深度学习与自然语言处理课程报告
——基于 LDA 模型对文本进行分类和预测

姓名：龙行健

学号：ZY2203110

一、 摘要

LDA 模型为一种聚类模型，其可以分为两个 Dirichlet 分布和两个多项式分布，采用文档-主题-单次的思想，对给定的文本库进行无监督学习。本报告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。验证与分析分类的有效性，结果表明，LDA 建模在一定程度上具有精确性，并且分类精度对应某个合适的主题数量选取，采用词对中文文本进行处理比采用字的方式效果更好。

二、 引言

LDA（Linear Discriminant Analysis），是一种文档主题生成模型，，它可以将文档中每篇文档的主题按照概率分布的形式给出。也称为一个三层贝叶斯概率模型，包含词、主题和文档三层结构。所谓生成模型，就是说，我们认为一篇文章的每个词都是通过“以一定概率选择了某个主题，并从这个主题中以一定概率选择某个词语”这样一个过程得到。文档到主题服从多项式分布，主题到词服从多项式分布。

LDA 是一种非监督机器学习技术，可以用来识别大规模文档集（document collection）或语料库（corpus）中潜藏的主题信息。它采用了词袋（bag of words）的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。但是词袋方法没有考虑词与词之间的顺序，这简化了问题的复杂性，同时也为模型的改进提供了契机。每一篇文档代表了一些主题所构成的一个概率分布，而每一个主题又代表了很多单词所构成的一个概率分布。

LDA 的核心思想是寻找到最佳的投影方法，将高维的样本投影到特征空间(feature space)，使得不同类别间的数据“距离”最大，而同一类别内的数据“距离”最小。

三、 实验原理

1. LDA 模型原理

如前文所述，LDA 模型通过两个 Dirichlet 分布和两个多项式分布进行单词的概率生成，具体如下所示：

$$P(\boldsymbol{W},\boldsymbol{Z},\boldsymbol{\theta},\boldsymbol{\varphi};\alpha,\beta)=\prod_{j=1}^{M}P(\theta_j;\alpha)\prod_{i=1}^{K}P(\varphi_i;\beta)\prod_{t=1}^{N}P(Z_{j,t}\mid\theta_j)P(W_{j,t}\mid\varphi_{Z_{j,t}})$$

等式左边为生成某一特定文档的概率，其由右边四个部分所决定，其中，W,Z 分别为单词、主题所代表的序列，θ,φ 为训练所得的 Dirichlet 分布参数，分别对应文档-主题以及主题-单词。$\prod_{j=1}^{M}P(\theta_j;\alpha)$ 项代表某次该文档生成的主题分布，对应要从 $\prod_{t=1}^{N}P(Z_{j,t}\mid\theta_j)$ 这个多项式分布中得到对应的主题，同理，$\prod_{i=1}^{K}P(\varphi_i;\beta)$ 项代表某次该主题生成的单词分布，对应要从 $\prod_{t=1}^{N}P(W_{j,t}\mid\varphi_{Z_{j,t}})$ 这个多项式分布中得到对应的单词。

2. LDA 模型生成

经典的 LDA 模型采用 Gibbs 采样原理进行生成，具体过程如下：

1. 初始化文档中每个单词的主题，随机选取即可；
2. 对于每一个文档中的单词，采用 Gibbs 采样算法进行主题的重新估计；
3. 重复步骤 2，直到所有单词的主题都被更新过一遍；
4. 重复步骤 2 和步骤 3，直到模型的参数估计收敛或者达到指定的迭代次数；

Gibbs 采样方法如下所示：

对于每个 word 在文档中：

对于每个 topic：

文档生成该单词的概率 = 该 topic 生成该单词的概率 * 文档生成该 topic 的概率

通过上述 topic 序列更新该单词的最大 topic 概率。

四、 实验过程

1. 实验环境

带有 jieba 开源库和基础科学运算的 python 环境

2. 数据来源

1) 统计数据集

金庸武侠小说 16 部。

2) 停词表

包括非中文字符，本次实验中没有用上无实义助词作为停词表。

3. 数据预处理

这部分主要对金庸武侠小说数据集进行三个操作：数据集读取、去除不计入统计的词汇、采用 jieba 中文语料库进行分词操作。

```
def Preproccess(data_root, del_root, aft_del_root):
    data_list_dir = os.listdir(data_root)
    del_list_dir = os.listdir(del_root)
    aft_del_list_dir = os.listdir(aft_del_root)
    data_corpus = []
    del_corpus = []
    aft_del_corpus = []
    cha_count = 0

    #First preprocess
    for del_file_name in  del_list_dir:
        del_file_path = del_root + '/' +str(del_file_name)
        print(del_file_path)
        with open(os.path.abspath(del_file_path), "r", encoding = 'utf-
8') as f:
            del_context = f.read()
            del_corpus.extend(del_context.split("\n"))
    for data_file_name in  data_list_dir:
```

```
        data_file_path = data_root + '/' +str(data_file_name)
        print(data_file_path)
        with open(data_file_path, "r", encoding = 'ANSI') as f:
            data_context = f.read()
            data_context = data_context.replace("本书来自 www.cr173.com 免
费 txt 小说下载站\n 更多更新免费电子书请关注 www.cr173.com", '')
            data_context = data_context.replace("\n", "")
            data_context = data_context.replace(" ", '')
            data_context = re.sub('\s','',data_context)

            for del_word in del_corpus:
                data_context = data_context.replace(del_word, "")
            cha_count += len(data_context)
            data_corpus.append(data_context)

    data_corpus = jiebaCut(data_corpus)
def jiebaCut(corpus):
    new_corpus = []
    for text in corpus:
        words = jieba.cut(text)
        new_corpus.extend(words)
    return new_corpus
```

