文本复制检测报告单(全文标明引文)

№:ADBD2018R_2018053015312720180530154839440174168657　　检测时间:2018-05-30 15:48:39

检测文献:　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用

作者:　邱聪荣

检测范围:　中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

CNKI大成编客-原创作品库

个人比对库

时间范围:　1900-01-01至2018-05-30

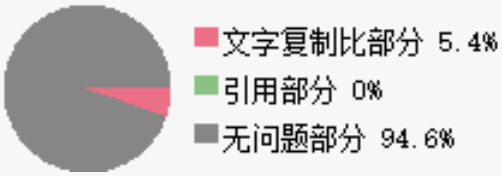
检测结果

总文字复制比：5.4%　　跨语言检测结果：0%

去除引用文献复制比：5.4%　　去除本人已发表文献复制比：5.4%

单篇最大文字复制比：1%（scikit-learn决策树学习 - 每天进步一点点2017 - CSDN博客）

重复字数：	[1627]	总段落数：	[7]
总字数：	[30342]	疑似段落数：	[6]
单篇最大重复字数：	[310]	前部重合字数：	[123]
疑似段落最大重合字数：	[589]	后部重合字数：	[1504]
疑似段落最小重合字数：	[34]		



指　标：☐疑似剽窃观点　☑疑似剽窃文字表述　☐疑似自我剽窃　☐疑似整体剽窃　☐过度引用

表格：0　　公式：2　　疑似文字的图片：0　　脚注与尾注：0

1.6%（87）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第1部分（总5295字）

0%（0）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第2部分（总420字）

9.8%（343）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第3部分（总3500字）

2.6%（236）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第4部分（总9244字）

11.3%（338）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第5部分（总2988字）

7.9%（589）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第6部分（总7481字）

2.4%（34）　53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第7部分（总1414字）

（注释：无问题部分　文字复制比部分　引用部分）

1. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第1部分 总字数：5295
相似文献列表 文字复制比：1.6%(87) 疑似剽窃观点：(0)
1 汪海伟_0503090712_液相还原法制备超细银粉 1.1%（57）
汪海伟 - 《大学生论文联合比对库》- 2013-06-04 是否引证：否
2 《游戏世界》第一期：变形金刚离我们有多远？_超好玩 0.6%（30）
- 《网络（http://blog.sina.com）》- 2015 是否引证：否
原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

吉林大学学士学位论文（设计）承诺书

本人郑重承诺：所呈交的学士学位毕业论文（设计），是本人在指导教师的指导下，独立进行实验、设计、调研等工作基础上取得的成果。除文中已经注明引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的作品成果。对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

学士学位论文（设计）作者签名：

2018年5月20日

摘要

基于sklearn框架KNN分类算法的实现与应用

随着互联网时代的来临，信息处理技术和计算机处理技术在企业的应用中得到了迅猛发展。伴随着电子信息行业的高速发展，行业所产生的数据也变得越来越多，动辄TB量级甚至PB量级的行业大数据使得一般的计算机技术难以处理。数据本身是有价值的，但是数据量太大的话数据本身的价值就难以被挖掘出来。因此，如何有效地分析海量的大数据已经成为各个企业数据分析所必须掌握的技术。大数据在当今的世界中扮演着重要的角色，近年来世界各地的学术界，企业和国家都在寻求有效的大数据处理技术[1]。一些国家机构也从科技战略技术层面提出了许多促进国家地区大数据技术发展的计划，以推进企业，大学和国家机关对大数据处理技术的研究和开发。而在大数据技术中，又以机器学习技术为核心技术。

机器学习技术是数据挖掘的一个重要手段，机器学习在三十多年的发张中已经成为了一门多领域交叉学科，涉及概率论，统计数学，凸分析，计算复杂性理论等多门学科。如果把数据当作资源，那算法就是让资源得以有效利用的机器。机器学习和数据的关系就犹如蒸汽机和煤炭的关系，机器学习理论主要就是设计和分析一些让计算机可以自动学习的算法。机器学习主要分四类：监督学习，无监督学习，半监督学习和增强学习。而在监督学习中又以分类算法为最重要。机器学习中的分类算法主要有朴素贝叶斯、SVM、KNN、决策树、逻辑回归等分类算法。其中朴素贝叶斯分类算法是基于贝叶斯理论的统计学分类算法；SVM分类算法把分类问题转化成寻找分类平面的问题，并通过最大化边界点和分类平面之间的距离来实现分类；KNN分类算法通过寻找和带预测样本距离最近的K个点中种类最多的种类来对待预测样本进行预测；决策树分类算法是一个预测模型，它是一个树形模型，每个非叶节点表示对某个特征属性值的测试，通过对特征属性值进行划分生成分支子树，每个叶节点表示决策树的一个标签类别。

本文的主要目的是分析和比较KNN算法，SVM算法和决策树算法的性能，主要比较的性能指标有算法的运行时间和算法的预测准确率。首先，从理论上了解这三个算法的实现原理以及它们的优缺点；然后在pycharm平台上通过python的sklearn包实现上述三个算法分类器；最后从UCI上分别下载高纬度，低纬度，大数据，小数据等具有代表性的数据，分别用KNN算法分类器，SVM算法分类器和决策树算法分类器进行预测，比较他们的运行时间和预测准确率。实验结果显示，和SVM，决策树算法比较，KNN算法具有实现原理简单，对于一般的中小型，中小维度的数据算法预测准确率高，预测速度快，对于一些高维数据算法也能达到较高的准确率。

关键词：机器学习，分类算法，KNN,SVM，决策树

Abstract

Implementation and application of KNN classification algorithm based on sklearn framework

With the advent of the Internet era, information processing technology and computer processing technology have been rapidly developed in the application of enterprises. As the fast development of electronic information industry, industry data generated by the also becomes more and more, hold a TB scale even PB level industry makes it hard for the general computer technology data processing[1]. The data itself is valuable, but too much data makes the value of the data difficult to be discovered. Therefore, how to effectively analyze large amounts of big data has become a necessary technology for every enterprise data analysis. Big data plays an important role in today's world. In recent years, academia, enterprises and countries all over the world are seeking effective big data processing technology[1]. Some state agencies are also from the technical level of science and technology strategy put forward a lot of plans, to promote the development of national region big data technology in order to promote enterprise, university and national office for research and development of the technology of data processing. In big data technology, machine learning technology is the core technology.

Machine learning technology is an important means of data mining, machine learning in more than 30 years of hair has become a multidisciplinary cross discipline, related to probability theory, mathematical statistics, convex analysis, computational complexity theory and so on different subjects. If you think of data as a resource, an algorithm is a machine that allows resources to be used efficiently. The relationship between machine learning and data is just like that between a steam engine and a coal. Machine learning is mainly divided into four categories: supervised learning, unsupervised learning, semi-supervised learning and enhanced learning. The classification algorithm is the most important in supervised learning. The classification algorithms in machine learning mainly include naive bayes, SVM, KNN, decision tree, logic regression and other classification algorithms. The naive bayesian classification algorithm is a statistical classification algorithm based on bayesian theory. The SVM classification algorithm transforms the classification problem into the problem of finding the classification plane, and realizes the classification by maximizing the distance between the boundary point and the classification plane. KNN classification algorithm is used to predict the prediction samples by finding the most kinds of the nearest K points with the predicted samples. Decision tree classification algorithm is a prediction model, it is a tree model, each non-leaf nodes according to characteristics of a particular attribute value test, based on the characteristics of attribute value division generated branching subtree, each leaf node said the decision tree of a tag categories.

The main purpose of this paper is to analyze and compare the performance of KNN algorithm, SVM algorithm and decision tree algorithm. First, the realization principle of these three algorithms and their advantages and disadvantages are understood theoretically. Then the above three algorithm classifiers are implemented on the pycharm platform through python sklearn package. Finally, respectively from the UCI download high latitudes and low latitudes, big data, such as small data from a nationally representative sample of data, each classifier using KNN algorithm, the SVM classifier classifier and decision tree algorithm to compare their running time and predictive accuracy. , according to the results of the experiment and the SVM decision tree algorithm, KNN algorithm is simple in principle, for small and medium-sized, average prediction accuracy is high, medium and small dimension data algorithm prediction speed, for some of the high-dimensional data algorithm can achieve high accuracy.

