毕业设计汇报

基于产品评论信息的用户观点提取

汇报人：武文娟

目录

- 研究背景
- 数据来源
- 流程及方法（重点）
- 进度及计划

研究背景

- 随着互联网和电子商务的发展，越来越多的人会进行网络购物，并且会根据以往用户对于产品的评论信息作出决策。
- 这些评论信息庞大而又复杂，属于无结构化数据，仅靠人工阅读的方式，很难精确快速地获得想要的信息。
- 因此，本文基于网络上产品的评论信息，运用一系列文本处理的技术，对用户的观点进行挖掘和提取。

关键词：产品评论，分词，SVD，LDA，主题提取

数据来源

- 本研究针对京东网站上比较畅销的笔记本电脑的评论数据，数据来源于数据堂。

  （详见http://more.datatang.com/data/46283）

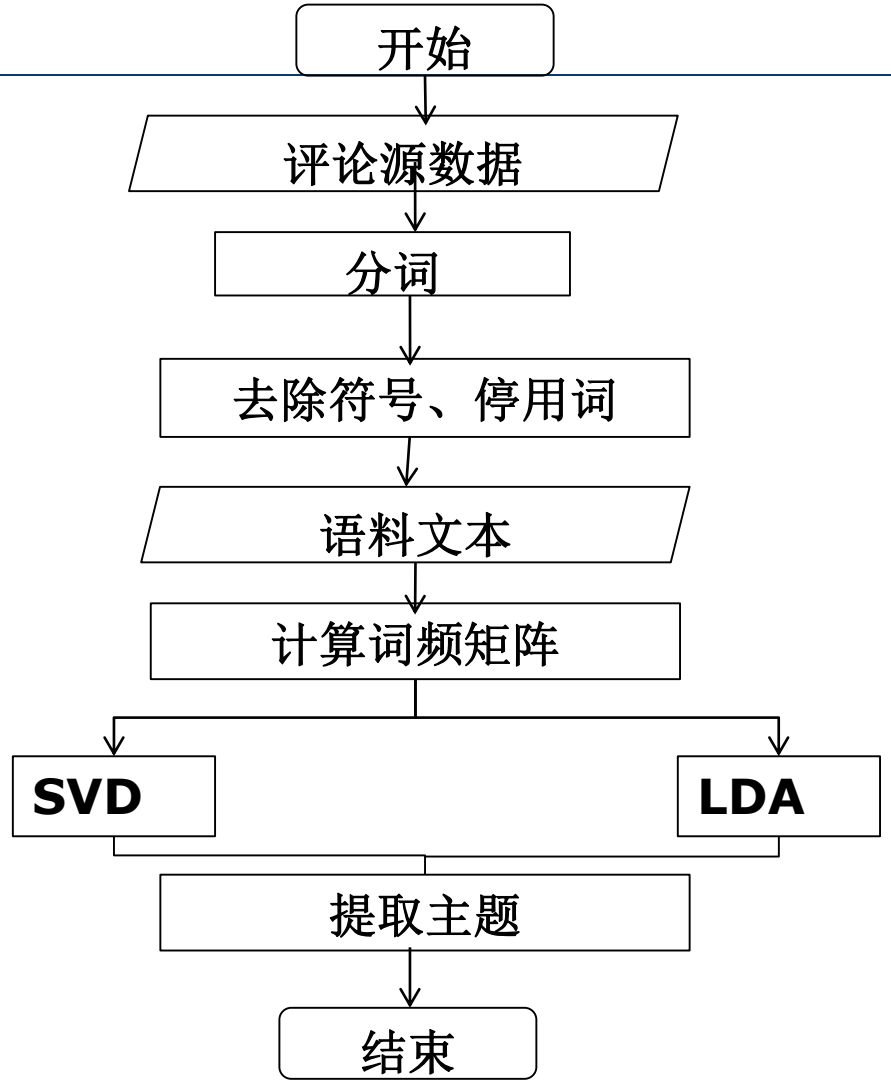
- 评论信息集中包括了 284个产品共20万条的信息，时间为截止2014年6月6日前。
- 数据包括产品编号、评论用户、购买时间、评论时间、评分、心得、省份、标签字段。

