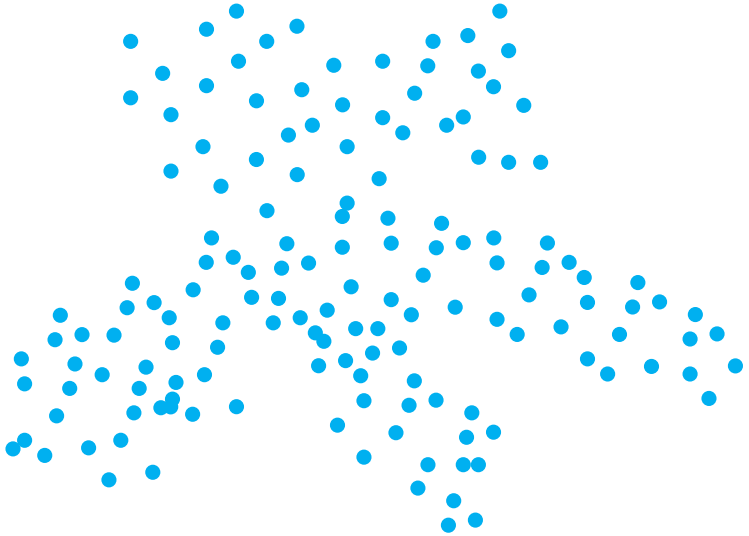
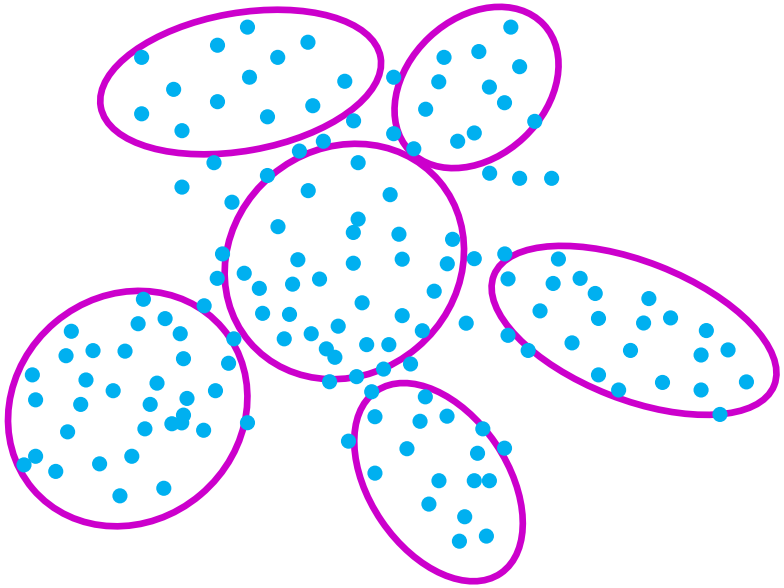
AI – Machine Learning

Artificial Intelligence
Research Group

Motivation

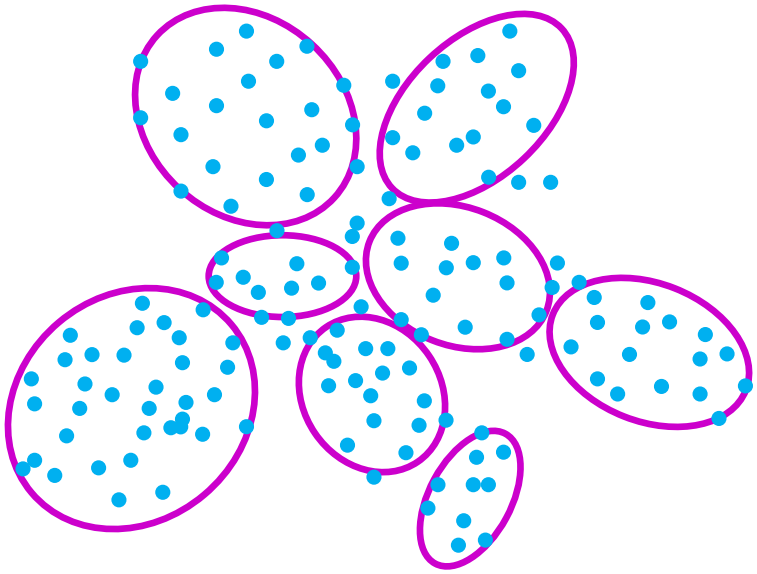


Motivation



$k = 6$

Motivation



$k = 8$

Dirichlet Distribution

The Dirichlet distribution of order $k \geq 2$ with parameters $\alpha_1, \alpha_2, \cdots, \alpha_k > 0$ has a probability density function with respect to Lebesgue measure on the Euclidean space $\mathbb{R}^{k-1}$ given by

$$f(x_1, x_2, \cdots, x_k; \alpha_1, \alpha_2, \cdots, \alpha_k) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

