关键词提取

关键词提取是指从文本中提取与文章主题或当前任务最相关的词语，例如，从文献中选取出能够反映文章主题的词语作为关键词。关键词提取可用于自动文摘、文献检索，文本分类等任务。提取关键词最简单的一个方法就是利用词频，将高频出现的词作为关键词，这种方法虽然简单，但是往往结果较差。本章将介绍以下几种关键词提取算法：基于图的TextRank算法、基于词频和逆文本频率的TF-IDF算法、基于分布的LDA算法和PLSA算法。

5.1 TextRank关键词提取算法

TextRank[1]是一个无监督的基于图的排序算法，它与Google的PageRank算法相似。PageRank是根据网页之间的链接关系来对网页的重要程度进行排序。也就是说，在进行网页排序的过程中，PageRank算法只需要知道网页间的结构信息，而无须知道网页的具体内容。从图的角度来看，网页可以作为节点，若两个网页间存在链接关系，则相应的两个节点间就存在一条边。图中每个节点的重要程度由图中的这些边决定，即由整张图的结构决定，而非由节点的内部信息决定。TextRank采用了与PageRank相似的思路。在提取关键词时，TextRank由文本中提取出词汇图，以单词作为节点，单词间的关系作为边，然后根据图对单词进行排序。

5.1.1 基于图的排序算法

基于图的排序算法会递归地从整张图中提取信息，最终完成对图中节点的重要程度的排序。基于图的排序算法的基本思想是投票，若图中存在由节点A指向节点B的边，则认为节点A为节点B投了一票，图中获得票数越多的节点就越重要。此外，不同的节点投出的票的重要程度是不同的，这取决于投票节点的重要程度。因此决定节点的重要程度的因素有两个：有多少节点为该节点投票；为该节点投票的那些节点自身的重要程度。

记有向图为$G=(V,\boldsymbol{E})$，其中V代表节点集，$\boldsymbol{E}$代表边集，$\boldsymbol{E}$是一个$|V|\times|V|$的矩阵。

若图 G 中存在由 V_j 指向 V_i 的边，则矩阵中第 i 行第 j 列的元素值为 1，否则为 0。对于节点 V_i，$S(V_i)$为节点 V_i 的得分，即节点 V_i 的所得的票数，每个节点都会被赋予一个随机的初始得分，然后对每个节点的得分进行迭代更新，直至节点的得分值收敛。节点得分的更新方式如下：

所有指向 V_i 的节点组成的集合记为 $\mathrm{In}(V_i)$；所有由 V_i 出发所指向的节点组成的集合记为 $\mathrm{Out}(V_j)$；则节点 V_i 经过一次更新后的得分为：

$$S(V_i)=(1-d)+d\times\sum_{V_j\in \mathrm{In}(V_i)}\frac{1}{|\mathrm{Out}(V_j)|}S(V_j) \tag{5.1}$$

式(5.1)中的 d 为阻尼系数，其值为 0～1，通常取 0.85[2]。d 用于模拟从一个节点跳转至另一个节点的概率。在用户浏览网页的情景下，可以理解为：若当前处于网页 V_i，用户通过单击此网页上的链接，跳转至新网页 V_j ($j\in \mathrm{In}(V_i)$)的概率为 d；用户没有单击此网页上的链接，而是直接打开了一个全新的网页 V_k ($k\notin \mathrm{In}(V_i)$)的概率为 $1-d$。

【例 5-1】 模拟图的更新过程。

如图 5-1 所示的有向图 G 可以表示为：

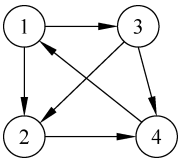


图 5-1 有向图 G

$$G=(V,\boldsymbol{E})$$

$$V=\{V_1,V_2,V_3,V_4\}$$

$$\boldsymbol{E}=\begin{bmatrix}0&0&0&1\\1&0&1&0\\1&0&0&0\\0&1&1&0\end{bmatrix}$$

假设分数的随机初始值为：

$$S(V_1)=1,\quad S(V_2)=4,\quad S(V_3)=2,\quad S(V_4)=6$$

取 $d=0.85$，对于节点 V_2，第一次的更新过程如下：

$$\mathrm{In}(V_2)=\{V_1,V_3\},\quad \mathrm{Out}(V_1)=\{V_2,V_3\},\quad \mathrm{Out}(V_3)=\{V_2,V_4\}$$

$$S(V_2)=(1-0.85)+0.85\times\left(\frac{1}{|\mathrm{Out}(V_1)|}S(V_1)+\frac{1}{|\mathrm{Out}(V_3)|}S(V_3)\right)$$

$$=0.15+0.85\times\left(\frac{1}{2}\times1+\frac{1}{2}\times2\right)=1.425$$

则第一次更新后：

$$S(V_2)=1.425$$

若从矩阵的角度计算，第一次更新的计算过程为：

$$\boldsymbol{E}\times\boldsymbol{D}^{-1}\times\boldsymbol{S}=\begin{bmatrix}0&0&0&1\\1&0&1&0\\1&0&0&0\\0&1&1&0\end{bmatrix}\times\begin{bmatrix}2&0&0&0\\0&1&0&0\\0&0&2&0\\0&0&0&1\end{bmatrix}^{-1}\times\begin{bmatrix}1\\4\\2\\6\end{bmatrix}$$

$$=\begin{bmatrix}0&0&0&1\\0.5&0&0.5&0\\0.5&0&0&0\\0&1&0.5&0\end{bmatrix}\times\begin{bmatrix}1\\4\\2\\6\end{bmatrix}$$

$$=\begin{bmatrix}6\\1.5\\0.5\\5\end{bmatrix}$$

其中，

$$D_{ii}=\sum_j e_{ji}$$

$$\boldsymbol{S}=(1-0.85)+0.85\times\begin{bmatrix}6\\1.5\\0.5\\5\end{bmatrix}=\begin{bmatrix}5.250\\1.425\\0.575\\4.400\end{bmatrix}$$

则第一次更新后：

$$S(V_1)=5.250,\quad S(V_2)=1.425,\quad S(V_3)=0.575,\quad S(V_4)=4.400$$

重复上述过程，对节点的得分进行迭代更新，直至节点的得分收敛，收敛后每个节点的得分即为最终能够反映节点重要程度的得分。

5.1.2　基于图的排序算法的拓展运用

传统的基于图的排序算法作用于有向图，但是也可运用于无向图和加权图中。

在无向图中，若想使用基于图的排序算法，只需假设无向图中节点的入度和出度相等且都等于该节点的度，即 $\mathrm{In}(V_i)=\mathrm{Out}(V_i)=\mathrm{Degree}(V_i)$。

加权图是指边有相应权重的图，根据自然语言文本构建的图往往都是加权图，TextRank 算法中使用的便是加权图。对于加权图，我们将节点的更新公式修改为：

$$\boldsymbol{W}S(V_i)=(1-d)+d\times\sum_{V_j\in\mathrm{In}(V_i)}\frac{w_{ji}}{\sum\limits_{V_k\in\mathrm{Out}(V_j)}w_{jk}}\boldsymbol{W}S(V_j)\tag{5.2}$$