Key words: machine learning, classification algorithm, KNN,SVM, decision tree

目录

2. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第2部分 总字数：420
相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)
原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

3. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第3部分

总字数：3500

相似文献列表　文字复制比：9.8%(343)　疑似剽窃观点：(0)

1	基于异常行为辨识的智能监控技术研究 钟志(导师：徐扬生) -《上海交通大学硕士论文》- 2008-06-01	3.2%（113） 是否引证：否
2	基于Mahout的植物识别系统 谢裕光 -《大学生论文联合比对库》- 2016-05-30	2.5%（89） 是否引证：否
3	面向半结构化文本的知识抽取研究 丁玉飞;王曰芬;刘卫江; -《情报理论与实践》- 2015-03-30	1.2%（41） 是否引证：否
4	基于决策树学习的柱状二极管表面缺陷检测系统设计 郭朝伟;张中炜; -《微型机与应用》- 2015-03-25	1.1%（40） 是否引证：否
5	股指期货市场系统风险预警指标体系研究 高一铭;阎国光;张阳;辛明辉; -《统计与决策》- 2013-03-30	1.1%（38） 是否引证：否
6	基于用户行为挖掘的数据流管理技术研究 李军(导师：方滨兴) -《北京邮电大学博士论文》- 2012-06-30	1.0%（36） 是否引证：否
7	基于教据挖掘的企业商务智能系统平台设计 张远新(导师：李银胜) -《复旦大学博士论文》- 2013-09-20	1.0%（34） 是否引证：否
8	大学英语成绩分析系统设计与应用 程绪琦(导师：于学军;刘志峰;刘春宇) -《北京工业大学博士论文》- 2011-12-01	0.9%（32） 是否引证：否
9	基于智能算法的鼠标手势识别的应用研究 成功(导师：陈玉华) -《大连海事大学博士论文》- 2013-06-01	0.8%（29） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第1章绪论

1.1 研究背景

随着计算机技术的高速发展和互联网的快速普及，网络上的数据量程指数型暴涨，人们也越来越意识到数据对于人类社会的重要性。数据正在逐渐成为与人力资源和物质资本同等重要的资源，合理地使用大数据将会成为提高企业竞争力，行业竞争力乃至国家核心竞争力的关键要素[2]。毋庸置疑，人们对大数据的分析是很有必要的，这也使得人们对数据分析的相关技术越来越关注和重视。目前，很多大企业，高校甚至国家都开展了对数据分析的相关研究和应用，力求从数据分析方面获得利益以增强自身在同行间的竞争力。由此可见，数据分析对当今社会的重要性是非常重要的。

在数据分析中，又以算法为其核心。如果把数据比作煤炭，那么算法就是让煤炭发挥其能源作用的蒸汽机。由此可见，虽然对于研究本身来说，数据分析中最重要的还是数据，但是没有算法的话，数据无非是一堆让人看不懂其真正内容的数字。事实上，有关数据分析的研究工作早在1888年Galton[3]研究人类身高和前臂长度的关系时就已经开始了。在数据分析算法中，机器学习算法算得上是数据分析中的常见算法。机器学习在历经三十多年的发展中已经演变成为了一门多领域交叉的学科，涉及到的知识有统计学，概率论以及计算机等多门学科。数据分析的主要任务是分类、聚类、关联分析、时序模式、预测和偏差分析。其中分类算法是通过对样本数据进行分析测试，进而从对样本数据的分析中获得规律以预测同类测试样本的所属类别。按照学习的方式可以将分类算法分成以下几种类别：监督学习（Supervised Learning），无监督学习(Unsupervised Learning)，半监督学习（Semi-supervised Learning）和增强学习(Reinforcement Learning)。监督式学习，是机器学习中的一种方法，可以从训练样本集中学到或建立一个模式，并依此模式推测新的实例的特征。训练样本集是由输入数据（通常是以向量形式输入）和与输入数据对应预期输出所组成。当函数的输出是一个连续的值的时候，就叫做回归分析。当函数的输出是一个分类标签时，称之为分类；无监督式学习，是属于人工智能网络领域的一种算法，其通过对原始资料进行分类，以便达到了

解资料内部结构的目的[4]。和监督式学习网络不同的是，无监督式学习网络在学习的时侯并不知道其分类的结果是否正

确，即没有给出分类标签。它仅对此种网络模型提供输入样本，并且它会自动从这些输入样本中找出其潜在的类别规则。当训练完毕并经过测试后，也可以将之应用到新的样本数据上；半监督学习，是在监督式学习和无监督式学习之间，它综合利用有类标签的数据和没有类标签的数据，来生成合适的分类函数以达到分类目的。它的的基本思想是利用数据分布上的模型假设，建立学习预测模型对未标签的样本进行标签；增强学习又称强化学习，它可以让计算机从一开始什么都不懂，通过不断地学习，从错误中寻找规律，最后找到规律。本文主要研究的是机器学习分类算法中的监督式分类算法。

在机器学习中，常用的分类算法有KNN、SVM、决策树、朴素贝叶斯、逻辑回归以及神经网络。其中，本文主要围绕KNN（K-邻近，K-Nearest Neighbors），SVM（支持向量机，Support Vector Machine）和决策树算法（Decision Tree）展开研究。在实际数据分析中，选择哪个分类算法主要是根据数据本身的特征决定的。k-近邻算法是一种基于实例的分类方法。KNN算法是一个很简单的算法，KNN分类算法的主要思想就是找出距离待预测样本最近的K个训练集的样本，然后看这K个样本中最多的标签类别是哪个，就把待预测样本归于那一类别。其中距离参数可以人为规定，可以是曼哈顿距离，欧式距离等。k-近邻算法属于惰性学习算法，它在训练阶段只存放样本，到分类阶段才开始进行计算，如果训练样本集数据量比较大的话，很可能会导致很大的计算开销，在分类时，很可能需要很大的存储开销。所以没有办法很实时地应用。KNN方法不足之处主要有以下几点: (1)算法分类速度较慢。(2)距离权重的选取将会影响准确率。(3)对训练样本集的依懒性较强。(4)需要反复尝试以选取最佳K值[5]。同时，由于KNN算法的简单性以及对于一般的数据的有效性使得目前KNN算法仍然受到业界的欢迎。SVM也叫做支持向量机，它是把数据从低维空间映射到高维空间。在高维空间中找出能够将这些数据分类的最优超平面，最后根据这个超平面来对数据进行分类。SVM对训练集之外的数据的预测效果也很好，它具有较小的计算开销，较低的泛化错误率以及结果易于解释等优点，其对高维数据效果依然很好，并且其能够在训练样本数量较少的情况下仍然能获得较好的分类效果[6]。SVM本质上是二分类算法，如果要用它进行多分类，其本质上也只是在向量空间中进行多次二分类而已。决策树是一种简单并且曾被广泛使用的分类算法，它是一个预测模型，决策树代表的是对象属性和对象值之间的一种映射关系。树中每个非叶节点分支代表对某个属性的预测，每个叶节点所代表的是样本的一个类别。决策树的预测效率很高，因为决策树只需要构建一次，便可以一直使用，决策树预测时的计算次数和其深度有关[7]。决策树的输出结果可以通过图的形式让人易于理解，对于中间值的缺失也不敏感，但是它容易产生过拟合问题，并且构建很耗时。同时，决策树不适合处理高维数据，对于高维数据会增加其树的高度，使得树结构变得复杂不易于理解。

1.2 论文工作

本文介绍分析了当前流行的三个机器学习分类算法KNN,SVM和决策树以及它们各自的优缺点，通过实际的数据验证三种分类算法的分类效率。算法通过python的机器学习包sklearn来实现，共实现了KNN、SVM以及决策树三种分类算法。运行系统为win8系统，编程语言为python，IDE为pycharm，实验数据来自UCI上的开源机器学习数据。本文的工作主要有以下几点：

(1) .分析当前流行的多种机器学习分类算法的实现原理，并且讨论它们的优缺点，总结目前人们对于KNN,SVM和决策树这三种分类算法的认识以及对其进行的改进。本文将会给出算法的程序框图，伪代码以及算法实现流程等，让读者易于理解其实现原理。本文要重点分析的是KNN分类算法，主要通过控制变量法和SVM，决策树的性能进行比较来分析KNN算法性能。

(2) .在pycharm平台上利用python的机器学习包sklearn来实现KNN、SVM和决策树分类算法。首先需要下载python3.5版本，然后下载IDE pycharm，之后下载python的机器学习类库，如numpy，scipy，scikit-learn等。一切准备就绪就可以用python的机器学习包sklearn来构造分类器了。本文共构造了三个分类器分别为KNN分类器，决策树分类器和SVM分类器。

(3) .从UCI上下载机器学习数据，通过比较运行在三个分类器上的运行时间和分类准确率来衡量算法的性能。为了进行算法性能分析，分别下载了高维数据，大数据量数据，低维数据和小数据量数据。由于不同分类算法对于不同类型的数据的分类效果不一样，这些代表性的数据会使得算法的运行时间和准确率等明显不一样，从而我们可以更容易清楚地比较出各种算法的特点，从而达到分析算法的目的。

1.3 论文组织

本论文的章节组织结构如下：

(1) 第一章是论文的绪论部分，该章节讲述了本文研究的背景知识，研究思路，实验采用的技术和手段，论文主体工作以及论文的组织结构。

(2) 第二章是分类算法的理论知识介绍部分。在这里我将会先介绍机器学习分类算法的前景提要。然后介绍三种主要的机器学习分类算法：KNN分类算法,SVM分类算法和决策树分类算法。最后分别对上述三种分类算法进行实现原理分析。在介绍完上述算法之后我们便可以开始算法的实现部分了。

(3) 第三章是本文章的主体部分，也就是算法的实现部分。本章将会先介绍实验所采用的编程环境和实验需要准备的类库。通过编程实现了三个分类器：KNN分类器，SVM分类器和决策树分类器。然后从UCI数据库上下载一些具有代表性的数据集。将下载的数据集分别带入生成的三个分类器里，然后比较各个分类器的运行时间和预测结果准确率，进而比较出各个算法的性能。

