广西大学

硕士学位论文

基于机器学习的文本分类算法研究

姓名：杨挚诚

申请学位级别：硕士

专业：计算机应用技术

指导教师：苏德富

20070601

基于机器学习的文本分类算法研究

摘要

随着计算机技术、数据库技术、网络技术的飞速发展和 Internet 的日益普及，在现实生活中，每一个领域都不断产生海量数据和信息，特别是海量的文本数据。如何自动将这些文本数据进行分类整理，挖掘出有效信息，给人们有效利用，成为一个日趋重要的问题。因此，文本数据挖掘作为一门新兴学科，逐渐成为了一门引人注目、发展迅速的领域。

文本分类是文本数据挖掘中的一个基本技术，其作用是根据文本的各项特征判断其所属的预先设计的类别。它在自然语言处理与理解、信息组织与管理、内容信息过滤等领域具有非常广泛的应用。早期的文本分类采用的是基于知识工程和专家系统的方法，可是这样的方法非常复杂和缺乏灵活性。随着机器学习的兴起和发展，很多机器学习的分类器模型被引入的文本分类领域中，从不同的方面取得了不错的效果。

目前，各种文本分类算法都在一定的领域里有好的效果，但都不能成为通用方法，因此，如何对现有的文本分类算法进行评估也是一个非常重要的问题。分类的精度是已经被广泛用于评估文本分类算法性能的主要度量标准之一，但是，当要处理的类分布不均匀或者分类出错的代价不相同时，精度的局限性就显示出来了。在这种情况下，AUC

被提出作为一个新的评估文本分类算法性能的度量标准。已有研究表明，AUC 比精度的健壮性要好，而且有它特有的排序评测功能。这样，原有的分类算法在新的评估标准下是否和原来一样有效，是一个值得关注的问题。

由于新的标准的提出，目前还没有完整的实验对原有文本分类算法进行评测。本文将采用统一的文本基准集，重新对支持向量机，决策树，最近邻，朴素贝叶斯几个主流的文本分类算法进行实验比较，主要工作有：一是介绍和分析了几种主流的文本分类算法的基本原理；二是介绍了一种新的文本分类器评估标准，分析了它的评测原理以及和原有评估标准的比较；三是设计了详细的实验对几种主流文本分类算法进行测评，指出它们在新标准下的不足和今后需要改进的方向。

关键词：文本分类，朴素贝叶斯，支持向量机，决策树，最近邻，ROC 曲线下面积

RESEARCH ON TEXT CLASSIFICATION ALGORITHMS BASED ON MACHINE LEARNING

Abstract

With the rapid development of the techniques of computer, database and networks as well as the popularity of the Internet, in the real world life, there are more and more data and information generated in every domain, especially a great deal of text data. How to auto catalog and pick up these text data, get useful information to help people, becomes a more and more important problem. Thus text data mining, as a new subject, has gradually become a remarkable and fast developed area.

Text classification is one of the base techniques of text data mining, whose function is to assign the document to the preassigned class in terms of its features. Text classification is widely used in natural language process and analyzes, information organization and management and content filer area. The method early text classification used was based on knowledge engineering and export system, which was very complex and lack of agility. With the arisen and developing of machine learning, lot of classifier models have been introduced into the text classification domain, which have effect in different aspect.

Recently, different text classification algorithms are just used in different applications with a good performance. So it is an important problem how to select a currently best proper algorithm to apply for some application. Accuracy is one of the widely used main measures to evaluate classifier' s performance. But when processing some instances whose class distribution is unbalance or the error cost is difference, we couldn' t get a accuracy result by using accuracy. In this situation, AUC is proposed to be a new evaluation measure for the text classification performance. Some researches have showed that AUC is more robust than accuracy, and AUC could give an evaluation for ranking. Thus, It is a remarkable problem whether the current "well" text classification algorithms are still effective for the new measure.

Although the new measure has been proposed, there isn' t a whole evaluation for the classic text classification algorithms. This paper will report a controlled study with uniform datasets, comparing the performances of the SVM, decision tree, nearest neighbor, naive bayes and Multinomial event model naive bayes. The main works are below:

Firstly, Introduce and Analyze several popular text classification algorithms and their basic principle; Secondly,

Introduce a new evaluation measure for text classifiers, and analyze its evaluate principle, finally make a comparison with the old measure. Thirdly, Design a particular experiment to evaluate the performance of several popular text classification algorithms, point out their scarcity in the new measure and indicate the direction how to improve.

Keywords: text classification, naive bayes, SVM, decision tree, nearest neighbor, AUC

第一章 绪论

1.1 课题研究背景及意义

随着计算机技术、数据库技术、网络技术的飞速发展，Internet 的广泛应用，信息交换越来越方便，使得各个领域都不断产生海量数据，尤其是海量的文本数据。怎样从这些海量数据中挖掘出有用的信息和知识，方便人们的查阅和应用，已经成为一个日趋重要的问题。因此，文本数据挖掘已经逐渐成为一个引人注目，发展迅速的领域。

文本分类是文本数据挖掘中的一个基本技术，其作用是根据文本的某个特征，把它分到预先定义的类别中。传统的文本分类模式是基于知识工程和专家系统的，在灵活性和分类效果上都有很大的缺陷。例如卡内基集团为路透社开发的 Construe 专家系统就是采用知识工程方法构造的一个著名的文本分类系统[1]，然而该系统的开发工作量达到了 10 个人年。当需要进行信息更新的时候，维护起来就更加困难。由此可见，知识工程方法已不适用于日益复杂的文本分类系统需求。分类是机器学习的核心问题之一，20 世纪 90 年代以来，机器学习的分类算法有了日新月异的发展，其中的很多分类器模型也逐步被应用到文本分类之中，比如支持向量机（SVM，Support Vector Machine）、最近邻法（Nearest Neighbor）、决策树（Decision tree）、朴素贝叶斯（Naive Bayes）等，它们在不同的领域体现了不同的分类效果。

文本分类在自然语言处理与理解、信息组织与管理、内容信息过滤等领域具有非常广泛的应用。在应用中如何选择一个合适的分类模型是一个重要的问题。预测的精确度（accuracy）是传统用于评估分类器性能的重要度量标准之一，很多关于如果改进分类算法精确度的研究已经展开和完成。但是精确度作为度量标准有不足之处：一是待分类实例的类分布是不变的或者是比较平衡的，当类分布不均衡的时候，精确度的误差会加大；二是采用精确度作为标准来分类的时候，51%的概率和 99%的概率是一样的，没有去考虑两种概率所体现的离目标的距离；三是默认不同种类的分类错误代价是一样的。但是在现实的应用中这些不足是不应该有的。比如，在内容信息过滤的时候，将有用的文本分类为无用的远比错误的分配无用文本所付出的代价会大得多，这样，我们需要根据一个文本可能有用的概率来对文本进行排序。

AUC（The Area Under the ROC Curve）是近年来提出的一个新的评估标准，

一些研究证明，与传统的精确度相比，AUC 具有更好的鲁棒性。同时，通过提高AUC可以有效的提高分类算法的排序性能。我们将采用AUC作为评估标准，在统一的数据集上对几个经典的文本分类算法进行实验评测，对比它们的排序性能。通过实验数据的对比，我们发现多项式贝叶斯不论在各个方面都强于其他用于比较的分类器。

1.2 国内外研究现状和发展

1.2.1 文本分类研究进展

文本分类的概念早在二十世纪六十年代就开始存在，早期的算法基于手工构建详细的分类规则集；直到八十年代，文本分类的主要手段还局限于所谓的知识工程方法，这种方法需要将语言学知识与具体领域的专家知识结合起来由人工整理出描述文本类别的逻辑规则集，然后利用这个规则集通过推理实现文本分类。但是用知识工程方法构造文本自动分类系统常常会遇到知识获取瓶颈的问题，而这个难以解决的问题也阻碍了知识工程方法运用在文本分类上的进一步的发展。到了九十年代知识工程的方法逐渐被机器学习的方法所取代[2]。

1971 年，Rocchio[3]提出了在用户查询中不断通过用户的反馈来向用户提供更有用的信息。之后，Mark[4]将这种方法引进文本分类中，根据分类器分类的正例和反例的数量来学习类的权向量，但不足的是这种方法只能对少量的文本进行学习。Mun[5]提出了一种基于错误驱动（error-driven）的学习方法，它通过乘除来修改正例和反例的权重参数。

在决策树方面，1986 年，针对训练集过大而导致内存不足的情况，Quinlan 引入基于窗口的增量学习方法，提出了著名的 ID3 方法[6]。该方法采用信息增益作为选择属性的标准。在 ID3 的基础上，研究者们又提出了 C4.5 算法[7]，C5 算法和 CART 方法，并且将决策树方法运用[2][6][8]在文本分类领域里。Lewis[43]在路透社的文档集上对决策树的分类效果作了评估，Provost[9]、Su[10][37]在他们的论文中提出了对基于树推导理论的算法的基于概率排序的改进，并用实验进行了求证。

Cover和Hart[11]在1967年就提出了近邻分类思想，这是一种基于实例的学习方法。最近邻算法是一种简单有效的分类算法，它与其它分类学习算法不同的是

它不需要事先构建分类器，但是要先存储所有的训练样例，当训练数据数量巨大是，这就需要很多的存储空间。因此，David和Dennis等人[12]提出了一种通过牺牲分类精度来减少空间消耗的方法。Yang [13]经过实验证实k近邻算法对于文本分类具有良好的效果，但是Sebastiani [14]指出了*k*近邻算法的不足之处。国内方面，李程雄，丁月华等[15]则将可kNN与SVM结合来提高分类精度，并应用在专利文本分类上。

贝叶斯文本分类器主要有两种，一种是基于独立性假设的朴素贝叶斯文本分类器，另一种是考虑了属性间依赖关系的贝叶斯网络分类器。前者的独立性假设虽然在现实中不能成立，毕竟单词间的关联是显而易见的，但是在实践中却依然取得了很好的分类效果，因此，在经典贝叶斯理论的基础上提出的朴素贝叶斯一直都是研究的热点。Nitesh在他的博士论文[16]上对比了常见的四种基于朴素贝叶斯的模型，指出多项式模型在不平衡数据集上有很好的分类效果，Andrew 和Kamal [17]则对其中两种模型在文本包含单词量大小的问题上做了评测。在Web 中文文档自动分类方面国内中科院的史忠植教授等使用朴素贝叶斯分类器作为分类模型，通过一定的EM 迭代算法实现了半监督Web 文本挖掘[18] 。他们使用网页收集器Spider 从http://www.fm365.com 收集了关于体育方面的网页近500 篇，在贝叶斯潜在语义（Bayesian Latent Semantic，BLS） 模型的框架下，首先利用潜在语义分析标注含有潜在类别变量的文档的类别，然后结合朴素贝叶斯模型以及未标注文档的知识对这些文档进行分类。他们的实验表明该方法具有很好的精确度和召回率。靳小波等[19]将Lee模型引入朴素贝叶斯中提高分类精度。

自1995 年Vapnik[20]与其领导的贝尔实验室研究小组系统地提出统计学习理论以来，作为一种构造性的统计学习方法，以追求间隔最大化为目标的支持向量机技术掀起了一股新的研究热潮。1997 年德国的Joachime[21]首次将支持向量机成功应用于大规模文本自动分类中。随后，研究者将主动学习算法与支持向量机结合[22]，取得了很好的效果。针对支持向量机中求解二次规划的问题，Platt提出了一种新的求解模型SMO[31][32]，提高了求解速度。Yang [13]采用统一的数据基准集将SVM和一些主流文本分类算法进行比较后指出SVM在分类精度上有着很好的表现。

1.2.2 分类评估标准

对文本分类器的评估标准有很多，出错率，查全率，查准率都是常用到的方法，为了解决查全率和查准率的评估矛盾问题，又提出了联合评估方式如 F_β，查准率/查全率平衡点(Precision- Recall-Break-Even-Point, PRBEP)等等。在基于机器学习的文本分类中，精确度（accuracy）是被广泛用于评估文本分类算法性能的主要度量标准。许多研究已经在精度上对各种文本分类算法进行了比较[13][21][40]，并提出了对各种算法的改进方法 [23]来提高分类精度。但是，精度在某些方面[24][38]有一定的局限性。在一些文本分类的应用中，排序精度比精度更有价值。比如，在搜索一个文本时，用户更希望得到的是这个文本可能所属分类的一个排序，而不仅仅是文本属于某一类。AUC(The Area Under The ROC Curve)非常适合用来作为数据分类排序精度的度量标准[25]。近年来，在数据分类的领域，越来越多以提高分类算法AUC的研究被展开，一些主流算法如SVM[26][34]、决策树[27][28]、贝叶斯[29]等都得到了比较和改进。将AUC用于评估和改进文本分类方面的研究[30][35][42]也在进行当中，但仍处于初始阶段，各种主流文本分类算法的排序性能比较以及如何提高排序性能将成为研究的热点。