【例 5-2】 基于图的排序算法在加权图中的运用。

如图 5-2 所示的带权有向图 G 可以表示为：

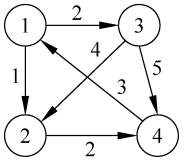


图 5-2　带权有向图 G

$$G=(V,\boldsymbol{E})$$

$$V=\{V_1,V_2,V_3,V_4\}$$

$$\boldsymbol{E}=\begin{bmatrix}0&0&0&1\\1&0&1&0\\1&0&0&0\\0&1&1&0\end{bmatrix}$$

边的权重矩阵为：

$$\boldsymbol{W}=\begin{bmatrix}0&1&2&0\\0&0&0&2\\0&4&0&5\\3&0&0&0\end{bmatrix}$$

假设分数的随机初始值为：

$$S(V_1)=1,\quad S(V_2)=4,\quad S(V_3)=2,\quad S(V_4)=6$$

取 $d=0.85$，对于节点 V_2，第一次的更新过程如下：

$$\mathrm{In}(V_2)=\{V_1,V_3\},\quad \mathrm{Out}(V_1)=\{V_2,V_3\},\quad \mathrm{Out}(V_3)=\{V_2,V_4\}$$

$$S(V_2)=(1-0.85)+0.85\times\left(\frac{w_{12}}{w_{12}+w_{13}}S(V_1)+\frac{w_{32}}{w_{32}+w_{34}}S(V_3)\right)$$

$$=0.15+0.85\times\left(\frac{1}{1+2}\times1+\frac{4}{4+5}\times2\right)=1.189$$

则第一次更新后：

$$S(V_2)=1.189$$

若从矩阵的角度计算，第一次更新的计算过程为：

$$\boldsymbol{W}^{\mathrm{T}}\times\boldsymbol{D}^{-1}\times\boldsymbol{S}=\begin{bmatrix}0&0&0&3\\1&0&4&0\\2&0&0&0\\0&2&5&0\end{bmatrix}\times\begin{bmatrix}3&0&0&0\\0&2&0&0\\0&0&9&0\\0&0&0&3\end{bmatrix}^{-1}\times\begin{bmatrix}1\\4\\2\\6\end{bmatrix}$$

$$=\begin{bmatrix}0&0&0&3\\1/3&0&4/9&0\\2/3&0&0&0\\0&2&5/9&0\end{bmatrix}\times\begin{bmatrix}1\\4\\2\\6\end{bmatrix}=\begin{bmatrix}18\\11/9\\2/3\\82/9\end{bmatrix}$$

其中，

$$D_{ii}=\sum_j w_{ij}$$

$$S=(1-0.85)+0.85\times\begin{bmatrix}18\\11/9\\2/3\\82/9\end{bmatrix}=\begin{bmatrix}15.450\\1.189\\0.717\\7.894\end{bmatrix}$$

则第一次更新后：

$$S(V_1)=15.450,\quad S(V_2)=1.189,\quad S(V_3)=0.717,\quad S(V_4)=7.894$$

重复上述过程，对节点的得分进行迭代更新，直至节点的得分收敛，收敛后每个节点的得分即能够反映节点的重要程度。

5.1.3 基于图的排序算法在关键词提取中的运用

为了将基于图的排序算法运用于自然语言文本，首先需要用图来表示文本。不同级别的文本单元均可以作为节点，例如，节点可以是单词、固定搭配、整个句子等。节点间的关系可以是语法或语义相关、位置相关等。

将基于图的排序模型运用于图，需要以下步骤。

(1) 根据当前任务将文本分割为一个个小的文本单元，并且将这些文本单元作为图的节点。

(2) 定义文本单元间存在哪些关系，并且根据这些关系在节点间构建边。

(3) 使用基于图的排序算法对图进行迭代更新直至节点对应的得分值收敛。

(4) 根据每个节点的得分对节点进行降序排列。

5.1.4 TextRank 算法

TextRank 算法的输入为自然语言文本，其输出为由文本中关键的单词或短语构成的集合。TextRank 算法的实现主要包含以下步骤。

(1) 对文本进行过滤和分割，将分割后的文本作为图中的节点。为了防止图的节点数过多，设分割后的一个单元中仅包含一个单词，即文本中一个单词对应图中的一个节点。每一个节点的得分都被初始化为 1。

(2) 为存在共现关系的节点建立边。单词间的共现关系定义为：设一个窗口中最多包含 N 个单词，若文本中的两个单词出现于同一个窗口，则称这两个单词间存在共现关系。

(3) 使用基于图的排序算法对图进行迭代更新直至收敛。通常需要进行 20～30 轮迭代。

(4) 根据每个节点的得分对节点进行降序排列，选择得分最高的 T 个节点对应的单词作为关键词。通常 T 的值设置为节点数量的三分之一。对于挑选出的关键词，若某些关键词在文本中是相邻的，则将它们组合起来作为一个关键词。

5.2 TF-IDF 关键词提取算法

TF-IDF(Term Frequency-Inverse Document Frequency)是一种用于信息检索与数据挖掘的常用加权技术。其中 TF 是词频，IDF 是逆文本频率指数。在关键词提取问题中，TF-IDF 可以用来评估某个词在文件集的某篇文档中的重要程度。若一个词在某篇文档中高频出现，但是在文件集其他文档中出现的频率较低，则说明这个词较能反映这篇文档的特征。据此，TF-IDF 中使用了两个指标，即词频 TF 和逆向文档词频 IDF，TF 的值与词语在当前文档中出现的频率成正比，IDF 的值与词语在文件集所有文档中出现的频率成反比，TF-IDF 的值即为 TF 与 IDF 的乘积。在某篇文档中，某个单词对应的 TF-IDF 值越大说明这个单词在文档中越重要。

TF(w,d)是指词 w 在文档 d 中出现的频率。设文档 d 中共包含 N_d 个词，其中词 w 出现了 $N_{d,w}$ 次，则

$$\mathrm{TF}(w,d)=\frac{N_{d,w}}{N_d} \tag{5.3}$$

在文件集中，设文档总数为 D，词 w 在 D_w 篇文档中出现过，则

$$\mathrm{IDF}(w)=\ln\left(\frac{D}{D_w+1}\right) \tag{5.4}$$

那么，单词 w 在文档 d 中的 $\mathrm{TF\text{-}IDF}_{d,w}$ 值为：

$$\mathrm{TF\text{-}IDF}_{d,w}=\mathrm{TF}(w,d)\times\mathrm{IDF}(w) \tag{5.5}$$

【例 5-3】 假设在 500 篇文章中，“自然语言处理”一词在 30 篇文章中出现过。第一篇文章共包含 1000 个词，其中“自然语言处理”一词出现了 80 次。求“自然语言处理”相应的 TF-IDF 值。

“自然语言处理”对应的 TF 值为：

$$\mathrm{TF}=80/1000=0.08$$

“自然语言处理”对应的 IDF 值为：

$$\mathrm{IDF}=\ln(500/31)\approx 2.78$$

那么，“自然语言处理”对应的 TF-IDF 值为：

$$\mathrm{TF} \times \mathrm{IDF} = 0.08 \times 2.78 \approx 0.22$$

通过上述过程可以计算出文档中每个词语对应的 TF-IDF 值，并选择 TF-IDF 值较大的词作为关键词。

5.3 LDA 与 PLSA 关键词提取算法

LDA（Latent Dirichlet Allocation[3]）模型和 PLSA（Probabilistic Latent Semantic Analysis[4]）模型都属于生成概率模型。模型中会用到以下数据：单词、文档和语料。

