**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

# 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。

# 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。

# 实验原理

## 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：



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，分别为单词、主题所代表的序列，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。项代表某次该文档生成的主题分布，对应要从这个多项式分布中得到对应的主题，同理，项代表某次该主题生成的单词分布，对应要从这个多项式分布中得到对应的单词。

## 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。

# 实验过程

## 实验环境

带有jieba开源库和基础科学运算的python环境

## 数据来源

1. **统计数据集**

金庸武侠小说16部。

1. **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。

## 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

## 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。

## 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。

# 实验结果

## 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),]

## 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。

## 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。