1.3 本文的主要工作和组织结构

1.3.1 本文的主要工作

本文的研究目的是采用新的性能度量标准对传统文本分类算法进行重新的测评分析，比较它们在不平衡类分布文本数据集和大文本数据集的排序性能。

（1）简要介绍文本分类在现实应用中重要性，分析了文本分类算法中几个经典算法支持向量机、k 最近邻、决策树和朴素贝叶斯分类器的理论基础；

（2）分析了几种分类评估标准的理论依据，探讨了它们的不足之处。详细分析了新提出的评测标准 AUC，并且和另一评测标准精度进行了对比。

（3）设置了一系列实验，采用统一的文本分类基准集，对支持向量机、k 最近邻、决策树和朴素贝叶斯分类器等主流文本分类算法进行各种性能的分析，主要研究了这些算法在不平衡类分布数据集和大文本数据集上精度和排序精度的优劣，为后续研究提供实验基础和数据支持，以及为应用时对分类器的选择提供指导意见。

1.3.2 本文的组织结构

本文的组织结构如下：

第一章 绪论。阐明了本文的研究背景和研究意义，综述了基于机器学习的文本分类算法国内外的研究现状以及文本分类评估标准的发展，说明了本文的所要进行的研究工作。

第二章 文本分类。简要说明了自动文本分类的过程和步骤，概述了文本表述和特征选取的方法，分析了一些主要文本分类算法的理论基础和实现步骤。

第三章 文本分类评估标准。简要介绍了目前几种常用的文本分类评估标准，重点介绍了 ROC 曲线下面积（AUC）的原理，对比了 AUC 和精度之间的特点，指出 AUC 评估优于精度而且可以用单一数值描述排序精度。

第四章 实验比较。说明了实验的软硬件环境，数据集的选取原则，实验设计的动机，最后给出了实验研究的结论。

第五章 总结和展望。对本文所做的工作进行总结，并指出今后需要继续研究和实现的方向。

第二章　文本分类

2.1 文本分类算法概述

早期的算法基于手工构建详细的分类规则集，一些精度很高的分类器通过这种方法构造，不过代价很高，需要一批领域专家具备大量的专业知识，花费大量的时间才能构造出正确且完整的规则集，而且以后研究领域的变更会使已有的规则集作大规模的修改甚至全部重新构建。所以除极少数的特例外，这种手工构建分类器的方法缺乏实用价值。自动文本分类通过监督学习自动构建出分类器，同样可以达到较高的分类性能，而且花费比人工建立分类规则集要小得多。

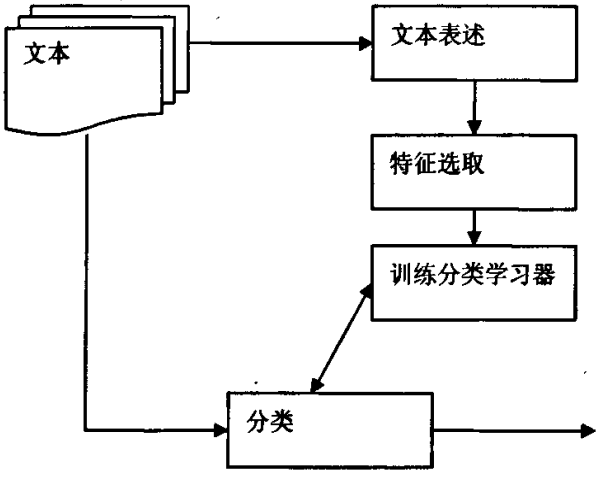


图 2-1 文本分类一般流程

Fig 2-1 general workflow of text classification

常用的自动文本分类算法主要有基于 TFIDF 权值计算方法的分类器算法，其基本思想是利用 TFIDF 权值公式计算一个词在文档中的重要性，然后用 cosine 距离公式计算两个文档的相似度，这类算法包括 Rocchio 算法，TFIDF 算法，k 邻近算法（k Nearest Neighbors，简称 kNN)等；基于概率和信息理论的分类算法，如朴素贝叶斯算法（Naive Bayes，简称 NB），最大熵法（Maximum Entropy）等；基于知识学习的分类算法，如决策树（Decision Tree），人工神经网络（Artificial Neural Networks，简称 ANN），支持向量机（Support Vector Machine,简称 SVM）等算法。自动文本分类的流程的一般流程基本如图 2-1 所示。

2.2 文本表述和特征选择

2.2.1 文本表述

记录文本内容的格式和实现方法有很多，文本分类器无法直接处理形式各样的文本文档，因此使用分类器之前首先要将平面文本文档性质转换为一种定量的形式，即转换成可以让分类器处理的数据格式。

我们可以把文本可以看成是段落的集合，或者是句子的集合，也可以看成是单词或字母的集合。单词是组成文本的一个基本单位，研究者通常把一个文本当作是一系列单词的集合来表示，即所谓的词包（Bag of Words）表示法。由于单词在不同的文本之间出现的频率高而且存在一定的统计规律，因此，以单词作为基本单位来表示文本时文本自动分类研究中常用的也是比较有效的方法，而Salton[33]提出的空间向量模型（Vector Space Modal, VSM）是实际应用中常见的文本表示模型。

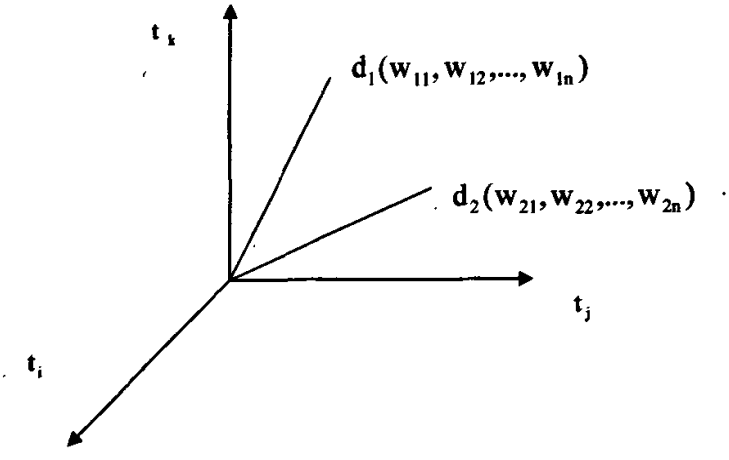


图 2-2 向量空间模型

Fig 2-2 Vector space modal

定义 2-1 向量空间模型

给定一个文本文档 $d = d(w_1, w_2, \ldots, w_r)$，项 t_k 可以在文档的不同位置重复出现，为了简化分析，通常不考虑项 t_k 在文档中出现的先后次序并要求项 t_k 互异，这时可以把 $t_1, t_2, \ldots, t_r$ 看成一个 r 维的坐标系，而 $w_1, w_2, \ldots, w_r$ 为响应的坐标值，

这样 $d(w_1,w_2,...,w_r)$ 可以被看成是 r 维空间中的一个向量，称 $d(w_1,w_2,...,w_r)$ 为文档 d 的向量表示，如图 2-2 所示。

向量空间模型已经在信息检索、文本分类等应用中取得了成功。目前，在文本表示上普遍还处在语法级别上，基于语义的研究也已逐步展开。

2.2.2 特征选择

采用词袋方式来表述文本，通常会遇到一些问题，比如一些连接词，助词等词出现的次数比较多，不但无法用来作为分类的特征，还会影响正常的判断；第二是动词的形态很多，比如一个动词有正常形态，过去时，正在进行时，第三人称表述等，但是从词义来看都是表达同样的意思。对于这两个问题，可以用 FOX 提出的禁用词表（Stop list）和取词根的方法来解决，这样的方法就是一种比较简单的特征选择方法。

通过特征选择出来的单词，也就是表述文档的各个特征相可以用很多种方法来给他们设定权值，比如通过专家给他们设定的语义来分配权值，或者通过判断他们是否在文档中出现，或者在文档中出现的次数等。最后提到的方法叫做词频法。

2.3 主流文本分类算法

2.3.1 支持向量机（Support Vector Machine，SVM）

支持向量机是 Vapnik 与其领导的贝尔实验室研究小组根据统计学理论提出的一种学习方法，基于结构风险最小化原则和核函数方法，通过构造间隔最大的最优分类超平面构造决策函数，很好地解决了学习机的泛化能力和复杂性问题。支持向量机方法有以下几个优点：

1）它是专门针对有限样本情况的，其目标是得到现有信息下的最优解而不仅仅是样本趋于无穷大时的最优值；

2）算法最终将转化成为一个二次型寻优问题，从理论上说，得到的将是全局最优点，解决了在神经网络方法中无法避免的局部
极值问题；

3）算法将实际问题通过非线性变换转换到原空间中的非线性判别函数，特殊

性质能保证机器有较好的推广能力。

支持向量机分类算法思想，是从训练样本中寻找能够确定一个最优超平面的支持向量。假设有大小为 m 的训练样本集$\{\langle x_1,y_1\rangle,\langle x_2,y_2\rangle,...,\langle x_m,y_m\rangle\}$，如果它是一个二分类任务，分类标识为 $y_i=\pm 1(i=1,2,...,m)$，那么，这个任务的决策函数可以表示为：

$$f(x)=\mathrm{sign}(w\cdot x+b) \qquad (2.1)$$

那么，支持向量机需要解决下面的一个优化问题：

$$\min_{w,b,\xi} \quad \frac{1}{2}w^{T}w+C\sum_{i=1}^{m}\xi_i \qquad (2.2)$$

并且上述公式满足条件：

$$y_i(w^{T}\Phi(x_i)+b)\geq 1-\xi_i \qquad (2.3)$$

$$\xi_i\geq 0$$

在这里，训练向量x_i通过函数Φ被映射到高维空间中，然后支持向量机将在这个高维空间中寻找一个带有最大间隔的线性可分超平面。可以使用拉格朗日优化方法将最优分类面问题转化为一个对偶最优化问题：

$$W(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j\Phi(x_i)^{T}\Phi(x_j) \qquad (2.4)$$

$K(x_i,x_j)=\Phi(x_i)^{T}\Phi(x_j)$，称之为核函数，常用的函数有：

1）线性函数$K(x_i,x)=x_i\cdot x$

2）多项式函数$K(x_i,x)=(x_i\cdot x+1)^{d}$

3）径向基函数$K(x_i,x)=\exp(\frac{-\|x-x_i\|^2}{\sigma^2})$

4）多层感知器函数$K(x_i,x)=\tanh(kx_i\cdot x+\theta)$

我们在计算时只需要计算这些核函数，而不用去直接计算复杂的高维空间中的非线性函数，这样就可以有效的避免了特征空间维数灾难问题。由于支持向量机机坚实的理论基础和它在很多领域表现出良好的推广性能，目前国际上正在广泛

开展对支持向量机方法的研究。研究者们相继开发出了很多 SVM 快速训练算法，例如 Joachims 的 SVM^{light} [11]，John Plat 的 SMO[12] (Sequential Minimal Optimization)等算法，使得 SVM 在文本分类领域取得了很大的成功。

SMO 方法是一种简单的算法，它能快速求解 SVM 的二次规划问题，从而提高训练 SVM 的速度。按照 Osuna 的理论，在保证收敛的情况下，把 SVM 的二次规划问题分解成一系列子问题来解决，其他算法相比， SMO 方法的优点在于，优化问题只有两个拉格朗日乘子，它用分析的方法即可解出，从而完全避免了复杂的数值解法；另外，它不需要巨大的矩阵存储，这样，即使是很大的 SVM 学习问题，也可在 PC 机上实现。

2.3.2 k 最近邻算法（k Nearest Neighbor, kNN）

最近邻算法是通过查询已知样例的情况，来判断新样例与已知样例是否属于同一类。最近邻算法存在许多变种，但其基本思路都是先存储全部或部分训练样例，对于测试样例，通过相似函数计算它与所存储的训练列子的距离以决定类别的归属。也就是说最近邻学习算法不需要训练分类器模型。

k 最近邻算法是最近邻学习算法的一个例子。kNN 法最初由 Cover 和 Hart 于 1968 年提出，是一个理论上比较成熟的方法。采用 kNN 方法进行文档分类基本思想是：对于给定的文档集 D，把 D 中所有的文本内容形式化为特征空间中的加权特征向量 **D**，其中向量 **D** 表示为 $D = D(T_1, W_1; T_2, W_2; \ldots; T_n, W_n)$。对于某一给定的测试文档 d，通过计算它与每个训练文档的相似度，找出 k 个最相似的文档。在此基础上，给每个文档类别加权打分，根据加权距离和判断测试文本所属的类别。具体步骤可描述如下：

1、根据各种规则将文本内容变换成文本特征向量。

2、根据相似度公式计算测试文本与每个训练文本的相似度，计算公式如下：

$$SIM(d_i, d_j) = \frac{\sum_{k=1}^{m} W_{ik} \times W_{jk}}{\sqrt{\left(\sum_{k=1}^{m} W_{ik}^2\right)\left(\sum_{k=1}^{m} W_{jk}^2\right)}} \qquad (2.5)$$

