主题模型

一眼看穿世界

七月在线 加号
微博：@翻滚吧_加号

主要内容

- 主题模型理论
  - 直观版
  - 标准版
  - 公式版
- 实战
  - 一眼看穿『希拉里邮件门』

什么是主体模型？

理论解释

理解整个过程，涉及到比较复杂数学推导。
一般来说，从公式1一直推导到公式100，
大部分同学会在公式10左右的时候，就关了直播，洗洗睡了

所以，我今天用3个不同版本的讲解，从简单到复杂，
来让大家一步步理解主体模型。

据我推测，大部分人是可以撑过前两个版本的。
这样，就算第三个版本太过枯燥，你也可以安心的洗洗睡，无妨。

么么

直观版

假设某企业想要招聘一个工程师，
他们收到了一把的简历，
他们想直接通过简历来看谁是大牛，谁是彩笔

直观版

简历里通常会包含这些个人特征：

上过七月在线课

面试满分

参加过七月在线的线上竞赛

百度实习

只穿杰克琼斯

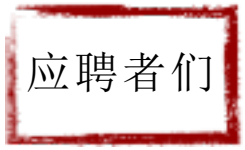
阿里实习

长相捉急

腾讯实习

着装有品位

融了一堆资的忽悠大师



条纹衬衫

逃离创业泡沫的CTO

笔试满分

中国top2毕业

自称top3毕业

直观版

这三个要素，构成了这家企业的人力总监判断的基础：

直观版

这位人力总监要做的事是：

拿出一份份简历
记录下每份简历包含的特征
然而，他并不知道，这一切代表着什么

于是他开始猜

拿起一份简历A，
他看到里面说A参加过七月课程
他就猜这位童鞋的水平应该很高，八成应该是个好工程师
但是他又看到A的学历只是小学毕业，心里又有了两成的担忧

他又看到B
又看到C
等等。。。

直观版

当然，这个猜，只是猜，没有任何证据可以证实。
但是这位人力总监是久经职场的老司机，他通过经验统计来调整自己的猜想：

- 选一份『张三』的简历，选一个特征『条纹衬衫』
- 为什么『张三』有可能喜欢穿『条纹衬衫』？也许是因为穿条纹衬衫是优秀程序员的信仰
- 也就是说，越多的优秀程序员穿『条纹衬衫』，越让人力总监猜想『张三』的其他个人特征也符合优秀程序员的喜好，并且『张三』本人穿『条纹衬衫』是一个优秀的程序员自我修养的体现
- 继续猜，继续拿『张三』和『条纹衬衫』两个元素。人力总监转念一想，也有可能爱穿条纹的都是彩笔。
- 于是他按照上面的逻辑，再看看『张三』穿『条纹衬衫』是『彩笔』的可能性有多少
- 把所有的简历和所有的特征都做个组合，都来猜一下是彩笔还是大牛。

直观版

久经沙场之后，老司机人力总监掌握了如下信息：

对于是不是优秀程序员的分类，它通过人头统计大概有了数

这让他在以后看到新简历的时候，一眼就知道他是不是个优秀程序员

对于每个特征C，他也能说出大概百分之多少的人拥有特征C可以说明他们是优秀的程序员。

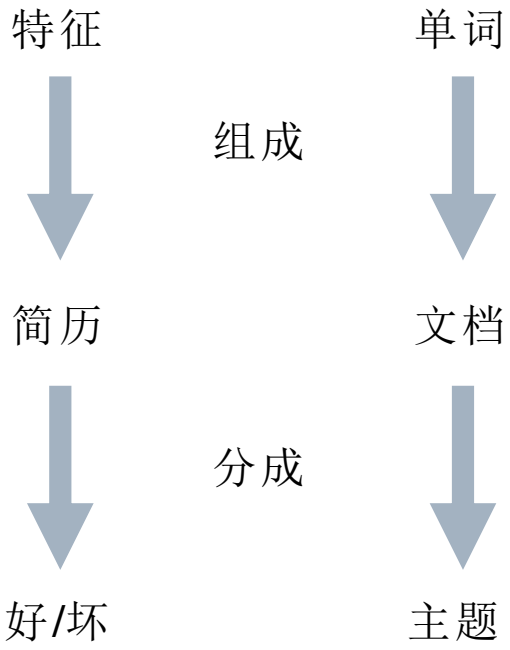
直观版

总结成公式就是：

$$P(\text{优秀程序员} \mid \text{特征，简历}) = \frac{\text{此特征在优秀程序员之中出现的次数}}{\text{优秀程序员拥有的所有特征}} \times \text{此简历中属于优秀程序员的特征个数}$$

