基于半监督与集成学习的文本分类方法

唐焕玲 ◎ 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

基于半监督与集成学习的文本分类方法

唐焕玲　著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

文本分类技术广泛应用于新闻媒体、网络期刊文献、数字图书馆、互联网等领域，是人类处理海量文本信息的重要手段。

本书重点探讨了利用信息论中的评估函数量化特征权值的方法；基于权值调整改进 Co-training 的算法；利用互信息或 CHI 统计量构造特征独立模型，进行特征子集划分的方法；基于投票熵维护样本权重的 BoostVE 分类模型；融合半监督学习和集成学习的 SemiBoost-CR 分类模型。

其中特征选择和权值调整方法、基于特征独立模型划分特征子集的方法适用于文本分类，其他算法不仅适用于文本分类，对机器学习和数据挖掘的其他研究也有较大的参考价值和借鉴作用。

本书适合研究方向为文本挖掘、机器学习的硕士、博士研究生及相关专业技术人员学习和参考。

图书在版编目（CIP）数据

基于半监督与集成学习的文本分类方法 / 唐焕玲著. —北京：电子工业出版社，2013.8
ISBN 978-7-121-21256-7

Ⅰ. ①基… Ⅱ. ①唐… Ⅲ. ①文字处理—研究 Ⅳ. ①TP391.1

中国版本图书馆 CIP 数据核字（2013）第 188126 号

责任编辑：张 京
文字编辑：薄 宇
印 刷：三河市鑫金马印装有限公司
装 订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：900×1 280 1/32 印张：5.875 字数：205 千字
印 次：2013 年 8 月第 1 次印刷
定 价：29.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

文本分类（Text/Document Categorization）是指按照预先定义的主题类别，通过一定的学习机制，在对带有类别标签的训练文本进行学习的基础上，给未知文本分配一个或多个类别标签的过程。文本分类技术广泛应用于新闻媒体、网络期刊文献、数字图书馆、互联网等领域，是人类处理海量文本信息的重要手段。数据挖掘技术在信息检索、邮件过滤、Web 个性化服务等领域的成功应用均在一定程度上依赖于准确的文本分类技术。因此，文本分类技术的相关研究一直是近年来国际学术界的研究热点。

本书对文本分类的关键技术进行了概述，阐述了基于半监督学习和集成学习的国内外相关研究，重点对基于半监督学习和集成学习的文本分类方法进行了深入探讨。

本书的第 1 章介绍了研究背景、文本分类及其面临的问题，阐述了基于半监督学习和集成学习的文本分类方法的研究意义和国内外研究现状。第 2 章对文本分类的关键技术进行了概述，主要包括文本预处理、文本的表示、特征选择、分本分类方法、实验数据集及分类模型的评估方法。第 3 章分析了特征选择存在的问题，采用信息论中的评估函数量化特征的重要性，调整特征的权值，提出 TEF-WA 权值调整技术；分析比较了文档频率、信息增益（Information Gain，IG）、期望交叉熵（Expected cross Entropy）、互信息（Mutual Information，MI）、χ^2 统计量（CHI）、文本证据权（Weight of Evidence for Text，WET）和几率比（Odds Ratio）等多种评估函数及实验结果。第 4 章分析了半监督学习中的代表方法 Co-training 算法，提出了利用 TEF-WA 技术对 Co-training 改进的算法 TV-SC 和 TV-DC，通过评估两个基分类器之间的差异性，可间接评估两个特征视图的独立性，并通过实验证明了所提方法的有效性。第 5 章针对 Co-training 方法的独立性假设问题，提出

了利用互信息（MI）或 CHI 统计量评估特征之间的相互独立性的方法，构造了一种特征独立模型（MID-Model）。基于该模型提出了特征子集划分方法——PMID 算法，以便把不存在自然划分的一个特征集合划分成两个独立性较强的子集，进而提出了改进的半监督分类算法——SC-PMID 算法。并且对由 PMID 算法划分得到的两个特征子集之间的独立性进行了理论论证。第 6 章分析了集成学习算法 AdaBoost 算法不能有效提升 Naïve Bayesian 分类器的原因，提出了基于投票信息熵和多视图的 AdaBoost 改进算法——BoostVE 算法，采用基于投票信息熵的样本权重维护新策略，能有效提高 Naïve Bayesian 文本分类器的泛化能力。理论分析证明改进的 BoostVE 算法的最小训练错误上界优于 AdaBoost。第 7 章基于半监督学习和集成学习，提出了置信度重取样的 SemiBoost-CR 分类模型，给出了基于最大差距和基于相似近邻两种置信度计算方法。实验表明利用少量标注样本和大量未标注样本，SemiBoost-CR 分类模型能够明显提升 Naïve Bayesian 文本分类器的性能指标。第 8 章介绍了采用 VC++ 6.0 实现的中英文文本分类系统 SECTCS，阐述了 SECTCS 系统的原有的功能与新扩展的功能、总体结构、主要的用户界面及操作。

本书的研究工作得到了山东省高校智能信息处理重点实验室（山东工商学院）、国家自然科学基金项目（No.61073133，No.61175053，No.61272369，No.61272244）及山东省优秀中青年科学家科研奖励基金计划项目（S2010DX021）的资助，特此表示感谢。

唐焕玲

2013 年 3 月

目　录

第1章 绪 论

❑1.1 研究背景及意义

1.1.1 数据挖掘和文本挖掘

随着信息技术和网络技术的迅速发展，网络数据规模呈指数增长，Internet 已发展成站点遍布全球的巨大信息服务网络，包含了涉及许多领域的丰富的信息资源。面对内容异构的海量信息，传统的数据分析方法只能获得数据的表层信息，无法获得数据属性的内在关系和隐含的信息，难以适应需求的不断发展。数据挖掘和知识发现（Data Mining & Knowledge Discovery in Database，DM&KDD）是 20 世纪 90 年代兴起的一门信息技术领域的前沿技术，它是在数据和数据库急剧增长远远超过人们对数据处理和理解能力的背景下产生的。

数据挖掘（Data Mining，DM）是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中采掘出隐含的、先前未知的、对决策有潜在价值的知识和规则[1]。知识发现（Knowledge Discovery in Databases，KDD）指识别出存在于数据库中有效的、新颖的、具有潜在效用的、最终可理解的模式的非平凡过程[2]。数据挖掘是一个交叉学科领域，受多个学科的影响，包括数据库系统、统计学、机器学习、可视化和信息科学等。此外，依赖于所用的数据挖掘方法及可使用的其他学科的技术，如神经网络、粗糙集理论、知识表示、归纳逻辑程序设计或高性能计算。依赖于所挖掘的数据类型或给

定的数据挖掘应用，数据挖掘技术也可能集成空间数据、信息检索、模式识别、图像分析、信号处理、计算机图形学、Web 技术、经济、商业、生物信息学或心理学领域的技术[1]。

传统的数据挖掘技术，主要针对的是结构数据，如关系的、事务的、数据仓库的数据。随着数据处理工具、先进数据库技术及网络技术的迅速发展，大量的形式各异的复杂类型的数据（如结构化与半结构化数据、超文本与多媒体数据）不断涌现。因此数据挖掘面临的一个重要课题就是针对复杂数据的挖掘，这包括复杂对象、空间数据、多媒体数据、时间序列数据、文本数据和 Web 数据。

文本挖掘是数据挖掘领域的一个分支，在国际上，文本挖掘是一个非常活跃的研究领域。从技术上说，它实际是数据挖掘和信息检索两门学科的交叉。文本挖掘与传统数据挖掘的差别在于文本数据与一般数据的巨大差异。传统数据挖掘所处理的数据是结构化的，如关系的、事务的、数据仓库的数据。其特征数通常不超过几百个，而文本数据没有结构，转换为特征矢量后特征数将达到几万甚至几十万。所以，文本挖掘既采用了很多传统数据挖掘的技术，又有自己的特性[5-14]。

近年来随着 Internet 的大规模普及和企业信息化程度的提高，信息积累越来越多，Internet 已经发展为当今世界上最大的信息库。Internet 上的信息，绝大多数是以网页形式存放的，而网页的内容又多以文本方式来表示，传统的信息检索技术已不适应日益增长的大量文本数据处理的需要。如何快速、准确地从来自异构数据源的大规模的文本信息资源中提取符合需要的简洁、精炼、可理解的知识，就要用到文本知识挖掘。Internet 的发展极大地促进了文本挖掘的发展。

文本挖掘（Text Mining，TM）：以计算语言学、统计数理分析为理论基础，结合机器学习和信息检索技术从文本数据中发现和提取独立于用户信息需求的文本集中的隐含知识。它是一个从文本信息描述到选取提取模式，最终形成用户可理解的信息知识的过程[6]。根据 KDD 的框架，结合文本挖掘的定义和特点，文本挖掘的过程示意图如图 1.1 所示。开始处是原始文本信息源，最终结果是用户获得的知识模式，经历了信息预处理→文本挖掘→质量评价三个主要阶段。

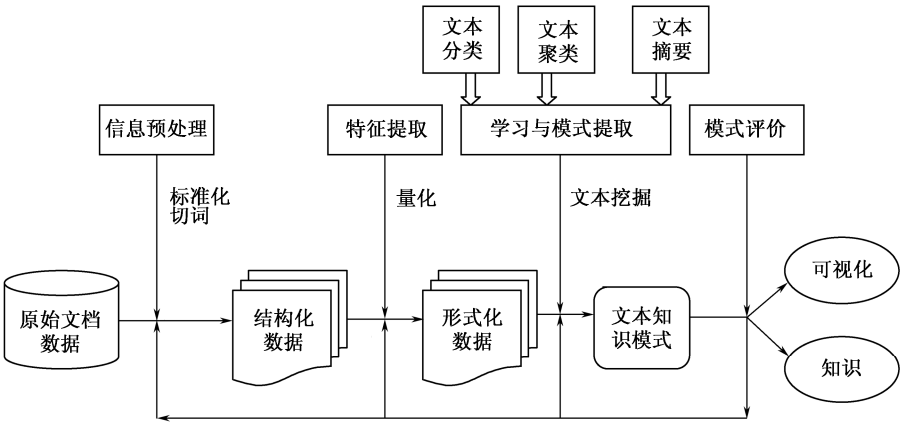


图 1.1　文本挖掘的过程示意图

1.1.2　文本分类及其面临的问题

文本分类（Text Categorization，TC）是文本挖掘中最重要的研究领域之一。对文本进行准确、高效的分类是许多数据管理任务的重要组成部分。数据挖掘技术在信息检索、邮件过滤和提供个性化的服务等方面，均在一定程度上依赖于准确的文本分类技术。

1．文本分类的定义

所谓文本分类，是指按照预先定义的主题类别 $C=\{c_1, c_2,\cdots, c_L\}$，通过一定的学习机制，在对带有类别标签的训练文本 $D=\{(x_1,y_1),\cdots,(x_N,y_N)\}$ 进行学习的基础上，给未知文本分配一个或多个类别的过程。其中 C 可以是并列的也可以分层次组织起来的。可以用一个目标函数 $\phi: D\times C\to\{T,F\}$ 来描述文本分类[1]，称 $\phi: D\times C\to\{T,F\}$ 为分类规则或假设或模型，对 $x_i\in D, c_j\in C$，$(x_i,c_j)\to T$ 表示 x_i 属于类别 c_j；而 $(x_i,c_j)\to F$ 表示 x_i 不属于类别 c_j。

这里，文本既可以是传统的纯文本，也可以是经过 HTML Parser 等网页

解析工具去掉网页标记、转换成纯文本的网页（Web 文档），网页可以看做是特殊的文本。Web 上的信息资源大多以 HTML 页面或 XML 页面的形式存在，与一般文本的表示不同，网上大量半结构化的文本及其之间的超链接提供了多于传统文本的有用信息，如标题、段落标题、超链接文字、链接及所用的字号等辅助信息为文本分类提供了更多的有用信息。根据 Web 网页的具体特性，一般可以从两个方面来选取网页特征项：① 通过提取网页内容中的关键词；② 利用网页中的有关标识符及其结构特征。Web 网页经过 HTML Parser 等网页解析工具，去掉网页标记，可转换成纯文本。

文本分类的过程一般包括文本预处理、特征约简、训练与分类、分类结果的评价和反馈等过程，如图 1.2 所示。

本书研究的基于半监督学习和集成学习的文本自动分类方法，既适用于纯文本的分类，也可应用于网页的分类。在后续的章节，没有特别说明时，文本泛指纯文本和经过预处理后的 Web 文档。

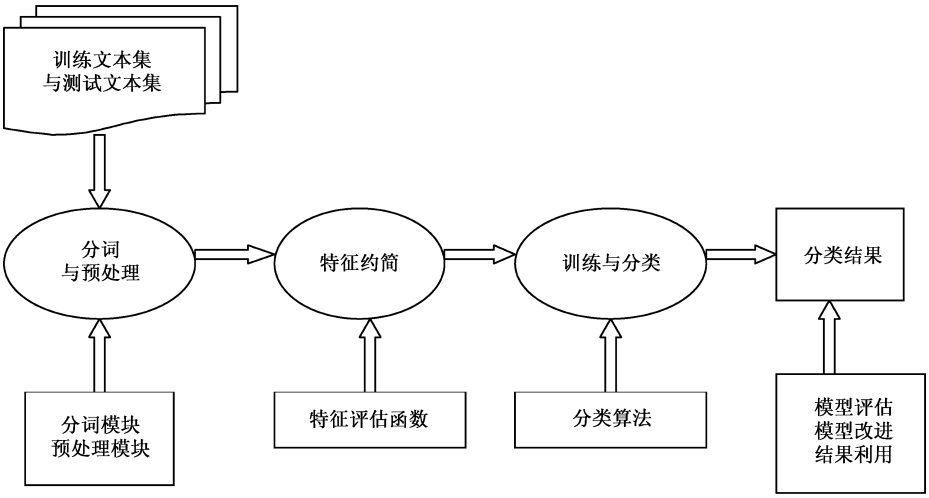


图 1.2　文本分类的一般过程

2．文本分类的发展

文本分类是信息检索与数据挖掘领域的研究热点。1960 年 Maron 在 Journal of ASM 上发表了有关自动分类的第一篇论文，标志着文本分类技术

的诞生[8]，而 H. P. Luhn 在这一领域进行了许多开创性的研究工作。随后许多著名的情报学家如 K. Sparch、G. Salton 及 R. M. Needham 等都在这一领域进行了卓有成效的研究[9-11]。

文本自动分类在国外大体经历了三个发展阶段：

第一阶段（1960—1964），主要进行文本自动分类的可行性研究；

第二阶段（1965—1974），进行自动分类的实验研究；

第三阶段（1975 至今），自动分类进入实用化阶段，新方法和新系统层出不穷。

我国的自动分类工作也经历了这三个阶段，只是起步较晚。1981 年侯汉清先生在国内首次对文本自动分类进行探讨，此后国内一些科研院所也相继开展了文本分类研究，在分类理论和应用特别是中文文本分类方面取得了众多成果。2007 年侯汉清教授承担的国家社科基金项目“基于知识库的网页自动标引和自动分类研究”结题。

目前国外开展文本自动分类研究比较著名的机构包括卡耐基梅隆大学（CMU）、麻省理工学院（MIT）、加州大学伯克利分校、康奈尔大学、马里兰大学、微软剑桥研究院、微软亚洲研究院、IBM 研究中心、卡耐基集团等。国内比较活跃的单位有清华大学、北京大学、复旦大学、上海交通大学、哈尔滨工业大学、东北大学、北京邮电大学、中国科学院（计算所、软件所、计算机语言信息工程研究中心）、南京大学、纳讯科技公司、西风网站等。此外国内外还有大批研究机构和公司也在从事同类研究。

从文本分类使用的方法上说，主要有：① 20 世纪 80 年代基于知识工程和专家系统的文本分类模式；② 20 世纪 90 年代逐渐成熟的基于机器学习的文本分类方法，更注重分类器的模型自动挖掘和生成及动态优化能力，在分类效果和灵活性上都比之前基于知识工程和专家系统的文本分类模式有所突破，成为相关领域研究和应用的经典范例。

文本分类的主要方法包括决策树（Decision Tree）、K 近邻算法（K Nearer Neighbors Classifier）、线性分类器（Linear Classifier）、回归模式（Regression Models）、简单 Bayesian 网络（Bayesian belief Networks）、规则学习算法（Rule Learning Algorithms ）、BP 神经网络（BP Neural Networks）、归纳学习技术（Inductive Learning Techniques）、支持向量机（Support Vector Machine，SVM）等[15-33]。20 世纪 90 年代出现了基于集成学习的分类方法，即组合多个分类

器以克服单个分类器的不足，有效提高了分类的精度，已成为一个研究热点，Boosting 和 Bagging 方法是其中的代表算法[34-38]。

3．文本分类面临的问题

文本分类技术能够较好地解决大部分具有数据量相对较小、标注比较完整及数据分布相对均匀等特点的应用问题，但是对大规模应用仍受到很多问题的困扰，目前面临着诸多挑战[33]，本书主要讨论以下两点。

（1）标注瓶颈问题

传统的有监督分类算法（Supervised Categorization Algorithm）需要提供足够的已标注训练样本（Labeled Data），但是已标注的训练样本集的建立需要专家知识，费时费力，代价高，获取困难，制约了分类模型的建立，致使许多实际问题的研究无法开展，这就是所谓的标注瓶颈问题。然而，互联网上存在大量未标注样本（Unlabeled Data），获取相对比较容易。因此，利用大量的未标注样本结合少量的标注样本的半监督分类（Semi-supervised Categorization）的研究引起了学术界比较广泛的关注。

针对标注瓶颈问题，基于半监督学习（Semi-supervised Learning）的分类方法是比较有效的解决方法[39-41] [47-56]。半监督学习在文本分类、图像分类、邮件过滤、机器翻译、主题词识别、词性标注等方面都有广泛的应用。基于半监督学习的分类称为半监督分类（Semi-supervised Categorization 或 Semi-supervised Classification），其研究重点在于使用大量的未标注（Unlabeled）样本，结合少量的标注（Labeled）样本训练生成分类器，提高分类器的性能指标。

半监督分类的研究虽然已经取得了不少成果，但是也存在许多值得探讨和亟待解决的问题。例如，半监督分类算法的应用存在一定的约束条件；半监督分类的精度还有待提高；半监督分类算法往往要付出大量的迭代代价，计算复杂度比较高等。如何提高半监督分类的精度、降低计算复杂度，是值得研究的问题。基于半监督学习的文本分类方法的研究，在理论和实践上都是比较有意义的研究方向。

（2）分类方法本身存在局限性

分类技术在其发展过程中出现了许多经典的分类方法，如决策树方法、Naïve Bayesian 学习方法、神经网络（Netware Net）、K 近邻法（K Nearest Neighbor）、支持向量机（Support Vector Machine，SVM）等，由于受到分类方法本身的局限性，这些经典方法的性能指标在原有基础上很难进一步提高[3] [15-33]。因此，基于集成学习的分类方法，即组合多分类器来提高分类的精度成为学术界比较关注的另一个研究方向。

集成学习（Ensemble Learning）是一种机器学习范式，多个学习器的单独决策被以某种方式组合起来（通过加权或无权重投票）解决同一个问题[34-38]。集成学习技术已经在行星探测、地震波分析、Web 信息过滤、生物特征识别、计算机辅助医疗诊断等众多领域得到了广泛的应用。在文本挖掘领域，集成学习技术可用于文本分类、文本过滤等；在网络挖掘方面，集成学习技术在网页分类、信息检索和网络用户行为分析——偏好排序方面都有应用。由于集成学习技术可以有效地提高学习系统的泛化能力，因此成为国际机器学习界的研究热点，并被国际权威 T. G. Dietterich 称为当前机器学习四大研究方向之首。

鉴于文本分类面临的这两种挑战，利用少量标注样本（Labeled Data）和大量的未标注样本（Unlabeled Data）的半监督分类（Semi-supervised Categorization）和组合多个分类器来提高分类性能的集成学习（Ensemble Categorization）是近年来模式识别和机器学习领域的研究热点，也是本书研究的主要内容。

❑1.2　国内外相关研究

1.2.1　半监督学习

半监督学习是模式识别和机器学习研究领域的热点之一，在文本分类、图像分类、邮件过滤、机器翻译、主题词识别、词性标注等方面得到了广泛的应用。根据半监督分类算法的实现方式，现有的典型方法大致分为五种：

基于生成模型（Generative Model）的半监督分类方法[39-43]、自训练方法（Self-Training）[44-46]、协同训练方法（Co-Training）[47-56]、基于图的半监督分类方法[57-59]、直推式支持向量机方法（Transductive SVM，TSVM）[60-64]等。

