潜在狄利克雷分配

1 潜在狄利克雷分配

1.1 狄利克雷分布

若多元连续随机变量 $\theta = (\theta_1, \theta_2, ..., \theta_k)$ 的概率密度函数为

$$f(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

其中 $\sum_{i=1}^{k} \theta_i = 1, \theta_i \geq 0, \alpha = (\alpha_1, \alpha_2, ..., \alpha_k), \alpha_i > 0, i = 1, 2, ..., k$，则称随机变量 θ 服从参数为 α 的狄利克雷分布，记作 $\theta \sim Dir(\alpha)$。

令

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}$$

$B(\alpha)$ 是规范化因子，称为多元贝塔函数。可将狄利克雷分布的概率密度函数记为

$$f(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}$$

假设随机变量 X 服从多项分布 $X \sim Multi(n, \theta), n = (n_1, n_2, ..., n_k), \theta = (\theta_1, \theta_2, ..., \theta_k)$，则其概率密度函数为

$$f(X|\theta) = \theta_1^{n_1} \theta_2^{n_2} ... \theta_k^{n_k} = \prod_{i=1}^{k} \theta_i^{n_i}$$

其中 X 的参数 θ 满足的先验分布为狄利克雷分布 $f(\theta|\alpha)$，参数为 $\alpha = (\alpha_1, \alpha_2, ..., \alpha_k)$。因此参数 θ 的后验分布为：

$$
\begin{aligned}
f(\theta|X,\alpha) &= \frac{f(X|\theta)f(\theta|\alpha)}{f(X|\alpha)} \\
&= \frac{\prod_{i=1}^{k}\theta_i^{n_i}\frac{1}{B(\alpha)}\theta_i^{\alpha_i-1}}{\int\prod_{i=1}^{k}\theta_i^{n_i}\frac{1}{B(\alpha)}\theta_i^{\alpha_i-1}d\theta} \\
&= \frac{1}{B(\alpha+n)}\prod_{i=1}^{k}\theta_i^{\alpha_i+n_i-1} \\
&= Dir(\theta|\alpha+n)
\end{aligned}
$$

因此，如果多项分布的先验分布是狄利克雷分布，则其后验分布也是狄利克雷分布。称狄利克雷分布是多项分布的**共轭先验**。

1.2 模型定义

潜在狄利克雷分配使用三个集合：一是**单词集合**$W=\{w_1,w_2,...,w_V\},V$ 是单词的个数。二是**文本集合**$D=\{d_1,d_2,...,d_M\},M$ 是文本数量；文本 $d_m=(w_{m1},w_{m2},...,w_{mN_m})$ 是一个单词序列，N_m 是文本 d_m 中的单词个数。三是**话题集合**$Z=\{z_1,z_2,...,z_K\},K$ 是话题个数。

由话题 z_k 生成单词是由其条件分布 $f(w|z_k)$ 决定，服从多项分布，参数为 $\phi_k=(\phi_{k1},\phi_{k2},...,\phi_{kV})$，该参数服从一个超参数为 β 的狄利克雷分布。所有话题的参数向量构成一个 $K\times V$ 矩阵 $\phi=\{\phi_k\}_{k=1}^{K}$，超参数 β 也是一个 V 维向量 $\beta=(\beta_1,\beta_2,...,\beta_V)$。

由话题生成文本 d_m 是由其条件分布 $f(z|d_m)$ 决定，服从多项分布，参数为 $\theta_m=(\theta_{m1},\theta_{m2},...,\theta_{mK})$，该参数服从一个超参数为 α 的狄利克雷分布。所有话题的参数向量构成一个 $M\times K$ 矩阵 $\theta=\{\theta_m\}_{m=1}^{M}$，超参数 α 也是一个 K 维向量 $\alpha=(\alpha_1,\alpha_2,...,\alpha_K)$。

因此，每一个文本 d_m 中的每一个单词 w_{mn} 由该文本的话题分布 $f(z|d_m)$ 和所有话题的单词分布 $f(w|z_k)$ 决定。潜在狄利克雷分配的生成算法如下：