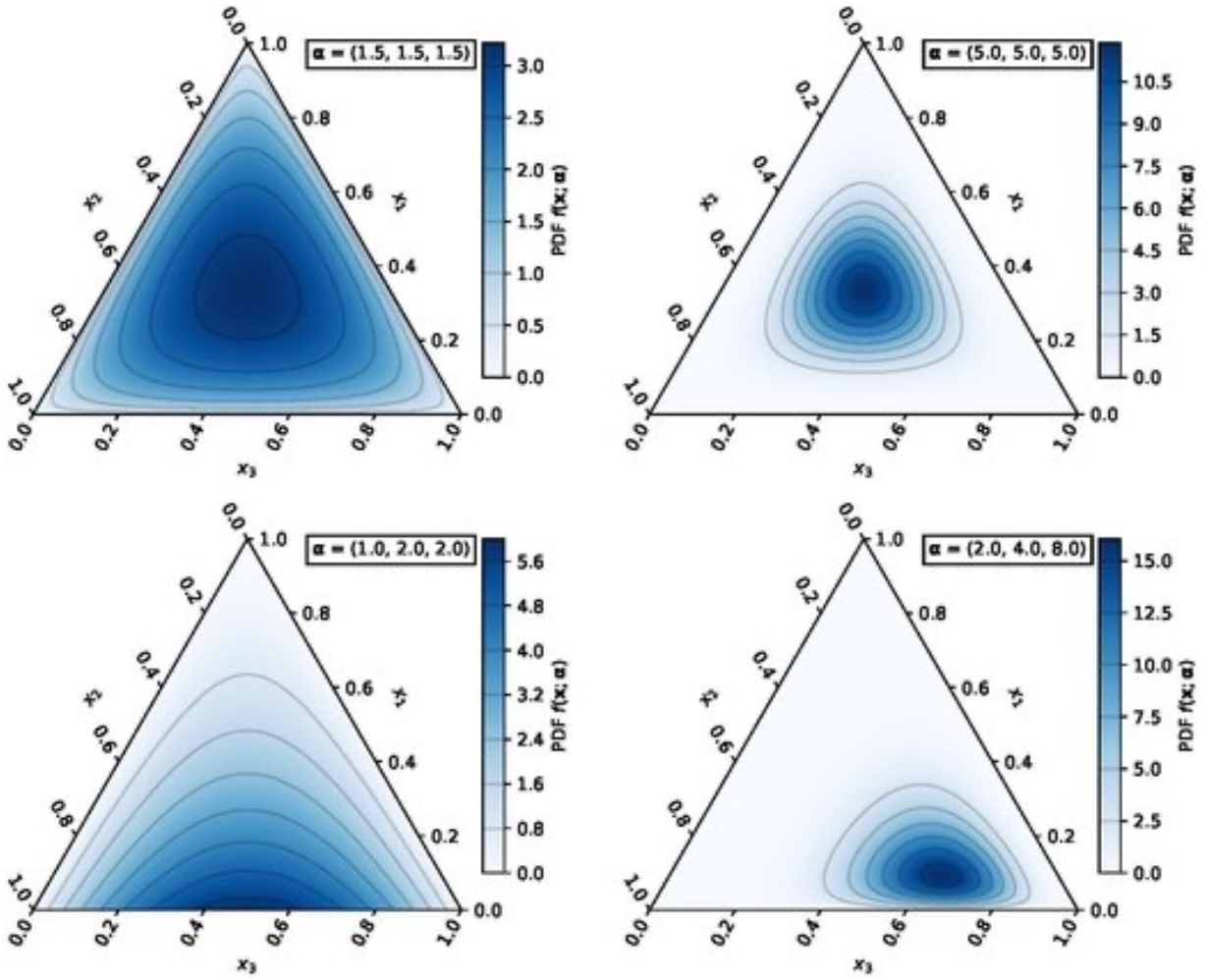
where $\sum_{i=1}^{k} x_i = 1$ and $x_i \geq 0$.

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)} \qquad \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_k)$$

Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_k) \sim \mathrm{Dir}(\alpha_1, \alpha_2, \cdots, \alpha_k)$ and $\alpha_0 = \sum_{i=1}^{k} \alpha_i$.

$$\mathrm{E}[x_i] = \frac{\alpha_i}{\alpha_0} \qquad \mathrm{Var}[x_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$$

