主题模型

七月算法 邹博

2015年5月9日

朴素贝叶斯的文本分类

- 给定M封邮件，每个邮件被标记为垃圾邮件或者非垃圾邮件，给出第M+1封非标记邮件是垃圾邮件的概率。
- 朴素贝叶斯的基本假设：
  - 一个特征出现的概率，与其他特征(条件)独立(特征独立性)
    - 其实是：对于给定分类的条件下，特征独立
  - 每个特征同等重要(特征均衡性)

分析

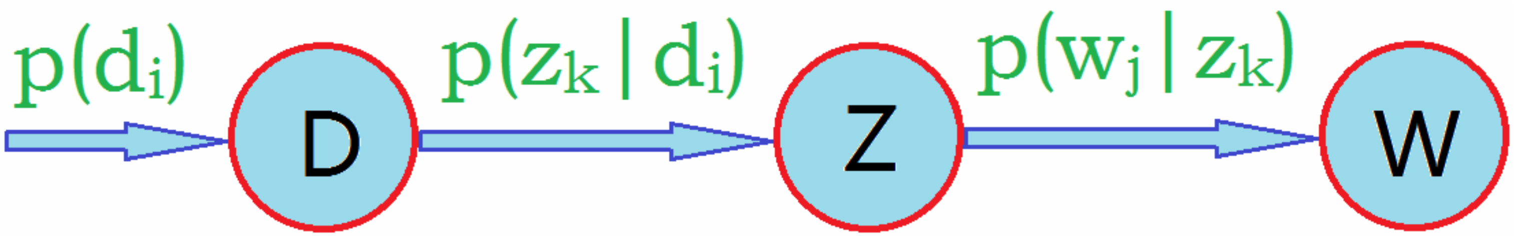
- 类别c：垃圾邮件c_1，非垃圾邮件c_2
- 词汇表，两种建立方法：
  - 使用现成的单词词典；
  - 将所有邮件中出现的单词都统计出来，得到词典。
  - 记单词数目为N
- 将每个邮件m映射成维度为N的向量$\mathbf{x}$
  - 若单词w_i在邮件m中出现过，则$x_i=1$，否则，$x_i=0$。即邮件的向量化：$m \rightarrow (x_1,x_2\ldots\ldots x_N)$
- 贝叶斯公式：$P(c|\mathbf{x})=P(\mathbf{x}|c)*P(c) / P(\mathbf{x})$
  - $P(c_1|\mathbf{x})=P(\mathbf{x}|c_1)*P(c_1) / P(\mathbf{x})$
  - $P(c_2|\mathbf{x})=P(\mathbf{x}|c_2)*P(c_2) / P(\mathbf{x})$
    - 注意这里$\mathbf{x}$是向量

朴素贝叶斯的分析

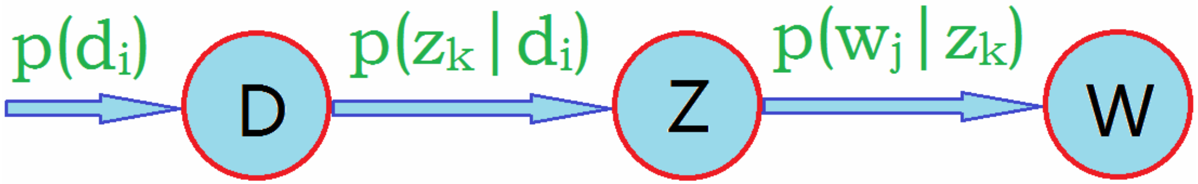
- 可以胜任许多文本分类问题。
- 无法解决语料中一词多义和多词一义的问题——它更像是词法分析，而非语义分析。
- 如果使用词向量作为文档的特征，一词多义和多词一义会造成计算文档间相似度的不准确性。
- 可以通过增加“主题”的方式，一定程度的解决上述问题：
  - 一个词可能被映射到多个主题中
    - ——一词多义
  - 多个词可能被映射到某个主题的概率很高
    - ——多词一义

pLSA模型

- 基于概率统计的pLSA模型(probabilistic latent semantic analysis, 概率隐语义分析)，增加了主题模型，形成简单的贝叶斯网络，可以使用EM算法学习模型参数。

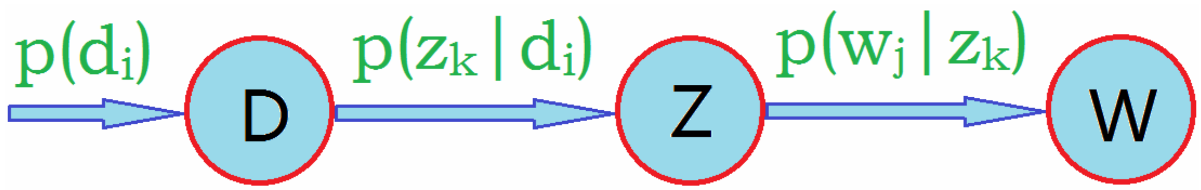


pLSA模型



- D代表文档，Z代表主题(隐含类别)，W代表单词；
  - $P(d_i)$表示文档d_i的出现概率，
  - $P(z_k|d_i)$表示文档d_i中主题z_k的出现概率，
  - $P(w_j|z_k)$表示给定主题z_k出现单词w_j的概率。
- 每个主题在所有词项上服从多项分布，每个文档在所有主题上服从多项分布。
- 整个文档的生成过程是这样的：
  - 以$P(d_i)$的概率选中文档d_i；
  - 以$P(z_k|d_i)$的概率选中主题z_k；
  - 以$P(w_j|z_k)$的概率产生一个单词w_j。

pLSA模型



- 观察数据为(d_i,w_j)对，主题z_k是隐含变量。
- (d_i,w_j)的联合分布为

$$P(d_i,w_j)=P(w_j \mid d_i)P(d_i)$$

$$P(w_j \mid d_i)=\sum_{k=1}^{K} P(w_j \mid z_k)P(z_k \mid d_i)$$

- 而$P(w_j \mid z_k)$,$P(z_k \mid d_i)$对应了两组多项分布，而计算每个文档的主题分布，就是该模型的任务目标。

极大似然估计：w_j在d_i中出现的次数 $n(d_i, w_j)$

$$L = \prod_{i=1}^{N} \prod_{j=1}^{M} P(d_i, w_j) = \prod_{i} \prod_{j} P(d_i, w_j)^{n(d_i, w_j)}$$

