算法交流与学习

Dirichlet process

合工大管院电子商务研究所 钱洋
2018-03-20

sober_qian

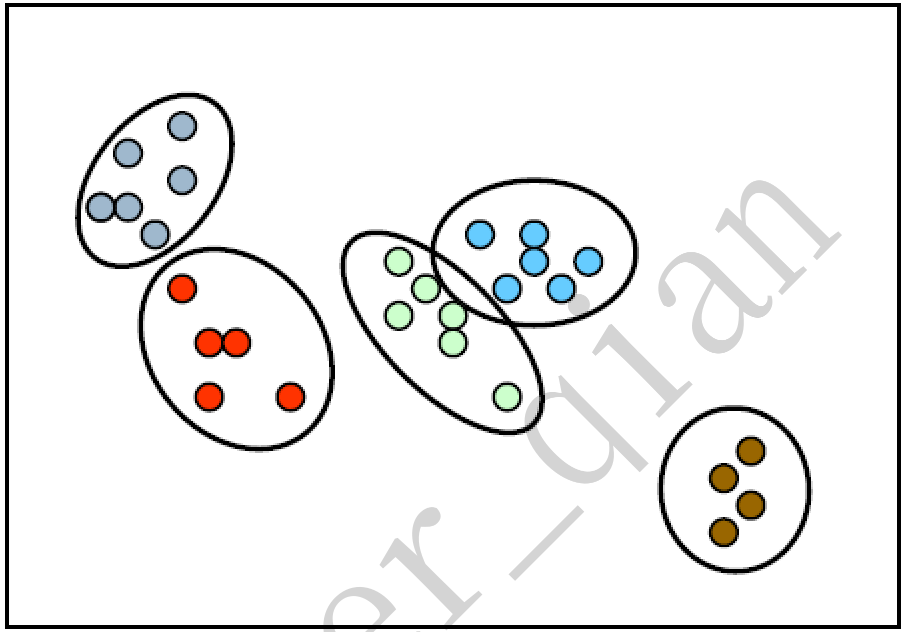
主要内容

- DP应用场景
- DP的定义及构造
- DPMM模型
- HDP模型

sober_qian

DP应用场景

一组产生自高斯分布混合的数据集：



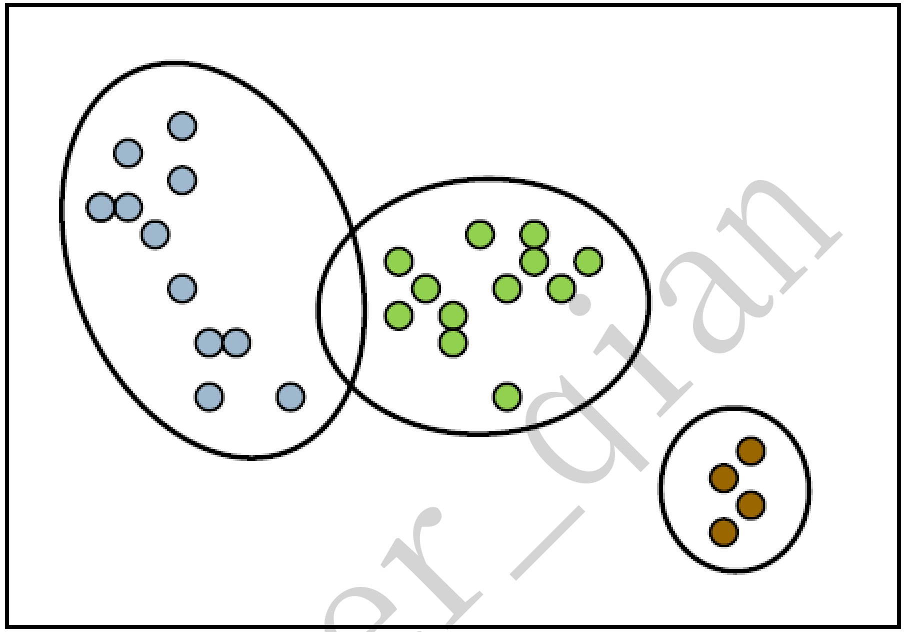
$$\theta_j = \{\boldsymbol{\mu}_j, S_j, \pi_j\}$$

$$p(\boldsymbol{x}|\theta_1,\ldots,\theta_K) = \sum_{j=1}^{K} \boldsymbol{\pi}_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, S_j)$$

sober_qian

DP应用场景

一组产生自高斯分布混合的数据集：



$$\theta_j = \{\boldsymbol{\mu}_j, S_j, \pi_j\}$$

到底有多少个K，
能聚多少类？

$$p(\boldsymbol{x}|\theta_1,\ldots,\theta_K) = \boxed{\sum_{j=1}^{K}} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, S_j)$$

sober_qian

DP应用场景

DP是一种有名的非参贝叶斯模型，特别适合解决各种聚类问题。其优势是用于建立混合模型使得类别数目无需人工设定，而是由模型自主学习。

在自然语言处理领域，很多问题都是挖掘语料中的新的或者未知的知识，这些信息往往是缺少先验知识的，而DP过程的有点在很多自然语言处理的应用问题上得到了充分的体现。

sober_qian

主要内容

- DP应用场景
- DP的定义及构造
- DPMM模型
- HDP模型

sober_qian

DP的引入

$$\begin{array}{ccccccc} x_1 & x_2 & x_3 & \cdots & x_i & \cdots & x_n \\ \downarrow & \downarrow & \downarrow & & \downarrow & & \downarrow \\ \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_i & \cdots & \theta_n \end{array}$$

- 每个数据 x 对应一个产生它的分布，其参数为θ。
- θ必定有一个分布

$$\theta_i \sim H(\theta)$$

- 如果H是连续的，每次产生x的分布必然不同：

$$\theta_i \sim G(\alpha_0, H) \qquad \Leftarrow \qquad \theta_1, \theta_2 \sim H \quad \Rightarrow \quad p(\theta_1 = \theta_2) = 0$$

- 生成θ的不能从连续分布中产生。

sober_dian

DP的数学定义

假设G_0是测度空间Θ上的随机概率分布，参数α_0是正实数，空间Θ上的概率分布G如果满足如下条件：

对测度空间 Θ 的任意一个有限划分 $A_1, \ldots, A_r$，均有一下关系存在：

$$(G(A_1), \cdots, G(A_r)) \sim \mathrm{Dir}(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r))$$

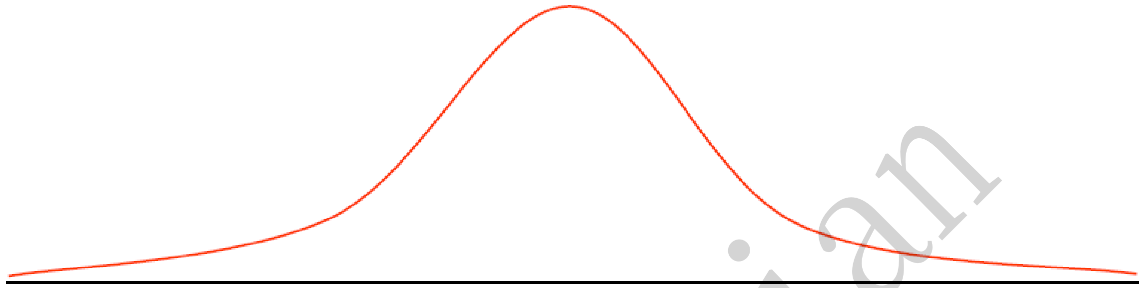
则G服从基分布为G_0和集中度参数为α_0组成的DP过程，即：

$$G \sim \mathrm{DP}(\alpha_0, G_0)$$

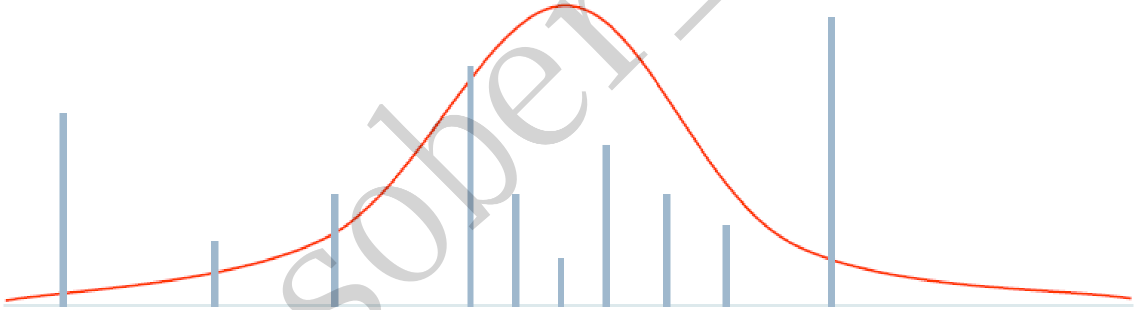
sober-qian

DP的理解

高斯分布G_0



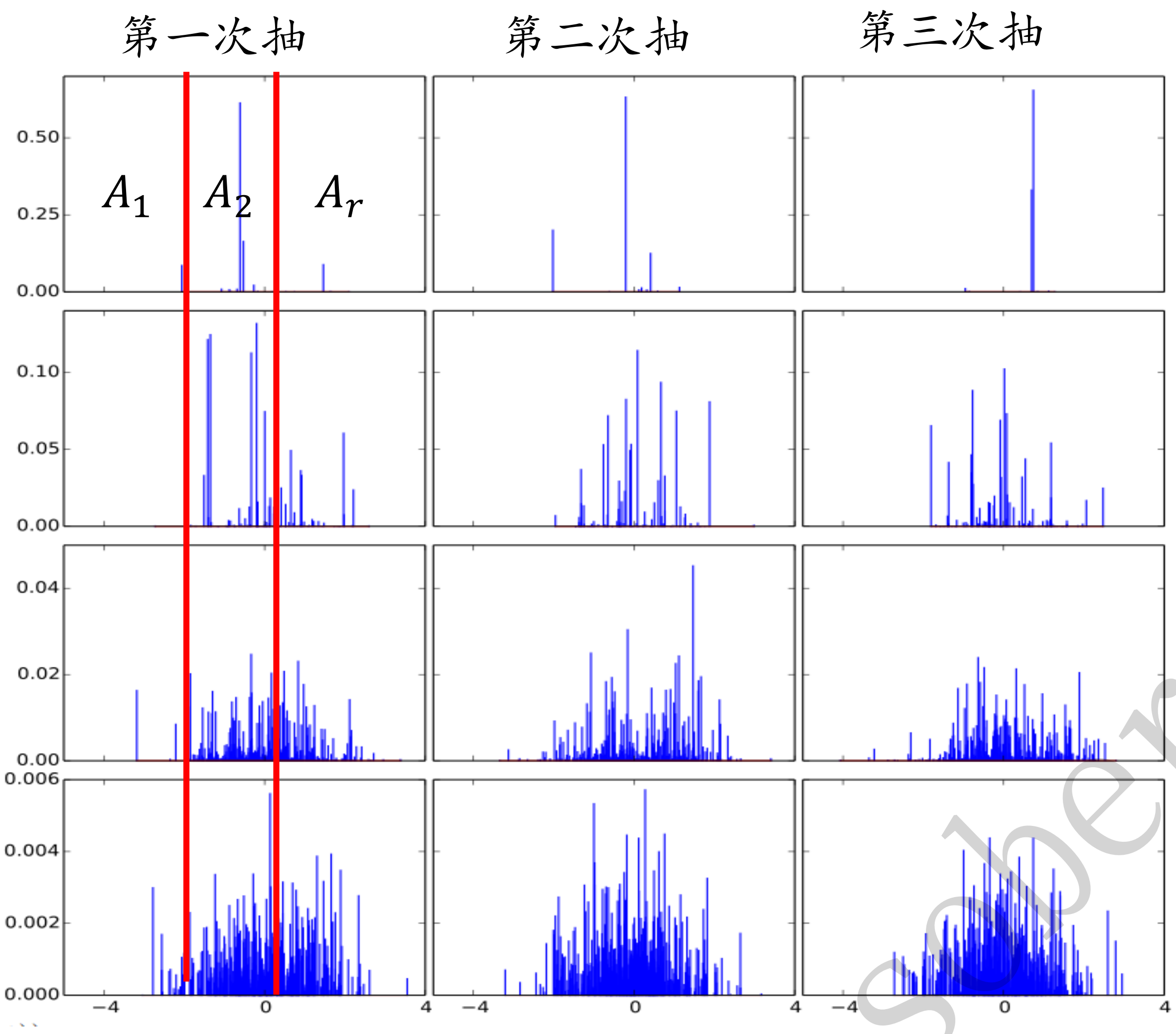
$G \sim DP(\alpha_0, G_0)$



DP中采样得到的分布是可数无限个离散概率，无法用有限数量的参数描述， 因此DP是非参数模型

sober_qian

DP的理解



来源：wikipedia

$G \sim DP(\alpha_0, N(0,1))$

α_0=(1,10,100,1000)