输入：单词集合 W，文本集合 D，话题集合 W 和狄利克雷分布的超参数 α,β。

1. 生成话题的单词分布：对于话题 $z_k,k=1,2,...,K$，生成多项分布的参数 $\phi_k\sim Dir(\beta)$，作为话题的单词分布 $f(w|z_k)$
2. 生成文本的话题分布：对于文本 $d_m,m=1,2,...,M$，生成多项分布的参数 $\theta_m\sim Dir(\alpha)$，作为文本的话题分布 $f(z|d_m)$
3. 按照多项分布 $Multi(\theta_m)$ 随机生成一个话题 $z_{mn}\sim Multi(\theta_m),m=1,2,...,M;n=1,2,...,N_m$
4. 按照多项分布 $Multi(\phi_{z_{mn}})$ 随机生成一个单词 $w_{mn}\sim Multi(\phi_{z_{mn}}),m=1,2,...,M;n=1,2,...,N_m$

输出：生成的文本 $\{d_i=\{w_{m1},w_{m2},...,w_{mN_m}\}\}_{m=1}^{M}$。

以概率图模型的视角，LDA 的图模型为 $\alpha \to \theta_m \to z_{mn} \to w_{mn} \leftarrow \phi_k \leftarrow \beta$。LDA 模型整体是由观测变量和隐变量组成的联合概率分布为

$$f(d,z,\theta,\phi|\alpha,\beta)=\prod_{k=1}^{K} f(\phi_k|\beta)\prod_{m=1}^{M} f(\theta_m|\alpha)\prod_{n=1}^{N_m} f(z_{mn}|\theta_m)f(w_{mn}|z_{mn},\phi)$$

其中，第 m 个文本的联合概率分布为

$$f(d_m,z_m,\theta_m,\phi|\alpha,\beta)=\prod_{k=1}^{K} f(\phi_k|\beta)f(\theta_m|\alpha)\prod_{n=1}^{N_m} f(z_{mn}|\theta_m)f(w_{mn}|z_{mn},\phi)$$

为得到关于 d 的概率分布，先求 d_m 关于 θ_m,ϕ 的分布：

$$f(d_m|\theta_m,\phi)=\prod_{n=1}^{N_m}\left[\sum_{k=1}^{K} P(z_{mn}=k|\theta_m)f(w_{mn}|\phi_k)\right]$$

所以，超参数 α,β 给定下第 m 个文本的生成概率为：

$$f(d_m|\alpha,\beta)=\prod_{k=1}^{K}\int f(\phi_k|\beta)\left[\int f(\theta_m|\alpha)\prod_{n=1}^{N_m}\left[\sum_{k=1}^{K} P(z_{mn}=k|\theta_m)f(w_{mn}|\phi_k)\right]d\theta_m\right]d\phi_k$$

所以，超参数 α,β 给定下所有文本的生成概率为：

$$f(d|\alpha,\beta)=\prod_{k=1}^{K}\int f(\phi_k|\beta)\left[\prod_{m=1}^{M}\int f(\theta_m|\alpha)\prod_{n=1}^{N_m}\left[\sum_{k=1}^{K} P(z_{mn}=k|\theta_m)f(w_{mn}|\phi_k)\right]d\theta_m\right]d\phi_k$$

1.3 LDA 的吉布斯抽样算法

首先介绍**吉布斯抽样**。吉布斯抽样是马尔可夫链蒙特卡洛方法 (MCMC) 中的一种方法，其可用于多元联合分布的抽样和估计。该算法如下：

输入：目标概率分布的密度函数 $p(x)$，函数 $f(x)$，收敛步数 m 和迭代步数 n。

1. 初始化：给出初始样本 $x^{(0)}=\{x_1^{(0)},x_2^{(0)},...,x_k^{(0)}\}^T$
2. 对 i 循环执行。此时，第 $i-1$ 次迭代结束后的样本为 $x^{(i-1)}=\{x_1^{(i-1)},x_2^{(i-1)},...,x_k^{(i-1)}\}^T$，不断执行以下操作：

- 从分布 $p(x_1|x_2^{(i-1)},...,x_k^{(i-1)})$ 抽取 $x_1^{(i)}$
- ...
- 从分布 $p(x_j|x_1^{(i)},...,x_{j-1}^{(i)},x_{j+1}^{(i-1)}...,x_k^{(i-1)})$ 抽取 $x_j^{(i)}$