$$l = \sum_{i} \sum_{j} n(d_i, w_j) \log P(d_i, w_j)$$

$$= \sum_{i} \sum_{j} n(d_i, w_j) \log P(w_j \mid d_i) P(d_i)$$

$$= \sum_{i} \sum_{j} n(d_i, w_j) \log \left(\sum_{k=1}^{K} P(w_j \mid z_k) P(z_k \mid d_i) \right) P(d_i)$$

$$= \sum_{i} \sum_{j} n(d_i, w_j) \log \left(\sum_{k=1}^{K} P(w_j \mid z_k) P(z_k \mid d_i) P(d_i) \right)$$

$$P(d_i, w_j) = P(w_j \mid d_i) P(d_i)$$

$$P(w_j \mid d_i) = \sum_{k=1}^{K} P(w_j \mid z_k) P(z_k \mid d_i)$$

目标函数分析

- 目标函数 $l=\sum_i\sum_j n(d_i,w_j)\log\left(\sum_{k=1}^{K}P(w_j\mid z_k)P(z_k\mid d_i)P(d_i)\right)$
- 未知变量/自变量 $P(w_j\mid z_k),P(z_k\mid d_i)$
- 使用逐次逼近的办法：
  - 假定$P(z_k|d_i)$、$P(w_j|z_k)$已知，求隐含变量z_k的后验概率；
  - 在(d_i,w_j,z_k)已知的前提下，求关于参数$P(z_k|d_i)$、$P(w_j|z_k)$的似然函数期望的极大值，得到最优解$P(z_k|d_i)$、$P(w_j|z_k)$，带入上一步，从而循环迭代；
  - 即：EM算法。

求隐含变量主题z_k的后验概率

- 假定$P(z_k|d_i)$、$P(w_j|z_k)$已知，求隐含变量z_k的后验概率；

$$P(z_k \mid d_i, w_j) = \frac{P(w_j \mid z_k) P(z_k \mid d_i)}{\sum_{l=1}^{K} P(w_j \mid z_l) P(z_l \mid d_i)}$$

- 在(d_i, w_j, z_k)已知的前提下，求关于参数$P(z_k|d_i)$、$P(w_j|z_k)$的似然函数期望的极大值，得到最优解$P(z_k|d_i)$、$P(w_j|z_k)$，带入上一步，从而循环迭代；

分析似然函数期望

- 在(d_i, w_j, z_k)已知的前提下，求关于参数$P(z_k|d_i)$、$P(w_j|z_k)$的似然函数期望的极大值，得到最优解$P(z_k|d_i)$、$P(w_j|z_k)$，带入上一步，从而循环迭代；

分析似然函数期望

$$
\begin{aligned}
l &= \sum_i \sum_j n(d_i, w_j) \log P(d_i, w_j) \\
&= \sum_i \sum_j n(d_i, w_j) \log\left(P(w_j \mid d_i) P(d_i)\right) \\
&= \sum_i \sum_j n(d_i, w_j) \left(\log P(w_j \mid d_i) + \log P(d_i)\right) \\
&= \left(\sum_i \sum_j n(d_i, w_j) \log P(w_j \mid d_i) \right) + \left(\sum_i \sum_j n(d_i, w_j) \log P(d_i) \right)
\end{aligned}
$$

$$\Rightarrow$$

$$l_{new} = \left(\sum_i \sum_j n(d_i, w_j) \log P(w_j \mid d_i) \right)$$

$$
\begin{aligned}
E(l_{new}) &= \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log P(w_j, z_k \mid d_i) \\
&= \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log P(w_j \mid z_k) P(z_k \mid d_i)
\end{aligned}
$$

完成目标函数的建立

- 关于参数P($z_k|d_i$)、P($w_j|z_k$)的函数E，并且，带有概率加和为1的约束条件：

$$E=\sum_i\sum_j n(d_i,w_j)\sum_{k=1}^{K}P(z_k\mid d_i,w_j)\log P(w_j\mid z_k)P(z_k\mid d_i)$$

$$s.t.\begin{cases}\sum_{j=1}^{M}P(w_j\mid z_k)=1\\ \sum_{k=1}^{K}P(z_k\mid d_i)=1\end{cases}$$

- 显然，这是只有等式约束的求极值问题，使用Lagrange乘子法解决。

目标函数的求解

- Lagrange函数为：

$$Lag = \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log P(w_j \mid z_k) P(z_k \mid d_i)$$

$$+ \sum_{k=1}^{K} \tau_k \left(1 - \sum_{j=1}^{M} P(w_j \mid z_k) \right) + \sum_{i=1}^{N} \rho_i \left(1 - \sum_{k=1}^{K} P(z_k \mid d_i) \right)$$

- 求驻点：

$$\frac{\partial Lag}{\partial P(w_j \mid z_k)} = \frac{\sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)}{P(w_j \mid z_k)} - \tau_k \overset{令}{==} 0$$

$$\frac{\partial Lag}{\partial P(z_k \mid d_i)} = \frac{\sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)}{P(z_k \mid d_i)} - \rho_i \overset{令}{==} 0$$

分析第一个等式

$$\frac{\partial Lag}{\partial P(w_j \mid z_k)} = \frac{\sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)}{P(w_j \mid z_k)} - \tau_k \overset{令}{==} 0$$

$$\Rightarrow \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j) = \tau_k P(w_j \mid z_k)$$

$$\Rightarrow \sum_{m=1}^{M} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j) = \sum_{m=1}^{M} \tau_k P(w_j \mid z_k)$$

$$\Rightarrow \sum_{m=1}^{M} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j) = \tau_k \sum_{m=1}^{M} P(w_j \mid z_k)$$

$$\Rightarrow \sum_{m=1}^{M} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j) = \tau_k$$

$$\xrightarrow{将\tau_k代回第二式} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j) = \sum_{m=1}^{M} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j) P(w_j \mid z_k)$$

$$\Rightarrow P(w_j \mid z_k) = \frac{\sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)}{\sum_{m=1}^{M} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)}$$

同理分析第二个等式

- 求极值时的解——M-Step：

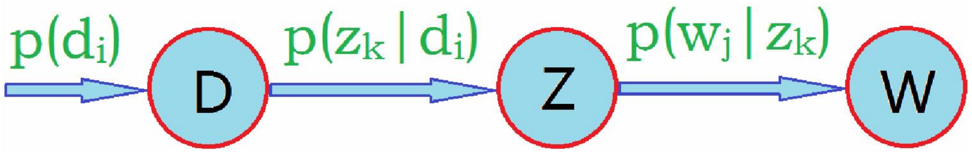
$$
\begin{cases}
P(w_j \mid z_k) = \dfrac{\sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)}{\sum_{m=1}^{M} \sum_i n(d_i, w_j) P(z_k \mid d_i, w_j)} \\
P(z_k \mid d_i) = \dfrac{\sum_j n(d_i, w_j) P(z_k \mid d_i, w_j)}{\sum_{k=1}^{K} \sum_j n(d_i, w_j) P(z_k \mid d_i, w_j)}
\end{cases}
$$