棍的长度即为混合的权重，有无限多个棍，所有棍的长度和为1.

$$G(\theta) = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}(\theta)$$

每次抽都是一个完整的分布，这些分布的特性(概率性质如何)：

$$(G(A_1), \cdots, G(A_r)) \sim \mathrm{Dir}(\alpha_0 G_0(A_1), \cdots, \alpha_0 G_0(A_r))$$

sober_qian

Stick-breaking构造

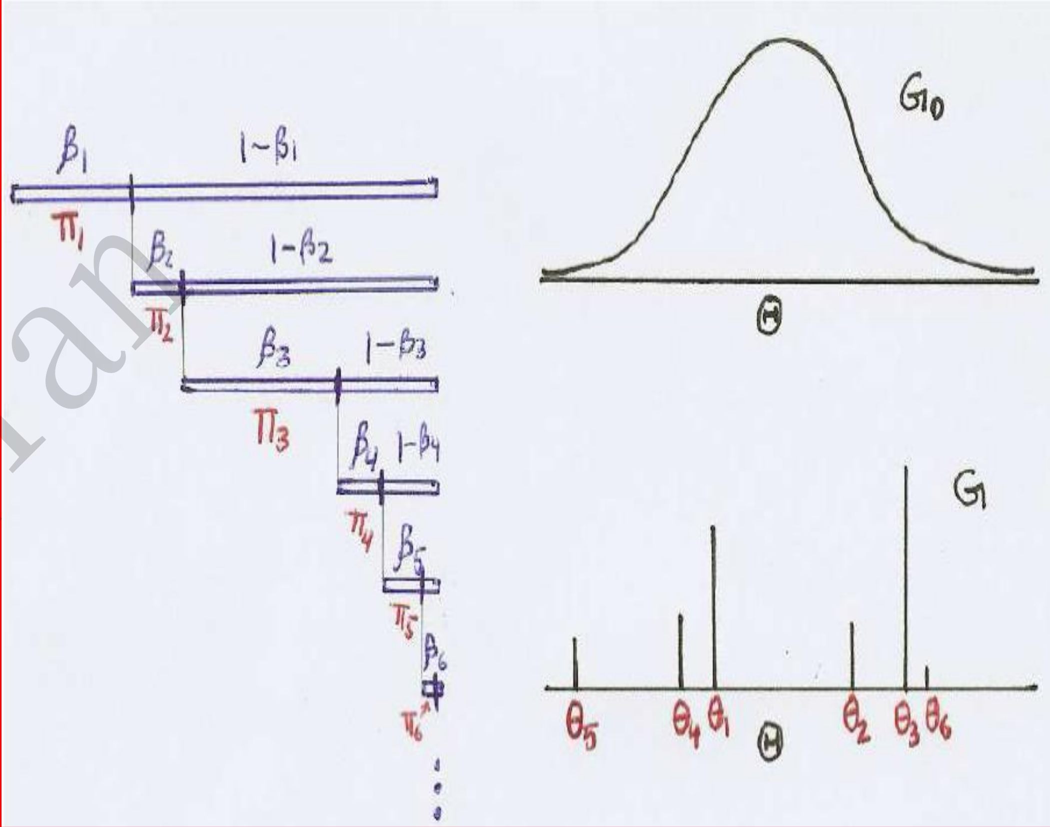
基于相互独立的变量序列 $(\beta_k)_{k=1}^{\infty}$ 和 $(\phi_k)_{k=1}^{\infty}$ 的 Stick-breaking 构造:

$$\beta_k|\alpha_0, G_0 \sim \mathrm{Beta}(1, \alpha_0), \quad \phi_k|\alpha_0, G_0 \sim G_0$$

定义一随机概率分布 G 如下:

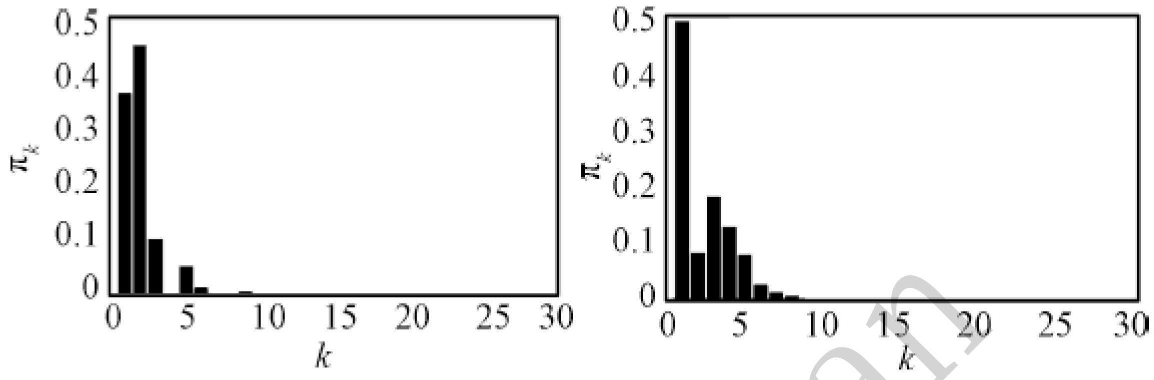
$$\pi_k = \beta_k \prod_{l=1}^{k-1}(1-\beta_l), \quad G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$$

$$\pi_1 = \beta_1, \pi_2 = (1-\beta_1)\beta_2, \ldots, \pi_c = \beta_c \prod_{j=1}^{c-1}(1-\beta_j), \ldots$$

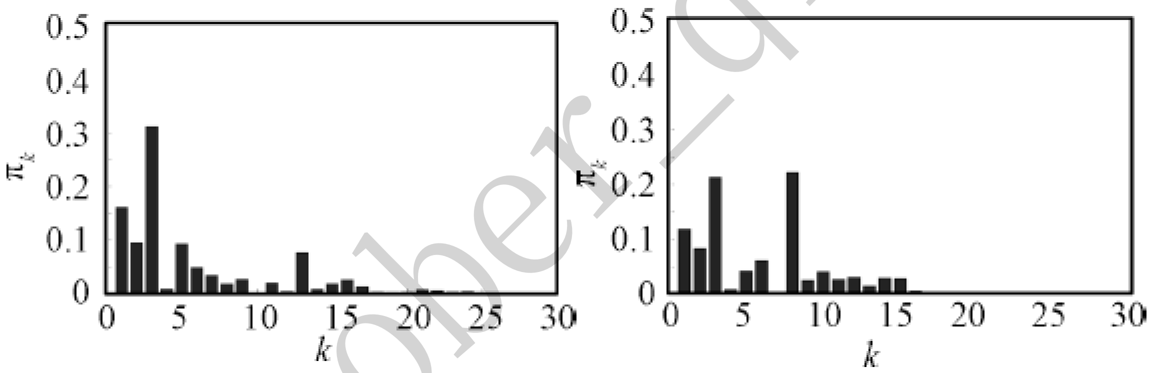


sober_qian

Stick-breaking构造



(a) $\alpha_0 = 1$ 时权重系数 1 (b) $\alpha_0 = 1$ 时权重系数 2



(c) $\alpha_0 = 5$ 时权重系数 1 (d) $\alpha_0 = 5$ 时权重系数 2

$\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha_0)$

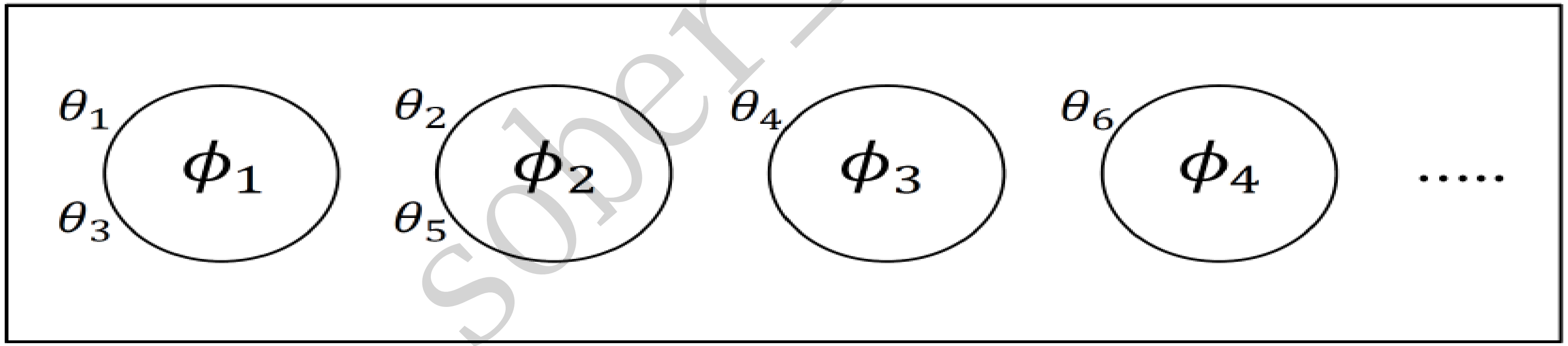
Griffiths, Engen, McCloskey

soben_qian

Chinese restaurant process构造

将θ_i比作进入餐厅的顾客，不同值的ϕ_k对应顾客就坐的桌子。令第i个参数θ_i的指示因子为z_i，则$\theta_i = \phi_{z_i}$：

- 餐厅内有无数多张桌子；
- 每个顾客坐一张桌子；
- 第一个顾客坐第一张桌子；
- 第i个顾客就坐于第k个桌子的概率与该座子的顾客数m_k成正比；就坐于一张新桌子的概率正比于α_0。



soher_qian

Chinese restaurant process构造

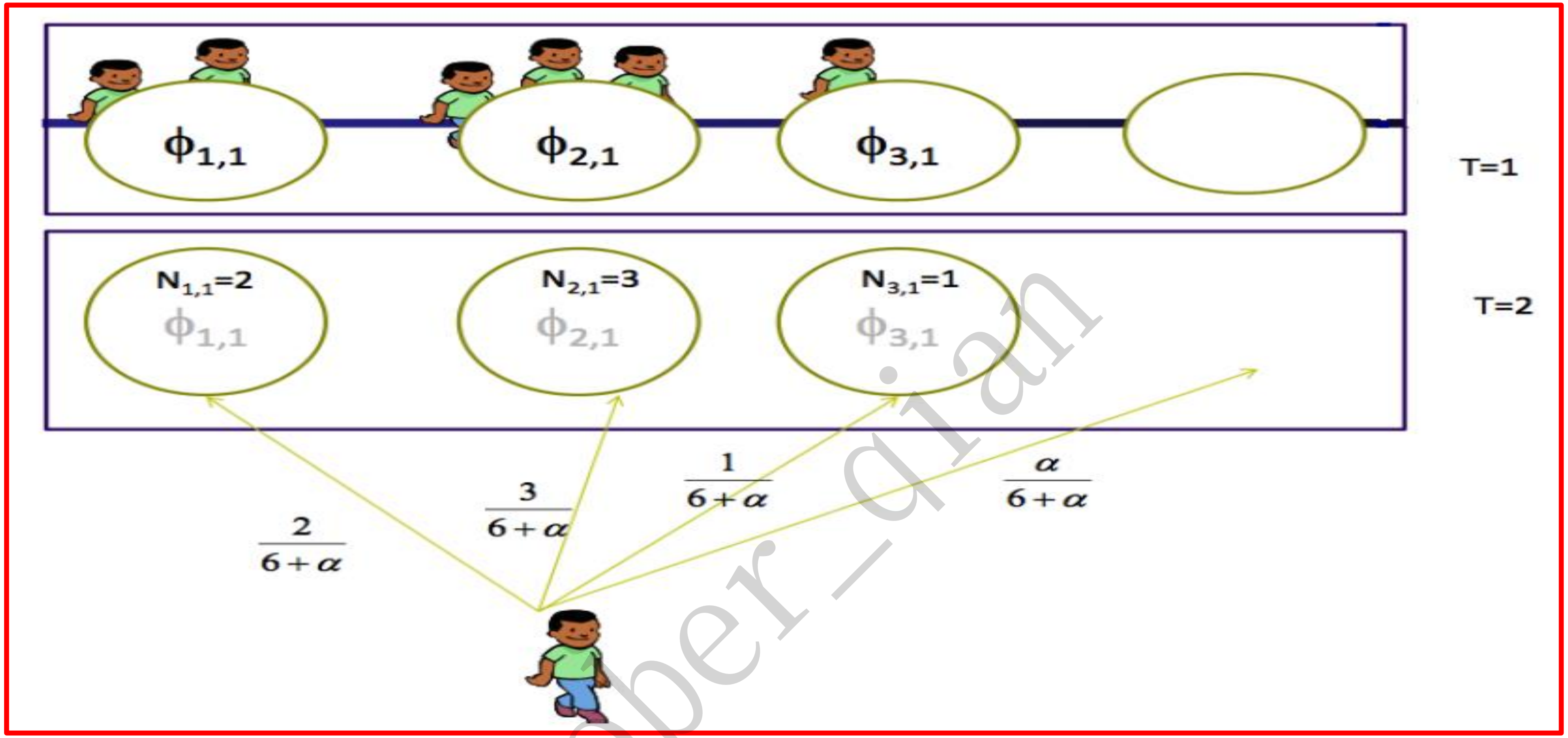
将θ_i比作进入餐厅的顾客，不同值的ϕ_k对应顾客就坐的桌子。令第i个参数θ_i的指示因子为z_i，则$\theta_i = \phi_{z_i}$：