(4) 第四章是本论文的总结部分，该章总结了论文的总体工作，分析了目前的多种机器学习分类算法各自的优缺点，并对未来机器学习分类算法的发展趋势提出了展望。

指　标
疑似剽窃文字表述

1. 它的的基本思想是利用数据分布上的模型假设，建立学习预测模型对未标签的样本进行标签；
2. 分类算法，它是一个预测模型，决策树代表的是对象属性和对象值之间的一种映射关系。树中每个非叶节点分支代表对某个

4. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第4部分

总字数：9244

相似文献列表　文字复制比：2.6%(236)　疑似剽窃观点：(0)

1	风电场输出功率的短期预测研究	0.7%（69）
	陈前程(导师：王晓兰) -《兰州理工大学博士论文》- 2012-05-01	是否引证：否
2	基于KNN的地基可见光云图分类方法	0.7%（68）
	朱彪;杨俊;吕伟涛;陈丽英;马颖;姚雯;张义军; -《应用气象学报》- 2012-12-15	是否引证：否
3	201405_201021430090_曾星	0.7%（66）
	曾星 -《大学生论文联合比对库》- 2014-05-20	是否引证：否
4	基于数据挖掘的能耗监管模型在校园节能监管平台中的应用研究	0.3%（30）
	胡良浩(导师：刘慧婷) -《安徽大学博士论文》- 2016-04-01	是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第2章

KNN分类算法以及常见分类算法的介绍

2.1 数据挖掘和分类算法

数据挖掘（Data Mining）技术[8]指的是从海量的数据中得到有用信息和价值的过程，它是数据库技术发展的必然趋势。数据挖掘是一门多学科交叉的计算机学科分支，数据挖掘涵盖了人工智能，机器学习，统计学和数据处理技术等多门学科。数据挖掘的总体目标是从一个数据集中提取出有用的信息，并将其转化成易于理解的模型，以用于进一步的数据分析[9]。而数据分析是属于数据挖掘的一个分支。数据分析是在已定的假设上利用计算方法和数理统计方法将数据分析转化为信息的技术，而如果要对这些信息进行进一步的处理，从而转化为具有预测和决策功能的模型，则需要数据挖掘技术，也就是说，数据挖掘技术是数据分析技术的更深入一步，当然分类算法也是属于数据挖掘算法的一部分。

分类（Classification）算法是数据挖掘算法的关键技术，分类算法首先构造分类器，通过分类器对训练数据集进行模型训练，从而发现数据集的分类规则，最后根据训练后的分类器模型对待测试数据进行预测。分类算法的流程主要包括以下两步：①构建模型阶段：根据已经获得的原始数据集和原始标签构建一个合适的分类器，并用该分类器对原始数据集进行训练，最终生成一个可以在一定误差范围内预测该特定类型数据的分类器。②使用模型阶段：使用训练后的分类器模型对上述数据类型的未知对象进行预测分类。目前分类算法已经被广泛应用到生物学，金融学，统计数学等多个领域。

由于分类算法种类繁多，不同的分类算法的性能和适用的情况不同，这使得研究者们对于分类算法的选择往往很困惑。以下是三种常见的分类算法的特点：KNN算法简单稳定而有效，对于中小型的低维的数据很适合，但是由于KNN分类算法是有数据预测时才训练数据的，所以对于高维的大数据分类效率就很低；SVM分类算法对于二分类的效率很好，但是对于多分类的话就会话费很多时间；决策树分类算法简单易懂，对于低维数据的分类效果很可观，但是对于高维数据，要生成的决策树高度就会变高，从而导致分类效率变低，并且容易过拟合。由此可见，选择合适的分类算法将会对数据集的分类效果起到很重要的促进作用。本文接下来将会介绍KNN以及几种常见的分类算法并对其性能进行研究。

2.2 KNN分类算法

2.2.1 KNN算法介绍

KNN算法全称为K Nearest Neighbors，即K邻近算法，就是找到距离测试样本最近的K个实例中所占比例最多的类别标签。KNN算法是一种可以用于数据分类和回归分析的算法。KNN算法最早是由Cover和Hart在1968年发明的，其思路十分简单并且直观，其基本思想大致如下：先根据距离函数算出待测试样本和训练数据集中每个数据点的距离，从中选出与待测试样本最近的K个样本，最后选取这K个样本中所占比例最多的类别标签作为该待测试样本的类别。KNN算法算得上是一个理论上比较成熟的分类算法，并且由于其实现起来较简单以及其准确率较高等优点使得目前还有很多学术界的学者和业界人士仍然在研究和使用该算法。

KNN算法是一个“懒惰”的算法，它和决策树，贝叶斯等分类算法不同，它是一种基于实例的学习算法。KNN分类算法不需要提前训练数据，当有待测试数据出现时，直接从训练数据集中找出距离待测试数据最近的K个实例，把这K个实例中占比最多的类别标签作为待测试数据的类别标签。因此，当训练集数据和待测试数据的数据量很大的时候将会导致计算量急剧增加，使得算法效率低下。同时选取合适的K值，距离计算函数和距离权重也是很重要的。不同的数据集的最佳K值不同，需要研究人员手动测试。距离计算函数的选取也会影响到算法的准确率，需要研究人员综合数据集的特点来决定选取哪种距离计算函数。

由上述分析可见，要最大化地发挥KNN分类算法的性能也不是那么简单的一件事。我们需要通过反复试验来选取最佳的

K值，同时根据数据集的特征和分布情况来决定最佳的距离计算函数。文章接下来将会分析KNN算法的具体实现原理以及如何选取最佳的K值，距离权重和距离计算函数以及算法的一些其他参数以提高算法的分类性能。

2.2.2 KNN算法实现原理

1.基本原理：

KNN算法是一个相对比较容易理解的算法，KNN算法的基本原理大致如下：输入训练样本和待测试数据，先根据距离函数计算出待分类样本和训练数据集中每个数据点的距离，从中选出与待测试样本最近的K个样本，最后选取这K个样本中所占比例最多的类别标签作为该待测试样本的类别标签。你可以简单的理解成由那些距离自己最近的K个点投票来决定待分类的数据属于哪一个类别，即“物以类聚，人以群分”！

KNN分类算法所选取的邻居对象都是已经确定类别的对象，该算法在觉定测试样本所属类别的时候只依据和测试样本最近的K个训练样本的类别来决定待分类样本所属类别，因此K的选取是很重要的。如图2-1 所示：

图2-1 KNN算法示例图

从上图我们可以发现，图中训练数据集有两种类型，一类是蓝色正方形，另一类是红色三角形，而处于图片中心的绿色圆形数据正式我们要预测的待分类样本。

如果K=3的话，那么待分类样本的5个最近样本中有2个红色三角形和1个个蓝色正方形，按照KNN分类算法让这3个点投票，则该待分类样本应该属于红色三角形。

如果K=5的话，那么待分类样本的3个最近样本中有2个红色三角形和3个蓝色正方形，按照KNN分类算法让这5个点投票，则该待分类样本应该属于红色三角形。

由此可见，KNN分类算法的分类效率很大一部分是取决于K值得选择。

2. 算法流程：

KNN分类算法是一个惰性算法，它和决策树，贝叶斯等分类算法不同，它是一种基于实例的学习算法。所以KNN分类算法不需要提前训练数据，当有待测试数据输入时再立马进行测试。KNN算法的大致步骤如下：收集数据，准备数据，分析数据，测试算法，使用算法。KNN分类算法的大致流程图如下所示：

开始

收集数据

准备数据（距离计算所需的值，最好结构化成向量形式）

分析数据

测试算法（计算错误率）

使用算法

输出

图2-2 KNN算法流程图

由上图2-2可知，KNN分类算法主要有以下五个步骤：

①收集数据：原始数据的收集可以是任何途径，可以在网上下载或者通过试验等方式获得。

②准备数据：该步骤主要为后面的距离计算做准备，由于距离计算只能计算数字类型数据，所以需要对原始数据进行处理转化成数字类型数据，并且需要将原始数据转化成向量形式的数据，这样易于后面的距离计算。

③分析数据：这里可以用任何方法分析数据，例如数据是二维的话，那么我们可以用python的Matplotlib包里的函数画二维散点图。

④测试算法：测试算法的目是计算算法的错误率，即算法的可行性，要是算法的错误率太高的话那么显然算法就是不可行的。首先需要将训练集的一部分当作测试样本，测试样本和非测试样本的区别在于：测试样本是已经分类完的数据，如果测试样本的预测分类结果和实际的类别不一样，则标记为一个错误。在获得测试样本数据之后我们还要确定K值和距离计算函数。对于不同的数据集的K值和距离计算函数的选取会有所不同。我们需要不断地取不同的K值和距离计算函数以及距离权重来测试测试数据的错误率，直到找到最优或者接近最优的K值和距离计算函数以及距离权重。测试算法步骤是KNN分类算法的核心步骤，我们需要不断地测试以达到良好的分类效果。K值，距离计算函数和距离权重的选取我们将会在后面提到。当选择完K值，距离计算函数和距离权重后KNN分类器就构造完成了，现在就可以开始使用算法了。