- 别忘了E-step： $P(z_k \mid d_i, w_j) = \dfrac{P(w_j \mid z_k) P(z_k \mid d_i)}{\sum_{l=1}^{K} P(w_j \mid z_l) P(z_l \mid d_i)}$

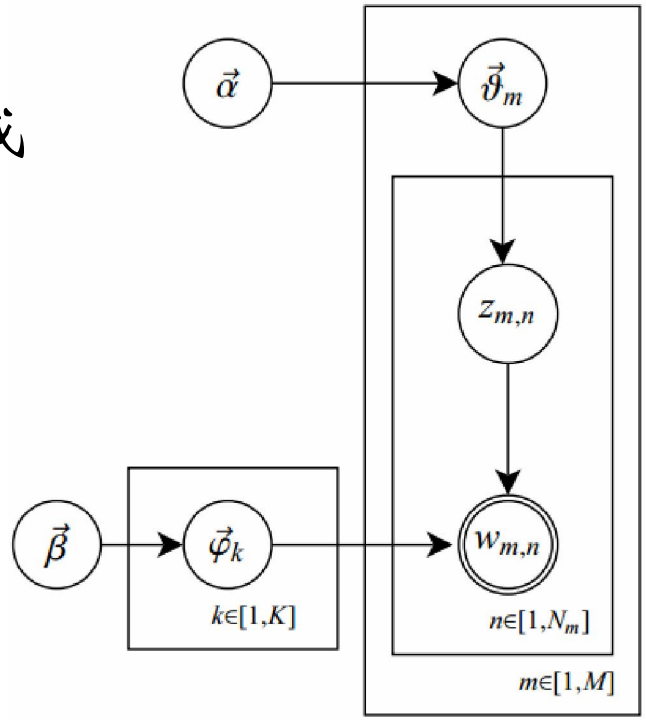
pLSA的总结

- pLSA应用于信息检索、过滤、自然语言处理等领域，pLSA考虑到词分布和主题分布，使用EM算法来学习参数。
- 虽然推导略显复杂，但最终公式简洁清晰，很符合直观理解，需用心琢磨；此外，推导过程使用了EM算法，也是学习EM算法的重要素材。

pLSA进一步思考



- 相对于“简单”的链状贝叶斯网络，可否给出“词”“主题”“文档”更细致的网络拓扑，形成更具一般性的模型？
- pLSA不需要先验信息即可完成自学习——这是它的优势。如果在特定的要求下，需要有先验知识的影响呢？
- 答：LDA模型；
  - 三层结构的贝叶斯模型
  - 需要超参数



LDA涉及的主要问题

- 共轭先验分布
- Dirichlet分布
- LDA模型
  - Gibbs采样算法学习参数

两种认识

- 给定某系统的若干样本，求该系统的参数。
- 矩估计/MLE/MaxEnt/EM等：
  - 假定参数是某个/某些未知的定值，求这些参数如何取值，能够使得某目标函数取极大/极小。
  - 频率学派
- 贝叶斯模型：
  - 假定参数本身是变化的，服从某个分布。求在这个分布约束下使得某目标函数极大/极小。
  - 贝叶斯学派
- 无高低好坏之分，只是认识自然的手段。只是在当前人们掌握的数学工具和需解决的实践问题中，贝叶斯学派的理论体系往往能够比较好的解释目标函数、分析相互关系等。
  - 前面章节的内容，大多是频率学派的思想；下面的推理，使用贝叶斯学派的观点。

贝叶斯公式 $P(A|B)=\frac{P(B|A)P(A)}{P(B)}$

- 给定某系统的若干样本x，计算该系统的参数，即

$$P(\theta|x)=\frac{P(x|\theta)P(\theta)}{P(x)}$$

  - P(θ)：没有数据支持下，θ 发生的概率：先验概率。
  - P(θ |x)：在数据x的支持下，θ 发生的概率：后验概率。
  - P(x| θ)：给定某参数 θ 的概率分布：似然函数。
- 例如：
  - 在没有任何信息的前提下，猜测某人姓氏：先猜李王张刘……猜对的概率相对较大：先验概率。
  - 若知道某人来自"牛家村"，则他姓牛的概率很大：后验概率——但不排除他姓郭、杨等情况。

共轭先验分布

- 由于x为给定样本，P(x)有时被称为“证据”，仅仅是归一化因子，如果不关心P(θ |x)的具体值，只考察 θ 取何值时后验概率P(θ |x)最大，则可将分母省去。

$$P(\theta|x)=\frac{P(x|\theta)P(\theta)}{P(x)}\propto P(x|\theta)P(\theta)$$

- 在贝叶斯概率理论中，如果后验概率P(θ |x)和先验概率p(θ)满足同样的分布律，那么，先验分布和后验分布被叫做共轭分布，同时，先验分布叫做似然函数的共轭先验分布。
- In Bayesian probability theory, if the posterior distributions p(θ |x) are in the same family as the prior probability distribution p(θ), the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function.

共轭先验分布的提出 $P(\theta|x) \propto P(x|\theta)P(\theta)$

- 某系统服从概率分布P(θ)时，
- 当观测到新的X数据时，有如下问题：
  - 根据新观测数据X，是否可以更新参数 θ
  - 根据新观测数据X可以在多大程度上更新参数 θ
    - θ ← θ + Δ θ
  - 当重新估计 θ 的时候，以新参数值 θ 的新概率分布最大作为估计依据。即：maxP(θ |x)

共轭先验分布的实践意义

- 根据贝叶斯法则 $P(\theta|x) \propto P(x|\theta)P(\theta)$
  - 似然函数P(x|θ)表示以先验θ为参数的概率分布，可以直接求得
  - 先验分布P(θ)是θ的分布率，可根据先验知识获得。
    - 从哪里获得先验知识？
- 方案：选取似然函数P(x|θ)的共轭先验作为P(θ)的分布，这样，P(x|θ)乘以P(θ)(然后归一化)得到的P(θ|x)的形式和P(θ)的形式一样。

举例说明

- 投掷一个非均匀硬币，可以使用参数为θ的伯努利模型，θ为硬币为正面的概率，那么结果x的分布形式为：$P(x|\theta)=\theta^{x}\cdot(1-\theta)^{1-x}$
- 两点分布/二项分布的共轭先验是Beta分布，它具有两个参数α和β，Beta分布形式为

$$P(\theta|\alpha,\beta)=\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1\theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}$$