1．EM 算法

Dempster、Laird 和 Rubin 提出的 EM（Expectation-Maximization）算法[40]是一种对不完整数据进行最大化参数估计的迭代算法。结合标注文本和未标注文本的信息进行半监督学习，未标注文本的类别可以看成缺失的值。Nigam 结合 EM 算法和 Naïve Bayesian 算法，从标注文本和未标注文本中学习，改进了 Bayesian 分类器的分类效果[41-42]，是基于生成模型（Generative Model）的半监督分类方法的典型代表。

初始时，只使用标注样本集，建立 Naïve Bayesian 分类器，然后重复交替执行 E 步骤和 M 步骤，达到使似然函数增加的目的[41]。直观地，EM 试图在未标注样本的分布上建立最大可能的分类假设，EM 算法可以看成未标注样本在初始的标注训练样本“周围”的聚类。

标注文本集和未标注文本集组成混合模型（Mixture Model），当标注文本集的数目远远小于未标注文本集时，EM 的参数估计在很大程度上来源于未标注文本集。当未标注文本集上的无监督聚类学习产生的类别与标注文本集的类别不一致时，反而会降低分类的正确性。EM-λ 算法是 Nigam 在基本的 EM 算法上，引入了一个参数$\lambda(0 \leqslant \lambda \leqslant 1)$，以调整未标注文本在 EM 算法中的权重[39][42]。

2．Self-training 算法

Self-training 首先使用由少量的标注样本训练生成分类器对未标注样本分类，选出置信度高的未标注样本，加上预测的类标记，添加到训练样本集中重新训练，迭代这个过程。Self-training 又称 Self-teaching 或 Bootstraping。为了避免一个分类器强化自己的错误，通常当置信度低于某个阈值时就停止迭代，或者使用 Co-training 方法。Self-training 在主题名词的识别[44]、emotional 和 non-emotional 的对话分类[45]、图像检测[46]等应用领域取得了比较好的效果。

3．Co-training 算法

Co-training 算法[47-49]最早由 Blum 和 Mitchell 提出，假设数据集可以被自然地分成两个独立的特征子集（视图），每个子集都包含足够的信息进行分类学习，在每个视图上建立各自的分类器，两个分类器每次互相标记一部分置信度高的样本给对方，重新训练，迭代直到没有更多适合的未标注样本加入。Blum 和 Mitchell 将 Web 文本集分成两个视图，视图 V_1 由 Web 网页上的单词组成，另一个视图 V_2 由指向其他网页的超链接的单词组成。任意样本 x 可以用一个三元组 (x_1,x_2,c) 表示，这里，x_1 和 x_2 是 x 在两个视图中的描述，c 是它的类别。Co-training 算法要求两个特征视图满足假设：① 一致性（Compatible），② 独立性（Uncorrelated）。前者表示，样本的分布与分类的目标函数是一致的，也就是说，对大多数样本 x：$f^1(x_1)=f^2(x_2)=f(x)$，目标函数在每个特征子集上预测的类别是完全相同的。后者表示对指定类别的任意的样本 (x_1,x_2,c)，$P(x_1 \mid f(x),x_2)=P(x_1 \mid f(x))$，也就是说，样本 x_1 和 x_2 在两个视图中的描述是独立的[47-49] [56]。

实际上由于多种原因，这两个假设并不能完全严格地满足，尤其是独立性，甚至在许多实际应用中不存在自然划分且满足这种假设的两个视图。Goldman 和 Zhou[50]使用两个不同的学习器在同一个特征视图进行共同训练，Zhou 和 Goldman 提出了在单个特征视图上多个分类器的 Democratic Co-learning 算法，如果大多数分类器给未标注样本 x_u 的分类一致，则把 x_u 连同它的标注一起加入训练样本集，所有分类器在更新的训练样本集上重新训练[51]。类似地，周志华提出了使用三个分类器的 Tri-training 算法[54]。Muslea 提出了多视图（Multi-views）与主动学习（Active Learning）相结合的 Co-EM 算法、CO-EMT 算法[56]，也可以看成是 Co-training 与 EM 相结合的算法。

4．直推式支持向量机

直推式支持向量机（Transductive SVM，TSVM）[57-61]使得分类器首先通过对已标注样本的学习，仅对当前的少量未知样本进行误差最小的预测，而且暂不考虑对未来所有实例预期性能的最优性，将这些样本加入到学习过程中来，以改进分类器的效果。由于成功地把未标注样本中所隐含的分布信息引入了支持向量机的学习过程中，TSVM 算法比单纯使用有标注样本训练得

到的分类器在性能上有了显著提高。

TSVM 算法的一个主要缺陷在于：算法执行之前必须人为指定待训练的未标注样本中的正类样本数 N，而在一般情况下 N 值是很难准确地估计的。

5．基于图形的方法

基于图形的半监督分类方法的典型代表是 Blum 和 Chawla 提出的 Mincut 方法[62, 63]，该方法定义了一种图形，图中的节点表示样本集中的已标注样本和未标注样本，两点之间的边的权重反映了两个样本点之间的相似度。在两类问题中，Mincut 方法的目标是寻找将两类样本点分开的最小割集[64]。

Lawrence 提出的高斯随机模型（Gaussian Process Model）[65]是另外一种比较受关注的半监督学习方法，为核函数的学习提供了一种既具有理论基础又可用于实践的概率模型，在模型的选择、学习和分类方面提供了一个完整的理论框架。Chu 结合成对的类之间的联系，改进了高斯随机模型[66, 67] 。

关于半监督学习的其他算法可参见文献[39]（Seeger, 2001）、文献[68]（Chapelle, 2006）和文献[69]（Zhu Xiaojin，2006）。Xiao Li Li 和 Bing Liu 等在半监督学习方面也做了很多工作[70-75]。

1.2.2 集成学习

基于集成学习的分类方法在文献中有许多称谓，如组合分类（Combining Classifiers）、集成分类（Ensembling Learning，Classifier Ensembles）、多分类系统（Multiple Classifiers Systems）等。集成分类器是分类器的集合，这些分类器的单独决策被以某种方式组合起来（通过加权或无权重投票）以给新样本分类，研究发现集成分类器通常比组成它们的单个分类器要精确得多，前提是用错误率小于 0.5 的单个分类器，而这些单个分类器的错误之间至少应是某种程度上无关的[76-80]。如图 1.3 所示，集成分类器将 T 个学习得到的分类器 $C_1, C_2, \cdots, C_T$ 组合起来，目的是创建一个改进的分类器 C^*。

集成分类器的模型一般分为两类：一是使用不同类型的单分类器的集成分类器，称分类委员会（Classifier Committees）[4][80]；二是使用相同类型的单分类器的集成分类器，典型代表是 Breiman 的 Bagging[35][76]和 Freund 及 Schapire 的 Boosting[34][37-38]。

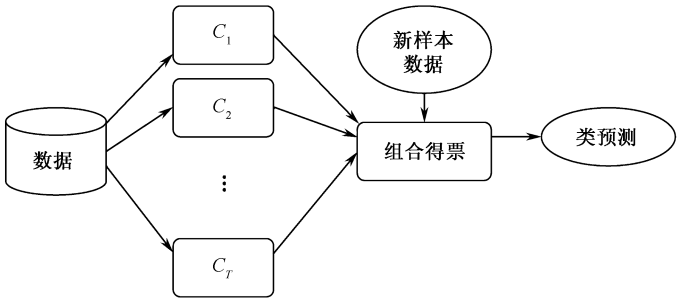


图 1.3　集成分类

1．分类委员会

分类委员会（Classifier Committees）所基于的假设是对一个需要专家进行判断的任务，k 个专家个人判断的有效组合优于个人的判断[4]。在文本分类中，选择 k 个不同的分类器 $\Phi_1,\Phi_2,\cdots,\Phi_k$，对同一个文本分类，将分类结果进行适当的组合得到最终的分类结果。一个分类委员会的特性就由两方面决定，一是 k 个分类器的选择，二是集成规则的选择。

有关 k 个分类器的选择，从机器学习的理论可以知道，为了保证结果的有效性，组成分类委员会的分类器要尽可能地互相独立，Tumer 和 Ghosh 在他们的文献[80]中有详细介绍。人们研究最多的是有关组合规则的选择，通常有如下几种。

① 多票表决（Majority Voting，MV）：是最简单组合规则，对于二值分类器，得到超过$(k+1)/2$ 个投票的分类结果被选为最终结果（k 必须是奇数）。

② 线性加权组合（Weighted Linear Combination，WLC）：对 k 个分类器的分类结果进行加权求和得到最终的分类结果。权重反映的是分类器 Φ_j 的分类有效性。

③ 动态分类器选择（Dynamic Classifier Selection，DCS）：考察与文本 x_i 最相近的校验文本集中的分类器的性能，效果最好的分类器作为分类委员会选择。

④ 自适应分类器组合（Adaptive Classifier Combination，ACC）：是介于 WLC 与 DCS 之间的一种策略。对分类委员会中的所有分类器的分类判断求和，但是分类器的权重与文本 x_i 最相近的校验文本集中的分类的效果

有关[25]。

Li 和 Jain 使用朴素贝叶斯分类器、近邻分类器、决策树做单分类器，基于 MV、DCS 和 ACC 三种组合规则实验比较之后，认为 ACC 规则是最好的[25]。由于这种评价是在小样本空间上做出的，还不能说是权威性的结论[4]。另外，分类委员会的分类结果并不总是比单分类器的分类效果好，而且它的计算代价高昂，是单个分类器的总和再加上组合规则的计算代价。这些都是需要改进的地方。

2．Bagging 方法

Bagging 方法（Bootstrap Aggregation）是由 Breiman 于 1996 年提出的，其基本思想是，将产生样本的重复 Bootstrap 实例作为训练集，每一回运行 Bagging 都给学习算法提供有替代的、随机从大小为 m 的原始训练集抽取出 m 个训练样本的集合。这种训练集被称为原始训练集合的 Bootstrap 复制，这种技术也叫做 Bootstrap 综合，即 Bagging[35]。

Bagging 方法是基于对训练集进行处理的集成方法中最简单、最直观的一种。使用 Bagging 算法的时候，理论上每个基本分类器的训练集中有 63.2% 的重复样例。Breiman 在文献[35]中同时指出，要使得 Bagging 有效，基本分类器的学习算法必须是不稳定的，所谓不稳定，是指训练样本发生小的变动会明显影响分类结果[35]，也就是说对训练数据比较敏感。基本分类器的学习算法对训练数据越敏感，Bagging 的效果越好，因此 Bagging 对于决策树和人工神经网络这样的学习算法是相当有效的。另外，由于 Bagging 算法本身的特点，使得 Bagging 算法非常适合用来并行训练多个基本分类器，这也是 Bagging 的优势，适用于大规模问题的研究。

3．Boosting 方法

Boosting（增强）方法的思想最早来源于 1984 年 Valiant 的 PAC-Learning Model[84]。学习算法根据准确度可分为“弱”学习器和“强”学习器。“弱”学习算法准确率不高，仅比随机猜测略好；“强”学习算法是准确率很高的学习算法。Valiant 提出了下面的问题：一个性能仅比随机猜稍好的“弱”学习器是否能被“提升”为一个“强”学习算法？1989 年 Schapire 提出了第一个可证明的多项式时间 Boosting 算法[34]，对这个问题给出了肯定的回答。

Boosting 算法的基本流程如下：

① 原始训练集输入，给每个样本赋予初始权重；

② 将训练集输入已知的弱分类器，弱分类器对每个样本给出假设；

③ 更新训练集中各样本的权重；

④ 对此次的弱分类器给出权重；

⑤ 转到步骤②，直到循环到达一定次数或某度量标准符合要求；

⑥ 将弱分类器按其相应的权重加权组合形成强分类器。

Boosting 方法的核心思想如下。

（1）样本的权重

Boosting 算法也通过操纵训练样本来产生多假设，每个训练样本赋予一个权重。

在没有先验知识的情况下，初始的分布应为等概分布，也就是训练集如果有 N 个样本，每个样本的分布概率为 $1/N$。

在每次的迭代中，按照一定的标准增加被分类错误的样本的权重，减少分正确样本的权重，使得下一次迭代的弱分类器能够集中力量对这些错误样本进行判断。

（2）弱分类器的权重

每个弱分类器也对应一个权重，分类精确度高的弱分类器会有大的投票权重。

Boosting 算法主要包括两个系列：Boost-by-majority 和 AdaBoost。在每一回迭代中 Boost-by-majority 通过重取样生成不同的训练集，与 Bagging 类似，只不过重取样的具体方法不同。而 AdaBoost 在每个样本上调整这种分布。AdaBoost 根据弱分类器在训练样本上的错误率来调整训练样本上的概率分布，被误分的样本将获得更大的权重，使得在下一次迭代中弱分类器更加关注这样的样本。最终的分类器通过单个弱分类器的加权投票建立起来。每个弱分类器按照其在训练集上的精度而加权。

4．Bagging 与 Boosting 的比较

Bagging 的训练集随机选择，每一次迭代的训练集相互独立；而 Boosting

的每一次迭代的训练集并不独立，它的选择与前一轮的学习结果有关。在生成弱假设时，Bagging 没有权重，可以并行生成；而 Boosting 有权重，只能顺序生成。因此 Bagging 比 Boosting 更易于修改为并行和分布处理的版本，这比较适用于大规模的数据挖掘。

J. R. Quinlan 在“Bagging，Boosting，and C4.5”[75]一书中指出，Bagging 和 Boosting 都可以有效地提高分类的准确性。在大多数数据集中，Boosting 的准确性比 Bagging 高。在有些数据集中，Boosting 会引起退化。

Bagging 和 Boosting 的学习过程都是运行多次，每次都在训练样例的子集上学习，最后综合各次迭代生成的基学习器以形成最终决策。虽然 Bagging 与 Boosting 算法因为要经过多次迭代而造成效率不够高，但是在大多数应用中，准确率比运算速度更为重要，因为计算机的性价比提高很快。Bagging 和 Boosting 都可以有效地提高分类的准确性。

Bagging 和 Boosting 在不稳定的学习算法上工作得尤其好，如决策树、神经网络，规则学习算法都是不稳定的。但是线性回归、K 近邻法、Naïve Bayesian 算法和线性阈值算法通常是很稳定的，Bagging 和 Boosting 在提升这类算法时，效果就比较差。如何利用 Bagging 和 Boosting 提升 Naïve Bayesian、K 近邻等算法的分类性能，也是比较值得探讨的问题。

半监督学习（Semi-supervised Learning）和集成学习（Ensemble Learning）作为机器学习（Machine Learning）的两大主流，在各自的领域已经取得了不少研究成果，但是也存在一些问题。本书重点对二者的代表方法 Co-training 算法和 AdaBoost 算法进行深入的研究，在理解前人研究的基础上，针对存在的问题提出几种改进方案。

❑1.3 本书内容组织

本章介绍了研究背景、文本分类及其面临的问题，阐述了基于半监督学习和集成学习的文本分类方法的研究意义和国内外研究现状。

第 2 章对文本分类的关键技术进行概述，主要包括文本分类预处理、文本的表示、特征选择、文本分类方法、实验数据集及分类模型的评估方法。

第 3 章分析了特征选择存在的问题，采用信息论中的评估函数量化特征

的重要性，调整特征的权值，提出 TEF-WA 权值调整技术；分析比较了文档频率、信息增益（Information Gain，IG）、期望交叉熵（Expected cross Entropy）、互信息（Mutual Information，MI）、χ^2 统计量（CHI）、文本证据权（Weight of Evidence for Text，WET）和几率比（Odds Ratio）等多种评估函数及实验结果。

第 4 章分析了半监督学习中的代表方法 Co-training 算法，它要求两个特征视图满足一致性和独立性的理论假设，但是直接判断两个视图是否满足独立性有一定的难度。本章提出了通过评估两个基分类器之间的差异性，间接评估二者独立性的方法。在此基础上提出了两种改进算法：TV-SC 和 TV-DC 算法。

第 5 章针对 Co-training 方法的独立性假设问题，提出了利用互信息（MI）或 CHI 统计量评估特征之间的相互独立性的方法，构造了一种特征独立模型（MID-Model）。基于该模型提出了特征子集划分方法——PMID 算法，以便把不存在自然划分的一个特征集合划分成两个独立性较强的子集，进而提出了改进的半监督分类算法——SC-PMID 算法，并且对由 PMID 算法划分得到的两个特征子集之间的独立性进行了理论论证。

第 6 章分析了集成学习算法 AdaBoost 算法不能有效提升 Naïve Bayesian 分类器性能的原因，提出了一种基于投票信息熵的样本权重维护新策略，即样本权重的调整不仅考虑是否被当前基分类器分错，还要考虑该样本在前几轮基分类器上的投票分歧，而且在错误率相同的情况下，对基分类器间的差异性贡献大的基分类器将会获得更大的置信度。在此基础上提出了对 AdaBoost 的改进算法——BoostVE 算法，并对 BoostVE 算法的最小训练错误上界进行了论证分析。理论分析和在 20-newsgroups 标准数据集上的对比实验结果表明 BoostVE 算法优于 AdaBoost 算法，能够有效提高 Naïve Bayesian 文本分类器的泛化能力。

第 7 章结合半监督学习对集成学习的 Boosting 方法进行了研究，提出了一种基于置信度重取样的 SemiBoost-CR 分类模型；提出了基于相似近邻和基于最大差距的两种置信度计算公式，按照置信度重采样，选取一定比例置信度较高和置信度较低的未标注样本，分别以不同的策略加入到已标注的训练样本集。对比实验表明使用少量的标注样本和大量的未标注样本，SemiBoost-CR 分类模型能够有效提升 NB 的分类效果。

第 8 章介绍了采用 VC++ 6.0 实现的中英文文本分类系统 SECTCS，该系统前期设计针对的是有监督分类算法，这里对原来的 SECTCS 系统进行了修改和功能扩展，进一步开发实现了半监督分类方法和集成分类方法中的经典算法，以及前几章所提出的基于半监督学习与集成学习的各种改进算法，并在 20-newsgroup 数据集和中文新闻数据集上进行了大量的对比实验和分析，验证了所提方法的有效性。本章阐述了 SECTCS 系统的原有的功能与新扩展的功能、总体结构、主要的用户界面及操作，描述了分类模型的多种评估方法和实验数据集。

第2章
文本分类技术概述

文本的自动分类技术（Automatic Text Categorization，ATC）是文本挖掘中最重要的研究领域之一。对文本进行准确、高效的分类是许多数据管理任务的重要组成部分。对文本、电子邮件的内容实时辨识和过滤并据此将其放置到相应的文件夹下，进行类别标识以便后续进行与类别相关的处理。结构化的搜索和浏览、提供个性化的服务等方面，均在一定程度上依赖于准确的文本分类技术。

❑2.1 文本分类预处理

文本分类预处理是文本分类的第一个步骤，也是比较重要的一个步骤。预处理过程对分类效果的影响至关重要，数据准备是否做好将直接影响文本挖掘的效率和准确度及最终模式的有效性。文本的预处理过程可能占据整个系统 80%的工作量。

与传统的结构化数据相比，文本分类处理的是大量的、非结构化的文本数据，这些数据一般是长期积累的结果，且没有统一的结构。因此不仅需要对这些文本数据进行数据挖掘中相应的标准化预处理，如数据的选择（选择相关的数据内容）、净化（消除噪声、冗余数据）、推测（推算缺失数据）、数据缩减（减少数据量），而且文本使用自然语言描述，计算机难以直接处

理，所以还需要进行文本数据的信息预处理。

Internet 上的大部分网页是 HTML 文档或 XML 文档，文本的预处理首先要做的是，利用网页信息抽取模块将网页的内容，去掉与分类无关的标记，转换成统一格式的 TXT 文本以备后续处理。

信息预处理的主要目的是抽取代表文本特征的元数据（特征项），对元数据进行标记、语言学分析、词性标注、短语边界辨认等。一般“词”能表达完整的语义对象，所以通常选用词作为文本特征的元数据。中文文本的预处理比英文文本的预处理复杂，因为中文的基元是字而不是词，字的信息量比较低，句子中各词语间没有固有的分隔符（如空格），因此对中文文本还需要进行词条切分处理。汉语语义及结构上的复杂性和多样性给中文自动分词带来了极大困难，这也成为中文文本信息处理中的技术难点之一。在中文信息处理领域，对中文自动分词的研究工作已经做了很多，下面重点介绍汉语分词的方法。