- ...
- 从分布 $p(x_k|x_1^{(i)},...,x_{k-1}^{(i)})$ 抽取 $x_k^{(i)}$

3. 得到样本集合 $\{x^{(m+1)},x^{m+2},...,x^{(n)}\}$
4. 计算结果 $f_{mn}=\frac{1}{n-m}\sum_{i=m+1}^{n}f(x^{(i)})$

输出：估计结果 f_{mn}。

记文本中的单词总集合为 $w=(w_{11},w_{12},...,w_{1N_1},w_{21},w_{22},...,w_{2N_2},...,w_{M1},w_{M2},...,w_{MN_M})$。话题集合是 $z=(z_1,z_2,...,z_M),z_m=(z_{m1},z_{m2},...,z_{mN_m}),m=1,2,...,M$。文本的话题分布和话题的单词分布参数分别为 $\theta=\{\theta_1,\theta_2,...,\theta_M\}$ 和 $\phi=\{\phi_1,\phi_2,...,\phi_K\}$。在超参数 α,β 已知的情况下，需要对联合概率分布 $p(w,z,\theta,\phi|\alpha,\beta)$ 进行估计，其中 w 是观测变量，z,θ,ϕ 是隐变量。

LDA 模型采用收缩的吉布斯抽样方法，基本思想是：

1. 首先对隐变量 θ,ϕ 积分，得到边缘概率分布 $p(w,z|\alpha,\beta)$
2. 转换为对不可观测的变量 z 的抽样，按后验分布 $p(z|w,\alpha,\beta)$ 进行吉布斯抽样
3. 得到分布 $p(z|w,\alpha,\beta)$ 的样本集合，使用该集合估计参数 θ,ϕ 的估计值

可以发现，对参数 θ,ϕ 的估计主要需要计算后验概率 $p(z|w,\alpha,\beta)$。由于

$$p(z|w,\alpha,\beta)=\frac{p(w,z|\alpha,\beta)}{p(w|\alpha,\beta)}\propto p(w,z|\alpha,\beta)$$

$p(w|\alpha,\beta)$ 中均是已知变量，可以不予考虑，所以可以考虑 $p(w,z|\alpha,\beta)$，而该概率分布可以进一步分解为

$$p(w,z|\alpha,\beta)=p(w|z,\alpha,\beta)p(z|\alpha,\beta)=p(w|z,\beta)p(z|\alpha)$$

对两个因子 $p(w|z,\beta)$ 和 $p(z|\alpha)$ 分别进行处理。对于 $p(w|z,\beta)$，首先

$$p(w|z,\phi)=\prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{kv}^{n_{kv}}$$

式中，ϕ_{kv} 是第 k 个话题生成单词集合中第 v 个单词的概率，n_{kv} 是第 k 个话题生成单词集合中第 v 个单词的次数。于是

$$
\begin{aligned}
p(w|z,\beta) &= \int p(w|z,\phi)p(\phi|\beta)d\phi \\
&= \int \prod_{k=1}^{K}\prod_{v=1}^{V}\phi_{kv}^{n_{kv}}\frac{1}{B(\beta)}\phi_{kv}^{\beta_v-1}d\phi \\
&= \prod_{k=1}^{K}\frac{1}{B(\beta)}\int\prod_{v=1}^{V}\phi_{kv}^{n_{kv}+\beta_v-1}d\phi \\
&= \prod_{k=1}^{K}\frac{B(n_k+\beta)}{B(\beta)}
\end{aligned}
$$