Dirichlet Distribution



Dirichlet Process

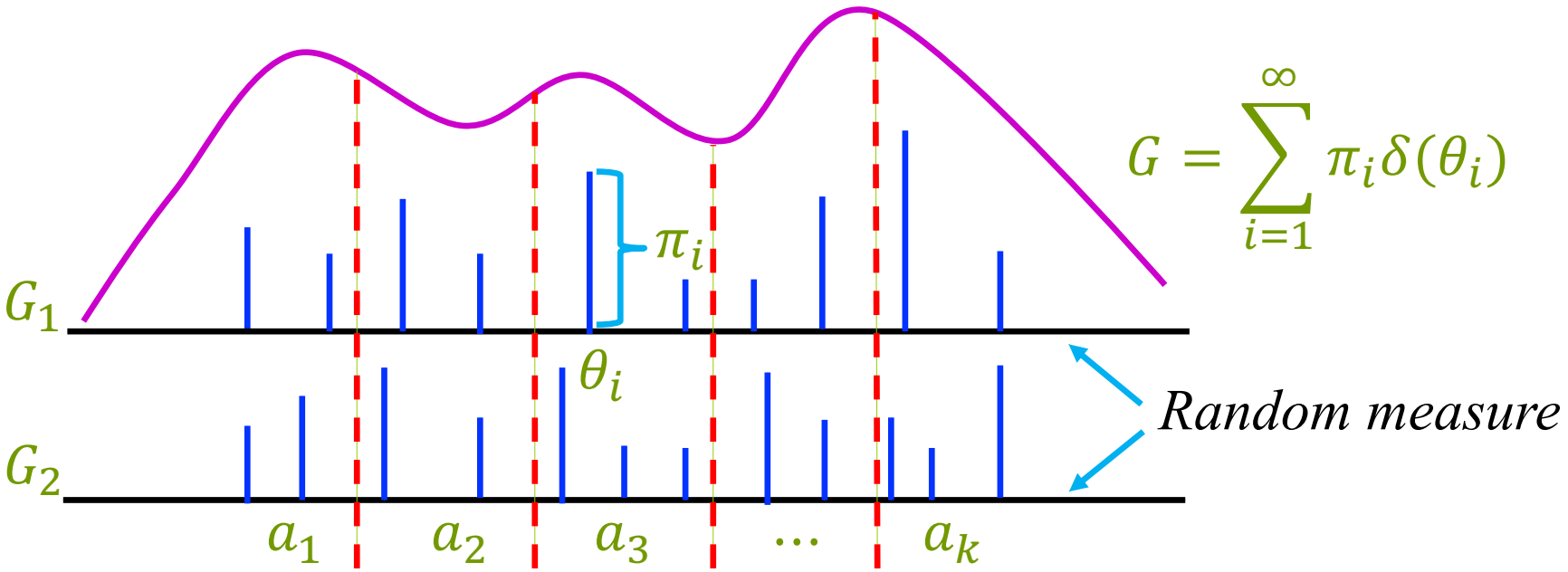
$\theta_i \sim G$

$G \sim \mathrm{DP}(\alpha, H)$

$E[k] \propto \log(N)$

$G = H$ if $\alpha \to \infty$

$G =$ *Very discrete distribution* if $\alpha \to 0$



$\forall a_1, a_2, \cdots, a_k$ (For any partition)

$G(a_1), G(a_2), \cdots, G(a_k) \sim \mathrm{Dir}(\alpha H(a_1), \alpha H(a_2), \cdots, \alpha H(a_k))$

Dirichlet Process

$G(a_1), G(a_2), \cdots, G(a_k) \sim \mathrm{Dir}(\alpha H(a_1), \alpha H(a_2), \cdots, \alpha H(a_k))$

Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_k) \sim \mathrm{Dir}(\alpha_1, \alpha_2, \cdots, \alpha_k)$ and $\alpha_0 = \sum_{i=1}^{k} \alpha_i$.

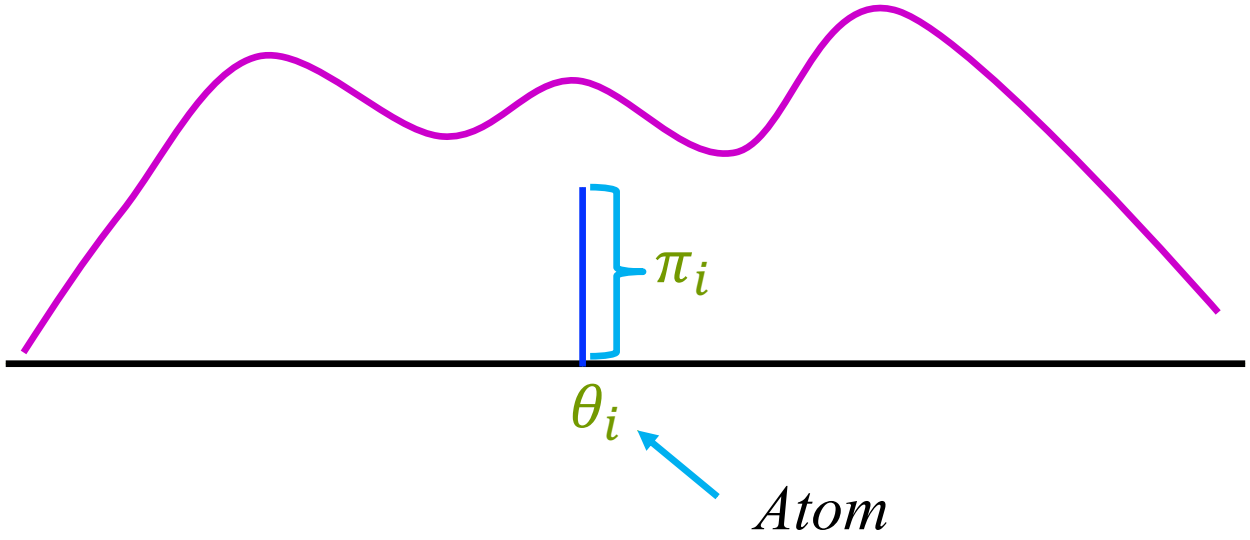
$$\mathrm{E}[x_i] = \frac{\alpha_i}{\alpha_0} \qquad \mathrm{Var}[x_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$$

$$\mathrm{E}[\mathrm{G}(a_i)] = \frac{\alpha H(a_i)}{\sum_{i=1}^{k} \alpha H(a_i)} = \frac{\alpha H(a_i)}{\alpha \underbrace{\sum_{i=1}^{k} H(a_i)}_{=1}} = H(a_i)$$

$$\mathrm{Var}[\mathrm{G}(a_i)] = \frac{\alpha H(a_i)(\alpha - \alpha H(a_i))}{\alpha^2(\alpha+1)} = \frac{H(a_i)(1 - H(a_i))}{\alpha+1}$$

- $G = H$ if $\alpha \to \infty$ imply $\mathrm{Var}[\mathrm{G}(a_i)] \to 0$
- $G =$ *Bernoulli Distribution* if $\alpha \to 0$ imply $\mathrm{Var}[\mathrm{G}(a_i)] = H(a_i)(1 - H(a_i))$