目前，汉语分词主要有两大类方法：基于词典与规则的方法和基于统计的方法。基于词典与规则的方法应用词典匹配、汉语词法或其他汉语语言知识进行分词，如最大匹配法（Maximum Matching）、最小分词方法等。这类方法简单、分词效率较高，但对词典的完备性、规则的一致性等要求比较高。基于统计的分词方法则将汉语基于字和词的统计信息，如相邻字间互信息、词频及相应的贡献信息等应用于分词，由于这些信息是通过训练集动态获得的，因而具有较好的鲁棒性能，但是完备性相对比较差。

在这两大类方法的基础上又可将分词的基本方法归纳为如下几种。

① 词典匹配法：如最大匹配法、逆向匹配法、增字或减字匹配法、双向扫描法、二次扫描法、逐词遍历法、部件词典法。

② 设立标志法：如切分标志法、统计标引法、多层次列举法。

③ 词频统计法：如高频优先法、基于期望法、最少分词词频法。

④ 联想词群法：如联想回溯 AB 法、词链法、多遍扫描联想法、联想树分析法、无词库法。

⑤ 语义语用法：如邻接约束法、扩充转移网络法、综合匹配法、后缀分词法。

⑥ 知识与规则法：如切词规则法、切分与语义校正法、规则描述切词法、生成-测试法、语境相关法、短语结构法、词语结构类比法。

⑦ 人工智能法：如专家系统法、神经网络方法等。

a. 专家系统分词法：将自动分词过程看成知识推理过程，力求从结构与功能上分离分词过程和实现分词所依赖的汉语词法知识、句法知识及部分语义知识。把知识的表示、知识库的逻辑结构与知识库的维护放在系统设计的首位考虑。其知识库按常识性知识与启发性知识分别进行组织。对于常识性分词知识采用“语义网络”表示，对于启发性分词知识采用“产生式规则”表示。知识库是使专家系统具有“智能”的关键部件。

b. 基于神经网络的分词方法：以模拟人脑运行，分布处理和建立数值计算模型工作。它将分词知识所分散的隐式的方法存入神经网络内部，通过自学习和训练修改内部权值，以达到正确的分词结果。这种方法的关键在于知识库（权重链表）的组织和网络推理机制的建立。

实验表明，文本分类对分词的精度要求不是很高，通常采用基于词典的“最大匹配法”。这一方法简单、高效，适合用于模型系统的设计实现。总之，汉语自动分词是中文信息处理的“瓶颈”问题，它的最终解决依赖于汉语的分词结构、句法结构、语义等语言知识的深入、系统的研究；依赖于对语言与思维的本质的揭示；同时，在很大程度上还依赖于神经网络、专家系统、知识工程等人工智能技术的研究进展。

❑2.2　文本的表示

文本的内容是人类所使用的自然语言，表达了丰富的信息，但是要把这些信息编码为一种标准形式是非常困难的。基于自然语言处理和统计数据分析的文本挖掘中的文本特征表示指的是对从文本中抽取出的元数据（特征项）进行量化，以结构化形式描述文档信息。这些特征项作为文档的中间表示形式，在信息挖掘时用以评价未知文档与用户目标的吻合程度，这一步又叫做目标表示。

常用的文本表示模型有：① 布尔逻辑模型：布尔逻辑模型通过定义一个二值变量集合来表示文档；② 向量空间模型（Vector Space Model，VSM）：1969 年 Gerard Salton 和 McGill 提出[10]；③ 潜在语义索引（Latent Semantic Indexing，LSI）：也用向量表示特征项，但是每一个向量代表一个“概念”，

由 Deerweater 和 Dumais 等于 1990 年提出[13]；④ 概率模型（Probabilistic Model）[14]：使用概率构架表示特征项，由 Belkin 和 Croft 于 1992 年提出。

本书的文本自动分类系统采用了近年来应用较多且效果较好的 VSM 方法，下面重点讨论。

向量空间模型的基本思想是使用词袋法（Bag-of-Word）表示文本，这种表示法的一个关键的假设，就是文本中的词条出现的先后次序是无关紧要的。每个特征词对应特征空间的一维，将文本表示成欧氏空间的一个向量，并成功应用到 SMART 系统[9]中，是文本分类技术使用最多的表示方法。它的核心概念可以描述如下。

① 特征项：组成文档的字、词、句子等。$x=(t_1,t_2,\cdots,t_k,\cdots t_n)$，其中 t_k 表示第 k 个特征项，作为一个维度。

② 项的权重：在一个文本中，每个特征项都被赋予一个权重 w，以表示特征项在该文本中的重要程度。

③ 向量空间模型（VSM）：在舍弃了各个特征项之间的顺序信息之后，一个文本就表示成向量，即特征空间的一个点。

如文本 x_i 的表示：$\boldsymbol{x}_i=(w_{i1},w_{i2},\cdots,w_{ik},\cdots w_{i|V|})$，其中，$w_{ik}=f(t_k,c_j)$，$f(t_k,c_j)$ 为权值函数，表示特征 t_k 决定文档 $\boldsymbol{x}_i$ 是否属于类 c_j 的重要性。V 表示特征词集合，$|V|$ 表示特征词数。

④ 相似度（similarity）：对于所有文档和用户目标都可映射到此文本向量空间，从而将文档信息的匹配问题转化为向量空间中的矢量匹配问题。n 维空间中点的距离用向量之间的余弦夹角来度量，即表示了文档间的相似程度。假设用户目标为 $\boldsymbol{u}$，未知文档为 $\boldsymbol{x}_i$，相似度计算公式如下：

$$\mathrm{sim}(\boldsymbol{x}_i,\boldsymbol{u})=\cos(\boldsymbol{x}_i,\boldsymbol{u})=\frac{\boldsymbol{x}_i\cdot\boldsymbol{u}}{\|\boldsymbol{x}_i\|\|\boldsymbol{u}\|}=\frac{\sum_{k=1}^{|V|}w_{ik}\cdot w_k}{\sqrt{\sum_{k=1}^{|V|}{w_{ik}}^2\sum_{k=1}^{|V|}{w_k}^2}} \qquad (2\text{-}1)$$

其中，“·”表示向量点积，$\|\boldsymbol{x}_i\|$ 是向量 $\boldsymbol{x}_i$ 的长度，$\mathrm{sim}(\boldsymbol{x}_i,\boldsymbol{u})\in[0,1]$。如果余弦相似度为 0，则 $\boldsymbol{x}_i$ 与 $\boldsymbol{u}$ 之间的夹角为 0°，除大小（长度）之外，二者是相同的；如果余弦相似度为 0，则 $\boldsymbol{x}_i$ 与 $\boldsymbol{u}$ 之间的夹角为 90°，并且它们不包含任何相同的特征词。因此夹角越小，余弦相似度越大，说明二者的相似度越高。

权重通常是特征项频率的函数，用 $\mathrm{TF}(t_k)$ 表示特征 t_k 出现的频率，权重

函数有多种。

① 布尔型：$w_{ik}=\begin{cases}1, & \mathrm{TF}(t_k)>0\\ 0, & \text{其他}\end{cases}$。　（2-2）

文本向量由 0 和 1 组成。

② 词频型：$w_{ik}=\mathrm{TF}(t_k)$。　（2-3）

③ 平方根型：$w_{ik}=\mathrm{TF}(t_k)^{1/2}$。　（2-4）

④ 对数型：$w_{ik}=\log(\mathrm{TF}(t_k)+1)$。　（2-5）

⑤ TF-IDF 公式：$w_{ik}=\mathrm{TF}(t_k)*\log(\frac{N}{N_k}+0.5)$。　（2-6）

比较著名的权值函数是由 Salton 在 1988 年提出的 TF-IDF 公式[9][23]，它是根据词条的重要性正比于词条的文档内频数，反比于训练文档中出现该词条的文档频数的原理构造的。N 为训练文本总数，N_k 为训练文本集中出现词条 t_k 的文本数。

归一化后处理后为：$w_{ik}=\frac{w_{ik}}{\sqrt{\sum_{t_j\in \mathbf{x}_i} w_{ij}{}^2}}$。归一化的目的是使不同的文本具有相同的长度。

文本经过分词程序分词后，首先使用停用词表去掉对分类没有贡献的词，还可采取特征词相关性分析、聚类、同义词和近义词归并等策略，最终表示成上面描述的文本向量。

❑2.3　特征选择

特征选择的目的有三个：

① 为了提高程序效率，提高运行速度。

② 数万维的特征对文本分类的意义是不同的，一些通用的、各个类别都普遍存在的特征对分类的贡献小，在某个特定的类中出现的比重大而在其他类中出现的比重小的特征对文本的贡献大。

③ 防止过拟合（Overfiting）。

一个有效的特征集合直观上说必须具备以下两个特点。

① 完全性：确实体现目标文档的内容。

② 区分性：能将目标文档同其他文档区分开来。

2.3.1 初始特征选择

在文本的向量空间模型中，向量的维数常常是数十万维，存在着大量的对分类无用的特征，也称无关特征，也就是各个类别中均可以出现的特征，它不代表类别的特点。举例来讲，“的”、“the”、“我们”、“所以”在所有的文档中都有很高的出现频率，对分类不起作用。而稀有词在全部的训练文档中的出现次数都很少，它对文档也不具代表性。这两种词都应该删除，否则会影响分类。另外，根据 ZIP 法则，频率低的词在所有单词中占的比例是很高的。例如，只出现一次的单词在所有的单词中的比例占大约 50%，因此合理地删除大量的低频词，可以降低特征空间的维数。通常，将“的”、“the”、“我们”、“所以”等常用词放在一个停用词表里，然后设置一个最低词频阈值，文本中属于停用词表中的单词和词频低于最低词频阈值的单词全部删除。为了减少冗余度、提高分类效果，还可以采取特征词相关性分析、聚类、同义词和近义词归并等策略，如“计算机”、“电脑”、“computer”应该作为一个词条处理。

2.3.2 特征选择算法

用向量空间法表示文档时，即使经过删除停用词表中的停用词及应用 ZIP 法则删除低频词，仍会有数万特征留下。最后一般只选择一定数量的最佳特征来作为分类依据。所以进一步对特征进行精选就显得异常重要。

1．机器学习中的特征选择算法

在统计学、模式识别、机器学习中都有以不同的搜索策略和评估函数进行特征选择的方法，搜索策略一般有以下几种。

（1）前向选择：将初始特征设为空集，用贪婪算法逐步增加特征。

（2）后向消除：将初始特征设为包括所有特征的全集，用贪婪算法逐步删除特征。

（3）前向逐步选择：将初始特征设为空集，用贪婪算法逐步增加或删除

特征。

（4）后向逐步消除：将初始特征设为包括所有特征的全集，用贪婪算法逐步增加或删除特征。

（5）随机应变：将初始特征设为随机选择的特征集，用给定的重复次数增加或删除特征。

机器学习中的特征选择方法可分为 Filter 和 Wrapper 两种模型[12]。在 Filter 模型中，特征选择算法是学习算法的预处理过程，与学习算法独立，两个著名的算法是 Relief 和 Fovcus 算法。在 Wrapper 模型中特征选择的算法和学习算法是交织在一起的，这种模型的代价相当高昂，即使处理几百个特征也很困难。

2．基于评估函数的特征选择

大多数机器学习的特征选择算法不适用于文本分类，因为文本分类中的特征的维数过于庞大，对于一个 n 维的向量，其各维的特征组合会有 2^n 种，即使采用一定的优化技术，这在计算复杂度上也是不可承受的。于是常常使用特征独立性的假设来简化特征选择，以达成计算时间和计算质量的折中。因为有了特征独立性的假设，文本挖掘中选择特征子集的方法和机器学习比起来是简单的。

特征子集提取的一般步骤是：通过构造一个特征评估函数，对特征集中的每个特征进行评估，每个特征获得一个评估分数，然后对所有的特征按照评估分大小进行排序，选取预定数目的最佳特征作为特征子集。评估函数一般使用逆文本频率（TF-IDF）、信息增益（Information Gain）、期望交叉熵（Expected Cross Entropy）。已经有很多研究者在特征选择方面进行了大量工作[12][15-18]，其中斯坦福大学的 Mechran Sahami 和美国卡耐基梅隆大学的 Yang Yiming 教授的文章较具代表性和总结性[12] [16]。本书将在第 3 章讨论基于评估函数的特征权重调整（Term Evaluate Function-Weight Adjustment，TEF-WA）技术。

3．潜在语义索引

潜在语义索引（Latent Semantic Indexing，LSI）的基本观点是：把高维的向量空间模型（VSM）中表示的文档映射到低维的潜在的语义空间（LSS）

中。Deerwester 等人利用线性代数的知识，通过矩阵的奇异值分解（Singular Value Decomposition，SVD）来进行信息滤波和潜在语义索引[13]。

奇异矩阵分解（SVD）的相关知识如下。

假设 $\boldsymbol{A}_{(m\times n)}$ 表示训练文档集的特征向量矩阵，m 表示文档中特征的个数，n 表示文档的个数。$\boldsymbol{A}$ 的奇异矩阵可由下面的式子得到：

$$\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}V^{\mathrm{T}} \tag{2-7}$$

其中，$\boldsymbol{U}_{(m\times K)}$ 和 $\boldsymbol{V}_{(K\times n)}$ 是正交矩阵（$\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}=1$），$\boldsymbol{\Sigma}_{(R\times R)}$ 是 $\boldsymbol{A}$ 的奇异值对角矩阵。$R\leqslant\min(m,n)$，从 $\boldsymbol{\Sigma}_{(R\times R)}$ 中选取 K 个最大的在奇异值，其余的设为 0，构成 $\boldsymbol{\Sigma}_{(K\times K)}$，此时有：

$$\boldsymbol{A}_K=\boldsymbol{U}_K\boldsymbol{\Sigma}_K\boldsymbol{V}_K^{\mathrm{T}} \tag{2-8}$$

其中，$\boldsymbol{U}_{K(m\times K)}$ 和 $U_{K(K\times n)}$ 是删除了 U、V 中相应的行和列构成的新矩阵。

$\boldsymbol{A}_K$ 从某种意义上说挖掘出了 $\boldsymbol{A}$ 的潜在的语义模式，在一定程度上克服了一词多用和多词同意的问题，同时滤掉了词的用法和多变性产生的噪声。矩阵的维数从 R 降到了 K，规模大大减少，实现了特征约减。

有了奇异矩阵 SVD，可以用 $\boldsymbol{A}_K$ 的两个列向量的余弦表示这两个文档的相似性，用两个行向量的余弦表示两个特征词的用法模式的相似性。

通过奇异值分解，将文档在高维向量空间模型中的表示投影到低维的潜在的语义空间中，有效地缩小了问题的规模。潜在语义分析在信息过滤、文本索引、视频检索等方面具有较为成功的应用。

❑2.4 文本分类算法

2.4.1 质心向量分类算法

质心向量分类算法（Centroid-based Classification，CC）也称质心分类算法，是基于向量空间模型（Vector Space Model，VSM）的一种最简单的有导师学习方法。在向量空间表示法中，每个文本用一个特征向量表示，这样通过衡量两个特征向量之间的距离，就可以度量两个文本的相似度，这也是矢量相似度法的基本思想。质心分类算法的基本思路是利用属于同类的训练

文本生成一个代表该类别的质心向量（Centroid Vector），然后在测试文本到来时确定新文本向量，计算该向量与每个类别的质心向量的距离（相似度），最后判定文本属于与该文本距离最近的类。该算法简单，也可以看成是其他分类算法的基础。

质心向量分类算法分训练（或学习）和分类两个阶段，具体如表 2.1 所示。

表 2.1　质心分类算法

（1）训练阶段

步骤 1：定义类别集合 $C=\{\boldsymbol{c}_j\}_{j=1}^{L}$。这些类可以是层次型的，也可以是并列式的。

步骤 2：给出训练文档集合 $D=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n}$，每个训练文档 $\boldsymbol{x}_i$ 都有所属类别标签 $\boldsymbol{y}_j\in C$。

步骤 3：统计 D 中的所有文档，确定每个文档的特征矢量 $\boldsymbol{x}_i, i=1,\cdots,n$。根据所有文档的特征矢量，利用算术平均法计算每个类别的特征矢量 $\boldsymbol{c}_j, j=1,\cdots,L$。

（2）分类阶段

步骤 1：对于测试文档集 $T=\{\boldsymbol{x}_i\}_{i=1}^{m}$ 中的每个待分类文档 $\boldsymbol{x}_k$，计算其特征矢量 $\boldsymbol{x}_k$ 与每个类别向量 $\boldsymbol{c}_j$ 之间的相似度 $\mathrm{sim}(\boldsymbol{x}_k,\boldsymbol{c}_j)$。

步骤 2：选取相似度最大的一个类别 $\arg\max\limits_{\boldsymbol{c}_j\in C}\mathrm{sim}(\boldsymbol{x}_k,\boldsymbol{c}_j)$ 作为 $\boldsymbol{x}_k$ 的类

有时只要 $\boldsymbol{x}_k$ 与这些类别间的相似度超过某个预定阈值，可为 $\boldsymbol{x}_k$ 指定多个类别。但若这种情况发生得太频繁，则说明预定义类别 $C=\{\boldsymbol{c}_j\}_{j=1}^{L}$ 不当，应加以修改。若文档 $\boldsymbol{x}_k$ 与所有类的相似度都低于该阈值，则将其标注为“其他”类。

通过计算两个特征向量之间的距离衡量两个特征向量的近似程度，存在 3 种最通用的距离度量：欧氏距离、余弦距离和内积。因此计算 $\mathrm{sim}(\boldsymbol{x}_k,\boldsymbol{c}_j)$ 时，有多种方法可供选择。

最简单的方法是仅考虑两个特征矢量 $\boldsymbol{x}_k$ 与 $\boldsymbol{c}_j$ 中所包含的词条的重叠程度，即

$$\mathrm{sim}(\boldsymbol{x}_k,\boldsymbol{c}_j)=\frac{\boldsymbol{x}_k\text{与}\boldsymbol{c}_j\text{具有的相同词条数}}{\boldsymbol{x}_k\text{与}\boldsymbol{c}_j\text{所有的词条数}} \tag{2-9}$$

最常用的方法是考虑两个特征矢量 $\boldsymbol{x}_k$ 与 $\boldsymbol{c}_j$ 之间的夹角余弦，即

$$\mathrm{sim}(\boldsymbol{x}_k,\boldsymbol{c}_j)=\frac{\boldsymbol{x}_k\cdot\boldsymbol{c}_j}{\|\boldsymbol{x}_k\|\times\|\boldsymbol{c}_j\|} \tag{2-10}$$

2.4.2 K 近邻分类算法

K 近邻（K Nearest Neighbor，KNN）分类算法实际上是矢量相似度法的一种改进。K 近邻分类算法的基本思想中，利用文本向量之间的夹角来衡量文本之间的相似度这一点不变，所不同的是，给定一个未标记类别的文档，对该文档所属类别的预测建立在对于与之最相似的 K 个文档所属类别的概率分布上[1][23]。

文档 $\boldsymbol{x}$ 属于 $\boldsymbol{c}$ 类的概率为

$$P(\boldsymbol{c}\mid\boldsymbol{x})=\frac{\sum_{i=1}^{K}\mathrm{sim}(\boldsymbol{x},\boldsymbol{x}_i)P(\boldsymbol{c}\mid\boldsymbol{x}_i)}{\sum_{j=1}^{L}\sum_{i=1}^{K}\mathrm{sim}(\boldsymbol{x},\boldsymbol{x}_i)P(\boldsymbol{c}_j\mid\boldsymbol{x}_i)} \tag{2-11}$$

式中，$\boldsymbol{x}_i$ 为与文档 $\boldsymbol{x}$ 最近邻的 K 个文档之一，它既可以按不同的概率属于不同的类别，也可以属于唯一的一个类别 $\boldsymbol{c}_k$，两个文档的相似程度 $\mathrm{sim}(\boldsymbol{x}_1,\boldsymbol{x}_2)$ 常用两个向量的夹角余弦或其变种来度量。显然其实质仍为比较向量夹角。当 K=1 时，待标记文档被赋予与它最临近的训练样本的类别号。Yang 在文献[23]里指出 K 的取值通常可以选 30 或 40。

K 近邻分类算法是基于要求的或懒散的学习法，即它存放所有的训练样本，并且直到新的（未标记）样本需要分类时才建立分类器。这与判定树归纳和向后传播这样的急切学习法形成鲜明对比，后者在接受新样本之前已经构造一个一般的分类模型。当与给定的未标记样本比较可能的近邻（即训练样本）数量很大时，懒散学习法的计算开销可能很大。

虽然如此，但实际上它始终是文本分类领域最有效的算法之一，而且只要在具体实现时运用一些编程技巧，它的速度并不慢。实验表明，K 近邻分类算法在实用文本系统中是非常有效的。

