目录

贾云飞 ZY2303207

摘要

本文研究了使用 Latent Dirichlet Allocation（LDA）模型对语料库中的 1000 个段落进行文本主题分类的方法，并用 200 个语料库新段落进行分类测试。LDA 是一种无监督的概率生成模型，用于发现文本数据中的隐藏主题。

1. 简介

在文本挖掘中，文本分类是一项基本任务，其目标是根据一定的标准将文本数据分组。随着互联网信息的爆炸式增长，研究文本分类的有效方法已成为一项重要课题。本文针对 1200 个段落的文本数据进行主题分类及测试，采用了 Latent Dirichlet Allocation（LDA）模型，以发现文本中的隐藏主题，并使用随机森林模型进行标签分类。本文将详细介绍实验方法、实验过程和实验结果，以验证 LDA

模型和随机森林模型在文本主题分类方面的有效性。

2. 实验方法

Latent Dirichlet Allocation（LDA）模型是一种基于概率的生成主题模型，其基本假设是文档由一定比例的主题构成，而每个主题又由一定比例的词汇构成。给定一个文档集合，LDA 模型可以通过对每个文档中的词进行统计分析，发现其中的隐藏主题。LDA 模型的输入为一个文本集合，输出则是每个文本所属于不同主题的概率分布。它首先对文本集合中的每一篇文档进行处理，并将文档表示为一个单词的序列。然后，对于每个主题，使用 Dirichlet 分布生成该主题下单词的概率分布。接着，对于每个文档，从主题分布中随机选择一个主题，再从所选主题的单词概率分布中随机选择一个单词。

LDA 是基于贝叶斯模型的，涉及到贝叶斯模型离不开“先验分布”，“数据（似然）”和"后验分布"三块。在朴素贝叶斯算法原理小结中我们也已经讲到了这套贝叶斯理论。

先验分布 + 数据（似然）= 后验分布

对于是非问题可以使用二项分布进行解答。为了使得后验分布可以作为下一次判断的先验分布，所以我们希望先验分布和后验分布的形式尽量一样，与二项分布共轭的为 Beta 分布

$$Beta(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{1}$$

其中Γ为 Gamma 函数，满足$\Gamma(x) = (x-1)!$

超过二维的 Beta 分布我们一般称之为狄利克雷(以下称为 Dirichlet)分布。也可以说 Beta 分布是 Dirichlet 分布在二维时的特殊形式。

Dirichlet 分布的表达式为：

$$Dirichlet(\vec{p}|\vec{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k-1} \tag{2}$$

并且 Dirichlet 分布的期望有如下性质：

$$E\big(Dirichlet(\vec{p}|\vec{\alpha})\big) = \left(\frac{\alpha_1}{\sum_{k=1}^{K}\alpha_k}, \frac{\alpha_2}{\sum_{k=1}^{K}\alpha_k}, \dots, \frac{\alpha_K}{\sum_{k=1}^{K}\alpha_k}\right) \tag{3}$$

Beta 分布和二项分布的共轭关系在 Dirichlet 中也可以体现：

$$Dirichlet(\vec{p}|\vec{\alpha}) + multi(\vec{m}) = Dirichlet(\vec{p}|\vec{\alpha} + \vec{m}) \quad (4)$$

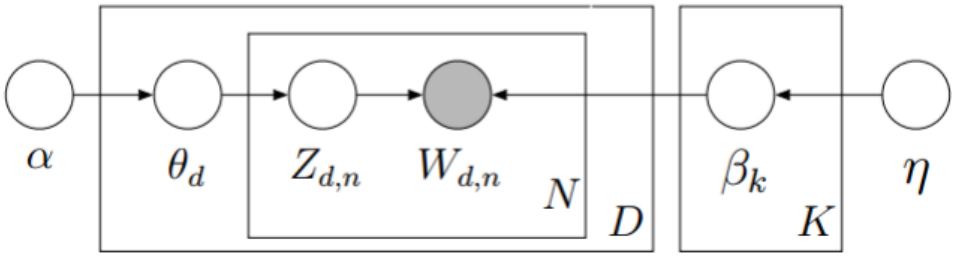

图 1 LDA 模型示意图

图中不同部分表示的含义为：α表示一个文档集；θ_d表示被抽取的某一个文档的主题分布；z_{dn}表示每一个词的在篇文档的主题分布，有先验 Dirichlet 分布得到；w_{dn}被观测到的词本身；β_k表示对应某一主题的词分布；η表示所有的词汇。