例子与理论的关系

以上，就是我们用现实的例子模拟的LDA模型来区分简历好坏
在文本的主题分类中，我们的例子和实际之间的联系如下：



什么是LDA？

Latent Dirichlet Allocation:

是一种无监督的贝叶斯模型

是一种主题模型，它可以将文档集中每篇文档的主题按照概率分布的形式给出。同时它是一种无监督学习算法，在训练时不需要手工标注的训练集，需要的仅仅是文档集以及指定主题的数量k即可。此外LDA的另一个优点则是，对于每一个主题均可找出一些词语来描述它。

是一种典型的词袋模型，即它认为一篇文档是由一组词构成的一个集合，词与词之间没有顺序以及先后的关系。一篇文档可以包含多个主题，文档中每一个词都由其中的一个主题生成。

— wikipedia

什么是贝叶斯模型？

理论

$$P(\theta, y) = P(\theta)P(y \mid \theta)$$

$$P(\theta \mid y) = \frac{P(\theta, y)}{P(y)} = \frac{P(y \mid \theta)P(\theta)}{P(y)}$$

模型

- 用概率作为『可信度』
- 每次看到新数据，就更新『可信度』
- 需要一个模型来解释数据的生成

先验，后验与似然

P(好工程师 | 简历) = P(好工程师) P(简历 | 好工程师)