先验概率和后验概率的关系

- 根据似然和先验：

$$P(x|\theta)=\theta^{x}\cdot(1-\theta)^{1-x} \qquad P(\theta|\alpha,\beta)=\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1\theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}$$

- 计算后验概率

$$\begin{aligned}&P(\theta|x)\\&\propto P(x|\theta)\cdot P(\theta)\\&\propto\left(\theta^{x}(1-\theta)^{x}\right)\left(\theta^{\alpha-1}(1-\theta)^{\beta-1}\right)\\&=\theta^{x+a-1}(1-\theta)^{1-x+\beta-1}\end{aligned}$$

- 该后验概率的形式与先验概率的形式完全一样——后验概率是参数为(x+α,x+β)的另一个Beta分布，即：伯努利分布的共轭先验是Beta分布。

伪计数

- 如果我们关心的是Bernoulli分布的参数θ，则参数α、β是决定参数θ的参数，常常称之为“超参数”。
- 观察Beta分布的定义和后验概率：

$$P(\theta|\alpha,\beta)=\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1\theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta} \qquad P(\theta|x)\propto\theta^{x+a-1}(1-\theta)^{1-x+\beta-1}$$

  - 可以发现，在后验概率的最终表达式中，参数α和β和x一起作为参数θ的指数——后验概率的参数为(x+α,x+β)。而这个指数的实践意义是：投币过程中，正面朝上的次数。α和β先验性的给出了在没有任何实验的前提下，硬币朝上的概率分配；因此，α和β常常被称作“伪计数”。

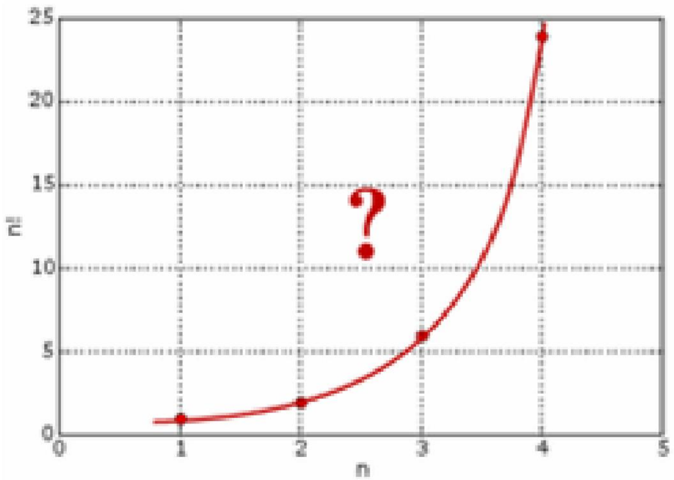
共轭先验的直接推广

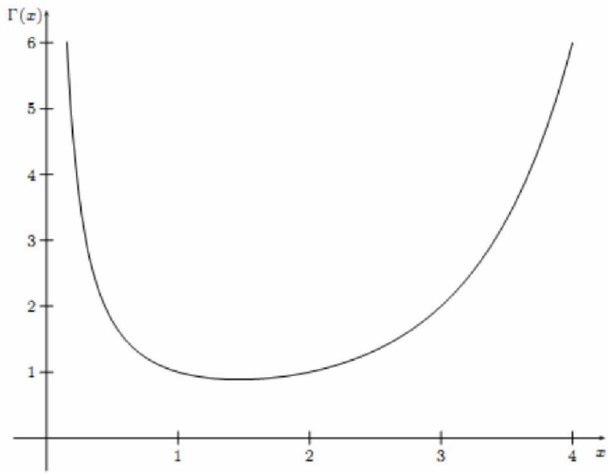
- 从2到K：
  - 二项分布→多项分布
  - Beta分布→Dirichlet分布

引：Γ函数

- Γ函数是阶乘在实数上的推广

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \qquad \Gamma(n) = (n-1)!$$





Dirichlet分布

- 参照Beta分布的定义：

$$P(\theta|\alpha,\beta)=\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}$$

- Dirichlet分布的定义：

$$p(\vec{p}|\vec{\alpha}) = \mathrm{Dir}(\vec{p}|\vec{\alpha}) \triangleq \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K} p_k^{\alpha_k-1} \triangleq \frac{1}{\Delta(\vec{\alpha})}\prod_{k=1}^{K} p_k^{\alpha_k-1}$$

$$\boxed{\Delta(\vec{\alpha}) = \frac{\prod_{k=1}^{\dim\vec{\alpha}}\Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{\dim\vec{\alpha}}\alpha_k)}}$$

Dirichlet分布分析

$$p(\vec{p}|\vec{\alpha}) = \frac{1}{\Delta(\vec{\alpha})}\prod_{k=1}^{K} p_k^{\alpha_k - 1}$$

- α 是参数向量，共K个
- 定义在$x_1, x_2 \ldots x_{K-1}$维上
  - $x_1 + x_2 + \ldots + x_{K-1} + x_K = 1$
  - $x_1, x_2 \ldots x_{K-1} > 0$
  - 定义在(K-1)维的单纯形上，其他区域的概率密度为0
- α 的取值对Dir(p|α)有什么影响？

Symmetric Dirichlet distribution

- A very common special case is the symmetric Dirichlet distribution, where all of the elements making up the parameter vector have the same value. Symmetric Dirichlet distributions are often used when a Dirichlet prior is called for, since there typically is *no prior knowledge* favoring one component over another. Since all elements of the parameter vector have the same value, the distribution alternatively can be parametrized by a single scalar value α, called the concentration parameter(聚集参数).