LDA 模型假设文档的先验分布为 Dirchlet 分布，即对于任一文档d，主题分布θ_d为：

$$\theta_d = Dirichlet(\vec{\alpha}) \quad (5)$$

其中，α为分布的超参数，是一个 K 维向量。

LDA 假设主题中词的先验分布是 Dirichlet 分布，即对于任一主题 k，词分布β_k

$$\beta_k = Dirichlet(\vec{\eta}) \quad (6)$$

其中η为分布的超参数，是一个V维向量。V用来表示词汇表里所有词的个数。

对于数据中任一一篇文档 d 中的第 n 个词，我们可以从主题分布θ_d中得到它的主题编号z_{dn}的分布为

$$z_{dn} = multi(\theta_d) \quad (7)$$

而对于这个主题编号，我们实际看到的词w_{dn}的概率分布为：

$$w_{dn} = multi(\beta_{z_{dn}}) \quad (8)$$

这个模型里，我们有 M 个文档主题的 Dirichlet 分布，而对应的数据有 M 个主题编号的多项分布，这样$(\alpha \rightarrow \theta_d \rightarrow \vec{z}_d)$就组成了 Dirichlet-multi 共轭。可以使用前面提到的贝叶斯推断的方法得到基于 Dirichlet 分布的文档主题后验分布。

如果在第 d 个文档中，第 k 个主题的词的个数为：$n_d^{(k)}$，则对应的多项分布的计数可以表示为

$$\vec{n}_d = \left(n_d^{(1)}, n_d^{(2)}, \ldots, n_d^{(K)}\right) \quad (9)$$

利用 Dirichlet-multi 共轭，得到θ_d的后验分布为

$$Dirichlet(\theta_d|\vec{\alpha}+\vec{n}_d) \tag{10}$$

对于主题与词的分布，有 K 个主题与词的 Dirichlet 分布，而对应的数据有 K 个主题编号的多项分布，我们可以根据前面的方法得到基于 Dirichlet 分布的后验分布。

利用 Dirichlet-multi 共轭，得到β_k的后验分布为

$$Dirichlet(\beta_k|\vec{\eta}+\vec{n}_k) \tag{11}$$

最后我们可以通过使用 EM 方法来进行参数的迭代求解。

LDA 模型的原理如下：

确定主题数量 K；

对于每个主题 k：

a. 按照狄利克雷分布生成一个主题-词分布；

对于每个文档 d：

a. 按照狄利克雷分布生成一个文档-主题分布；

b. 对于文档 d 中的每个词 w：

i. 从文档-主题分布中采样一个主题 z；

ii. 从主题-词分布中采样一个词 w；

iii. 将词 w 分配给主题 z；

通过迭代优化，得到最终的文档-主题和主题-词分布。

3. 实验过程

3.1 数据预处理

3.1.1 停词处理

停词包括文章中的标点以及常见语气助词等无实意的字或词，在查看语料库中，我们发现文章中仍有一些英文出现，并且有一些标点在被特定语言读取过程中可能会出现转义的情况，为了防止以上情况出现对于 LA 模型构建的影响，我们在原有停词表中做了以下处理：

在停词表中增加小写、大写英文共 52 个字母；在停词表中，对于一些特定

标点增加转义字符比如将“\”调整为”\\”；新增一些停词比如英文的逗号“,”等。

3.1.2 分词处理

本文使用了 python 的 jieba 库来进行中文词汇的分词，该库的主要任务是将读取的字符串，按照数据库中的中文词汇，将中文字符串分成多个词组，便于后面进行 LDA 模型构建以及分类。

3.1.3 数据的获取和处理

数据集为金庸先生的 16 本小说，其中包含了大量乱码与无用或重复的中英文符号，因此需要对该实验数据集进行预处理。

具体来说就是，首先要删除无意义的字符比如空格、回车、制表符、段落符；其次根据已经获取的停词表，将相应的停词，如标点、英文字母、阿拉伯数字、无实意语气词等进行删除；最后可以选择直接使用汉字，或者使用 jieba 库将长字符串进行分词，得到多个词组。

