利用Python实现LDA聚类算法

简介

这是一篇面向工程师的LDA入门笔记，并且提供一份开箱即用Python实现。本文只记录基本概念与原理，并不涉及公式推导。文中的LDA实现核心部分采用了arbylon的LdaGibbsSampler并力所能及地注解了，在数学题库分类应用上测试良好，开源在github上。

主题模型

在我的博客上，有篇文章《python 结巴分词(jieba)》被归入分词目录，分词即为该文章的主题。而该文章因为涉及到Python语言知识讲解，又被我归入了Python目录。所以该文章的主题并不单一，具体来说文中90%在讲分词，10%稍微提了一下Python语言知识讲解。

传统的文本分类器，比如贝叶斯、kNN和SVM，只能将其分到一个确定的类别中。假设我给出3个分类“文学”“分词”“Python”让其判断，如果某个分类器将该文归入Python类，我觉得还凑合，如果分入文学，那我觉得这个分类器不够准确。

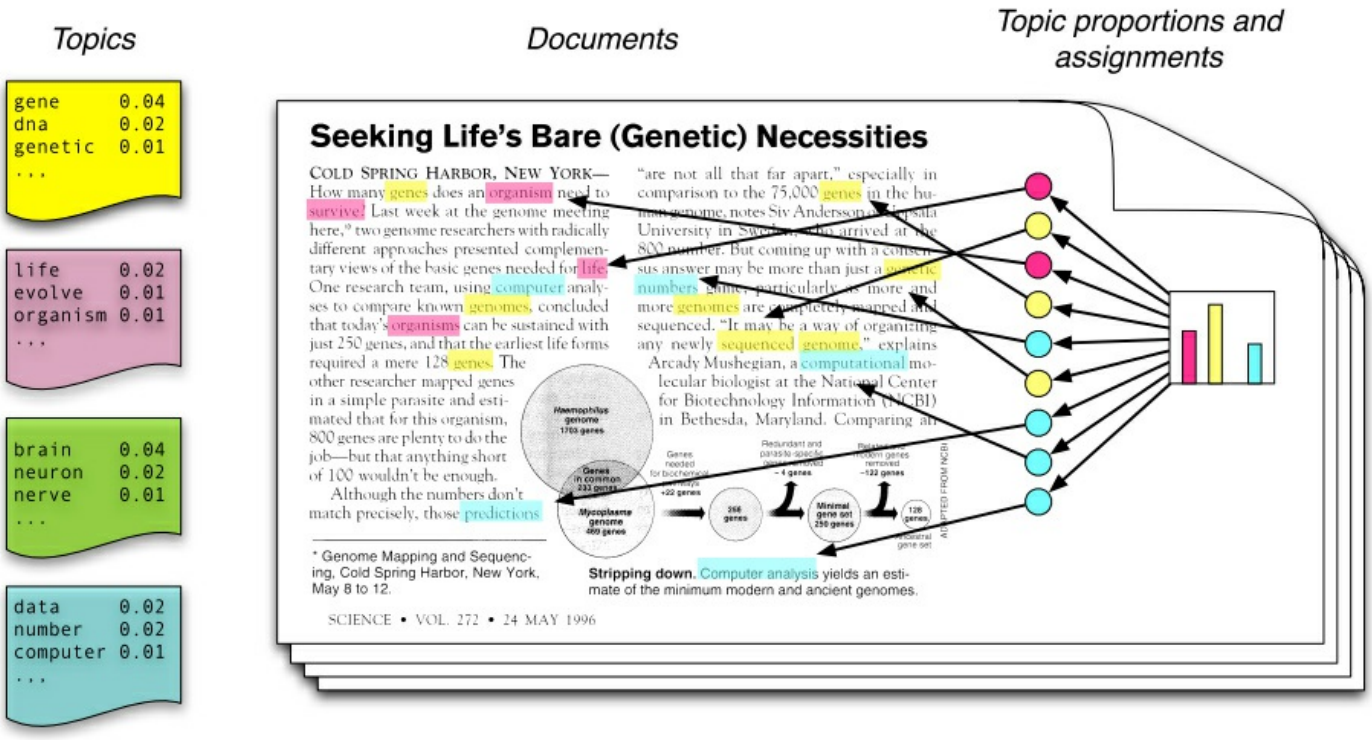
假设一个文艺小青年来看我的博客，他完全不懂Python你和分词，自然也给不出具体的备选类别，有没有一种模型能够告诉这个白痴，这篇文章很可能(90%)是在讲分词，也可能(9%)是在讲Python，几乎不可能(1%)是在讲其它主题呢？