后验 *先验* *似然*

应聘者概率模型　　简历数据生成模型

（具体解释，待我白板撸，做不动PPT了。。。）

标准版

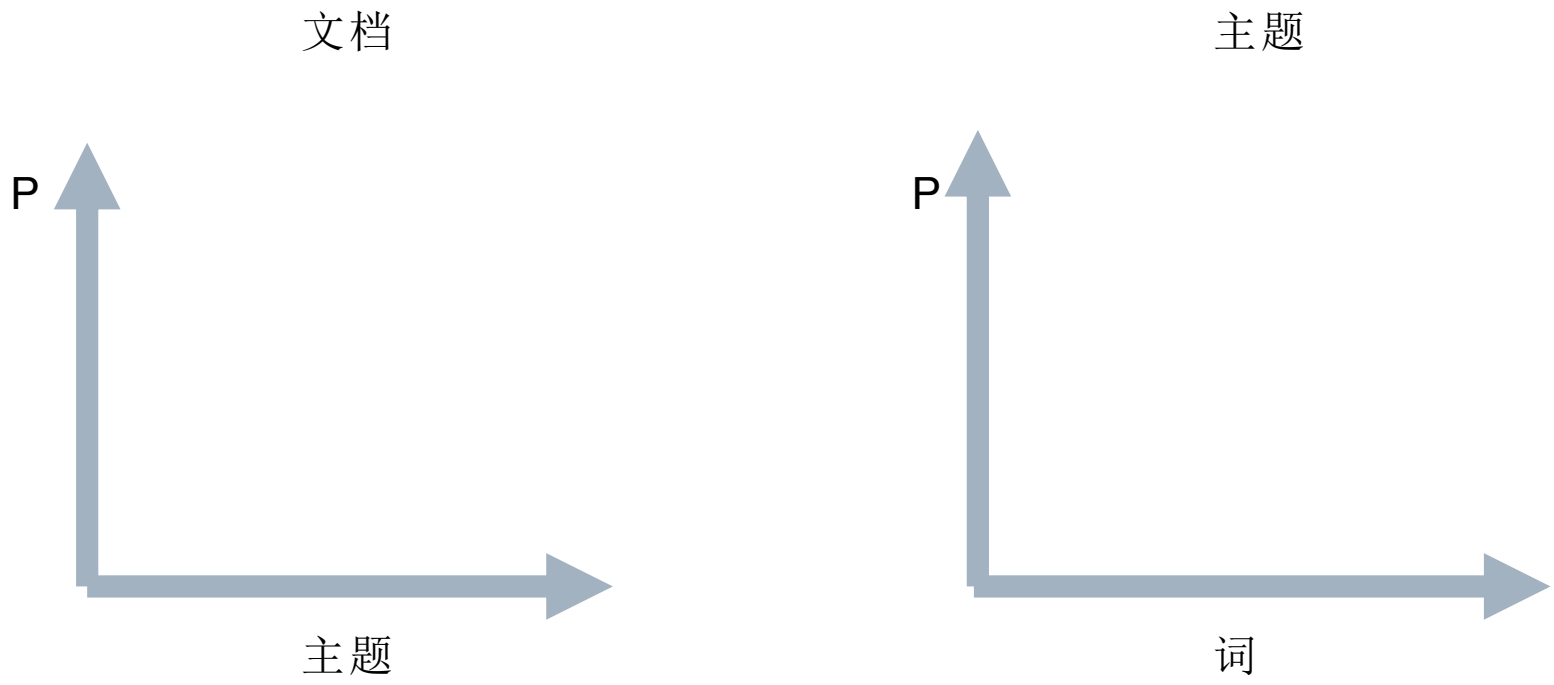
我们用LDA找寻的就是之前例子里总监大人统计出来的经验：

一份简历的每个特征都是因为本人有一定概率是好/坏程序员，并从好/坏这个分类中以一定概率选择某些特征而组成的

一篇文章的每个词都是以一定概率选择了某个主题，并从这个主题中以一定概率选择某个词语而组成的

P(单词 | 文档) = P(单词 | 主题) * P(主题 | 文档)

标准版



标准版

LDA生成过程

对于语料库中的每篇文档，LDA定义了如下生成过程（generative process）：

1.对每一篇文档，从主题分布中抽取一个主题；

2.从上述被抽到的主题所对应的单词分布中抽取一个单词；

3.重复上述过程直至遍历文档中的每一个单词。

标准版

稍微具体点儿讲：
（w代表单词；d代表文档；t代表主题；大写代表总集合，小写代表个体。）

D中每个文档d看作一个单词序列<w1,w2,...,wn>，wi表示第i个单词。

D中涉及的所有不同单词组成一个词汇表大集合V (vocabulary)，LDA以文档集合D作为输入，希望训练出的两个结果向量（假设形成k个topic，V中一共m个词）：

+ 对每个D中的文档d，对应到不同Topic的概率θd<pt1,...,ptk>，其中，pti表示d对应T中第i个topic的概率。计算方法是直观的，pti=nti/n，其中nti表示d中对应第i个topic的词的数目，n是d中所有词的总数。

+ 对每个T中的topict，生成不同单词的概率φt<pw1,...,pwm>，其中，pwi表示t生成V中第i个单词的概率。计算方法同样很直观，pwi=Nwi/N，其中Nwi表示对应到topict的V中第i个单词的数目，N表示所有对应到topict的单词总数。

标准版

所以，LDA的核心公式如下：

P(w|d)=P(w|t)*P(t|d)

直观的看这个公式，就是以Topic作为中间层，可以通过当前的θd和φt给出了文档d中出现单词w的概率。其中p(t|d)利用θd计算得到，p(w|t)利用φt计算得到。

实际上，利用当前的θd和φt，我们可以为一个文档中的一个单词计算它对应任意一个Topic时的p(w|d)，然后根据这些结果来更新这个词应该对应的topic。然后，如果这个更新改变了这个单词所对应的Topic，就会反过来影响θd和φt

标准版

LDA学习过程

LDA算法开始时，先随机地给θd和φt赋值（对所有的d和t）。然后：

1.针对一个特定的文档ds中的第i单词wi，如果令该单词对应的topic为tj，可以把上述公式改写为：**$P_j(w_i|d_s)=P(w_i|t_j)*P(t_j|d_s)$**

2.现在我们可以枚举T中的topic，得到所有的pj(wi|ds)。然后可以根据这些概率值结果为ds中的第i个单词wi选择一个topic。最简单的想法是取令pj(wi|ds)最大的tj（注意，这个式子里只有j是变量）

3.然后，如果ds中的第i个单词wi在这里选择了一个与原先不同的topic（也就是说，这个时候i在遍历ds中所有的单词，而tj理当不变），就会对θd和φt有影响了。它们的影响又会反过来影响对上面提到的p(w|d)的计算。对D中所有的d中的所有w进行一次p(w|d)的计算并重新选择topic看作一次迭代。这样进行n次循环迭代之后，就会收敛到LDA所需要的结果了。