以下为数据节选：

产品编号|评论用户|购买时间|评论时间|评分|心得|省份|标签
1000025|jd_351529599|2014-03-13|2014-04-11|5|刚好感觉还不错||
1000025|坤歌沉寂|2014-03-19|2014-04-08|5|急哦急哦急哦，急哦急哦急哦，WDOS太难受啦|江苏|速度快 系统很好 外观漂亮
...

1000026|jd_lfyxrd|2013-11-12|2013-11-24|5|嘛，还是相当不错的，果然是不失风度。|上海|
899379|斯特斯8300|2013-11-25|2014-04-21|5|还可以，在第京东买的二台电脑了。||屏幕清晰 键盘很舒服 重量轻 外观漂亮

流程介绍



1、分词

- 分词工具：jieba，中文分词工具。
- 使用方式：在python中import jieba即可。
- 分词原理：

1）基于Trie树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG)；

2）采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合；

3）对于未登录词，采用了基于汉字成词能力的HMM模型，使用了Viterbi算法。

- 分词模式：

1）精确模式，试图将句子最精确地切开，适合文本分析。**jieba.cut(“”,cut_all=False)**

2）全模式，把句子中所有的可以成词的词语都扫描出来。**jieba.cut(“”,cut_all=True)**

3）搜索引擎模式，在精确模式的基础上，对长词再次切分。**jieba.cut_for_search(“”)**

Example

原句：小明硕士毕业于中国科学院计算所，后在日本京都大学深造

- 精确模式：

小明/ 硕士/ 毕业/ 于/ 中国科学院/ 计算所/ ，/ 后/ 在/ 日本京都大学/ 深造

- 全模式：

小/ 明/ 硕士/ 毕业/ 于/ 中国/ 中国科学院/ 科学/ 科学院/ 学院/ 计算/ 计算所/ ，/ 后/ 在/ 日本/ 日本京都大学/ 京都/ 京都大学/ 大学/ 深造

- 搜索模式：

小明/ 硕士/ 毕业/ 于/ 中国/ 科学/ 学院/ 科学院/ 中国科学院/ 计算/ 计算所/ ，/ 后/ 在/ 日本/ 京都/ 大学/ 日本京都大学/ 深造

2、去掉符号、停用词

- 停用词：Stop Words大致为如下两类：

1）这些词应用十分广泛，在Internet上随处可见，如web等词。

2）包括了语气助词、副词、介词、连接词等，通常自身 并无明确的意义，只有将其放入一个完整的句子中才有一定作用，如常见的“的”、“在”之类。

- 使用工具：jieba

Example

原句：小明硕士毕业于中国科学院计算所，后在日本京都大学深造

设置符号和停用词：于 在 ，

```
于
在
，
```

结果：

```
Prefix dict has been built succesfully.
小明
硕士
毕业
计算所
后
日本京都大学
深造
[Finished in 3.9s]
```