在处理的过程中，为了尽可能选择分词数比较多的段落满足不同 token 需要，在选择段落的时候段落尽可能长。在此 16 本武侠小说中选择字数较长的五篇小说，分别为《倚天屠龙记》、《笑傲江湖》、《天龙八部》、《射雕英雄传》、《鹿鼎记》，每篇小说随机选取 240 段用于 LDA 实验，其中 200 段用作训练集，40 段用作测试集。通过实际处理发现，这五篇文章同时满足字数最大段落数约为 250。因此在每本小说中随机抽取字数大于 250 的段落共 1200 段用于训练和测试。

同时，为了便于后续检测准确率，我们需要生成两个标签列表，分别用于训练集和测试集，标签列表长度分别为 1000、200，每个段落的标签就是该段落出自的小说，我们用 0-4 的数字来便捷表示。

3.2 建立 LDA 模型

3.2.1 模型的输入和搭建

首先建立词频的数据表。使用 gensim 的库，之后将文本数据转换为一个词项（token）的集合，然后使用 Dictionary 对象来构建一个文档-词项矩阵（词袋模型）。构建词袋模型后，使用 gensim.models.ldamodel.LdaModel 来创建模型的实例。

3.2.2 参数设置

- num_topics: 设计一个 T 列表代表主题数，为[5，10，50，100，200，400]。
- corpus：训练集词袋模型，用于生成模型。
- alpha: 集中参数（也称为狄利克雷分布的超参数），用于主题分布的平滑，在此设置为’auto’。
- eta: 集中参数，用于词项分布的平滑。
- id2word: 一个 Dictionary 对象，它将词项映射到它们的整数引。
- chunksize: 在每次迭代中处理的文档数。
- passes:整个数据集的迭代次数。
- update_every: 指定了在训练过程中，模型的状态更新（即执行吉布斯采样并更新主题分布）的频率，设置为默认值 1。
- per_word_topics: 设置为 True，为每个词项在每个文档中生成一个主题分布。

为检验 LDA 模型效果，使用 gensim 库中 lda_model.log_perplexity 计算 Logarithmic Perplexity。

Logarithmic Perplexity（对数困惑度）：困惑度是衡量语言模型性能的一个指标，它衡量的是模型对真实数据分布的拟合程度。由于直接计算困惑度可能会得到非常小的数值，因此通常取其对数。对数困惑度不仅数值更易于处理，而且当用于比较不同模型的性能时，对数形式的加和减操作更为直观。log_perplexity 越低，表示模型对训练数据的预测越准确，模型性能越好。

在主题建模中，评估一个主题模型的质量是非常重要的。其中一个常用的评估指标是主题的一致性（coherence）。CoherenceModel 是 gensim 库中用于评估 LDA 模型主题一致性的类。它通过分析主题中词项的分布来评估模型的一致性。执行 coherence_model_lda.get_coherence()后，coherence_lda 将包含计算得到的一致性得分，该得分可以用来评估 LDA 模型的质量。一致性得分的范围通常在 0 到 1 之间，得分越高表示主题的一致性越好，模型的质量越高。

3.3 建立分类模型

3.3.1 模型的输入和搭建

对于分类模型采用随机森林模型，随机森林（Random Forest，简称 RF）是一种集成学习算法，它属于监督学习算法中的一种，常用于分类和回归问题。随机森林通过构建多个决策树来进行训练，并利用这些树的预测结果来进行最终的决策。下面是随机森林的一些关键特点和工作原理：

1.集成多个决策树：随机森林由多个决策树组成，每棵树都是独立构建的，并且在构建过程中引入随机性。

2.自助采样（Bootstrap sampling）：每棵决策树都是在数据集的一个随机子集上训练的，这个子集是通过有放回抽样（即自助采样）得到的。

3.特征选择的随机性：在决策树的每个决策点，随机森林不是考虑所有可能的特征，而是随机选择一部分特征，然后从中选择最佳分裂特征。

4.投票机制：对于分类问题，随机森林通过多数投票的方式来确定最终的预测结果。即，对于一个输入样本，森林中的每棵树都会给出一个预测，然后统计各个类别的票数，票数最多的类别将作为最终预测结果。