其中，m 是特征向量维数， K 值的确定目前还没有很好的方法，一般采用先定一

个初始值，然后根据实验测试的结果调整 K 值，一般初始值定为几百到几千之间，但是要小于训练文档总数。

3、从 2 的结果中选出 k 个相似度最大的训练集文档，计算分类权重：

$$P(d,c_i)=\sum_{d_j\in kNN} SIM(d,d_j)y(d_j,c_i)-b_i \qquad (2.6)$$

其中d表示文本特征向量，$y(d_j,c_i)\in\{0,1\}$，即如果文档属于该类别值为1，反之为0；b_i为阈值，对于某一特定类来说，b_i是一个有待优化选择的值，可以通过一个验证文档集来进行调整。

kNN 分类算法的优点是易于快速实现，在基于统计的模式识别中非常有效，对于未知和非正态分布可以取得较高的分类准确率，并且，它对训练数据中的噪声有很好的健壮性，当给定足够大的训练集合时也非常有效。同时，作为一种懒散的学习算法，它也存在一些限制：一是空间开销大。因为要事先存储全部训练样例，当训练样例增大时存储空间也随之增大。二是计算相似度时，实例间的距离是根据实例的所有属性来计算的，这与那些只选择全部实例属性的一个子集的方法不同，例如决策树。比如每个实例由 20 个属性描述，但在这些属性中仅有 2 个与它的分类有关。在这种情况下，这两个相关属性的值一致的实例　可能在这个 20 维的实力空间中相距很远。结果，依赖着 20 个属性的相似性度量会误导 k-近邻算法的分类。近邻间的距离会被大量的不相关属性所支配。这种由于存在很多不相关属性所导致的难题，有时被称为唯度灾难（curse of dimensionality）。最近邻方法对这个问题特别敏感。解决的方法一般有两种，一是在计算两个实例间的距离时对每个属性加权，二是从实例空间中完全消除最不相关的属性。

2.3.3　决策树算法

决策树学习时应用最广泛的归纳推理算法之一，它是一种逼近离散值函数的方法，对噪声数据有很好的健壮性且能够学习析取表达式。决策树着眼于从一组无次序无规则的事例中推理出决策树表示形式的分类规则，它通过把实例从根结点排序（sort）到某个叶子结点来分类实例，叶子结点即为实例所属的分类。在构造分类模型时，树上的每个结点指定了对实例属性集测试后选择出的属性，并且该结点的每一个后继分支对应于该属性的一个可能值。分类实例的时候，就是从

树的结点开始，测试这个结点指定的属性，然后按照给定实例的该属性值对应的树枝向下移动，之后在新的结点上重复这个过程直到叶子结点，即获得分类。

图2-3画出了一棵典型的决策树样例。这棵树根据天气情况分类“星期六上午是否适合打网球”。

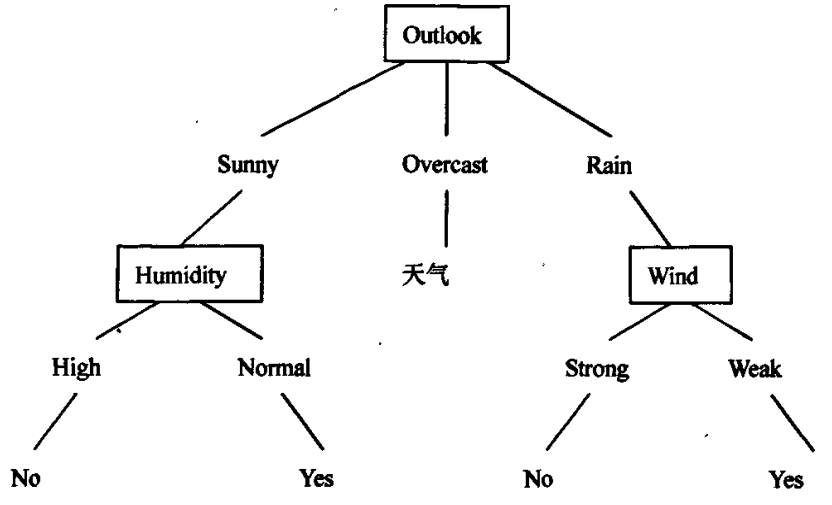


图2-3 概念“Play Tennis”的决策树

Fig2-3 Decision Tree of concept “Play Tennis”

ID3算法和C4.5算法是决策树算法中两个经典算法，许多后续研究都是在这两个算法的基础上进行的，它们都采用的是自顶向下的贪婪搜索遍历可能的决策树空间。算法的核心就是如何选取在树结点上要测试的属性，哪个属性是对分类最有效的属性。这里引用了信息论中熵的概念。 设一个样例集S，相对于c个状态的分类的熵定义为：

$$\text{Entropy}(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i \qquad (2.7)$$

其中p_i是S中属于类别i的比例。在以熵作为衡量训练样例结合纯度的标准后，可以定义为分类训练数据选择属性的标准，这个标准称之为信息增益（Information Gain）：

$$\text{Gain}(S,A) \equiv \text{Entropy}(S) - \sum_{v \in \text{Value}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \qquad (2.8)$$

其中，Values（A）是属性A所有值的一个集合，S_v是S中属性A的值为v的子集。

信息增益是ID3算法中生成树结点时选择最佳属性的度量标准，但是它存在一个内在偏置，就是值越多的属性其增益也越大。因此，还有很多其他的属性选择度量方法。

2.3.4 朴素贝叶斯算法（Naive Bayes，NB）

贝叶斯分类是一种统计学的分类方法。通过贝叶斯公式，我们可以由先验概率(Prior Probability)计算出后验概率(Posterior Probability)。后验概率比先验概率提供更多的信息，从而可以作为分类的标准。贝叶斯方法对新实例分类时的方法是在给定描述实例的属性值$\langle a_1, a_2 \ldots a_n \rangle$下，得到最可能的目标值$v_{MAP}$：

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j \mid a_1, a_2 \ldots a_n) \qquad (2.9)$$

通过使用贝叶斯公式将此表达式重写为：

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \ldots a_n \mid v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \ldots a_n \mid v_j) P(v_j) \qquad (2.10)$$

朴素贝叶斯（Naive Bayes）是贝叶斯学习方法中一种实用性很高的学习器，它有一个“独立性假设”，即假定不同属性对分类结果的影响是独立的，观察到联合的$a_1, a_2 \ldots a_n$的概率等于单个单独属性的概率乘积：

$$P(a_1, a_2 \ldots a_n \mid v_j) = \prod_i P(a_i \mid v_j) \qquad (2.11)$$

因此，将它带入公式（2.10），朴素贝叶斯分类器可以表述为：

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i \mid v_j) \qquad (2.12)$$

v_{NB}表示朴素贝叶斯分类器输出的目标值，这样，概括的讲，朴素贝叶斯方法需要估计的对象为$P(a_i \mid v_j)$和$P(v_j)$。对应于文本分类问题，v_j是分类标签j，而a是文本的特征向量，我们需要求的就是类别j里文本特征a_i的概率和类别j在训练文档集里出现的概率。根据文本特征抽取选择方法和用于计算的概率模型的不同，

$P(a_i|v_j)$和$P(v_j)$的计算方法也不同。

多变元伯努利事件模型（Multi-Variate Bernoulli event model）、多项式事件模型（Multinomial event model）和泊松模型（Poisson model）是比较常用基于朴素贝叶斯独立性假设的先序概率模型。其中多项式朴素贝叶斯被证明是比较好的一种方法。假设有训练文档集D，$d_i \in D$，i=1,2,...,|D|，文档d_i属于分类C中某一个分类c_j，$c_j \in \{c_1,c_2,...,c_{|C|}\}$，那么，对于测试文档集中任意文档$d_i$属于哪一个分类的的描述为：

$$\text{Max}\, P(c_j|d_i)=\frac{P(c_j)P(d_i|c_j)}{P(d_i)} \qquad (2.13)$$

设文档d_i是由一系列属于词汇表V中的单词组成的，Nit表示单词ω_t在文档d_i中出现的次数，如果文档符合多项式分布，则：

$$P(d_i|c_j)=P(|d_i|)|di|!\prod_{t=1}^{|V|}\frac{P(\omega_t|c_j)^{N_{it}}}{N_{it}!} \qquad (2.14)$$

其中，$P(\omega_t|c_j)$可以采用拉普拉斯先验概率估算，表示为：

$$P(\omega_t|c_j)=\frac{1+\sum_{i=1}^{|D|}N_{it}P(c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|D|}N_{is}P(c_j|d_i)} \qquad (2.15)$$

其中：

$$P(c_j|d_i)=\begin{cases}1 & d_i\text{的分类是}c_j\\ 0 & d_i\text{的分类不是是}c_j\end{cases}$$

由此计算出文档d_i可能属于的分类。

第三章　文本分类的评估标准

对文本分类器的评估有很多，不同的分类方法可能会偏好某些评估方法，也就是说对分类方法的改进也是基于某一种标准上的改进。下面我们将介绍一些常用的评估标准，并在下一节重点介绍一种新的评估标准-ROC下的区域。

3.1常用评估标准

对于两分类问题，样本集中有n个样本，分类$C \in \{0,1\}$，N_p和N_n分别表示正例和反例的个数，即$N_p+N_n=N$；另外设正例被判断为正例和反例的个数分别是T_p、T_n，反例被判断为正例和反例的个数分别为F_p和F_n。表3-1显示了上面描述的一个邻接表。

表3-1

Table3-1

	正例	反例
判断为正	T_p	F_p
判断为反	T_n	F_n

在机器学习领域，错误率是一个比较重要的评估标准。由上面的定义可以得到错误率的公式：

$$Err=\frac{判断错误的样例数}{样本集中所有样例数}=\frac{T_n+F_p}{T_p+T_n+F_p+F_n}=\frac{T_n+F_p}{N}$$

查准率（Precision）和查全率(Recall)是两个在信息检索领域中常用的评估标准,他们的公式如下：

$$Pre=\frac{判断正确的正例数}{判断为正例的样例数}=\frac{T_p}{T_p+F_p}$$

$$Rec=\frac{判断正确的正例数}{样例集中的正例数}=\frac{T_p}{T_p+T_n}$$

显然，如果所有的文档都被分类为正，查全率就是100%，但是查准率只有N_p/N；而如果只有一个分类为正的样例被分类为正，同时全部分类为反的样例都被正确

分类，这样，查准率是100%，但是查全率就只有$1/N_p$了。所以，如果查全率和查准率都具有一定的片面性。为了在这两个评估值上取一个平衡点，综合评价分类性能，这样就提出了F_β，公式如下：

$$F_\beta = \frac{(1+\beta^2)Pre * Rec}{\beta^2 Pre + Rec}$$

最常用的是令$\beta=1$，也就是 $F_1 = \frac{2Pre * Rec}{Pre + Rec}$。

另外一个常用的评估标准是预测精度（predict accuracy），简称精度，其公式为：

$$Acc = \frac{判断正确的样例数}{全部样例数} = \frac{T_p + F_n}{T_p + F_n + T_n + F_p}$$

3.2 受试者工作特征曲线(Receive Operating Characteristic Curve，ROC曲线)

ROC分析是一种早期用于信号检测理论中的评估技术，迄今为止，ROC分析已经被广泛用于医疗诊断、模式识别和数据挖掘等领域。在用于分类器性能评估时，对于二分类问题，ROC曲线展示了分类器预测所得到的正确正例率（True Positive Rate，TPR）和错误正例率（False Positive Rate，FPR）之间的对映关系。根据本节开始时做出的假设，TPR和FPR可以用公式来表示：$TPR = \frac{T_p}{N_p}$、$FPR = \frac{F_p}{N_n}$。

绘制ROC曲线时，纵轴对应于TPR，横轴对应于FPR，基本图形如图3-1所示。图中点（0，1）表示所有的正例都被判断为正例，而所有的反例都没有被判断为正例，显然，这是最好的分类效果；点（1，0）表示所有的正例都没有被判断为正例，而所有的反例都被判断为正例，这是最坏的分类效果。图3-1中，曲线的弧顶越接近点（0，1）表示分类的性能越好，如曲线A代表的分类器优于曲线B代表的分类器，因为A可能出现的错误分类代价期望比B的期望要低。如果一条ROC曲线连接着原点（0，0）和（1，1）的直线，表示了该分类器的分类效果和用随机法一样差。

当不同分类器的ROC曲线在坐标图中没有交叉的时候，很容易判断出哪个分类器最好，但是实际情况并不是这么简单。如图3-2 所示，曲线A和曲线B是有交点的，在交点的左边A优于B，而在交点右边B优于A，这样，从整体上就很难判断分