（1）单词 w：文本由一些离散的基本单元构成，文本中的一个单词被认为是一个基本单元，LDA 和 PLSA 模型中忽略了单词间的顺序关系。

（2）文档 d：由 N 个单词的序列构成，表示为 $d=\{w_1,w_2,\cdots,w_N\}$。

（3）语料 D：由 M 篇文档组成，$D=\{d_1,d_2,\cdots,d_M\}$。

对于给定的语料库，LDA 模型和 PLSA 模型都能估计出语料库中每篇文档的主题分布和词分布。

（1）主题是某些关键词的集合，如果一篇文档中出现这些关键词，认为该文档中包含该主题的内容。一篇文档可能包含多种主题，某一主题的关键词在文中出现的次数越多，则该主题在文档的主题分布中相应的概率值越大。例如，对于一篇新闻文档，其中可能包含“金融”“教育”“科技”3 个主题的相关内容，而这 3 个主题在文档中的分布情况即为该文档的主题分布，比如主题分布可能为{金融：0.1，教育：0.6，科技：0.3}。

（2）对于某一主题，其中会包含许多关键词，与该主题越相关的词在主题的词分布中相应的概率值越大。

主题分布可以反映主题与文档的相关度，而词分布可以反映主题与词语的相关度，那么这两个相关度的乘积就能够反映词语与文档的相关度。得到文档中各个词语与该文档的相关度后，可以提取与文档相关度较高的词语作为文档的关键词。

LDA 模型和 PLSA 模型都属于词袋模型，即 LDA 模型和 PLSA 模型不关注词语间的顺序。LDA 模型和 PLSA 模型中抽取一个词语的过程如下：对于某文档，根据该文档的主题分布抽取一个主题，再根据主题上的词分布抽取一个词。对于一篇有 N 个词的文档来说，需要将词语抽取过程重复 N 次，便可得到一篇文档。在此过程中，已知文档中有哪些词，即已知文档中的词分布，要据此推断文档的主题分布和主题的词分布。为了更好地理解这一过程，可以拿常见的抛硬币问题进行类比：假设当前有两个质地不均匀的硬币 A 和硬币 B，一次只抛硬币 A 和硬币 B 中的一个。抛硬币的过程可以描述为：先按照一定的概率，从硬币 A 和硬币 B 中选择一个，然后多次抛这枚硬币，记录抛出的硬币是正面朝上还是反面朝上。也就是已知多轮抛硬币的结果，求硬币 A 和硬币 B 被选中的概率和每枚硬币正面朝上的概率。在这个例子中，选硬币的过程类似选主题的过程，而抛硬币的过程与选词语的过程类似，这个例子在下面还会详述。

5.3.1 相关基础知识

1. 分布

（1）泊松分布：泊松分布的参数 λ 是单位时间（或单位面积）内随机事件的平均发生次

数。在 LDA 模型中用泊松分布来模拟文档中词语个数的分布情况。泊松分布的概率密度函数为：

$$P(X=k)=\frac{\lambda^k}{k!}e^{-\lambda},\quad k=0,1,\cdots \tag{5.6}$$

(2) 二项式分布(Binomial distribution)：二项式分布是 n 个独立的“是/非试验”中成功的次数的离散概率分布，其中每次试验的成功概率为 p。当 $n=1$ 时，二项式分布就是伯努利分布。二项式分布的概率密度函数为：

$$P(K=k)=\binom{n}{k}p^k(1-p)^{n-k},\quad \binom{n}{k}=\frac{n!}{k!(n-k)!} \tag{5.7}$$

(3) 多项式分布(Multinomial Distribution)：多项式分布是二项式分布在高维度上的推广，二项式分布的试验结果只有两个(是和非)，而多项式分布的试验结果则多于两个。参照二项式分布试验的特点，多项式分布试验的特点如下：每种结果都有各自发生的概率，所有结果的发生概率之和为 1；各次试验相互独立，每次试验结果都不受其他各次试验结果的影响。

$$\sum_{i=1}^{k}p_i=1,\quad p_i>0$$

$$P(x_1,x_2,\cdots,x_k;N,p_1,p_2,\cdots,p_k)=\frac{N!}{x_1!\ x_2!\ \cdots x_k!}p_1^{x_1}p_2^{x_2}\cdots p_k^{x_k} \tag{5.8}$$

$$N=x_1+x_2+\cdots+x_k$$

(4) Beta 分布：Beta 分布是一种连续型概率密度分布。给定参数 $\alpha>0$ 和 $\beta>0$，取值范围为[0,1]的随机变量 x 的概率密度函数为：

$$p(x;\alpha,\beta)=\frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1} \tag{5.9}$$

$$B(\alpha,\beta)=\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\mathrm{d}t$$

Beta 函数相关性质：

$$B(\alpha,\beta)=\frac{(\alpha-1)!(\beta-1)!}{(\alpha+\beta-1)!}$$

函数相关性质为 $\Gamma(n+1)=n!$，因此

$$B(\alpha,\beta)=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

(5) 狄利克雷分布(Dirichlet Distribution)：狄利克雷分布是 Beta 分布在高维度上的推广，概率密度函数为：

$$p(x_1,x_2,\cdots,x_k;\alpha_1,\alpha_2,\cdots,\alpha_k)=\frac{1}{B(\alpha)}\prod_{i=1}^{k}x_i^{\alpha^i-1} \tag{5.10}$$

其中，

$$B(\alpha)=\int_0^1\prod_{i=1}^{k}t_i^{\alpha^i-1}\mathrm{d}t,\quad \sum_{x_i}=1$$

$$B(\alpha)=\int_0^1\prod_{i=1}^{k}t_i^{\alpha^i-1}\mathrm{d}t=\frac{\prod_{i=1}^{k}(\alpha^i-1)!}{\left(\sum_{i=1}^{k}\alpha^i-1\right)!}=\frac{\prod_{i=1}^{k}\Gamma(\alpha^i)}{\Gamma\left(\sum_{i=1}^{k}\alpha^i\right)}$$

上述分布的关系为：二项式分布和多项式分布类似，Beta 分布和狄利克雷分布类似，Beta 分布是二项式分布的共轭先验概率分布，而狄利克雷分布是多项式分布的共轭先验概率分布。

2. 共轭先验分布

贝叶斯公式为：

$$P(A_i \mid B)=\frac{p(B \mid A_i)p(A_i)}{\sum_{j=1}^{\infty} p(B \mid A_j)P(A_j)} \tag{5.11}$$

其中，$p(A_i)$称为先验分布，$p(A_i|B)$称为后验分布。

贝叶斯学派里的一个基本观点是：对于任何一个未知量都可以使用概率分布来描述其未知的状况。在抽样之前，基于已有的知识对于未知量的概率分布进行的预估，这在贝叶斯公式里面被称作先验分布，即式(5.11)中的 $p(A_i)$，然后再基于样本的分布情况得出后验分布，即式(5.11)中的 $p(A_i|B)$。

共轭分布的定义：在贝叶斯公式中，如果后验分布与先验分布属于同类，则先验分布与后验分布被称为共轭分布。

共轭先验分布：设 θ 是总体分布中的参数(或参数向量)，$p(\theta)$是 θ 的先验密度函数，假如由抽样信息算得的后验密度函数 $p(\theta|x)$与 $p(\theta)$有相同的函数形式，则称 $p(\theta)$是参数 θ 的共轭先验分布。

