作业二：探究和利用 LDA 模型

Abstract:

从给定的中文语料库中均匀抽取 1000 个段落作为数据集（每个段落可以有 K 个 token, K 可以取 20，100，500, 1000, 3000），每个段落的标签就是对应段落所属的小说。利用 LDA 模型在给定的语料库上进行文本建模，主题数量为 T，并把每个段落表示为主题分布后进行分类（分类器自由选择），分类结果使用 10 次交叉验证（i.e. 900 做训练，剩余 100 做测试循环十次）。

实现和讨论如下的方面：

（1）在设定不同的主题个数 T 的情况下，分类性能是否有变化？

（2）以"词"和以"字"为基本单元下分类结果有什么差异？

（3）不同的取值的 K 的短文本和长文本，主题模型性能上是否有差异？

Introduction

LDA(Latent Dirichlet Allocation)中文翻译为：潜在狄利克雷分布。LDA 主题模型是一种文档生成模型，是一种非监督机器学习技术。它认为一篇文档是有多个主题的，而每个主题又对应着不同的词。一篇文档的构造过程，首先是以一定的概率选择某个主题，然后再在这个主题下以一定的概率选出某一个词，这样就生成了这篇文档的第一个词。不断重复这个过程，就生成了整篇文章。它最早由 Blei 等人在 2003 年提出，旨在通过对文本数据进行分析，自动发现其隐藏的主题结构。被广泛应用于文本挖掘、信息检索、自然语言处理等领域。

LDA 算法的输入，是一个文档的集合：D={d1, d2, d3, ..., dn}，同时也需要主题的类别数量 m。算法的处理过程是，将每一篇的文档的 di 对应主题上的一个概率值 p，由此，每一篇文档都将会得到一个概率的集合，也即表示文档 di 在 m 个主题上的概率值。文档中的所有词汇会求出它对应的每个主题的概率，就可以获得一个文本到主题、一个词到主题的两个矩阵。其核心公示为：P(词 | 文档)=P（词 | 主题）P（主题 | 文档），表达式表达如下：

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

以主题 t 作为它的中间层，可以通过 θd 和 φt 给出文档 d 中出现单词 w 的概率。其中 p(t|d)可以通过 θd 计算取得，p(w|t)可以通过 φt 计算得出。

利用 θd 和 φt 能够为任意一个文档中的任意一个单词计算它对应任何一个主题时的 p(w|d)，然后根据这些结果来更新这个词应该对应的主题。之后，如果该项更新改变了这个单词所对应的主题，都会影响 θd 和 φt。

LDA 算法开始阶段，先随机给 θd 与 φt 赋值，（对所有的 d 与 t）。之后不断迭代重复上述过程，最终会收敛到一个结果，即 LDA 算法的输出。LDA 算法将文档与词投影到一组主题上。通过主题可以找出文档和词之间、文档和文档之间、词和词之间的潜在关系。LDA 算法属于无监督算法，每个主题并不会被要求要指定条件，但是聚类之后，通过统计出各个主题上词的概率的分布情况，寻找出在该主题上概率高的词，就能比较好地描述该主题的意义。

Methodology

M1 ： 验证步骤

1、将小说文件录入，构建语料库，包括文档与对应的标签。

2、对语料库中的词语进行分词或分字，在此过程中去除停用词，并取 K 个 token 作为一个段落，形成语料库。

3、均匀地在语料库中取 1000 个段落，将 1000 个段落分成 900 个训练样本和 100 个测试样本。

4、指定主题数 T,对 900 个训练样本进行 DLA 模型建模，输出 900*T 的矩阵，代表所有训练样本的主题分布。

5、利用上述的训练输出和 900 个训练样本对应的标签来训练一个分类器，得到训练样本和预测样本的标签，获得分类器模型，将训练样本和测试样本进行 10 次的交叉验证，从而获得训练样本的进度和测试样本的精度。

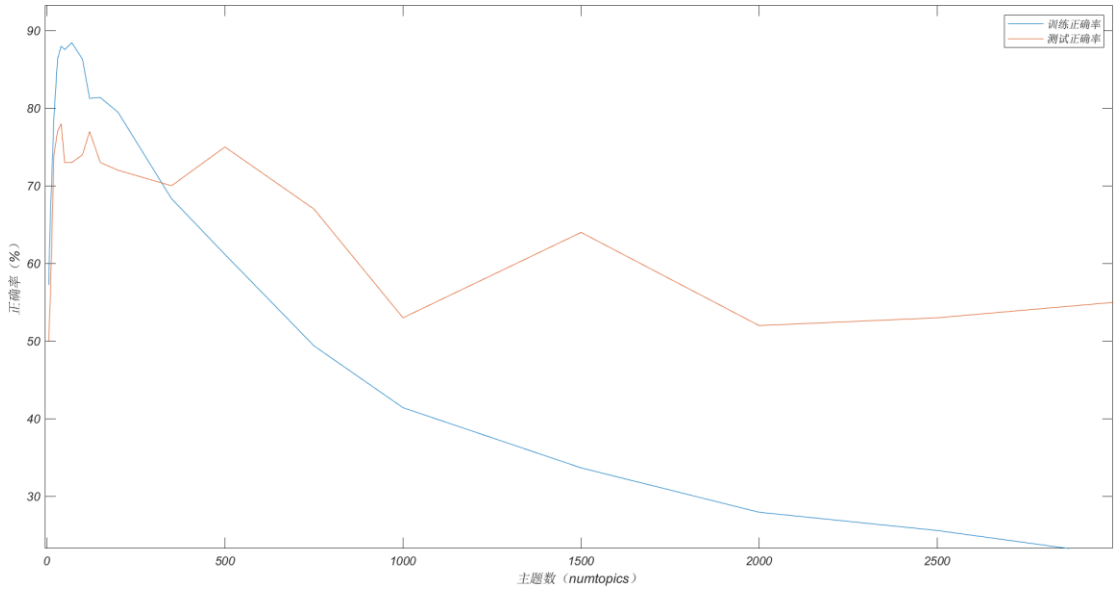
Experimental Studies

M1 ： 设定不同的主题个数 T，分类性能的变化情况？

抽取段落数：num_paragraphs=1000
Token 个数：K=1000
分类器选择：支持向量机（SVC）
主题数:
num_topics=[5,10,20,30,40,50,70,100,120,150,200,350,500,750,1000,1500,2000,2