K 近邻分类算法的步骤如表 2.2 所示。

表 2.2 K 近邻分类算法

步骤 1：根据特征集合描述训练文本向量 $D=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n}$，定义类别集合 $C=\{\boldsymbol{c}_j\}_{j=1}^{L}$。
步骤 2：测试文本到达，分词和特征抽取，确定测试文本 $\boldsymbol{x}$ 的向量表示。
步骤 3：根据相似度计算公式（2-10），在训练文本集中选出与测试文本最相似的 K 个文本。
步骤 4：依次计算测试文本 $\boldsymbol{x}$ 的 K 个邻居属于每类 $\boldsymbol{c}$ 的概率 $P(\boldsymbol{c}\mid\boldsymbol{x})$，计算公式如式（2-11）。
步骤 5：比较步骤 4 的结果，将测试文本 $\boldsymbol{x}$ 分到概率最大的那个类，或预定阈值，可为 $\boldsymbol{x}$ 指定多种类别

2.4.3　贝叶斯分类算法

贝叶斯分类（Bayesian Classification 或 Bayes Classification）是基于贝叶斯定理（Bayes theorem）的一种统计学分类方法。该方法假设分类决策问题可以用概率的形式描述，并且假设所有的概率分布已知。贝叶斯分类方法根据属性值和类变量的概率关系建立分类模型，给定一个未标注样本，可以预测属于某个类的概率。

首先介绍贝叶斯定理（Bayes theorem），然后阐述贝叶斯定理在分类中的应用，再介绍两种贝叶斯分类模型：朴素贝叶斯和贝叶斯信念网络。

1．贝叶斯定理

假设 X，Y 是一对随机变量，它们的联合概率 $P(X=x,Y=y)$ 是指 X 取值 x 且 Y 取值 y 的概率，条件概率指一随机变量在另一随机变量取值已知的情况下取某一特定值的概率。例如，$P(Y=y \mid X=x)$ 指在变量 X 取值 x 的情况下变量 Y 取值 y 的概率。X 和 Y 的联合概率和条件概率满足如下关系：

$$P(X,Y)=P(Y \mid X)P(X)=P(X \mid Y)P(Y) \tag{2-12}$$

式（2-12）经过变换可以得到下面的公式，称贝叶斯定理。

$$P(Y \mid X)=\frac{P(X \mid Y)P(Y)}{P(X)} \tag{2-13}$$

利用贝叶斯定理，可以进行分类决策。

2．贝叶斯分类器

基于贝叶斯定理，如式（2-13）所示，可以构造贝叶斯分类器。首先，从统计学的角度形式化分类问题。在分类问题中，设 X 表示属性值，Y 表示类变量。如果类变量与属性之间的关系式不确定，那么 X 和 Y 可以看做随机变量，用 $P(Y \mid X)$ 的概率方式反映二者之间的关系。条件概率 $P(Y \mid X)$ 称为 Y 的后验概率（Posterior Probability），相应地 $P(Y)$ 称为先验概率（Prior Probability）。

在训练阶段，给定带有类别标注的训练数据集，可以统计 Y 的先验概率 $P(Y)$、类条件概率（Class-conditional Probability）$P(X \mid Y)$ 和属性集 X 的概

率 $P(X)$。在分类阶段，给定一个样本的属性值组合，根据贝叶斯定理可以估算该样本属于某一类别的后验概率 $P(Y|X)$。

比较样本属于不同 Y 值的后验概率时，式（2-13）中的分母 $P(X)$ 是常数，因此 $P(X)$ 可以忽略，即样本属于某一类的后验概率 $P(Y|X)$ 与类条件概率（Class-conditional Probability） $P(X|Y)$ 和 Y 的先验概率 $P(Y)$ 的乘积成正比。从而得到：

$$P(Y|X) \propto P(X|Y)P(Y) \tag{2-14}$$

给定标注训练集，先验概率 $P(Y)$ 可以通过计算训练集中属于每个类的训练样本所占的比例计算。对于类条件概率 $P(X|Y)$ 的估计，下面讨论两种应用于文本分类的贝叶斯分类方法的实现：朴素贝叶斯分类算法和贝叶斯信念网络。

3．朴素贝叶斯分类算法

朴素贝叶斯（Naïve Bayesian，NB）分类算法，在估算类条件概率时，假设属性之间条件独立。在文本分类问题中，类变量 Y 的取值范围为 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_L$，随机变量 X 取值对应的是一个文本特征向量 $\boldsymbol{x}=(w_1,w_2,\cdots,w_k,\cdots,w_{|V|})$，其中，$V$ 表示特征词集合，即属性集合。每一个特征词就是一个属性，属性之间的条件独立假设表示各属性（特征词）独立地作用于决策类别变量。也就是说，各个特征词分布相互独立，所有的特征词节点只有唯一的父节点（类节点），如图 2.1 所示。

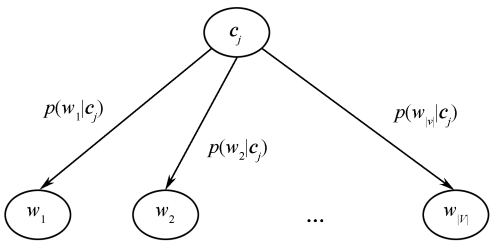


图 2.1　朴素贝叶斯网络

如果没有特征的独立假设前提，$P(\boldsymbol{x}|\boldsymbol{c}_j)$ 的计算开销非常大。在独立假设的前提下，文本向量 $\boldsymbol{x}=(w_1,w_2,\cdots,w_k,\cdots,w_{|V|})$ 的各个属性之间条件独立，

每个属性 w_k 只与父节点类 $\boldsymbol{c}_j$ 相关（如图 2.1 所示）。这样，类条件概率 $P(\boldsymbol{x}\mid\boldsymbol{c}_j)$ 的计算就简化如下：

$$P(\boldsymbol{x}\mid\boldsymbol{c}_j)=\prod_{t=1}^{|V|}P(w_t\mid\boldsymbol{c}_j) \tag{2-15}$$

虽然这一假设在一定程度上限制了简单贝叶斯分类器的适用范围，然而在实际应用中，不仅以指数级降低了贝叶斯网络构建的复杂性，而且在许多领域，在违背这种假定的条件下，朴素贝叶斯分类器也表现出相当的健壮性和高效性[2]。它已成功地应用到分类、聚类及模型选择等数据挖掘任务中。

根据贝叶斯定理，每个文本 $\boldsymbol{x}$ 属于每个类 $\boldsymbol{c}_j$ 的后验概率 $P(\boldsymbol{c}_j\mid\boldsymbol{x})$ 可以计算如下：

$$P(\boldsymbol{c}_j\mid\boldsymbol{x})\propto P(\boldsymbol{x}\mid\boldsymbol{c}_j)P(\boldsymbol{c}_j) \tag{2-16}$$

由式（2-15）和式（2-16）可得：

$$P(\boldsymbol{c}_j\mid\boldsymbol{x})\propto P(\boldsymbol{c}_j)\prod_{t=1}^{|V|}P(w_t\mid\boldsymbol{c}_j) \tag{2-17}$$

这样，由式（2-17）可知，朴素贝叶斯分类器的参数由先验类概率值 $P(\boldsymbol{c}_j)$ 和基于类的词条的条件概率 $P(w_t\mid\boldsymbol{c}_j)$ 组成，$P(\boldsymbol{c}_j)$ 和 $P(w_t\mid\boldsymbol{c}_j)$ 的值完全由已标注的训练集文档本确定。其中，每个类 $\boldsymbol{c}_j$ 的先验类概率值 $P(\boldsymbol{c}_j)$ 的计算公式为

$$P(\boldsymbol{c}_j)=\frac{1+\sum_{\boldsymbol{x}_i\in D}P(\boldsymbol{c}_j\mid\boldsymbol{x}_i)}{|C|+|D|} \tag{2-18}$$

式（2-18）中 $|C|$ 为类数目，$|D|$ 为训练集合中的文本数目。

基于类的词条的条件概率 $P(w_t\mid\boldsymbol{c}_j)$ 由公式（2-19）估计：

$$\begin{aligned}P(w_t\mid\boldsymbol{c}_j)&=\frac{1+\mathrm{TF}(w_t,\boldsymbol{c}_j)}{|V|+\sum_{k=1}^{|V|}\mathrm{TF}(w_k,\boldsymbol{c}_j)}\\&=\frac{1+\sum_{\boldsymbol{x}_i\in D}N(w_t,\boldsymbol{x}_i)P(\boldsymbol{c}_j\mid\boldsymbol{x}_i)}{|V|+\sum_{k=1}^{|V|}\sum_{\boldsymbol{x}_i\in D}N(w_k,\boldsymbol{x}_i)P(\boldsymbol{c}_j\mid\boldsymbol{x}_i)}\end{aligned} \tag{2-19}$$

其中，$\mathrm{TF}(w_t,\boldsymbol{c}_j)$ 为特征词条 w_t 在 $\boldsymbol{c}_j$ 类文本中出现的频度，$N(w_t,\boldsymbol{x}_i)$ 为特征词条 w_t 在文本 $\boldsymbol{x}_i$ 中出现的次数。$|V|$ 代表文本集合中全部不同特征词条

的数目。对于训练文本集合中的文本 $\boldsymbol{x}_i$，定义当 $\boldsymbol{x}_i$ 属于类别 $\boldsymbol{c}_j$ 时，$P(\boldsymbol{c}_j \mid \boldsymbol{x}_i)=1$，否则 $P(\boldsymbol{c}_j \mid \boldsymbol{x}_i)=0$。

对于测试文本集中的无标注文本，利用已经训练好的分类器，可以求出文本 $\boldsymbol{x}$ 属于类别 $\boldsymbol{c}_j$ 的后验概率 $P(\boldsymbol{c}_j \mid \boldsymbol{x})$，用 w_k 表示文本 $\boldsymbol{x}$ 中的第 k 个特征词条，公式为

$$\begin{aligned} P(\boldsymbol{c}_j \mid \boldsymbol{x}) &\propto P(\boldsymbol{c}_j)\prod\nolimits_{w\in V} P(w \mid \boldsymbol{c}_j) \\ &\propto P(\boldsymbol{c}_j)\prod_{k=1}^{|\mathrm{x}|} P(w_k \mid \boldsymbol{c}_j) \end{aligned} \tag{2-20}$$

当 $\boldsymbol{x}$ 的后验概率 $P(\boldsymbol{c}_j \mid \boldsymbol{x})$ 超过某个预定阈值时，也可以为 $\boldsymbol{x}$ 指定多种类别。但若这种情况发生得太频繁，则说明预定义的类别集合 $C=\{\boldsymbol{c}_j\}_{j=1}^{L}$ 不恰当，应加以修改。也可以设置阈值，当文档 $\boldsymbol{x}$ 对所有类的后验概率都低于该阈值时，则将其标注为“其他”类。

朴素贝叶斯算法如表 2.3 所示。

表 2.3　朴素贝叶斯算法

步骤 1：训练阶段，在训练文本集 $D=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{n}$ 和类别集合 $C=\{\boldsymbol{c}_j\}_{j=1}^{L}$ 上计算每个类的先验概率 $P(\boldsymbol{c}_j)$，计算特征词属于每个类的条件概率 $P(w_t \mid \boldsymbol{c}_j)$，分别如式（2-18）和式（2-19）所示。

步骤 2：测试阶段，测试文本生成特征向量，按公式（2-20）计算文本 $\boldsymbol{x}$ 属于每个类 $\boldsymbol{c}_j$ 的后验概率 $P(\boldsymbol{c}_j \mid \boldsymbol{x})$。

步骤 3：比较测试文本属于每个类别的后验概率，将其分到最大的那个类别 $\boldsymbol{c}_k$。

$$\boldsymbol{c}_k = \arg\max_{k}\{P(\boldsymbol{c}_j \mid \boldsymbol{x})\}$$

4．贝叶斯信念网络

朴素贝叶斯分类器的条件独立假设有些太严格，特别是对于属性之间有一定相关性的分类问题。贝叶斯信念网络（Bayesian Belief Networks，BBN），简称贝叶斯网络（Bayesian Networks），用带权无环有向图表示一组随机变量之间的概率关系。该模型不要求给定类的所有属性条件独立，而是允许指定哪些属性条件独立。

贝叶斯网络由两部分组成：

（1）一个有向无环图，节点表示随机变量，弧表示变量之间的依赖关系（相关性）；

（2）一个概率表，把各节点和它的双亲节点关联起来。

有向无环图中的每个节点表示一个变量，每条弧表示变量之间的依赖关系（相关性、因果关系），没有弧连接的节点之间彼此条件独立。如果存在一条从节点 Y 到节点 X 的有向弧，则表示 Y 是 X 的双亲，X 是 Y 的孩子，该依赖关系由条件概率 $P(X|Y)$ 表示。如果网络中存在一条从 Z 到 X 的路径，则表示 Z 是 X 的祖先，而 X 是 Z 的后代。

节点与节点之间的弧定义了网络结构，而条件概率是给定结构的参数。有向无环图中的每个节点关联一个概率表。

（1）如果节点 X 没有双亲节点，则表中只包含先验概率 $P(X)$。

（2）如果节点 X 只有一个双亲节点 Y，则表中包含条件概率 $P(X|Y)$。

（3）如果节点 X 有多个双亲节点 $\{Y_1, Y_2, \cdots, Y_m\}$，则表中包含条件概率 $P(X|Y_1, Y_2, \cdots, Y_m)$。

例如，如图 2.2 所示的贝叶斯网络，有向无环图定义了网络结构，描述了四个随机变量 A、B、C 和 D 的依赖关系，其中 A 与 B 条件独立，且都是 C 的孩子节点，D 是 C 的双亲，D 是 A 的祖先，A 是 D 的后代，而且 B 和 D 不是 A 的后代。从节点 C 到节点 A 存在一条弧，意味着 A 的概率以 C 的值为条件，C 在 A 上有直接影响（Direct Influence）。

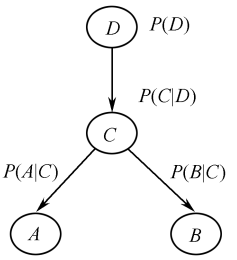


图 2.2　贝叶斯网络

贝叶斯网络的一个重要性质：条件独立贝叶斯网络中的一个节点，如果它的双亲已知，则它独立于它的所有非后代节点。

如图 2.2（a）所示，给定 C，A 条件独立于 B 和 D，因为 B 和 D 都是 A 的非后代节点。

朴素贝叶斯分类器中的条件独立假设也可以用贝叶斯网络来表示，如图 2.3 所示，其中 c_j 是目标类，$\{w_1, w_2, \cdots, w_5\}$ 是特征词集合。每个特征词只

有一个双亲 c_j，特征词 $w_1, w_2, \cdots, w_5$ 之间条件独立。

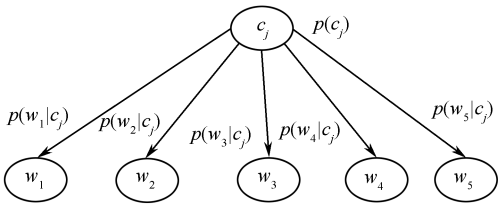


图 2.3　朴素贝叶斯网络

在图 2.3 所示的贝叶斯网络中，因为 w_1, w_2, w_3, w_4, w_5 之间彼此条件独立，所以 $p(w_1, w_2, w_3, w_4, w_5, c_j) = p(c_j)\prod_{k=1}^{5} p(w_k \mid c_j)$。这样，对此问题求解的参数就大大减少了，如果没有上述条件独立性，则需要 $2^6-1=63$ 个参数。

贝叶斯网络的本质作用就是通过引入条件独立性，大大简化了对随机变量概率的计算。使用贝叶斯网络的好处之一便是可以方便地运用先验知识，同时因为贝叶斯网络是以图形的关系给出的，所以很容易理解并加以人工干预，如可以采用手工方式去掉不合理的弧。

在使用贝叶斯网络处理文本分类时，一般把文本中的每个特征作为一个节点，所属类别也作为一个节点。显然，由于文本数据的高维特性，必须引入一些特征独立性假设，才能控制计算复杂度。问题是引入多强的独立性假设才能既控制计算量又不至于太失真。而 Naïve Bayesian 网络是最简单，同时也是引入独立性假设最多的贝叶斯网络，其结构如图 2.1 和图 2.3 所示。

在图 2.1 和图 2.3 中，c_j 是类别变量，$w_1, w_2, \cdots, w_k, \cdots, w_{|V|}$ 都是属性变量，在文本分类问题中，它们就是文本中出现的各个特征词。Naïve Bayesian 分类器假设这些特征词节点彼条件独立，每个特征词节点只有唯一的一个父节点，那就是类节点 c_j。这一假设显然与事实不符。但经验表明，Naïve Bayesian 分类器在实际的分类工作中可以得到很理想的结果。

贝叶斯网络的建模包括两个步骤：① 创建网络结构；② 估计每个节点的概率表中的概率值。网络的拓扑结构可以由领域专家知识编码获取。

贝叶斯网络具有以下特点。

（1）贝叶斯网络提供了一种图形模型来获取特定领域的先验知识的方

法。网络将先验知识以概率方式描述变量之间的依赖关系。

（2）构造贝叶斯网络比较费时，然而对于构造好的网络，添加新变量将非常容易。

（3）贝叶斯网络适合处理不完整数据。

贝叶斯网络是带有概率注释的有向无环图，利用贝叶斯定理揭示学习和统计推断功能，可以实现预测、分类、聚类、因果分析等数据挖掘任务[117]。

2.4.4　关联规则分类算法

关联规则挖掘的研究是近年来研究较多的数据挖掘方法，在数据挖掘的各种方法中应用得也比较广泛。关联规则的概念是由 Agrawal、Imielinski、Swami 提出的，是数据中一种简单但实用的规则[19]。关联规则模式属于描述型模式，发现关联规则的模式属于描述型模式，发现关联规则的算法属于无监督学习的方法。关联规则分类器是一种很通用的分类算法，Foil 系统是其典型代表。

关联规则分类器的原理是：每个类对应一个规则集，其中每个规则对应此类的一个子集。一个问题是可能存在许多规则，它们对于训练集合的精度差不多，对测试集合性能却差得很远。程序在这些差不多的规则间如何取舍呢？这时可以将先验知识编码到程序中，来帮助决定到底用哪些规则。可以允许只要绝大多数 c 类样本符合其某规则的前提条件，就认为此规则是符合类 c 的。符合此规则的其他类的样本数目所占比例应是很小的。这样每个规则都有一个对应概率。允许规则包含少量不属于本类的样本，也是解决噪声的一个好办法。因为在噪声下，负样本集中也许有少数样本实际应是正样本。用关联规则分类比用统计方法分类一般可以得到更高的精度，不过也付出了牺牲覆盖度的代价。

2.4.5　支持向量机

支持向量机（Support Vector Machine，SVM）是一种近年来发展很快的机器学习方法[20-22]，它在多种分类问题表现出了优异的推广性能，其基本思想是基于统计学习理论的结构风险最小化。支持向量机可以很好地应用于高

维数据，避免维数灾难问题。该方法的独特之处是使用训练样本的一个子集来表示决策边界，该子集称为支持向量（Support Vector，SV）。

图 2.4 显示了的一个数据集，包含两类不同的样本，分别用圆圈和方块表示。这个数据集是线性可分的，即可以找到一个超平面，使得所有的圆圈位于这个超平面的一侧，而所有的方块位于超平面的另一侧，如图 2.4 中的 B_1 和 B_2。这样的超平面可能存在无穷个，虽然它们的训练误差都等于零，但是不能保证这些超平面在未知样本中也很好，根据在测试样本上的分类结果，分类器需要从多个超平面中选择一个作为决策边界。

为了更好地泛化，不仅希望不同类别样本在超平面的两侧，而且还希望它们离超平面有一定的距离。如图 2.4 所示，考虑两个决策边界 B_1 和 B_2，这两个决策边界能够准确无误地把两类训练样本划分到各自的类中。其中 B_1 两侧的虚线 B_{11} 和 B_{12} 分别表示过两类中离决策边界 B_1 最近的样本且平行于决策边界 B_1 的两个超平面。B_{11} 和 B_{12} 之间的距离称为边缘或分类间隔（margin）。具有较大分类间隔的决策边界比较小分类间隔的决策边界具有更好的泛化能力。因此，为了更好地泛化，希望分类间隔（margin）最大化。

