毕业项目结题报告

问题定义

项目概述

本项目旨在训练一个机器学习模型来对 20 个类别的新闻数据进行自动分类。数据来源于公开数据集 20Newsgroups，网址为 http://www.qwone.com/~jason/20Newsgroups/。该数据集的形式为英文文本类型，包含 20 个类别的新闻数据，各个类别文本数量大体一致，并已经经过一定的处理。

文本分类问题是自然语言处理领域的经典问题，文本分类在垃圾短信识别、社交网站有害评论过滤等方面有着广泛的应用。针对文本分类的算法包括朴素贝叶斯、支持向量机、决策树、主题模型、神经网络等，对于不同类型的文本，算法各有优劣。然而由于人类自然语言的复杂性，还没有统一的算法框架解决文本分类问题。

之所以选择文本分类问题是因为在此之前参加过一个 kaggle 比赛，该比赛是针对垃圾文本的分类，因而对自然语言领域的处理方法有所了解，希望能通过这个项目把自己在比赛中所学到的文本处理方法加以应用，同时加深对自然语言处理的理解。

问题陈述

数据集包含了 20 个类别的新闻文本，每篇文本归属于一个类别，对它们进行分类是机器学习问题中的有监督学习问题。本项目需要对数据集划分出训练集和测试集，对每一篇文本抽取出文本特征，以文本特征作为输入，文本类别作为标签，训练一个监督学习模型。

首先我会做数据探索和预处理。数据探索即探索文本特点，如文本长度，文本所包含的词汇数量，不同词汇的词频，文本类别是否均衡等。通过数据探索达到对数据的深入理解，为数据预处理做准备。数据预处理即从非结构化的文本中

抽取文本特征，包含数据清洗和特征提取。数据清洗即去掉无效的文本，去掉文本中的非 utf8 编码、标点符号、进行小写转化等。特征提取主要是对英文单词词形还原并提取出文档的 tf-idf 特征。

特征提取完成后即可训练模型，本项目中我选取了 80%数据做训练和交叉验证，剩余 20%数据做测试集，通过交叉验证选取最佳模型作为最终的模型。理想的分类器应该能够对大部分文本做到正确分类，并在评价指标上有较好的分数。本项目中选取了朴素贝叶斯模型、用于多分类的 LR 模型、XGBOOST 模型作为分类器，分别训练得到最优模型，并通过评价指标比较它们之间的差异。

评价指标

分类问题可以采用的评估指标有准确率、查准率、查全率、F1 分数、AUC 等。准确率的定义如下，

准确率=预测正确的样本数量/样本总数量

准确率从总体上衡量了有多少比例的样本被正确预测，准确率越高，模型预测效果越好。在样本类别不平衡时，除了采用准确率指标还应结合查准率、查全率、F1 分数等指标综合评估模型效果。对于二分类问题，查准率 P，查全率 R 和 F1 分数定义如下，

$$P=TP/(TP+FP)$$

$$R=TP/(TP+FN)$$

其中 TP,FP,TN,FN 分别为真正例数，假正例数，真负例数，假负例数。对于多分类问题有相应的“宏查准率”，“宏查全率”，“微查准率”，“微查全率”，“宏 F1”，“微 F1”指标，详细定义可参考[1]。

由于本项目所使用的数据集中，各个类别的样本数量比较均衡，因此本项目采用准确率作为模型预测效果的评估指标，同时辅以宏 F1 分数，以便于基准模型作比较。

除了模型预测效果以外，同等硬件条件下，模型复杂度及训练所需要的时间也是需要考量的因素，实际工业级的应用中，有时会在模型预测效果和所需时间之间做出权衡。

分析

数据的探索

数据集中的数据是一些英文短文本，示例如下：

From: mathew <mathew@mantis.co.uk>
Subject: Re: university violating separation of church/state?
Organization: Mantis Consultants, Cambridge. UK.
X-Newsreader: rusnews v1.01
Lines: 29

dmn@kepler.unh.edu (...until kings become philosophers or philosophers become kings) writes:
> Recently, RAs have been ordered (and none have resisted or cared about
> it apparently) to post a religious flyer entitled _The Soul Scroll: Thoughts
> on religion, spirituality, and matters of the soul_ on the inside of bathroom
> stall doors. (at my school, the University of New Hampshire) It is some sort
> of newsletter assembled by a Hall Director somewhere on campus. It poses a
> question about 'spirituality' each issue, and solicits responses to be
> included in the next 'issue.' It's all pretty vague. I assume it's put out
> by a Christian, but they're very careful not to mention Jesus or the bible.