⑤使用算法：首先输入未知属性的待测试样本，KNN分类器会计算出训练数据集中每个数据点和当前测试样本的距离，然后从中选取K个与待测试样本距离最近的点来决定测试样本的类别标签。算法的使用步骤如下图所示：

输入未知待测试点

计算训练样本集中的点与当前点之间的距离

按照距离递增排序

选取与当前点距离最小的K个点

确定K个点所属类别出现的概率

返回K个点中类别出现频率最高的类别

图2-3 KNN算法的使用步骤

由图2-3知在使用KNN算法时，当我们输入未知待测试样本后，KNN分类器会首先计算出训练样本集中每个点和当前点的

距离，然后按照距离的大小递增排序，选取出距离最小的前K个点，再计算出这K个点中每个类别出现的概率，最后返回K个点出现次数最多的类别作为待预测样本的类别标签。

2.2.3 KNN算法的性能分析

决定KNN算法的分类性能的因素主要有数据集本身的分布特点，K值的选取，距离计算函数的选取以及距离权重的选取。其中数据集本身的分布特点是我们无法认为改变，因此我们想要提高算法的分类性能，我们就要从后面三者下手。

1. K值选取：

K值的选取可以说的上是最为关键的，不同的K值将会对算法产生重大影响。如果K值选的太小，则相当于只在待测试样本较近的范围内进行预测，这样会使预测结果对近邻样本点非常敏感，当测试样本点附近有噪声点的时候，那么预测结果就很容易出错。换而言之，K值太小会导致分类模型过于复杂，容易产生过拟合。相反的，要是K值太大的话，那么预测结果将会取决于与待测试样本距离较大的范围内的数据。这样会导致较远的数据本身与待测试数据无关，但是仍然被牵连进来，显然这样做会降低算法的分类准确率。K值得增加会使得分类模型变得简单，在极端情况下K=N，那么无论输入什么都将输出训练样本集中出现次数最多的类别。

K值的选取不能过大也不能过小，它决于训练样本集的规模，对于较小的训练集K值取3~10。一种常见的取值方法是取K值为训练样本集的样本数的开平方，还有一种做法是进行交叉检验，从训练样本集中选取出多个测试样本集，用多个K值来对测试样本集进行预测，根据算法预测的准确率来选择最佳的K值。

2. 距离计算函数的选取：

KNN算法中常用的距离度量计算方法有如下：

闵可夫斯基距离（Minkowski Distaice）度量：

dist(X , Y) = (n为样本的维度，p=1，2，3...)

闵可夫斯基距离度量是对多个距离度量公式的概括：

当p=1的时候，就变成曼哈顿距离（Manhattan Distance）：

dist(X , Y) = (n为样本的维度)

曼哈顿距离来自城市的区块间距离，它是在标准坐标系上的两个点的各个坐标系的值相减绝对值的总和[10]。

当p=2的时侯，就变成欧式距离（Euclidean Distance）：

dist(X , Y) = (n为样本的维度)

欧式距离是最容易理解的距离度量，也是最常用的，它的物理意义就是空间中的两点之间的直线距离[11]。

当p=∞的时侯，就变成切夫雪比距离（Chebyshev Dictance）：

dist(X , Y) = (n为样本的维度)

切夫雪比距离是多维向量空间中的一种距离度量方法，两点间的距离是其各坐标轴上数值差的最大值。即：

dist(X , Y) = （i=1，2，3 . . . n，n为向量维数）

余弦距离（Cosine Distance）：

在二维空间里，两向量A(x1，x2)和B(y1，y2)的夹角的余弦公式为：

Cosθ =

在n为空间中A(x1，x2，...xn) 与B(y1，y2，..,yn)的夹角余弦为：

Cosθ =

夹角余弦的取值范围是[-1，1]，值越大表示两个向量之间的夹角就越小，两向量就越相似。

马氏距离（Mahalanobis Distance）：

马氏距离可以用来衡量一个样本点X与数据集的分布距离，也可以衡量两个来自同一分布的样本X和Y的相似性，设X=(x1，x2，x3，. . .xn)T，

Y=(y1，y2，y3, . .yn)T，数据集分布的均值为：μ=(μ1，μ2，μ3，. . .μn)，协方差矩阵为S，则样本点X与数据集的马氏距离为：

DM(X) =

样本X与样本Y的马氏距离为：

D(X，Y) =

马氏距离最早由印度的统计学家哈马拉斯诺提出，表示两数据的协方差距离，它考虑了数据的个指标之间相关性的干扰，它的缺点是对变化微小的变量很敏感。

在实际运用中，常用的是欧式距离，它可以较好的度量空间中两点间的距离，本文实验选用的距离度量方式也是欧式距离。

3. 距离权重的选取：

在KNN算法中，如果训练样本集的分布很不均匀的话，那么算法的分类效率将会降低，图2-4展示了当样本数据分布呈区域性集中时的情况：

图2-4 当K=17时的KNN分分类算法示意图

从图2-4我们可以看到，待预测点为绿色三角形的点，当K=17时，蓝色棱形有10个，红色圆形有7个，按照传统的KNN算

法，带预测点应该预测为和蓝色棱形一类。但是如果我们仔细观察的话会发现，在带预测点附近的点基本都是红色的点，而且相当密集，从直观角度上来看，待预测点应该归为红色圆形这一类，所以这个预测模型是失败的。

为了改进算法，我们应该给距离赋上权值，使得距离预测点近的点有更大的权重。在KNN算法中，常用的距离加权函数有两个：

（1）反函数法：

反函数是比较简单的加权函数，设距离为d，则返回权重为1/d。当两个样本完全一样或者非常相近时，返回的权重就会很大甚至会无穷大，这显然是不可以的，基于这个问题，在原来的加权函数里加入一个常量a：

Weight = 1/（dictance + a）

反函数法的优点是为近邻邻居分配了很大的权重，稍微远的会衰减的很快，这种情况虽然是我们想要的，但它有时候也会使算法对噪声因素更加敏感。

（2）高斯函数法：

高斯函数相对于反函数来说，其表达式比较复杂，但是其加权效果比反函数好。高斯函数的表达式如下：

f(x) = (a , b , c ∈ R)

图2-5 高斯函数图像

图2-5是高斯函数的函数图像，和正态分布的图像有点相似，函数参数中，a是曲线的高度，b是曲线中心线在x轴的坐标，c是半峰高度（FWHM）。

图2-6 当a=5,b=0,c=10时的高斯函数图像

图2-6是当a=5,b=0,c=10时的高斯函数图像，由图可知，当距离为0时，权重为1，随着距离的增加，权重的减小幅度很平缓，这克服了反函数当距离远一些的时候急剧下降的缺点，高斯函数的抗噪声性比反函数强。

在加权KNN算法的实际应用中，首先我们要获得排序后的样本距离值，然后取钱K个样本距离，在处理离散型数据时，我们需要将这K个数据用权重来衡量相似度，预测结果与第i个数据的标签相同的概率为：Pi = 。在处理连续型数据时，我们需要对这K个数据加权平均：通过对每个样本的距离值乘于样本对应得权重，然后再累加，再求出所有权重之和，两者相除既得最终结果。算法公式为：f(x) = 。其中Di代表邻近点i与待预测样本的距离，Wi代表邻近点i的权重，f(x)是预测结果。

2.3 SVM分类算法

2.3.1 SVM算法介绍

SVM（支持向量机）算法是由Vapnik和其小组于1995年在贝尔实验室提出的[12]，SVM算法是一种基于统计学理论的算法，它是一种监督式学习算法，广泛应用与统计分类和回归分析，目前SVM分类算法广泛应用于文本和超文本分类，图像分类，手写字体识别以及医学中的蛋白质分类等。SVM分类器的特点是它能够在控制误差最小的情况下使超平面的两边区域最大化。

SVM分类算法的主要思想是在高维空间中找到一个可以将训练集中的两个类别的样本之间的距离最大化的超平面。SVM算法最初是用于解决二分类问题而提出来的，当它处理多分类问题时，其实就是将多分类问题转化为多个二分类问题，然后依次用多个SVM分类器解决问题。

SVM分类算法是目前机器学习最流行的分类算法之一，它在高维空间中非常有效[13]，即使在样本维度比样本集数量大的情况下依然有效，SVM在决策函数中使用训练集的子集，因此它对内存的利用是很高效的。SVM中不同的和函数有相对应的决策函数，我们也可以自定义核函数。但是SVM分类算法也是有缺点的，如果样本的维度比样本数要大得多，那么就很容易发生过拟合，SVM分类算法不提供直接的概率估计，若要概率估计则需要进行五次交叉验算来获得。

2.3.2 SVM算法实现原理

SVM是一个二分类模型，它的主体思想是在高维空间中找出一个可以将样本集中的两类数据分割开的超平面。首先需要将数据集从低维空间映射到高维空间，这就需要核函数的来实现，之后在这个高维空间中找到一个可以将两类样本分割的超平面，使得它们之间的样本距离最大化。如果一个样本集是多维的，那么分类器将会随机产生一个超平面，然后不停地移动这个超平面，直到该超平面可以分割训练样本集中的两类样本点为止。可能选取到的超平面可能有很多个，我们的目的是找到一个能使得超平面两侧空白区域最大化的超平面，从而使训练样本集的分类效果达到最好。