Stick-break Construction



$$\theta_1 \sim H \qquad \theta_2 \sim H \qquad \mathrm{E}[\beta_i] = \frac{1}{1+\alpha}$$

$$\beta_1 \sim \mathrm{Beta}(1,\alpha) \qquad \beta_2 \sim \mathrm{Beta}(1,\alpha)$$

$$\pi_1 = \beta_1 \qquad \pi_2 = (1-\pi_1)\beta_2$$

$G = \textit{Continuous.}$ if $\alpha \to \infty$

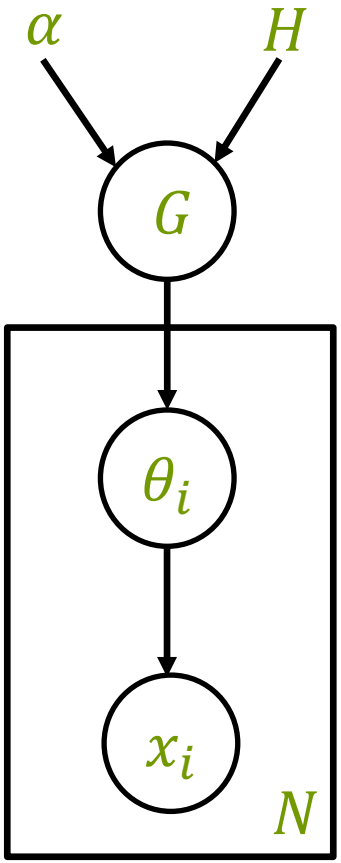
$G = \textit{Discrete.}$ if $\alpha \to 0$

Graphical Model

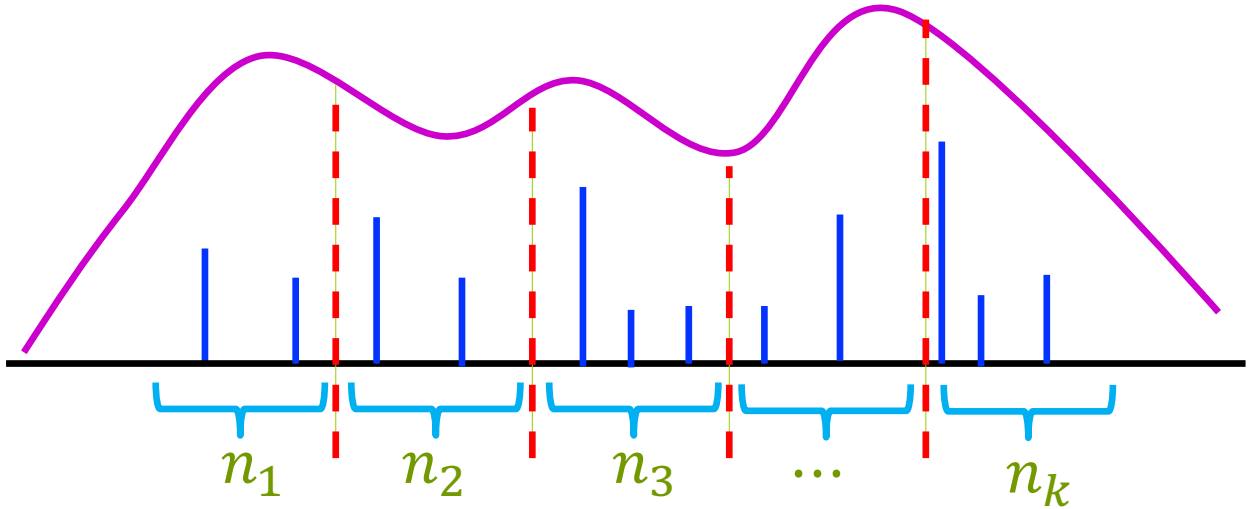
$$G \sim \mathrm{DP}(\alpha, H)$$

$$\theta_1, \theta_2, \cdots, \theta_k \sim G$$

$$x_i \sim F(\theta_i)$$



Posterior Distribution



$$p(G(a_1), G(a_2), \cdots, G(a_k) | n_1, n_2, \cdots, n_k) \propto$$

$$\text{Mult}(n_1, n_2, \cdots, n_k | G(a_1), G(a_2), \cdots, G(a_k)) \, \text{Dir}(\alpha H(a_1), \alpha H(a_2), \cdots, \alpha H(a_k))$$

$$\text{Dir}(\alpha H(a_1) + n_1, \alpha H(a_2) + n_2, \cdots, \alpha H(a_k) + n_k) \sim \text{DP}(\alpha + n, \frac{\alpha H + \sum_{i=1}^{n} \delta(\theta_i)}{\alpha + n})$$

$$\underbrace{\frac{\alpha}{\alpha + n} H}_{\textit{Continuous}} + \underbrace{\frac{\sum_{i=1}^{n} \delta(\theta_i)}{\alpha + n}}_{\textit{Discrete}}$$

Spike and Slab

Predictive Distribution

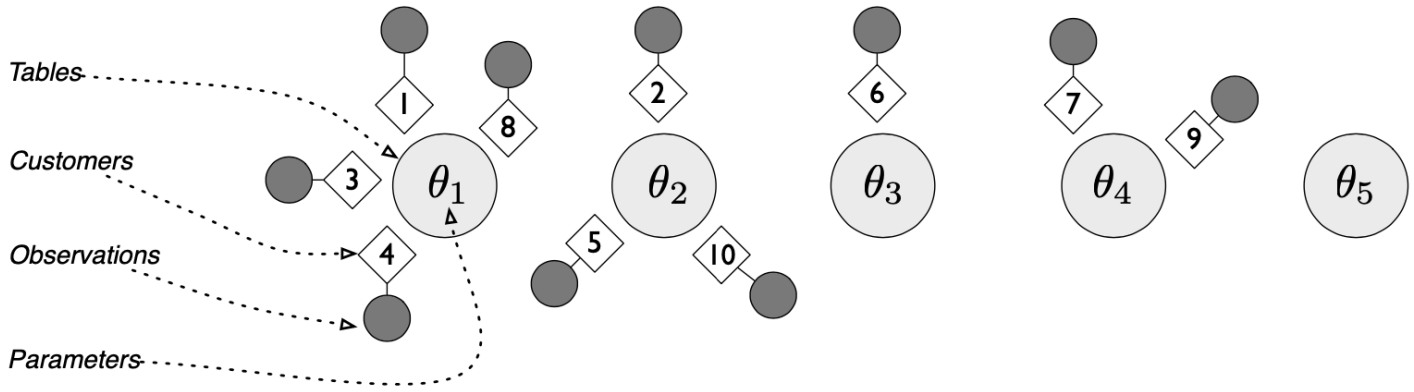
$$p(\theta_i|\overline{\theta_{-i}}) = \int_G p(\theta_i|G)p(G|\overline{\theta_{-i}})dG$$