文本头部包含文本的来源、组织、主题等，正文部分是约 200~300 词的文本。文本中除了大写和小写形式的常用英文单词外还有邮箱地址，人名和地名等，此外还有标点符号和一些特殊符号，如>，@,_等，此外文本中还含有一些非 UTF8 编码的字符，在数据预处理中这些特殊字符都被当作噪音去掉。

文本预处理中除了对标点符号和一些特殊符号去掉之外，还对所有大写字母进行了小写转换，对名词的复数形式进行了词形还原并去掉了文章中的停用词。值得注意的是，在垃圾文本分类的任务中，特殊符号和标点符号不能简单地去掉，

因为垃圾文本与正常文本相比通常包含许多特殊字符（如表情符号），如果把这些符号去掉，便失去了重要特征。而在本项目的分类任务中，并没有特别多的特殊符号，且对鉴别文本来说并没有特殊的含义，所以本项目中去掉特殊符号和标点符号只保留英文单词并没有太多特征信息上的损失。对单词进行词形还原和去停用词是英文自然语言处理中的常用手段，单词的单复数形式在语义上并没有太大的差别，而停用词并不能提供语义上的一些信息，所以这样来处理是恰当的。将所有的大写字母转换为小写字母也是基于以上的原因。
数据集中各个类别的文本数量如下表，可以看出各个类别的数量基本均衡。

category	numbers	category	numbers
comp.graphics	973	comp.os.ms-windows.misc	985
talk.religion.misc	628	talk.politics.guns	910
comp.sys.mac.hardware	963	rec.motorcycles	996
comp.sys.ibm.pc.hardware	982	sci.crypt	991
alt.atheism	799	sci.electronics	984
talk.politics.mideast	940	rec.sport.baseball	994
sci.space	987	comp.windows.x	988
sci.med	990	soc.religion.christian	997
rec.sport.hockey	999	rec.autos	990
misc.forsale	975	talk.politics.misc	775

数据集中最常出现的 20 个单词及出现的次数如下，

('ax', 62548), ('edu', 20692), ('m', 17275), ('wa', 13606), ('line', 13089), ('x', 12526), ('subject', 12294), ('com', 11996), ('organization', 11394), ('c', 11315), ('u', 11200), ('re', 10590), ('d', 10194), ('one', 9193), ('w', 8937), ('q', 8732), ('would', 8543), ('r', 8392), ('p', 8094), ('g', 8087)

这些单词出现频率较高，但是并不能提供关于文章类别的一些信息，这里并没有对这些单词进行特殊的处理，后面进行 tf-idf 特征提取时，df 值比较大的词汇权重自然会变小。

算法和技术

在特征提取和模型训练中，本项目用到了以下算法。

朴素贝叶斯算法：

朴素贝叶斯分类是一种十分简单的分类算法，叫它朴素贝叶斯分类是因为这种方法的思想真的很朴素，朴素贝叶斯的思想基础是这样的：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。朴素贝叶斯算法优缺点如下，

朴素贝叶斯的主要优点有：

1）朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。

2）对小规模的数据表现很好，能个处理多分类任务，适合增量式训练，尤其是数据量超出内存时，我们可以一批批的去增量训练。

3）对缺失数据不太敏感，算法也比较简单，常用于文本分类。

朴素贝叶斯的主要缺点有：

1） 理论上，朴素贝叶斯模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为朴素贝叶斯模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。而在属性相关性较小时，朴素贝叶斯性能最为良好。对于这一点，有半朴素贝叶斯之类的算法通过考虑部分关联性适度改进。

2）需要知道先验概率，且先验概率很多时候取决于假设，假设的模型可以有很多种，因此在某些时候会由于假设的先验模型的原因导致预测效果不佳。

3）由于我们是通过先验和数据来决定后验的概率从而决定分类，所以分类决策存在一定的错误率。

4）对输入数据的表达形式很敏感。

多分类逻辑回归算法

普通的 logistic 回归只能针对二分类(Binary Classification)问题，要想实现多个类别的分类，必须要改进 logistic 回归，让其适应多分类问题。改进方法是修改 logistic 回归的损失函数，让其适应多分类问题。这个损失函数不再笼统地只考虑