500,3000]

主题数（num_topics）	训练正确率（%）	测试正确率（%）
5	57.24	50.00
10	66.10	56.00
20	78.90	74.00
30	86.30	77.00
40	87.99	78.00
50	87.54	72.99
70	88.44	73.00
100	86.30	74.00
120	81.257	76.99
150	81.372	73.00
200	79.46	72.00
350	68.34	70.00
500	61.16	75.00
750	49.38	67.00
1000	41.41	52.99
1500	33.66	63.99
2000	27.94	51.99
2500	25.59	52.99
3000	22.44	55.00



总结规律：着主题数增多，正确率逐步提高，当主题数到达 40 左右时，正确率开始逐步下降。因此最优主题数在 40 左右

M2 ： 以"词"和以"字"为基本单元下分类结果有什么差异?

由 M1 可以知道最佳的主题数在 40 左右，因此此处将主题数设为 40

抽取段落数：num_paragraphs=1000
Token 个数：K=1000
分类器选择：支持向量机（SVC）
主题数：num_topics=40

以”字”或以”词”	训练正确率（%）	测试正确率（%）
以”字”	27.52	27.11
以”词”	88.43	73.00

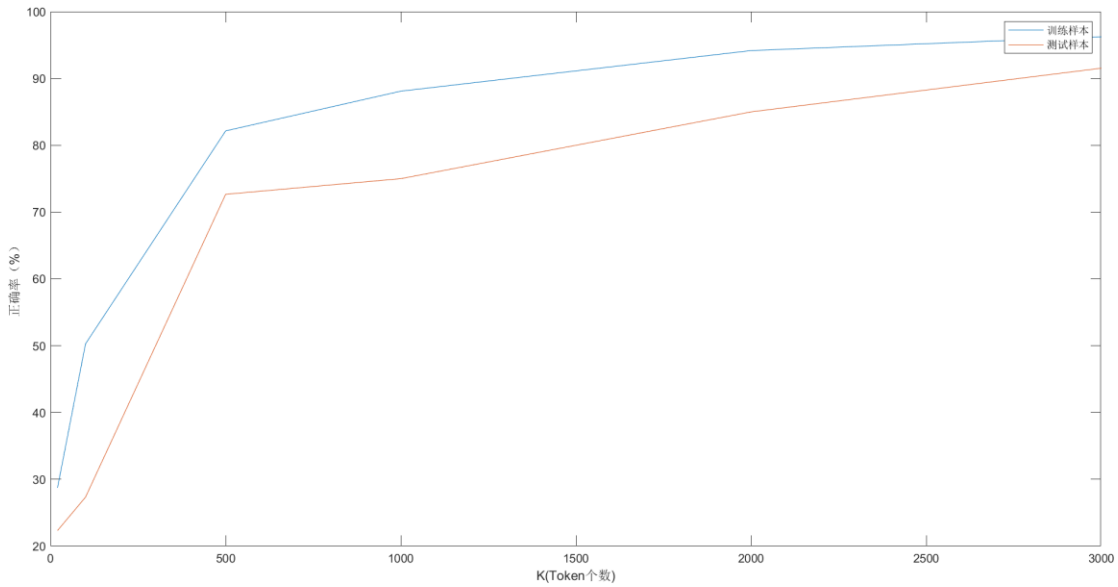
总结规律：以“字”为基本单位的正确率小于以“词”为基本单位

M3 ： 不同的取值的 K 的短文本和长文本，主题模型性能上是否有差异?

抽取段落数：num_paragraphs=1000
分类器选择：支持向量机（SVC）
主题数：num_topics=40

Token 个数：K=[20,100,500,1000,3000]

K(Token 个数)	训练正确率（%）	测试正确率（%）
20	28.73	22.33
100	50.28	27.33
500	82.15	72.66
1000	88.10	75.00
2000	94.17	85.00
3000	96.23	91.54



总结规律：随着 K 的提升，模型的正确率显著提升

M4：不同的分类器，正确率是否不同?

抽取段落数：num_paragraphs=1000
主题数：num_topics=40
Token 个数：K=1000

分类器选择['SVC','RandomForestClassifier','MultinomialNB','LogisticRegression']

分类器有：支持向量机（SVC），随机森林分类器（RandomForestClassifier），多项式朴素贝叶斯分类器（MultinomialNB），逻辑回归分类器（LogisticRegression）

分类器	训练正确率（%）	测试正确率（%）
支持向量机	88.33	78.99
随机森林分类器	90.23	72.00
多项式朴素贝叶斯分类器	83.16	57.00
逻辑回归分类器	87.74	67.00

总结规律：机森林分类器的效果最好，更适合成为分类器

M5 总结

主题数选择：T=40

以“词“为基本单元

文本长度选择：3000

分类器选择随机森林分类器

训练正确率（%）	测试正确率（%）
93.49	85.00