有，这样的模型就是主题模型。

LDA

简述

潜在狄立克雷分配（Latent Dirichlet Allocation，LDA）主题模型是最简单的主题模型，它描述的是一篇文章是如何产生的。如图所示：



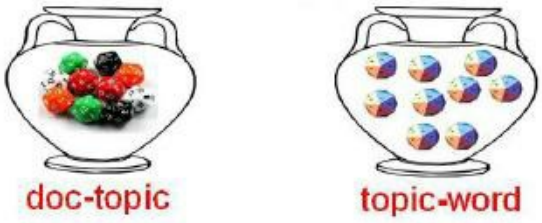
从左往右看，一个主题是由一些词语的分布定义的，比如蓝色主题是由2%几率的data，2%的number……构成的。一篇文章则是由一些主题构成的，比如右边的直方图。具体产生过程是，从主题集合中按概率分布选取一些主题，从该主题中按概率分布选取一些词语，这些词语构成了最终的文档（LDA模型中，词语的无序集合构成文档，也就是说词语的顺序没有关系）。

如果我们能将上述两个概率分布计算清楚，那么我们就得到了一个模型，该模型可以根据某篇文档推断出它的主题分布，即分类。由文档推断主题是文档生成过程的逆过程。

在《LDA数学八卦》一文中，对文档的生成过程有个很形象的描述：

Game 12 LDA Topic Model

1: 上帝有两大坛子的骰子，第一个坛子装的是doc-topic 骰子,第二个坛子装的是topic-word 骰子；



2: 上帝随机的从第二个坛子中独立的抽取了K 个topic-word 骰子，编号为1到K；

3: 每次生成一篇新的文档前，上帝先从第一个坛子中随机抽取一个doc-topic 骰子，然后重复如下过程生成文档中的词

- 投掷这个doc-topic 骰子,得到一个topic 编号z
- 选择K 个topic-word 骰子中编号为z的那个，投掷这个骰子，于是得到一个词

概率模型

LDA是一种使用联合分布来计算在给定观测变量下隐藏变量的条件分布（后验分布）的概率模型，观测变量为词的集合，隐藏变量为主题。

联合分布

LDA的生成过程对应的观测变量和隐藏变量的联合分布如下：

$$p\left(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}\right)=\prod_{i=1}^{K} p\left(\beta\right) \prod_{d=1}^{D} p\left(\theta_{d}\right)\left(\prod_{n=1}^{N} p\left(z_{d,n} | \theta_{d}\right) p\left(w_{d,n} | \beta_{1:K}, z_{d,n}\right)\right)$$

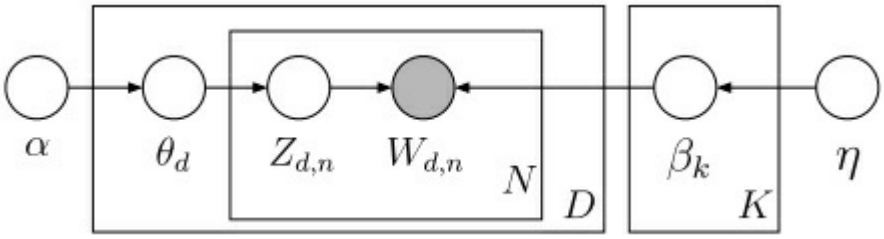
式子有点长，耐心地从左往右看：

式子的基本符号约定——β表示主题，θ表示主题的概率，z表示特定文档或词语的主题，w为词语。进一步说——

β1:K为全体主题集合，其中βk是第k个主题的词的分布（如图1左部所示）。第d个文档中该主题所占的比例为θd，其中θd,k表示第k个主题在第d个文档中的比例（图1右部的直方图）。第d个文档的主题全体为zd，其中zd,n是第d个文档中第n个词的主题（如图1中有颜色的圆圈）。第d个文档中所有

词记为wd，其中wd,n是第d个文档中第n个词，每个词都是固定的词汇表中的元素。

p(β）表示从主题集合中选取了一个特定主题，p(θd)表示该主题在特定文档中的概率，大括号的前半部分是该主题确定时该文档第n个词的主题，后半部分是该文档第n个词的主题与该词的联合分布。连乘符号描述了随机变量的依赖性，用概率图模型表述如下：



比如，先选取了主题，才能从主题里选词。具体说来，一个词受两个随机变量的影响（直接或间接），一个是确定了主题后文档中该主题的分布θd，另一种是第k个主题的词的分布βk（也就是图2中的第二个坛子）。

后验分布

沿用相同的符号，LDA后验分布计算公式如下：

$$p\left(\beta_{1:K}, \theta_{1:D}, z_{1:D} | w_{1:D}\right) = \frac{p\left(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}\right)}{p\left(w_{1:D}\right)}$$

分子是一个联合分布，给定语料库就可以轻松统计出来。但分母无法暴力计算，因为文档集合词库达到百万（假设有w个词语），每个词要计算每一种可能的观测的组合（假设有n种组合）的概率然后累加得到先验概率，所以需要一种近似算法。