【例 5-4】 证明 Beta 分布是二项式分布的共轭先验分布。

设事件 A 发生的概率 $P(A)=\theta$，为了估计 θ 的值，进行了 n 次独立实验，其中事件 A 出现了 x 次，因此 $X\sim B(n,\theta)$。概率密度函数为：

$$p(X=x \mid \theta)=\binom{n}{x}\theta^x(1-\theta)^{n-x} \tag{5.12}$$

根据贝叶斯公式，为了得到参数 θ 的后验概率 $p(\theta|x)$，需要知道 $p(\theta)$，由于没有其他有用的信息，因此只能认为 θ 在区间[0,1]上均匀分布，即 $\theta\sim U(0,1)$。

计算联合概率分布 $p(x,\theta)$：

$$p(x,\theta)=p(x \mid \theta)p(\theta)$$

根据联合概率分布计算 $p(x)$的边缘概率分布：

$$\begin{aligned} p(x)&=\int_0^1 p(x,\theta)\mathrm{d}\theta \\ &=\int_0^1 \binom{n}{x}\theta^x(1-\theta)^{n-x}\mathrm{d}\theta \end{aligned}$$

由于 Beta 函数 $B(\alpha,\beta)=\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\mathrm{d}t$，故可推出：

$$p(x)=\binom{n}{x}B(x+1,n-x+1)$$

计算 $p(\theta|x)$：

$$p(\theta \mid x)=\frac{p(x,\theta)}{p(x)}=\frac{1}{B(x+1,n-x+1)}\theta^x(1-\theta)^{n-x}$$

即 $p(\theta|x)$满足参数为$(x+1)$和$(n-x+1)$的 β 分布，即

$$p(\theta \mid x) \sim \text{Beta}(x+1, n-x+1)$$

而先验分布 $p(\theta)$满足区间(0,1)上的均匀分布,区间(0,1)上的均匀分布是一种特殊的 β 分布,即

$$p(\theta) \sim \text{Beta}(1,1)$$

由此可见,参数 θ 的先验概率 $p(\theta)$与其后验概率 $p(\theta|x)$都属于 Beta 分布,因此 Beta 分布是二项式分布的共轭先验分布。

【例 5-5】 证明 Beta 分布是伯努利分布的共轭先验分布。

参数为 θ 的伯努利模型,其结果 x 的分布情况为:

$$P(x \mid \theta) = \theta^x (1-\theta)^{1-x}, \quad x = 0, 1$$

设参数 θ 满足参数为 α 和 β 的 Beta 分布,即

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

由于给定样本后 $P(x)$为定值,假设$\int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\mathrm{d}t$ 也为定值,计算参数 θ 的后验概率:

$$\begin{aligned} P(\theta \mid x) &= \frac{P(x \mid \theta)P(\theta)}{P(x)} \\ &\propto P(x \mid \theta)P(\theta) \\ &\propto [\theta^x (1-\theta)^{1-x}][\theta^{\alpha-1}(1-\theta)^{\beta-1}] \\ &= \theta^{x+\alpha-1}(1-\theta)^{\beta-x} \end{aligned}$$

将 $\theta^{x+\alpha-1}(1-\theta)^{\beta-x}$ 进行归一化后得:

$$P(\theta \mid x) = \frac{\theta^{x+\alpha-1}(1-\theta)^{\beta-x}}{\int_0^1 t^{x+\alpha-1}(1-t)^{\beta-x}\mathrm{d}t} \sim \text{Beta}(x+\alpha, \beta-x+1)$$

因此,若假定伯努利分布的参数 θ 的先验概率为 Beta 分布,那么其后验概率也属于 Beta 分布,因此 Beta 分布是伯努利分布的共轭先验分布。

LDA 模型中会使用到狄利克雷(Dirichlet)分布和多项式分布(Multinomial Distribution)。由于词分布和主题分布都服从多项式分布,因此 LDA 模型需要估计此多项式分布的参数。为了估计此参数,LDA 模型设它的先验分布为狄利克雷分布,而 LDA 模型中选用狄利克雷分布作为参数的先验分布的原因就是:狄利克雷分布是多项式分布的共轭先验概率分布。

【例 5-6】 证明:狄利克雷分布是多项式分布的共轭先验概率分布。

多项式分布:

$$P(x_1, x_2, \cdots, x_m; N, \alpha_1, \alpha_2, \cdots, \alpha_m) = \frac{N!}{x_1!\ x_2!\ \cdots x_m!}\alpha_1^{x_1}\alpha_2^{x_2}\cdots\alpha_k^{x_m}$$

$$N = x_1 + x_2 + \cdots + x_m, \quad \sum_{\alpha_i} = 1, \alpha_i > 0$$

设参数 $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m)$满足参数为 $\boldsymbol{k}$ 的狄利克雷分布,即参数 $\boldsymbol{\alpha}$ 的先验分布为 $\text{Dir}(\boldsymbol{\alpha} \mid \boldsymbol{k})$:

$$p(\boldsymbol{\alpha};\boldsymbol{k})=p(\alpha_1,\alpha_2,\cdots,\alpha_m;k_1,k_2,\cdots,k_m)=\frac{\prod_{i=1}^{m}\alpha_i^{k_i-1}}{B(\boldsymbol{k})},\quad \sum_{i=1}^{m}k_i=1$$

计算参数$\boldsymbol{\alpha}$的后验概率：

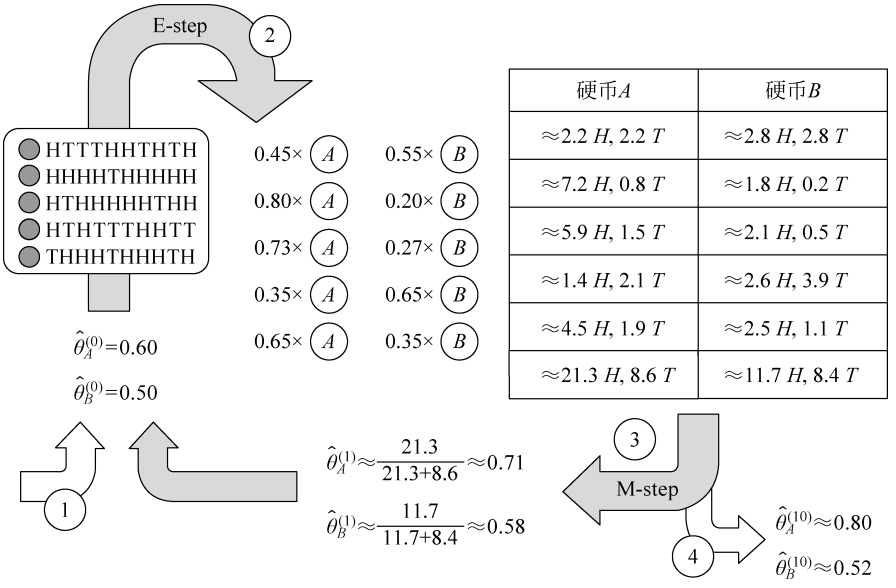
$$\begin{aligned}P(\boldsymbol{\alpha}\mid\boldsymbol{x})&=\frac{P(\boldsymbol{x}\mid\boldsymbol{\alpha})P(\boldsymbol{\alpha})}{P(\boldsymbol{x})}\\&\propto P(\boldsymbol{x}\mid\boldsymbol{\alpha})P(\boldsymbol{\alpha};\boldsymbol{k})\\&=\left[\frac{N!}{x_1!\ x_2!\ \cdots x_m!}\alpha_1^{x_1}\alpha_2^{x_2}\cdots\alpha_k^{x_m}\right]\left[\frac{\prod_{i=1}^{m}\alpha_i^{k_i-1}}{B(\boldsymbol{k})}\right]\\&\propto\left[\alpha_1^{x_1}\alpha_2^{x_2}\cdots\alpha_k^{x_m}\right]\left[\prod_{i=1}^{m}\alpha_i^{k_i-1}\right]\\&=\prod_{i=1}^{m}\alpha_i^{x_i+k_i-1}\end{aligned}$$