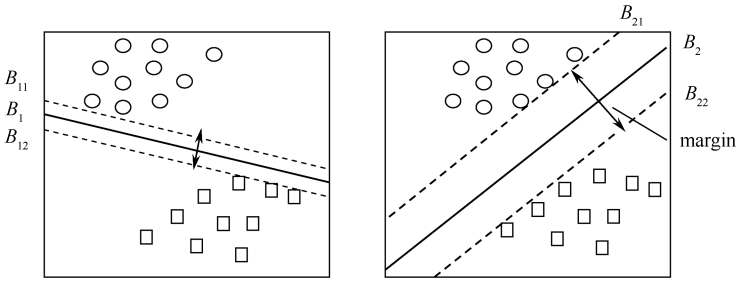


图 2.4　分类超平面和最优分类超平面

最优分类超平面（Optimal Separating Hyperplane）就是不仅能够把两类正确分开（训练错误率为 0），而且使分类间隔（margin）最大的超平面，简称最优超平面。如图 2.4 中的 B_2 就是最优分类超平面，B_{21} 和 B_{22} 之间的距离最大，B_{21} 和 B_{22} 上的样本点称为支持向量。

统计学理论给出了线性分类器分类间隔（margin）与其泛化误差之间关系的形式化解释，称之为结构风险最小化（Structural Risk Minimization，SRM）理论[21]。该理论的根据是分类器的训练误差 R_{emp}（经验风险）、训练样本数 N、

模型的复杂度 h（即模型的能力），分类器的泛化误差 R（实际风险）的上界。即在概率 $1-\eta(0\leqslant\eta\leqslant 1)$ 下，分类器的泛化误差 R 在最坏情况下满足式（2-21）。

$$R\leqslant R_{\text{emp}}+\varphi\left(\frac{h}{N},\frac{\log\eta}{N}\right) \tag{2-21}$$

其中，φ 是分类模型能力 h 的单调递增函数。分类模型的能力通常用 VC 维（Vapnik-Chervonenkis Dimension）来刻画。模式识别中 VC 维的直观定义是：对一个指示函数集，如果存在 h 个样本能够被函数集中的函数按所有可能的 2^h 种形式分开，则称函数集能够把 h 个样本打散，函数集的 VC 维就是它能打散的最大样本数目 h，若对任意数目的样本都有函数能将它们打散，则函数集的 VC 维是无穷大的。VC 维反映了函数集的学习能力，VC 维越大则分类器越复杂，所以 VC 维又是分类器复杂程度的一个衡量标准。

结构风险最小化（SRM）体现了训练误差与模型复杂度的折中。根据 SRM 原理，随着能力 h 的提高，泛化误差 R 的上界也随之提高。因此，支持向量机（SVM）在高维空间需要寻找一个最优超平面作为两类的分割，以保证最坏情况下泛化误差 R 的上界最小。如果给出两类线性可分样本，在给出线性分类超平面的时候，人们直观地趋向于将分类超平面取在离两类样本点都距离较远的地方，因为感觉上这种做法比较保险。Vapnik 从数学理论上给出了这种做法的理论依据，并推导出了这种方法风险性能的衡量，以及一整套求解的步骤。支持向量机的一个重要的优势是处理线性不可分的情况。

在线性可分的情况下，假设存在训练样本 $(\boldsymbol{x}_1,\boldsymbol{y}_1),\cdots,(\boldsymbol{x}_n,\boldsymbol{y}_n)$，$\boldsymbol{y}_i\in\{+1,-1\}$，$i=1,\cdots,n$（$n$ 为样本数，在线性可分的情况下就会有一个超平面使得这两类样本完全分开，设该超平面的形式为：

$$g(\boldsymbol{x})=\omega\cdot\boldsymbol{x}+b=0 \tag{2-22}$$

公式中的圆点“·”表示向量点积，分类如下：

$g(\boldsymbol{x})\geqslant 0$，　对应于 $\boldsymbol{y}_i=+1$

$g(\boldsymbol{x})<0$，　对应于 $\boldsymbol{y}_i=-1$

如果训练样本可以无误差地被划分，且每一类样本与超平面距离最近的向量与超平面之间的距离最大，则称这个超平面为最优分类超平面，如图 2.4 中的 B_2。

根据结构风险最小化（SRM）理论，需要寻找有最小 VC 维的超平

面[20, 21]，使得正样本和负样本之间的距离最大化。

这样，当面临一个未知测试样本时，如果它在此超平面上方，则判断其类标为 $y_i=+1$，反之 $y_i=-1$。它可以通过最大化 margin 来求得。此处的 margin $\rho=\frac{2}{\|w\|^2}$ 被定义为某训练样本点与分类超平面的最小距离，这里所说的支持向量就是满足 $g(\boldsymbol{x})=\omega\cdot\boldsymbol{x}+b=1$ 的点 (x_i,y_i)。因此分类问题就被转化为一个二次规划问题。

将判别函数归一化，使得两类所有样本都满足 $|g(\boldsymbol{x})|\geqslant 1$，即

$$y_i[(\omega\cdot\boldsymbol{x}_i)+b]-1\geqslant 0, i=1,2,\cdots,n \tag{2-23}$$

据此可以定义 Lagrange 函数：

$$L(\omega,b,\alpha)=\frac{1}{2}(\omega\cdot\omega)-\sum_{i=1}^{n}\alpha_i\{y_i[(\omega\cdot\boldsymbol{x}_i)+b]-1\} \tag{2-24}$$

其中，$\alpha_i>0$ 为 Lagrange 乘数，对 ω 和 b 求偏微分并令其为 0，原问题转换成如下对偶问题：在约束条件

$$\sum_{i=1}^{n}y_i\alpha_i=0,\ \alpha_i\geqslant 0, i=1,\cdots,n \tag{2-25}$$

下对 α_i 求解下列函数的最大值：

$$Q(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) \tag{2-26}$$

如果 α_i^* 为最优解，那么

$$\omega^*=\sum_{i=1}^{n}\alpha_i^* y_i\boldsymbol{x}_i \tag{2-27}$$

对于线性不可分的情况，可以引入松弛因子，在求最优解的限制条件中加入对松弛因子的惩罚函数。完整的支持向量机还包括通过核函数的非线性变换将输入空间变换到一个高维空间，然后在高维空间中求取线性分类面。常见的核函数包括多项式核函数、径向基函数、Sigmoid 函数等。值得指出的是，最终判别函数只包括与支持向量的内积的求和，所以识别时计算复杂性只取决于支持向量的个数。

SVM 很大的一个优点是它不受问题维数的限制，所以特别适宜在高维空间中工作，这对于文本分类来说是非常适宜的。但是，当样本数很大时，用

标准的数学方法解决二次规划问题在计算量上是不可行的。所以人们又提出了种种算法，使 SVM 能够实用化。由于具有较好的泛化性能，支持向量机被用于多个模式识别领域。在文本分类方面也有多种研究试验结果。在多个实验结果中，SVM 均取得了比原有多种分类方法更高的分类精度。更多关于 SVM 的问题可以参考 Vapnik 博士的《统计学习理论的本质》[21]。

2.4.6　其他分类算法

1．决策树（Decision Tree）

决策树学习是以实例为基础的归纳学习算法。它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。它的每个内部节点表示一个给定特征词是否在文本中出现的测试，每个分支代表一个测试输出，而每个叶子节点代表类或类分布。数的顶层节点是根节点。决策树归纳的基本算法是贪心算法，它以自顶向下递归各个击破方式构造决策树。常用的归纳决策树算法有 Bayesian 方法、CART 算法、ID3 算法[1]、C4.5（C5.0）[2]。最为典型的决策树学习系统是 ID3，它采用自顶向下的不回溯策略，能保证找到一颗简单的树。算法 C4.5 和 C5.0 都是 ID3 的扩展，它们将分类领域从类别属性扩展到数值型属性。

在决策树学习算法中，除了考虑分类正确性以外，决策树的复杂度是另外一个重要因素。表示不当、噪声、存在重复的子树等原因造成决策树过大。通常通过剪枝（预剪枝 pre_pruning 和后剪枝 post_pruning）、修改测试属性空间、对实例数据控制、采用其他数据结构等方法来简化决策树。Oliver 将决策树转化成简约决策图（RODG）[2]，Gaines 提出了例外有向无环图（EDAG）[2]，Quinlan 在 C4.5 系统中将决策树转化为规则[2]，之后对规则进行剪枝。通过简化决策树，剪去最不可靠的分支，有助于提高决策树对外来数据正确分类的能力。

2．神经网络

神经网络（Neural Network，NN）领域采用感知算法进行分类，最常见的有反向传播算法（Back Propagation，BP）[12]等。神经网络的性质主要取

决于以下两个因素：一是网络的拓扑结构；另一个是网络的权值、工作规则。二者结合起来就可以构成一个网络的主要特征。

神经网络的学习问题就是网络的权值调整问题。在这种模型中，分类知识存储在连接的权值上，使用迭代算法确定权值。当网络输出判别正确时，权向量不变，否则进行增大或减小的调整。对线性可分的情况，感知算法是收敛的，对于线性不可分的情况，一般不收敛，可以采用最小均方差误差准则。

3．线性最小方差匹配分类器

线性最小方差匹配分类器（Linear Least Squares Fit，LLSF）是由 Yang 提出的[23]。一个多变量回归模型从训练文本集合和它们的类别中被自动学习。训练数据表示成输入输出向量对的形式，输入向量仍是传统的向量空间模型中的一个文本，输出向量由对应文本的类别组成。通过解线性最小方差匹配，可以获得由词-类别回归因子组成的矩阵。

$$\mathbf{Fls} = \arg\min \| \boldsymbol{FA} - \boldsymbol{B} \|^2 \qquad (2\text{-}28)$$

其中，矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 表示训练数据（它们对应的列是一对输入输出向量），**Fls** 表示解矩阵，定义了一个从任意文件到加权类向量的一个匹配。权重越大，就说明文本越可能属于此类别。Yang 在文献[23]中对各种分类算法做了综合比较，她的结论对以后的研究具有很重要的参考价值。

❑2.5 实验数据集

各种文本分类算法的分类性能优劣的比较，需要一个共同的训练数据集和测试数据集，这里采用国际上通用的英文文本分类标准数据集 20-newsgroups，以及从易宝中文下载的中文新闻数据集。

1．英文文本分类标准数据集 20-newsgroups

20-newsgroups 数据集①包含了大约 20 000 个新闻文本，按照主题划分成 20 个新闻组，最初由 Ken Lang 收集整理，是机器学习中文本分类、文

① http://people.csail.mit.edu/jrennie/20Newsgroups/.

本聚类等研究领域广泛使用的通用实验数据集。20-newsgroups 按照类别主题划分成 20 个新闻组，其中的一些新闻组是比较相近的，如 comp.sys.ibm.pc.hardware/comp.sys.mac.hardware，而有的是高度不相关的，如 misc.forsale/soc.religion.christian。20 newsgroups 数据集的主题类别如表 2.4 所示。

表 2.4　20-newsgroups 数据集的主题类别

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast talk.religion.misc	alt.atheism soc.religion.christian

2．中文文本分类数据集

从易宝中文下载的包括国际、经济、体育、文教、政治 5 个主题类别的 20 341 篇新闻文本可以看出，经济、政治、国际是比较难以区分的类别，文教和体育两个类别比较相近。实验中，将训练集划分为不同数量的文本组成训练文本集和测试文本集，然后做交叉验证，按照 Lewis 划分的要求，训练文本集与测试文本集没有交集。令 L_n、U_m、T_k 分别表示包含 n 篇的训练文本集、包含 m 篇没有类标签的未标注文本集和包含 k 篇的测试文本集。

❑2.6　分类模型的评估方法

文本分类模型的常用评估方法有预留法（hold-out）和交叉验证法（cross-validation）。两种方法均将数据分为训练集和测试集两部分。学习和测试反复进行，最后用一个平均值来衡量模型的好坏。在预留法中从数据集中随机抽取预定大小的一个子集作为测试集，其余数据作为训练集；在交叉验证法中从数据集中按照所要进行的学习-测试循环次数分成相当数目的子集，每次循环中，其中的一个子集作为测试集，而其他子集的并集作为

训练集[3, 4]。本书采用的是预留法。

文本分类模型的评价指标，除了来源于信息检索所用的精度（precision）、召回率（recall）等，还有来源于数据挖掘中的收益率（gain）、置信度（certainty）、简洁性（simplicity）、分类正确率（classification accuracy）、精度与召回率的几何平均数、信息估值（information score）等，目的是衡量所发现知识的有效性、可用性和可理解性[3, 4]。下面对其中几种主要的，也是本书实验中使用的指标进行简要描述。

令 TP(c_j)表示属于 c_j 类的样本且被正确分为 c_j 类的样本数；FN(c_j) 表示属于 c_j 类的样本，但是没有被分为 c_j 类的样本数；FP(c_j) 表示不属于 c_j 类的样本但是被分为 c_j 类的样本数。分类模型的精度 Precision、召回率 Recall、F1 值、宏平均精度 Macro-recision、宏平均召回率 Macro-Recall、宏平均 F1 值 Macro-F1、微平均 F1 值 Micro-F1 分别计算如下。

① 精度（Precision）：被分为目标类的样本集合中正确样本所占的比例。

$$\text{Precision}(c_j)=\frac{\text{TP}(c_j)}{\text{TP}(c_j)+\text{FP}(c_j)} \tag{2-29}$$

② 召回率（Recall）：实际是目标类的样本集合中正确样本所占的比例。

$$\text{Recall}(c_j)=\frac{\text{TP}(c_j)}{\text{TP}(c_j)+\text{FN}(c_j)} \tag{2-30}$$

③ F 方法。

有时可将精度和召回率两者结合起来，如在信息检索中常用的 F 方法，两者相对重要性用一个参数β来刻画。

$$F_\beta(c_j)=\frac{(1+\beta^2)\text{Precision}(c_j)\times\text{Recall}(c_j)}{\beta^2\text{Precision}(c_j)+\text{Recall}(c_j)} \tag{2-31}$$

式中，β取值[0，∞]，β=0 时 F_β即为 Precision，β=∞时 F_β即为 Recall。

当β=1 时，即 precision 与 recall 在估计模型 M 中有着同样的重要性，称为 F1 度量：

$$\text{Fl}(c_j)=\frac{2\ \text{Precision}(c_j)\times\text{Recall}(c_j)}{\text{Precision}(c_j)+\text{Recall}(c_j)} \tag{2-32}$$

β<1 时强调 Precision 的作用，β>1 时强调 Recall 的作用。

④ 宏平均精度 Macro-Precision：

$$\text{Macro-Precision} = \frac{1}{|C|}\sum_{j=1}^{|C|}\text{Precision}(c_j) \tag{2-33}$$

⑤ 宏平均召回率 Macro-Recall：

$$\text{Macro-Recall} = \frac{1}{|C|}\sum_{j=1}^{|C|}\text{Recall}(c_j) \tag{2-34}$$

⑥ 宏平均 F1 值 Macro-F1：

$$\text{Macro-F1} = \frac{1}{|C|}\sum_{j=1}^{|C|}\text{F1}(c_j) \tag{2-35}$$

⑦ 微平均精度 Micro-Precision：

$$\text{Micro-Precision} = \frac{\sum_{j=1}^{|C|}\text{TP}(c_j)}{\sum_{j=1}^{|C|}[\text{TP}(c_j)+\text{FP}(c_j)]} \tag{2-36}$$

⑧ 微平均召回率 Micro-Recall：

$$\text{Micro-Recall} = \frac{\sum_{j=1}^{|C|}\text{TP}(c_j)}{\sum_{j=1}^{|C|}[\text{TP}(c_j)+\text{FN}(c_j)]} \tag{2-37}$$

⑨ 微平均 F1 值 Micro-F1：

$$\text{Micro-F1} = \frac{2\text{Micro-Precision}\times\text{Micro-Reccall}}{\text{Micro-Precision}+\text{Micro-Reccall}} \tag{2-38}$$

❑2.7　本章小结

本章阐述了文本分类的关键技术，包括文本分类预处理、文本的表示方法、特征选择方法、分类常用算法，以及文本分类通用数据集和文本分类模型的几种评估方法。

第3章

TEF-WA 权值调整技术

❑3.1 特征选择存在的问题

特征选择是文本分类的关键步骤，特征选择的好坏直接影响分类学习的结果。通常，特征子集的提取过程为：通过构造一个特征评估函数，对训练文档集的特征集 V 中的每个特征进行评估，每个特征获得一个评估分数，然后对所有的特征按照评估分大小进行排序，选取预定数目的最佳特征作为特征子集 V'。特征选择后，特征的删除情况通常用式（3-1）表示，称 ξ 为删除因子。

$$\xi = \frac{|V| - |V'|}{|V|} \tag{3-1}$$

经过特征选择，能够降低文本的维数、提高文本分类算法的时间效率。但是文本的特征选择策略是只留下特定数量个对分类最有用的特征词，其余的无用特征词完全删掉，在一定程度上对提高分类精度有帮助，但是对留下来的特征词同等对待，没有再做处理。

权重调整技术根据各词条对分类的有用程度调整权重，有用词条赋予较高的权重，无用词条赋予较低的权重。以往的权重调整是先按某个评估函数（用得比较多是信息增益）给每个特征词条打分，根据评估分排队，然后使用权重函数给特征词条赋予一个权重。如果权重的值或者是 0 或者是 1，那就与特征选择无异。权重函数用得较多的是 TF-IDF 公式。

TF-IDF 权值函数以 TF 词频和逆文本频度 IDF 的乘积作为特征空间坐标

系的取值测度。它建立在这样基本假设之上。第一，在一个文本中出现次数很多的单词，在另一个同类文本中出现的次数也会很多，反之亦然。所以如果将特征空间坐标系取 TF 词频作为测度，就可以体现同类文本的特点，同类文本向量的距离很近，而不同类文本向量彼此距离相对较远。第二，TFIDF 法认为一个词条出现的文本频数越小，它区别不同类别的能力就越大，所以引入了逆文本频度 IDF 的概念。

近年来，一些研究者[17][28, 29]对使用 TF-IDF 权重函数给特征词加权的合理性提出了异议，许多实验也证明使用 TF-IDF 权重函数给特征词加权并不一定就能得到高的分类精度。原因在于一个文本中对分类有用的词条只占一小部分，而大部分词条与所要判别的类无关，属于“噪声单词”。结果两个文本之间的夹角在很大程度上是由这些噪声词条的词频差异而非有用词条的词频差异决定。这些噪声完全可能淹没有用信息，从而导致以 TF 为坐标系测度的分类方法精度极低。Joachims Thorsten[17]运用概率理论对 TF-IDF 法进行了理论上的分析和解释，并成功地得到了一种介于传统 TF-IDF 法和朴素贝叶斯法之间的概率 TFIDF 法。在他的实验中，概率 TFIDF 法的分类精度比传统 TF-IDF 法有较大提高。文献[28，29]中的权值调整方法，基于特征的评估函数进行权值调整，也取得了很好的效果，但是它保留了全部特征，一方面导致算法的时间效率降低，另一方面容易出现过拟合。

❑3.2　TEF-WA 权值调整技术

3.2.1　TEF-WA 权值调整的基本思想

文本的表示通常用向量空间模型（Vector Space Model，VSM）[10]表示。文本经过分词处理、词频统计、舍弃词条后，每个文本 $\boldsymbol{x}_i$ 都可映射为由一组词条矢量张成的向量空间一个点，或者说特征空间中的一个规范化的特征向量 $\boldsymbol{x}_i = (ws_{i1}, ws_{i2}, \cdots, ws_{ik}, \cdots, ws_{im})$，其中 ws_{ik} 表示文本 $\boldsymbol{x}_i$ 的第 k 个特征 s_{ik} 的权重。

一般的特征权重调整要经过如下三个步骤。

① 计算每个特征的辨别能力；

② 根据特征的辨别能力，筛选出一定数量的特征；

③ 调整特征的权重，强调辨别能力强的特征，抑制没有辨别能力或辨别能力低的特征。

步骤①的实现通常是构造一种特征评估函数来计算每个特征的辨别能力。常用的评估函数是从信息论中延伸出来的，用于给各个特征词条打分，很好地反映了词条与各类之间的相关程度，如文本频率（Document Frequency）、信息增益（Information Gain）、期望交叉熵（Expected Cross Entropy）、互信息（Mutual Information）、χ^2 统计量（CHI）、单词权（Term Strength）、文本证据权（the Weight of Evidence for Text）和几率比（Odds Ratio）等。特征的辨别能力由评估分的高低来衡量。