类器A与B哪一个更好。

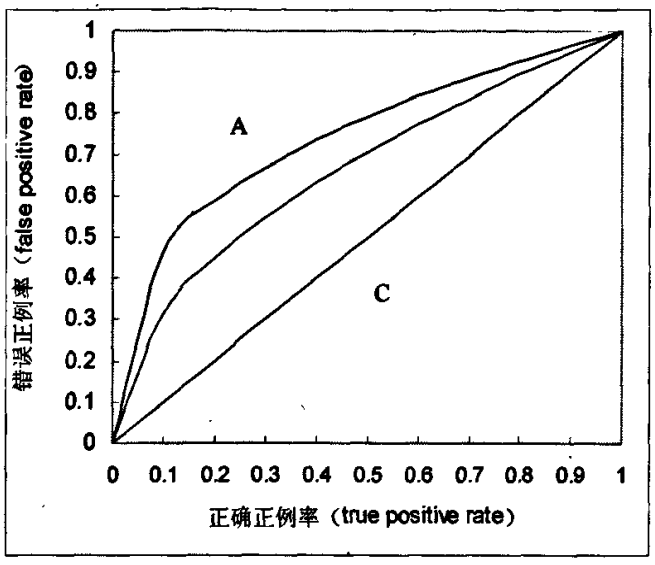


图3-1 一般的ROC曲线图

Table3-1 A normal ROC curve

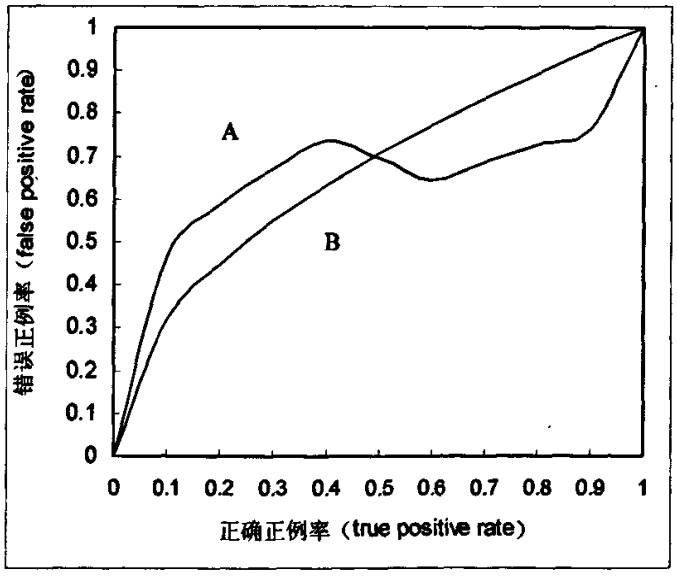


图3-2 交叉的ROC图

Table3-2 ROC diagram of cross curves

3.3 ROC 曲线下的区域（The Area Under the ROC Curve, AUC）

通过一个无交叉点的 ROC 图可以很容易看出不同分类器的性能优劣，但是当

ROC 曲线存在交点时就难以从整体判断了。ROC 曲线下的区域的面积，如图 3-3 所示，简称 AUC，作为全局数字评估标准能够很好得解决这个问题。这个面积代表了从测试集中随机选择一个正例比随机选择一个反例的概率要高[25]。

计算 AUC 的方法有很多种，我们选择的是 Provost 和 Fawcett[24]提出的无参求法。二分类任务时计算 AUC 的公式如下：

$$\hat{A} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}$$

其中，n_0和n_1分别是测试样例集中正例和反例的个数，$S_0 = \sum r_i$，r_i是第 i 个正例在实例排序上排列的序号。

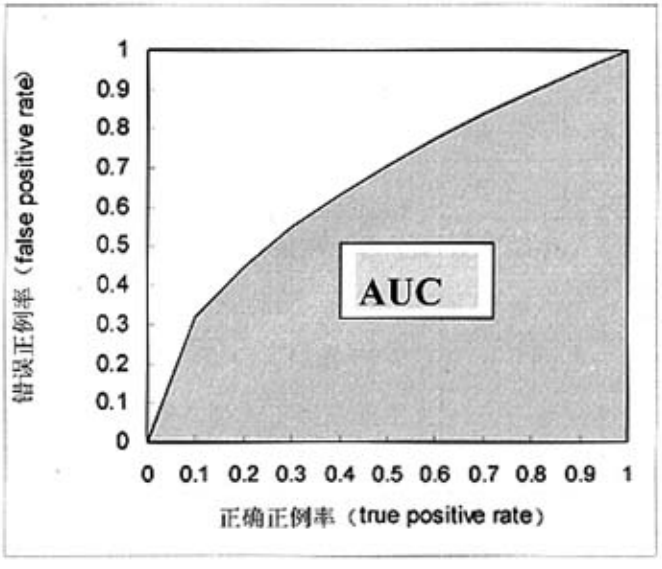


图 3-3 ROC 曲线下区域

Fig3-3 The area under then ROC Curve

例如，在一个测试集中有五个正例和五个反例，即$n_0 = n_1 = 5$，如果这十个测试样例被预测为正的概率分别为{0.9，0.99，0.6，0.4，0.7}U{0.55，0.23，0.1，0.44，0.3}，放在一起排序后如表 3-1 所示，这样$S_0 = 4+7+8+9+10 = 38$，所以，AUC=（38-5*（5+1）/2）/（5*5）=（38-15）/25=23/25。由公式和例子可以看出，正例在测试集中排序的位置和S_0是决定 AUC 的关键。表 3-2 显示了上例中

表 3-1 排序例子

Table3-1 A example of ranking

序号	1	2	3	4	5	6	7	8	9	10
标示	负	负	负	正	负	负	正	正	正	正
数值	0.1	0.23	0.3	0.4	0.44	0.5	0.6	0.7	0.9	0.99

几个特殊排序下 AUC 的值。

表 3-2 几个特殊的排序

Table3-2 several special ranking

情况	正例序号	S_0	$n_0(n_0+1)/2$	AUC
A	1, 2, 3, 4, 5	15	15	0
B	1, 3, 5, 8, 10	27	15	12/25
C	2, 4, 5, 7, 9	27	15	12/25
D	6, 7, 8, 9, 10	40	15	1

A 情况是分类最差的情况，所有正例被判断为正的概率都小于反例被判断为正例的概率，反之，D 是分类效果最好的情况，即所有的正例被判断为正的概率都大于反例被判断为正的概率。这里，我们可以看到 AUC 的一个特点：不受分类阈值的影响。不论是概率大于 0.5 就判断为正还是概率大于 0.6 就判断为正，只考虑正例在整个测试样例中的排序位置，或者说是正例被判断为正的总体趋势是否大于相反的情况。

如果是多分类任务， David 和 Robert[25]提出通过 M 估量来计算 AUC。假设有 c 个分类的数据集，$c \geq 2$，A（i, j）表示 i 类标签作为正例，j 类标签作为反例时的 AUC 值。当 c=2 时 A（i, j）=A（j, i）；当 c>2 时，$A(i,j) \neq A(j,i)$。此时，简单的用算术平均值求二分类{i，j}的 AUC 值：$\hat{A}(i,j)=\frac{A(i,j)+A(j,i)}{2}$。对于 c 个分类的数据集，需要求 c(c-1)/2 个二分类对，所以：

$$M=\frac{2}{c(c-1)}\sum_{i<j}\hat{A}(i,j)$$

在下一章的实验中我们求的 AUC 采用的就是这个公式。

3.4 预测精度与 AUC

预测精度（后面都简称为精度）是传统用于评估分类器性能的重要度量标准之一，它可以明确给出在指定阈值的情况下一个分类器正确分类的百分比。但是，如果分类阈值有所变化，那么原先评测的结果就不再可用，必须重新设定。因此它的健壮性比较差。另外，绪论中也提到了精度其他方面的不足：

（1） 待分类实例的类分布是发生变化或者类分布不均衡的时候，精确度的误差会加大；

（2） 许多分类算法在分类的过程中会产生一些概率估计信息，采用精确度作为标准来分类的时候，精度没有利用这些信息，因此，当分类不论以 51%的概率还是 99%的概率成立都是一样的，没有去考虑两种概率所体现的离目标的距离；

（3） 默认不同种类的分类错误代价是一样的。

另外，在现实应用中很多时候需要一个精确的排序结果而不仅仅是判断的精确度。比如在一个购物营销方案运用中，我们需要通过以往顾客的购物习惯来判断是否向客户发购物邀请。那么，我们希望不仅仅是简单的得到客户是否会购物，而是更希望得到一个根据客户会购物的可能性的排序，因为失去一个客户远比多投一张邀请的代价要高得多。同样，如果在一个文本检索系统中，我们也需要一个感兴趣的重要资料的排序。因此，我们需要一个能够评价分类器排序性能的度量标准，而上面提到的精度在忽略概率信息方面的不足使得它不能用来评估排序精度。前一节中介绍的 AUC 恰恰能满足这一需求。让我们看看表 3-3 的精度和 AUC 对比的例子。简单起见，仍然假设在一个二分类任务测试集中有五个正例和五个反例，即 $n_0=n_1=5$。求精度（公式中用 acc）是以 50%为求精阈值，表中，A1 和 A8 是两个最极端的情况，显然 acc(A1)=auc（A1）=1, acc（A8）=auc（A8）=0；精度值和 AUC 值相等；在 A2-A5 的例子中，精度值都是相等的，acc(A2)=acc(A3)=acc(A4)=acc(A5)=0.8；而 AUC 充分利用了正例在排序位置的不同，各不相同。auc(A2)=0.8, auc(A3)=0.84, auc(A4)=0.96, auc(A5)=0.72。这样，我们可以看到：

$$acc(A2)=auc(A2),\ acc(A3)<auc(A3)$$

$$acc(A4)<auc(A4),\ acc(A5)>auc(A5)$$

表 3-3 精度和 AUC 的比较

Table3-3 A comparison between accuracy and AUC

序号	实例排序（概率递增）	精度	AUC
A1	负负负负负|正正正正正	1	1
A2	负负正负负|正正负正正	4/5	4/5
A3	负正负负负|负正正正正	4/5	21/25
A4	负负负负正|负正正正正	4/5	24/25
A5	负负正负负|正正正正负	4/5	18/25
A6	负正正负负|正负负正正	3/5	3/5
A7	负负负正正|负负正正正	3/5	21/25
A8	正正正正正|负负负负负	0	0

在精度不变的情况下，AUC 的值则有不同的值，有的大，有的小。再来看看 A6 和 A7 的情况，acc(A6)=acc(A7)=0.6，而 auc(A6)=0.6<auc(A7)=0.84。综合 A2 到 A7 的情况看：

$$acc(A2)=acc(A3)=acc(A4)>acc(A7)$$

$$而\ auc(A2)<auc(A3)=auc(A7)<auc(A4)$$

由上可见，AUC对分类器性能描述比精度对分类器性能的描述要丰富得多。当然，哪一个评估标准更好不能通过描述是否丰富来判断，Charles X. Ling 和Jin Huang[39]第一次通过实验和演算证明了AUC不但可以用于评估排序的精度，甚至可以替代精度。

他们首先做了四个定义：一致性（Consistency）、判别性（discriminancy）、一致度（Degree of Consistency）、判别度（Degree of Discriminancy）。

定义 1 （一致性）对于度量标准 f 和 g，在区域Ψ内如果不存在$a,b\in\tilde{\Psi}$，使得 $f(a)>f(b)$ 并且$g(a)<g(b)$，则 f 和 g 是（严格）一致的。

定义 2（判别性）对于度量标准 f 和 g，在区域Ψ内存在$a,b\in\Psi$使得$f(a)>f(b)$ 并且$g(a)=g(b)$，同时，不存在$a,b\in\Psi$使得$g(a)>g(b)$并且$f(a)=f(b)$，那么 f 比 g

更具有判别性。

定义 3（一致度）对于度量标准 f 和 g，在区域 Ψ 内，让 $R=\{(a,b)\mid a,b\in\Psi,f(a)>f(b),g(a)>g(b)\},S=\{(a,b)\mid a,b\in\Psi,f(a)>f(b),g(a)<g(b)\}$。F 和 g 的一致度是 C（$0\le C\le 1$），$C=\frac{|R|}{|R|+|S|}$。

定义 4（判别度）对于度量标准 f 和 g，在区域 Ψ 内，让 $P=\{(a,b)\mid a,b\in\Psi,f(a)>f(b),g(a)=g(b)\},Q=\{(a,b)\mid a,b\in\Psi,g(a)>g(b),f(a)=f(b)\}$。这样 f 优于 g 的判别度 $D=\frac{|P|}{|Q|}$。

在进行了以上基础定义之后，为了判别那个标准更好，还做了一个定义：

定义5 度量标准 f 在统计意义上与 g 一致，并且比 g 更具备判别性，当且仅当 C>0.5 并且 D〉1。在这种情况下我们称 f 是比 g 更好的标准。

第四章 文本分类算法比较实验

4.1 实验设计

4.1.1 设计动机

在精度作为评估标准的时候，研究者通过各种方法改进进分类算法的预测精度，同时，各种基础测试也不断在进行，测试比较的结果成为其他研究者改进的基石。从 Provost[24]将 ROC 曲线用于整体序列和错误代价可视化研究到 Bradly[43]用 AUC 作为评估标准对流行分类算法评估，基于 AUC 和排序精度的研究已经一步步的展开。起初 AUC 只能用于二分类研究，对多分类数据集就无法评估；David J. Hand[25]在 2001 年提出了基于 AUC 的 M 估计用于多分类评估，进一步促进了 AUC 在文本分类上的评估作用。03 年 Charles X. Ling 等人[39]不但从经验上分析了 AUC 的优点和适用性，还从理论上证明了 AUC 不但是作为排序精度的度量标准，而且比精度更健壮。同时，不少研究者已经对支持向量机、朴素贝叶斯、决策树进行了精度方面的改进和评测[44][45][46]，Huang[43]等在一些手工数据集上对上述分类器进行了比较，但是还没有研究者对这几种流行的分类器在 AUC 方面进行综合的评价。因此我们从选择文本基准语料集开始，针对普通文本数据，不平衡类分布文本数据和大文本集数据进行实验综合评测，试图给出一个综合的评价。

4.1.2 算法选择

实验中使用的是分类算法是 weka 中已经实现的算法，我们要进行实验对比的支持向量机，朴素贝叶斯，多项式朴素贝叶斯，k 最近邻和决策树算法分别使用 weka 软件中的 SMO、Naive Bayes、NaiveBayesMultinomial、IBk 和 J48 算法。在测试过程中，对于 k 最近邻算法，在尝试了各种 k 值后均得不到理想的结果——分类精度小于 30%，排序精度小于 60%，这两项结果均远低于其他四种分类器，因此不具可比性，因此在下一节实验结果中没有列出来。