对称Dirichlet分布

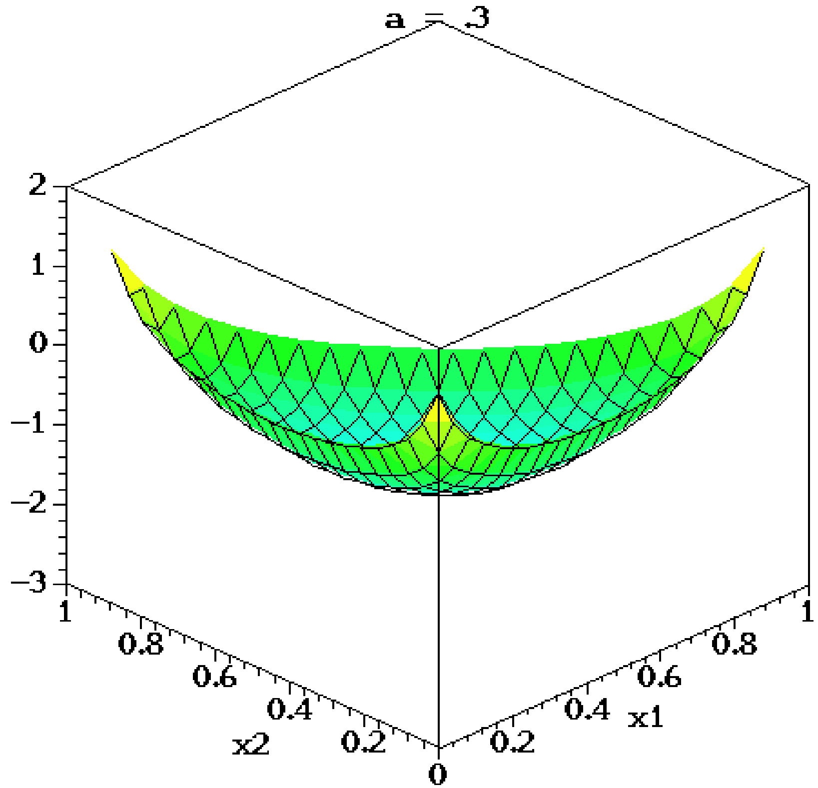
$$p(\vec{p}|\vec{\alpha}) = \mathrm{Dir}(\vec{p}|\vec{\alpha}) \triangleq \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}p_k^{\alpha_k-1} \triangleq \frac{1}{\Delta(\vec{\alpha})}\prod_{k=1}^{K}p_k^{\alpha_k-1}$$

$$p(\vec{p}|\alpha, K) = \mathrm{Dir}(\vec{p}|\alpha, K) \triangleq \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\prod_{k=1}^{K}p_k^{\alpha-1} \triangleq \frac{1}{\Delta_K(\alpha)}\prod_{k=1}^{K}p_k^{\alpha-1}$$

$$\boxed{\Delta_K(\alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(K\alpha)}}$$

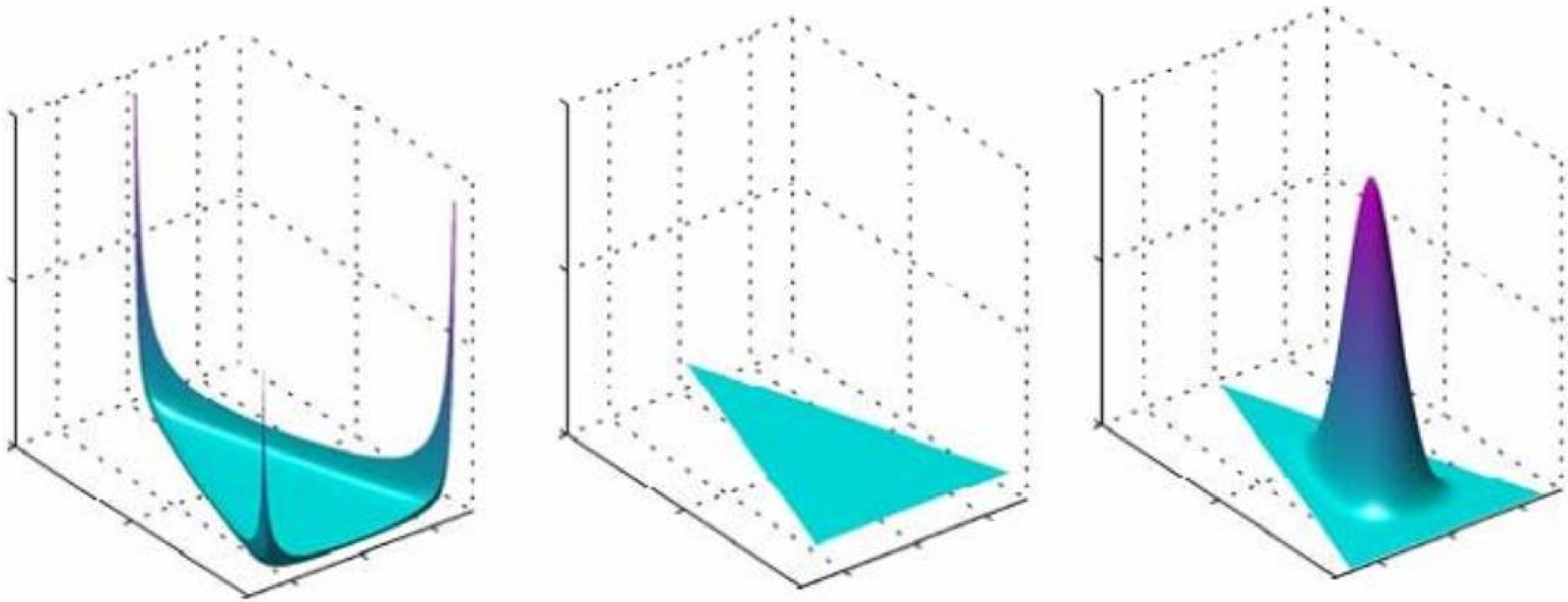
对称Dirichlet分布的参数分析

- $\alpha=1$ 时
  - 退化为均匀分布
- 当 $\alpha>1$ 时
  - $p_1=p_2=\ldots=p_k$ 的概率增大
- 当 $\alpha<1$ 时
  - $p_i=1$，$p_{非i}=0$ 的概率增大



图像说明：将Dirichlet分布的概率密度函数取对数，绘制对称Dirichlet分布的图像，取K=3，也就是有两个独立参数x1,x2，分别对应图中的两个坐标轴，第三个参数始终满足x3=1−x1−x2且α1=α2=α3=α，图中反映的是α从0.3变化到2.0的概率对数值的变化情况。