步骤②的实现有两种方法：方法一，设置一个评估分阈值，低于该阈值的特征被删除；方法二，设置一个保留特征数阈值，必须先按照特征的评估分排序，保留排在前面的预定数量的特征。这两种方法各有优缺点。方法一的优点是不需要排序算法，时间效率高；缺点是评估分的阈值难以确定，它与评估函数有关，并且随着训练文本集的改变而变化。方法二的阈值比较好确定，缺点是必须按排序评估分排序，即使采用快速排序法，时间复杂度也是 $O(n\log n)$，n 是训练文本集的特征总数。

作者提出了一种改进方法——评估分阈值比率法，综合了二者的优点。评估分阈值比率法先计算出所有特征的评估分的平均值 aver_score，然后设置一个比率阈值 thred_pi，这比方法一指定常数阈值要容易得多，而且不需要方法二中的排序过程，提高了时间效率。

步骤③一般是构造一种权重调整策略。如果没有这一步，就是普通的特征选择。权重调整的目的是突出重要的特征、抑制次要的特征。TF-IDF 权重函数根据特征的逆文本频率 IDF 调整权重。分析 TFIDF 权重函数，逆文本频度 IDF 不能很好地反映特征的重要性，然而使用文本处理中的一些常用的评估函数独立地给每个特征打分，评估分的高低能够很好地代表特征的重要性，因此很自然地想到使用一些常用的评估函数代替逆文本频度 IDF 进行权重调整，这就是作者的 TEF-WA（Term Evaluation Function –Weight Adjustment）权值调整技术的基本思想。新的权值函数称为 TF-TEF 权重函数，TEF（Term Evaluation Function）代表特征评估函数，TF-TEF 权重公式如下：

$$ws_{ik} = \text{TF-TEF}(s_{ik}) = \text{TF}(s_{ik}) \times \text{TEF}(s_{ik}) \quad (3\text{-}2)$$

式中，$\text{TF}(s_{ik})$ 表示文本 x_i 的第 k 个特征的词频。$\text{TEF}(s_{ik})$ 表示常用的评估函数，用于给各个特征词条打分，反映特征词条与各类之间的相关程度。常用的特征评估函数有文档频率（Document Frequency，DF）、信息增益（Information Gain，IG）、期望交叉熵（Expected cross Entropy）、互信息（Mutual Information，MI）、χ^2 统计量（CHI）、文本证据权（Weight of Evidence for Text，WET）和几率比（Odds Ratio）[3][22][27] [102]，下面将详细讨论。

3.2.2　各种评估函数的 TEF-WA 权值调整

特征权重调整根据特征的重要程度赋予其权重，是对特征重要程度的量化，以便后续进行特征选择。量化特征的重要性有多种不同的方法。

特征权重调整中使用的评估函数是从信息论中延伸出来的，用于给各个特征词条打分，反映了词条与各类之间的相关程度。常用的评估函数有文本频率、信息增益、期望交叉熵、互信息、χ^2 统计量（CHI）、单词权、文本证据权和几率比等。已经有很多研究者在特征选择方面进行了大量工作[12, 15-18][26-29]，其中美国卡耐基梅隆大学的 Yang Yiming 教授和斯坦福大学的 Mechran Sahami 的文章较具代表性和总结性[12, 15]。

1．文本频率（Document Frequency，DF）

它是最简单的评估函数，$\text{freq}(s,c)$ 表示特征词 s 在 c 类文本中出现的频率。依据的基本理论假设是 $\text{freq}(s,c)$ 值很小的特征词要么不含有对于类别预测的信息量，要么太少而不足以对分类产生影响，所以可以删去。

$$\text{TEF}_{df}(s) = \text{freq}(s,c) \quad (3\text{-}3)$$

2．信息增益（Information Gain，IG）

在机器学习领域，信息增益是一种衡量特征是否良好的常用指标。信息增益重要的衡量标准是该特征能够为分类系统带来多少信息量，信息量越多，该特征越重要。对某个特征来说，分类系统中出现它和不出现它时信息量将发生变化，其信息量的差值就是该特征给系统带来的信息量。信息论中，所谓信息量就是“熵”。本书使用信息增益特征词 s 打分，所衡量的是特征词

s 在一个文本分类系统中出现或不出现时对分类系统的影响，计算如式（3-4）。

$$\mathrm{TEF}_{\mathrm{InfoGain}}(s)=P(s)\sum\nolimits_j P(c_j\mid s)\log\frac{P(c_j\mid s)}{P(c_j)}+P(\bar{s})\sum\nolimits_t P(c_j\mid \bar{s})\log\frac{P(c_j\mid \bar{s})}{P(c_j)} \tag{3-4}$$

s 表示特征词出现，$\bar{s}$ 表示特征词 s 不出现；$P(s)$表示特征词 s 出现的概率，$P(\bar{s})$ 表示特征词 s 不出现的概率；$P(c_j)$是类 c_j 的先验概率，$P(c_j|s)$是基于 s 的类 c_j 的条件概率。

在用公式（3-1）计算特征 s 的权重时，乘以 $\mathrm{TF}(s)$，即特征 s 的词频，因此，信息增益的计算修改为式（3-5），称信息增益_M（infoGain_M）。

$$\mathrm{TEF}_{\mathrm{InfoGain\_M}}(s)=\sum\nolimits_j P(c_j\mid s)\log\frac{P(c_j\mid s)}{P(c_j)}+\sum\nolimits_t P(c_j\mid \bar{s})\log\frac{P(c_j\mid \bar{s})}{P(c_j)} \tag{3-5}$$

3．期望交叉熵（Expected Cross Entropy，ECE）

期望交叉熵所衡量的是特征词 s 在文本中出现时所获得的信息量，与信息增益不同。

$$\mathrm{TEF}_{\mathrm{CrossEntropy}}(s)=P(s)\sum\nolimits_{j=1}^{|C|}P(c_j\mid s)\log\frac{P(c_j\mid s)}{P(c_j)} \tag{3-6}$$

同样，为去掉式（3-6）中的 $P(s)$，称之为期望交叉熵_M（CrossEntropy_M），计算如式（3-7）所示。

$$\mathrm{TEF}_{\mathrm{CrossEntropy\_M}}(s)=\sum\nolimits_{j=1}^{|C|}P(c_j\mid s)\log\frac{P(c_j\mid s)}{P(c_j)} \tag{3-7}$$

4．互信息（Mutual Information，MI）

互信息（Mutual Information）是信息论里一种有用的信息度量，用于表示信息之间的关系，是两个随机变量统计相关性的测度。互信息的定义与交叉熵近似。使用互信息进行特征抽取基于如下假设：在某个特定类别出现频率高，但在其他类别出现频率比较低的词条，与该类的互信息比较大。通常用互信息作为衡量某个特征词和类别之间的统计独立关系，即某个特征词和类别之间的测度，如果特征词属于该类，它们的互信息量最大，反之较小。由于该方法不需要对特征词和类别之间关系的性质做任何假设，因此非常适合于文本分类中的特征和类别的测度。对特征词 s 和某个类别 c_j 的互信息定

义如下：

$$\mathrm{TEF}_{\mathrm{MutualInfo}}(s)=\sum_{j=1}^{|C|}P(c_j)\log\frac{P(s\mid c_j)}{P(s)} \tag{3-8}$$

使用互信息进行特征选择时效果比较差，原因在于互信息评估函数没有$P(s)$项，即没有考虑特征词发生的频度，致使互信息倾向于选择稀有特征词，删掉了很多有用的高频特征词[103]。这也是互信息与期望交叉熵的本质不同。但如果用互信息进行权值调整，虽然高频特征词加权并不很大，但与其很高的 TF 值相乘后，总的权重不会太低，所以效果大大提高。

5．文本证据权（Weight of Evidence for Text，WET）

文本证据权衡量类的概率和给定特征时类的条件概率之间的差别，它不需要计算 s 的所有可能值，而只考虑 s 在文本中是否出现。

$$\mathrm{TEF}_{\mathrm{weightEvidTxt}}(s)=P(s)\sum_j P(c_j)\left|\log\frac{P(c_j\mid s)(1-P(c_j))}{P(c_j)(1-P(c_j\mid s))}\right| \tag{3-9}$$

与信息增益_M、期望交叉熵_M 类似，文本证据权_M（weightEvidTxt _M）计算如式（3-10）所示。

$$\mathrm{TEF}_{\mathrm{weightEvidTxt\_M}}(s)=\sum_j P(c_j)\left|\log\frac{P(c_j\mid s)(1-P(c_j))}{P(c_j)(1-P(c_j\mid s))}\right| \tag{3-10}$$

6．几率比（Odds Ratio，Odds）

$$\mathrm{TEF}_{\mathrm{OddsRatio}}(s)=\log\frac{P(s\mid \mathrm{pos})(1-P(s\mid \mathrm{pos}))}{P(s\mid \mathrm{neg})(1-P(s\mid \mathrm{neg}))} \tag{3-11}$$

其中，pos 代表正类，neg 代表负类，P(s|pos)表示特征 s 在正类 pos 中出现的概率，P(s|neg)表示 s 在负类 neg 中出现的概率。几率比特别适用于二元分类器。在二元分类中，希望能识别出尽可能多的正类，而不关心识别出负类，这时 Odds Ratio 比其他评估函数有额外的优势。

7．χ^2 统计量

χ^2 统计量（CHI 统计量）用于衡量一个特征词和一个类别之间的关联性，关联性越强，特征得分越高，该特征越应被保留。令 a 为训练集中包含 s 的

c_j 类文本数，b 为训练集中包含 s 的非 c_j 类文本数，d 为训练集中不包含 s 的 c_j 类文本数，e 为训练集中不包含 s 的非 c_j 类文本数，N 为训练集中的总文本数。那么特征词 s 和类别 c_j 之间的 χ^2 统计量定义为：

$$\mathrm{TEF}_{\chi^2}(s)=\sum_j P(c_j)\frac{N(ae-bd)^2}{(a+d)(b+e)(a+b)(d+e)} \qquad (3\text{-}12)$$

❑3.3 实验结果与分析

3.3.1 TEF-WA 权值调整的有效性

实验数据来自易宝中文新闻，包括国际新闻、经济新闻、体育新闻、文教新闻、政治新闻 5 个主题类别的几万篇文档，从中随机选取不同数量的文档组成训练文档子集和测试文档子集。

1．实验数据 1：如表 3.1 所示

训练样本集：_train5000（5000 篇）；测试样本集_check1027（1027 篇）。
特征选择保留：1000（特征选择基于文档频数）。
实验目的：比较评估函数用于特征选择和权值调整的区别。

表 3.1 评估函数用于特征选择和权值调整的比较

分类精度（%） / 评测函数		朴素贝叶斯分类器	
		基于单词频数	基于文档频数
信息增益_M	特征选择	76.32	77.75
	权值调整	**82.75**	78.45
期望交叉熵_M	特征选择	76.32	77.61
	权值调整	**84.32**	**83.62**
互信息	特征选择	36.76	37.06
	权值调整	**84.46**	**84.32**
文本证据权_M	特征选择	76.32	77.51
	权值调整	**84.86**	**85.36**

续表

评测函数 \ 分类精度（%）		朴素贝叶斯分类器	
		基于单词频数	基于文档频数
几率比	特征选择	77.75	76.32
	权值调整	**84.23**	78.35
χ^2 统计	特征选择	77.56	76.43
	权值调整	74.85	77.67

表 3.1 列出了基于各种评估函数单纯的特征选择和 TEF-WA 权值调整技术在朴素贝叶斯分类器上的分类精度。从表 3.1 中可以明显看出，除了结合 χ^2 统计评估函数权值调整，分类精度下降，其他大部分评估函数如期望交叉熵改进型、互信息、文本证据权改进型、几率比、信息增益改进型在权值调整时的分类精度都比特征选择提高了 2%～8%，其中互信息提高得最多，文本证据权改进型的文档模式分类精度最好为 85.36%。由此可见，使用 TEF-WA 权值调整技术结合评估函数进行权值调整，而非单纯的特征选择，能够很有效地提高分类精度。

2．实验数据 2：如表 3.2 所示

训练文档集：_train1000（1000 篇），测试文档集_check115（115 篇）。
实验目的：TF-TEF 权值公式与 TF-IDF 权值公式的实验比较。

表 3.2　TF-TEF 权值公式与 TF-IDF 权值公式的实验比较

分类精度		质心分类		KNN		Naïve Bayesian	
		词频型	文档型	词频型	文档型	词频型	文档型
TF-IDF 权值公式		77.39	80.87	78.26	80	80	80.87
TF-TEF 权值公式	信息增益_M	75.65	71.3	77.39	75.65	**80.87**	**81.74**
	交叉熵_M	**81.74**	**81.74**	**81.74**	**80.87**	**83.48**	**82.61**
	互信息	**82.61**	**84.35**	**80.87**	**82.61**	**80.87**	**82.61**
	文本证据权_M	**82.61**	**82.61**	**79.13**	**82.61**	**82.61**	**81.74**
	几率比	**81.74**	74.78	77.39	79.13	**81.74**	78.26
	χ^2 统计	55.65	69.57	70.43	78.26	73.91	**84.35**

从表 3.2 可以看到：新的 TF-TEF 权值公式结合期望交叉熵改进型、互信息、文本证据权、几率比进行权值调整比 TF-IDF 权值公式的分类精度要高（粗体数字），提高了 1%～5%，其中，结合互信息、文本证据权的基于词频型在质心分类器上分类精度提高了 5.22%。基于文档型的 χ^2 统计在 Naïve Bayesian 分类器上分类精度达到了 84.35%，基于词频型的期望交叉熵改进型在 Naïve Bayesian 分类器上的精度也很高，为 83.48%。

3．实验数据 3：如表 3.3 所示

训练文档集：_train5000（5000 篇），测试文档集：_check238（238 篇）。
实验目的：TF-TEF 权值公式与 TF-IDF 权值公式的实验结果比较。

表 3.3　TF-TEF 权值公式与 TF-IDF 权值公式的比较

分类精度		质心分类		KNN		Naïve Bayesian	
		词频型	文档型	词频型	文档型	词频型	文档型
TF-IDF 权值公式		78.15	81.93	81.93	82.35	83.19	83.19
TF-TEF 权值公式	信息增益_M	75.65	71.3	81.51	80.67	**86.97**	81.93
	交叉熵_M	**82.35**	**83.19**	**84.45**	**84.03**	**86.55**	**87.39**
	互信息	**82.35**	**84.03**	**85.71**	**84.45**	**87.39**	**86.97**
	文本证据权_M	**83.19**	**82.35**	**85.71**	**85.29**	**87.39**	**87.39**
	几率比	**84.03**	76.89	**84.03**	**82.35**	**89.08**	**86.97**
	χ^2 统计	65.55	71.85	76.05	81.93	73.53	**87.39**

从表 3.3 可以看出，TF-TEF 函数结合几率比的词频型的最适合用于 Naïve Bayesian 分类器的权值调整，分类精度为 89.08%，比 TF-IDF 权值函数提高了 6%。基于文档型的 χ^2 统计也很适合用于 Naïve Bayesian 分类器的权值调整，分类精度为 87.39%，比 TF-IDF 权值函数提高了 6%。期望交叉熵改进型、互信息、文本证据权在三种分类器上权值调整的效果都比较好，对比 TF-IDF 权值公式分类精度提高了 2%～4%（如表 3.3 中粗体数字所示）。

4．实验数据 4：如图 3.1 所示

训练文本集：_train3000b（3000 篇）；测试文本集：_check115（115 篇）。
分类算法：朴素贝叶斯、质心分类、K 近邻分类。

权值调整：TF-TEF 权值公式结合几率比，基于词频模式。

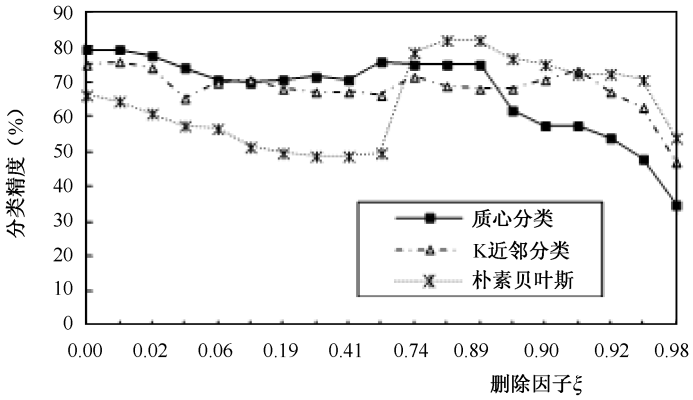


图 3.1　结合几率比权值调整后的分类效果

在图 3.1 中，横坐标表示删除因子 ξ，纵坐标表示相应删除因子 ξ 下的分类算法的分类精度。图 3.1 表示的是采用基于词频型的几率比（Odds Ratio）进行权重调整，质心分类算法、K 近邻分类算法、朴素贝叶斯（Naïve Bayesian）分类算法的分类精度和分类速度随着删除因子 ξ 的变化趋势。

分析图 3.1 所示分类精度随删除因子的变化趋势，在删除因子为 0%～0.1%，几乎保留全部特征的情况下，质心分类算法和 K 近邻分类算法的分类精度最高，分别为 79.13%和 75.65%，随着删除因子的增大，分类精度都逐渐下降；而朴素贝叶斯在保留全部特征情况下，分类精度反而很差，只有 64.35%～66.09%，删除因子达到 90%时，即保留 10 %的特征时，分类精度最高为 81.74%。折中考虑分类精度和算法时间复杂度、存储空间复杂度，选择 10%左右的特征，质心分类精度为 74.78%，K 近邻分类精度为 73.04%，朴素贝叶斯最高为 81.74%。

这说明 TF-TEF 权值公式结合几率比评估函数，进行权值调整，设置评估分阈值比率 thred-pi 降维，对朴素贝叶斯分类算法来说不仅可以实现非常有效的降维（$\xi = 0.9$），还可以大幅提高分类精度，$\xi = 0.9$ 比 $\xi = 0$ 的分类精度提高了 15.65%，既降低了算法的计算复杂度又提高了分类精度。

对于权值计算公式 TF-TEF 与 TFIDF 权值公式的比较，从实验数据表 3-2 和表 3-3 可以明显地看到二者的区别，实验数据表 3-2 和表 3-2 的数据纵向

比较，期望交叉熵_M 型、互信息、文本证据权_M 的词频模式和文档模式在三种分类方法上都有明显的提高，信息增益_M 型、几率比、χ^2 统计在部分分类器上有改进，有的反而下降。χ^2 统计的文档模式、期望交叉熵的词频模式对朴素贝叶斯分类器非常有效，分类精度分别为 84.35%和 83.48%，对比 TFIDF 提高约 4%；互信息的文档模式在质心分类器上的精度为 84.35%，提高约 4%。

从时间效率上分析，特征选择需要先按评估分数排序，即使使用快速排序法，时间复杂度是 $O(n\log_2 n)$，n 是初始特征空间的特征词条数。使用 TEF-WA 权值调整技术和阈值比率法，不需要按照特征的评估分排队，没有排序的过程，TF-TEF 权值公式中蕴含了评估函数 $\mathrm{TEF}(s_k)$，只需按照 TF-TEF 公式（3-2）计算权重，就提高了时间效率，时间复杂度为 $O(n)$。但是 TEF-WA 权值调整技术并不保留全部的特征，而是通过评估分阈值比率法，设置一个评估分阈值比率 thred_pi，这比指定常数阈值要容易得多，而且不需要排序过程，提高了时间效率；同时，减少了下一步分类学习算法要处理的特征数，也达到了降维的目的。

更重要的是使用 TEF-WA 权值调整技术筛选出的特征，对文本分类贡献大（即辨别能力强）的特征赋予的权重大，对分类影响小的特征赋予的权重小，并进一步调整它们在分类算法中的作用，权重大的特征在分类算法中的作用也得到了加强，权重小的特征在分类算法中的作用相对较弱，权重是 0 的就不起作用了。

TEF-WA 权值调整技术无论在时间效率还是在提高分类精度上都要优于特征选择，尤其是在分类精度的提高方面。

3.3.2 不同评估函数的权值调整

本节在不同规模的训练文档集上，采用信息增益改进型、期望交叉熵改进型、互信息、χ^2 统计（CHI）、文本证据权和几率比等不同的评估函数进行权值调整，计算公式分别采用词频型和文档型两种方法，运行质心分类算法、K 近邻算法（KNN）、朴素贝叶斯分类算法，进行交叉实验，从不同的角度进行了实验评估。

