作业 2：利用 LDA 模型进行文本建模

黄家懿 ZY2303203

zy2303203@buaa.edu.cn

Abstract

从给定的语料库中均匀抽取 1000 个段落作为数据集，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，分类结果使用 10 次交叉验证。分别实现和讨论不同主题数量 T、不同 token 数量 k、不同基本单元“词”和“字”对 LDA 主题模型分类性能的影响。

Introduction

- LDA 模型

LDA（Latent Dirichlet Allocation，潜在狄利克雷分布）模型是自然语言处理中常用来进行文本分类的主题模型。这种模型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。如图 1 所示是 LDA 模型下构造一篇文档的过程。假设语料库 D 由 M 篇文档组成，$D=\{W_1,W_2,...,W_M\}$，其中一篇文档 W 又包含 N 个词，$W=\{w_1,w_2,...,w_N\}$，则从语料库 D 中生成文档 W 的过程可表示为：

Step1：从狄利克雷分布中随机采样文档 W_m 的主题分布分布 $\vec{\vartheta}_m$，$\vec{\vartheta}_m \sim Dir(\vec{\alpha})$；

Step2：对文档 W 中的第 n 个词 ω_n 进行话题指派，且 $z_{m,n} \sim Multionmial(\vec{\vartheta}_m)$；

Step3：根据指派话题 z_n 所对应的词频分布 $\vec{\varphi}_k \sim Dir(\vec{\beta})$ 随机采样生成词 $w_{m,n}$。其中，$\beta \in \mathbb{R}^{k\times V}$，$k$ 为话题数目，V 为词表大小，且 $\beta_{ij}=p(w^j \mid z^i=1)$。

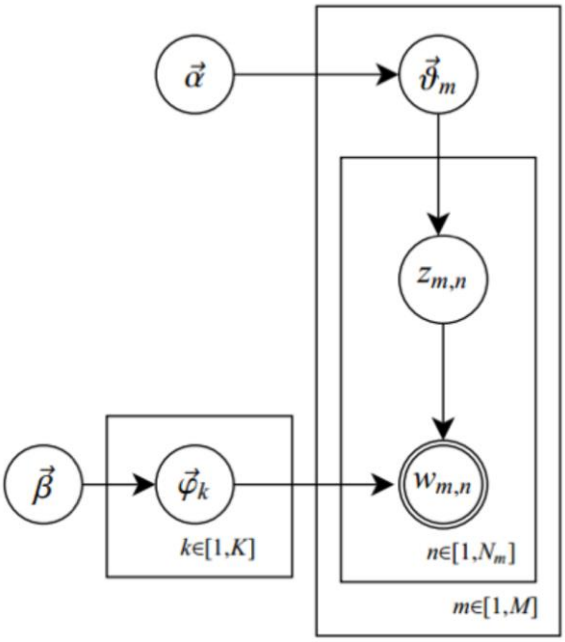

图 1 使用 LDA 模型构造一篇文档

- SVM 分类器

SVM（support vector machines，支持向量机）分类器是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。SVM 学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。如图 2 所示，$\mathbf{w}\cdot x+b=0$即为分离超平面，对于线性可分的数据集来说，这样的超平面有无穷多个（即感知机），但是几何间隔最大的分离超平面却是唯一的。