x_1	x_2	x_3	$\cdots$	x_k
$\theta_1 = 6.0$	$\theta_2 = 4.8$	$\theta_3 = 6.0$	$\ldots$	$\theta_k = 4.8$
$z_1 = 1$	$z_2 = 2$	$z_3 = 1$	$\ldots$	$z_k = 2$

$$p(z_i = m|\overline{z_{-i}}) = \frac{p(z_i = m, \overline{z_{-i}})}{p(\overline{z_{-i}})}$$

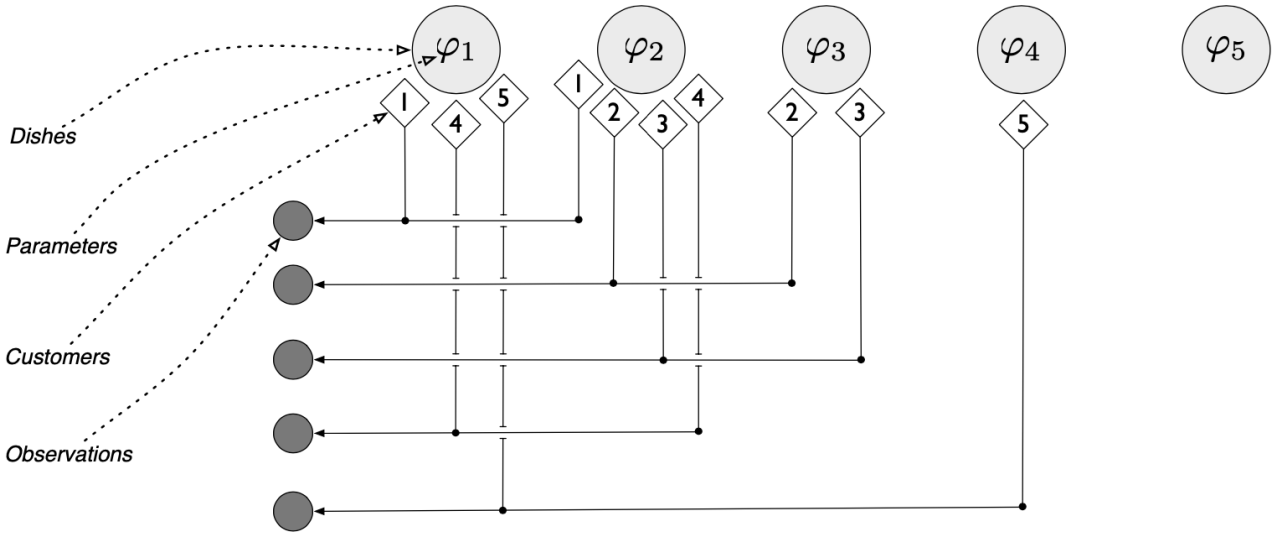
$$\begin{cases} \dfrac{n_{m,-i}}{n + \alpha - 1} & \textit{Existing} \\ \dfrac{\alpha}{n + \alpha - 1} & \textit{New} \end{cases}$$

Chinese Restaurant Process



The ***Chinese restaurant process***. The generative process of the CRP, where numbered diamonds represent customers, attached to their corresponding observations (shaded circles). The large circles represent tables (clusters) in the CRP and their associated parameters (θ). Note that technically the parameter values $\{\theta\}$ are not part of the CRP *per se*, but rather belong to the full mixture model.

Indian Buffet Process



The ***Indian buffet process***. The generative process of the IBP, where numbered diamonds represent customers, attached to their corresponding observations (shaded circles). Large circles represent dishes (factors) in the IBP, along with their associated parameters (φ). Each customer selects several dishes, and each customer's observation (in the latent factor model) is a linear combination of the selected dish's parameters. Note that technically the parameter values $\{\varphi\}$ are not part of the IBP *per se*, but rather belong to the full latent factor model.

Predictive Distribution

$$G \sim \mathrm{DP}(\alpha, H)$$

$$\theta_1, \theta_2, \cdots, \theta_k \sim G$$

$$x_i \sim F(\theta_i)$$

$$\theta_i | \theta_1, \cdots, \theta_{i-1} \sim \frac{1}{i + \alpha - 1} \sum_{j}^{i-1} \delta(\theta_j) + \frac{\alpha}{i + \alpha - 1} H$$

Zip' Law

Word	Freq. (f)	Rank (r)	$f \cdot r$	Word	Freq. (f)	Rank (r)	$f \cdot r$
the	3332	1	3332	turned	51	200	10200
and	2972	2	5944	you'll	30	300	9000
a	1775	3	5235	name	21	400	8400
he	877	10	8770	comes	16	500	8000
but	410	20	8400	group	13	600	7800
be	294	30	8820	lead	11	700	7700
there	222	40	8880	friends	10	800	8000
one	172	50	8600	begin	9	900	8100
about	158	60	9480	family	8	1000	8000
more	138	70	9660	brushed	4	2000	8000
never	124	80	9920	sins	2	3000	6000
Oh	116	90	10440	Could	2	4000	8000
two	104	100	10400	Applausive	1	8000	8000

- $f \propto \frac{1}{r}$ or

  $f \cdot r = k$
- $m \propto \sqrt{f}$

The relationship among the frequency of a word f, its position in the list, know as its rank r, and the number of meanings m of the word.