1．基于信息增益的实验效果

训练文档集：_train3000a。初始特征总数：34755。

测试文档集：_check115。评估函数：信息增益_M。

基于词频型的实验结果如图 3.2 所示，基于文档型的实验结果如图 3.3 所示。

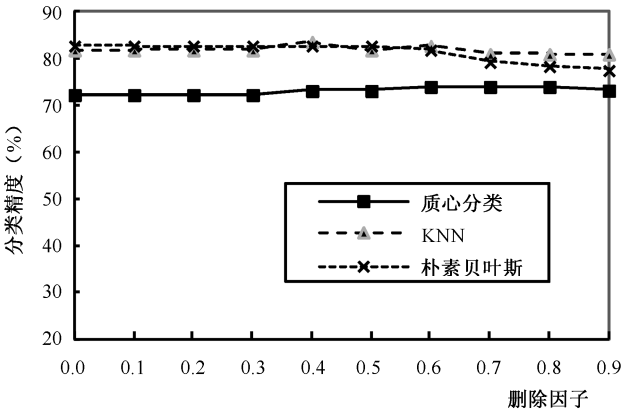


图 3.2　基于词频型的信息增益_M 的分类结果

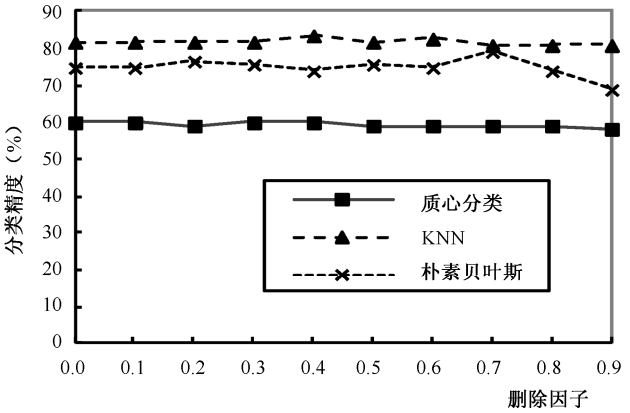


图 3.3　基于文档型的信息增益_M 的分类结果

在图 3.2 和图 3.3 中，横坐标表示删除因子ξ，纵坐标表示相应删除因

子ξ下的分类算法的分类精度。

从图 3.2 和图 3.3 中的数据可以看到以下几点。

① 在使用信息增益_M 进行权值调整时，比较三种分类算法，KNN 分类算法的分类精度最高，其次是 Naïve Bayesian 分类算法，最差的是质心分类算法。

② 信息增益_M 的最大的特点是：随着删除因子ξ的增大，分类精度的下降趋势较为平缓，即随着特征维数的大幅降低，仍然能保持较高的分类精度。因此信息增益_M 评估函数最常用来降维。

③ 文档型与词频型比较，质心分类和 Naïve Bayesian 的词频型明显优于文档型，KNN 的词频型和文档型相差不大。

2．基于期望交叉熵的实验效果

训练文档集：_train3000a（2811 篇）；初始特征总数：34 755。

测试文档集：_check115（115 篇）。评估函数：期望交叉熵_M 型。

基于词频型的实验结果如图 3.4 所示，基于文档型的实验结果如图 3.5 所示。

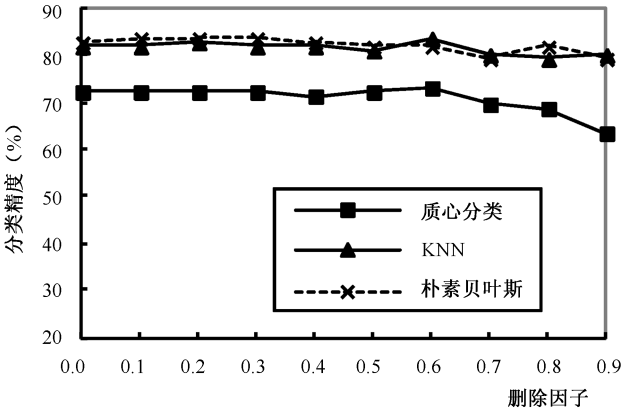


图 3.4　基于词频型的期望交叉熵_M 实验结果

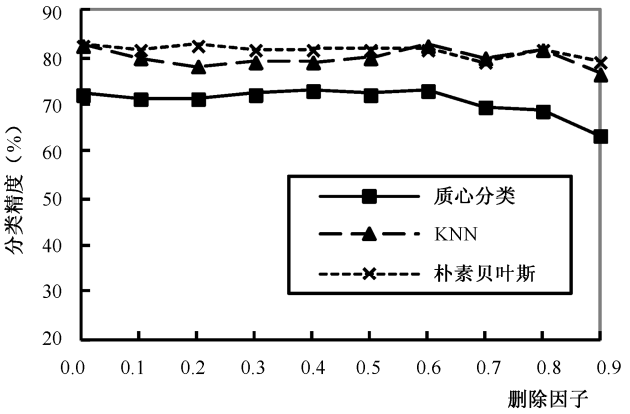


图 3.5　基于文档型的期望交叉熵_M 实验结果

在图 3.4 和图 3.5 中，横坐标表示删除因子ξ，纵坐标表示相应删除因子ξ下的分类算法的分类精度。在训练文档集_train3000 上，分别以基于词频型和文档型的期望交叉熵_M 型进行权重调整，在不同删除因子ξ下，显示质心分类算法、K 近邻分类算法、Naïve Bayesian 分类算法的分类精度。

分析图 3.4 和图 3.5 中的数据，可以看到以下几点。

① 在使用期望交叉熵_M 型进行权值调整时，比较三种分类算法，在删除因子$\xi = 0.6$时，KNN 算法的分类精度最高，词频型为 83.48%，而文档型为 82.61%；Naïve Bayesian 分类算法的词频型为 81.74%，文档型为 81.74%；分类精度最低的是质心分类算法，词频型为 72.17%，文档型为 72.17%；当$\xi \geqslant 0.8$时，三种分类算法的精度都有损失，但是 Naïve Bayesian 分类算法和 KNN 分类算法损失不大，而且 Naïve Bayesian 的分类精度超过了 KNN，质心分类的精度下降较大。因此可以说，Naïve Bayesian 分类算法分类最好，其次是 KNN 分类算法，最差的是质心分类算法。

② 文档型与词频型比较，Naïve Bayesian 分类算法和 KNN 分类算法的词频型要优于文档型，分类精度高出 2%左右，质心分类的词频型和文档型相差不大。

总的来说，期望交叉熵_M 型是一种比较好的权值调整函数。

3．基于互信息的实验效果

训练文本集：_train3000a（2811 篇）。初始总特征：34 755。

测试文本集：_check115 篇。评估函数：互信息、词频型。

基于词频型的实验结果如图 3.6 所示，基于文档型的实验结果如图 3.7 所示。

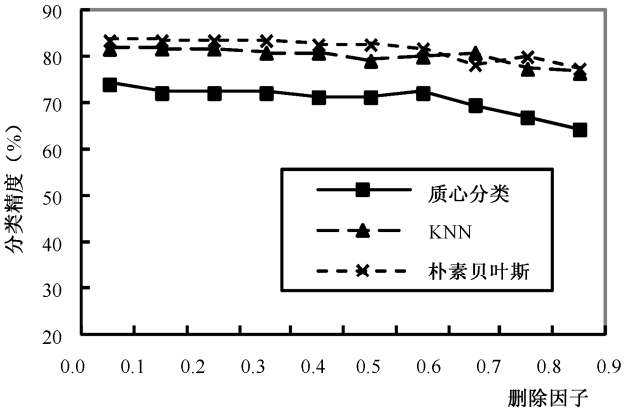


图 3.6　基于词频型的互信息实验结果

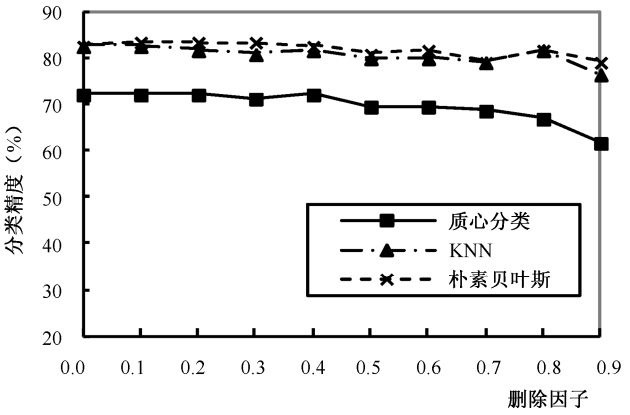


图 3.7　基于文档型的互信息实验结果

在图 3.6 和图 3.7 中，横坐标表示删除因子ξ，纵坐标表示相应删除因子ξ下的分类算法的分类精度。图 3.6 和图 3.7 分别表示基于词频型和文档型的互信息进行权重调整，质心分类算法、K 近邻分类算法、Naïve Bayesian 分类算法的分类精度和分类速度随删除因子ξ的变化趋势。

从图 3.6 和图 3.7 中可以看出，使用互信息作为评估函数权值调整时，无论是基于文档型还是基于词频型，在ξ<0.5 时，Naïve Bayesian 算法的分类精度比较高，为 82.61%～83.48%，KNN 的分类精度为 80%～82.61%，质心分类算法的精度为 70%～74.04%。但随着ξ的继续增大，即保留特征的减少，三种分类算法的分类精度都开始下降，质心分类的精度下降得比较快，Naïve Bayesian 算法和 KNN 算法下降缓慢。

这与期望交叉熵_M、文本证据权_M 类似，和信息增益_M、几率比不同。

4．基于几率比的实验效果

训练文本集：_train 3000b（共 2911 篇）。总特征数：35 612。

测试文本集：_check115 篇（115 篇）。评估函数：Odds Ratio。

基于词频型、基于文档型的实验结果分别如图 3.8 和图 3.9 所示。

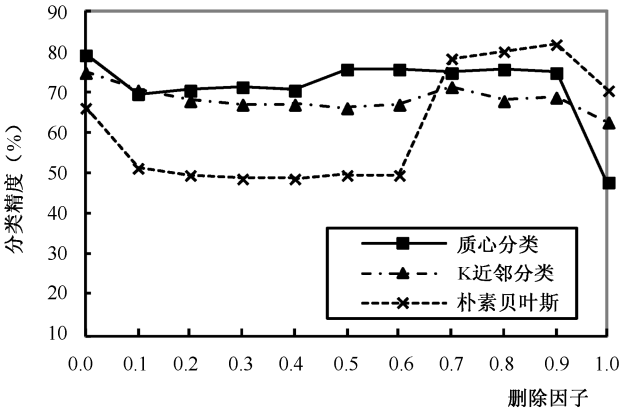


图 3.8　基于词频型的几率比实验结果

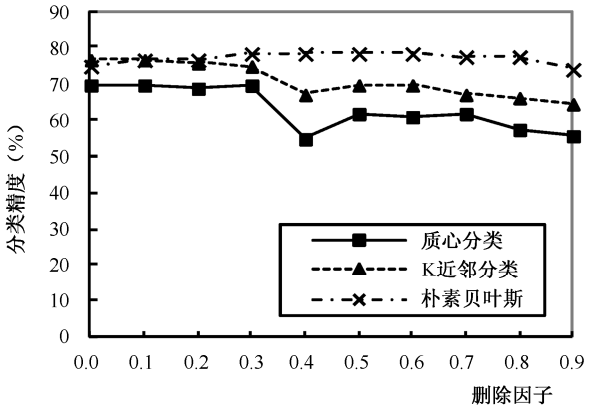


图 3.9　基于文档型的几率比实验结果

在图 3.8 和图 3.9 中，横坐标表示删除因子ξ，纵坐标表示相应删除因子ξ下的分类算法的分类精度。图 3.8 和图 3.9 表示的是在训练文档集_train3000b 上，分别以基于词频型和文档型的几率比进行权重调整，质心分类算法、K 近邻分类算法、Naïve Bayesian 分类算法的分类精度和分类速度随删除因子ξ的变化趋势。

使用几率比权值调整时，基于词频型的几率比（见图 3.8），质心分类和 KNN 分类算法在$\xi=0$时，即保留全部特征情况下，二者分类精度最高，分别为 79.13%和 74.78%，随着ξ的增大，分类精度呈下降趋势。Naïve Bayesian 分类算法则相反，在$\xi=0$时，分类精度反而很差，只有 66.09%，在$\xi>0.6$时，分类精度呈上升趋势，在$\xi=0.1$时，即仅保留 10%左右的特征时，分类精度最高，达 81.74%。当删除因子$\xi=0.95$时，Naïve Bayesian 仍有很高的分类精度 70.43%，而 K 近邻的分类精度已降为 62.61%，质心分类的精度已经低于 50%，为 47.83%。

因此，基于词频型的几率比很适合用于 Naïve Bayesian 算法，不仅可以实现非常有效的降维（$\xi=0.90$），还可以大幅提高分类精度，$\xi=0.90$时比$\xi=0$时的分类精度提高了 15.65%。

5．基于文本证据权的实验效果

文本证据权_M 型计算公式如 3.3.2 节中式（3-10）所示。

训练文档集：_train3000a。初始总特征数：34 755。

测试文档集：_check115。权值调整函数：文本证据权_M。

基于词频型、基于文档型的实验结果分别如图 3.10 和图 3.11 所示。

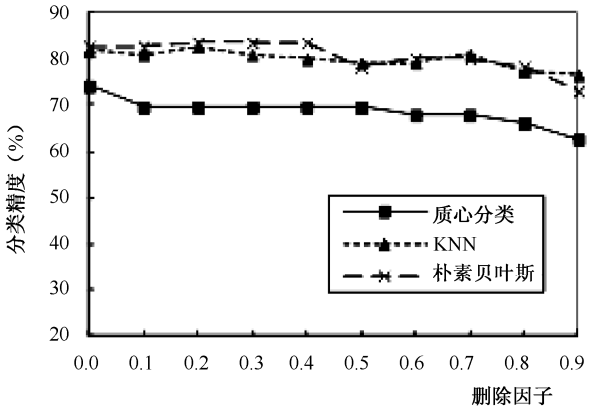


图 3.10　基于词频型的文本证据权_M 实验结果

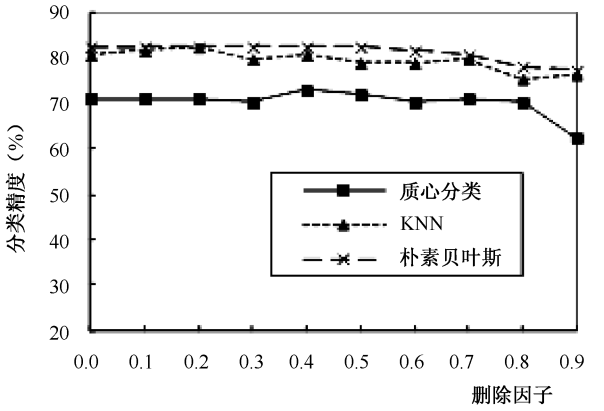


图 3.11　基于文档型的文本证据权_M 实验结果

在图 3.10 和图 3.11 中，横坐标表示删除因子ξ，纵坐标表示相应删除因子ξ下的分类算法的分类精度。图 3.10 和图 3.11 表示的是在训练文档集_train3000a 上，分别以基于词频型和文档型的文本证据权_M 进行权重调整，质心分类算法、K 近邻分类算法、Naïve Bayesian 分类算法的分类精度随删

除因子ξ的变化趋势。

从图 3.10 和图 3.11 可以看出以下几点。

① 使用文本证据权_M 作为评估函数权值调整时，无论是基于文档型还是词频型，在$\xi<0.40$时，Naïve Bayesian 算法的分类精度比较高，为 82.61%～83.48%，KNN 的分类精度为 80%～82.61%，质心分类的精度为 70%～74.04%。但随着ξ的继续增大，即随着保留特征的减少，三种分类算法的分类精度都开始下降，质心分类的精度下降得比较大，Naïve Bayesian 算法和 KNN 算法下降缓慢。

这与期望交叉熵_M、互信息类似，但与信息增益_M、几率比不同。

② 文档型与词频型相比，Naïve Bayesian 的词频型稍好于文档型，KNN 的文档型和词频型相差不多。

总之，文本证据权_M 是一种很好的权值调整的评估函数。

6．基于χ^2统计（CHI）的实验效果

训练文档集：_train3000a（2811 篇）。总特征数：34 755。

测试文档集：_check115（115 篇）。评估函数：χ^2统计。

基于词频型、基于文档型的实验结果分别如图 3.12 和图 3.13 所示。

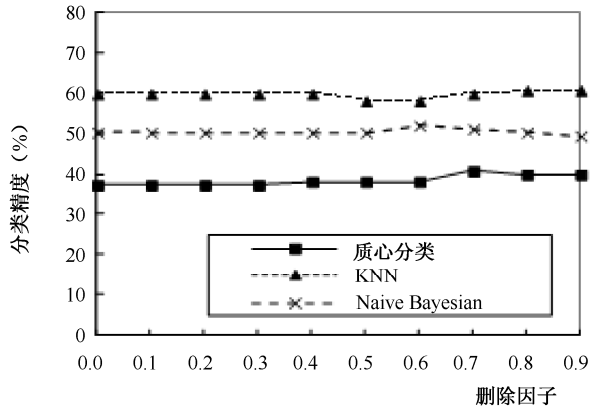


图 3.12　基于词频型的 CHI 实验结果

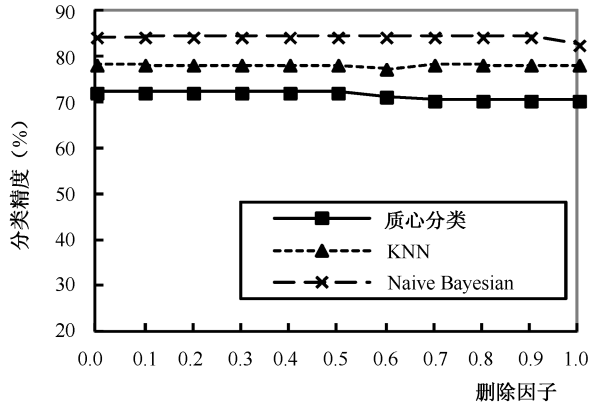


图 3.13　基于文档型的 CHI 实验结果

在图 3.12 和图 3.13 中，横坐标表示删除因子ξ，纵坐标表示相应删除因子ξ下的分类算法的分类精度。图 4-11 和图 4-12 分别表示在训练文档集_train3000a 上，以基于词频型和文档型的χ^2统计进行权重调整，质心分类算法、K 近邻分类算法、Naïve Bayesian 分类算法的分类精度随删除因子ξ的变化趋势。

对比图 3.12 和图 3.13，可以看出以下几点。

① χ^2统计的文档型明显优于词频型。

② 从图3.13的走势中，可以看到基于文档型的χ^2统计在Naïve Bayesian分类算法上的分类效果明显比质心分类算法、K 近邻分类算法好，随着训练样本规模的增大更为明显。Naïve Bayesian 分类算法的分类精度为 84.35%，比质心分类算法的 70.34%高出 14%，比 K 近邻分类算法的 78.26%高 6%，而且降维效果也很好，删除因子$\xi = 0.90$。

总的来说，使用χ^2统计时，应当采用文档型计算公式，基于文档型的χ^2统计很适合用于 Naïve Bayesian 分类算法，但是不太适合用于 K 近邻分类算法和质心分类算法。

3.3.3 评估比较

1．从分类精度方面比较

从前面的实验结果看，总的来说，Naïve Bayesian 分类器的分类精度最高，其次是 K 近邻分类器，最差的是质心分类器。

2．从分类速度方面比较

Naïve Bayesian 分类器的分类速度最快，其次是 K 近邻分类器，最慢的是质心分类器，见图 3.14。删除因子低于 0.05 时，K 近邻分类器的速度比质心分类器的速度慢，随着特征删除因子的增大，K 近邻分类器的速度明显比质心分类器快，Naïve Bayesian 的速度提高得更为显著。在删除因子为 0.50，即保留 50%的特征词时，质心分类器的分类速度为 8 个/秒，K 近邻分类器的速度为 14 个/秒，Naïve Bayesian 分类器的速度为 28 个/秒；当删除因子为 0.96 时，即保留 4%的特征词时，K 近邻分类器的速度为 16 个/秒，Naïve Bayesian 分类器的速度为 38 个/秒。图 3.14 以期望交叉熵改进型为例，使用其他评估函数进行权值调整时三种分类方法的速度也如此。[①]

在以往的研究文献中，K 近邻分类一直是效果很好的分类方法，其缺陷是速度较慢。鉴于此，在 K 近邻算法的实现上采取了一些技巧，不是把测试文本直接和训练集合中的每个样本计算相似度，而是把测试文本中的每个单词与此单词出现过的文本进行计算，大大提高了运算速度，使得系统对于数量超过数万级的训练文本、数千级的测试文本，都能在合理时间内完成分类任务。