参数 α 对Dirichlet分布的影响



$\{\alpha_k\} = 0.1$ $\{\alpha_k\} = 1$ $\{\alpha_k\} = 10$

参数选择对对称Dirichlet分布的影响

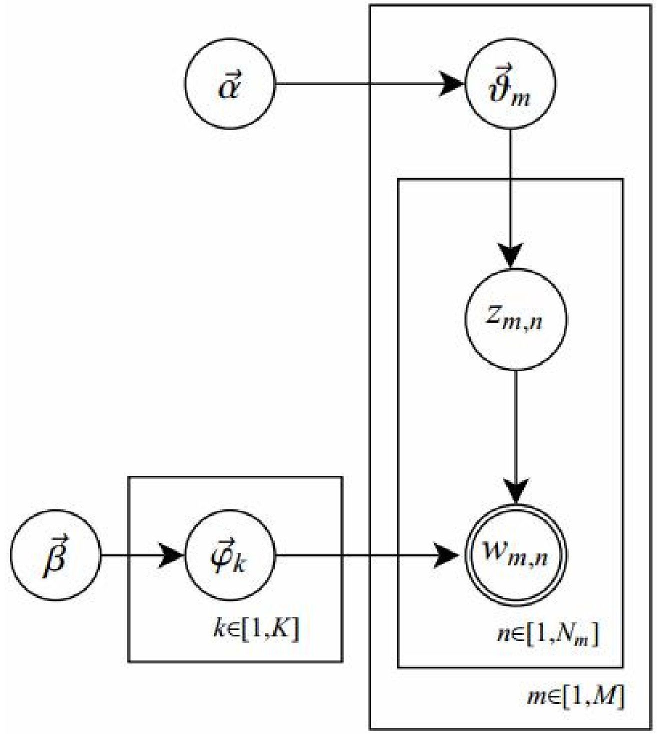
- When α =1, the symmetric Dirichlet distribution is equivalent to a uniform distribution over the open standard (K−1)-simplex, i.e. it is uniform over all points in its support. Values of the concentration parameter above 1 prefer variants that are dense, evenly distributed distributions, i.e. all the values within a single sample are similar to each other. Values of the concentration parameter below 1 prefer sparse distributions, i.e. most of the values within a single sample will be close to 0, and the vast majority of the mass will be concentrated in a few of the values.

多项分布的共轭分布是Dirichlet分布

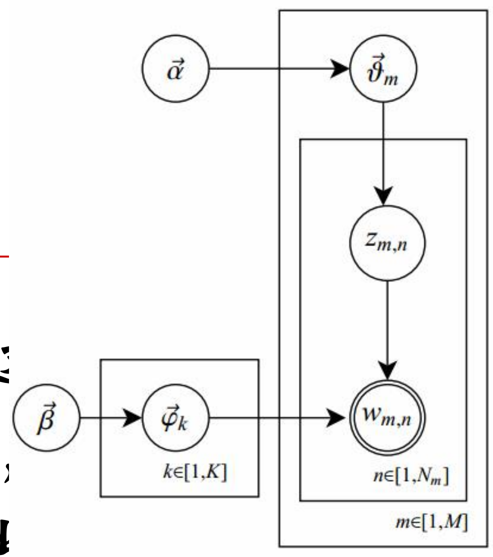
$$
\begin{aligned}
\boldsymbol{\alpha} &= (\alpha_1, \cdots, \alpha_K) = \text{concentration hyperparameter} \\
\mathbf{p} \mid \boldsymbol{\alpha} &= (p_1, \cdots, p_K) \sim \operatorname{Dir}(K, \boldsymbol{\alpha}) \\
\mathbb{X} \mid \mathbf{p} &= (\mathbf{x}_1, \cdots, \mathbf{x}_K) \sim \operatorname{Cat}(K, \mathbf{p}) \\
\mathbf{p} \mid \mathbb{X}, \boldsymbol{\alpha} &\sim \operatorname{Dir}(K, \mathbf{c} + \boldsymbol{\alpha}) = \operatorname{Dir}(K, c_1 + \alpha_1, \cdots, c_K + \alpha_K)
\end{aligned}
$$

LDA的解释

- 共有m篇文章，一共涉及了K个主题；
- 每篇文章(长度为N_m)都有各自的主题分布，主题分布是多项分布，该多项分布的参数服从Dirichlet分布，该Dirichlet分布的参数为α；
- 每个主题都有各自的词分布，词分布为多项分布，该多项分布的参数服从Dirichlet分布，该Dirichlet分布的参数为β；
- 对于某篇文章中的第n个词，首先从该文章的主题分布中采样一个主题，然后在这个主题对应的词分布中采样一个词。不断重复这个随机生成过程，直到m篇文章全部完成上述过程。



详细解释

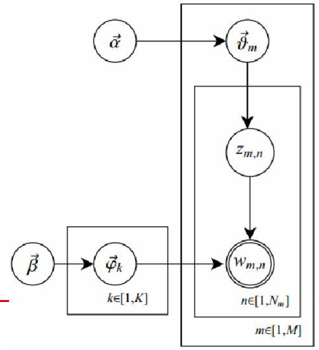


- 字典中共有V个term(不可重复)，
  现在具体　　　　的文章中，
  word——在具体某文章中的word当
  是有可能重复的。
- 语料库中共有m篇文档$d_1,d_2\ldots d_m$；
- 对于文档d_i，由N_i个word组成，可重复；
- 语料库中共有K个主题T_1，$T_2\ldots T_k$；
- α和β为先验分布的参数，一般事先给定：如取0.1的对称Dirichlet分布——表示在参数学习结束后，期望每个文档的主题不会十分

详细解释

- 图中K为主题个数，M为文档总数，Nm是第m个文档的单词总数。β是每个Topic下词的多项分布的Dirichlet先验参数，α是每个文档下Topic的多项分布的Dirichlet先验参数。zmn是第m个文档中第n个词的主题，wmn是m个文档中的第n个词。两个隐含变量θ和φ分别表示第m个文档下的Topic分布和第k个Topic下词的分布，前者是k维(k为Topic总数)向量，后者是v维向量(v为词典中term总数)

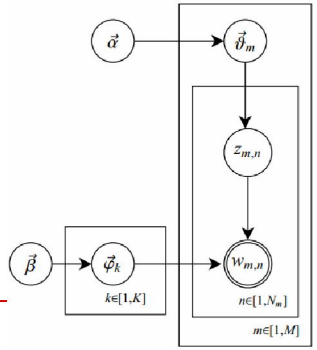
参数的学习



- 给定一个文档集合，w_{mn}是可以观察到的已知变量，α和β是根据经验给定的先验参数，其他的变量z_{mn}、θ、φ都是未知的隐含变量，需要根据观察到的变量来学习估计的。根据LDA的图模型，可以写出所有变量的联合分布：

$$p(\vec{w}_m, \vec{z}_m, \vec{\vartheta}_m, \underline{\Phi} | \vec{\alpha}, \vec{\beta}) = \prod_{n=1}^{N_m} p(w_{m,n} | \vec{\varphi}_{z_{m,n}}) p(z_{m,n} | \vec{\vartheta}_m) \cdot p(\vec{\vartheta}_m | \vec{\alpha}) \cdot p(\underline{\Phi} | \vec{\beta})$$