图2-7 SVM分类算法最优超平面选取

图2-7展示了SVM分类算法选取最优超平面的过程，图中能将样本集中的两类数据分开的超平面有很多，但是我们能找到一个超平面能达到最大化边缘，这个超平面即最优超平面。

核函数：核函数是SVM分类算法的一个很重要的辅助函数，它实现了将样本集数据从低维空间转化到高维空间的映射过程，而且这过程可以把低维空间中的线性不可分的数据集转化成高维空间中的线性可分的数据集[14]。

常见的核函数一般有以下几类：

①线性函数：

K(x1 , x2) = x1·x2

②多项式核函数：

K(x1 , x2) = (γ< v1 , v2 > + c)n

③高斯径向基核函数：

K(x1 , x2) = exp (-γ|| x1 - x2 ||2)

④Sigmoid核函数：

K(x1 , x2) = tanh (γ< x1 , x2 > + c)

其中γ，c，n为常数。

在上述众多核函数中，我们最常用的是高斯径向及函数，应为它具有较好的学习能力，它具有很强的适应性，无论低维，高维，小样本还是大样本等都适用，且具有较大的收敛域，是性能比较好的核函数[15]。

SVM分类算法的实现原理如下[16]：

设待分类问题为二分类问题，训练样本集为（x1，y1），（x2，y2），...，（xn，yn），其中xi∈Rn代表样本的特征向量，yi∈{+1，-1}为类别标签。SVM算法就是使用核函数ϕ(x)将训练集的数据x从低维的空间映射到高维的空间，然后在高维空间找到一个最优化超平面：

w·ϕ(x) + b = 0

公式中，w∈Rm，x∈Rm，b为偏置量。下面我们需要构造一个判别函数，使得两类样本能够被正确分割，并且使它们之间的分割间隔距离最大，判别函数如下：

y(x) = sign [w·ϕ(x) + b]

通常情况下，最优超平面问题可以被描述为

约束条件为：

yi (w·ϕ(x))+ b) ≥ 1-i

i ≥0 (i=1 ,2 ,3 ,...,l)

其中，i 为松弛项，C为惩罚参数。要想求解上述二次规划问题，我们可以通过采用拉格朗日乘子式求解它的对偶形式来求解该问题。即求解下述问题：

约束条件为：

0 ≤αi ≤ C，

上式中，K(xi，xj)=ϕ(xi)·ϕ(xj) 称为核函数，判别函数为：

y(x) = sign[w*·ϕ(x) + b*] =sign []

其中，w*，b*，ai*表示最优解。

SVM分类算法不仅可以处理线性规划问题，也可以处理非线性规划问题。对于非线性规划问题，我们可以通过非线性变换将它转化成高维空间的线性规划问题[17]，然后在高维特征空间中使用支持向量机，这里的非线性转化即为核函数。

2.3.3 SVM算法性能分析

支持向量机最先是发明用于计算二分类问题的，但是现实世界中大多数分类问题都是多分类的，因此仅仅二分类的支持向量机是不够用的。为此，需要对二分类向量机进行了改进使之能能适应于多分类问题。一种很好的改进方案就是将多分类问题分解为多个二分类的问题，然后通过生成多个SVM分类器对其进行分类。常用的多分类方案有[18]：

①one-against-rest（一对多）：

该方案将训练样本集的某一类样本当作二分类的其中一类，将其他的种类当成另一类来进行分类。对于n个类别的样本数据集，则需要训练n个SVM分类器，同时可以得到n个最优超平面分类规则的指示函数结果。对于待分类样本，用这n个分类器进行预测，取指示函数结果最大的类别作为最终的类别。该方法实现原理比较简单所需的分类器也比较少，因此分类速度也比较快，但它也有一定的缺点：由于构造决策平面时所有的样本集均需要参加训练，因此它在训练数据时所花的时间比较多。同时由于训练时总是将某一类看作一类，其他的类看多另一类，所以训练时两类的数目会相差很大，这将会对预测的结果精度产生很大影响。

②one-against-one（一对一）：

该方案将训练样本集中的每两类都当作一个二分类问题，这样的话，如果样本集有n个类，那么它将会构造n(n-1)/2个分类器。当需要预测数据时，用这 n(n-1)/2个分类器来对待预测样本进行分类，根据这n (n-1) /2个分类器的分类结果中投票数最多的类作为待测试样本的所属类别。这种方法的的思路简单，也容易实现，但其生成的分类器数目是O(n2)的，当训练样本集的类别数目较多时，将会产生很多的分类器，这会大大影响样本集训练的速度。同时，在测试时要对待测试数据进行n(n-1) /2次分类预测，这会导致样本预测的时间很长。另外，要是在投票时出现两个类别得票数目相同的情况，这时的类别判定就会很麻烦。

③BT - SVM（二叉树支持向量机）：

BT - SVM对于k个类的训练样本集，生成k - 1个分类器。第一个分类器将第一个类别作为一类，将其他的2，3，....，n类作为另一类来训练第一个分类器SVM1，第二个分类器将第二个类别作为一类，将其他的3，4，....，n类作为另一类来训练第二个分类器SVM2，同理，第i个分类器将第i个类别作为一类，将其他的i+1，i+2，....，n类作为另一类来训练第i个分类器SVMi，直到第k - 1个分类器将第k-1个类别作为一类，将第k个类作为另一类来训练第k-1个分类器SVMk-1。

指　标
疑似剽窃文字表述

1. 测试样本，测试样本和非测试样本的区别在于：测试样本是已经分类完的数据，如果测试样本的预测分类结果和实际的类别不一样，则标记为一个错误。

5. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第5部分

总字数：2988

相似文献列表　文字复制比：11.3%(338)　疑似剽窃观点：(0)

1	基于决策树的分类算法研究 胡江洪(导师：熊盛武) -《武汉理工大学硕士论文》- 2006-04-01	3.0%（89） 是否引证：否
2	2011301500212_黄蕴熙_基于决策树的分类算法在试卷分析系统中的应用 黄蕴熙 -《大学生论文联合比对库》- 2016-05-21	2.1%（64） 是否引证：否
3	入侵检测模糊分类算法研究 张俊丰(导师：彭新光) -《太原理工大学硕士论文》- 2007-05-01	2.0%（61） 是否引证：否
4	自04_2010011443_庞海天 庞海天 -《大学生论文联合比对库》- 2014-06-09	1.6%（49） 是否引证：否
5	数据挖掘在农信社客户关系管理中的应用研究 龙亚平(导师：陆国庆;陈勇) -《湖南大学博士论文》- 2012-10-01	1.4%（41） 是否引证：否
6	基于数据挖掘技术的警务智能信息系统的构建与应用 祝捷(导师：肖会敏) -《河南财经政法大学博士论文》- 2010-12-01	1.3%（39） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

算法生成的二叉树结构类似于一棵哈夫曼树。算法只用构造k-1个分类器，测试时并不一定要进行k-1次分类器预测，从而降低了训练和预测所需的时间。但是它也有一定的缺点，如果前面的分类器判断错误，那么这个错误也会扩散到后面的分类器上，从而影响到分类器的性能。因此，我们需要构造合理的二叉树结构以提高分类性能，这就需要我们选择合适的二叉树生成算法。

④DAG - SVM（有向无环图支持向量机）：

该方案需要构造n(n - 1) /2个分类器，和“one-against-one”方案一样。在预测阶段，首先要构造一个有向无环图，该图有n(n-1) /2个非叶节点和n个叶节点，每个非叶节点代表一个SVM分类器，每个叶节点代表一个类别标签。DAG - SVM的树结构示意图如下图所示[19]：

图2-8 DAG - SVM树结构示意图

由图2 - 8可知，一共有9个类别，生成了36个分类器，当对样本进行预测时时，先用根节点分类器对其进行预测，根据预测的结果选择下一层的分类器，以此类推，直到最后到达叶节点，就得到了预测的类别。该方案的优点是预测速度快，只要n-1次预测就能得到结果，这比一对一和一对多都要快。和一对一一样，它需要构造的分类器数目很多，这使得训练阶段所花费的时间很多。同时，该算法根节点的选择很重要，根节点的选择将会对后面的选择有很大影响。

2.4决策树分类算法

2.4.1决策树算法介绍

决策树（Decision Tree）算法是一种典型的数据分类算法，它产生于20世纪60年代。到70年代，科学家昆兰提出了基熵的ID3算法[20]，此算法有效的减少了生成的决策树的深度，但是它没有考虑到叶子数目的研究。之后，有人在ID3的基础上进行了进一步的改进，提出了C4.5算法。C4.5算法在剪枝技术，预测变量的缺值处理，派生规则等方面进行了许多改进，使之不仅适用于分类算法，也适用于回归问题。

决策树算法通过对训练样本集进行训练生成决策树，然后利用决策树对数据进行分析。决策树对样本集训练的过程实质上就是利用归纳算法发现数据集中蕴含的分类规则的从而生成决策树过程，决策树算法的关键在于构造一颗规模小，精度高的决策树。构建决策树的过程可以分两个步骤：第一步是通过归纳算法对训练数据集进行训练从而生成决策树的过程；第二步是对上一步生成的决策树中多余的节点进行剪枝的过程。剪枝主要是对上一步生成的决策树的节点进行校验和修正，以消除噪声和过拟合问题。