SMO 实现的是 Platt 提出的 SVM 解决方法，该算法会自动填补缺值属性，通过成对分类解决多类分类问。SMO 算法有一些列参数，不同的参赛选择对计算的结果会有影响。经过一些前期的测试，对它选择了如下参数，能取得支持向量机最好的性能：

buildLogisticModels - False。不采用逻辑模型。

c - 复杂性参数选择 2.0。

checksTurnedOff - False，不关闭时间消耗的检测。

debug - False，不需要在 console 端输出一些额外的调试信息。

epsilon -- 1.0E-12。ε的值。

filterType - Normalize training data。规格化训练数据。

kernel - PolyKernel -C 250007 -E 1.0。采用多项式核函数。

numFolds -- -1。折叠的次数。

randomSeed - 1。交叉验证的随机数。

toleranceParameter -- 0.0010。

朴素贝叶斯和多项式贝叶斯没有什么特别的参数，于是都采用默认的参数配置。J48 决策树算法采用如下配置：

binarySplits - False，在实型属性上不采用二元划分

confidenceFactor -0.25，进行树裁剪的是的置信因子。

debug -- False，不需要在 console 端输出一些额外的调试信息。

minNumObj -每个叶子节点上允许存在 5 个样例

reducedErrorPruning - 不需要进行错误递减裁剪。

unpruned - yes，需要裁剪树

useLaplace - 不使用 Laplace。

4.2 实验配置

4.2.1 实验环境

本实验的进行的硬件环境为 CPU P4 2.4G，RAM 512M，硬盘 80G。软件环境为数据挖掘工具 weka。

Weka（Waikato Environment for Knowledge Analysis）系统[36]是新西兰 Waikto 大学开发的一个开源代码的机器学习及数据挖掘工具。它是基于 Java 语言实现的，因此可以运行在多个操作系统之下，目前已在 Linux、Windows、Macintosh 等多个平台下通过测试。weka 的操作环境下提供了很多功能，包括对数据的预处理，分类、回归、聚类、关联规则和可视化。weka 自带有很多数据挖掘算法，由

于是开源软件，除了 Weka 本身已经包含的数据挖掘算法，用户还可以很方便的依据 weka 提供的接口函数改进已有算法或者添加自己的算法，与原有的算法进行比较，因此 weka 逐渐成为数据挖掘领域的一个流行的工具。

Weka 还提供了多个连接多种数据的接口，有简单的.arff 和.cvs 的文件格式，还可以通过 JDBC 接口与 sqlserver、oracle 等数据库连接来完成数据挖掘任务。我们的实验中使用的是.arff 格式文件，下面将简要的介绍一下这种文件的存储格式：

```
文件 weather.arff
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,85,85,FALSE,no
```

文件中各项属性都以@符号起头，@relation 表示数据集名称，此数据集名称为 weather；@attribute 表示属性和类别，就是可以作为对某个类别做出判断的依据的特征及判断的类别。每个数据集众可以有很多特征，如上例中有 outlook、temperature、humidity、windy 五个特征，play 是类别；在 weka 中只支持 3 种类型，离散型、实数型和字符串型。比如 outlook 特征的{sunny, overcast, rainy}就是离散性；temperature 就是实数型。最后是属性@data，表示的就是一条判断记录，上例中，如果 outlook=sunny, temperature=85,humidity=85,windy=FALSE,则 play 属性判断为 no。

4.2.2 数据集

实验的数据集是 15 个文本分类基准数据集，这些数据集被广泛用于文本分类实验，它们主要来源于 Reuters[49]、TREC[48] 、WebACE[50][51][52][53]和 OHSUMED[47]，表 4.1 列举了我们实验中用到的这些数据集的详细信息，包括每个数据集中包含的文档数量，数据集的词汇表单词总数，每个数据集的类别数，最

小类别中包含的文档数和最大类别中包含的文档数以及每个数据集中包含文档最少的类别的文档数在文档中的百分比。

Reuters 路透社于1987年播发的21578篇财经新闻，是文本分类研究中应用最广泛的基准语料之一。Re0，和Re1是从Reuters中选取出的只由一个分类标签的文档集。

OHSUMED 一个医学文摘语料库。

TREC Text REtrieval Conference，是由美国国家标准技术局(National Institute of Standards and Technology,简称NIST)和国防部高级研究计划局(Defense Advanced Research Projects Agency,简称DARPA)组织召开的一年一度的国际会议,是文本检索领域最权威的国际会议之一,代表了当今世界文本检索领域的最高水平。Tr11, tr12, tr21, tr23, tr31, tr41, tr45和Fbis是从TREC-5以及TREC-6中分离出来的。

WebACE WebACE工程。Wap中每个文档对应一个雅虎网页。

表4-1 数据集总结

Table4-1 Summary of data sets

数据集来源	数据集	文档数	分类数	分类中最小的文档数	分类中最多的文档数	最小文档数百分比	单词数
OHSUMED	Oh0	1003	10	51	194	5.08%	3182
OHSUMED	Oh5	918	10	59	149	6.43%	3012
OHSUMED	Oh10	1050	10	52	165	4.95%	3238
OHSUMED	Oh15	913	10	53	157	5.81%	3100
TREC	Fbis	2463	17	38	506	1.54%	2000
TREC	Tr11	414	9	6	132	1.45%	6429
TREC	Tr12	313	8	9	93	2.88%	5804
TREC	Tr21	336	6	4	231	1.19%	7902
TREC	Tr23	204	6	6	91	2.94%	5832
TREC	Tr31	927	7	2	352	0.22%	10128
TREC	Tr41	878	10	9	243	1.03%	7454
TREC	Tr45	690	10	14	160	2.03%	8261
Reuters	Re0	1504	13	11	608	0.73%	2886
Reuters	Re1	1657	25	10	371	0.60%	3758
WebACE	wap	1560	20	5	341	0.32%	8460

这些数据集的文件格式均符合 weka 要求的 arff 文件格式，其中@data 属性下每一行代表一个文件，文件的表示形式为词汇表中单词在文件中出现的次数，不考虑单词出现的位置，文档的最后一个属性为该文档的分类类别。比如，词汇表中有六个单词 {IBM, sport, game, football, match, man} 和两个类别 {computer, sports}，一个类别为 sports 的文件用{0，1，0，3，4，2，sports}表示，也就是在文档中单词 IBM 和 game 都没有出现过，单词 sport 出现过一次，单词 football、match、man 分别出现过 3、4、2 次。由于在真实文件中词汇表单词数是比较多的，而出现在每个文档中的单词相对较少，因此如果以上面的形式存储将存在大量的“0”单词，所以，我们实验中用到的数据集文件采用了 arff 文件的另一种格式，稀疏数据存储格式来存储。例如，上面的例子就用这样的方式：{2 1，4 3，5 4，6 1，7 sports}。

在这些数据集中，tr12, tr21, tr23, tr31, tr41 和 wap 的最小分类中的文档数均小于 10 个，在这里我们称之为不平衡数据集；tr31, re0, re1 和 wap 的最小分类的文档数占总文档数的百分均比小于 1%，在这里我们称之为失衡数据集。另外，tr21 数据集的详细情况如表 4-2 所示：

表 4-2 tr21 数据集类分布

Table4-2 class distribution of dataset tr21

类编号	文档数	总数	百分比
0	231	336	0.6875
1	16	336	0.0476
2	9	336	0.0268
3	41	336	0.122
4	35	336	0.1042
5	4	336	0.0119

表 4-2 显示了数据集 tr21 的类分布情况，其中类型 0 占了总样例数将近 70%，而其他的分类最高不超过 2%，类分布非常不平衡。在其他的数据集中没有这样的情况。

4.3 实验数据及分析

实验中采用的测试方法是 10 折交叉验证（10-folder cross validation），对每一个数据集进行 3 次重复的 10 折交叉验证，然后取平均值进行比较。在同一个数据集下不同分类器效果的比较采用统计学意义的 T 测试。下面表格及正文中出现的 SMO、NBM、NB 和 J48 分别表示 SMO、多项式朴素贝叶斯、朴素贝叶斯和决策树。表格第一列的算法作为基准比较算法，在这里我们选择 SMO,用其余的分类器和 SMO 比较，NBM、NB 和 J48 列后的符号表示采用置信度为 95% 双尾 T 测试与 SMO 比较的结果，v 表示在统计学的意义上数值大于 SMO，*表示数值小于 SMO，如果没有符号表示统计学意义上没有区别。在下列图表中，SMO 表示采用 SMO 方法训练 SVM 的支持向量机方法，MNB 表示多项式模型朴素贝叶斯，NB 表示朴素贝叶斯算法，C4.5 表示 Weka 中的 J48 算法。

4.3.1 精度分析

表 4-3 列出了支持向量机、多项式朴素贝叶斯、朴素贝叶斯和决策树代表算法在 15 个文本数据集上测试得到的精度值。从平均值可以看出，支持向量机在精度上的表现好于朴素贝叶斯但略逊于多项式朴素贝叶斯和决策树算法。多项式朴素贝叶斯在精度上的表现最好，而朴素贝叶斯的精度比较差，还没有达到 70%，尤其是在 tr21 数据集上的精度还没有达到 50%。同样的情况也发生在多项式朴素贝叶斯上，平均值在 80%以上的多项式朴素贝叶斯在 tr21 数据集上只等到了 63.37% 的预测精度。因此，朴素贝叶斯方法在极不均衡类分布的数据集上表现不如支持向量机和决策树算法。

表 4-3 SMO、MNB、NB 和 C4.5 的精度结果

Table4-3 The Accuracy of the SMO、MNB、NB and C4.5

数据集	SMO	MNB		NB		C4.5	
fbis.mat	78.4±2.47	76.9±2.37		62.61±2.28	*	73.81±2.2	*
oh0.mat	81.96±4.5	89.03±2.91	v	79.66±4.9		83.05±5.73	
oh10.mat	74.86±3.84	81.24±3.65	v	72.67±2.88		75.52±4.54	
oh15.mat	72.72±3.62	83.78±3.09	v	75.24±3.23		77.54±2.58	v
oh5.mat	77.45±2.46	86.7±3.8	v	77.88±2.73		81.47±4.64	
re0.mat	75.47±3.2	80.38±3.23	v	57.51±4	*	75.73±2.91	
re1.mat	74.29±2.12	83.35±3.8	v	66.33±4.2	*	80.27±3.47	v
tr11.mat	74.17±8.65	84.79±4.84	v	54.83±8.41	*	72.25±5.82	
tr12.mat	74.46±7.96	83.05±8.16	v	54.67±5.19	*	77.93±4.94	
tr21.mat	79.46±4.33	63.37±8.61	*	46.39±8.96	*	82.75±5.84	
tr23.mat	74.12±11.65	71.55±12.81		55.79±7.87	*	92.1±7.54	v
tr31.mat	92.13±3.65	94.61±3.2		80.69±5.13	*	92.56±1.55	
tr41.mat	87.02±4.11	94.42±2.35	v	85.65±3.71		91.57±2.29	
tr45.mat	81.16±5.34	83.04±5.02		65.51±7.69	*	89.57±3.12	v
wap.mat	82.18±2.13	81.09±2.1		72.76±2.69	*	67.37±2.25	*
平均值	78.66	82.49		67.21		80.90	

图 4-1(a)显示了四个分类算法在不平衡数据集上的表现，图标的 X 轴表示的是数据集，按照数据集中文档数排序，Y 轴是分类算法在数据集上的精度。从图中可以看到，MNB 的趋势线高于 SMO，SMO 的趋势线高于 NB，决策树 J48 在趋势线上的表现不明显。图 4-1(b)显示了四个分类算法在失衡数据集上的表现，与图 4-1（a）坐标轴表示一致。从图中看出，在失衡数据集上，MNB 依然表现最好，NB 最差，决策树算法不稳定。

