribution*  $G_0$  over domain  $\mathcal{Y}$   
( $Y_j$  is the dish on table  $j$ , shared by all customers at table  $j$ )
  - ▶ The labeled CRP generates samples from  $\mathcal{Y}$  as follows.  
The  $i$ th sample is  $Y_{X_i}$ , i.e., the label on table  $X_i$ , where  $X_i$  is the table that customer  $i$  sits at.
- This distribution is called a *Dirichlet process*  $DP(\alpha, G_0)$  with *base distribution*  $G_0$  and *concentration parameter*  $\alpha$
- Labels can be virtually anything at all
  - ▶ Multinomials from a Dirichlet prior (e.g., topic models)
  - ▶ Grammar rules (e.g., “infinite” grammars)
  - ▶ Samples from another Dirichlet process (Teh et al (2006) *Hierarchical Dirichlet Processes*)

# Topic modeling using Chinese restaurants

- Customers correspond to documents  $\mathbf{X}_i$  in topic model
- Tables correspond to hidden classes  $k$  in topic model
- Don't explicitly represent hidden classes  $k$  with no documents (i.e.,  $N_k(\mathbf{Y}) = 0$ )
  - ▶ generate a new hidden class  $k$  (table) when a document is assigned to it
  - ▶ garbage-collect a hidden class when all documents have been removed from it
- Uncollapsed sampler samples assignment of documents to hidden classes  $\mathbf{Y}$  and probabilities of words  $\theta_k$  in each class  $k$ 
  - ▶ probability of hidden classes integrated out (collapsed) since not all classes are explicitly represented
- Collapsed sampler samples assignment of documents to hidden classes  $\mathbf{Y}$ , but integrates out word probabilities
- Exchangability still holds  $\Rightarrow$  treat each customer as if they were your last ...

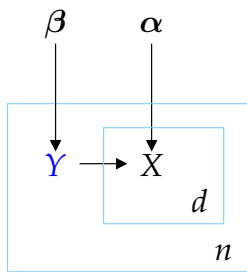
# Review: Fixed- $\ell$ uncollapsed sampler



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

$$P(Y_i = k | \phi, \theta, \mathbf{X}_i) \propto \phi_k \prod_{j=1}^m \theta_{k,j}^{N_j(\mathbf{X}_i)}$$

# Review: Fixed- $\ell$ collapsed sampler



- Initialize by *randomly* assign each document  $\mathbf{X}_i$  to a topic  $Y_i$
- Repeat the following:
  - ▶ For each document  $i$  in  $1, \dots, n$  (in random order):
    - Replace  $Y_i$  with a random sample from  $P(Y_i | \mathbf{Y}_{-i}, \alpha, \beta)$

$$P(Y_i = k | \mathbf{Y}_{-i}, \mathbf{X}, \alpha, \beta) \propto \frac{N_k(\mathbf{Y}_{-i}) + \beta_k}{n - 1 + \beta_{\bullet}} \frac{C(\alpha + \mathbf{N}_{Y=k}(\mathbf{X}_{-i}) + \mathbf{N}(\mathbf{X}_i))}{C(\alpha + \mathbf{N}_{Y=k}(\mathbf{X}_{-i}))}$$

where  $\mathbf{N}_{Y=k}(\mathbf{X}_{-i}) = \sum_{i' \neq i, Y_{i'}=k} \mathbf{N}(\mathbf{X}_{i'})$

# CRP uncollapsed sampler

- Each table  $k$  (hidden class) is *labeled with class*  $\rightarrow$  *word multinomial*  $\theta_k$ , where  $\theta_{kj}$  = prob. of word  $j$  in class  $k$
- Empty classes not represented  $\Rightarrow$  integrate out (collapse) multinomial parameters associated with empty classes

$$P(Y_i = k | \mathbf{X}, \mathbf{Y}_{-i}, \alpha, \beta, \theta) \propto \begin{cases} \frac{N_k(\mathbf{Y}_{-i})}{n-1+\beta} \prod_{j=1}^m \theta_{kj}^{N_j(\mathbf{X}_i)} & \text{if } k \leq \ell \\ \frac{\beta}{n-1+\beta} \frac{C(\mathbf{N}(\mathbf{X}_i) + \alpha)}{C(\alpha)} & \text{if } k = \ell + 1 \end{cases}$$

- $\alpha$  = Dirichlet prior for  $\theta_k$  (word multinomials),  $\beta$  = CRP parameter,  $\ell$  = number of currently occupied classes (tables)
- Sample  $\theta_k$  from  $\alpha$  and the  $\mathbf{X}_i$  associated with class  $k$ , as in fixed- $\ell$  uncollapsed model  
(This is Algorithm 2 of Neal (2000))

# CRP collapsed sampler

- Integrate out class  $\rightarrow$  word multinomial parameters  $\theta_k$
- $\Rightarrow$  Existing class  $k$  generates  $\mathbf{X}_i$  with probability

$$\frac{C(\mathbf{N}(\mathbf{X}_i) + \mathbf{N}_{Y=k}(\mathbf{X}_{-i}) + \alpha)}{C(\mathbf{N}_{Y=k}(\mathbf{X}_{-i}) + \alpha)}$$

where  $\mathbf{N}_{Y=k}(\mathbf{X}_{-i}) = \sum_{i' \neq i, Y_{i'}=k} \mathbf{N}(\mathbf{X}_{i'})$ , so:

$$P(Y_i = k | \mathbf{X}, \mathbf{Y}_{-i}, \alpha, \beta) \propto \begin{cases} \frac{N_k(\mathbf{Y}_{-i})}{n-1+\beta} \frac{C(\mathbf{N}(\mathbf{X}_i) + \mathbf{N}_{Y=k}(\mathbf{X}_{-i}) + \alpha)}{C(\mathbf{N}_{Y=k}(\mathbf{X}_{-i}) + \alpha)} & \text{if } k \leq \ell \\ \frac{\beta}{n-1+\beta} \frac{C(\mathbf{N}(\mathbf{X}_i) + \alpha)}{C(\alpha)} & \text{if } k = \ell + 1 \end{cases}$$

(This is Algorithm 3 of Neal (2000))

# Summary

- It's straight-forward to extend Dirichlets to an unbounded number of outcomes
  - ▶ the resulting distribution is called a *Dirichlet process*
  - ▶ the conditional (sampling) distribution is called a *Chinese Restaurant process*
- These can replace finite dimensional Dirichlet-Multinomials in our models
- Gibbs sampling can be performed on the resulting models
  - ▶ Have to recycle (garbage-collect) vacated tables
  - ▶ Not clear how best to initialize (randomly assign documents to an infinite number of tables? give each document a unique table? sample initial assignment?)