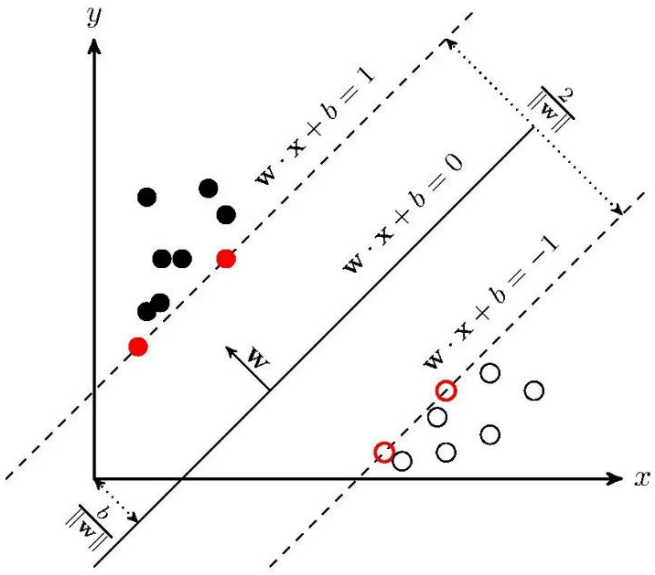

图 2 SVM 分类器

Methodology

M1: LDA 模型

- 实验环境：

  Python 3.7

  PyCharm Community Edition 2023.1

  jieba 0.42.1

- 实验数据：

  中文语料库：包括《白马啸西风》、《碧血剑》等小说的 txt 格式文档

  标点符号库：cn_punctuation.txt

  中文停词库：cn_stopwords.txt

- 建立 LDA 模型的程序基本流程如图 1 所示：

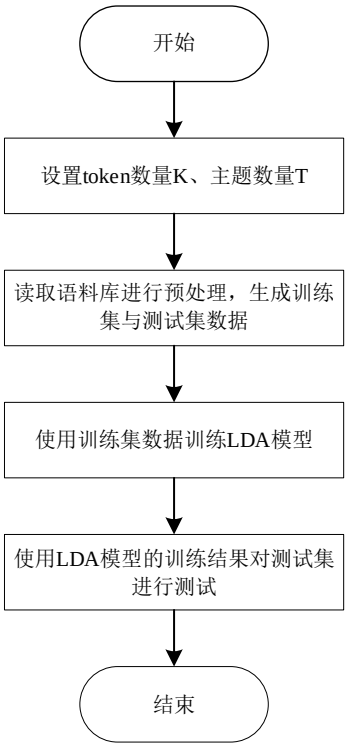

图 3 LDA 模型的程序流程图

- 实验代码：

  使用训练集训练 LDA 模型：

  Step1：给每个文档 *Document* 的每一个单词 *Word* 随机分配一个主题 *Topic* ；

  Step2：计算每个 *Document* 中每个 *Topic* 的概率分布：

$$Freq(Topic_j \mid Document_i) / Count(Document_i) = P(Topic_j \mid Document_i) \quad (1)$$

其中，$Freq(Topic_j \mid Document_i)$ 是每个 *Document* 中，属于各个 *Topic* 的 *Word* 数量，$Count(Document_i)$ 是每个 *Document* 中的 *Word* 总数量

Step3：计算每个 *Topic* 中，每个 *Word* 的概率分布：

$$Freq(Word_k \mid Topic_j) / Count(Topic_j) = P(Word_k \mid Topic_j) \quad (2)$$

其中，$Freq(Word_k \mid Topic_j)$ 是每个 *Topic* 中，各个 *Word* 的数量，$Count(Topic_j)$ 是每个 *Topic* 中的 *Word* 总数量。

Step4：假设每个 *Document* 中的每个 *Word* 是由最大概率生成，即：

$$P(Word_k \mid Document_i) = P(Word_k \mid Topic_j) \times P(Topic_j \mid Document_i) \quad (3)$$