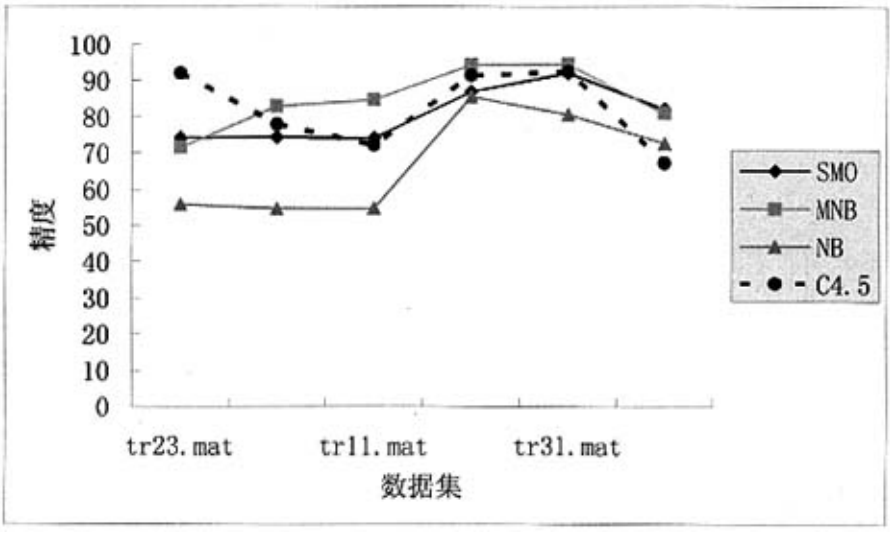


图 4-1(a) 分类算法在不平衡数据上的表现

Fig4-1(a) Performance of the classification algorithms in unbalance dataset

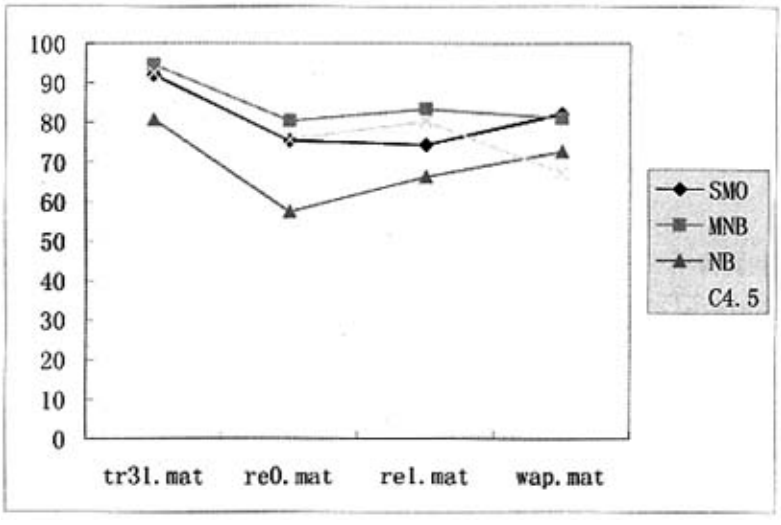


图 4-1(b) 分类算法在不平衡数据上的表现

Fig4-1(b) Performance of the classification algorithms in unbalance dataset

表 4-4 显示了通过置信度为 95%的双尾 t 测试的比较结果，这些数据从统计学意义上表明多项式朴素贝叶斯在 15 个数据集中有 9 个好于支持向量机而仅有一个表现差于它；朴素贝叶斯算法没有在一个数据集上表现好于支持向量机并且在三

分之二的数据集上逊于它；决策树算法则与支持向量机相当，表现略好，优于后者的数据集数为 4 个，另有 2 个不如后者。在多项式朴素贝叶斯和决策树的比较中，多项式贝叶斯有 4 个好于决策树而 3 个差于它。

表 4-4 所有数据集 T 测试结果

Table4-4 T-test summary of all datasets

	SMO	MNB	NB	C4.5
SMO		5-3-1	0-5-10	4-9-2
MNB			0-0-15	3-8-4
NB				10-4-1

表 4-5 不平衡数据集 T 测试结果

Table4-5 T-test summary of unbalance datasets

	SMO	MNB	NB	C4.5
SMO		9-5-1	0-1-8	2-6-1
MNB			0-0-9	1-4-4
NB				6-2-1

4.3.2 AUC 分析

表 4-6、表 4-7 分别显示了支持向量机、多项式朴素贝叶斯、朴素贝叶斯与决策树的在全部数据集上的排序精度值和采用置信度为 95% 双尾 T 测试后决策树、多项式朴素贝叶斯、朴素贝叶斯与支持向量机比较的结果。朴素贝叶斯以 88%的平均排序精度略低于 SMO，根据 T-测试比较的结果看，朴素贝叶斯在其中 11 个数据集上的表现与 SMO 不相上下，但其余 4 个均不如 SMO；多项式朴素贝叶斯的表现则好于 SMO，其中平均精度高于 SMO3.62 个百分点，T-测试的结果表明多项式朴素贝叶斯在所有数据集上都不比 SMO 差，其中更在 8 个数据集上优于 SMO；而多项式贝叶斯的运行速度也远比 SMO 和朴素贝叶斯快得多，排序精度也高于其他分类算法。

表 4-6 文本分类算法在 AUC 上的性能

Table4-6 Performance of Text classification algorithms in AUC

Dataset	SMO	MNB		NB			C4.5		
fbis.mat	95.2±1.39	96.2±1.17		90.34	±2.37	*	88.66	±1.89	*
oh0.mat	95.81±2.1	98.58±0.67	v	95.26	±1.85		93.99	±2.75	
oh10.mat	93.04±2	96.57±1.1	v	92.75	±0.91		92.13	±3.07	
oh15.mat	92.49±1.81	95.98±2.19	v	91.73	±2.17		93.15	±2.37	
oh5.mat	94.84±1.15	98.05±1.04	v	94.68	±1.66		94.46	±1.91	
re0.mat	91.13±2.83	94.7±3.18		87.34	±5.15		79.02	±5.63	*
re1.mat	93.92±2.48	94±2.71		88.77	±2.93	*	89.08	±2.79	*
tr11.mat	90.2±3.09	93.39±2.44		83.26	±5.81	*	84.38	±4.07	
tr12.mat	89.22±6.1	96.3±2.93	v	85.47	±4.5		90.8	±5.74	
tr21.mat	80.26±8	86.73±5.18		69.99	±6.44		81.78	±8.28	
tr23.mat	84±8.93	92.17±5.91	v	83.19	±7.77		92.08	±5.88	v
tr31.mat	92.56±3.36	92.7±2.97		91.76	±2.18		93.06	±2.5	
tr41.mat	94.68±3.39	97.4±2.01		93.7	±2.82		92.94	±2.94	
tr45.mat	89.65±4.48	95.02±2.6	v	88.25	±3.37		96.33	±3.41	v
wap.mat	90.68±2.49	94.23±1.99	v	83.46	±3.42	*	77.36	±5.18	*
平均	91.18	94.80		88.00			89.28		

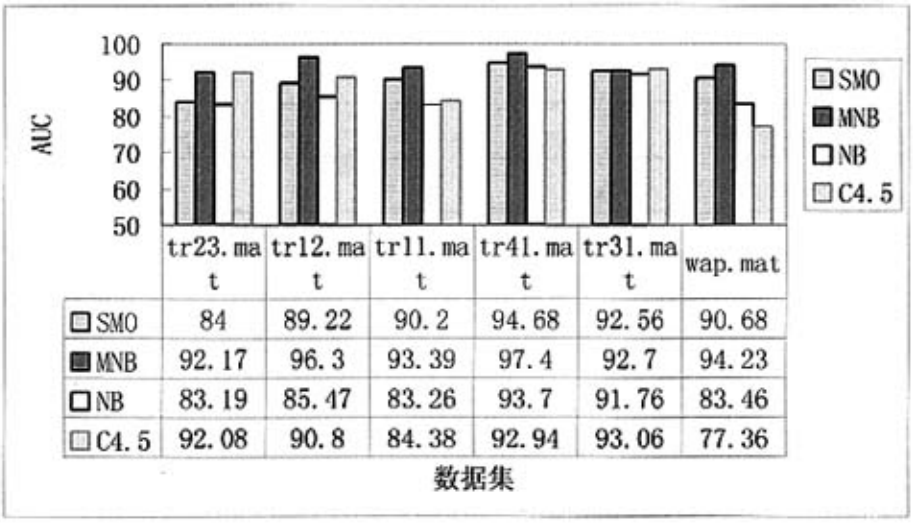
决策树算法在排序精度上的表现并不好，它在 4 个数据集上差于 SMO，只在两个数据集上高于 SMO；而与多项式朴素贝叶斯比较的时候，决策树更是在 11 个数据集上不如多项式。

表 4-7 AUC 的 T 测试结果

Fig4-7 T-test summary of AUC

	SMO	MNB	NB	C4.5
SMO		8-7-0	0-11-4	2-9-4
MNB			0-1-14	0-4-11
NB				2-11-2

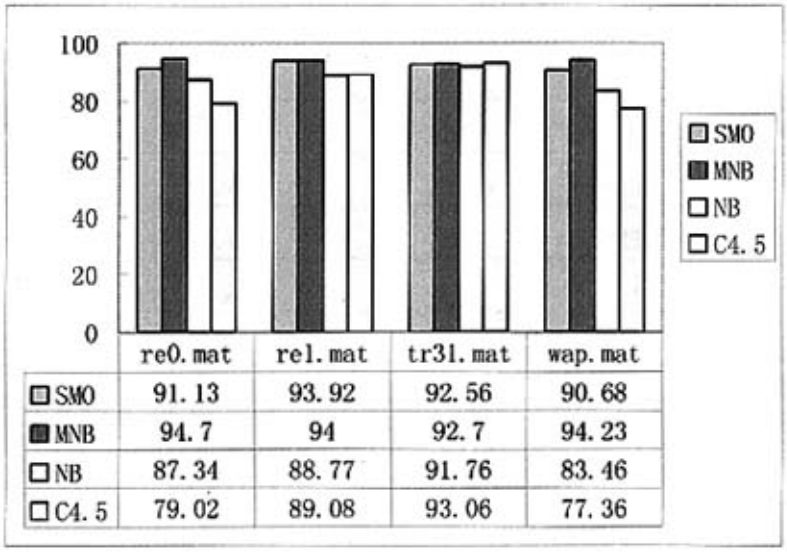
表 4-2(a)显示了四个文本分类算法在不平衡数据集上的排序精度，MNB 在所有数据集上都好于其他三种文本分类器。



	tr23.mat	tr12.mat	tr11.mat	tr41.mat	tr31.mat	wap.mat
SMO	84	89.22	90.2	94.68	92.56	90.68
MNB	92.17	96.3	93.39	97.4	92.7	94.23
NB	83.19	85.47	83.26	93.7	91.76	83.46
C4.5	92.08	90.8	84.38	92.94	93.06	77.36

图 4-2(a) 文本分类器在不平衡数据集上的排序性能

Table4-2(a) Ranking performance of text classifier in unbalance datasets



	re0.mat	rel.mat	tr31.mat	wap.mat
SMO	91.13	93.92	92.56	90.68
MNB	94.7	94	92.7	94.23
NB	87.34	88.77	91.76	83.46
C4.5	79.02	89.08	93.06	77.36

图 4-2(b) 文本分类器在不平衡数据集上的排序性能

Fig4-2(a) Ranking performance of text classifier in unbalance datasets

在 tr21 极不平衡数据集上，表 4-8 综合对比了四个分类器在精度和排序精度上的表现。精度上多项式朴素贝叶斯和贝叶斯都不如 SMO，但是，在排序精度上，

表 4-8 SMO、MNB、NB、C4.5 在数据集 tr21 上的比较

Table4-8 A comparison among SMO、MNB、NB、C4.5 in dataset tr21

度量	SMO	MNB		NB		C4.5	
精度	79.46±4.33	63.37±8.61	*	46.39±8.96	*	82.75±5.84	
AUC	80.26±8	86.73±5.18		69.99±6.44		81.78±8.28	

多项式朴素贝叶斯和朴素贝叶斯与 SMO 在统计意义上没有显著差别。

从运行时间上看，表 4-9 和图 4-3 显示出决策树算法用时最长，朴素贝叶斯在

表 4-9 分类器训练时间

Table4-5 Training time of classifiers

Dataset	SMO	MNB		NB		C4.5	
fbis.mat	50.53	0.14	*	26.89	*	476.28	v
oh0.mat	11.34	0.04	*	8.18	*	130.17	v
oh10.mat	12.43	0.03	*	9	*	149.72	v
oh15.mat	10.97	0.03	*	7.34	*	130.48	v
oh5.mat	10.85	0.03	*	7.07	*	83.35	v
re0.mat	22.53	0.03	*	12.51	*	297.91	v
re1.mat	73.62	0.06	*	20.19	*	493.09	v
tr11.mat	10.02	0.08	*	8.01	*	63.85	v
tr12.mat	7.26	0.06	*	5.23	*	32.21	v
tr21.mat	7.07	0.08	*	8.09	v	90.38	v
tr23.mat	4.34	0.05	*	3.34	*	14.74	v
tr31.mat	15.47	0.1	*	36.42	v	315.2	v
tr41.mat	17.79	0.09	*	23.76	v	179.41	v
tr45.mat	16.27	0.1	*	19.85	v	116.89	v
wap.mat	56.93	0.18	*	57.85	v	2199.41	v
平均	21.83	0.07		16.92		318.21	

平均时间上用时长于 SMO，采用 T-测试结果比较后发现，在 10 个数据集上朴素贝叶斯杯 SMO 用时要短，另外 4 个用时长于 SMO，平均训练时间也少于 SMO；而多项式贝叶斯的运行速度远比 SMO 和朴素贝叶斯快得多。