- 餐厅内有无数多张桌子；
- 每个顾客坐一张桌子；
- 第一个顾客坐第一张桌子；
- 第i个顾客就坐于第k个桌子的概率与该座子的顾客数m_k成正比；就坐于一张新桌子的概率正比于α_0。

$$z_i | z_1, \cdots, z_{i-1}, \alpha_0, G_0 \sim \sum_k^K \frac{m_k}{i-1+\alpha_0} \delta(z_i, k) + \boxed{\frac{\alpha_0}{i-1+\alpha_0} \delta(z_i, \bar{k})}$$

sober-qian

Chinese restaurant process构造



- CRP很好的体现了DP的聚类性质。其中桌子就是所要聚的类。

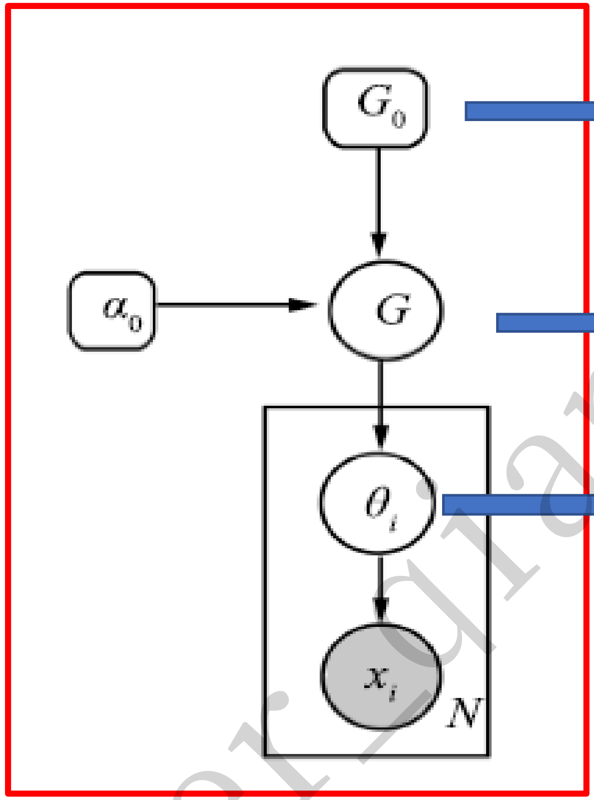
sober_qian

主要内容

- DP应用场景
- DP的定义及构造
- DPMM模型
- HDP模型

sober_qian

Basic DPMM



G_0为基分布

G 产生无穷多的点(point masses)

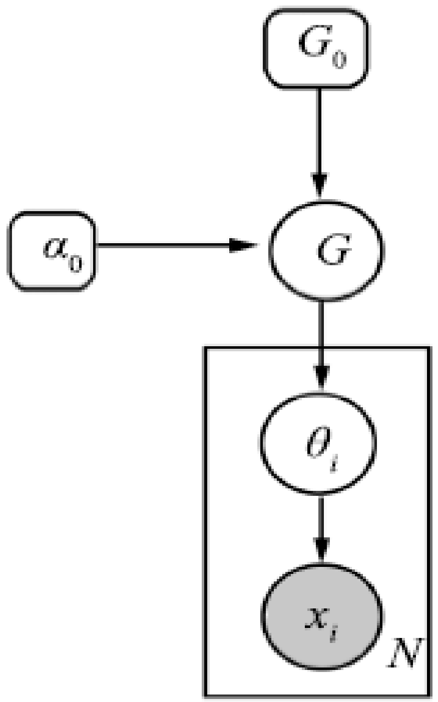
集中度参数为α_0决定了离散程度，决定了与G_0分布的相似度

从G中抽样N次获取的参数代表不同的混合成分（ mixture components）

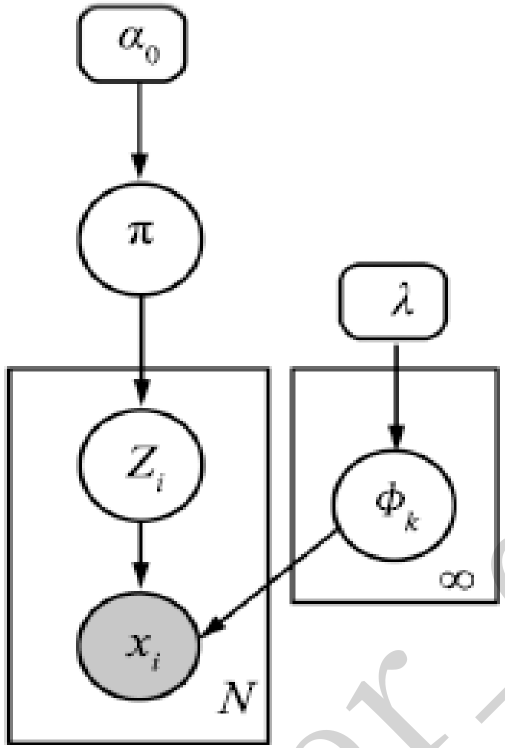
$$x_i|\theta_i \sim F(\theta_i) \qquad \theta_i|G \sim G \qquad G \sim \mathrm{DP}(\alpha_0, G_0)$$

sober_qian

DPMM的Stick-breaking构造



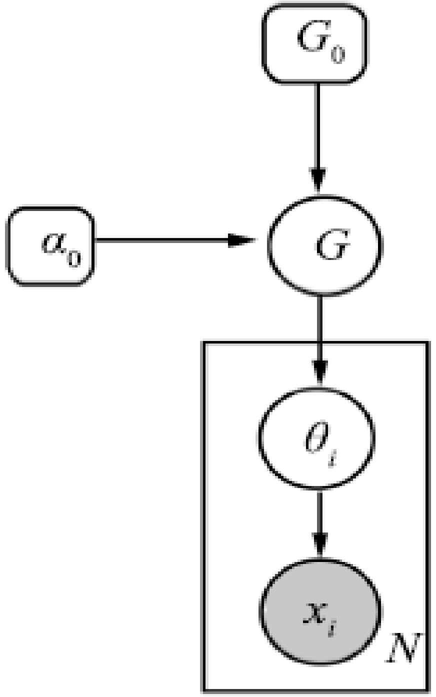
$$x_i|\theta_i \sim F(\theta_i)$$
$$\theta_i|G \sim G$$
$$G \sim \mathrm{DP}(\alpha_0, G_0)$$



$$\boldsymbol{\pi}|\alpha_0 \sim \mathrm{GEM}(\alpha_0), \quad z_i|\boldsymbol{\pi} \sim \boldsymbol{\pi}$$
$$\phi_k|G_0 \sim G_0, \quad x_i|z_i, \quad (\phi_k)_{k=1}^{\infty} \sim F(\phi_{z_i})$$
$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k} \qquad G_0 = g(\lambda)$$

sober_qian

DPMM的采样



- 关于指示因子z_i的条件分布为：

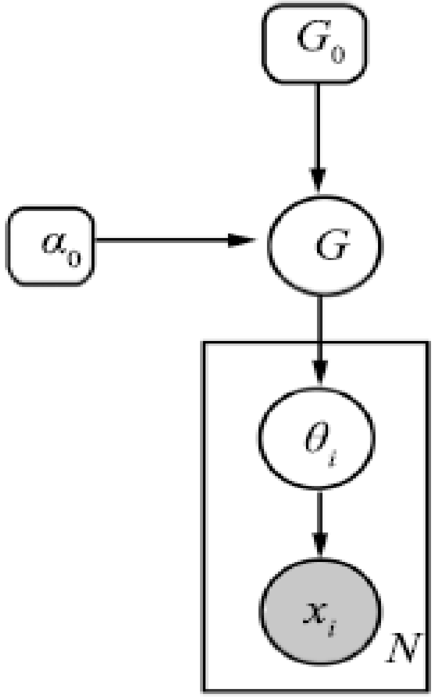
$$p(z_i|x_1,\cdots,x_N,\mathcal{Z}_{\setminus i},\lambda,\alpha_0)\propto p(z_i|\mathcal{Z}_{\setminus i},\alpha_0)p(x_i|z_1,\cdots,z_N,\mathcal{X}_{\setminus i},\lambda)$$

- 基于CRP(中餐馆过程)：

$$z_i|\mathcal{Z}_{\setminus i},\alpha_0\sim\sum_{k}^{K}\frac{N_k^{\setminus i}}{N-1+\alpha_0}\delta(z_i,k)+\frac{\alpha_0}{N-1+\alpha_0}\delta(z_i,\bar{k})$$

sober_qian

DPMM的采样



- 若指示因子$z_i = k$（已有的簇）：

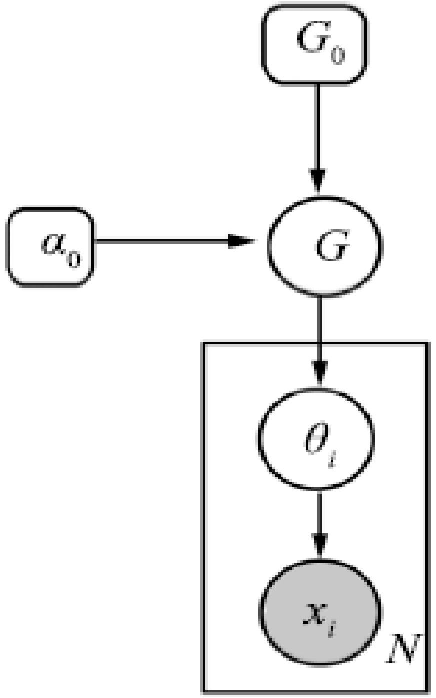
$$p(x_i|z_i = k, \mathcal{X}_{\setminus i}, \lambda) = p(x_i|\{x_j|z_j = k, j \neq i\}, \lambda) = \frac{\int_\theta f(x_i|\theta) \prod_{z_j=k, j\neq i} f(x_j|\theta) g(\theta|\gamma) \mathrm{d}\theta}{\int_\theta \prod_{z_j=k, j\neq i} f(x_j|\theta) g(\theta|\lambda) \mathrm{d}\theta}$$