二分类非 1 就 0 的损失，而是具体考虑每个样本标记的损失。这种方法叫做 softmax 回归。逻辑回归算法优缺点如下，

优点：

1）预测结果是界于 0 和 1 之间的概率；

2）可以适用于连续性和类别性自变量；

3）容易使用和解释；

缺点：

线性算法，难以拟合复杂的非线性函数

Xgboost

Xgboost 算法是一种梯度提升方法。算法的主要思想是训练多个基模型，取多个基模型的线性加和作为最后的预测结果。基模型的训练中，后训练的基模型对先训练的模型预测错误的样本给与比较大的权重，从而提高预测的准确率。Xgboost 中的基模型可以采用线性模型和树模型。本项目中采用线性模型。

Xgboost 相对于传统的 GBDT 算法，并行性更好，Xgboost 中增加了正则项来控制模型的过拟合。近年来 Xgboost 在各类数据挖掘比赛中得到了广泛的应用，且通常能得到不错的结果。但 Xgboost 中超参数较多，调参比较复杂。

朴素贝叶斯、多分类的逻辑斯蒂回归和 Xgboost 算法都以样本点 Tf-idf 特征作为输入，得到样本属于某个类别的概率，取概率最大的类别作为样本的类别，从而实现分类。

基准模型：

基准模型选择斯坦福大学 NLP 组对相同数据集做的分类，链接 https://nlp.stanford.edu/wiki/Software/Classifier/20_Newsgroups。研究人员尝试了多种方法尝试提高模型的效果，包括对文本进行更好的分词使能包含更多的细节，添加进文本长度等人为构造的特征并按照长度分组作为类别特征，同时使用 L1 和 L2 正则，调整正则项超参数，对大小写不同处理等方法。数据集中 60%用于训练集，40%用于测试集。模型采用的评价方法为准确率、Micro-F1 分数和 Macro-F1 分数，得到的最好结果是训练集上 Micro-F1 分数为 0.90361，Macro-F1

分数为 0.90277，测试集上 Micro-F1 分数为 0.81731，Macro-F1 分数为 0.81158。

方法

数据预处理

部分数据预处理的工作如去除标点符合和特殊符号，大写全部转化为小写等前文已述及，文本特征提取主要用的 Tf-idf 算法，

文档中单词的 Tf-idf 值计算公式如下

Tf-idf = Tf*idf

Tf 表示文章中出现该单词的次数， idf 计算方法为 idf = log(词料库的文档总数/包含该词的文档数+1)。

Tf-idf 算法的优缺点如下，

1.优点是算法的容易理解，便于实现。

2.缺点：IDF 的简单结构并不能有效地反映单词的重要程度和特征词的分布情况，使其无法很好的完成对权值的调整功能，所以在一定程度上该算法的精度并不是很高。除此之外，算法也没有体现位置信息，对于出现在文章不同位置的词语都是一视同仁的。

本项目中采用 ngram 为 3 的模型提取文档 Tf-idf 特征，特征总个数为 239123。

执行过程

文本的 Tf-idf 特征提取出来之后就可以进行模型的训练工作。

首先对数据集中的所有数据提取 tf-idf 特征，并把数据顺序随机打乱，取其中的 80%作为训练集，剩余的 20%作为预测集。

模型训练中采用 10 折交叉验证方法调节超参数，分别得到最优的朴素贝叶斯、多分类逻辑回归和 Xgboost 模型，并用最优的模型在训练集和测试集上分别进行预测，进而评估模型表现。

朴素贝叶斯模型需要调节的超参数为拉普拉斯平滑中使用的参数 alpha，alpha 越大模型复杂度越低，越不容易过拟合反之越容易过拟合。

多分类逻辑回归需要调节的超参数为平方正则项系数 C，C 越大模型越简单，越不容易过拟合，反之越容易过拟合。

Xgboost 调节的超参数为最大树身 max_depth,学习率 learning_rate,平方项正则系数 reg_lambda,绝对值正则系数 reg_alpha。最大树身控制模型的复杂度，max_depth 越大模型越复杂，越容易过拟合，反之越简单；learning_rate 控制模型训练过程中的学习速率，越大训练的越快，但也越不容易达到最优；reg_lambda 和 reg_alpha 同样控制模型的复杂度，越大模型越简单，越不容易过拟合。