其中 $n_k=\{n_{k1},n_{k2},...,n_{kV}\}$。第二个因子 $p(z|\alpha)$ 也可通过类似的方法进行计算，由于

$$
p(z|\theta)=\prod_{m=1}^{M}\prod_{k=1}^{K}\theta_{mk}^{n_{mk}}
$$

式中，θ_{mk} 是第 m 个文本生成第 k 个话题的概率，n_{kv} 是第 m 个文本生成第 k 个话题的次数。于是

$$
\begin{aligned}
p(z|\alpha) &= \int p(z|\theta)p(\theta|\alpha)d\theta \\
&= \int\prod_{m=1}^{M}\prod_{k=1}^{K}\theta_{mk}^{n_{mk}}\frac{1}{B(\alpha)}\theta_{mk}^{\alpha_k-1}d\theta \\
&= \prod_{m=1}^{M}\frac{1}{B(\alpha)}\int\prod_{k=1}^{K}\theta_{mk}^{n_{mk}+\alpha_k-1}d\theta \\
&= \prod_{m=1}^{M}\frac{B(n_m+\alpha)}{B(\alpha)}
\end{aligned}
$$

其中 $n_m=\{n_{m1},n_{m2},...n_{mK}\}$。联立两式有

$$
p(z,w|\alpha,\beta)=\prod_{k=1}^{K}\frac{B(n_k+\beta)}{B(\beta)}\prod_{m=1}^{M}\frac{B(n_m+\alpha)}{B(\alpha)}
$$

因此，收缩的吉布斯抽样分布的公式为

$$
p(z|w,\alpha,\beta)\propto\prod_{k=1}^{K}\frac{B(n_k+\beta)}{B(\beta)}\prod_{m=1}^{M}\frac{B(n_m+\alpha)}{B(\alpha)}
$$

根据该联合密度函数，可以对 z 进行吉布斯抽样。由于 $p(z|w,\alpha,\beta)$ 是满条件分布，因此该函数的吉布斯抽样分布可以写成

$$
p(z_i|z_{-i},w,\alpha,\beta)=\frac{1}{Z_{z_i}}p(z|w,\alpha,\beta)
$$

其中此处的 i 可以取到所有单词，$z_{-i}=\{z_j:j\neq i\}$，Z_{z_i} 是规范化因子，使左端可以变成一个概率密度函数。由此可以推出：

$$p(z_i|z_{-i},w,\alpha,\beta)\propto\frac{n_{kv}+\beta_v}{\sum_{v=1}^{V}(n_{kv}+\beta_v)}\frac{n_{mk}+\alpha_k}{\sum_{k=1}^{K}(n_{mk}+\alpha_k)}$$

通过该函数的吉布斯抽样，可以得到一系列关于话题 z 的样本，从而估计参数 $\theta=\{\theta_m\}$ 和 $\phi=\{\phi_k\}$。可以由共轭先验，写出 θ,ϕ 的后验分布。

$$p(\theta_m|z_m,\alpha)=\frac{1}{Z_{\theta_m}}\prod_{n=1}^{N_m}p(z_{mn}|\theta_m)p(\theta_m|\alpha)\sim Dir(\theta_m|n_m+\alpha)$$

$$p(\phi_k|z,w,\beta)=\frac{1}{Z_{\phi_k}}\prod_{i=1}^{I}p(w_i|\phi_k)p(\phi_k|\beta)\sim Dir(\phi_k|n_k+\beta)$$

使用极大似然估计，可以得到：

$$\theta_{mk}=\frac{n_{mk}+\alpha_k}{\sum_{k=1}^{K}(n_{mk}+\alpha_k)},m=1,2,...,M;k=1,2,...,K$$

$$\phi_{kv}=\frac{n_{kv}+\beta_v}{\sum_{v=1}^{V}(n_{kv}+\beta_v)},k=1,2,...,K;v=1,2,...,V$$

在算法实现时，需要存储两个矩阵：话题-单词矩阵 $N_{K\times V}=[n_{kv}]$ 和文本-话题矩阵 $N_{M\times K}=[n_{mk}]$。每次在抽样前，需要先将对应位置的话题数减 1，再进行抽样，抽样后按抽样结果在对应位置将话题数加 1。潜在狄利克雷分配的吉布斯抽样算法如下：

输入：单词总集合为 $w=(w_{11},w_{12},...,w_{1N_1},w_{21},w_{22},...,w_{2N_2},...,w_{M1},w_{M2},...,w_{MN_M})$，超参数 α,β 和话题数 K。

1. 将计数矩阵的元素 n_{mk},n_{kv}，计数向量 n_m,n_k 初值置为 0
2. 对所有单词 $w_{mn},m=1,2,...,M;n=1,2,...,N_m$ 进行以下操作

