SY2103801 陈真

- **题目：**从给定的语料库中均匀抽取 200 个段落（每个段落大于 500 个词），每个段落的标签就是对应段落所属的小说。利用 LDA 模型对于文本建模，并把每个段落表示为主题分布后进行分类。验证与分析分类结果。

- **背景:**

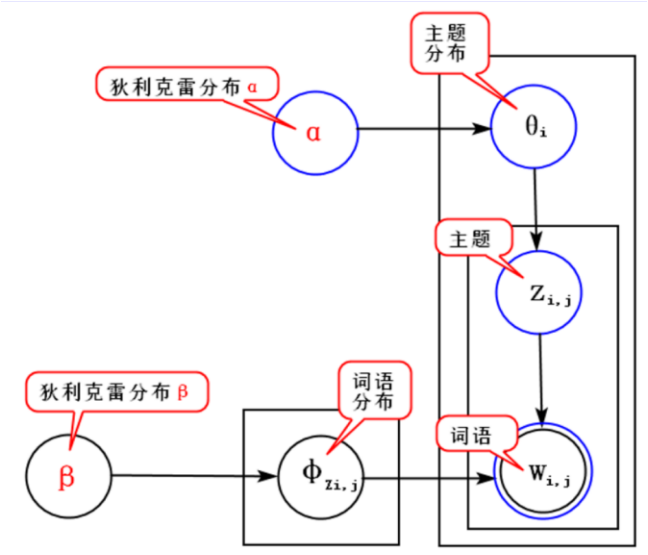
一篇文档可以包含多个主题，所以会有主题分布这个概率．可以这么理解一篇文章的生成：先以一定的概率选取某个主题，然后再以一定的概率选取该主题下的某个词，不断重复这两步，直到完成整个文档。LDA 解决的问题就是，分析给定的一篇文章都有什么主题，每个主题出现的占比大小是多少。LDA 对短文本的主题分类效果比较差。

从宏观上来看，在 LDA 模型中，以 topic 作为中间层，问题可以用如下形式进行表示：P(单词|文档) = P(单词|主题) × P(主题|文档)，即 P(w|d) = P(w|t)×P(t|d)，其中的 P(单词|文档)是已知的，直接从给定的文档中就可以计算得出的，P(单词|主题)和 P(主题|文档)是用来进行拟合的，调整这两个分布，直至符合单词文档的实际分布。也就是说，最终需要求的是 P(单词|主题)和 P(主题|文档)这两个分布，而真正对文档主题分布有用的，是 P(主题|文档)这个分布，即 P(t|d)。

- **LDA 模型生成文档步骤:**

在 LDA 模型中，一篇文档的生成方式如下：

1. 从迪利克雷分布α中取样生成文档 i 主题分布θi；
2. 从主题的多项式分布θi 中取样生成文档 i 第 j 个词的主题 zi，j；
3. 从迪利克雷分布β中取样生成主题 zi，j 对应的词语分布Φzi,j；
4. 从词语的多项式分布Φzi,j 中采样最终生成词语 wi,j。



- **题解**

1. **解题路线：**

我们将全部 16 篇金庸小说作为语料进行 LDA 模型构建，在 16 篇小说中随机均匀抽取长度不小于 500 的 248 个片段，将每个小说名作为片段标签。利用已经构筑好的 LDA 模型得到片段的主题分布。利用片段的主题分布与标签，划分训练集和测试集，选择 SVM 模型进行分类任务。

2. LDA 模型构建：

首先进行数据预处理。将原 16 篇文档整理合并，按行划分，形成合并文档 process.txt，接着进行分词并去掉停用词，分词这里使用 jieba 工具包，停用词表这里使用 https://github.com/goto456/stopwords 中的中文停用词表。

接着利用 gensim 进行 LDA 模型构建。读取文本数据 59423 条，利用 corpora.Dictionary()进行词典构建，利用 doc2bow()计算文本向量。利用 gensim.models.TfidfModel()进行模型构建，选择构建的主题数量为 20，最后利用 gensim.models.ldamodel()进行模型的训练。训练参数设置如下：

```
lda = models.LdaModel(corpus tfidf, num topics=num topics,
id2word=dictionary,
                alpha=0.01, eta=0.01, minimum probability=0.001,
                update_every=1, chunksize=3000, passes=20)
```