2.4.2决策树算法实现原理

决策树是一种简单并且曾被广泛使用的分类算法，它是一个预测模型，决策树实质上是对象属性和对象值之间的一种映射关系，通过对象属性去决定要走的决策树分支，最终找到叶节点，即预测的样本类别。决策树中每个非叶节点分支代表对某个属性特征的预测，每个叶节点代表样本的一个类别标签。决策树算法的效率很高，因为决策树只需构建一次，便可以反复只用[21]，每次预测的最大计算次数不超过决策树的深度。决策树算法是基于信息熵的算法，要了解决策树，首先要知道以下相关概念。

熵：熵（entropy）是衡量一个系统中物质的混乱程度的，通常用于化学中衡量物质的浓度，在其他领域中也有使用，是很重要的概念。

信息熵（香农熵）：信息熵是用来衡量信息混乱程度的，信息和信息熵成反比，信息越有序，则信息熵越低，反之信息熵越高。例如：书本有序的排列在图书馆里里，熵值很低，相反的话，要是书本无序地乱放在地上，则熵值就很高。

信息增益：信息增益为数据集划分变化前后的信息熵的差值。

决策树算法伪代码如下：

CreateDT(){

检测样本数据集中所有的类别标签是否都相同：

If(相同)：return 类别标签；

Else：

寻找最适合分裂的属性特征（信息增益最大的特征）；

划分数据集；

创建分支节点；

For（每一个划分的分支子集）：

调用函数CreateDT（ ）；

Return 分支节点；

}

目前流行的决策树算法有两种：ID3算法和C4.5算法。下面我们来介绍一下ID3算法和C4.5算法的原理：

ID3算法：

ID3算法是根据熵的变化进行决策树的构造的，熵的变化可以看成是信息增益，熵越小，则信息增益越大。ID3算法以信息增益作为分支属性选择，选分之后信息增益最大的属性进行分支。设D为用类别标签对训练集进行的划分，则D的熵为：

info(D) = -

其中pi表示第i个类别标签占整个训练集数据类别标签的比例。熵的实际意义是D中元组的类别标签所需的平均信息量。如果将训练集D按照属性值A进行划分，那么A对于D的期望信息为：

infoA(D) =

则属性值A的信息增益为上述两者之差：

gain(A) = info(D) - infoA(D)

ID3算法的原理就是用到上述的信息增益，在每次要分裂的时候会计算每个属性的信息增益，选择信息增益最大的属性进行分裂[22]，每次分裂都会使树生长，最终生成一颗完整的决策树。

ID3算法为决策树分类算法提出了新思路，但它还是有很多缺点：ID3算法对于缺失值没有考虑进去，并且容易出现过拟合问题，由于取值比较多的特征在计算信息增益时比取值较少的特征大，所以在进行分裂时ID3偏向于多值属性，然而他们都是完全不确定的变量，取值多的特征未必比取值少的特征好。基于ID3的确定，提出了C4.5算法[23]。

C4.5算法：

ID3容易选择一些取值较多的属性来分裂，尤其是像ID这种每个取值都不一样的属性，这样虽然会使得数据集划分得很纯净，但是这样的划分对分类效果来说却没什么意义。C4.5对于这点做出了改进，C4.5算法定义了分裂信息，表示为：

split_infoA(D) = -

其中，pi =，可以当作属性分裂的熵，种类越多就越混乱，熵就越大。在这里定义信息增益率为：

gain_ratio(A) =

C4.5选择信息增益率最大的属性来进行分裂。其具体应用和ID3类似，这里就不多说了。

2.4.3 决策树算法性能分析

决策树算法的算法复杂度很低，算法本身也也很好理解，并且中间值的缺失对它没什么影响，并且可以处理没有关联的特征的数据，但是其可能产生过拟合问题。

决策树的创建过程是一个递归过程，需要确定一个递归出口。如果把递归出口定义为训练集只有一种类型时退出，那么会使树产生过多的节点，从而导致过拟合。另一种方法是在当前节点中的训练集数目低于一定阈值时返回，将max(Pi)作为当前节点的分类。

上述两种方法生成的决策树都会产生过拟合问题，其原因是树节点过多会产生很多离群点和噪声点，一种减少过拟合的方案就是剪枝。剪枝对于决策树的正确率影响很大，主要有两种剪枝方法：

①前置剪枝：前置剪枝就是在构造决策树的时候提前停止，可以将切分节点的条件设置得苛刻一些，但是这样会导致决策树变得很小，使得决策树无法达到较好的分类效果。

②后置剪枝：后置剪枝是在构建好决策树之后再进行剪枝处理。主要有两种方法：1)用一个叶节点替换整个子树，叶节点的分类标签采用子树种类别比重最多的类别。2)用一个子树替代另一个子树。后置剪枝的一个缺点就是计算效率低，有些节点计算后就被剪枝了，导致白白计算。

指　标
疑似剽窃文字表述

1. 算法。C4.5算法在剪枝技术，预测变量的缺值处理，派生规则等方面进行了许多改进，使之不仅适用于分类算法，也适用于回归问题。
2. 信息增益越大。ID3算法以信息增益作为分支属性选择，选分之后信息增益最大的属性进行分支。

6. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第6部分

总字数：7481

相似文献列表　文字复制比：7.9%(589)　疑似剽窃观点：(0)

1	scikit-learn决策树学习 - 每天进步一点点2017 - CSDN博客 - 《网络（http://blog.csdn.net）》- 2017	4.1%（310） 是否引证：否
2	决策树DTC数据分析及鸢尾数据集分析_图文 - 《互联网文档资源（http://wenku.baidu.c）》- 2016	3.7%（278） 是否引证：否
3	Python机器学习算法库——决策树（scikit-learn学习 - 决策树） - Yeoman92的博客 - CSDN博客 - 《网络（http://blog.csdn.net）》- 2017	3.3%（249） 是否引证：否
4	Google机器学习二 鸢尾花数据集 load_iris - tz - 《网络（http://blog.csdn.net）》- 2017	3.1%（235） 是否引证：否
5	机器学习2-决策树的可视化 - 博客频道 - CSDN.NET - 《网络（http://blog.csdn.net）》- 2017	2.7%（200） 是否引证：否
6	数据分类技术的研究及在大数据平台上的运用 陈留锁 - 《大学生论文联合比对库》- 2016-06-16	2.6%（198） 是否引证：否
7	20112478_李春雪_模糊聚类的有效性判定算法实现 李春雪 - 《大学生论文联合比对库》- 2015-07-09	1.0%（76） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第三章实现与测试分析

3.1 实验环境

本实验实在win8系统上实现的，实验机型为dell笔记本电脑，电脑参数配置如下表：

表3-1 实验环境配置

参数配置

处理器 4核 Intel(R) core(TM) i5-5200U CPU @ 2.20GHz

内存 8G RAM

系统类型 64为操作系统，基于x86的处理器

硬盘 450GB

本实验采用了python语言实现编程python版本为python3.5，编码平在为pycharm。pycharm是目前最流行的python IDE，带有一整套可以帮助使用者便捷开发的开发工具，比如代码调试，智能提示，python代码自动对齐等功能。实验为了方便地实现机器学习算法，引入了多个机器学习包，包括numpy，matplotlib，sklearn等，其中，sklearn包为本文的主要包。

sklearn全称为scikit learn，它是一个简单高效的数据挖掘和数据分析工具，建立在Numpy，Scipy，和matplotbib之上，是业界相当出名的一个开源项目。它不仅实用方便，而且功能也很强大，广受研究者们的热爱。是sklearn之所以强大的原因之一是它对很多的机器学习方法进行了封装，使得用户想要用哪种机器学习方法可以直接通过sklearn包调用。sklearn包广泛地支持各种分类，聚类，回归算法，并且对数据降维，模型选择和数据预处理都有着很好的支持。本文中，我们将使用sklearn包的工具直接构造三个分类器。用KNeighborClassifier()构造KNN分类器，用SVC()构造SVM分类器，用DecisionTreeClassifier()构造决策树分类器。

本实验的数据集来自UCI数据库。UCI数据库里的数据很适合用来进行机器学习，它是由加州大学欧文分校(University of CaliforniaIrvine)提出的，目前UCI数据库上有436个数据集，其数据集数目还将会不断的增加。本实验为了比较出各个分类算法的性能，引用了高维，低维，大数据，小数据等多个具有代表性的数据，本文引用了UCI数据库上的iris，wine，madelon，skin_nonskin等数据集[24]。本实验对数据集的分类效果采用了训练时间，预测时间，预测准确率等指标，对每个数据集用不同的分类器进行分类，通过比较他们的指标来比较算法的性能。

3.2算法实现

3.2.1 KNN算法实现

Python sklearn包提供了监督式学习的最近邻分类算法的实现，在sklearn.neighbors里实现了两种最近邻分类算法，KNeighborsClassifier ()分类器实现了基于最近邻居点数的最近邻分类算法，RadiusNeighborsClassifier()实现了基于最近半径的最近邻分类算法。其中常用的是前者，当数据集分布不均匀的时候采用后者效果会比较好，这里我们采用KNeighborsClassifier ()来构造分类器。