- 若指示因子$z_i = \bar{k}$（新簇）：

$$p(x_i|z_i = \bar{k}, \mathcal{X}_{\setminus i}, \lambda) = p(x_i|\lambda) = \int_\Theta p(x_i|\theta) g(\theta|\lambda) \mathrm{d}\theta$$

sober_qian

DPMM的采样

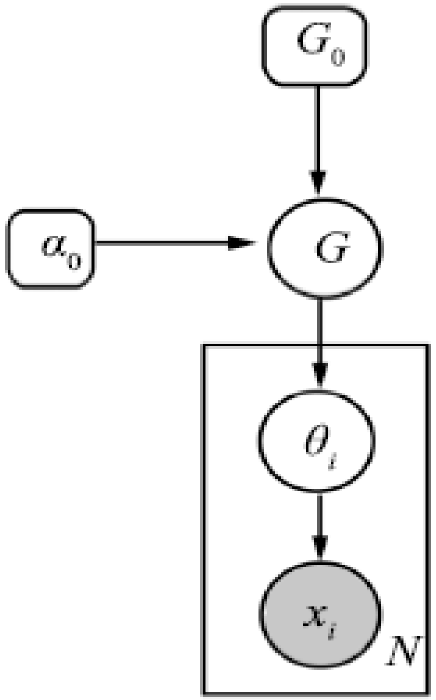


➢ 因此：

$$p(z_i|x_1,\cdots,x_N,Z_{\backslash i},\lambda,\alpha_0) \propto \sum_k^K \frac{N_k^{\backslash i}}{N-1+\alpha_0} \times$$
$$p(x_i|\{x_j|z_j=k,j\neq i\},\lambda)\delta(z_i,k) +$$
$$\frac{\alpha_0}{N-1+\alpha_0}\int_\Theta p(x_i|\theta)g(\theta|\lambda)\mathrm{d}\theta\delta(z_i,\bar{k})$$

sober_qian

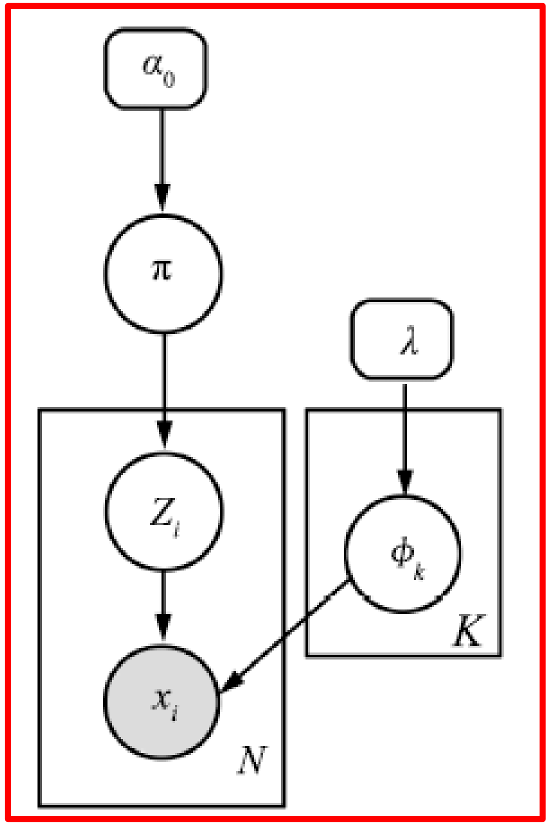
DPMM的采样



- 注：在采样过程中，选择$\theta_i \sim G_0$和$x_i \sim F(\theta_i)$是共轭的。常用的是Dirichlet分布和其共轭多项式分布，Gaussian-Wishart分布和其共轭Gaussian分布。

sober_dian

有限混合模型的无限近似



有限单元混合模型的概率图

$$\boldsymbol{\pi}|\alpha_0 \sim \mathrm{Dir}\left(\frac{\alpha_0}{K}, \cdots, \frac{\alpha_0}{K}\right), \quad z_i|\boldsymbol{\pi} \sim \boldsymbol{\pi}$$

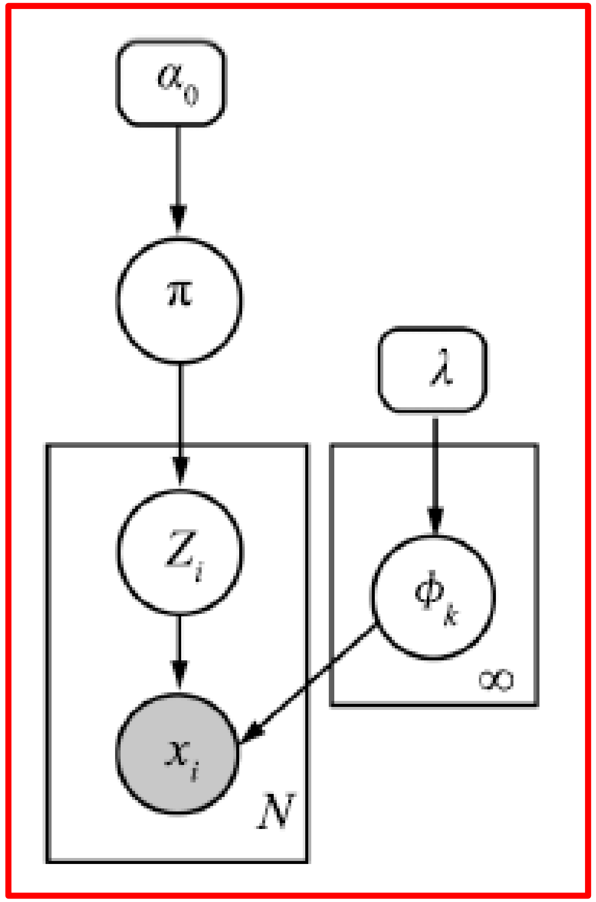
$$\phi_k|G_0 \sim G_0, \quad x_i|z_i, (\phi_k)_{k=1}^K \sim F(\phi_{z_i})$$

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

基于Dirichlet与多项式分布共轭的性质：

$$p(z_i = k|Z_{\setminus i}, \alpha_0) = \frac{N_k^{\setminus i} + \frac{\alpha_0}{K}}{N - 1 + \alpha_0}$$

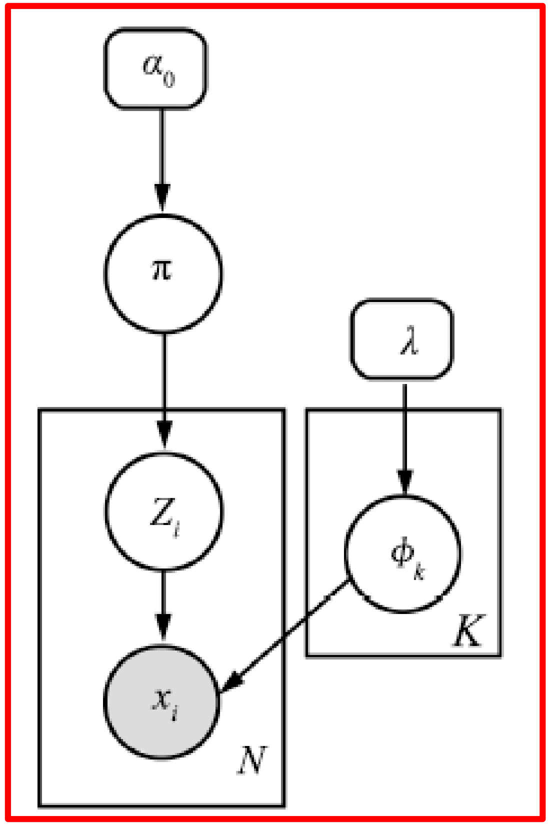
有限混合模型的近似过程为我们提供了近似求解DP的一种途径



DP混合模型

sober_qian

有限混合模型的无限近似

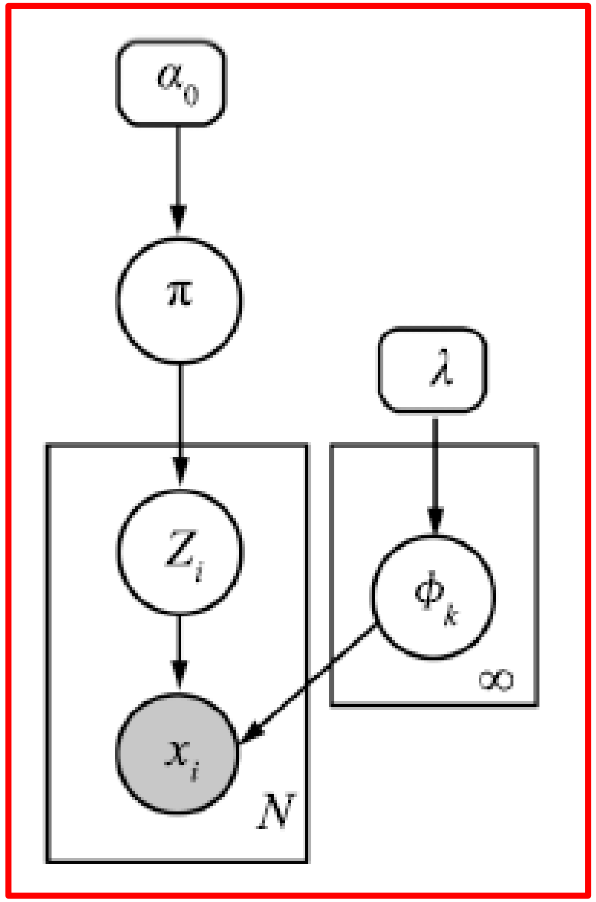


有限单元混合模型的概率图

$$p(z_i = k | Z_{\setminus i}, \alpha_0) = \frac{N_k^{\setminus i} + \frac{\alpha_0}{K}}{N - 1 + \alpha_0}$$

$K \to \infty$:

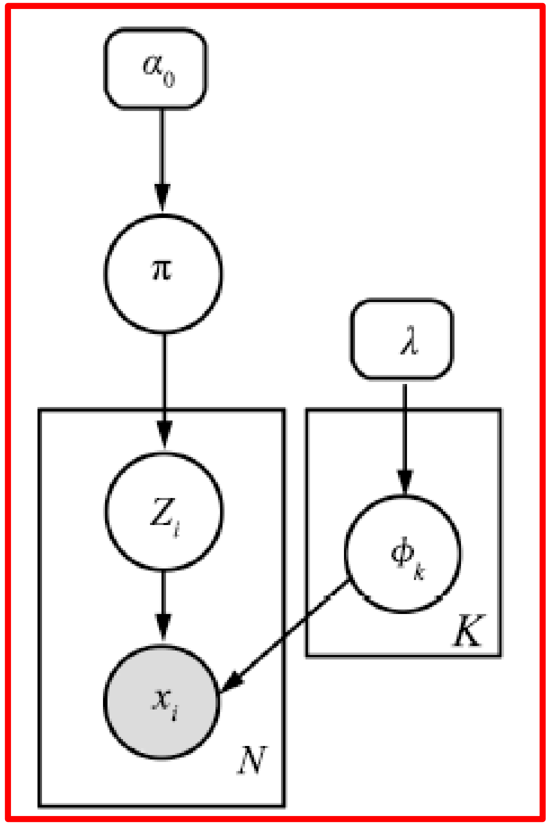
$$p(z_i = k | Z_{\setminus i}, \alpha_0) = \frac{N_k^{\setminus i}}{N - 1 + \alpha_0}$$



DP混合模型

sober_qian

有限混合模型的无限近似



有限单元混合模型的概率图

$$p(z_i = k|\mathcal{Z}_{\setminus i}, \alpha_0) = \frac{N_k^{\setminus i} + \frac{\alpha_0}{K}}{N - 1 + \alpha_0}$$