公式版

现在，我们对LDA的玩法基本了解了，
我们终于可以安静的刷刷公式了（注：尿点时刻）：

这一部分的解释，可以参照七月在线创始人July的CSDN博客：
http://blog.csdn.net/v_july_v/article/details/41209515

公式版

正经的理解LDA，分为下述5个步骤：

一个函数：gamma函数
四个分布：二项分布、多项分布、beta分布、Dirichlet分布
一个概念和一个理念：共轭先验和贝叶斯框架
两个模型：pLSA、LDA
一个采样：Gibbs采样

公式版

共轭分布与共轭先验：

后验概率（posterior probability）∝ 似然函数（likelyhood function）*先验概率（prior probability）

公式版

Gamma函数

阶乘函数在实数上的推广。

我们知道，对于整数而言：

$$\Gamma(n) = (n-1)!$$

对于实数：

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

公式版

二项分布（Binomial distribution）

二项分布是从伯努利分布推进的。伯努利分布，又称两点分布或0-1分布，是一个离散型的随机分布，其中的随机变量只有两类取值，非正即负{+，-}。而二项分布即重复n次的伯努利试验，记为 。简言之，只做一次实验，是伯努利分布，重复做了n次，是二项分布。二项分布的概率密度函数为：

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

公式版

多项分布，是二项分布扩展到多维的情况

多项分布是指单次试验中的随机变量的取值不再是0-1的，而是有多种离散值可能（1,2,3...,k）。比如投掷6个面的骰子实验，N次实验结果服从K=6的多项分布。当然啦，他们加起来的P应该是等于1的。

多项分布的概率密度函数为：

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

公式版

Beta分布，二项分布的共轭先验分布

给定参数a>0和b>0，取值范围为[0,1]的随机变量 x 的概率密度函数：

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

其中：

$$\frac{1}{B(\alpha,\beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}, \quad \Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$$

公式版

Dirichlet分布，是beta分布在高维度上的推广

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

其中

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha^i)}{\Gamma(\sum_{i=1}^{k} \alpha^i)}, \sum x_i = 1$$

公式版

贝叶斯派的思考方式：

先验分布$\pi(\theta)$ + 样本信息$\chi \Rightarrow$ 后验分布$\pi(\theta|x)$

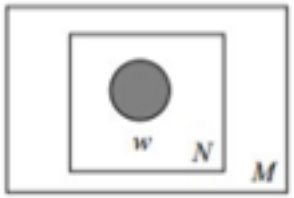
公式版

几个主题模型（循序渐进）：

Unigram model

对于文档$\mathbf{w} = (w_1, w_2, \cdots, w_N)$，用$p(w_n)$表示词$w_n$的先验概率，生成文档$\mathbf{w}$的概率为：