模型中超参数的搜索采用网格搜索法。

结果

模型的评价和验证

交叉验证得到的最优模型超参数值如下，

朴素贝叶斯：alpha=0.03

多分类逻辑回归：C=10

Xgboost：max_depth=4，learning_rate=0.1，reg_lambda=0.1, reg_alpha=0.01

三个模型在训练集和测试集上的准确率、宏精确度、宏召回度和宏 F1 分数如下表：

训练集

	准确率	宏精确度	宏召回度	宏 F1 分数
朴素贝叶斯	0.992	0.992	0.991	0.991
多分类逻辑回归	0.999	0.999	0.999	0.999
Xgboost	0.998	0.998	0.998	0.998

测试集

	准确率	宏精确度	宏召回度	宏 F1 分数
朴素贝叶斯	0.885	0.890	0.888	0.888
多分类逻辑回归	0.904	0.910	0.906	0.907
Xgboost	0.901	0.907	0.903	0.904

由于测试集在模型训练过程中从未暴露给模型，而三个模型在测试集上的表现都不错，达到了预期结果，因此结果是可信的。

合理性分析

三个模型在训练集和测试集上的表现都超过了基准模型。

三个模型在训练集上的准确率和 F1 分数都超过了 0.99，在测试集上的表现则在 0.90 附近，因此 3 个模型都存在一定程度的过拟合。三个模型中朴素贝叶斯模型的表现最差，多分类逻辑回归模型表现最佳，Xgboost 次之。逻辑回归模型是线性模型，在本项目中表现最佳可能是因为本项目中提取了文本的 Tf-idf 特征，且采用了 ngram 为 3 的语言模型，每个样本都被一个高维向量表示，因此可能是线性可分的。同样采用线性模型作为基学习器的 Xgboost 模型稍逊于逻辑回归模型有些出乎我的意料，这可能因为 Xgboost 模型超参数较多，而本项目只搜索了部分超参数组合，得到的结果是局部最优的，因此模型表现稍差于逻辑回归模型，若能很好的调整模型的超参数其表现应该优于逻辑回归模型。

模型最终结果表现理想，切实解决了分类的问题。

项目结论

对项目的思考

项目中首先对数据集进行一定程度的探索，随机挑选几个类别的文章打开看文章的大体形式、文档长度、所包含的各种符号等，并阅读文档，以获得对语义的感性认识。根据数据探索过程中对文档的感性认识进行下一步的数据预处理，去掉标点符号、特殊符号等。模型选择中选择了简单的朴素贝叶斯和多分类逻辑

回归模型，复杂一点的集成模型 Xgboost 模型。这三个模型都选择文档的 Tf-idf 特征作为模型的输入，使用交叉验证结合网格搜索法找到模型最佳的超参数。

项目中比较困难的地方在数据预处理部分和 Xgboost 模型超参数调优的过程。数据预处理对整个项目相当关键，预处理的质量决定了模型的上限，特征工程和模型超参数调优只是使结果更接近于这个上限。本项目中数据预处理的难点在于文本中的哪些特征应该保留，哪些应该舍去，比如标点符号、特殊符号等，还有大小写的处理，英文单词不同词形的处理等。本项目中直接去掉了标点和特殊符号，大写全部转化为小写，英文名词复数词形还原并得到了不错的结果。但本项目中的预处理方法并不具有通用性，预处理的细节需要根据不同任务的特点决定，如社交文本处理中不能简单地去掉特殊符号。

Xgboost 调参的难度在于参数众多，网格搜索法耗时较长，因此需要对 Xgboost 各参数对结果的影响有比较深刻的理解，减少超参数的搜索数量。

三个模型在测试集上的表现都超过了基准模型，基本达到了预期效果。然而正如前面所讨论的，由于文本预处理方法依任务不同而有所不同，所以本项目中的方法和模型并不具有通用性，需要根据不同的任务，采取不同的处理方法。

需要作出的改进

数据预处理中简单地去掉了标点符号和大写字母转化为了小写，可以同时在文档特征中加入标点符号数量，大写字母数量等特征弥补去掉这些特征所带来的信息上的损失。

模型超参数搜索中，可以采用随机搜索方法，这样可以迭代更少的超参数，降低超参数落到局部最优的概率，随后在随机搜索得到的最优超参数附近进行网格搜索得到更好的参数组合。

特征提取中本项目也可以根据文本样本训练 PLSA，LDA 等主题模型，得到每篇文档的主题分布作为文档特征进行文档的分类。模型训练中也可以采用 SVM，多层神经网络等模型进行训练。

本项目得到的是单个模型的训练结果，可以多训练一些性能较优的单模型，对它们进行融合和堆叠得到更优的模型。