There is ***a constant ratio*** by which words of length n are more frequent than word of length $n + 1$.

What makes ***frequency-based approaches*** to language ***hard*** is that almost all words are raw.

Pitman-Yor Process

$$G \sim \mathrm{PY}(d, \theta, H) \qquad E[k] \propto \theta N^{d}$$

Where the three parameters are: a *discount parameter* $0 \leq d < 1$, a *strength parameter* $\theta > -d$ and a *base distribution* H.

When $d = 0$ the Pitman-Yor process reduces to a Dirichlet distribution with parameters $\mathrm{DP}(\theta, H)$.

The discount parameter gives the Pitman-Yor process more flexibility over tail behavior than the Dirichlet process, which has ***exponential tails***. This makes Pitman-Yor process useful for modeling data with power-law tails (e.g., word frequencies in natural language).

Hierarchical Pitman-Yor LMs

Given a context $\mathbf{u}$ consisting of a sequence of up to $n-1$ words, let $G_{\mathbf{u}}(w)$ be the distribution over the current word w.

We use a Pitman-Yor process as the prior for $G_{\mathbf{u}}(w)$, in particular,

$$G_{\mathbf{u}}(w) \sim \mathrm{PY}(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}(w))$$

where $\pi_{(\mathbf{u})}$ is the suffix of consisting of $\mathbf{u}$ all but the first word. The base distribution is $G_{\pi(\mathbf{u})}(w)$, the distribution over the current word given all but the earliest word in the context. We believe that without observing any data the earliest word is the least important in determining the distribution over the current word.

$$G_{\mathbf{0}}(w) \sim \mathrm{PY}(d_0, \theta_0, G_0)$$

where $G_{\mathbf{0}}$ is the global base distribution, which is assumed to be uniform over the vocabulary.

Chinese Restaurant Representation

$G_{\mathbf{u}}$: Restaurant

x: Customer

k: Table

w: Dish

Routine to draw a new word given context **u** using the Chinese restaurant representation.

Function DrawWord($\mathbf{u}$):

- If $j = 0$, return word $w \in W$ with probability $G_0(w) = 1/V$.
- Else with probabilities proportional to:

  $\max(0, c_{\mathbf{u}wk} - d_{|\mathbf{u}|})$: sit customer at table k (increment $c_{\mathbf{u}wk}$); return word w.

  $\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot}$: let $w \leftarrow$ DrawWord($\pi(\mathbf{u})$); sit customer at an unoccupied table k^{new} serving dish w (increment $t_{\mathbf{u}w}$, set $c_{\mathbf{u}wk^{\text{new}}} = 1$); return w.

Inference Schemes

$$p(w|\mathbf{u},\mathcal{D}) = \int p(w|\mathbf{u},\mathcal{S},\Theta)p(\mathcal{S},\Theta|\mathcal{D})d(\mathcal{S},\Theta)$$

where $\Theta = \{\theta_m, d_m : 0 \le m \le n-1\}$.

We can approximate the integral with sample $\{\mathcal{S}^{(i)}, \Theta^{(i)}\}_{i=1}^{I}$ drawn from $p(\mathcal{S},\Theta|\mathcal{D})$:

$$p(w|\mathbf{u},\mathcal{D}) \approx \sum_{i=1}^{I} p(w|\mathbf{u},\mathcal{S}^{(i)},\Theta^{(i)})$$

while $p(w|\mathbf{u},\mathcal{S},\Theta)$ is given by the function:

$$p(w \mid 0,\mathcal{S},\boldsymbol{\Theta}) = 1/V$$

$$p(w \mid \mathbf{u},\mathcal{S},\boldsymbol{\Theta}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|}t_{\mathbf{u}w\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} p(w \mid \pi(\mathbf{u}),\mathcal{S},\boldsymbol{\Theta})$$

Sampling for Seating Arrangements

We can obtain a Gibbs sampler for the hierarchical Pitman-Yor language model directly using the Chinese restaurant representation. The Gibbs sampler only keep track of which table each customer sits at, while the other pieces of information in the seating arrangement can be reconstructed from this. The sampler then iterates over all customers present in each restaurant, resampling the table at which each customer sits.

This resampling can be performed most easily using two routines: a **RemoveCustomer** routine that removes a customer from the restaurant, and an **AddCustomer** routine which adds the customer back into the restaurant, sitting her at some random table.

Sampling for Seating Arrangements

Function AddCustomer($\mathbf{u}$,w):

Adds a new customer eating dish w into restaurant $\mathbf{u}$.

- If $\mathbf{u} = 0$ then increment $c_{0w\cdot}$.
- Else with probabilities proportional to:

  $\max(0, c_{\mathbf{u}wk} - d_{|\mathbf{u}|})$: sit customer at k^{th} table in restaurant $\mathbf{u}$ (increment $c_{\mathbf{u}wk}$).

  $(\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|}t_{\mathbf{u}\cdot})$DishProbability$(\pi(\mathbf{u}),w)$: sit customer at a new table k^{new} serving dish w in restaurant $\mathbf{u}$ (increment $t_{\mathbf{u}w}$, set $c_{\mathbf{u}wk^{\text{new}}} = 1$); AddCustomer$(\pi(\mathbf{u}),w)$.

Function RemoveCustomer($\mathbf{u}$,w):

Removes a customer eating dish w from restaurant $\mathbf{u}$.

- If $\mathbf{u} = 0$ then decrement $c_{0w\cdot}$.
- Else with probabilities proportional to:

  $c_{\mathbf{u}wk}$: remove a customer from k^{th} table in restaurant $\mathbf{u}$ (decrement $c_{\mathbf{u}wk}$).
- If as a result the k^{th} table becomes unoccupied then RemoveCustomer$(\pi(\mathbf{u}),w)$.