似然概率



- 一个词w_{mn}初始化为一个词t的概率是

$$p(w_{m,n}=t|\vec{\vartheta}_m,\underline{\Phi})=\sum_{k=1}^{K}p(w_{m,n}=t|\vec{\varphi}_k)p(z_{m,n}=k|\vec{\vartheta}_m)$$

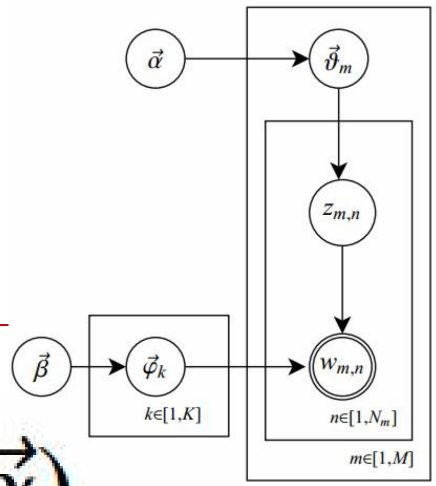
- 每个文档中出现主题k的概率乘以主题k下出现词t的概率，然后枚举所有主题求和得到。整个文档集合的似然函数为：

$$p(\mathcal{W}|\underline{\Theta},\underline{\Phi})=\prod_{m=1}^{M}p(\vec{w}_m|\vec{\vartheta}_m,\underline{\Phi})=\prod_{m=1}^{M}\prod_{n=1}^{N_m}p(w_{m,n}|\vec{\vartheta}_m,\underline{\Phi})$$

Gibbs Sampling

- Gibbs Sampling算法的运行方式是每次选取概率向量的一个维度，给定其他维度的变量值采样当前维度的值。不断迭代，直到收敛输出待估计的参数。
- 初始时随机给文本中的每个词分配主题$z^{(0)}$，然后统计每个主题z下出现词t的数量以及每个文档m下出现主题z的数量，每一轮计算$p(z_i|\mathbf{z}_{-i},\mathbf{d},\mathbf{w})$，即排除当前词的主题分布：
  - 根据其他所有词的主题分布估计当前词分配各个主题的概率。
- 当得到当前词属于所有主题z的概率分布后，根据这个概率分布为该词采样一个新的主题。
- 用同样的方法更新下一个词的主题，直到发现每个文档的主题分布θ_i和每个主题的词分布ϕ_j收敛，算法停止，输出待估计的参数θ和ϕ，同时每个单词的主题z_{mn}也可同时得出。
- 实际应用中会设置最大迭代次数。每一次计算$p(zi|\mathbf{z}_{-i},\mathbf{d},\mathbf{w})$的公式称为Gibbs updating rule。

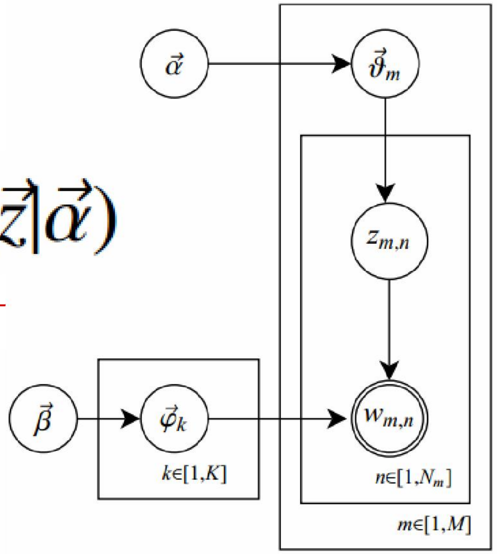
联合分布



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

- 第一项因子是给定主题采样词的过程
- 后面的因子计算，$n_z^{(t)}$表示词t被观察到分配给主题z的次数，$n_m^{(k)}$表示主题k分配给文档m的次数。

计算因子 $p(\vec{w},\vec{z}|\vec{\alpha},\vec{\beta}) = p(\vec{w}|\vec{z},\vec{\beta})p(\vec{z}|\vec{\alpha})$



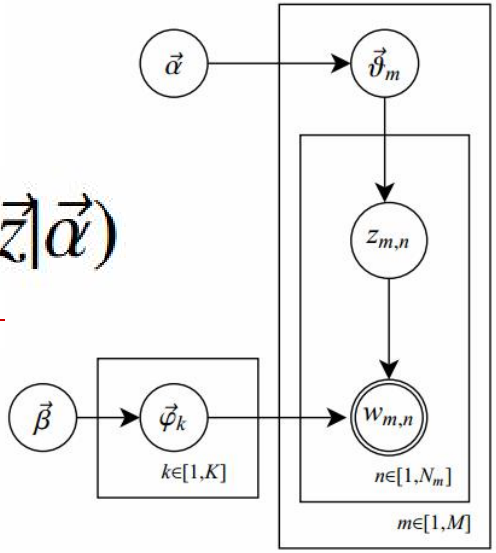
$$p(\vec{w}|\vec{z},\vec{\beta}) = \int p(\vec{w}|\vec{z},\underline{\Phi})\ p(\underline{\Phi}|\vec{\beta})\ \mathrm{d}\underline{\Phi}$$

$$= \int \prod_{z=1}^{K} \frac{1}{\Delta(\vec{\beta})} \prod_{t=1}^{V} \varphi_{z,t}^{n_z^{(t)}+\beta_t-1}\ \mathrm{d}\vec{\varphi}_z$$

$$= \prod_{z=1}^{K} \frac{\Delta(\vec{n}_z+\vec{\beta})}{\Delta(\vec{\beta})}, \quad \vec{n}_z = \{n_z^{(t)}\}_{t=1}^{V}$$

$$\int_{\vec{p}} \prod_{k=1}^{K} p_k^{\alpha_k-1} d\vec{p} = \Delta(\vec{\alpha})$$

计算因子 $p(\vec{w},\vec{z}|\vec{\alpha},\vec{\beta}) = p(\vec{w}|\vec{z},\vec{\beta})p(\vec{z}|\vec{\alpha})$



$$p(\vec{z}|\vec{\alpha}) = \int p(\vec{z}|\underline{\Theta})\ p(\underline{\Theta}|\vec{\alpha})\ \mathrm{d}\underline{\Theta}$$

$$= \int \prod_{m=1}^{M} \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{K} \vartheta_{m,k}^{n_m^{(k)}+\alpha_k-1}\ \mathrm{d}\vec{\vartheta}_m$$

$$= \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}, \qquad \vec{n}_m = \{n_m^{(k)}\}_{k=1}^{K}$$

$$\int_{\vec{p}} \prod_{k=1}^{K} p_k^{\alpha_k-1} d\vec{p} = \Delta(\vec{\alpha})$$