在sklearn.neighbors包中，KNeighborsClassifier ()的构造函数如下：

KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=1, **kwargs)

其中参数n_neighbors表示最近邻居数，也就是k值，默认为5；weights表示距离权重，默认为‘uniform’，可选项‘distance’即基于距离的权重，‘callable’表示用户自定义这个参数；algorithm表示用户计算最近邻时使用的方法，可选项有如下：‘auto’表示分类器根据数据集的特点自动选择效果较好的计算方法；‘ball_tree’表示分类器用球树算法计算最近邻；‘kd_tree’表示KD树算法；‘brute’表示暴力计算，当样本数据集数量少的时候用这个参数比较好。leaf_size当‘algorithm’=‘ball_tree’或者‘kd_tree’的时候才有效，表示建树时叶节点所能存放的最大数据点的数目；metric表示距离度量方法，默认参数为‘minkowski’即闵科夫斯基距离；metric_params表示对距离度量方式的附加参数，当metric=‘minkowski’，metric_params=2时就表示用欧式距离计算距离度量，大多数时候我们计算距离度量就是用这个值；n_jobs表示进行邻居搜索时并行作业的数量，默认为1。

KNeighborsClassifier ()有几个比较重要的方法：

KNeighborsClassifier ().fit(X,y)方法表示对训练样本集的训练，其中X为训练样本集的属性值，y表示训练样本集的类标签。

KNeighborsClassifier ().predict(x)方法表示KNN分类器对数据x进行预测的结果，返回和x中数据数量相同的一维数组。

KNeighborsClassifier ().predict_proba(x)表示KNN分类器对数据x的概率预测，返回可能预测类型的概率。

KNeighborsClassifier ().score(x，y)方法表示KNN分类器对数据x的预测打分，表示正确预测到的样本数的比例，其中x表示待预测样本，y表示待预测样本的标签，这个函数是在一直待预测样本标签的情况下使用的，对于分类器的预测性能分析很有用。

下面是一个简单的KNN分类器算法的实现：

```
train_data = [[0], [1], [2], [3]]
Train_target = [1, 1, 2, 2]
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier(n_neighbors=3)
KNN.fit(train_data, train_target)
print(KNN.predict([[1.1]]))
print(KNN.predict_proba([[0.8]]))
```

算法输出为：

[1]表示算法对数据[1.1]的预测结果，[[0.66666667 0.33333333]]表示算法对数据[0.9]的概率预测结果。至此KNN分类算法已经可以实现。

3.2.2 SVM算法实现

Python的sklearn.svm提供中了SVM算法的实现，共提供了SVC()，NuSVC()和LinearSVC()三种方法。其中SVC()和NuSVC()两种方法是相似的方法，他们对于多远分类都实现了“one-against-one”也就是“一对一”策略，他们的区别是能接受一些不同的参数设置并且有不同的数学方程。LinearSVC()方法是基于线性核函数的支持向量机，他对于多分类实现了“one-vs-the-rest”也就是“一对多”策略，它不接收参数kernel，也就是不接收别的核函数，因为它是线性的。他也缺少一些SVC()和NuSVC()所拥有的一些成员，例如support_。在这里我们采用的是SVC()方法构造SVM分类器。

在sklearn.svm包中，SVM分类器SVC()的构造函数如下：

SVC(C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None)

其中参数C为错误项惩罚参数；参数kernel为算法使用的核函数，默认为“rbf”即高斯径向基核函数，可选参数有“linear”即线性核函数，“poly”即多项式核函数，“sigmoid”即sigmoid核函数；参数degreee只有当kernel=“poly”时才适用，表示多项式饿次数；参数gamma是“rbf”，“poly”和“sigmoid”的核系数，默认情况下是“auto”，那么它的值将是1/n_features；参数coef0表示核函数中的独立项，它只在“poly”和“sigmoid”中才适用；参数shrinking为布尔值，表示是否要使用缩小的启发式，默认为“True”；参数probability表示是否启用概率估计，这必须在启用fit()函数之前启用，默认“false”，启用的话将会使fit()的训练时间变长；参数tol表示对停止准则；参数cache_size表示内核缓存的大小，单位MB；参数class_weight表示类的权重，值为一个词典，他会将第i类的参数C设置为SVC的class_weight[i]*C，如果没有给出，所有的类都应该有权重。当参数值为“balance”时使用y值自动调整权重，与输入数据中的类频率成反比(n_samples / np.bincount(y))；参数verbose表示是否启动详细输出，默认为“false”即关闭，这个设置利用了libsvm的每个进程运行时设置，如果启用，将会对多线程造成很大负担，可能使多线程无法正常工作；参数max_iter表示求解时的最大迭代次数，默认是没有限制的；参数decision_function_shape表示对多分类采用“ovr”策略或者“ovo”策略；参数random_state表示在对数据进行变换时使用伪随机树生成器的种子。

SVC()有以下一些比较重要的方法：

SVC().fit(X,y)方法表示对训练样本集X进行训练，y为X对应的样本标签；

SVC().score(X,y)方法表示对待预测样本集X进行预测，y为待预测样本集的标签，方法返回对待预测样本集的预测准确率。

SVC().predict(X)方法表示对待预测样本集X进行预测，返回预测的结果标签集。

SVC()的很多方法和KNeighborsClassifier ()是类似的，用起来也基本一样，下面是SVC()的简单实现代码：

```
from sklearn import svm
train_data = [[0, 0], [1, 1]]
train_target = [0, 1]
SVM = svm.SVC()
SVM.fit(train_data, train_target)
print("[2,2]的类别是：",SVM.predict([[2, 2]]))
```

上述程序的返回结果为：

上述结果表示待预测数据[2.0,2.0]的预测标签为[1]。至此，SVM分类器的基本构造方法已经实现了。

3.2.3 决策树算法实现

决策树算法是一种监督式的可用于分类和回归的算法，它的目标是创建一个从数据中学习分类规则的模型以对待预测数据进行预测。决策树算法便于理解，树的结构也可以可视化出来。Python 的sklearn.tree包给出了决策树的算法实现。该包用DecisionTreeClassifier()来实现决策树sss分类器，决策树分类器的构造函数如下：

DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False)

其中参数criterion是衡量分裂质量的函数，默认是“gini”表示衡量gini的纯度，也可以选择“entropy”，表示熵的信息增益；参数splitter表示每个节点的拆分策略，默认“best”表示选择最好的分裂策略，也可以选择“random”表示选择最好的随机分裂策略；参数max_depth表示构建的决策树的最大深度；参数min_samples_split表示分割内部节点所需的最小样本数，可以指定具体数目也可以指定为总样本数的百分比；参数min_sample_leaf表示叶节点所需的最小样本数，同样可以指定具体数目也可以指定为总样本数的百分比；参数min_weight_fraction_leaf表示在叶节点上所要求的加权总数的最小加权分数；参数max_features表示寻找最佳分裂时所需要考虑的特征数；参数random_state表示随机数生成器；参数max_leaf_nodes表示叶节点的最大节点数；参数min_impurity_decrease用来判断是否可以分割这个节点，如果分裂导致不纯度的降低大于或等于这个值，那么这个节点将被分割；参数min_impurity_split表示树生长停止的阈值；参数class_weight表示与类相关的权重；参数presort表示是否要对数据进行预分类；

决策树是能够进行多分类的分类器，它的一个优点是它生成的决策树可以可视化出来，使得我们可以更清楚地了解决策树。

首先我们需要对样本集进行训练：

```
from sklearn import tree
train_data = [[0, 0], [1, 1]]
train_target = [0, 1]
DT = tree.DecisionTreeClassifier()
DT = clf.fit(train_data, train_target)
```

训练完之后我们可以预测样本的类别：

```
DT.predict([[2, 2]])
```

输出为： array([1])

另外，我们还可以对样本进行概率预测：

```
DT.predict_proba([[2., 2.]])
```

输出为： array([[0. , 1.]])