5.减少过拟合：由于引入了随机性和集成多个模型，随机森林通常能够有效地减少过拟合，提高模型的泛化能力。

6.特征重要性评估：随机森林可以评估各个特征对预测的贡献度，这有助于特征选择和了解数据。

7.适用性广：随机森林可以处理数值型和类别型数据，不需要太多的数据预处理步骤。

8.并行处理：由于每棵树是独立训练的，随机森林可以很容易地并行化，这有助于在大规模数据集上提高训练效率。

3.3.2 参数设置

设置决策树个数 n_estimators 为 300， 随机种子 random_state 为 10， 最高树高 max_depth 为 9。

3.3.3 十折交叉验证

十折交叉验证（10-fold cross-validation）是一种常用的模型评估方法，用于评估机器学习模型的性能。它将数据集分成 10 个大小相等（或尽可能相等）的子集，然后执行以下步骤：

1. 选择一个子集作为测试集：从 10 个子集中选择一个作为测试集（也称为验证集或保留集），剩下的 9 个子集合并作为训练集。

2. 训练模型：使用训练集上的数据训练模型。

3. 评估模型：使用测试集上的数据评估模型的性能。通常会计算一个性能指标，如准确率、召回率、F1 分数等。

4. 重复过程：将剩下的 9 个子集轮流作为测试集，重复步骤 1 到 3。

5. 计算平均性能：计算 10 次评估的平均性能指标，得到模型的最终性能估计。

在此使用十折交叉验证对分类模型结果进行评估，并训练随机森林模型进行测试集测试分类结果与实际结果对比得到准确性。

4. 实验结果

生成一个 csv 表格 lda_topics 来表示不同段落对于符合不同主题的的概率，如下图所示，表格数值为某个段落符合不同主题的概率。

图 2 不同段落符合不同主题的概率

以主题数为 50，分词分类为例，下图数据包括训练集段落数、测试集段落数、LDA 模型对数困惑度、LDA 模型主题一致性得分、随机森林分类器交叉验

证分数、平均分数、测试集实际标签、测试集分类标签，模型准确率。

```
训练集段落数: 1000
测试集段落数: 200
Perplexity:  -28.417725502080785
Coherence Score:  0.4264618538065562
交叉验证分数: [0.48 0.53 0.01 0.   0.52 0.52 0.02 0.05 0.   0.  ]
平均分数: 0.213
实际标签: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3
, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
预测标签: [0, 0, 3, 1, 3, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 1, 3, 0, 0, 3, 0, 2, 0, 0, 0, 3, 0, 1,
4, 0, 4, 1, 1, 1, 3, 3, 1, 1, 1, 4, 4, 2, 4, 1, 0, 4, 3, 3, 2, 4, 3, 4, 4, 1, 2, 4, 4, 2, 4, 4, 2, 2, 2, 4, 0, 1, 4, 2, 2, 2, 4, 2, 2,
 2, 2, 2, 0, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 4, 2, 2, 2, 2, 1, 1, 3, 4, 2, 2, 2, 2, 2, 1, 4, 2, 1, 2, 2, 3, 0, 0, 4, 0, 4, 1, 4, 4, 3, 4
, 3, 3, 3, 3, 4, 1, 2, 4, 1, 3, 4, 4, 3, 4, 2, 3, 2, 3, 3, 3, 0, 3, 3, 2, 0, 3, 3, 3, 4, 4, 4, 4, 4, 4, 1, 1, 4, 4, 0, 2, 4, 1, 0, 4,
4, 4, 4, 0, 1, 4, 4, 4, 4, 0, 4, 1, 4, 3, 4, 4, 4, 2, 4, 1, 3, 3, 3, 3, 4]
模型准确率: 0.53
```

图 3 实验结果示意图

接下来根据实验要求进行不同控制变量实验。

4.1 设置不同的主题数进行分类

分类主题数如下图所示：

```
T = [5, 10, 50, 100, 200, 400] #主题数
```

图 4 段落获取情况

设置将段落进行分词分类，不限制 token 数，得到结果如下表：

表 1 不同主题分词分类后分类性能表

主题数	Log Perplexity	Coherence Score	十折交叉验证平均分数	模型准确率
5	-11.291	0.425	0.233	0.54
10	-12.176	0.432	0.194	0.45
50	-28.418	0.426	0.213	0.53
100	-47.386	0.444	0.175	0.55
200	-84.808	0.436	0.276	0.61
400	-160.233	0.437	0.239	0.60

4.2 分别使用分词和分字进行主题分类

设置主题数为 50，分别以"词"和以"字"为基本单元下进行分类，得到分类结果如下表：

表 2 相同主题数（50）分词和分字分类后分类性能表

基本单元	Log Perplexity	Coherence Score	十折交叉验证平均分数	模型准确率
字	-9.231	0.510	0.716	0.86
词	-28.418	0.426	0.213	0.53

