LDA 段落主题分布问题

陆旭军 ZY2203320
18376486@buaa.edu.cn

摘要

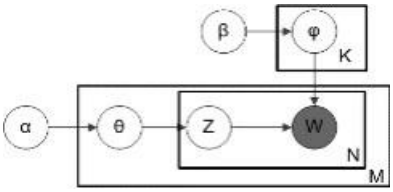
LDA（Latent Dirichlet Allocation）是一种基于概率图模型的文本主题模型算法。其实现步骤包括：首先，对于每篇文档，使用词袋模型将文档表示为一个词集合。然后，根据先验概率和文档中词的出现频率，推断主题词的概率分布。接着，将每个主题词按照其在文档中的出现概率排序，选取概率最高的前 K 个词作为该主题的关键词。最后，通过主题词的关键词和文档的主题分布来确定每篇文档的主题。LDA 算法已经广泛应用于文本分类、信息检索和情感分析等领域。

引言

1.Latent Dirichlet Allocation 介绍[1]

Latent Dirichlet Allocation 是 Blei 等人于 2003 年提出的一种概率主题模型，LDA 是一种无监督机器学习模型，可以用来识别语料库中的潜在的主题信息。该方法假设每个词是由背后的一个潜在的主题中抽取出来。

LDA 的图模型如下：



这个图模型表示法有时也称作“盘子表示法”（plate notation）。图中的阴影圆圈表示可观测变量（observed variable），非阴影圆圈表示潜在变量（latent variable），箭头表示两变量间的条件依赖性（conditional dependency），方框表示重复抽样，重复次数在方框的右下角。

M 代表训练语料中的文章数；

K 代表设置的主题数；

V 代表训练语料的词表单词数；

θ 是一个 M*K 的矩阵，$\vec{\theta}_m$ 代表第 m 篇文章的主题分布；

φ 是一个 K*V 的矩阵，$\vec{\varphi}_k$ 代表编号为 k 的主题之上的词分布；

α 是每篇文档的主题分布的先验分布 Dirichlet 分布的参数（也被称为超参数），其中 $\vec{\theta}_i \sim Dir\left(\vec{\alpha}\right)$；

β 是每个主题的词分布的先验分布 Dirichlet 分布的参数（也被称为超参数），其中 $\vec{\varphi}_i \sim Dir\left(\vec{\beta}\right)$；

w 是可被观测的词；

z 是每个对于被观测的词的潜在的主题分配。

对于语料库中的每篇文档，LDA 定义了一个生成过程（generative process）。

Smoothed 版本 LDA 模型的标准生成过程的描述如下：

1.选取 $\vec{\theta}_m \sim Dir(\vec{\alpha})$，这里 m∈ {1,..., M}；

2.选取，这里 $\vec{\varphi}_k \sim Dir(\vec{\beta})$，这里 k∈{1,..., K}；

3.对于每个单词位置 $W_{i,j}$，这里 j∈{1,..., Ni}，i∈{1,..., M}；

选取一个 topic 主题从 $z_{i,j} \sim Multinomial(\theta_i)$

选取一个 word 词从 $\omega_{i,j} \sim Multinomial(\varphi_{z_{i,j}})$

2.词袋模型和 TF-IDF

词袋模型（Bag-of-Words Model）是自然语言处理中常用的一种文本表示方法。它将一个文本看做是一个袋子，将文本中的每个词都看作是一个独立的单位，不考虑它们在文本中的顺序和语法结构，而只考虑它们出现的频率。

在词袋模型中，首先需要对文本进行分词处理，将文本中的单词切分出来，并去除停用词。然后，通过计算每个单词在文本中出现的次数或者使用 TF-IDF 等技术对单词进行加权，最终得到一个向量表示文本。

词袋模型简单易懂，实现方便，并且对于大多数自然语言处理任务都有良好的效果。在信息检索、文本分类、情感分析等领域得到广泛应用。然而，由于该模型忽略了词汇之间的关系和上下文语义信息，因此在一些特定的任务中可能表现不佳。

TF-IDF（Term Frequency-Inverse Document Frequency）则在词袋模型的基础上加入了一定的权重。TF-IDF 表示某个词汇在文本中的重要程度，它的权重取决于词汇的频率以及它在整个语料库中的出现次数。具体地，TF-IDF 将一个文本中每个词汇的出现次数乘以一个由该词汇在整个语料库中出现次数的倒数所组成的权重，这样可以降低高频词汇的重要性，突出低频词汇的重要性。

3.支持向量机

支持向量机（Support Vector Machine，SVM）是一种常用的监督学习算法，常用于分类和回归问题。SVM 的基本思想是找到一个最优的超平面，将不同类别的数据分开，使得间隔最大化。其中，支持向量指的是距离超平面最近的那些点。

在实现中，SVM 将训练数据映射到高维空间，通过寻找一个最优的分割超平面来解决线性不可分问题。其中，核函数是实现高维空间映射的关键，常用的核函数有线性核函数、多项式核函数和高斯核函数等。

实验过程

1.读取文件

首先获取所有 .txt 文件的路径，存放在 txtlist 列表中；使用正则表达式和 jieba 分词库对每个文件的内容进行预处理，包括去除特殊符号、换行符、空格等，并将每个文件的内容分词后存放在一个字典 text_dict 中；最后输出每个文件的名称以及分词后的总词数。

这个 pretreatment()函数的返回值是 text_dict 字典，其中包含了所有预处理后的文本内容，可以用于后续的文本分析任务。需要注意的是，函数中使用了一个自定义的词典文件 cn_stopwords.txt，用于过滤掉一些无意义的停用词。