$$p(\mathbf{w}) = \prod_{n=1}^{N} p(w_n)$$

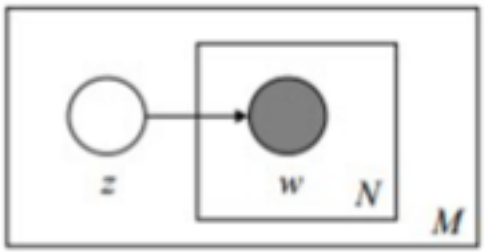


公式版

Mixture of unigrams model

该模型的生成过程是：给某个文档先选择一个主题z，再根据该主题生成文档，该文档中的所有词都来自一个主题。假设主题有z1,z2,z3,...zk，生成文档的概率为：

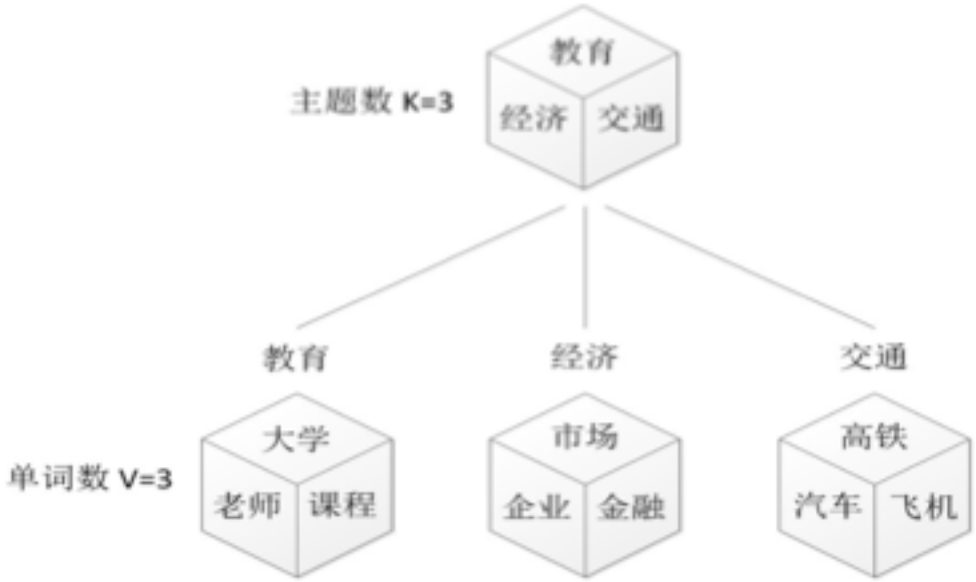
$$p(\mathbf{w}) = p(z_1)\prod_{n=1}^{N} p(w_n|z_1) + \cdots + p(z_k)\prod_{n=1}^{N} p(w_n|z_k) = \sum_z p(z)\prod_{n=1}^{N} p(w_n|z)$$



公式版

PLSA模型

刚刚的mix unigram模型里面，一篇文章只给了一个主题。但是现实生活中，一篇文章可能有多个主题，只不过是『出现的几率』不一样。



公式版

我们定义：

- $P(d_i)$表示海量文档中某篇文档被选中的概率。
- $P(w_j|d_i)$表示词w_j在给定文档d_i中出现的概率。
  - 怎么计算得到呢？针对海量文档，对所有文档进行分词后，得到一个词汇列表，这样每篇文档就是一个词语的集合。对于每个词语，用它在文档中出现的次数除以文档中词语总的数目便是它在文档中出现的概率$P(w_j|d_i)$。
- $P(z_k|d_i)$表示具体某个主题z_k在给定文档d_i下出现的概率。
- $P(w_j|z_k)$表示具体某个词w_j在给定主题z_k下出现的概率，与主题关系越密切的词，其条件概率$P(w_j|z_k)$越大。

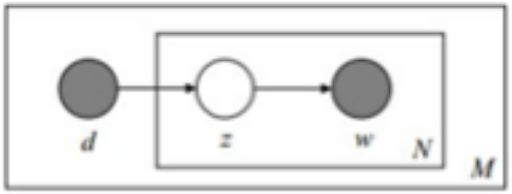
公式版

我们的文本生成模型就是：
（还记得什么是文本生成模型嘛？返回标准版part回顾一下）

1. 按照概率$P(d_i)$选择一篇文档d_i
2. 选定文档d_i后，从主题分布中按照概率$P(z_k|d_i)$选择一个隐含的主题类别z_k
3. 选定z_k后，从词分布中按照概率$P(w_j|z_k)$选择一个词w_j

公式版

我们通过观测，得到了『知道主题是什么，我就用什么单词』的文本生成模型，那么，根据贝叶斯定律，我们就可以反过来推出『看见用了什么单词，我就知道主题是什么』



从而可以根据大量已知的文档-词项信息$P(w_j|d_i)$，训练出文档-主题$P(z_k|d_i)$和主题-词项$P(w_j|z_k)$，如下公式所示：

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

故得到文档中每个词的生成概率为：

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

由于$P(d_i)$可事先计算求出，而$P(w_j|z_k)$和$P(z_k|d_i)$未知，所以$\theta = (P(w_j|z_k), P(z_k|d_i))$就是我们要估计的参数（值），通俗点说，就是要最大化这个θ。

