LDA主题模型介绍

<https://www.jianshu.com/p/fa97454c9ffd>

LDA（Latent Dirichlet Allocation）是一种文档主题生成模型，也称为一个三层[贝叶斯](https://links.jianshu.com/go?to=https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF" \t "_blank)概率模型，包含词、主题和文档三层结构。所谓生成模型，就是说，我们认为一篇文章的每个词都是通过“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”这样一个过程得到。文档到主题服从多项式分布，主题到词服从多项式分布。

****1 LDA的目的就是要识别主题，即把文档—词汇矩阵变成****

****文档—主题矩阵（分布）****

****主题—词汇矩阵（分布）****

语料库中的每一篇文档与T个主题的一个多项分布相对应，将该多项分布记为θ

即模型对该文档给出的分属各个主题的概率分布，即我们生成的文本衍生特征。

每个主题又与词汇表（vocabulary）中的V个单词的一个多项分布相对应，将这个多项分布记为φ。即该主题的词汇描述

Tp1 (1, '0.068\*"手机" + 0.037\*"贷" + 0.036\*"我" + 0.036\*"借款" + 0.035\*"腾讯" + 0.034\*"百度" + 0.033\*"视频" + 0.032\*"你" + 0.031\*"支付宝" + 0.030\*"微信"')

2 LDA整体流程

先定义一些字母的含义：文档集合D，主题（topic)集合T

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，设d有n个单词。（LDA里面称之为wordbag，实际上每个单词的出现位置对LDA算法无影响，也就是说LDA没有引入上下文信息）

D中涉及的所有不同单词组成一个大词袋VOC，LDA以文档集合D作为输入，希望训练出的两个结果向量然后聚类（设聚成k个topic，VOC中共包含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。

因每个主题还是由词构成（一般10个词来描述这个主题），故的nti是计算了该文档中的词在该主题中命中了几个，n是该文档词的数目

Tp1 (1, '0.068\*"手机" + 0.037\*"贷" + 0.036\*"我" + 0.036\*"借款" + 0.035\*"腾讯" + 0.034\*"百度" + 0.033\*"视频" + 0.032\*"你" + 0.031\*"支付宝" + 0.030\*"微信"')

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VOC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VOC中第i个单词的数目，N表示所有对应到topic的单词总数。

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(topic|docu) = θd

p(w|t)利用φt计算得到。 p(word|topic) = φt

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。

LDA主题模型在短文本上的应用

数据：地址文本

目标：通过主题一致性，困惑度等指标找到，最佳语料以及最佳主题数的边界，通过程序自动对最优主题寻优

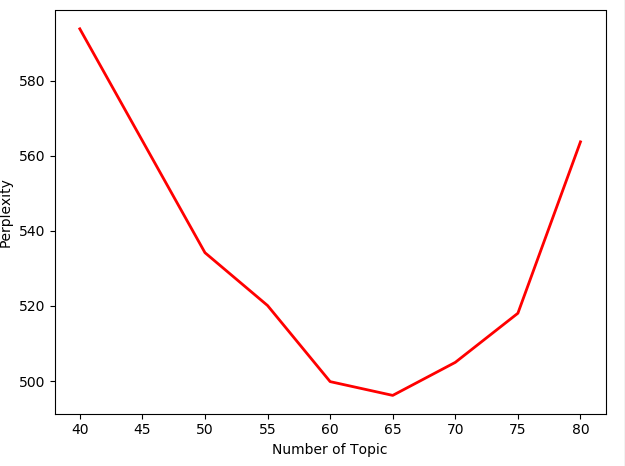
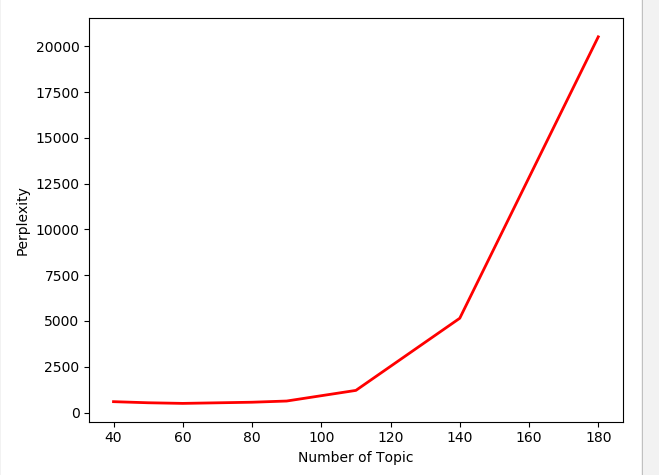
困惑度：困惑度可以理解为对于一篇文本，所训练出来的模型对文本属于哪个主题有多不确定，这个不确定程度就是困惑度。困惑度越低，说明聚类的效果越好。