进行归一化处理后$P(\boldsymbol{\alpha}\mid\boldsymbol{x})=\frac{\prod_{i=1}^{m}\alpha_i^{x_i+k_i-1}}{B(\boldsymbol{x}+\boldsymbol{k})}$，即参数$\boldsymbol{\alpha}$的后验概率满足$\mathrm{Dir}(\boldsymbol{\alpha}\mid\boldsymbol{k}+\boldsymbol{x})$

综上所述，多项式分布的参数的先验概率为狄利克雷分布时，其后验概率也为狄利克雷分布，因此狄利克雷分布是多项式分布的共轭先验概率分布。

3. EM 算法

EM 算法是一种迭代算法，用于含有隐含变量的概率模型参数的极大似然估计或极大后验估计。Nature Biotech 在他的文章 *What is the expectation maximization algorithm?*[5] 中，用了以下投硬币的例子讲述了 EM 算法的思想，如图 5-3 所示。



硬币A	硬币B
≈2.2 H, 2.2 T	≈2.8 H, 2.8 T
≈7.2 H, 0.8 T	≈1.8 H, 0.2 T
≈5.9 H, 1.5 T	≈2.1 H, 0.5 T
≈1.4 H, 2.1 T	≈2.6 H, 3.9 T
≈4.5 H, 1.9 T	≈2.5 H, 1.1 T
≈21.3 H, 8.6 T	≈11.7 H, 8.4 T

图 5-3　EM 算法估计参数

第一步，为硬币 A 和硬币 B 的参数 θ_A 和 θ_B 分别赋值为 0.6 和 0.5，这是根据经验人为设定的值。

第二步(E-step)，根据 θ_A 和 θ_B 的估计值判断第一轮至五轮分别抛的是哪个硬币，例如，对于第一轮抛硬币的结果服从二项式分布：

$$P(X=k)=\binom{n}{k}p^k(1-p)^{n-k}$$

硬币 A 抛出 5 正 5 反的概率：

$$P_A(X=5)=\binom{10}{5}0.6^5(1-0.6)^{10-5}$$

硬币 B 抛出 5 正 5 反的概率：

$$P_B(X=5)=\binom{10}{5}0.5^5(1-0.5)^{10-5}$$

对结果进行归一化后可得，第一次抛硬币抛的是硬币 A 的概率：

$$P(A)=\frac{P_A(X=5)}{0P_A(X=5)+P_B(X=5)}\approx 0.449$$

抛的是硬币 B 的概率：

$$P(B)=\frac{P_B(X=5)}{0P_A(X=5)+P_B(X=5)}\approx 0.551$$

将 $P(A)$和 $P(B)$作为权重，对于硬币 A：

$$0.449\times(5H,5T)\approx(2.2H,2.2T)$$

对于硬币 B：

$$0.551\times(5H,5T)\approx(2.8H,2.8T)$$

采用同样的方法对剩下的 4 轮进行计算。

第三步(M-step)，对于硬币 A，将其相应的 H 系数与 T 系数分别相加，即为硬币 A 在抛硬币过程中抛得正面和反面的频数，对于硬币 B 同理。

已知硬币 A 和硬币 B 抛得正面和反面的频数后就可进一步求出新的 θ_A 和 θ_B，例如，硬币 A 抛得的正面的频数(H 数相加)：

$$2.2+7.2+5.9+1.4+4.5=21.2$$

$$2.2+0.8+1.5+2.1+1.9=8.5$$

θ_A 新的估计值为：

$$\frac{21.2}{21.2+8.5}=0.71$$

同理，可得 θ_B 新的估计值为 0.58。

迭代进行第二步和第三步，直至 θ_A 和 θ_B 收敛，在这个例子中 θ_A 和 θ_B 收敛于 0.8 和 0.52，即 θ_A 和 θ_B 最终的估计结果为 0.8 和 0.52。

5.3.2 PLSA 模型

1. PLSA 模型简介

PLSA 模型中的相关定义如下。

(1) $P(d_i)$表示文档 d_i 被选中的概率。

(2) $P(w_j|d_i)$表示词 w_j 在文档 d_i 中出现的概率,其数值等于词 w_j 在文档 d_i 中出现的概率。

(3) $P(z_k|d_i)$表示主题 z_k 在文档 d_i 中出现的概率。

(4) $P(w_j|z_k)$表示词 w_j 在主题 z_k 下出现的概率,词 w_j 与主题 z_k 越相关概率值越大。

(5) PLSA 是一种词袋模型,不考虑词与词之间出现的顺序。

在 PLSA 模型中,为文档 d_i 抽取一个单词的过程如下。

(1) 根据概率分布 $P(z|d_i)$,为文档 d_i 抽取一个主题 z_k。

(2) 根据概率分布 $P(w|z_k)$,在主题 z_k 下抽取一个词语 w_j。

重复步骤(1)和(2),直至生成文档 d_i 中所需的 N 个词。

上述过程每篇文档的主题分布 $P(z|d_i)$和每个主题下的词分布 $P(w|z_k)$即为 PLSA 模中需要求的参数。

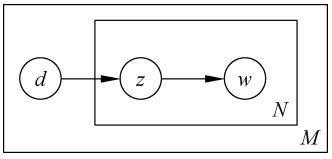


图 5-4　PLSA 模型生成文档

PLSA 模型生成文档的过程可由图 5-4 表示。

图 5-4 中的 d 代表文档,w 代表词语,z 代表主题,M 代表文档数量,N 代表文档中的单词数量。图 5-4 中的 d 和 w 为可以人为观测到的数据,而图 5-4 中的 z 不可观测,称之为隐含变量。图 5-4 中的方框代表重复,方框右下角的值代表重复的次数。

图 5-4 所描述的过程如下。

For $i=1,2,3,\cdots,M$:
　　选择一篇文档;
　　For $j=1,2,3,\cdots,N$:
　　　　根据当前文档的主题分布 $P(z|d_i)$,为当前文档选择一个主题;
　　　　根据选定的主题下的词分布 $P(w|z_k)$,选择一个词;

将图 5-4 中所描述的过程做进一步抽象:

给定文档 d_i,词语 w_j 出现的概率为

$$P(w_j \mid d_i)=\sum_{k=1}^{K}P(w_j \mid z_k)P(z_k \mid d_i) \tag{5.13}$$

w_j 和 d_i 的联合概率分布为

$$\begin{aligned}P(w_j,d_i)&=P(d_i)P(w_j \mid d_i)\\&=P(d_i)\sum_{k=1}^{K}P(w_j \mid z_k)P(z_k \mid d_i)\end{aligned} \tag{5.14}$$