4.3 使用不同 token 个数进行分类

分析所有段落最小分词数为 59，故设置 token 个数区间为[5，10，20，40，50，基于段落]，同时设置主题数为 50，得到使用不同 token 数分词分类结果如下表：

表 3 不同 token 数分词分类后分类性能表

Token	Log Perplexity	Coherence Score	十折交叉验证平均分数	模型准确率
5	-52.704	0.539	0.035	0.26
10	-48.666	0.463	0.056	0.23
20	-44.763	0.329	0.046	0.24
40	-40.279	0.349	0.086	0.30
50	-38.901	0.399	0.082	0.27
基于段落	-28.418	0.426	0.213	0.53

4.4 实验结果分析

对数困惑度衡量的是模型对真实数据分布的拟合程度，主题一致性反映了 LDA 模型发现的主题与实际文本内容的相关性，而主题分类准确率则反映了 LDA 模型将文档正确分配到对应主题的能力。

（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情况：

由表 1 可知，在其他参数相同的情况下，随着主题个数的增加，Log Perplexity 越来越高，Coherence Score 基本保持在一定范围内，模型准确率越来越高，但增大到一定程度基本不变。说明随着主题数增加，模型对真实数据分布的拟合程度越来越高，主题与实际文本内容的相关性波动比较小，同时对提高模型标签分配的准确性有帮助。

（2）以"词"和以"字"为基本单元下分类结果差异：

由表 2 可知，在其他参数相同情况下，分字的 Coherence Score 和模型准确率均高于分词，而且模型准确率高三分之一，分析原因可能分词的准确性不高，导致丢失重要信息或引入噪声，从而影响 LDA 模型和随后的随机森林分类器的性能，同时以字为单位可能会捕捉到更多的信息，还可能会产生更大的特征空间，这可能导致模型更为复杂，能够捕捉更多的细节，使生成模型的准确率更高。但分字的 Log Perplexity 低于分词，可能分词得到的模型更能反应对真实数据分布

的拟合程度。上述结论是基于表 2 得到，可能随着主题数变化或模型参数变化或者测试集、训练集发生变化等产生不一样的结果。

（3）不同的取值的 K 的短文本和长文本，主题模型性能上差异：

由表 3 可知，在其他参数相同的情况下，随着 token 个数增加，Log Perplexity 值越来越小，可能分词数较小，更能与真实文本数据进行匹配，但匹配程度还是不如主题数较大的情况。Coherence Score 也有减小趋势。但模型预测准确率只有 token 数基于段落的一半左右，分析原因可能是 token 列表的选取是基于所有段落中最小的分词数，实际情况下大部分段落的分词数都远大于 token，因此在进行模型训练和分类的时刻可能无法准确判断段落所属标签，之后实验情况下可据此进行优化。

最后观察表 1、2、3 发现基于分词的十折交叉验证结果均很不理想，与模型测试的结果出入较大，没有达到想要的结果。实验中采用 sklearn.model_selection 库中的 KFold 和 cross_val_score 函数进行十折交叉验证。分析原因可能是，每一段分词的次数不相同，交叉验证划分训练集和测试集时的分层采样中，训练集和测试集在类别分布上存在较大差异，同时模型可能在训练集上存在过拟合问题。对于十折交叉验证结果的优化也是下次实验中需要解决的。

综上所述，基于 LDA 模型和随机森林分类器对中文段落进行建模不同情况下结果分析如上。由于实验抽取五本武侠小说一部分段落，根据抽取段落所属章节不同，其分类结果可能也有所不同，同时所选小说均为同一作者的武侠小说，除人名外其他有意义分词可能在分类过程中对分类器造成混淆，影响结果。但根据划分出的训练集和测试集得出 LDA 模型以及随机森林分类模型仍有一定的准确度，可用于作为文本分类和主题探究实验的参考。

5. 结论

本文通过对 1200 个段落进行文本主题分类及测试的实验研究，验证了 LDA 模型在文本主题分类方面的有效性。实验结果表明，LDA 模型能够有效地发现文本中的隐藏主题，同时也有一定的准确率。但 LDA 模型在处理大规模文本数据时可能面临计算复杂度和收敛速度的挑战，未来研究可探讨如何优化 LDA 模型以应对这些挑战。