公式版

LDA模型

LDA就是在pLSA的基础上加层贝叶斯框架，即LDA就是pLSA的贝叶斯版本

PLSA与LDA对比

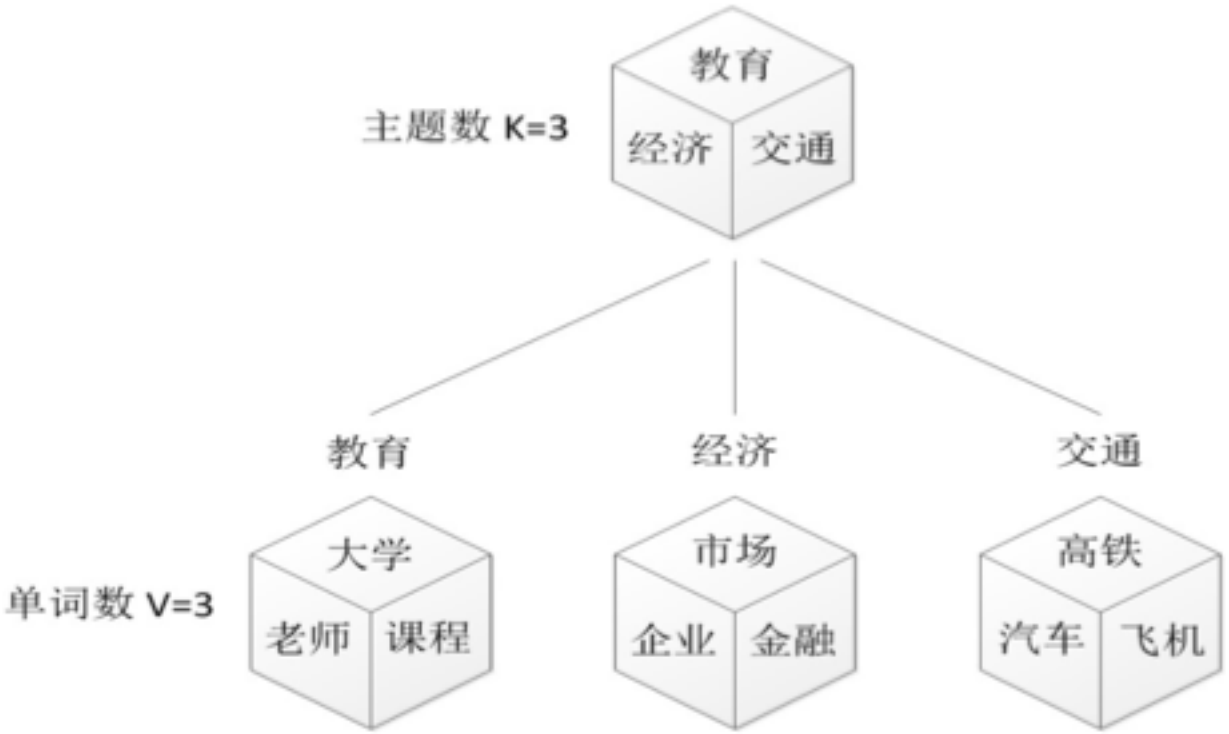
PLSA

1. 按照概率$P(d_i)$选择一篇文档d_i
2. 选定文档d_i后，确定文章的主题分布
3. 从主题分布中按照概率$P(z_k|d_i)$选择一个隐含的主题类别z_k
4. 选定z_k后，确定主题下的词分布
5. 从词分布中按照概率$P(w_j|z_k)$选择一个词w_j "

LDA

1. 按照先验概率$P(d_i)$选择一篇文档d_i
2. 从狄利克雷分布（即Dirichlet分布）α中取样生成文档d_i的主题分布θ_i，换言之，主题分布θ_i由超参数为α的Dirichlet分布生成
3. 从主题的多项式分布θ_i中取样生成文档d_i第 j 个词的主题$z_{i,j}$
4. 从狄利克雷分布（即Dirichlet分布）β中取样生成主题$z_{i,j}$对应的词语分布$\phi_{z_{i,j}}$，换言之，词语分布$\phi_{z_{i,j}}$由参数为β的Dirichlet分布生成
5. 从词语的多项式分布$\phi_{z_{i,j}}$中采样最终生成词语$w_{i,j}$ "

PLSA与LDA对比



PLSA与LDA对比

pLSA跟LDA的本质区别就在于它们去估计未知参数所采用的思想不同，前者用的是频率派思想，后者用的是贝叶斯派思想。

主体模型

就是这样，喵

实战

【详见随堂iPython Notebook】

感谢大家！

恳请大家批评指正！