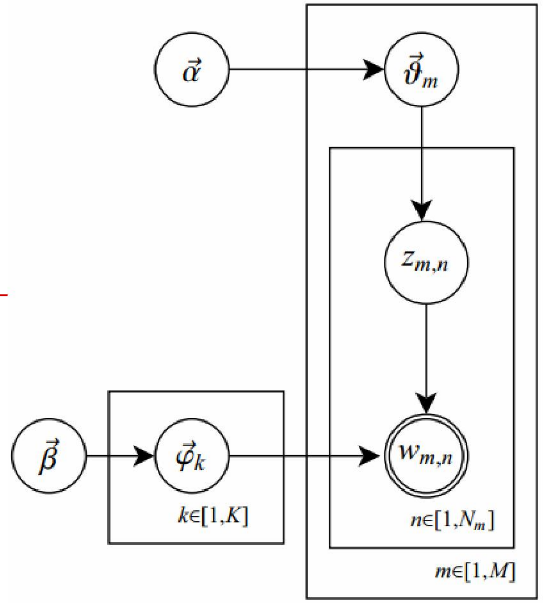
Gibbs updating rule

$$
\begin{aligned}
p(z_i{=}k|\vec{z}_{\neg i},\vec{w}) &= \frac{p(\vec{w},\vec{z})}{p(\vec{w},\vec{z}_{\neg i})} = \frac{p(\vec{w}|\vec{z})}{p(\vec{w}_{\neg i}|\vec{z}_{\neg i})p(w_i)} \cdot \frac{p(\vec{z})}{p(\vec{z}_{\neg i})} \\
&\propto \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{n}_{z,\neg i} + \vec{\beta})} \cdot \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{n}_{m,\neg i} + \vec{\alpha})} \\
&= \frac{\Gamma(n_k^{(t)} + \beta_t)\,\Gamma(\sum_{t=1}^V n_{k,\neg i}^{(t)} + \beta_t)}{\Gamma(n_{k,\neg i}^{(t)} + \beta_t)\,\Gamma(\sum_{t=1}^V n_k^{(t)} + \beta_t)} \cdot \frac{\Gamma(n_m^{(k)} + \alpha_k)\,\Gamma(\sum_{k=1}^K n_{m,\neg i}^{(k)} + \alpha_k)}{\Gamma(n_{m,\neg i}^{(k)} + \alpha_k)\,\Gamma(\sum_{k=1}^K n_m^{(k)} + \alpha_k)} \\
&= \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^V n_{k,\neg i}^{(t)} + \beta_t} \cdot \frac{n_{m,\neg i}^{(k)} + \alpha_k}{[\sum_{k=1}^K n_m^{(k)} + \alpha_k] - 1} \\
&\propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^V n_{k,\neg i}^{(t)} + \beta_t}(n_{m,\neg i}^{(k)} + \alpha_k)
\end{aligned}
$$

词分布和主题分布

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t}$$

$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k}$$



Gibbs采样算法

Algorithm LdaGibbs($\{\vec{w}\}, \alpha, \beta, K$)

Input: word vectors $\{\vec{w}\}$, hyperparameters α, β, topic number K

Global data: count statistics $\{n_m^{(k)}\}$, $\{n_k^{(t)}\}$ and their sums $\{n_m\}$, $\{n_k\}$, memory for full conditional array $p(z_i|\cdot)$

Output: topic associations $\{\vec{z}\}$, multinomial parameters $\underline{\Phi}$ and $\underline{\Theta}$, hyperparameter estimates α, β

```
// initialisation
zero all count variables, n_m^(k), n_m, n_k^(t), n_k
for all documents m ∈ [1, M] do
    for all words n ∈ [1, N_m] in document m do
        sample topic index z_{m,n}=k ~ Mult(1/K)
        increment document–topic count: n_m^(k) += 1
        increment document–topic sum: n_m += 1
        increment topic–term count: n_k^(t) += 1
        increment topic–term sum: n_k += 1
// Gibbs sampling over burn-in period and sampling period
while not finished do
    for all documents m ∈ [1, M] do
        for all words n ∈ [1, N_m] in document m do
            // for the current assignment of k to a term t for word w_{m,n}:
            decrement counts and sums: n_m^(k) −= 1; n_m −= 1; n_k^(t) −= 1; n_k −= 1
            // multinomial sampling acc. to Eq. 78 (decrements from previous step):
            sample topic index k̃ ~ p(z_i | z⃗_¬i, w⃗)
            // for the new assignment of z_{m,n} to the term t for word w_{m,n}:
            increment counts and sums: n_m^(k̃) += 1; n_m += 1; n_k̃^(t) += 1; n_k̃ += 1
    // check convergence and read out parameters
    if converged and L sampling iterations since last read out then
        // the different parameters read outs are averaged.
        read out parameter set Φ according to Eq. 81
        read out parameter set Θ according to Eq. 82
```

LDA总结

- 由于在词和文档之间加入的主题的概念，可以较好的解决一词多义和多词一义的问题。
- 在实践中发现，LDA用于短文档往往效果不明显——这是可以解释的：因为一个词被分配给某个主题的次数和一个主题包括的词数目往往无法收敛。往往需要通过其他方案“连接”成长文档。
  - 用户评论
  - Twitter/微博
- LDA可以和其他算法想结合。如：使用LDA，可以将长度为Ni的文档“降维”到K维(主题的数目)，同时可以给出每个主题的概率(主题分布)，因此可以将其使用if-idf(词频-逆文档频率)继续分析或者直接作为文档的特征进入聚类或者标签传播算法——用于社区挖掘或其他目的。

参考文献

- Gregor Heinrich, Parameter estimation for text analysis
- David M. Blei, Andrew Y. Ng, Michael I. Jordan, Latent Dirichlet Allocation, 2003
- http://blog.csdn.net/yangliuy/article/details/8330640(pLSA&EM)
- http://en.wikipedia.org/wiki/Dirichlet_distribution(Dirichlet分布)
- http://en.wikipedia.org/wiki/Conjugate_prior(共轭分布)

我们在这里

- 更多算法面试题在 7 七月算法
  - http://www.julyedu.com/
    - 免费视频
    - 直播课程
    - 问答社区

- contact us：微博
  - @研究者July
  - @七月问答
  - @邹博_机器学习

感谢大家！

恳请大家批评指正！