- 抽样话题 $z_{mn}=z_k\sim Multi(\frac{1}{K})$
- 将计数矩阵和计数向量中的对应元素 n_{mk},n_{kv},n_m,n_k 加 1

3. 对所有单词 $w_{mn},m=1,2,...,M;n=1,2,...,N_m$ 进行以下操作，直至进入燃烧期

- 当前的单词 w_{mn} 是第 v 个单词，话题 z_{mn} 是第 k 个话题
- 将计数矩阵和计数向量中的对应元素 n_{mk},n_{kv},n_m,n_k 减 1
- 按满条件分布抽样

$$p(z_i|z_{-i},w,\alpha,\beta)\propto\frac{n_{kv}+\beta_v}{\sum_{v=1}^{V}(n_{kv}+\beta_v)}\frac{n_{mk}+\alpha_k}{\sum_{k=1}^{K}(n_{mk}+\alpha_k)}$$

- 得到新话题 k'，分配给 z_{mn}。将计数矩阵和计数向量中的对应元素 $n_{mk'}, n_{k'v}, n_m, n_{k'}$ 加 1

4. 根据计数矩阵 $N_{K\times V}=[n_{kv}]$ 和 $N_{M\times K}=[n_{mk}]$ 计算参数的值

$$\theta_{mk}=\frac{n_{mk}+\alpha_k}{\sum_{k=1}^{K}(n_{mk}+\alpha_k)}, m=1,2,...,M; k=1,2,...,K$$

$$\phi_{kv}=\frac{n_{kv}+\beta_v}{\sum_{v=1}^{V}(n_{kv}+\beta_v)}, k=1,2,...,K; v=1,2,...,V$$

输出：模型的参数 θ,ϕ。

1.4 LDA 的变分 EM 算法

变分推理的目标是学习模型的后验概率分布 $f(z|x)$。其使用一个概率分布 $q(z)$ 去近似复杂的概率分布 $f(z|x)$。使用 KL 散度 $D(q(z)||f(z|x))$ 计算两个分布之间的相似度，从中找出 KL 散度最小的变分分布 $q^*(z)$ 去近似 $f(z|x)$，即 $q^*(z)\approx f(z|x)$。KL 散度可以写成如下形式：

$$\begin{aligned}D(q(z)||f(z|x)) &= E_q[\log q(z)]-E_q[\log f(z|x)]\\ &= E_q[\log q(z)]-E_q[\log f(x,z)]+E_q[\log p(x)]\\ &= \log p(x)-\{E_q[\log f(x,z)]-E_q[\log q(z)]\}\end{aligned}$$

注意到 KL 散度大于等于 0，因此有

$$\log p(x)\ge E_q[\log f(x,z)]-E_q[\log q(z)]$$

不等式左端称为*证据*，右端称为*证据下界*。将证据下界记为 $L(q)=E_q[\log f(x,z)]-E_q[\log q(z)]$。KL 散度的最小化等价于证据下界 $L(q)$ 的最大化。

此外，为了防止变分分布 $q(z)$ 的搜索范围过大，致使出现不可计算问题。变分分布 $q(z)$ 需要定义在**平均场**上，其对 z 的所有变量都是相互独立的，即 $q(z)=q(z_1)q(z_2)...q(z_n)$。因此，变分推理主要有以下几个步骤：

1. 定义变分分布 $q(z)$
2. 推导其证据下界表达式
3. 使用最优化方法对证据下界进行优化，得到最优分布 $q^*(z)$，作为 $p(z|x)$ 的近似

其中，最优化方法可以选用 EM 算法，得到**变分 EM 算法**。假设模型的概率分布是 $p(x,z|\theta)$, x 是观测变量，z 是隐变量，θ 是参数。导入平均场 $q(z)=\prod_{i=1}^{n}q(z_i)$，则可定义证据下界

$$L(q,\theta)=E_q[\log p(x,z|\theta)]-E_q[\log q(z)]$$

变分 EM 算法分别对 q,θ 进行迭代以求证据下界的最大值，其步骤如下：

1. E 步：固定 θ，求 $L(q,\theta)$ 关于 q 的最大化
2. M 步：固定 q，求 $L(q,\theta)$ 关于 θ 的最大化

将变分 EM 算法用于 LDA 模型中时，可以如下引入证据下界。简单起见，每次只考虑一个文本。文本的单词序列为 $w=(w_1,w_2,...,w_N)$，对应的话题序列为 $z=(z_1,z_2,...,z_N)$，话题分布的参数为 θ，其联合分布为：

$$p(\theta,w,z|\alpha,\phi)=p(\theta|\alpha)\prod_{n=1}^{N}p(z_n|\theta)p(w_n|z_n,\phi)$$