据此，可以选出该 *Word* 下概率最大的 *Topic*，并重新分配给 *Word*，重复 Step4，直到没有概率更大的新 *Topic*。

M2: SVM 分类器模型

- 实验环境：

  Python 3.7

  PyCharm Community Edition 2023.1

  jieba 0.42.1

- 通过 SVM 分类器模型实现分类的程序基本流程图如图 2 所示：

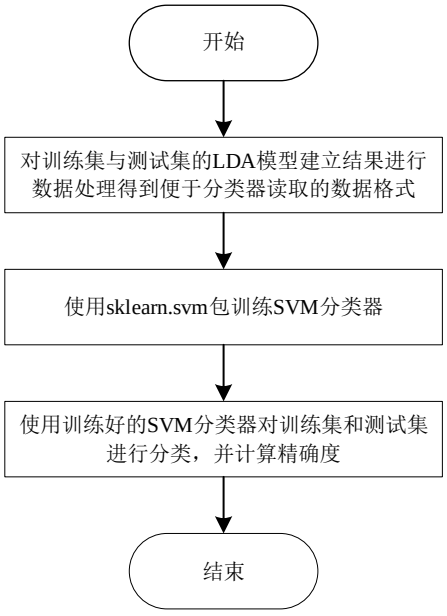

图 4 计算中文信息熵的程序流程图

- 实验代码：

  使用 sklearn.svm 包中的 SVM()方法训练 SVM 分类器。

Experimental Studies

（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？

表 1 所示是不同主题个数下，K=500，以词为基本单元的实验结果。

表 1 不同主题个数

主题数量 T	训练集分类精确度	测试集分类精确度
2	0.9978	0.9300
3	0.9967	0.9706
4	0.9989	0.8300
5	1.0000	0.8700
6	0.9911	0.8235
7	0.9843	0.8286
8	1.0000	0.8269
9	0.9668	0.7130
10	1.0000	0.7300
11	0.9591	0.8455
12	0.9550	0.7407
13	0.9491	0.7692
14	0.9410	0.8036
15	0.9288	0.7714
16	0.9193	0.7946

根据实验结果可以看出，随着主题数量 T 的增加，训练集与测试集的分类精确度均下降。这是因为当主题数量 T 增加，分类的可能性变多，但数据量不增加，因此分类的难度变大，要想在主题数量增加的情况下保持较高的分类精确度，可以使用更多的段落数或者提高每个段落的 token 数量。

（2） 以"词"和以"字"为基本单元下分类结果有什么差异？

表 2 是在不同主题个数下，分别以“词”和以“字”为基本单元，K=500 的实验结果。

表 2 分别以“词”和“字”为基本单元

主题数量 T	以“词”为基本单元		以“字”为基本单元	
	训练集分类精确度	测试集分类精确度	训练集分类精确度	测试集分类精确度
2	0.9978	0.9300	0.9611	0.9100
3	0.9967	0.9706	0.9654	0.9188
4	0.9989	0.8300	0.9522	0.9400
5	1.0000	0.8700	0.95	0.9100
6	0.9911	0.8235	0.9430	0.9216
7	0.9843	0.8286	0.9348	0.8571
8	1.0000	0.8269	0.9509	0.8558
9	0.9668	0.7130	0.9259	0.7870
10	1.0000	0.7300	0.9378	0.7900
11	0.9591	0.8455	0.9148	0.8818
12	0.9550	0.7407	0.9178	0.7778
13	0.9491	0.7692	0.9095	0.7500
14	0.9410	0.8036	0.8980	0.7768
15	0.9288	0.7714	0.8972	0.7238
16	0.9193	0.7946	0.8568	0.7143

对比同一主题下分别以“词”为基本单元和以“字”为基本单元的实验结果，可以发现，以“词”为基本单元的训练集精确度高于以“字”为基本单元的训练集精确度，二者的测试集精确度差别不大。

对比不同主题数量下以“字”为基本单元的实验结果，可以发现，随着主题数量的增加，训练集分类精确度与测试集分类精确度均逐渐下降，这一点与以“词”为基本单元的实验结果是相似的。

（3） 不同取值的 K 的短文本和长文本，主题模型性能上是否有差异？

表 3 所示是主题数量 T=10 下，K 取不同值，以词为基本单元的实验结果。

表 3 不同 token 数量 K

token 数量 K	训练集分类精确度	测试集分类精确度
20	0.8767	0.21
100	1	0.43
500	1	0.73
1000	1	0.9
3000	1	0.99

观察实验结果，可以发现，随着 token 数量的增加，训练集分类精确度和测试集分类精确度都提高。这是因为随着 token 数量的增加，每个文段包含的数据增多，在相同的数量下，提高数据量能提高模型训练精度，从而提升分类精确度。

Conclusions

- 在其他条件相同的情况下，随着主题数量 T 的增加，无论是以“词”还是“字”为基本单元，分类的效果变差；
- 在其他条件相同的情况下，以“词”为基本单元和的训练集分类精确度高于以“字”为基本单元的训练集分类精确度，者的测试集精确度差别不大；
- 在其他条件相同的情况下，随着 token 数量 K 的增加，训练集分类精确度和测试集分类精确度都提高。

References

[1] https://zhuanlan.zhihu.com/p/31470216

[2] https://www.zhihu.com/tardis/zm/art/31886934?source_id=1005