其中,$P(w_j,d_i)$和 $P(d_i)$可以由数据集中直接求得,而 $P(w_j|z_k)$和 $P(z_k|d_i)$为未知值,是 PLSA 模型中要估计的参数。

2. PLSA 模型的参数估计方法

PLSA 模型采用 EM 算法对参数 $P(w|z_k)$和 $P(z|d_i)$进行估计。EM 算法用于含有

隐含变量的概率模型参数的极大似然估计或极大后验估计。在 PLSA 模型中，主题分布即为 EM 算法中所说的隐含变量。

利用 EM 算法求解 PLSA 模型的参数过程如下。

PLAS 中需要估计的参数为 $P(w_j|z_k)$和 $P(z_k|d_i)$，即需要计算得到文档的主题分布和每个主题下的词分布。设主题在文档 d_i 上服从参数为 θ_i 的多项式分布，其中 $\theta_{i,k}$ 表示主题 z_k 在文档 d_i 中出现的概率，即

$$P(z_k \mid d_i)=\theta_{i,k},\quad \sum_{k=1}^{K}\theta_{i,k}=1 \tag{5.15}$$

设单词在主题 z_k 上服从参数为 α_k 的多项式分布，其中 $\alpha_{k,j}$ 表示单词 w_j 在主题 z_k 中出现的概率，即

$$P(w_j \mid z_k)=\alpha_{k,j},\quad \sum_{k=1}^{K}\alpha_{k,j}=1 \tag{5.16}$$

有了上述定义后，模型所需求解的参数用矩阵可以表示为：

$$\boldsymbol{\Theta}=[\theta_1,\theta_2,\cdots,\theta_M],\quad M \text{ 为文档数量}$$

$$\boldsymbol{A}=[\alpha_1,\alpha_2,\cdots,\alpha_K],\quad K \text{ 为主题数量}$$

$\boldsymbol{\Theta}$ 中包含每篇文档的主题分布，$\boldsymbol{A}$ 中包含每个主题的词分布。

整个语料库的词分布为：

$$P(W \mid D)=\prod_{i=1}^{M}\prod_{j=1}^{N}P(d_i,w_j)^{n_{i,j}} \tag{5.17}$$

其中，$n_{i,j}$ 表示单词 w_j 在文档 d_i 中出现的次数，则文档 d_i 中单词的个数为 $n_i=\sum_{w_j\in V}n_{i,j}$

为了计算得到 $P(W|D)$，需得到文档和词语的联合概率分布：

$$\begin{aligned}P(d_i,w_j)&=P(d_i)P(w_j \mid d_i)\\&=P(d_i)\sum_{k=1}^{K}P(w_j \mid z_k)P(z_k \mid d_i)\\&=P(d_i)\sum_{k=1}^{K}\alpha_{k,j}\theta_{i,k}\end{aligned} \tag{5.18}$$

参数$\boldsymbol{\Theta}$ 和 $\boldsymbol{A}$ 的对数似然函数为：

$$\begin{aligned}l(\boldsymbol{\Theta},\boldsymbol{A})&=\log P(W \mid D)\\&=\sum_{i=1}^{M}\sum_{j=1}^{N}n_{i,j}\log P(d_i,w_j)\\&=\sum_{i=1}^{M}\sum_{j=1}^{N}n_{i,j}\log P(d_i,w_j)\\&=\sum_{i=1}^{M}\sum_{j=1}^{N}n_{i,j}\log\left[P(d_i)\sum_{k=1}^{K}\alpha_{k,j}\theta_{i,k}\right]\\&=\sum_{i=1}^{M}\sum_{j=1}^{N}n_{i,j}\left[\log P(d_i)+\log\sum_{k=1}^{K}\alpha_{k,j}\theta_{i,k}\right]\\&=\sum_{i=1}^{M}\sum_{j=1}^{N}\left[n_{i,j}\log P(d_i)+n_{i,j}\log\sum_{k=1}^{K}\alpha_{k,j}\theta_{i,k}\right]\end{aligned}$$

$$= \sum_{i=1}^{M} n_i \log P(d_i) + \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \log \sum_{k=1}^{K} \alpha_{k,j} \theta_{i,k} \tag{5.19}$$

由于 $\sum_{i=1}^{M} n_i \log P(d_i)$ 中没有涉及参数$\boldsymbol{\Theta}$和$\boldsymbol{A}$，因此删去也不会影响似然函数的计算。因此：

$$l(\boldsymbol{\Theta}, \boldsymbol{A}) = \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \log \sum_{k=1}^{K} \alpha_{k,j} \theta_{i,k} \tag{5.20}$$

由于式(5.20)中存在和的对数，计算较为麻烦，可以使用 Jensen 不等式对似然函数进行进一步的简化。Jensen 不等式可以表述为：

$$\begin{aligned} \sum_i \log\ p(x^{(i)}; \theta) &= \sum_i \log \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \\ &= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \\ &\geqslant \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \end{aligned}$$

其中，

$$Q_i(z^{(i)}) = \frac{p(x^{(i)}, z^{(i)}; \theta)}{\sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta)}$$

根据 Jensen 不等式，似然函数可以进一步简化为：

$$Q_i(z) = \frac{\alpha_{k,j} \theta_{i,k}}{\sum_{k=1}^{K} \alpha_{k,j} \theta_{i,k}} = P(z_k \mid d_i, w_j) \tag{5.21}$$

$$\begin{aligned} l(\boldsymbol{\Theta}, \boldsymbol{A}) &= \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \log \sum_{k=1}^{K} \alpha_{k,j} \theta_{i,k} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \log \sum_{k=1}^{K} Q_i(z) \frac{\alpha_{k,j} \theta_{i,k}}{Q_i(z)} \\ &\geqslant \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \sum_{k=1}^{K} Q_i(z) \log \frac{\alpha_{k,j} \theta_{i,k}}{Q_i(z)} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log \frac{\alpha_{k,j} \theta_{i,k}}{P(z_k \mid d_i, w_j)} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \sum_{k=1}^{K} P(z_k \mid d_i, w_j) (\log \alpha_{k,j} \theta_{i,k} - \log P(z_k \mid d_i, w_j)) \end{aligned}$$

由于 $\log P(z_k \mid d_i, w_j)$中涉及参数$\boldsymbol{\Theta}$和$\boldsymbol{A}$，且在给定样本后它为定值，因此删去也不会影响似然函数的计算，因此似然函数转化为：

$$l(\boldsymbol{\Theta}, \boldsymbol{A}) = \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log \alpha_{k,j} \theta_{i,k} \tag{5.22}$$

下面采用 EM 算法来估算参数，首先根据经验为参数$\boldsymbol{\Theta}$、$\boldsymbol{A}$ 赋予一个初始的估计值，随后进行 EM 算法的步骤：E-step 和 M-step。

(1) E-step 计算隐含变量的后验概率

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

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

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

$$= \frac{\alpha_{k,j}\theta_{i,k}}{\sum_{l=1}^{K} \alpha_{l,j}\theta_{i,l}}$$

(2) M-step 实现最大化似然函数,并将 E-step 求出的后验概率值代入$\boldsymbol{\Theta}$、$\boldsymbol{A}$ 的表达式,求出相应参数的解。首先,似然函数为:

$$l(\boldsymbol{\Theta}, \boldsymbol{A}) = \sum_{i=1}^{M} \sum_{j=1}^{N} n_{i,j} \sum_{k=1}^{K} P(z_k \mid d_i, w_j) \log \alpha_{k,j}\theta_{i,k}$$