由于 θ,z 是隐变量，可定义平均场 $q(\theta,z|\gamma,\eta)=q(\theta|\gamma)\prod_{n=1}^{N}q(z_n|\eta_n)$。其中 γ 是 θ 服从的狄利克雷分布的参数，$\eta=(\eta_1,\eta_2,...,\eta_N)$ 是 $z_1,z_2,...,z_N$ 服从的多项分布的参数。因此，文本的证据下界为

$$L(\gamma,\eta,\alpha,\phi)=E_q[\log p(\theta,w,z|\alpha,\phi)]-E_q[\log q(\theta,z|\gamma,\eta)]$$

其中，γ,η 是变分分布的参数，α,ϕ 是 LDA 模型的参数。通过对函数 $L(\gamma,\eta,\alpha,\phi)$ 最大化即可求得变分分布的参数和 LDA 模型的参数。变分分布的参数优化方法是

$$\eta_{nk}\propto\phi_{kv}\exp\left(\Psi(\gamma_k)-\Psi\left(\sum_{l=1}^{K}\gamma_l\right)\right)$$

$$\gamma_k=\alpha_k+\sum_{n=1}^{N}\eta_{nk}$$

其中 $\Psi(.)$ 为对数伽马函数的导函数，即 $\frac{\partial}{\partial x}\log\Gamma(x)=\Psi(x)$。在得到 η_{nk} 后，需要做放缩使得 $\sum_{k=1}^{K}\eta_{nk}=1$。

模型参数的更新方法是

$$\phi_{kv}=\sum_{m=1}^{M}\sum_{n=1}^{N_m}\eta_{mnk}w_{mn}^{v}$$

其中，η_{mnk} 表示第 m 个文本的第 n 个单词属于第 k 个话题的概率；w_{mn}^{v} 当第 m 个文本的第 n 个单词是单词集合中的第 v 个单词时为 1，否则为 0。

α 的更新需要通过牛顿法得到 $\alpha:=\alpha-H^{-1}(\alpha)g(\alpha)$。其中，$g(\alpha)$ 为一阶导，$H(\alpha)$ 为黑塞矩阵，其元素参数如下：

$$\frac{\partial L}{\partial\alpha_k}=M\left[\Psi\left(\sum_{l=1}^{K}\alpha_l\right)-\Psi(\alpha_k)\right]+\sum_{m=1}^{M}\left[\Psi(\gamma_{mk})-\Psi\left(\sum_{l=1}^{K}\gamma_{ml}\right)\right]$$

$$\frac{\partial^2 L}{\partial\alpha_k\partial\alpha_l}=M\left[\Psi'\left(\sum_{l=1}^{K}\alpha_l\right)-I(k=l)\Psi'(\alpha_k)\right]$$

因此，LDA 的变分 EM 算法步骤如下：

输入：单词总集合为 $w=(w_{11},w_{12},...,w_{1N_1},w_{21},w_{22},...,w_{2N_2},...,w_{M1},w_{M2},...,w_{MN_M})$ 和话题个数 K。

1. 初始化变分参数 γ,η 和模型参数 α,ϕ
2. 固定模型参数 α,ϕ，更新变分参数 γ,η

- 更新 $\eta_{nk}:=\phi_{kv}\exp\left(\Psi(\gamma_k)-\Psi\left(\sum_{l=1}^K\gamma_l\right)\right)$
- 规范化使得 $\sum_{k=1}^K\eta_{nk}=1$
- 更新 $\gamma=\alpha+\sum_{n=1}^N\eta_n$
- 重复以上直至收敛