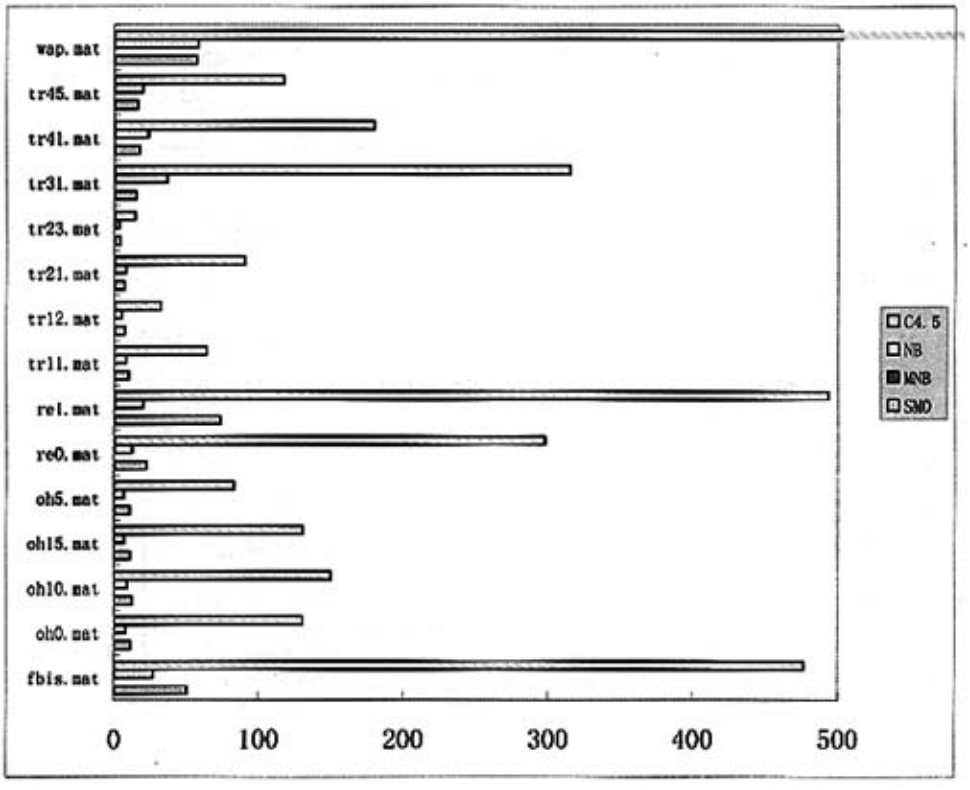


图 4-3 训练时间

Fig4-3 Train time

4.3.3 大容量数据集下的比较

前一个实验中对一般容量的数据集进行了测试，但是现实接触到文本容量——词汇表和文档数量都是巨大的。这里，我们选择了三个大文本数据集，详细信息如表 4-10 所示：

表 4-10 大文本数据集信息

Table4-10 information of large scale dataset

数据集来源	数据集	文档数	分类数	分类中最小的文档数	分类中最多的文档数	最小文档数百分比	单词数
OHSUMED	Ohscal	11162	10	709	1621	6.35%	11465
TREC	La1	3204	6	273	943	8.52%	31472
TREC	La2	3075	6	248	905	8.07%	31472

选择的数据集一个来源于 OHSUMED 的 Ohscal 数据集，另外两个来源于 TREC 的 La1 和 La2 数据集。由于决策树算法用时过长，效率上远低于其它几个分类算法，我们只对 SMO、MNB 和 NB 进行了比较。

表 4-11 SMO、MNB、NB 在大文本数据集下的精度

Table4-11 Accuracy of SMO、MNB and NB in large text datasets

dataset	SMO	MNB		NB	
la1	84.44±1.89	88.36±1.39	v	74.95±2.4	*
la2	86.29±1.9	89.87±1.39	v	75.23±1.85	*
ohscal	74.38±1.59	74.69±1.29		63.37±1.54	*
平均	81.70	84.31		71.18	

从表 4-11 可以看出，在大文本数据集下多项式朴素贝叶斯的分类精度平均值依然是最高的，而朴素贝叶斯最差，SMO 的表现不如其他文献中表现得这么好。从统计学角度来看，MNB 在 la1 和 la2 数据集上的表现都显著好于 SMO，在 ohscal 上没有显著差别。

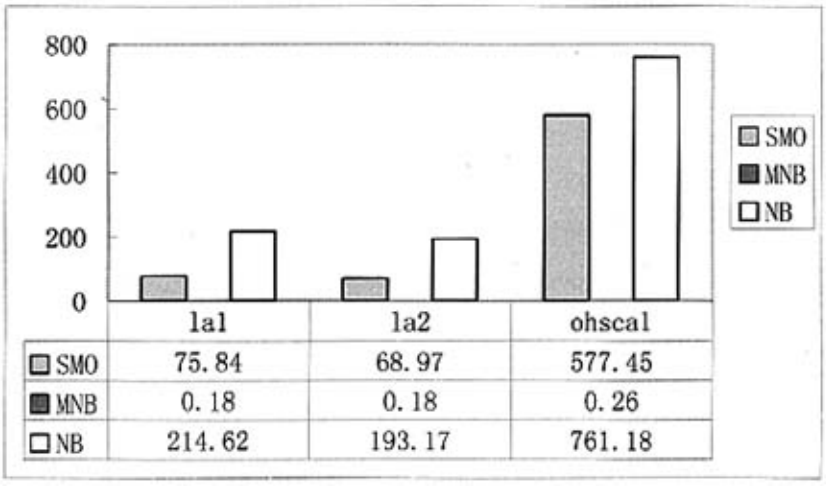
表 4-12 SMO、MNB、NB 在大文本数据集下的排序精度

Table4-12 Ranking Accuracy of SMO、MNB and NB in large text datasets

dataset	SMO	MNB		NB	
la1	94.81±0.7	97.24±0.65	v	91.93±1.26	*
la2	95.27±0.75	97.86±0.53	v	91.72±1.43	*
ohscal	94.38±0.54	95.28±0.65	v	89.85±0.77	*
平均	94.82	96.79		91.17	

表 4-12 列出了三个文本分类算法在大文本数据集下的 AUC 值，从平均值来看 MNB 略好于 SMO，但从统计学角度来看，MNB 全面显著优于 SMO。朴素贝叶斯的排序精度仍不如 SMO 和 MNB。

图 4-3 显示了三种文本算法的需要的训练时间，MNB 并没有受到单词和文档数增加的影响，训练速度非常快，而 SMO 和 NB 受到的影响就大的，NB 需要的时间要多于 SMO。



	la1	la2	ohscal
SMO	75.84	68.97	577.45
MNB	0.18	0.18	0.26
NB	214.62	193.17	761.18

图 4-3 训练时间

Fig4-3 Train Time

4.4 结论

在上面的实验中，我们选择了 15 个数据集对四个主流的文本分类算法 SMO、MNB、NB 和 C4.5 进行了测试，其中有 9 个数据集的类分布是不平衡的数据集；另外用了 3 个大文本数据集对三个文本分类算法 SMO、MNB 和 NB。这些数据集的文本表述方法采用的是空间向量法，文本特征选取方法采用的是词频法，文本以稀疏格式存储。

在 15 个文本数据集的测试中，多项式朴素贝叶斯算法整体表现最好，平均精度高于其他三种文本分类方法，平均排序精度也是四个算法中最高的。决策树算法的预测精度不如多项式朴素贝叶斯，但是略好于支持向量机；朴素贝叶斯的算法的预测精度最差。我们专门讨论了四个分类算法在 9 个不平衡数据集上的分类精度，多项式朴素贝叶斯表现依然是最好的，决策树算法在不平衡文本数据集上

的精度好于朴素贝叶斯但是和 SMO 没有显著差别。值得注意的是在极不平衡的数据集 tr21 上多项式朴素贝叶斯差和朴素贝叶斯都显著差于 SMO 和决策树算法。

在随后的排序精度测试中，从四个文本算法在 15 个数据集上的整体表现来看，MNB 拥有最好的排序效果，因为它在所有的数据集上都不显著差于其他三个分类算法，有点出乎意料的是决策树算法在排序精度上与朴素贝叶斯不相上下，在 15 个数据集的平均排序精度方面虽然略高于朴素贝叶斯，但是从显著性差别方面来判别，决策树在两个数据集上好于朴素贝叶斯，两个数据集上不如朴素贝叶斯，其余的没有显著性差别。SMO 虽然不如 MNB 排序精度高，但是略好于决策树。在 9 个不平衡数据集中，各分类器的表现与整体表现一致。MNB 和 NB 在极不平衡数据集上的排序精度与 SMO 持平，没有显著差别。

在训练时间开销上，决策树算法是最耗时的算法，它的时间开销远大于其他三个分类算法，多项式朴素贝叶斯训练速度非常快，时间开销最小。

最后，我们在三个大文本数据集上测试了 SMO、MNB 和 NB 算法，多项式朴素贝叶斯不论是在精度，排序精度和运行时间上都显著优于 SMO 和 NB 算法。

第五章 总结与展望

5.1 工作总结

信息时代的发展，海量的数据的产生，尤其是海量的电子文本——电子邮件、新闻报道、学术论文、电子病历等不断的累积，为了有效的利用已经存在的和即将产生的信息，文本数据挖掘作为数据挖掘的一个子学科应运而生，而文本分类作为文本数据挖掘的一项基本技术也越来越受到关注。

在经历了复杂而且缺乏灵活性的基于知识工程和专家系统的文本分类模式之后，随着机器学习技术日新月异的发展，成熟的机器学习分类模型被引入到文本分类领域中。支持向量机、贝叶斯、决策树等各种分类模型都基于不同的原理，有各自的特点，如何评价和在应用中选择一个合适的模型成为了一个重要的问题。随着排序问题的提出，原有的文类度量标准无法有效度量排序性能，因此需要新的度量标准的出现。

本文首先概述了几种文本分类算法的实现原理以及一些过去使用和正在使用的分类度量标准。然后介绍了一种新提出的度量标准 AUC，描述了它的评估原料和实现方法。在总结其他人研究的基础上提出了自己设计的实验手段，用新的标准来验证几种常用的文本分类算法，以期为今后的改进打下实验基础并且可以作为选择分类器的指导意见。

本文所做的主要工作是通过选择丰富的基准语料，对几种流行的文本分类算法进行新的评测，尤其关注在新的度量标准下文本分类算法在带有不平衡类分布的文本数据集和大文本数据集中的表现，对这些表现进行评价。

5.2 工作展望

本文所做的工作着重于实验研究，还有需要后续工作需要补充：

1） 文本选用的数据集来源于几个常用的基准语料，采用的特征选取方法是较为简单的词频法，因此在后续工作中可以继续选用些新的基准语料，扩大测试面，同时采用其他特征选取方法进行预处理，从而了解更多分类算法在不同语料条件下的排序性能；

2） 从本文的实验发现，多项式朴素贝叶斯在基于词频法的文本数据集上表现出很强的分类和排序能力，而且速度很快，但是在极不平衡类分布的数据集上

表现有所不足，如何改进使得它适应偏斜类分布的文本数据集是一个研究的方向；

3） 决策树方法在本实验中时间开销最大，因为它在每个叶结点都需要大量计算求熵来选择最佳分裂属性，已有研究者提出了降低决策树算法运算复杂度的方法；另外也有研究者提出了解决决策树方法缺乏概率估计信息问题的方法，如果将这两个改进结合到文本分类中也将是今后研究的一个方向。

参考文献

[1] Hayes, P. and S. Weinstein. Construe-TIS: A System for Content-Based Indexing of a Database of News Stories. In Second Annual Conference on Innovative Applications of Artificial Intelligence. 1991: AAAI Press / MIT Press. p. 49-64.

[2] Mitchell, T., Machine Learning. 1997, New York: McGraw-Hill.

[3]J. J. Rocchio. Relevance feedback in information retrieval. In the Smart Retrieval System -Experiments in Automatic Document Processing. Pages313-323. PrenticeHall, 1971

[4]Mark van Uden, Rocchio: Relevance Feedback in Learning Classification algorithms. http://citeseer. ist. psu. edu/57872. html

[5]Pim van Mum, Text Classification in Information Retrieval Using Winnow. http://citeseer . ist. psu. edu/133034. html

[6] Quinlan J. R., Induction of decision trees, Machine Learning, 1(1):81-106, 1986

[7]Quinlan, J. C4. 5: Programs for Machine Learning. Morgan Kaufmann: San Mateo, CA. 1993

[8] Moulinier, I. A framework for comparing text categorization approaches. In AAAI Spring Symposium on Machine Learning and Information Access. 1996.

[9] Provost F, Domingos P. Tree Induction for Probability-Based Ranking[J]. Machine Learning, 2003, 52(3):199-215.

[10]Jiang Su, Harry Zhang. Learning HyBrid Decision trees for Ranking. http://www. cs. unb. ca/profs/hzhang/

[11]Cover, T.M., & Hart, P.E. (1967). Nearest neighbor pattern classification. Institute of Electrical and Electronics Engineers Transactions on Information Theory, 13, 21-27.

[12]Aha D W, Dennis K, Marc K A. Instance-based Learning Algorithms. Machine learning, 1991; (6): 37-66

[13] Yang Y, Liu X. A Re-examination of Text Categorization Methods[J]. In The 22nd Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval, 1999:42-49.

[14]Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys, 2002, 34(1):1-47.

[15]李程雄，丁月华，文贵华 SVM_KNN 组合改进算法在专利文本分类中的应用，计算机工程与应用，第 20 期，2006

[16] Susana Eyheramendy. Bayesian text categorization. Ph. D. Rutgers University, New Brunswick 2004

[17] McCallum A, Nigam K. A Comparison of Event Models for Naive Bayes Text Classification[A]. AAAI-98 Workshop on Learning for Text Categorization[C]. Madison, Wisconsin:AAAI Press, 1998: 509-516.

[18]宫秀军，史忠植，基于Bayes潜在语义模型的半监督Web挖掘软件学报Vol. 13, No. 8, 2002

[19]靳小波，夏清国，基于Lee模型的文本分类，计算机工程，第32卷第2期，2006. 1

[20] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[21] Joachims T. Text Categorization with Suport Vector Machines: Learning with Many Relevant Features[J]. Proceedings of the 10th European Conference on Machine Learning.