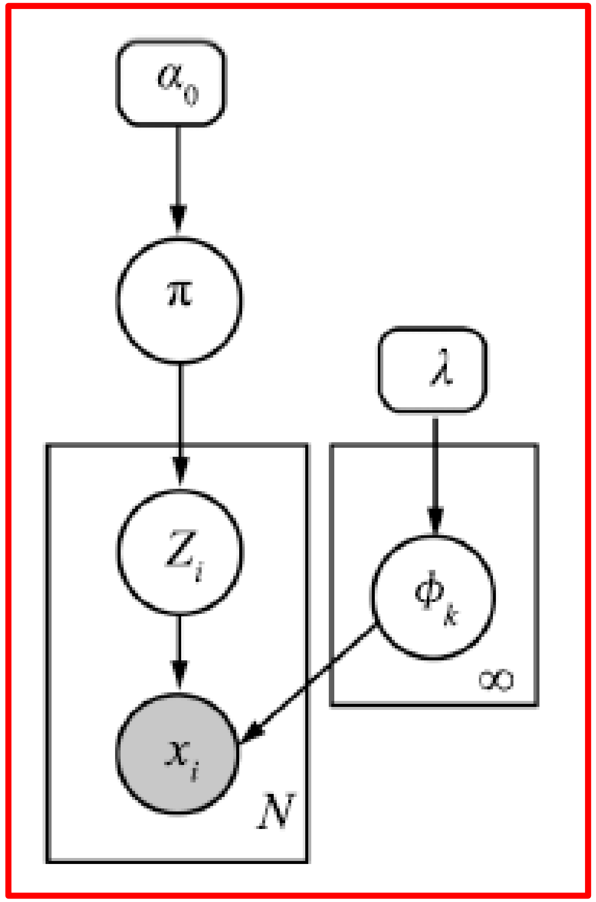
x_i为空类时：

$$p(z_i = \bar{k}|\mathcal{Z}_{\setminus i}, \alpha_0) =$$

$$1 - \sum_{k, N_k^{\setminus i} \geq 0} p(z_i = k|\mathcal{Z}_{\setminus i}, \alpha_0) =$$

$$1 - \sum_{k} \frac{N_k^{\setminus i}}{N - 1 + \alpha_0} =$$

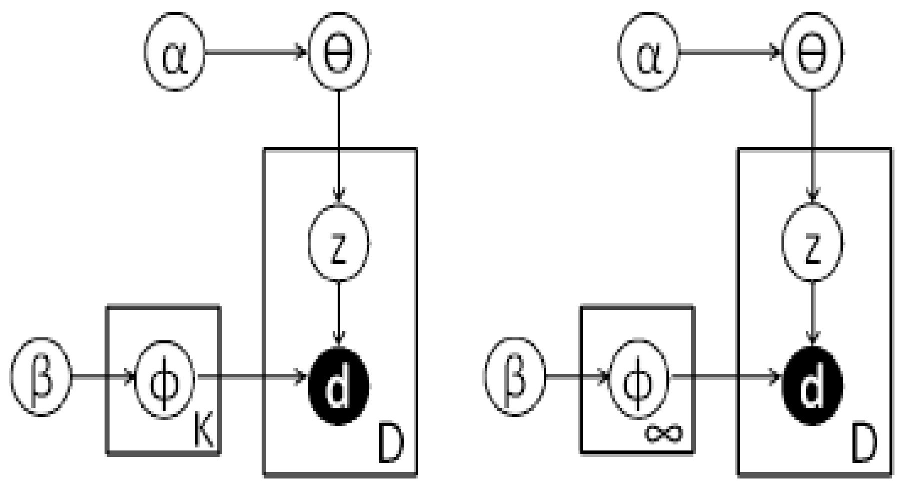
$$\frac{\alpha_0}{N - 1 + \alpha_0}$$



DP混合模型

sober-qian

基于DPMM的文档聚类(案例)



(a) DMM (b) DPMM

DMM的生成过程：

$$\Theta|\alpha \sim Dir(\alpha)$$
$$z_d|\Theta \sim Mult(\Theta) \qquad d = 1, ..., D$$
$$\Phi_k|\beta \sim Dir(\beta) \qquad k = 1, ..., K$$
$$d|z_d, \{\Phi_k\}_{k=1}^{K} \sim p(d|\Phi_{z_d})$$

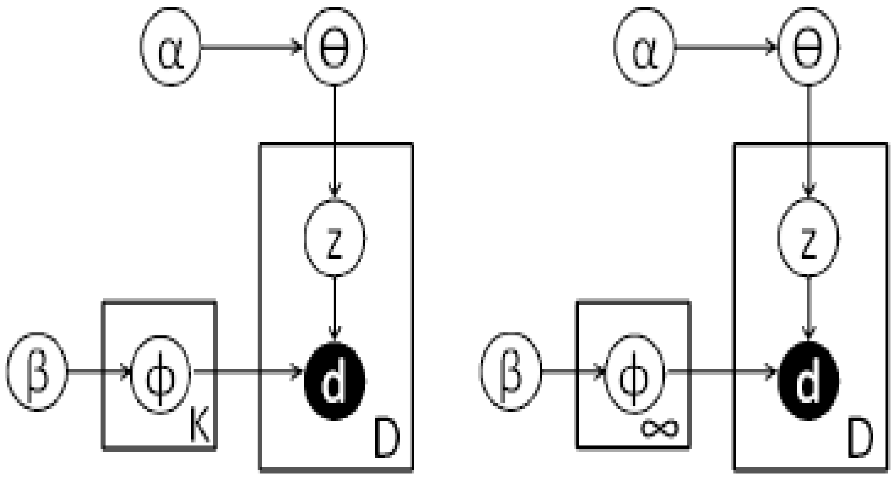
DPMM的生成过程：

$$\Theta|\alpha \sim GEM(1, \alpha)$$
$$z_d|\Theta \sim Mult(\Theta) \qquad d = 1, ..., D$$
$$\Phi_k|\beta \sim Dir(\beta) \qquad k = 1, ..., \infty$$
$$d|z_d, \{\Phi_k\}_{k=1}^{\infty} \sim p(d|\Phi_{z_d})$$

Dirichlet Process Multinomial Mixture model

sober_qian

基于DPMM的文档聚类(案例)



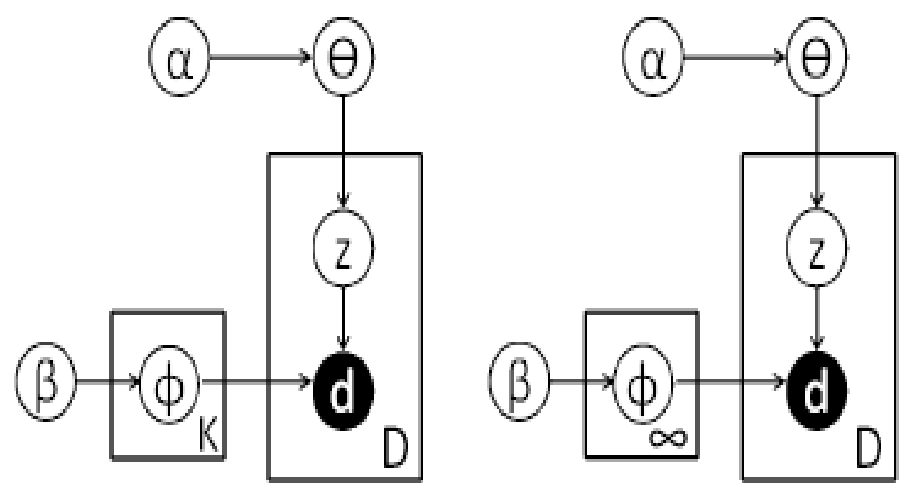
(a) DMM (b) DPMM

在其他文档簇已知的情况下，文档d所属的簇：

$$
\begin{aligned}
& p(z_d = z|\vec{z}_{\neg d}, \alpha) \\
& = \int Dir(\Theta|\vec{m}_{\neg d} + \alpha/K) Mult(z_d = z|\Theta) d\Theta \\
& = \int \frac{1}{\Delta(\vec{m}_{\neg d} + \alpha/K)} \Theta_z \prod_{k=1,k\neq z}^{K} \Theta_k^{m_{k,\neg d}+\alpha/K-1} d\Theta \\
& = \frac{\Delta(\vec{m} + \alpha/K)}{\Delta(\vec{m}_{\neg d} + \alpha/K)} \\
& = \frac{\prod_{k=1}^{K} \Gamma(m_k + \alpha/K)}{\Gamma(\sum_{k=1}^{K}(m_k + \alpha/K))} \frac{\Gamma(\sum_{k=1}^{K}(m_{k,\neg d} + \alpha/K))}{\prod_{k=1}^{K} \Gamma(m_{k,\neg d} + \alpha/K)} \\
& = \frac{\Gamma(m_{z,\neg d} + \alpha/K + 1)}{\Gamma(m_{z,\neg d} + \alpha/K)} \frac{\Gamma(D - 1 + \alpha)}{\Gamma(D + \alpha)} \\
& = \frac{m_{z,\neg d} + \alpha/K}{D - 1 + \alpha}
\end{aligned}
$$

sober_qian

基于DPMM的文档聚类(案例)



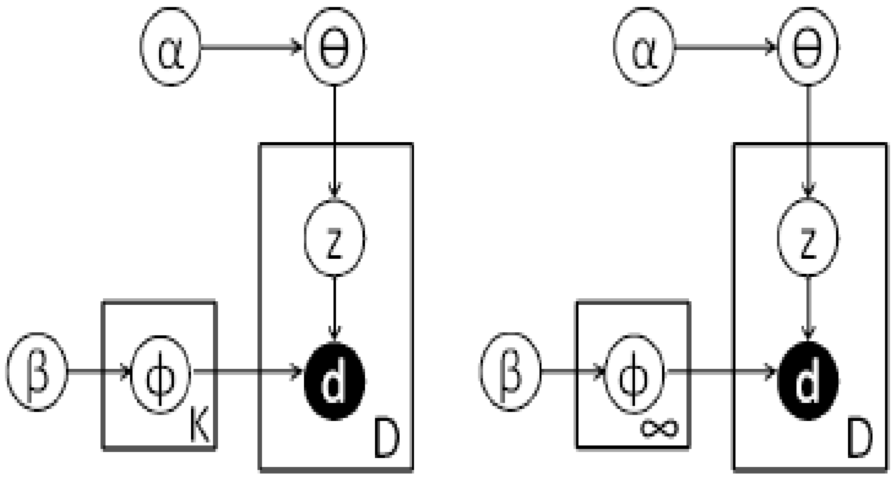
(a) DMM (b) DPMM

在其他文档簇已知的情况下，文档选择一个新簇：

$$
\begin{aligned}
p(z_d = K+1|\vec{z}_{\neg d}, \alpha) \\
&= 1 - \sum_{k=1}^{K} p(z_d = k|\vec{z}_{\neg d}, \alpha) \\
&= 1 - \frac{\sum_{k=1}^{K} m_{k,\neg d}}{D-1+\alpha} \\
&= 1 - \frac{D-1}{D-1+\alpha} \\
&= \frac{\alpha}{D-1+\alpha}
\end{aligned}
$$

sober_qian

基于DPMM的文档聚类(案例)



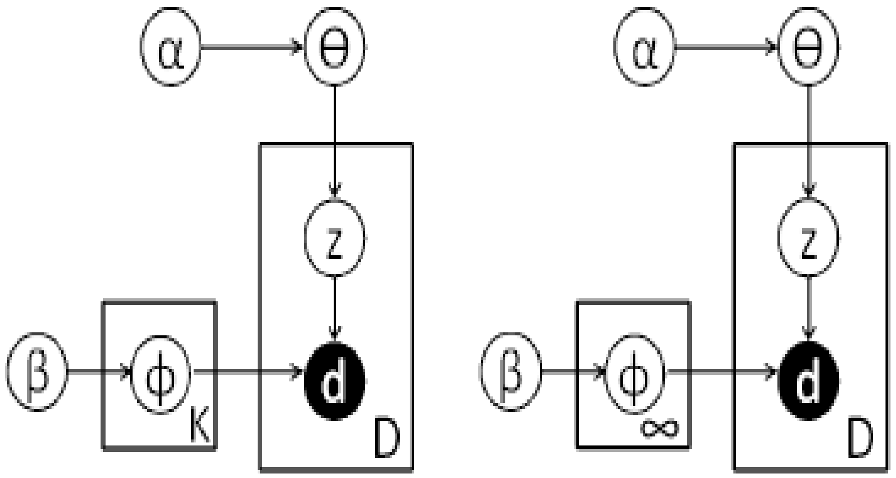
(a) DMM (b) DPMM

生成文档的条件概率(已有簇的情况下)：

$$
\begin{aligned}
& p(d|z_d = z, \vec{d}_{z,\neg d}, \beta) \\
& = \int Dir(\Phi_z|\vec{n}_{z,\neg d}+\beta) \prod_{w\in d} Mult(w|\Phi_z) d\Phi_z \\
& = \int \frac{1}{\Delta(\vec{n}_{z,\neg d}+\beta)} \prod_{t=1}^{V} \Phi_{z,t}^{n^t_{z,\neg d}+\beta-1} \prod_{w\in d} \Phi_{z,w}^{n_d^w} d\Phi_z \\
& = \frac{\Delta(\vec{n}_z+\beta)}{\Delta(\vec{n}_{z,\neg d}+\beta)} \\
& = \frac{\prod_{t=1}^{V}\Gamma(n_z^t+\beta)}{\Gamma(\sum_{t=1}^{V}(n_z^t+\beta))} \frac{\Gamma(\sum_{t=1}^{V}(n^t_{z,\neg d}+\beta))}{\prod_{t=1}^{V}\Gamma(n^t_{z,\neg d}+\beta)} \\
& = \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n^w_{z,\neg d}+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\beta+i-1)}
\end{aligned}
$$

sober_qian

基于DPMM的文档聚类(案例)