3. 固定变分参数 γ,η，更新模型参数 α,ϕ

- 更新 $\phi_{kv}=\sum_{m=1}^M\sum_{n=1}^{N_m}\eta_{mnk}w_{mn}^v$
- 更新 $\alpha:=\alpha-H^{-1}(\alpha)g(\alpha)$
- 重复以上直至收敛

输出：变分分布的参数 γ,η 和 LDA 模型的参数 α,ϕ。

2 代码实现

本次使用模拟的数据集。数据集中共 100 个文本，每个文本中共 20 个单词，单词集合中共 12 个单词。

```
from sklearn.datasets import make_multilabel_classification
X, _ = make_multilabel_classification(random_state=0)
X
```

```
## array([[3., 1., 4., ..., 4., 1., 3.],
##        [5., 0., 6., ..., 0., 0., 3.],
##        [3., 4., 1., ..., 3., 2., 5.],
##        ...,
##        [2., 1., 2., ..., 1., 0., 3.],
##        [6., 4., 1., ..., 1., 3., 5.],
##        [2., 4., 2., ..., 5., 4., 2.]])
```

2.1 sklearn 实现

潜在狄利克雷分配的 sklearn 接口位于 sklearn.decomposition.LatentDirichletAllocation, 其文档可见**此处**。其中较为重要的参数有：

- n_components：话题个数 K

- doc_topic_prior：α
- topic_word_prior：β

注意，在 sklearn 中，LDA 是使用变分 EM 算法进行参数估计的。

```
from sklearn.decomposition import LatentDirichletAllocation
clf = LatentDirichletAllocation(n_components=5)
clf.fit(X)
```

后两个文本的话题概率分布为：

```
print(clf.transform(X[-2:,:]))
```

```
## [[0.14655242 0.12308133 0.00360182 0.72315697 0.00360747]
##  [0.31658403 0.00360047 0.10941854 0.0039823  0.56641466]]
```

2.2 吉布斯抽样算法

使用如下类实现吉布斯抽样算法的 LDA 模型参数估计：

```
import numpy as np

class GibbsLDA:

    def __init__(self,K,alpha=None,beta=None,sim_time=1000):
        self.K = K
        self.alpha = alpha
        self.beta = beta
        self.sim_time = sim_time

    def fit(self,X):
        M,N = X.shape
        V = len(np.unique(X.reshape(-1)))
        K = self.K

        N_kv = np.zeros((K,V))
        N_mk = np.zeros((M,K))
        if self.alpha is None:
            alpha = [1 for _ in range(K)]
        else:
            alpha = self.alpha
```

```
        if self.beta is None:
            beta = [1 for _ in range(V)]
        else:
            beta = self.beta

        topic_mat = np.zeros((M,N))
        for m in range(M):
            for n in range(N):
                word = X[m,n]
                topic = int(K*np.random.random())
                topic_mat[m,n] = topic
                N_mk[m,topic] += 1
                N_kv[topic,int(word)] +=1

        for _ in range(self.sim_time):
            for m in range(M):
                for n in range(N):
                    word = X[m,n]
                    topic = int(topic_mat[m,n])
                    N_mk[m,topic] -= 1
                    N_kv[topic,int(word)] -= 1

                    prob = (N_kv[:,int(word)]+beta[int(word)])/(np.sum(N_kv+beta,axis=1))\
                          *(N_mk[m,:]+alpha[topic])
                    prob = list(prob/np.sum(prob))
                    topic = list(np.random.multinomial(1,prob)).index(1)

                    topic_mat[m,n] = topic
                    N_mk[m,topic] += 1
                    N_kv[topic,int(word)] += 1

        theta = N_mk+alpha
        self.theta = theta/np.sum(theta,1).reshape(-1,1)
        phi = N_kv+beta
        self.phi = phi/np.sum(phi,1).reshape(-1,1)


    def transform(self):
```

```
        return self.theta
```

后两个文本的话题概率分布为，此处仅优化 100 次：

```
clf = GibbsLDA(K=5,sim_time=100)
clf.fit(X)
print(clf.transform()[-2:])
```

```
## [[0.04 0.16 0.2  0.28 0.32]
##  [0.24 0.36 0.04 0.28 0.08]]
```