实践1 3410: 同信云盾数据 文本数80860 dict len 36014 DT len 80860

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 一致性 | 困惑度 | 测试集AUC |  |
| Tp40 | -12.073197 | 593.87015 |  |  |
| Tp50 |  | 534.20089 | 0.783270 |  |
| Tp60 | -13.031725 | 499.83862 |  |  |
| Tp80 |  | 563.74753 | 0.779730 |  |
| Tp90 | -13.021619 | 628.83943 |  |  |
| Tp110 |  | 1209.44703 | 0.747014 |  |
| Tp140 | -16.125137 | 5149.64382 |  |  |
| Tp180 | -16.647381 | 20515.11013 |  |  |
| Tp250 | -16.673315 | 24071.90215 |  |  |
| Tp300 | -16.673315 |  |  |  |
|  |  |  |  |  |

通过困惑度缩小范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 一致性 | 困惑度 |  |  |
| Tp55 | -12.073197 | 520.11836 |  |  |
| Tp65 | -12.073197 | 496.16712 | 0.795351 |  |
| Tp70 | -12.073197 | 504.97288 |  |  |
| Tp75 | -12.073197 | 518.06380 |  |  |
| Tp90 | -13.021619 | 628.83943 |  |  |



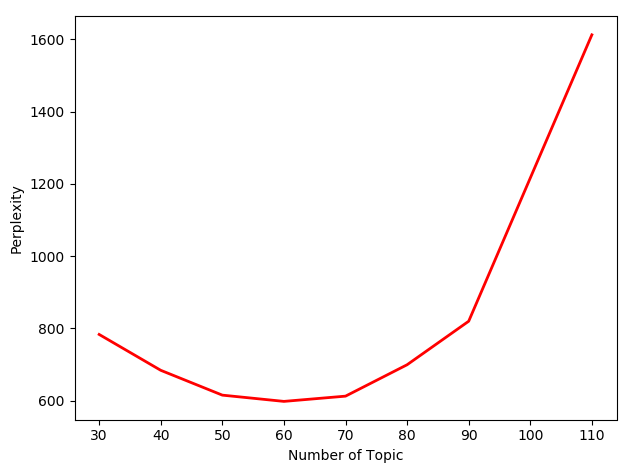
实践2 3410: 同信云盾数据 文本数172349 dict len 43739 DT len 172349

我们可以看到虽然文本数增加了不止一倍，但是词袋变化3.6--4.3，说明新增新词并不多，词袋再慢慢的变得饱和，随着文本的增加，很多词重复出现

如果模型可以随着文档的变化而自动的调整最佳的主题数，这样是最好的，因为虽然我们有困惑度可以确定最佳主题数，但是我们还没有找到最佳训练文档数，

其他大佬，融入了分层狄利克雷过程（HDP），构成一种非参数主题模型，好处就是不需要预先指定个数，模型可以随着文档的变化而自动的对主题个数进行调整，这个后面需要去关注一下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 一致性 | 困惑度 | 测试集AUC |  |
| Tp30 | -11.346477 | 783.48359 |  |  |
| Tp40 | -12.376871 | 684.26496 |  |  |
| Tp50 |  | 615.55030 | 0.767938 |  |
| Tp60 | -13.702856 | 598.24407 | 0.770681 |  |
| Tp70 | -13.500750 | 612.76128 |  |  |
| Tp80 |  | 699.65867 | 0.757691 |  |
| Tp90 | -13.021619 | 820.04785 |  |  |
| Tp110 | -13.121567 | 1612.48937 |  |  |



实践3 3月份pdl 文本数6.7w dict len 44017 DT len 56879

文本特点：不同机构的地址数据准确度不同，有的机构地址数据杂乱无章

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 一致性 | 困惑度 | AUC |  |
| Tp30 | -12.653218 | 1433.79935 |  |  |
| Tp40 | -13.887622 | 1291.30237 |  |  |
| Tp50 |  | 1190.82393 | 0.639619 |  |
| Tp60 | -14.462109 | 1219.53827 |  |  |
| Tp70 |  | 1416.46896 |  |  |
| Tp80 |  | 1489.76287 |  |  |
| Tp90 | -12.877761 | 1637.69123 |  |  |
| Tp100 |  | 1612.48937 |  |  |