空间限制，这里列举某 4 个主题前 10 个主要词概率分布。

主题 1	孩儿	解药	门派	西域	二哥	道人	经脉	食指	不该	真气
	0.0135 6217	0.0077 6301	0.0059 6932	0.0059 6429	0.0058 8608	0.0050 4877	0.0049 2453	0.0047 3175	0.0045 9261	0.0041 8687
主题 2	少林	少林寺	大师	方丈	弟子	赵钱孙	下山	敝	糊涂	蒙古
	0.0116 3895	0.0111 6088	0.0103 9371	0.0082 5011	0.0078 9407	0.0064 7364	0.0050 0936	0.0044 57	0.0043 9796	0.0043 7793
主题 3	师兄	刀法	较量	恶	落下	打听	苍老	惊讶	不忍	岩石
	0.0205 1514	0.0150 0916	0.0078 7329	0.0059 2551	0.0057 7093	0.0055 4486	0.0052 3014	0.0049 3958	0.0047 9358	0.0046 7229
主题 4	壁上	各派	老前辈	崖	毒	伯伯	人均	殿下	奔到	六个
	0.0088 2462	0.0058 0436	0.0052 8313	0.0051 4028	0.0045 6607	0.0044 5411	0.0040 6776	0.0040 5323	0.0040 2811	0.0040 0323

3. 片段主题分布生成：

由于 LDA 对短文本的主题分类效果比较差。所以摘取的片段不宜过短。我们首先将各个文档分别进行预处理，再进行长段落选择。发现并不是每一个小说都有足够数量的 500 词以上的段落，因此我们将部分小说段落进行合并。共摘取 248 个片段。代码如下

```
quanji x=[]
quanji y=[]
we10=0
```

```
for we in range(len(ttext0)):
    jj=0
    for we1 in range(len(ttext0[we])):#每一篇
        temp=[]
        if len(jieba.lcut(ttext0[we][we1])) >=40 and we1>we10+40:#某一
段大于 10
            for juju in range(40):
                cd 100=[]
                cd 99=jieba.lcut(ttext0[we][we1+juju])
                for dd in cd 99:
                    if dd not in stop words:
                        cd 100.append(dd)
                temp+=cd 100
            assert len(temp)>700
            quanji x.append(temp)
            quanji y.append(we)
            jj+=1
        if jj>30:#大于 20 就下一篇
            we10 = we1
            break
```

所得到的 248 个片段，其中片段最短长度为 50795，最短为 884，平均长度为 5713。可见我们每个片段的词数较多，主题特征相应应该也较多，应该更加有利于后续分类的效果。

依建立片段的词袋模型并输入 LDA 模型得到各个片段的主题分布概率值。将其作为特征值，将片段的所属小说名作为标签，得到总数据集。数据集特征以及标签见 Test.xlsx

4. **SVM 分类**：

进行数据集随机划分，代码如下：

```
X train, X test, y train, y test =
model selection.train test split(X, y, test size = 0.2, random state
= 1234)
```

用 sklearn 的 SVM 包进行训练预测，由于测试集合数量不多，这里不一一按照类别进行召回率与精确率计算，直接按照预测正确除以总数计算分类效果。测试集标签以及最终预测结果如下：

1）．在 test_size=0.2,random_state=1234 时候，准确率为 96%;

```
test_size: 0.2
random_state: 1234
准确率: 96.0 %
```

2）．在 test_size=0.1,random_state=4231 时候，准确率为 100%。

```
test_size: 0.1
random_state: 4321
准确率: 100.0 %
```

5. **结论分析：**

结果验证了所构建 LDA 模型的有效性，通过对于片段的主题分布进行分类，可以较为准确的识别出属于哪一篇小说。

通过本次实验，我掌握了构建 LDA 模型的流程，更加熟练得掌握了文档的数据预处理，并且可以利用 LDA 模型进行一些简单的文字片段分类任务。

6. **代码汇总：**

详细工程文档见。这里展示文档功能划分：

pre.py　文档预处理

exp3.py 片段生成与 LDA 模型构建以及数据集主题分布生成

svm.py 支持向量机分类