通过最大化似然函数可得:

$$\theta_{i,k} = \frac{\sum_{j=1}^{N} n_{i,j} P(z_k \mid d_i, w_j)}{n_i} \tag{5.23}$$

$$\alpha_{k,j} = \frac{\sum_{i=1}^{M} n_{i,j} P(z_k \mid d_i, w_j)}{\sum_{n=1}^{N} \sum_{i=1}^{M} n_{i,m} P(z_k \mid d_i, w_j)} \tag{5.24}$$

将 E-step 求得的 $P(z_k|d_i, w_j)$的值代入 $\theta_{i,k}$ 和 $\alpha_{k,j}$ 的表达式,可以得到新的 $\theta_{i,k}$ 和 $\alpha_{k,j}$ 值。

迭代 E-step 和 M-step,直至参数$\boldsymbol{\Theta}$、$\boldsymbol{A}$ 的值收敛。收敛后参数$\boldsymbol{\Theta}$、$\boldsymbol{A}$ 的值即为模型所求的参数值。

5.3.3 LDA 模型

LDA(Latent Dirichlet Allocation)是一个生成式概率模型,它与 PLSA 模型十分相似。对于一篇文档,LDA 和 PLSA 都会估计样本的主题分布和词分布,不同的是 PLSA 模型认为主题的分布和词的分布是未知的,但值是固定的,也就是说,PLSA 模型得到的 $P(z_k|d_i)$和 $P(w_j|z_k)$对应两个固定的实数值。而 LDA 模型则认为主题分布和词分布为服从某一分布的随机变量,也就是说,无论 $P(z_k|d_i)$和 $P(w_j|z_k)$是不是固定的值,但它们的值服从一定的分布。LAD 模型可以看作在 PLSA 模型的基础上融合了贝叶斯框架得到的。

【例 5-7】 假设有如下主题“财经”“娱乐”“科技”“体育”“教育”,PLSA 模型和 LDA 模型在处理主题分布上有何不同?

设“财经”“娱乐”“科技”“体育”“教育”这几个主题在文档中出现的概率$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3,$

θ_4,θ_5)，文档 d_i 的抽样结果满足以$\boldsymbol{\theta}$ 为参数的多项式分布，即

$$z_i \sim \mathrm{Mul}(\boldsymbol{\theta})$$

PLSA 模型和 LDA 模型的目标都是获得$\boldsymbol{\theta}$，但是它们分别采用了两种方式获得$\boldsymbol{\theta}$，PLSA 模型认为各个主题出现的概率$\boldsymbol{\theta}$ 虽然未知但都为固定值，比如，经过计算后可以得到主题“财经”“娱乐”“科技”“体育”“教育”在文档中出现的概率$\boldsymbol{\theta}=(\theta_1,\theta_2,\theta_3,\theta_4,\theta_5)=(0.2,0.05,0.4,0.05,0.3)$，即每个主题都会对应一个固定的概率值。

LDA 模型则认为各个主题出现的概率$\boldsymbol{\theta}$ 是服从某一分布的随机变量。主题的抽样结果服从以$\boldsymbol{\theta}$ 为参数的多项式分布，且多项式分布的共轭先验分布为狄利克雷分布，因此设$\boldsymbol{\theta}$ 服从狄利克雷分布，即$\boldsymbol{\theta}\sim\mathrm{Dir}(\boldsymbol{\alpha})$。也就是说，在 LDA 模型中最终不会计算得到$\boldsymbol{\theta}$ 的具体值，而是计算得到$\boldsymbol{\theta}$ 的分布。

1. LDA 模型简介

LDA 将词分布和主题分布看作服从某一分布的随机变量。由于词分布和主题分布本身是服从多项式分布的，而前文中证明过狄利克雷分布是多项式分布的共轭先验分布，因此在求词分布和主题分布的过程中，首先人为设定词分布和主题分布的先验分布为狄利克雷分布，然后根据已知数据计算词分布和主题分布的后验分布，这个后验分布即为 LDA 模型所要求的内容。

LDA 模型抽取一个词语的过程如下。

(1) 从狄利克雷分布 $\mathrm{Dir}(\alpha)$中取样生成文档 i 的主题分布中相应的参数 θ_i。

(2) 主题服从参数为 θ_i 的多项式分布，即 $z_i\sim\mathrm{Multinomial}(\theta_i)$，从中取样生成文档 i 的第 k 个主题 $z_{i,k}$。

(3) 从狄利克雷分布 $\mathrm{Dir}(\beta)$中取样生成主题 $z_{i,k}$ 的词分布中相应的参数 ϕ_k。

(4) 词语服从参数为 ϕ_k 的多项式分布，即 $w_k\sim\mathrm{Multinomial}(\phi_k)$，从中取样生成单词 $w_{k,j}$。

对于一篇包含有 N 个词语的文档(N 服从泊松分布)，重复上述步骤 N 次直至生成文档所需的 N 个词。

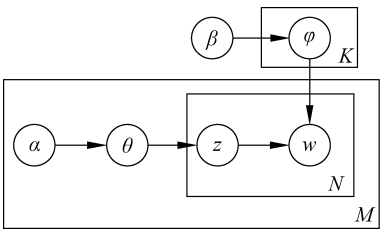


图 5-5　LDA 模型生成文档

上述过程与 PLSA 模型中生成文档的过程相比，LDA 在 PLSA 的基础上，为主题分布和词分布各加了一个狄利克雷先验分布。LDA 模型将主题分布和词分布当作先验分布为狄利克雷分布的随机变量，最终 LDA 模型会基于已知数据，计算主题分布和词分布的后验概率，即参数 θ 和参数 β 是 LDA 模型中需要估计的参数。

LDA 模型生成文档的过程可由图 5-5 表示。

图 5-5 中 θ 是满足参数为 α 的狄利克雷分布的随机变量：

$$p(\theta_1,\theta_2,\cdots,\theta_k;\alpha_1,\alpha_2,\cdots,\alpha_k)=\frac{\Gamma\left(\sum_{i=1}^{k}\alpha^i\right)}{\prod_{i=1}^{k}\Gamma(\alpha^i)}\prod_{i=1}^{k}\theta_i^{\alpha^i-1},\quad \sum\theta_i=1 \tag{5.25}$$

文档的边缘分布为：

$$p(\boldsymbol{w} \mid \alpha,\beta)=\int p(\theta \mid \alpha)\left(\prod_{n=1}^{N}\sum_{z_k} p(z_k \mid \theta)p(w_n \mid z_k,\beta)\right)\mathrm{d}\theta \tag{5.26}$$

其中，$p(w|z_k,\beta)$为主题 z_k 相应的词分布，此分布的参数由以 β 为参数的狄利克雷分布中抽样得到。

语料库中共包含 M 篇文档，因此语料库的边缘分布为：

$$p(\boldsymbol{D} \mid \alpha,\beta)=\sum_{d=1}^{M}\int p(\theta_d \mid \alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn} \mid \theta_d)p(w_{dn} \mid z_{dn},\beta)\right)\mathrm{d}\theta_d \tag{5.27}$$