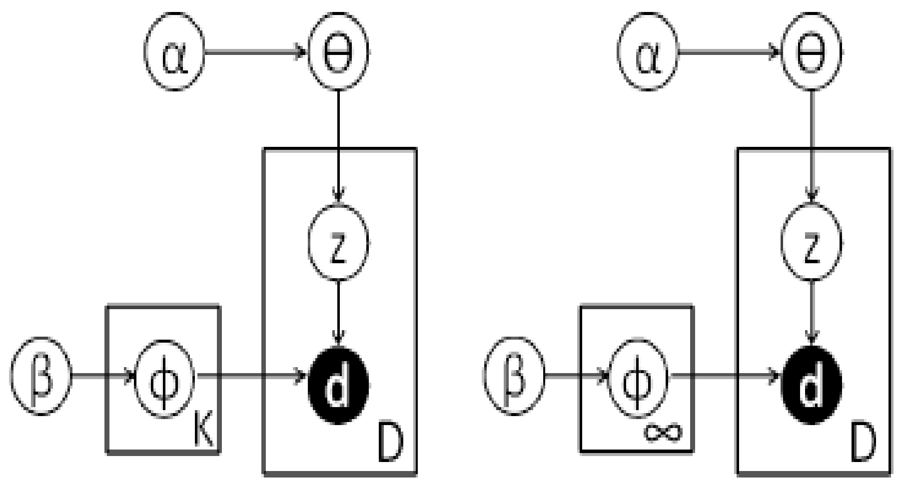
(a) DMM (b) DPMM

生成文档的条件概率(新簇的情况下)：

$$
\begin{aligned}
& p(d|z_d = K+1, \beta) \\
&= \int p(d, \Phi_{K+1}|z_d = K+1, \beta) d\Phi_{K+1} \\
&= \int p(\Phi_{K+1}|z_d = K+1, \beta) p(d|\Phi_{K+1}, z_d = K+1, \beta) d\Phi_{K+1} \\
&= \int p(\Phi_{K+1}|\beta) p(d|\Phi_{K+1}, z_d = K+1) d\Phi_{K+1} \\
&= \int Dir(\Phi_{K+1}|\beta) \prod_{w\in d} Mult(w|\Phi_{K+1}) d\Phi_{K+1} \\
&= \int \frac{1}{\Delta(\beta)} \prod_{t=1}^{V} \Phi_{K+1,t}^{\beta-1} \prod_{w\in d} \Phi_{K+1,w}^{N_d^w} d\Phi_{K+1} \\
&= \frac{\Delta(\vec{n}_{K+1}+\beta)}{\Delta(\beta)} \\
&= \frac{\prod_{t=1}^{V} \Gamma(n_{K+1}^t + \beta)}{\Gamma(\sum_{t=1}^{V}(n_{K+1}^t + \beta))} \frac{\Gamma(\sum_{t=1}^{V} \beta)}{\prod_{t=1}^{V} \Gamma(\beta)} \\
&= \frac{\prod_{w\in d} \prod_{j=1}^{N_d^w} (\beta + j - 1)}{\prod_{i=1}^{N_d} (V\beta + i - 1)}
\end{aligned}
$$

sober_qian

基于DPMM的文档聚类(案例)



(a) DMM　　(b) DPMM

因此：

$$p(z_d = z|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta) \propto \frac{m_{z,\neg d}}{D-1+\alpha} \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w+\beta+j-1)}{\prod_{i=1}^{N_d}(n_{z,\neg d}+V\beta+i-1)}$$

$$p(z_d = K+1|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta) \propto \frac{\alpha}{D-1+\alpha} \frac{\prod_{w\in d}\prod_{j=1}^{N_d^w}(\beta+j-1)}{\prod_{i=1}^{N_d}(V\beta+i-1)}$$

sober_qian

基于DPMM的文档聚类(案例)

```java
//对文档的每个单词进行计算
private int sampleCluster(int d, Document document)
{
    double[] prob = new double[K+1];
    //计算属于已有簇的概率
    for(int k = 0; k < K; k++){
        //第一项
        prob[k] = (m_z[k]) / (D - 1 + alpha);
        double valueOfRule2 = 1.0;
        int i = 0;
        //计算连乘积
        for(int w=0; w < document.wordNum; w++){
            int wordNo = document.wordIdArray[w];
            int wordFre = document.wordFreArray[w];
            //依据公式进行计算
            for(int j = 0; j < wordFre; j++){

                valueOfRule2 *= (n_zv[k][wordNo] + beta + j) / (n_z[k] + V*beta + i);
                i++;
            }
        }
        prob[k] = prob[k] * valueOfRule2 ;
    }
```

核心代码

```java
    //计算属于新簇的概率
    prob[K]= (alpha) / (D - 1 + alpha);
    double valueOfRule3 = 1.0;
    int i = 0;
    //这里可以进行近似计算的
    for(int w=0; w < document.wordNum; w++){
        int wordFre = document.wordFreArray[w];
        for(int j = 0; j < wordFre; j++){
            valueOfRule3 *= (beta + j) /( beta*V + i);
            i++;
        }

    }
    prob[K] = prob[K] * valueOfRule3 ;
    //基于轮盘赌选择是已有的簇还是旧的簇
    for(int k = 1; k < K+1; k++){
        prob[k] += prob[k - 1];
    }

    double thred = Math.random() * prob[K];
    int kChoosed;
    for(kChoosed = 0; kChoosed < K+1; kChoosed++){
        if(thred < prob[kChoosed]){
            break;
        }
    }

    return kChoosed;
}
```

核心代码

sober_dian

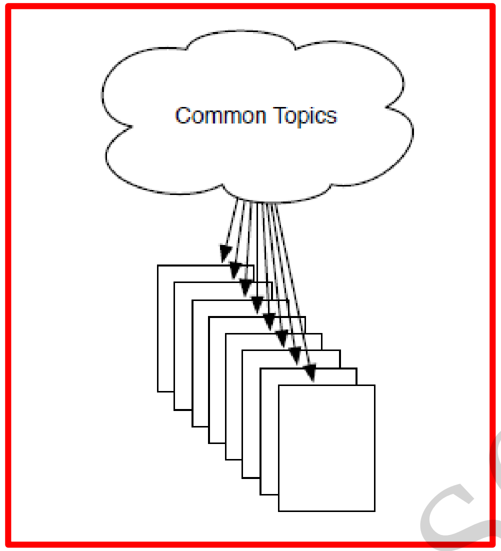
主要内容

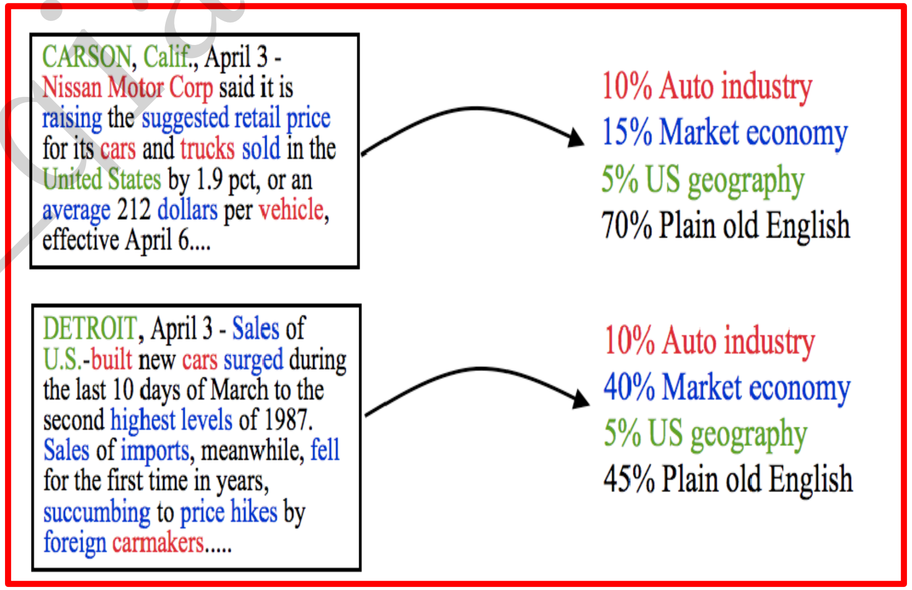
- DP应用场景
- DP的定义及构造
- DPMM模型
- HDP模型

sober_qian

HDP

- DP可以实现一组数据的聚类和分析（类似于文档聚类，一个文档相当于一个数据）
- 在研究多组数据的聚类问题时，单纯利用DP混合模型是无法实现建模分析的.

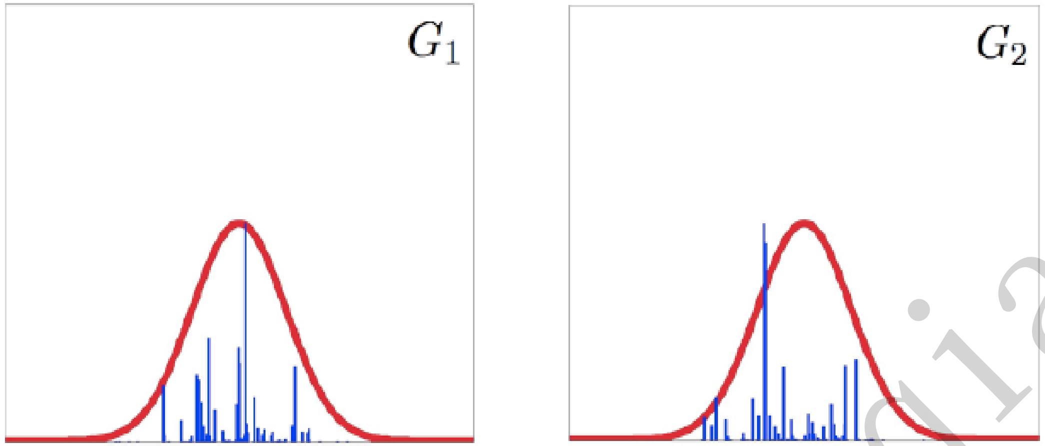




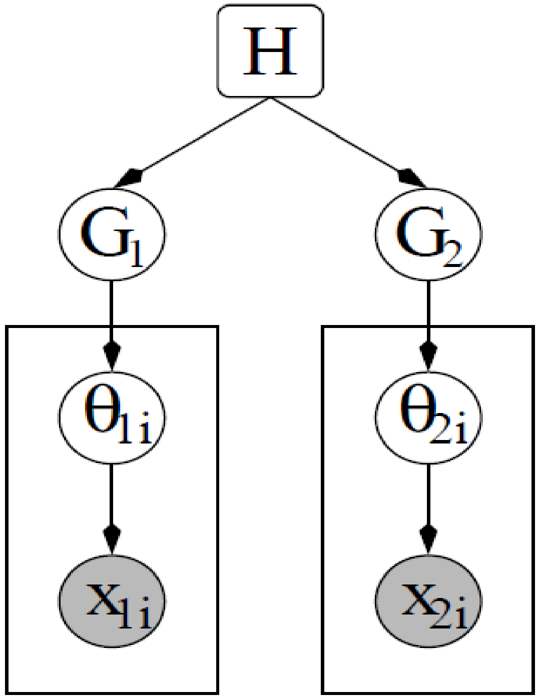
sober-qian

HDP

- DP为每一组数据聚类



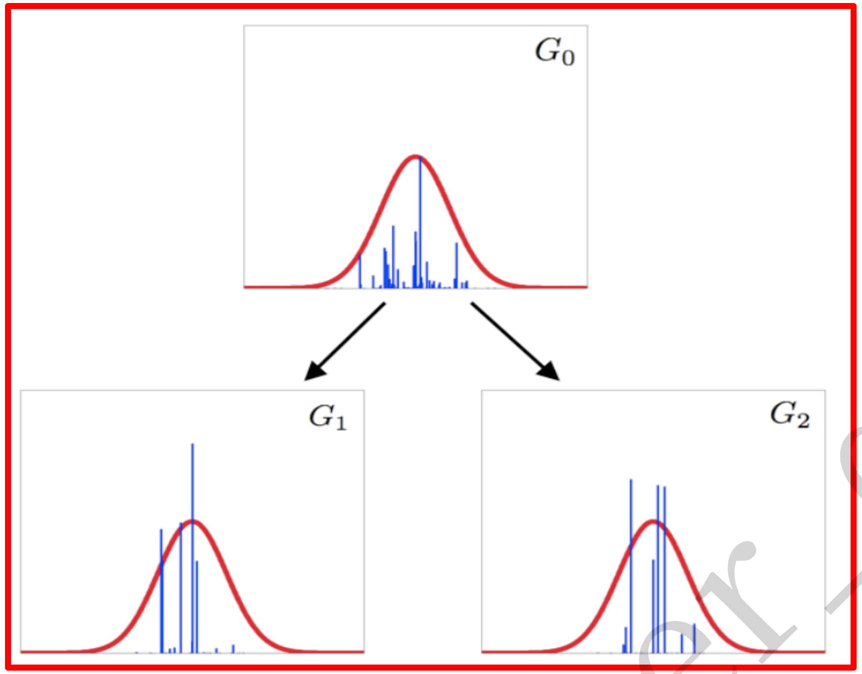
- H是连续分布，组与组间没办法共享聚类结果。
- 解决办法：构造H，使其离散化。