1998: 137 - 142.
[22] Tong, S. and D. Koller, Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, 2001.2: p. 45-66.
[23] Rennie J D M, Rifkin R. Improving Multiclass Text Classification with the Support Vector Machine[D]. Master's thesis, Massachusetts Institute of Technology, 2001.
[24]Provost, F. and Fawcett, T.: Analysis and Visualization of Classifier Performance:. Comparison under Imprecise Class and Cost Distributions[C], Proc of 3rd Int Conf on Knowledge Discovery and Data Mining. California, 1997:43-48.
[25] David J. Hand, Robert J. Till: A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning[J], 2001, 45(2): 171-186
[26] Ataman K, W Nick Street. Optimizing area under the ROC curve using ranking SVMs[C]. In KDD-05, 2005.
[27]FERRI C, FLACH P. Learning Decision Trees Using the Area Under the ROC Curve[J]. ICML, 2002, 65(10):78-81
[28] Jiang Su, Harry Zhang: Probabilistic Inference Trees for Classification and Ranking[C]. Canadian Conference on AI, 2006: 526-537.
[29] H. Zhang, L. Jiang and J. Su, Augmenting Naive Bayes for Ranking[A], Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)[C], ACM, 2005:pp. 1025-1032.
[30] Alex K. S. Wong, John W. T. Lee, Daniel S. Yeung. Improving Text Classifier Performance based on AUC[J]. ICPR, 2006, (3): 268-271.
[31] Platt J. Sequential minimal optimization: a fast algorithm for training support vector machines. Technical Report, MSR-TR-98-14, Microsoft Research, 1998.
[32] Platt J. Fast Training of Support Vector Machines using Sequential Minimal Optimization[A]. In: Schoelkopf B, Burges CJC, SmolaAJ eds. Advances in Kernel Methods - Support Vector Learning. Cambridge, MA: MIT Press, 1999:185-208.
[33] G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison Wesley, 1989.
[34] Joachims T. Making large-scale SVM learning practical[C]. In: BScholkopf, C J C Burges, A J Smola eds. Advances in Kernel Methods-Support Vector Learning, MIT Press, 1999: 169~184
[35] U. Brefeld and T. Scheffer. Auc maximizing support vector learning. In Preceedings of ICML 2005 workshop on ROC Analysis in Machine Learning, 2005.
[36] http://www.cs.waikato.ac.nz/ml/weka/
[37] J. Su and H. Zhang. A Fast Decision Tree Learning Algorithm. Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06) , AAAI Press(2006): 500-505.
[38] Huang Jin, Ling C X. Using AUC and Accuracy in Evaluating Learning Algorithms[J]. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2005, 17(3):299-310.
[39] C. X. Ling, J. Huang and H. Zhang, AUC: a statistically consistent and more discriminating measure than accuracy, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI2003), pp. 519-526, Morgan Kaufmann(2003).

[40] C. Hsu and C. Lin. A comparison on methods for multiclass support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.
[41] Huang J, Jingjing L, Charles X. Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. 3rd IEEE International Conference on Data Mining (ICDM 2003), 2003 Nov 19-22: Melbourne (FL).
[42] Charles X. Ling, Robert J. Yan .Decision Tree with Better Ranking. Proceedings of th Twentieth International Conference on Machine Learing(ICML-2003), Washington DC, 2003
[43] Bradley A P. The use of the area under the ROC curve in the evaluation of machine learning algorithms[J]. Pattern Recognition, 1997, 30:1145-1159.
[44] Forman G, Cohen I. Learning from little: Comparison of classifiers given little training. In: Jean FB, Floriana E, Fosca G, Dino P, eds. Proc. of the 8th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD-04). Pisa: Springer-Verlag, 2004. 161-172.
[45] HAN E H, KARYPIS G. Centroid-based document classification: Analysis & experimental results[A]. Technical Report 00-017, Computer Science[R]. University of Minnesota, 2000.
[46] Lewis, D. D. and M. Ringuette. A comparison of two learning algorithms for text categorization. in Proceeding of the Third Annual Symposium on Document Analysis and Information Retrieval. 1994. Las Vegas, US. p. 81-93.
[47] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In SIGIR-94, pages 192 - 201, 1994.
[48] TREC. Text REtrieval conference. http://trec.nist.gov.
[49] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. http://www.research.att.com/lewis, 1999.
[50] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the world wide web using WebACE. AI Review (accepted for publication), 1999.
[51] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-based clustering for web document categorization. Decision Support Systems (accepted for publication), 1999.
[52] E.H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploartion. In Proc. of the 2nd International Conference on Autonomous Agents, May 1998.
[53] J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. Web page categorization and feature selection using association rule and principal component clustering. In 7th Workshop on Information Technologies and Systems, Dec. 1997.
[54] M. F. Porter. An algorithm for suffix stripping. Program, 14(3):130 - 137, 1980.

致谢

研究生生活即将结束之际，首先我要对我的导师——苏德富教授表示最诚挚的感谢，感谢他多年来对我的关心、鼓励和帮助。几年来我所取得的每一点进步和提高都得益于他的言传身教。这篇论文从选题到确定研究方法，到最后的修改定稿，苏老师都给了我悉心的指导和帮助。导师渊博的知识、严谨的治学态度、实事求是的精神是我一生中宝贵的财富。

感谢钟诚教授提供了良好的实验环境以及他和宋玲副教授、唐天兵副教授、刘连芳研究员在预答辩时对论文提出的宝贵意见。

感谢李陶深教授、苏一丹教授、霍玲教授、严毅老师和黄文玲老师的教导和鼓励。

感谢我的父母，他们一直以来都在牺牲自己的一切来支持我的学习研究工作，让我时时感到身后亲情的温暖。

感谢广西大学、研究生院和计算机与电子信息学院为我提供了良好的学习环境。

攻读硕士学位期间发表的学术论文

[1] 杨挚诚，苏德富．文本分类算法比较．计算机技术与发展，录用.

基于机器学习的文本分类算法研究

作者：　　杨挚诚
学位授予单位：　　广西大学

相似文献(5条)

1.学位论文 张志华 中文文本分类算法的研究与实现 2004

文本分类最初是应文本信息检索的要求出现的,但是随着文本数据的激增,传统的研究方法已经不适合大规模文本分类,文本数据挖掘应运而生.作为文本数据挖掘的一个重要功能,文本分类技术日益成为研究热点.文本分类目的是对文本集有序组织,便于文本信息高效管理,为人的决策提供支持.但是传统的人工分类的做法存在许多弊端,不仅是耗费大量人力、物力和精力,而且受人为因素影响较大,分类结果一致性不高.与之相比,文本自动分类具有快速、高效的特点,且分类准确率较高.该文主要研究中文文本分类算法及其实现,采用基于关联分析和聚类分析两种方法,设计和开发了中文文本分类系统ACTC,实现中文文本分类功能,在理论和实践上论证两种方法的可行性和正确性.系统用向量空间模型表示中文文本,采用基于统计的文本分类模型.该文从理论和应用角度对现有文本分类算法进行了较为深入的研究,提出一种新的关联分类算法频繁模式增长PFP_Growth算法,并基于信息粒度原理将聚类算法用于中文文本分类.ACTC系统能够快速高效对大规模中文文本分类,具有良好的自适应性的和可扩充性,而且为研究文本分类算法提供了一个的实验平台.

2.学位论文 符燕华 Web文本数据挖掘研究 2006

随着Internet技术的日益成熟，尤其是万维网的普及化，使得数据挖掘技术的研究重点已经从传统的基于结构化数据库的应用转移到基于Web的应用上来。Web文本挖掘采用数据挖掘、机器学习、自然语言处理、信息检索和知识管理等领域的技术来处理和分析非结构或半结构化的文本，从中提取有价值的知识。文本分类是文本挖掘的一个主要研究分支，本文主要研究的是基于关联规则的文本分类方法。　　目前主要的文本分类方法有：最近邻分类、贝叶斯分类、决策树、支持向量机、向量空间模型、回归模型和神经网络等。本文通过分析现有的文本分类方法，提出了基于关联规则的文本分类算法。首先，本文介绍了数据挖掘的概念及方法，Web挖掘的主要方法；接着，介绍了关联规则的主要算法；然后介绍了文本数据挖掘中使用的主要技术，如文本的特征表示，文本的特征项提取，文本分类的方法；最后，提出了基于关联规则的文本分类算法，并通过实验，验证了方法的可行性。
由于关联规则频繁集的查找速度及特征项的提取对分类器的性能都起着及其重要的作用，所以，在今后的工作中，要加强频繁集的查找速度，选择更好的特征提取方法以提高分类器的性能。

3.学位论文 王俊英 基于科技文献的中文文本分类算法研究 2007

文本分类最初是应文本信息检索的要求出现的，但是随着文本数据的激增，传统的分类研究方法已经不适合大规模文本分类，于是文本数据挖掘应运而生。作为文本数据挖掘的一个重要功能，文本分类技术日益成为研究热点。　　科技文献的行文和格式都有规范的特点，但其科技文献的自动分类问题却没有得到足够的关注；与此同时，科技文献分类问题的需求却与日俱增。针对这一现实需求，本文以计算机类科技文献为例，对科技文献的分类问题进行了深入研究。　　首先，对中文文本分类算法进行了深入研究，从分类算法的应用和分类效果角度出发，分析了各个算法的分类思想、文本预处理方法、特征项的选择和特征提取方法以及算法实现关键步骤等，并提出了评价和分析几个分类算法的定理和方法。　　其次，分析了科技文献的行文规范特点，提出了关键词抽取算法。科技文献的标题、关键词和摘要部分很精简的反映了文章的核心内容，同时与文档主题内容不相关的描述很少，算法直接从该部分内容抽取关键词集，取代了传统文本分类算法的中文分词。　　然后，提出了一种基于科技文献的文本分类算法，实现了对计算机类科技文献的层次化分类。应用科技文献自身明显的层次关系结构特点，抽取各个类别文档的关键词集，构建层次化分类模型，有效地提高了科技文献的分类精度。　　实验结果充分表明，所提出的层次化分类算法的分类效果明显优于传统的平面化分类算法，有更高的准确率和查全率。

4.学位论文 李明 面向地理信息系统的文本数据挖掘系统的研究与实现 2004

该课题来自国家"863"项目《多源空间数据挖掘技术》.数据挖掘是致力于数据分析和理解、揭示数据内部蕴藏知识的技术,它成为未来信息技术应用的重要目标之一.随着INTERNET的大规模普及和企业信息化程度的提高,有越来越多的信息积累,其中绝大部分是以文本形式存在的.人们急需一种能从大规模的文本信息资源中提取符合需要的,简洁的,可靠性高的信息的工具,数据挖掘中的文本数据挖掘正是解决这个问题的方向.同时近年来,地理信息系统无论是在理论上还是应用上都处于一个飞速发展的阶段,并成为一个跨学科、多方向的研究领域.把文本数据挖掘和地理信息系统结合起来,面向地理信息领域提取地理信息系统需要的与地理相关的信息正是该文讨论的问题.该文主要讨论了面向地理信息系统的文本数据挖掘系统的研究与实现.对文本数据挖掘中的文本分类和文本信息提取分别进行了分析,进而提出了由分类关键词形成分类规则的文本分类算法设计、由人工经验规则与特征词典提取所需信息的文本信息提取算法设计,并实现了这两个算法,构造了一个文本数据挖掘子系统.通过4000多篇10类文本数据对文本数据挖掘子系统的测试,发现文本分类和文本信息提取均达到了符合预定要求的效果,实现了课题研究的初步目的.

5.学位论文 李永波 基于数据挖掘的军事情报分析系统研究 2005

文本挖掘（Text Mining,简称TM）,也称作文本数据挖掘(Text Data Mining),通俗地说，就是从文本或大量文本的集合中发现有用信息和知识过程。　　随着新军事革命进一步深化，世界各国为寻求军事制高点，下大力建设信息化军队。军队信息化建设的核心就是军事信息系统的建设，而军事情报分析系统则是信息系统的重要组成部分，如何提高情报分析效益，缩短指挥信息环路周期，提高信息系统的辅助决策效益，成为信息化建设中重中之重的问题。　　目前，各级情报侦察部门收集汇总的大量分析报告主要采用传统的处理方式进行分析利用。特别是在定量分析中主要采用IR和IE 两种技术，制约了情报分析的发展。本文在军事情报分析中引入文本挖掘技术，并利用ACTC作为分类工具，采用关联分析分类方法，对大量的中文文本情报，进行了分类，明显提高了情报分析效益。　　运用数据挖掘技术对情报进行分析处理，将有利于提高情报分析系统的效率和精度。本论文主要就以下问题进行了研究。　　1）文本挖掘的概念、过程、研究课题、应用领域，及其与DB、IR、IE的区别与联系。　　2）基于统计模型的常见文本分类技术，包括文本的向量空间模型〔VSM〕表示、文本预处理，文本分类算法和分类评估标准。　　3）军事情报分析系统的构成及功能，及其在军事指挥系统中的位置。　　4）基于数据挖掘的军事情报分析系统的总体设计，系统需求，结构和处理流程，文本训练模块和文本分类模块的设计框架，最后用实验数据佐证论文观点。　　本文的研究工作与研究成果对于中文文本挖掘技术研究有一定的理论意义；对于改进情报分析系统结构，建立具有可伸展性和自适应性的情报分析系统具有重要的参考价值；为基于数据挖掘的军事情报分析系统的研究与开发提供了一条可行的途径。