基于采样的算法通过收集后验分布的样本，以样本的分布求得后验分布的近似。

θd的概率服从Dirichlet分布，zd,n的分布服从multinomial分布，两个分布共轭，所谓共轭，指的就是先验分布和后验分布的形式相同：

$$Dir(\vec{p}|\vec{\alpha}) + MultCount(\vec{m}) = Dir(\vec{p}|\vec{\alpha} + \vec{m})$$

两个分布其实是向量的分布，向量通过这两个分布取样得到。采样方法通过收集这两个分布的样本，以样本的分布近似。

马氏链和Gibbs Sampling

马氏链

所谓马氏链指的是当前状态只取决于上一个状态。马氏链有一个重要的性质：状态转移矩阵P的幂是

收敛的，收敛后的转移矩阵称为马氏链的平稳分布。给定p(x)，假如能够构造一个P，转移n步平稳分布恰好是p(x)。那么任取一个初始状态，转移n步之后的状态都是符合分布的样本。

Gibbs Sampling

Gibbs Sampling是高维分布（也即类似于二维p(x,y)，三维p(x,y,z)的分布）的特化采样算法。

更深入的数学知识请参考附录，能力有限，等完全看懂了再更新这部分。

Python代码实现

LDA实现源码：https://github.com/huiyang865/material/tree/master/python/LDA

python-LDA

lda模型的python实现，算法采用sampling抽样

- 项目基于python2.7.10如果发现计算概率为0，可能是python的兼容性问题，暂时没时间修复（发现python3.0以上版本会出现此问题）

使用方法

模型训练文件

`zzz_math_todo_part` 是数学应用题的原题，可将每道题看成一篇独立的文章。

```
一个数，最高位千位上是10以内的最大质数，十位上是最小的合数，其他数位上的数都是0，这个数是 \__.
把「\frac{1}{7}」化成小数后，小数点后第13位上的数字是 \__.
镇海雅乐学校二年级的小朋友到一条小路的一边植树．小朋友们每隔2米种一棵树（马路两头都种了树），最后发
黄色卡片6张，红色卡片4张，蓝色卡片5张放在袋子里，至少要摸出4张，就可以保证摸出两张颜色相同的卡片．
两端都在圆上的线段中， \__最长，用字母__表示．
某校参加军训队列表演比赛，组织一个方阵队伍．如果每班60人，这个方阵最多要有4个班的同学参加，如果每班
「\frac{1}{2}」与「\frac{2}{3}」的和除以「\frac{5}{18}」，商是多少？
分母是13的最简真分数有 \__个，它们的和是__.
一条直线若同时平行于两个相交平面，那么这条直线与这两个平面的交线的位置关系是 \__.
钢铁厂有一块棱长是6分米的正方体钢坯，现在要把它熔铸成一个横截面为边长2厘米正方形的长方体方钢，这个
```

数据预处理

运行./preprocess/preprocess.py文件，模型将利用结巴分词工具(具体分词方法可见本人另一篇文章)预处理源数据，使之成为本模型的LDA算法的输入数据。

LDA聚类分析

`train.dat` 是数据预处理结果，显示格式如下：（一行表示一篇文档）

```
质数 数是 合数 千位 数位 十位 以内 最小 最高 最大 其他 这个 一个
位上 小数点 小数 化成 数字
小朋友 小路 米种 雅乐 植树 二年级 镇海 一棵树 马路 每隔 两头 一共 学校 一边 一条 最后 发现
卡片 袋子 两张 蓝色 黄色 红色 颜色 相同 放在 至少 保证 可以
线段 字母 两端 最长 表示
方阵 个班 每班 参加 同学 某校 最多要 这个 军训 队列 应为 如果 表演 比赛 队伍 人数 至少 组成 组织
商是 除以 多少
真分数 最简 分母 它们
直线 平面 交线 相交 两个 平行 位置 一条 关系 那么 同时
方钢 分米 正方体 长方体 棱长 横截面 熔铸 钢坯 边长 正方形 钢铁厂 厘米 一块 多少 现在 这个 一个
```

模型输出文件

```
`model_parameter.dat` 保存模型训练时选择的参数
`wordidmap.dat` 保存词与id的对应关系，主要用作topN时查询
`model_twords.dat` 输出每个类高频词topN个
`model_tassgin.dat` 输出文章中每个词分派的结果，文本格式为词id:类id
`model_theta.dat` 输出文章与类的分布概率，文本一行表示一篇文章，概率1  概率2 ...表示文章属于类
`model_phi.dat` 输出词与类的分布概率，是一个K*M的矩阵，其中K为设置分类的个数，M为所有文章的词的
```

使用说明

- 模型需要人为设定算法迭代次数和主题数量，详细文档设定方式请查看setting.conf
- cd 到lda.py所在目录，执行命令:python lda.py

总结

本文的部分内容转自http://www.hankcs.com/nlp/lda-java-introduction-and-implementation.html。LDA主要将文件和词语用主题这个概念联系起来。首先分析文件中的主题，然后将所有文件中的词语主题划分。最后将文件与词语建立联系。