Sampling for Seating Arrangements

We use auxiliary variable sampling routine that is easy to implement using basic operations. We assumes that each discount parameter has prior distribution $d_m \sim \text{Beta}(a_m, b_m)$ while each strength parameter has prior $\theta_m \sim \text{Gamma}(\alpha_m, \beta_m)$.

$$d_m \sim \text{Beta}\left(a_m + \sum_{\mathbf{u}:|\mathbf{u}|=m, t_{\mathbf{u}\cdot}\geq 2} \sum_{i=1}^{t_{\mathbf{u}\cdot}-1} (1-y_{\mathbf{u}i}), b_m + \sum_{\mathbf{u},w,k:|\mathbf{u}|=m, c_{\mathbf{u}wk}\geq 2} \sum_{j=1}^{c_{\mathbf{u}wk}-1} (1-z_{\mathbf{u}wkj})\right)$$

$$\theta_m \sim \text{Gamma}\left(\alpha_m + \sum_{\mathbf{u}:|\mathbf{u}|=m, t_{\mathbf{u}\cdot}\geq 2} \sum_{i=1}^{t_{\mathbf{u}\cdot}-1} y_{\mathbf{u}i}, \beta_m - \sum_{\mathbf{u}:|\mathbf{u}|=m, t_{\mathbf{u}\cdot}\geq 2} \log x_{\mathbf{u}}\right)$$

$$x_{\mathbf{u}} \sim \text{Beta}(\theta_{|\mathbf{u}|}+1, c_{\mathbf{u}\cdot\cdot}-1)$$

$$y_{\mathbf{u}i} \sim \text{Bernoulli}\left(\frac{\theta_{|\mathbf{u}|}}{\theta_{|\mathbf{u}|}+d_{|\mathbf{u}|}i}\right)$$

$$z_{\mathbf{u}wkj} \sim \text{Bernoulli}\left(\frac{j-1}{j-d_{|\mathbf{u}|}}\right)$$

Language Model

- n-gram language models:

$$p(w_n|w_{n-2}, w_{n-1}) =$$

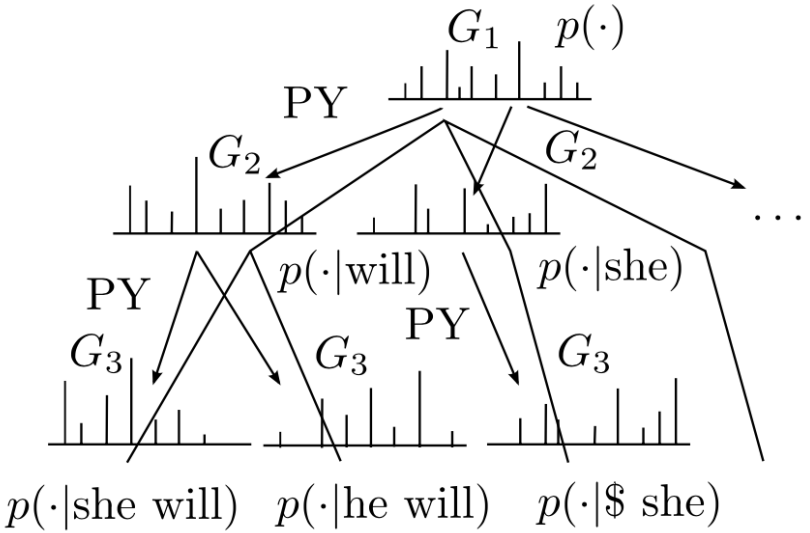
$$\lambda_1 p_1(w_n) + \lambda_2 p_2(w_n|w_{n-1}) + \lambda_3 p_3(w_n|w_{n-1}, w_{n-2})$$

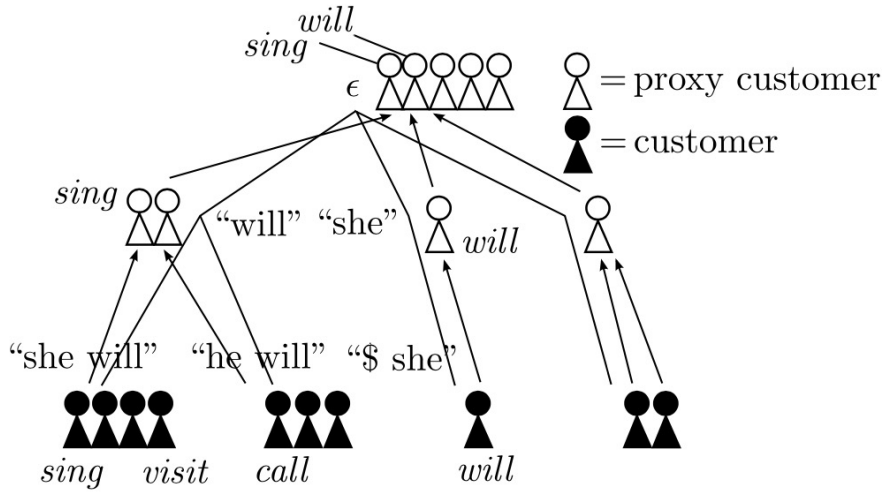
- Language model based on Pitman-Yor process:

$$p(w \mid 0, \mathcal{S}, \boldsymbol{\Theta}) = 1/V$$

$$p(w \mid \mathbf{u}, \mathcal{S}, \boldsymbol{\Theta}) = \frac{c_{\mathbf{u}w\cdot} - d_{|\mathbf{u}|} t_{\mathbf{u}w\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\cdot\cdot}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\cdot\cdot}} p(w \mid \pi(\mathbf{u}), \mathcal{S}, \boldsymbol{\Theta})$$