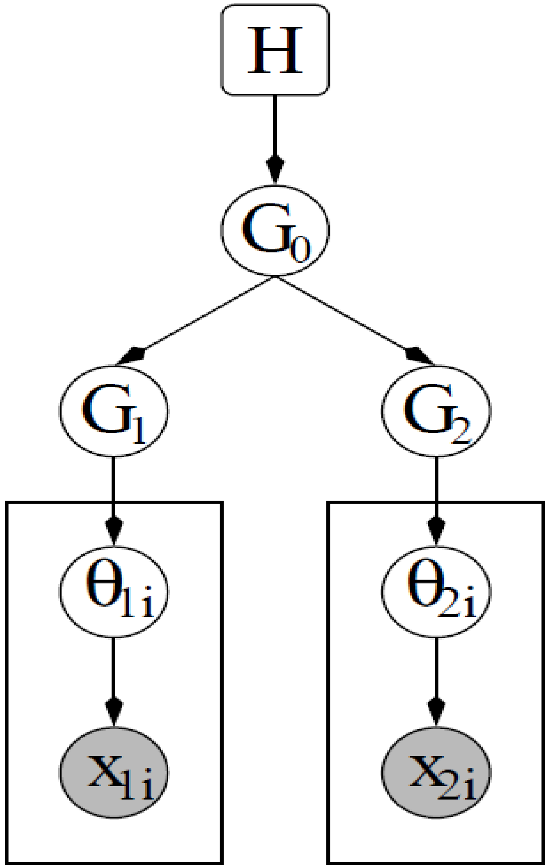
sohei-qian

HDP

➢ 在共同的基分布上引入一个DP先验



$$G_0 \sim \mathrm{DP}(\alpha_0, H)$$
$$G_1, G_2 | G_0 \sim \mathrm{DP}(\alpha, G_0)$$



sober_qian

HDP

- 各文档的主题均是服从基分布H分布(保障各文档之间的主题共享)
- 以基分布H和集中度参数γ，构造DP

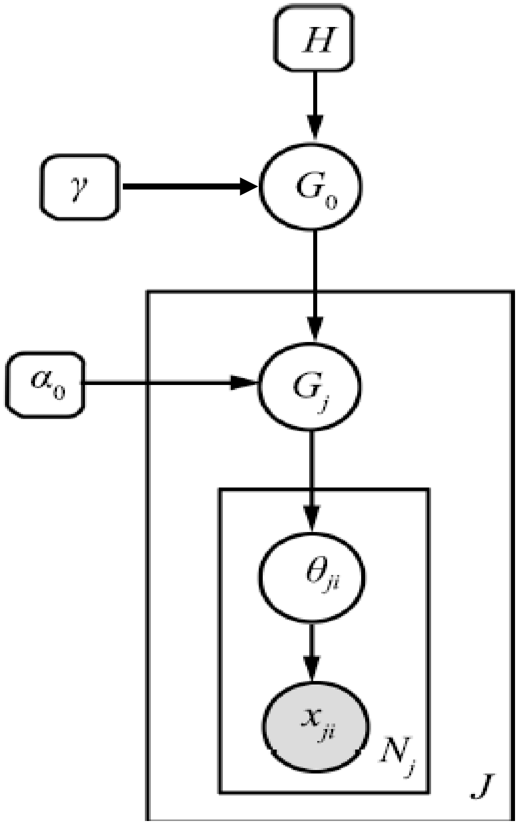
$$G_0 \sim \mathrm{DP}(\gamma, H)$$

- 以G_0为基分布和集中度参数α_0，对每一组数据构造DP

$$G_j | G_0 \sim \mathrm{DP}(\alpha, G_0) \quad \text{for } j = 1, \ldots, J$$

- 以G_j为先验分布，构造DP混合模型

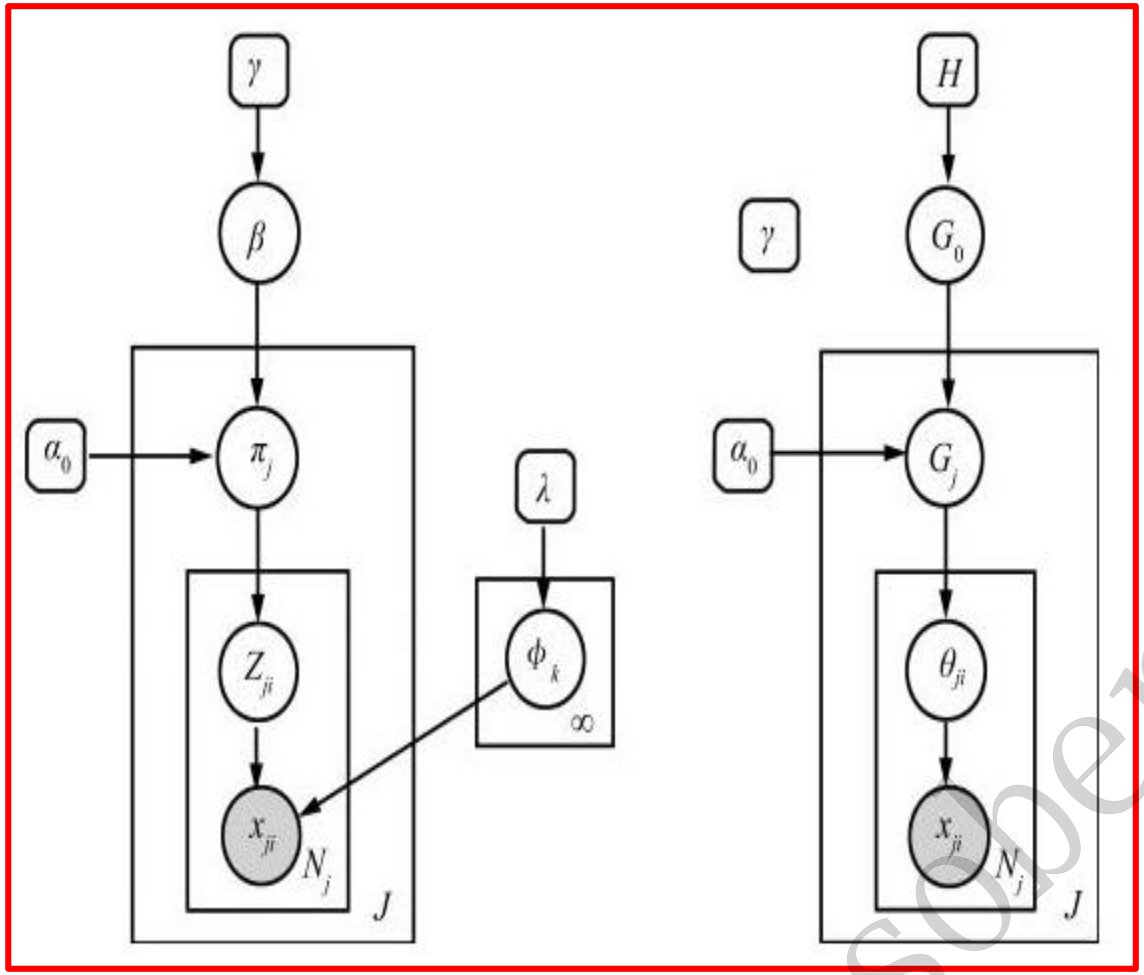
$$\theta_{ji} | G_j \sim G_j, \quad x_{ji} | \theta_{ji} \sim F(\theta_{ji})$$



HDP的有向图表示

sober_qian

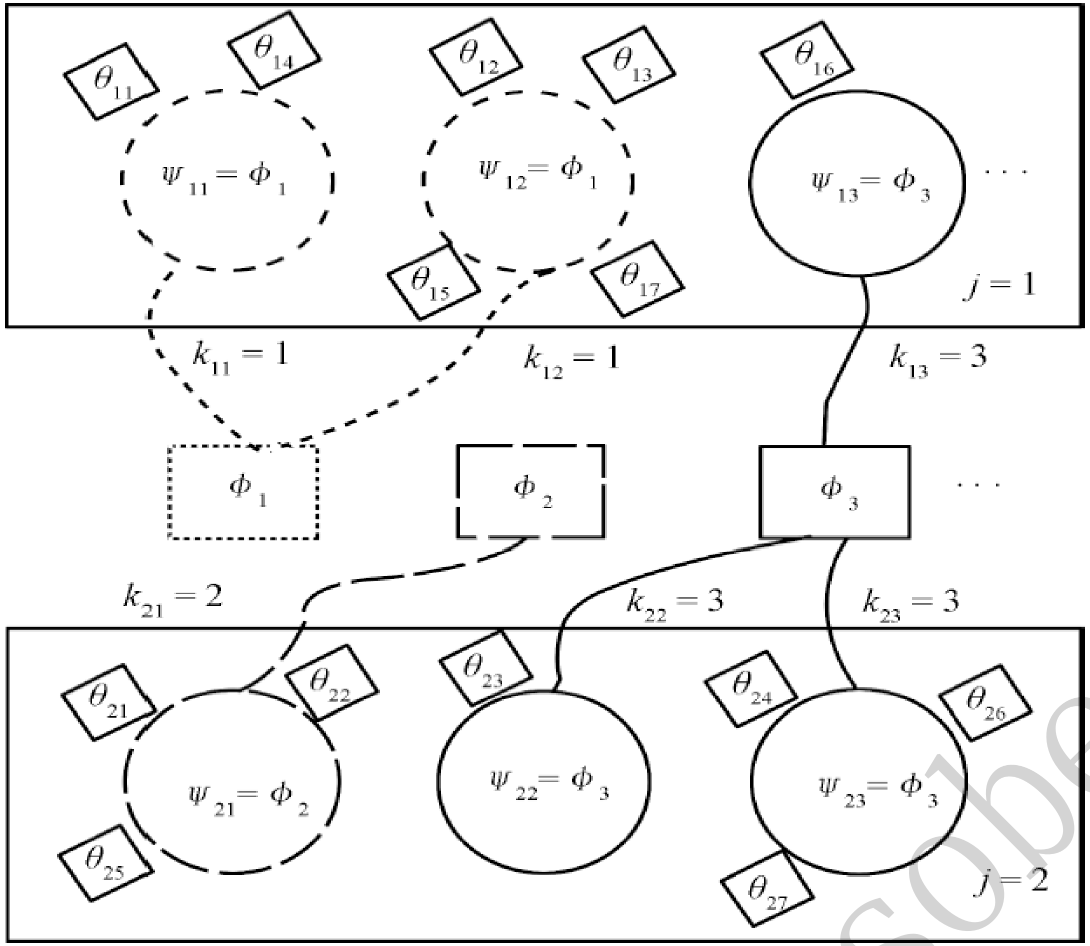
HDP的stick-breaking构造



$$\boldsymbol{\beta}|\gamma \sim \text{GEM}(\gamma), \quad \boldsymbol{\pi}_j|\alpha_0, \quad \boldsymbol{\beta} \sim \text{DP}(\alpha_0, \boldsymbol{\beta})$$
$$z_{ji}|\boldsymbol{\pi}_j \sim \boldsymbol{\pi}_j, \quad \phi_k \sim H(\lambda)$$
$$x_{ji}|z_{ji}, \quad (\phi_k)_{k=1}^{\infty} \sim F(\phi_{z_{ji}})$$