4. LDA 模型训练

在给定了训练参数 alpha，beta 和主题个数下，对模型采用前文提到过的 Gibbs 采样原理进行训练，关键代码如下：

```
                # Gibbs sampling update the word topic
                gibbs_p = []
                for k in range(topic_num):
                    p = (topic_set[k].get(word, 0) +
alpha)/(topics_word_num[k] + alpha)
                    p *= (doc_topic_num[i][k] + beta)/(doc_word_num +
beta)
                    gibbs_p.append(p)
                gibbs_p = np.array(gibbs_p)
                upd_toc = np.random.choice(topic_num, p = gibbs_p /
gibbs_p.sum())
```

其中，topic_set 为存储了每个 topic 中对应当前单词及其个数的一个词典，可以用来查询该单词在不在当前 topic 中，并且数量有多少。

5. LDA 模型测试

给定一篇文章，同样看可以利用 LDA 模型得到的参数对其进行分类，具体做法和 LDA 模型的训练类似，关键在于此时每个 topic 的单词概率分布是已知的，我们只用训练得到该文档每个单词对应的主题即可。

五、 实验结果

1. 测试数据结果：

在设定的 topic 数量为 40 的情况下，经过训练后得到的前 10 个 topic 数据如下所示：

topic 1：[('范蠡', 139), ('道', 70), ('勾践', 52), ('吴国', 35), ('薛烛', 34)]

topic 2：[('胡一刀', 33), ('只见', 24), ('金面佛', 24), ('左手', 20), ('竟', 19)]

topic 3：[('萧中慧', 56), ('周威信', 48), ('卓天雄', 48), ('麽', 45), ('著', 41)]

topic 4：[('说道', 26), ('王夫人', 18), ('举人', 18), ('兆', 15), ('田伯光', 14)]

topic 5：[('道', 58), ('万震山', 32), ('水笙', 28), ('狄云', 24), ('原来', 23)]

topic 6：[('说道', 96), ('听', 65), ('见', 65), ('想', 60), ('倒', 40)]

topic 7：[('道', 180), ('便', 175), ('中', 122), ('走', 103), ('听', 57)]

topic 8：[('道', 79), ('曹云奇', 55), ('麽', 48), ('田青文', 30), ('众人', 29)]

topic 9：[('一个', 23), ('便', 23), ('中', 20), ('林平之', 19), ('小说', 15),]

topic 10：[('阿青', 48), ('道', 40), ('姑娘', 34), ('竹棒', 32), ('一个', 23),]

2. 分类结果：

在训练集选取每篇文章部分段落 500 个单词，测试集选取每篇文章部分段落后 500 单词的情况下，最后得到的测试集文章推测如下表所示：

[[14 14 14 14 14 14 14 14 14 14 5 14 14]

[14 14 1 14 5 14 14 14 14 14 14 14 14]

[11 14 14 11 14 14 14 14 11 14 14 14 14]

[14 14 11 14 12 12 14 14 14 14 3 14 3]

[14 6 14 5 14 5 14 14 5 14 14 14 14]

[14 5 5 11 14 5 14 5 14 14 5 5 5]

[6 6 6 6 6 6 6 6 6 6 6 6 6]

[14 14 14 14 14 7 14 14 14 14 14 14 5]

[6 5 14 14 14 8 14 8 14 14 14 11 14]

[14 14 14 11 11 14 14 14 11 14 14 14 6]

[10 10 10 10 10 10 10 10 10 10 10 15 15]

[11 11 11 11 11 11 11 11 11 11 11 11 11]

[12 12 12 12 12 12 5 12 12 14 12 14 14]

[14 14 14 14 12 14 14 14 14 3 14 14 14]

[14 14 14 14 14 14 14 14 14 14 14 14 14]

[14 15 14 14 15 14 15 15 14 15 15 14 14]]

可以看到，LDA 模型的训练结果不理想，模型的准确性只有 20%左右。

六、 结论

上述实验和结果表明，LDA 模型的分类还是非常具有局限性的，可能的原因有如下几点：对于大型的文本来说，不同的 topic 和迭代次数对于结果有非常大的影响，合适的 topic 和迭代次数选取需要很好的经验；每个 topic 所包含的词和个数不一样，往往会有重复性的无意义的词对其权重形成影响。后续对 LDA 模型进行改进，可以考虑加入 N 元词语模型，使得其文本生成更加完好。