Word Segmentation with Nested Pitman-Yor Language Modeling

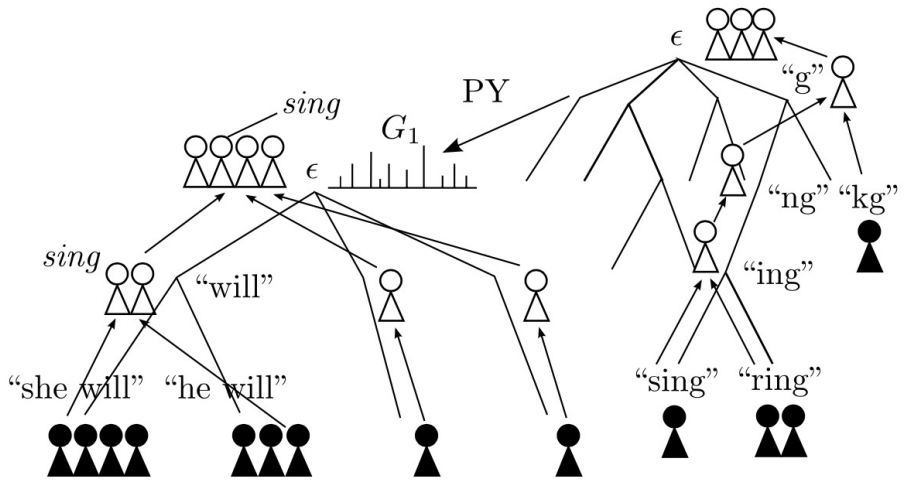




Generating *n*-gram distributions *G* hierarchically from the Pitman-Yor process.

Equivalent representation using a hierarchical Chinese Restaurant representation.

Nested Pitman-Yor Language Model



Chinese restaurant Representation

$$G_0(w) = p(c_1 \cdots c_k)$$
$$= \prod_{i=1}^{k} p(c_i | c_1 \cdots c_{i-1})$$

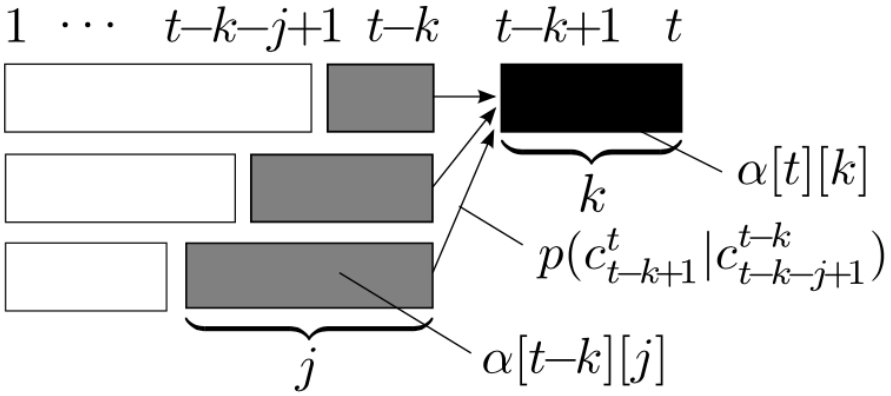
$$\mathrm{Po}(k|\lambda) = e^{-\lambda} \frac{\lambda^k}{k!}.$$

$$p(c_1 \cdots c_k) = p(c_1 \cdots c_k, k)$$
$$= \frac{p(c_1 \cdots c_k, k|\Theta)}{p(k|\Theta)} \mathrm{Po}(k|\lambda)$$

Blocked Gibbs Sampler

1: **for** $j = 1 \cdots J$ **do**
2: **for** s in randperm $(s_1, \cdots, s_D)$ **do**
3: **if** $j > 1$ **then**
4: Remove customers of $\mathbf{w}(s)$ from Θ
5: **end if**
6: Draw $\mathbf{w}(s)$ according to $p(\mathbf{w}|s, \Theta)$
7: Add customers of $\mathbf{w}(s)$ to Θ
8: **end for**
9: Sample hyperparameters of Θ
10: **end for**

Forward Filtering



$$\alpha[t][k] = \sum_{j=1}^{t-k} p(c_{t-k+1}^{t} | c_{t-k-j+1}^{t-k}) \cdot \alpha[t-k][j] \quad (7)$$

Forward-Backward Inference

Backward Sampling. Once the probability table $\alpha[t][k]$ is obtained, we can sample a word segmentation backwards. Since $\alpha[N][k]$ is a marginal probability of string c_1^N with the last k characters being a word, and there is always a sentence boundary token \$ at the end of the string, with probability proportional to $p(\$|c_{N-k}^N) \cdot \alpha[N][k]$ we can sample k to choose the boundary of the final word. The second final word is similarly sampled using the probability of preceding the last word just sampled: we continue this process until we arrive at the beginning of the string

for $t = 1$ to N **do**
  for $k = \max(1, t-L)$ to t **do**
    Compute $\alpha[t][k]$ according to (7).
  end for
end for
Initialize $t \leftarrow N$, $i \leftarrow 0$, $w_0 \leftarrow \$$
while $t > 0$ **do**
  Draw $k \propto p(w_i|c_{t-k+1}^t, \Theta) \cdot \alpha[t][k]$
  Set $w_i \leftarrow c_{t-k+1}^t$
  Set $t \leftarrow t - k$, $i \leftarrow i + 1$
end while
Return $\mathbf{w} = w_i, w_{i-1}, \cdots, w_1$.

Any questions?

AI Research Group
Fudan University