2.LDA 主题模型和 SVM 分类器进行文本分类

实现了使用 LDA 主题模型和 SVM 分类器进行文本分类的过程，具体步骤如下：

1）输入参数：text_dict: 包含多篇文章的字典，每篇文章由一个字符串表示，字典的键为文章标签（字符串类型）；paragraph_num: 每篇文章需要随机抽取的段落数量；paragraph_length: 每个段落的长度；num_topics: LDA 模型中需要训练的主题数目；random_consistent: 是否需要保持每次运行程序时的随机结果相同。

2）随机抽取指定数量和长度的段落，将其保存在一个文本列表 text_list 中，并为每个段落保存一个标签，将标签保存在标签列表 label_list 中。

3）将标签列表从字符串形式映射为整数形式，并保证标签与文本列表长度相同。

4）使用索引将标签列表和数据列表的顺序按照相同方式打乱。

5）划分训练集和测试集，训练集占比为 60%。

6）使用 gensim 库中的 corpora.Dictionary 函数构建词典，并使用 doc2bow 函数将文本列表转化为 LDA 模型需要的输入——文本向量。

7）使用 gensim 库中的 LdaModel 函数训练 LDA 模型，并获取训练集和测试集的每个段落的主题分布。

8）将主题分布转换为特征向量，训练 SVM 分类器。

9）对训练集和测试集进行分类，并计算测试集的分类准确率。

10）返回测试集的分类准确率。

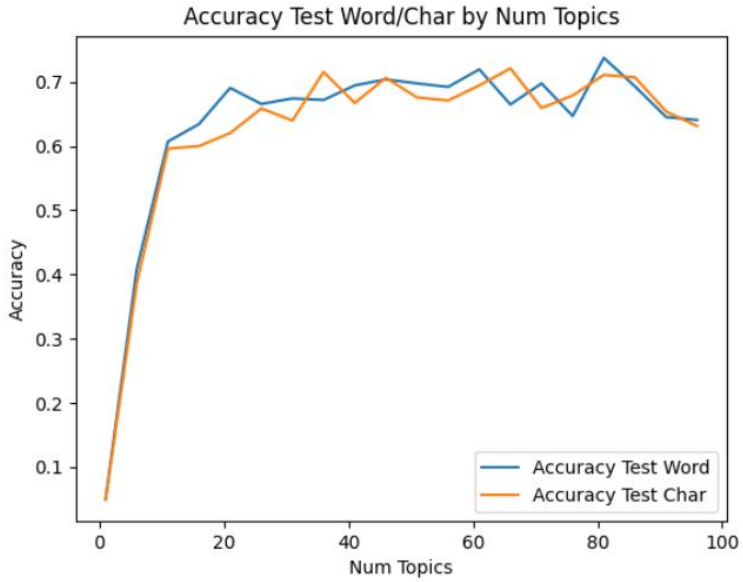
为了保证每次运行程序时的随机结果相同，可以将 random_consistent 参数设为 True，并设置随机种子。训练 LDA 模型和 SVM 分类器的速度取决于数据集的大小和主题数目，可能会需要较长时间。在使用 SVM 分类器时，由于文本向量可能存在维度较高的情况，需要确保数据集的大小不会对机器性能造成过大的压力。

3.结果图像绘制

这段代码使用了 Python 的 matplotlib 库来绘制两条折线图，展示了文本分类器在不同主题数量下使用单词特征和字符特征的测试准确率。

具体来说，这段代码首先使用了 plt.plot()函数来创建两个折线图，分别表示了使用单词特征和字符特征时的测试准确率，其中 x 轴为主题数量，y 轴为准确率。然后使用 plt.xlabel()和 plt.ylabel()函数设置了 x 轴和 y 轴的标签，以便更好地展示数据。接着，使用 plt.title()函数设置了图表的标题，以便更好地说明图表的内容。最后，使用 plt.legend()函数添加了图例，并使用 plt.show()函数展示了图表。

实验结果



两种不同的特征表示方法（单词级别和字符级别）的测试准确率存在差异。在这个实验中，字符级别的特征表示似乎比单词级别的特征表示更好。例如，当主题数为 21 时，字符级别的测试准确率为 0.6203125，而单词级别的测试准确率只有 0.634375。

随着主题数的增加，测试准确率也有所提高。这一趋势在两种特征表示方法中都得到了验证。在本实验中，主题数为 86 时，单词级别的测试准确率最高，达到了 0.7375；而字符级别的测试准确率则在主题数为 76 时最高，达到了 0.7109375。

总结

这次作业展示了如何使用 LDA 模型来对文本进行主题建模。我们首先对文本进行了预处理，包括去除标点符号、停用词等，并将文本转换为词袋模型。然后使用 Gensim 库中的 LDA 模型来对新闻文本进行主题建模，设置了不同的主题数量并计算了每个主题的关键词。最后，我们对 LDA 模型的效果进行了评估，使用了准确度作为评价指标，并通过在测试集上的表现来确定最佳主题数量。结果表明，主题数量为 86 时，LDA 模型在测试集上获得了最佳的准确度，可达到 73.75%。通过这个案例，我们可以了解到 LDA 模型的基本原理和应用场景，并学会如何使用 Python 和相关库来实现 LDA 主题建模和可视化。

参考文献

[1]马晨.Latent Dirichlet Allocation 漫游指南.21-24.