3、计算词频矩阵

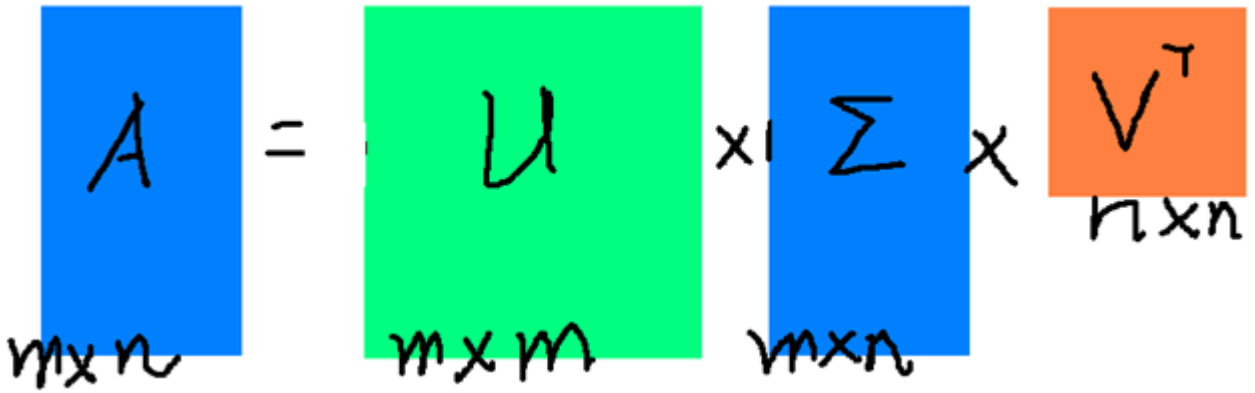
- 用Python中的正则表达式进行中文匹配后进行词频统计，之后生成一个词频矩阵，这是一个比较大的稀疏矩阵。
- 形如：

	评论1	评论2	评论3	评论4	…	评论n
好用	2	1	0	0	…	0
美观	1	0	1	0	…	0
方便	0	0	1	2	…	1
…	…	…	…	…	…	…
词语m	0	1	0	0	…	1

m*n的矩阵

4、SVD(奇异值分解)

SVD（奇异值分解）是一个能适用于任意形状的矩阵的降维方法。分解方法为 $A = U\Sigma V^T$



假设A是一个N * M的矩阵，那么得到的U是一个N * N的方阵（里面的向量是正交的，U里面的向量称为左奇异向量），Σ是一个N * M的矩阵（除了对角线的元素都是0，对角线上的元素称为奇异值），V’(V的转置)是一个N * N的矩阵，里面的向量也是正交的，V里面的向量称为右奇异向量）。

SVD

- 奇异值跟特征值类似，在矩阵Σ中也是从大到小排列，而且奇异值的减少特别的快，在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上了。所以说，我们可以用前r大的奇异值来近似描述矩阵，即：

$$\hat{A}_{m\times n} = U_{m\times r} \sum_{r\times r} V^{T}_{r\times n}$$

（ r是一个远小于m、n的数）

$\hat{A}_{m\times n}$ 将会是一个接近于A的矩阵。此时，V的每一行可以代表一个评论信息，评论信息将会降维为r维的向量，大大解决了稀疏性和数据维度太大的问题。