经过训练我们可以用export_graphviz导出决策树的图像：

```
import graphviz
dataset = tree.export_graphviz(DT, out_file=None)
gv = graphviz.Source(dataset)
gv.render("iris")
dot = tree.export_graphviz(DT, out_file=None,feature_names=iris.feature_names, class_names=iris.target_names,filled=True, rounded=True,special_characters=True)
gv = graphviz.Source(dot)
gv
```

执行上述代码必须现在系统中安装graphviz包，读者可自行去网上查看安装教程，下图是代码生成的决策树：

图3-1 iris数据集生成的决策树

3.3实验结果

KNN分类算法是懒惰的算法，因为他不需要训练数据，而是在测试的时候计算距离的，因此他的训练时间很少，而SVM和决策树是需要提前训练数据的，因此他们的训练时间会比较长。下面我们测试一个来着UCI的低维大数据量的数据集skin_segmentation（3维，数据量245057，二分类），用前80%数据进行训练，后20%数据进行测试，以下是用三个分类算

法进行多次测试的结果：

表3-2 三个分类算法对skin_segmentation的预测结果

分类算法 KNN SVM 决策树

训练时间 0.53s 4553s 3.7s

预测时间 8.4s 119s 0.045s

预测准确率 99.5% 99.6% 99.8%

由表3-2可知，对于skin_segmentation数据集，用决策树算法是最好的。因为数据集数量比较大，所以SVM分类器训练的速度特别慢，训练出来的分类器比较复杂，因此预测时间也比较长；而KNN算法不需要训练，所以KNN分类器的训练时间很短，预测时间时间稍微长了点，为8.4s，可以说对于这类数据KNN算法是个不错的选择。而决策树算法之所以训练时间和预测时间很少是因为数据的维度很低并且是二分类，这使得决策树的构造变得很简单，由于决策树分类器的预测时间和决策树的高度成正比，决策树高度低自然就导致预测时间短。

下面我们再用wine数据集（13维，数据量178，3分类）测试，测试数据如下：

表3-2三个分类算法对wine的预测结果

分类算法 KNN SVM 决策树

训练时间 0.00065s 0.3s 0.075s

预测时间 0.0009s 0.0007s 0.00027s

准确率 72% 49% 65%

由表3-2可知，对于wine数据集用KNN算法是比较好的，KNN算法对于数据量较少的数据集分类效果是比较好的。

下面是对于madelon（500维，数据量2600,二分类）的预测结果：

表2-3 三个分类算法对madelon的预测结果

分类算法 KNN SVM 决策树

训练时间 0.083s 3.43s 0.72s

预测时间 1.28s 0.77s 0.0087s

准确率 76% 78% 53%

由表2-3可知，对于一些高维数据，用KNN也是可以的，SVM不是对于所有的高维数据都适用的。

KNN算法在处理很多数据集的时候是有优势的，一方面它不需要提前训练数据集，这使得对于大数据集的数据他的训练时间就很少，另一方面，对于那些分布不均匀的数据集，用基于半径的KNN分类算法，其效果也是很好的。

指　标

疑似剽窃文字表述

1. DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
2. =None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None,

7. 53140714-邱聪荣-计算机科学与技术-基于sklearn框架KNN分类算法的实现与应用.doc_第7部分　总字数：1414

相似文献列表　文字复制比：2.4%(34)　疑似剽窃观点：(0)

1	托卡马克等离子体中TEM模和ITG模的研究 张能(导师：龚学余) - 《南华大学博士论文》- 2016-05-01	2.4%（34） 是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第四章总结与展望

4.1 工作总结

本文详细地描述了目前流行的三种分类算法KNN分类算法，SVM分类算法和决策树分类算法的工作原理和优缺点。并且用python的sklearn包对三种算法进行了实现。然后从UCI上下载了一些具有代表性的数据集，用三种算法对数据集进行测试，通过观察三种算法的运行时间和准确率对算法性能进行分析。

实验结果表明，在中小型数据集中，用KNN算法的分类性能是不错的，不管是算法的训练时间还是算法的预测准确率，都是不比其他两个算法差的，甚至比他们好很多。由于KNN算法实现原理简单，预测的效果也不错，在对数据进行分类时选用KNN算法是个不错的选择。

在如今的信息高度膨胀的时代，如何从海量而又看起来毫无规律的数据中分析出有用的信息是一项很有意义的课题[13]。数据对于企业，行业甚至国家来说是很宝贵的资源，如何对它进行有效的分析是很有必要的，KNN算法的简便易懂和高效率为企业数据分析提供了一种很有效的方法。

4.2研究展望

由于KNN的懒惰性，使得KNN在训练是几乎不耗时间，而在测试时承担了所有的计算量，并且每次都是计算待测试样本和所有训练集数据的距离，这使得KNN对于较大的测试数据量的数据会花费很长的时间，基于这一点SVM和决策树就比较有优势，因为他们提前训练了数据，使得测试时所花的时间大大减少。

KNN分类算法目前还不够完善，对于处理大数据是效果不是很好，这需要进一步改进算法以克服这一缺点，并且KNN对于高维数据的预测效果也不是很好。KNN分类算法的改进之路还有很长，需要我们不断地探索。而一些其它的算法也一样，SVM算法的训练时间太长，需要进一步改进，决策树算法怎么生成最优的决策树至今任是一个难题，需要人们不断探索以改进。

在当今数据遍布的时代，如何找出一个好的数据分析算法是很有必要的，这需要众多研究者们对分类算法进行进一步的研究和改进。当然，在人们的不断探索下，最终肯定能寻找出一个非常高效的分类算法。

参考文献

[1]汲磊举. 大数据环境下动车组故障关联关系分析关键技术研究与实现[D].北京交通大学,2016.

[2]张兰廷. 大数据的社会价值与战略选择[D].中共中央党校,2014.

[3]Galton F.Co-relations and their measurement, chiefly from anthropometric data. Proceedings of the Royal Society of London,1888,45:135-145.

[4]徐树良,王俊红.结合无监督学习的数据流分类算法[J].模式识别与人工智能,2016,29(07):665-672.

[5]闭小梅,闭瑞华.KNN算法综述[J].科技创新导报,2009(14):31.

[6]徐渊,许晓亮,李才年,姜梅,张建国.结合SVM分类器与HOG特征提取的行人检测[J].计算机工程,2016,42(01):56-60+65.

[7]瞿合祚,刘恒,李晓明,黄建明.基于多标签随机森林的电能质量复合扰动分类方法[J].电力系统保护与控制,2017,45(11):1-7.

[8]周庆,牟超,杨丹.教育数据挖掘研究进展综述[J].软件学报,2015,26(11):3026-3042.

[9]Data Mining Curriculum. ACM SIGKDD. 2006-04-30 [2014-01-27].

[10]许静. 基于隐私保护的LBSNS（Location-Based Social Network Service）系统的设计与实现[D].南京邮电大学,2015.

[11]田彬, 胡瑾秋, 仝刚. 一种基于聚类思想的输油管道泄漏信号模式识别方法[C]// 天燃气管道技术研讨会. 2014.

[12]CJC: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery.

[13]陆俊儒. 基于支持向量机的高维不平衡数据二分类方法的研究[D].哈尔滨工业大学,2017.

[14]高远. 基于核方法的人脸识别[D]. 北京工业大学, 2010.

[15]李盼池，许少华，支持向量机在模式识别中的核函数特性分析[J].计算机工程与设计2005,26(2): 302-304.

[16]马宁.基于SVM分类与回归的图像去噪方法[N].兰州理工大学学报，2009-2：2.

[17]瞿益丹. 基于HHT和SVM的滚动轴承故障振动信号的诊断研究[D].中南大学,2012.

[18]刘冰.多类SVM分类算法的研究和改进[J].电脑知识与技术(学术交流),2007(06):1590-1593.

[19]基于KNN的文本分类特征选择与分类算法的研究与改进_黄娟娟

[20]张棪,曹健.面向大数据分析的决策树算法[J].计算机科学,2016,43(S1):374-379+383.

[21]A survey of cost-sensitive decision tree induction algorithms[J] . Susan Lomax,Sunil Vadera. ACM Computing Surveys (CSUR) . 2013 (2).

[22]Pruning belief decision tree methods in averaging and conjunctive approaches[J] . Salsabil Trabelsi,Zied Elouedi,Khaled Mellouli. International Journal of Approximate Reasoning . 2007 (3)

[23]黄文.决策树的经典算法:ID3与C4.5[J].四川文理学院学报,2007(05):16-18.

[24]UCI DataBase[DB].http://archive.ics.uci.edu/ml/index.php.

致谢

光阴似箭，转眼间大学四年即将过去，回首大学四年，感慨良多。大学四年里我经历了许许多多的事情，从懵懵懂懂的书呆子到即将步入社会的成年人，我经历了很多光荣和耻辱的事情。正是这坎坷的四年，才让我有了成长，有了进步。此时正值毕业论文完成之际，在此我要感谢一路走来帮助过我的老师，同学和家人。

首先，我要感谢刘桂霞教授。刘桂霞教授是我的毕业设计指导老师，之前考研期间，老师体谅我学习忙，给我的毕业设计出谋划策，让我能更专心的应付考研。之后是我找工作期间，刘桂霞老师又体谅我找工作压力大，给我的毕设又出了很多建议，这让我能在找工作的同时也完成毕业设计。对于刘桂霞老师在我毕设期间给我的帮助我十分的感谢！

其次我要感谢我们班的同学们，在做毕业设计的时候遇到了许许多多的困难，同学们给我提出了许许多多的建议，这对我有着很大的帮助，让我能解决难题，从而完成毕业设计。对于同学们这期间对我的帮助，我表示十分感谢！

再者我要感谢一路上在背后默默支持我的爸爸妈妈，哥哥姐姐，在我人生最低落的时候，你们永远站在我身后默默的支持着我，绝不会背叛我，让我即时掉到人生的低谷也会马上提起精神，积极面对生活。

最后我要感谢吉林大学，感谢计算机学院，感谢教过我的老师们，是你们培养了我，我会终生牢记你们。以及各位评委们，希望你们对我的论文给出宝贵的意见和建议，让我能更好的进步！

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例

2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例

4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的

6.红色文字表示文字复制部分;绿色文字表示引用部分

7.本报告单仅对您所选择比对资源范围内检测结果负责

amlc@cnki.net

http://check.cnki.net/

http://e.weibo.com/u/3194559873/

“中国知网”大学生论文检测系统