2. LDA 模型参数估计方法

LDA 模型采用变分 EM 算法对参数进行估计，变分 EM 算法即变分推断和 EM 算法的结合，其基本思路与 PLSA 中参数的求解方法类似。

若要根据已有数据推断分布 P，当 P 不容易直接求解时，可以使用变分推断的方法，即寻找容易求解的分布 Q，使分布 Q 不断逼近 P，当分布 Q 足够逼近 P 时，可以将 Q 作为 P 的近似分布。

在 LDA 模型中，希望找到合适的 α 和 β 使对似然函数最大化，并求出隐含变量的条件概率分布。但由于隐含变量 α 和 β 之间存在耦合关系，使用 EM 算法时 E-step 无法直接求解它们基于条件概率分布的期望，且隐含变量 α 和 β 的分布很难直接求得，因此使用变分法，假设所有的隐含变量都是通过各自独立的分布生成的，即去掉隐含变量之间的连线和 w 节点，并赋予 θ、z、φ 各自独立分布，γ、η、ρ 为相应的变分参数，引入变分参数的目的是增加各个隐含变量之间的独立性，如图 5-6 所示。

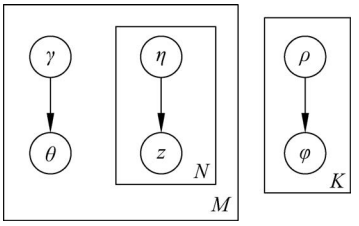


图 5-6　增加了变分参数后的 LDA 模型

设模型的联合概率分布为 $p(x,z)$，其中 x 为观测变量，z 为隐含变量，求后验概率 $p(z|x)$。变分 EM 算法用概率分布 $Q(z)$来近似条件分布概率 $p(z|x)$，之后用 KL 散度 $\mathrm{KL}(Q(z)\|p(z|x))$计算两者之间的相似度，$Q(z)$和 $p(z|x)$越相似 KL 散度越小，因此为了让变分分布 $Q(z)$能够近似地表示真实后验 $p(z|x)$，需要让二者的 KL 散度尽可能小。KL 散度的计算方法如下：

$$\begin{aligned}\mathrm{KL}(Q(z)\|p(z \mid x)) &= \sum_z Q(z)\log\frac{Q(z)}{p(z \mid x)}\\ &= \sum_z Q(z)\log Q(z)-\sum_z Q(z)\log p(z,x)+\log p(x)\\ &= \log p(x)-\{E_Q[\log p(x,z)]-E_Q[\log Q(z)]\}\end{aligned} \tag{5.28}$$

KL 散度的值大于或等于 0，当且仅当两个分布一致时 KL 散度为 0，即

$$\log p(x)\geqslant E_Q[\log p(x,z)]-E_Q[\log Q(z)]$$

由于 x 为观测值，所以在给定样本的情况下，$p(x)$为定值，因此可以设：

$$L=E_Q[\log p(x,z)]-E_Q[\log Q(z)] \tag{5.29}$$

称 $E_Q[\log p(x,z)]-E_Q[\log Q(z)]$为证据下界，可以通过最大化证据下界来求取 $Q(z)$。

图 5-7 中的 w 为可观测变量，α 和 β 为参数，θ、φ、z 为隐含变量，图中的虚线圆圈对应的 γ、η、ρ 为变分参数。

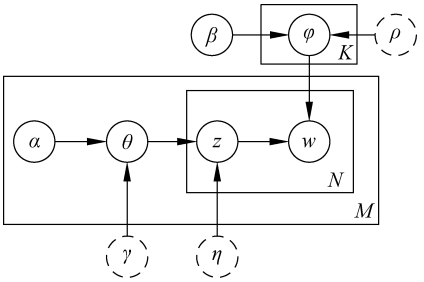


图 5-7 LDA 模型

随机变量 θ、z、w、φ 的联合概率分布为：

$$p(\theta,z,w,\varphi \mid \alpha,\beta)=p(\theta \mid \alpha)p(z \mid \theta)p(\varphi \mid \beta)p(w \mid z,\varphi)$$

隐含变量的后验分布为：

$$\begin{aligned}Q(\theta,z,\varphi \mid \gamma,\eta,\rho)&=Q(\theta \mid \gamma)Q(z \mid \eta)Q(\varphi \mid \rho)\\ L(\gamma,\eta,\rho,\alpha,\beta)&=E_Q[p(\theta,z,w,\varphi \mid \alpha,\beta)]-E_Q[Q(\theta,z,\varphi \mid \gamma,\eta,\rho)]\\ &=E_Q[p(\theta \mid \alpha)p(z \mid \theta)p(\varphi \mid \beta)p(w \mid z,\varphi)]-\\ &\quad E_Q[Q(\theta \mid \gamma)Q(z \mid \eta)Q(\varphi \mid \rho)]\\ &=E_Q[p(\theta \mid \alpha)]+E_Q[p(z \mid \theta)]+E_Q[p(\varphi \mid \beta)]+E_Q[p(w \mid z,\varphi)]-\\ &\quad E_Q[Q(\theta \mid \gamma)]+E_Q[Q(z \mid \eta)]+E_Q[Q(\varphi \mid \rho)]\end{aligned} \tag{5.30}$$

EM 算法求解过程如下。

(1) E-step：求解变分参数 γ、η、ρ。求解参数 γ 的过程如下：首先提取出 $L(\gamma,\eta,\rho,\alpha,\beta)$中含有 γ 的部分记为$L(\gamma)$；对提取出的部分对 γ 求偏导，即求$\frac{\partial L(\gamma)}{\partial \gamma}$；令偏导等于 0，即$\frac{\partial L(\gamma)}{\partial \gamma}=0$，求出 γ。变分参数 η 和 ρ 的求法与上述 γ 的求法类似，都先从 $L(\gamma,\eta,\rho,\alpha,\beta)$提取相关内容，然后求偏导，最后令偏导的值为零，求出参数值。

(2) M-step：求模型的参数 α、β。以参数 α 为例，提取出 $L(\gamma,\eta,\rho,\alpha,\beta)$中含有 α 的部分

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

其中，Ψ 是 digamma 函数，是对数伽马函数的一阶导数。对 α_k 求偏导：

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

再对 α_l 求偏导：

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

对参数 α 的值进行更新：

$$\alpha_{\text{new}}=\alpha_{\text{old}}-H(\alpha_{\text{old}})^{-1}g(\alpha_{\text{old}})$$

参数 β 的更新过程与 α 的类似：

$$\beta_{new} = \beta_{old} - H(\beta_{old})^{-1} g(\beta_{old})$$

迭代 E-step 和 M-step，直至参数 α 和 β 的值收敛。收敛后参数 α 和 β 的值即为模型所求的主题分布和词分布的后验分布。

参考文献

[1] Mihalcea R，Tarau P. TextRank：Bringing order into texts[C]//Conference on Empirical Methods in Natural Language Processing，2004.

[2] Brin S. The anatomy of a large-scale hypertextual web search engine[C]//7th World Wide Web Conference. Brisbane，1998.

[3] Blei D M，Ng A，Jordan M I. Latent dirichlet allocation[J]. The Journal of Machine Learning Research，2003(3)：993-1022.

[4] Hofmann T，Hofmann T. Unsupervised learning by probabilistic latent semantic analysis[J]. Machine Learning，2001，42(1-2)：177-196.

[5] Do C B，Batzoglou S. What is the expectation maximization algorithm?[J]. Nature Biotechnology，2008，26(8)：897-901.