SVD实现

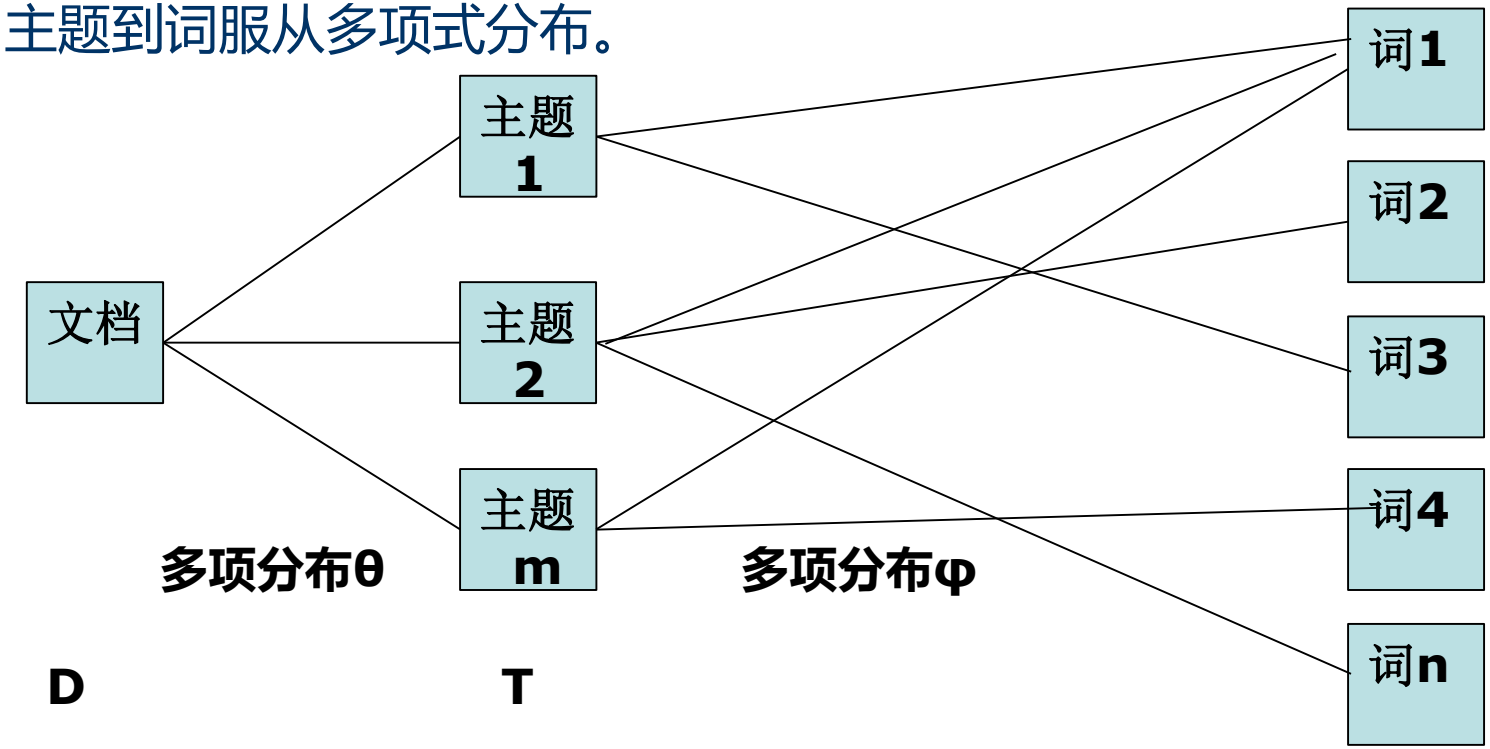
- SVD实现可以使用python中的拓展包numpy，scipy。（这部分正在学习）

```
import numpy as np
```

学习网址：http://www.cnblogs.com/nlp-yekai/p/3848528.html

5、LDA（文档主题生成模型）

- LDA是一种文档主题生成模型，也称为一个三层贝叶斯概率模型，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。



核心公式：

$$p(w|d)=p(w|t)*p(t|d)$$

LDA的学习过程

- LDA算法开始时，先随机地给θd和φt赋值（对所有的d和t）。详细迭代学习过程：

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：

$$pj(wi|ds)=p(wi|tj)*p(tj|ds)$$

2.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，即argmax[j]pj(wi|ds)。

3.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，就会对θd和φt有影响，它们的影响又会反过来影响上面提到的p(w|d)的计算。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。

LDA实现

- LDA实现可以在python中导入lda。（目前还未实现）

```
import lda
import lda.datasets
```

学习网址：https://my.oschina.net/letiantian/blog/616413?fromerr=ThbaouNJ

进度及计划

已完成：

- ✓ 查阅论文，了解课题的研究背景、研究现状及基本思路。
- ✓ 根据论文研究的应用场景，获取数据源。
- ✓ 学习Python语言的基本语法、函数、类、正则表达式。
- ✓ 利用jieba等分词工具对评论信息进行分词及去停用词。
- ✓ 学习SVD和LDA的理论知识并安装所需工具包。

进度及计划

待完成：

- 产生词频矩阵代码实现。
- SVD和LDA的代码实现。
- 对预处理后的文本信息进行主题词提取。（预计下周组会前达到此步骤）
- 开始写论文。（边做实验边开始写，4月初有初稿）
- 课题与流程优化。

THANKS!