sober_qian

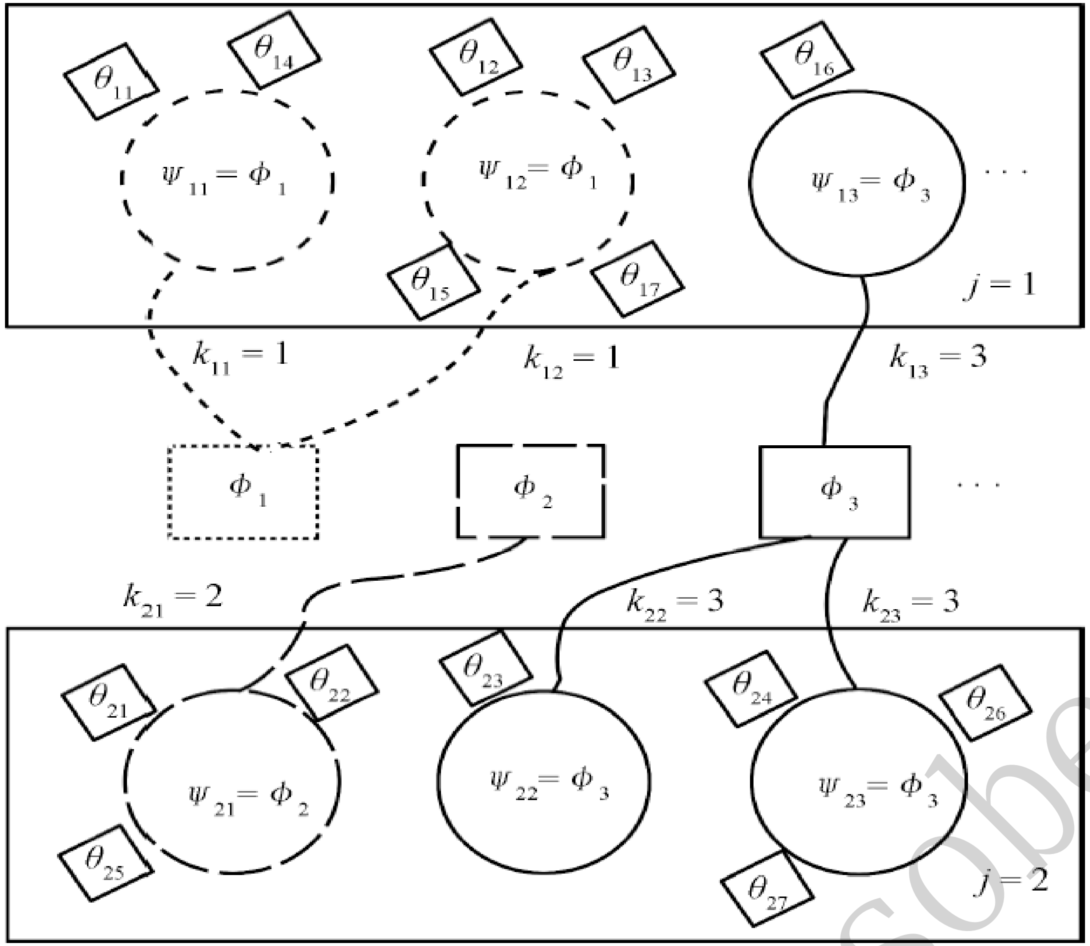
Chinese restaurant franchise构造



- 餐厅是连锁餐厅，所有餐厅共用一份菜单，菜的种类有无穷多个。
- 每个餐厅可拥有无限个桌子，每张餐桌可容纳无穷多位顾客。
- 第一位顾客就座第一张餐桌，每一张餐桌上的第一位客人负责点菜，一张餐桌只有一道菜，其他后来就座于该餐桌的客人共同享用该道菜。
- 不同餐厅的不同餐桌可以点用同一道菜，同一餐厅的不同餐桌也可点用同一道菜。

sober_qian

Chinese restaurant franchise构造



- CRP构造即是为顾客分配餐桌和菜的过程。
- 首先为每位顾客分配餐桌(可视为文档中单词聚类的过程，一层DP)，顾客就坐于哪张餐桌与该餐桌的顾客数成正比；也可以选择一张新桌子。
- 分配完餐桌后，为每张餐桌点菜，每道菜被得到的概率与已点到用这道菜的桌子数成正比；也可以点一道新菜。

sober_qian

基于CRP的后验采样算法(两层)

➢ 第一层：为每位顾客分配餐桌

$$p(t_{ji}=t|,\mathcal{T}^{\backslash ji},\mathcal{K})\propto\begin{cases}n_{jt.}^{\backslash ji}f_{k_{jt}}^{\backslash x_{ji}}(x_{ji}), & t \text{ 为已有顾客就座的餐桌}\\ \alpha_0 p(x_{ji}|\mathcal{T}^{\backslash ji},t_{ji}=t^{\text{new}},\mathcal{K}), & t=t^{\text{new}}\end{cases}$$

其中，t_{ji}表示第j个餐厅的第i个顾客就坐的餐桌，$n_{jt.}$表示第j个餐厅就坐于第t个餐桌上的顾客总数。

$$f_k^{\backslash x_{ji}}(x_{ji})=\frac{\int f(x_{ji}|\phi_k)\prod_{j'i'\neq ji,z_{j'i'}=k}f(x_{j'i'}|\phi_k)h(\phi_k)\mathrm{d}\phi_k}{\int\prod_{j'i'\neq ji,z_{j'i'}=k}f(x_{j'i'}|\phi_k)h(\phi_k)\mathrm{d}\phi_k}$$

→ 观测数据的条件概率

sober_qian

基于CRP的后验采样算法(两层)

- 第一层：为每位顾客分配餐桌

$$p(t_{ji}=t|,\mathcal{T}^{\backslash ji},\mathcal{K})\propto \begin{cases} n_{jt.}^{\backslash ji} f_{k_{jt}}^{\backslash x_{ji}}(x_{ji}), & t \text{ 为已有顾客就座的餐桌} \\ \alpha_0 p(x_{ji}|\mathcal{T}^{\backslash ji},t_{ji}=t^{\text{new}},\mathcal{K}), & t=t^{\text{new}} \end{cases}$$

其中，t_{ji}表示第j个餐厅的第i个顾客就坐的餐桌，$n_{jt.}$表示第j个餐厅就坐于第t个餐桌上的顾客总数。

$$p(x_{ji}|\mathcal{T}^{\backslash ji},\ t_{ji}=t^{\text{new}},\ \mathcal{K})=\sum_{k=1}^{K}\frac{m_{.k}}{m_{..}+\gamma}f_k^{\backslash x_{ji}}(x_{ji})+\frac{\gamma}{m_{..}+\gamma}f_{k^{\text{new}}}^{\backslash x_{ji}}(x_{ji})$$

等号右边第1项是新的餐桌点用已有顾客点过的菜之概率和；
第2项是该新的餐桌点一道新的菜的概率.

sober_qian

基于CRP的后验采样算法(两层)

➢ 第二层：对每个餐厅中的餐桌分配菜(只有是新桌子时执行)：

$$p(k_{jt} = k | \mathcal{K}^{\backslash jt}, \mathcal{T}) \propto \begin{cases} m_{.k}^{\backslash jt} f_k^{\backslash \mathcal{X}_{jt}}(\mathcal{X}_{jt}), & k \text{ 为已有顾客点用的菜} \\ \gamma f_{k^{\text{new}}}^{\backslash \mathcal{X}_{jt}}(\mathcal{X}_{jt}), & k = k^{\text{new}} \end{cases}$$

$m_{.k}$为所有餐厅里点了第k道菜的桌子总数。

sober_qian

HDP采样核心代码解读

```java
/**
 * 一步一步向前执行Gibbs Sampling
 */
public void nextGibbsSweep() {
    int table;
    //对每篇文档,每个单词循环
    for (int d = 0, len = docStates.length; d < len; d++) {
        for (int i = 0; i < docStates[d].docLen; i++) {
            removeWord(d, i); // remove the word i from the state
            table = sampleTable(d, i);
            //如果是新桌子去抽桌子的主题
            if (table == docStates[d].tablesNum) // new Table
                addWord(d, i, table, sampleTopic()); // sampling its Topic
            else
                addWord(d, i, table, docStates[d].tableToTopic[table]); // existing Table
        }
    }
    defragment();
}
```

针对每篇文档的每个单词进行桌子和主题分配(两层for循环)

移除该单词，并统计该单词对应的桌子上的词的数量-1;该单词对应的主题生成的总单词数量-1;该单词对应的主题生成的该单词的数量减1，并判断是否移除该桌子

sober_qian

HDP采样核心代码解读

```
int sampleTable(int docID, int i) {
    int k, j;
    double pSum = 0.0, vb = V * eta, fNew, u;
    DOCState docState = docStates[docID];
    f = ensureCapacity(f, K);
    p = ensureCapacity(p, docState.tablesNum);
    //这里是gamma,
    fNew = gamma.getValue() / V;
    //计算fNew
    for (k = 0; k < K; k++) {
        //计算f值
        f[k] = (phi[k][docState.words[i].termIndex] + eta) /
                (wordNumByTopic[k] + vb);
        //计算fNew的前半部分
        fNew += tablesNumByTopic[k] * f[k];
    }
    for (j = 0; j < docState.tablesNum; j++) {
        if (docState.wordCountByTable[j] > 0)
            //桌子对应的单词数，这里计算的旧桌子加和
            pSum += docState.wordCountByTable[j] * f[docState.tableToTopic[j]];
        p[j] = pSum;
    }
    //加上新桌子的概率
    pSum += alpha.getValue() * fNew / (totalTablesNum + gamma.getValue()); // Probability for t = 
    //轮盘赌
    p[docState.tablesNum] = pSum;
    u = random.nextDouble() * pSum;
    for (j = 0; j < docState.tablesNum; j++)
        if (u < p[j])
            break;  // decided which table the word i is assigned to
    return j;
}
```

sober_qian

HDP采样核心代码解读

```
/**
 *
 * 桌子的主题抽样，决定将桌子赋予哪个主题
 * 语料层的抽样
 * @return the index of the topic
 */
private int sampleTopic() {
    double r, pSum = 0.0;
    int k;
    p = ensureCapacity(p, K);
    //抽主题公式
    for (k = 0; k < K; k++) {
        pSum += tablesNumByTopic[k] * f[k];
        p[k] = pSum;
    }
    //加上新主题的
    pSum += gamma.getValue() / V;
    //轮盘赌
    p[K] = pSum;
    r = random.nextDouble() * pSum;
    for (k = 0; k < K; k++)
        if (r < p[k])
            break;
    return k;
}
```

sober_qian

相关学习资料

Neal R M. Markov chain sampling methods for Dirichlet process mixture models[J]. Journal of computational and graphical statistics, 2000, 9(2): 249-265.

Teh Y W, Jordan M I, Beal M J, et al. Sharing clusters among related groups: Hierarchical Dirichlet processes[C]//Advances in neural information processing systems. 2005: 1385-1392.

Teh Y W. Dirichlet process[M]//Encyclopedia of machine learning. Springer US, 2011: 280-287.

Görür D, Rasmussen C E. Dirichlet process gaussian mixture models: Choice of the base distribution[J]. Journal of Computer Science and Technology, 2010, 25(4): 653-664.

Murugiah S. Bayesian nonparametric clustering based on Dirichlet processes[D]. UCL (University College London), 2010.

周建英, 王飞跃, 曾大军. 分层 Dirichlet 过程及其应用综述[J]. 自动化学报, 2011, 37(4): 389-407.

sober_qian

END!