另外，传统的 K 近邻分类只是挑出 K 个最相似的样本，然后求属于同一类的样本数，哪一类样本数最多就选取哪一类。在算法实现上则把属于同一类的样本与测试文本的相似度分值进行数学处理后相加，最后属于哪一类的相似度分值之和最大，就选取哪一类。实验比较发现，数学处理时取开方效

① 说明：第 3 章中的内容是作者在清华大学读硕士学位期间的研究工作，实验时间是 2002—2003 年，地点是清华大学人工智能国家重点实验室，使用的计算机的计算速度远远不及现在。第 4～7 章的内容是作者在大连海事大学读博士学位时的研究工作，时间为 2006—2009 年。

果较好。还实现了 K 近邻的增量学习，即当已经对大量文本训练完毕并生成了一个分类模型后，如果又获得了少量新的训练文本，则无须从头训练，只要在原来模型的基础上进行增量学习，把少量新训练文本的信息归并到原来的模型中即可。增量学习方法在实际应用中可以有效提高 K 近邻分类的速度。

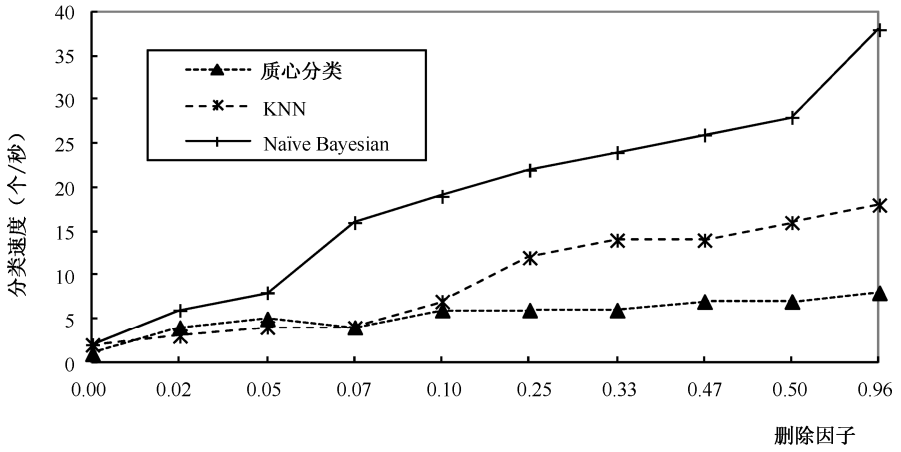


图 3.14　分类速度比较

3．从评估函数的效果方面比较

关于评估函数之间的对比，以前已有研究者做了许多工作。但他们的实验是在特征选择的背景下进行的，而且背景不同，结果也不尽相同。在 Yang 和 Pedersen 的实验[16]中，信息增益是最好的测度之一。而在 Dunja Mladenic 的实验[15]中，几率比是最好的测度，交叉熵和单词频度 TF（s）是较好的，较差的是互信息，最差的是信息增益。二者的学习算法和对数据域定义的不同可能是出现这种差异的原因。Yang 采用的学习算法是 K 近邻算法和线性最小方差匹配，而 Dunja Mladenic 用的是 Naïve Bayesian。在类别体系的定义上，Yang 采取的是平面文本分类，使用具有多个类值的一个分类器。而 Dunja Mladenic 采取的是等级文本分类，将数据域划分成许多子问题，每个子问题对应一个只有 2 类值的分类器。

在文本分类系统 SECTCS 实验平台上，运行质心分类、K 近邻和 Naïve

Bayesian 三种分类算法，分别采用词频、文档型计算方法，对各种评估函数在 TEF-WA 权值调整技术中的效果进行了综合比较，实验结果如图 3.15～图 3.20 所示。

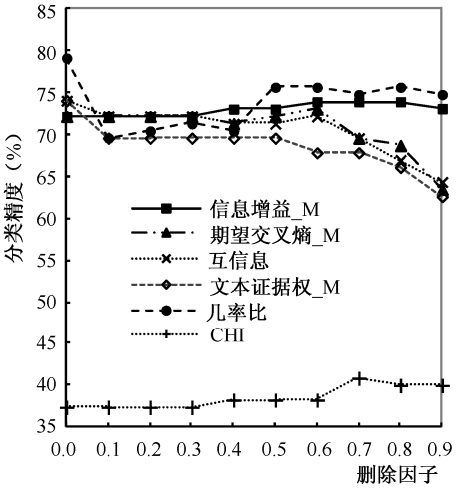


图 3.15　词频型的各种评估函数在质心分类上的比较

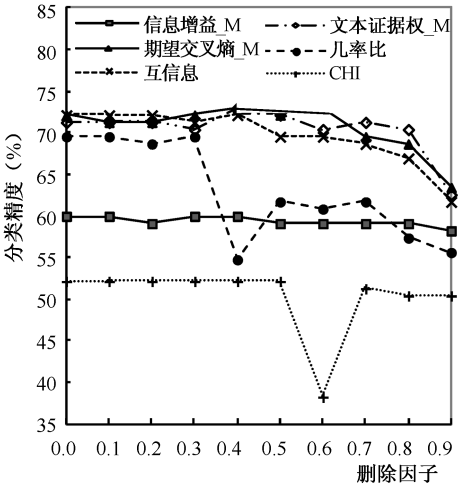


图 3.16　文档型的各种评估函数在质心分类上的比较

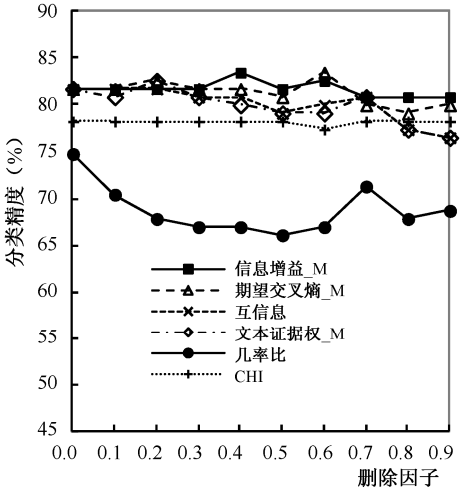


图 3.17　词频型的各种评估函数在 KNN 上的比较

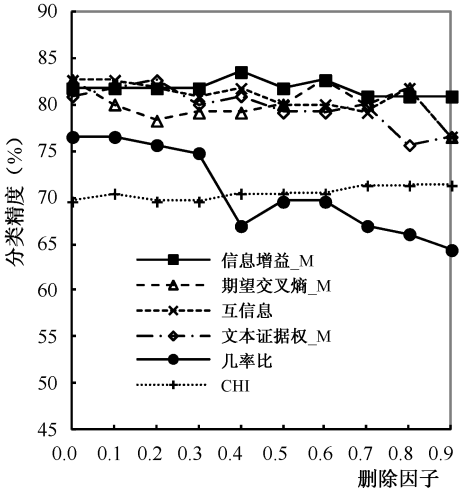


图 3.18　文档型的各种评估函数在 KNN 上的比较

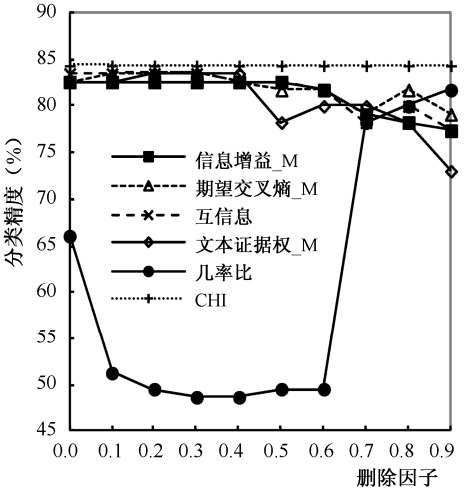


图 3.19　词频型的各种评估函数在 Naïve Bayesian 上的比较

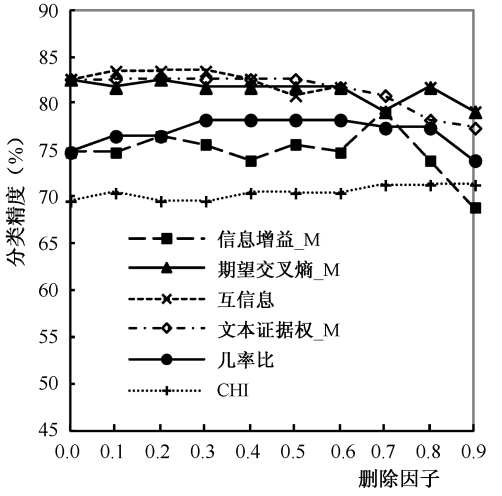


图 3.20　文档型的各种评估函数在 Naïve Bayesian 上的比较

图 3.15～图 3.20 中表示的是基于词频型和文档型的信息增益_M、期望交叉熵_M、互信息、文本证据权_M、几率比、χ^2 统计（CHI）六种评估函数在质心分类器、KNN 分类器和 Naïve Bayseian 分类器上的综合比较。训练

文档集为_train3000a，测试文档集为_check115。综合分析评估如下所述。

（1）质心分类器上各种评估函数的比较

如图 3.15 和图 3.16 所示，基于词频型计算公式，对质心分类器来说，最好的评估函数是几率比和信息增益_M，再依次是期望交叉熵_M、互信息和文本证据权_M，最差的是 χ^2 统计（CHI）；基于文档型计算公式，对质心分类器来说，最好的评估函数是文本证据权_M 和期望交叉熵_M，再依次是互信息、信息增益_M、几率比，最差的也是 χ^2 统计（CHI）。

对比图 3.15 和图 3.16，信息增益_M、几率比的词频型优于文档型，互信息、期望交叉熵_M 的词频型和文档型相差不大。

（2）KNN 分类器上各种评估函数的比较

如图 3.17 和图 3.18 所示，基于词频型计算公式，对 KNN 分类器来说，最好的评估函数是信息增益_M 和期望交叉熵_M，再依次是 χ^2 统计（CHI）、互信息和文本证据权_M，最差的是几率比；基于文档型计算公式，对 KNN 分类器来说，最好的评估函数是信息增益_M、期望交叉熵_M 和互信息，再依次是文本证据权_M、χ^2 统计（CHI），几率比的效果最差。

对比图 3.17 和图 3.18，期望交叉熵_M、几率比和 χ^2 统计（CHI）的词频型优于文档型，信息增益_M、互信息和文本证据权_M 的文档型和词频型相差不大。

（3）Naïve Bayseian 分类器上各种评估函数的比较

如图 3.19 和图 3.20 所示，基于词频型计算公式，对 Naïve Bayseian 分类器来说，最好的评估函数是 χ^2 统计（CHI）和几率比，再依次是期望交叉熵_M、互信息和信息增益_M，比较差的是文本证据权_M；基于文档型计算公式，对 Naïve Bayseian 分类器来说，最好的评估函数是互信息、期望交叉熵_M，再依次是文本证据权_M、几率比和信息增益_M，比较差的是 χ^2 统计（CHI）。

对比图 3.19 和图 3.20，χ^2 统计（CHI）和信息增益_M 的词频型很明显优于文档型，期望交叉熵_M、互信息和文本证据权_M 的文档型和词频型相

差不大，几率比在删除因子ξ＞0.7 时词频型优于文档型。

总的来说，可以得出如下结论。

① 基于词频型的χ^2统计（CHI）、基于词频型和文档型的几率比（Odds ratio）都很适合用于 Naïve Bayesian 分类算法，但是在质心分类器和 K 近邻分类器上的效果比较差。

② 期望交叉熵_M、互信息在 Naïve Bayesian 分类器、K 近邻分类器和质心分类器上的权值调整的效果都比较好，对比 TFIDF 权值公式，分类精度提高了 2%～4%。

③ 基于词频型的信息增益_M 很适用于特征选择，随着删除因子ξ的增大，类精度的损失很小。

④ 词频型或文档型的文本证据权_M 在删除因子ξ<0.6 时效果也不错，但是随着删除因子的增大，分类精度损失比较大（与信息增益_M 和几率比不同）。

随着训练样本集规模的增大，三种分类算法的精度都有提高（如图 3.15 和图 3.20 所示）。

❑3.4 本章小结

特征选择是文本分类的关键步骤，特征选择的好坏直接影响分类学习的结果。本章首先分析了目前特征选择中存在的问题，然后提出了一种结合评估函数的 TEF-WA（Term Evaluation Function –Weight Adjustment）权重调整技术，设计了一种新的权值函数，即将某种特征评估函数 TEF（Term Evaluation Function）引入权值函数中，称为 TF-TEF 权值函数。分析比较了文档频率（Document Frequency，DF）、信息增益（Information Gain，IG）、期望交叉熵（Expected cross Entropy）、互信息（Mutual Information，MI）、χ^2统计量（CHI）、文本证据权（Weight of Evidence for Text，WET）和几率比（Odds Ratio）等各种评估函数，结合不同评估函数的 TEF-WA 权值调整技术在质心分类、K 近邻（KNN）和朴素贝叶斯（Naïve Bayesian）三种分类算法中的效果进行了实验比较。对比实验结果表明 TEF-WA 权值调整技术在提高分类精度和降低算法的计算复杂度方面都是有效的。

第 4 章

结合 TEF-WA 技术的 Co-training 改进算法

❑4.1 Co-training 算法及其存在的问题

Co-training 算法[47-49]是半监督学习中比较有代表性的算法，最早由 Blum 和 Mitchell 提出，假设数据集可以被自然地分成两个独立的特征子集（视图），每个子集都包含足够多的信息进行分类学习，在每个视图上建立各自的分类器，两个分类器每次互相标记一部分置信度高的样本给对方，重新训练，迭代，直到没有更多适合的未标注样本加入，如表 4.1 所示。

Blum 和 Mitchell 将 Web 文本集分成两个视图，一个视图 V_1 由 Web 网页上的单词组成，另一个视图 V_2 由指向其他网页的超链接的单词组成。任意样本 x 可以用一个三元组（x_1，x_2，c）表示，这里，x_1 和 x_2 是 x 在两个视图中的描述，c 是它的类别。Co-training 算法要求两个特征视图满足以下假设。① 一致性（compatible）：即样本的分布与分类的目标函数是一致的，也就是说，对大多数样本 x：$f^1(x_1)=f^2(x_2)=f(x)$，目标函数在每个特征子集上预测的类别是完全相同的。② 独立性（uncorrelated）：表示对指定类别的任意的样本（x_1，x_2，c），$P(x_1 \mid f(x),x_2)=P(x_1 \mid f(x))$，也就是说，样本 x_1 和 x_2 在两个视图中的描述是独立的[47-49][56]。

表 4.1　Co-training 算法

Input：标注文本集 D^l，未标注文本集 D^u，视图 V_1 和 V_2 分类学习算法 f，迭代次数 k。

Loop　for k iterations

（1）在特征视图 $V_1(D^l)$和 $V_2(D^l)$上分别建立分类器 f^1 和 f^2；

（2）For each class c_j do

将由分类器 f^1 和 f^2 标注的 c_j 类且具有最大置信度的未标注文本 du_1 和 du_2 添加到 D^l，并从 D^u 中删除。

Output：组合的分类器 $f(x)=f^1(x)+f^2(x)$。

实际上由于多种原因，这两个假设并不能完全严格地满足，尤其是独立性，甚至在许多实际应用中不存在自然的视图分割方法。另外，直接判断两个视图是否满足独立性也有一定的难度。如何把一个特征集合分割成两个或多个一致的、独立的特征子集是一个有趣的、亟待解决的问题。Nigam 和 Ghani 人工随机将特征集合分割成两个子集[49]。Goldman 和 Zhou[50]使用两个分类器在一个特征集进行共同学习。Zhou 和 Goldman 在文献[51]中提出，在一个视图上训练多个分类器共同学习，根据最大加权投票决定未标注样本的类别[51]，类似地，Zhou Z-H 和 Li 提出了 tri-training 算法[54]。这些都放松了 Co-training 算法的假设约束。为此，本章提出基于 TEF-WA（Term Evaluation Function-Weight Adjustment）技术[103]创建特征多视图，训练产生多个基分类器，通过评估两个基分类器之间的差异性，间接评估二者的独立性的方法。在此基础上提出了两种改进算法：TV-SC 算法和 TV-DC 算法。

❑4.2　基于 TEF-WA 的特征多视图

4.2.1　TEF-WA 技术

文本通常用向量空间模型（Vector Space Model，VSM）[10]表示。文本经过分词处理、词频统计、舍弃词条之间的顺序，每个文本 x_i 都可映射为由一组词条矢量张成的向量空间一个点，或者说特征空间中的一个规范化的特征向量 $\boldsymbol{x}_i=(ws_{i1},ws_{i2},\cdots,ws_{ik},\cdots,ws_{im})$，其中 ws_{ik} 表示文本 x_i 的第 k 个特征 s_{ik} 的权重，它通常是词频的某个函数。

特征选择（Feature Selection）和权值调整（Weight Adjustment）是文本分类的关键步骤，传统的分类算法常采用 TF-IDF 函数[3][23]来计算特征的权重，但该函数难以从文本数据中区分出有用词条和噪声词条，本书采用 TEF-WA 技术[103]来调整特征的权重。TEF-WA 权值调整技术利用信息论中常用的评估函数代替逆文本频率（IDF）给每个特征独立地打分，评估分的高低能够很好地代表特征的重要性，根据评估分调整特征的权重。特征权重计算 $ws_{ik}=\text{TF-TEF}(s_{ik})=\text{TF}(s_{ik})\times\text{TEF}(s_{ik})$（见第 3 章的式（3-2））。文档频率、信息增益、期望交叉熵、互信息、χ^2 统计量（CHI）、文本证据权和几率比等评估函数的计算公式与分析比较见第 3 章的式（3-3）至式（3-12）。

4.2.2　基于 TEF-WA 的特征多视图

本章使用评估函数不仅是为了调整权重，更重要的目的是利用 TEF-WA 技术创建不同的特征视图。根据上述评估函数，在同一个训练集上构造不同的特征视图。每个视图包含足够的信息训练生成一个较好的分类器，而且这些视图之间在一定程度上是一致的（compatible）和独立的（independent）。

令 D' 表示带类别标注的训练文本集，V_i 表示根据第 i 个评估函数建立的特征视图。

$$V_i=\text{create\_view}(\text{TEF},\text{TForDF},m) \tag{4-1}$$

其中，TEF 表示特征评估函数，TForDF 表示特征权重计算时使用的是词频型（TF）公式还是文档型（DF）公式[3]，m 表示保留特征数量或比率。

计算特征词的权值时，有词频型（TF）公式与文档频数型（DF）公式两种方法。文档频数型公式也叫做布尔型，顾名思义，这种公式不考虑一个特征词在一个文本中到底出现了多少次，而只考虑此特征词在此文本中是否出现过，如果出现了，则值为 1，否则为 0。词频型公式则具体考虑特征词在文本中出现的次数。举例来说，求一个类的先验概率时，如果用词频型公式，应该用此类所有特征词发生总数除以训练集合所有特征词发生总数，而如果用文档频数型公式，则应该用此类所含训练文本数除以训练集合中所有文本数。

任意标注样本 x 可以表示为：$(x_1, x_2, \cdots, x_i, \cdots, c)$，$x_i$ 代表 x 在第 i 个视图 V_i 上的描述，c 是它的类标签。而且 $f^1(x_1)=f^2(x_2)=\cdots=f(x)=c$。可以

选择一对特征视图 V_i 和 V_j，二者在一定程度上条件独立，即 $P(x_i \mid c, x_j) = P(x_i \mid c),\ i \neq j$ 。

例如，对应式（4-1）中参数 TEF、TForDF 和 m，如果选择使用评估函数 MI，文本型公式，保留 1000 个特征词建立的特征视图，用 MI/DF/1000 表示建立该视图的参数，也可以这样说，MI/DF/1000 对应了一个特征视图。而 Odds/TF/900 代表使用评估函数 Odds，词频型公式，保留 900 个特征词建立的另一个特征视图。当选择不同的 TEF、TForDF 和 m 时，就建立了不同的特征视图。那么，对同一个文本 x_i 来说，它的特征向量 $(ws_{i1}, ws_{i2}, \cdots, ws_{ik}, \cdots, ws_{im})$ 在不同的视图上将会是不同的。每个视图包含足够多的信息训练生成一个较好的分类器，同一种分类算法在不同的特征视图上训练学习生成的分类器也会有所不同。因此，可以通过调整式（4-1）中的参数，在同样的训练集 L 上建立多个不同的特征视图，这些视图之间尽可能在一定程度上是一致的和独立的。

❑4.3 基分类器间的差异性评估

通过组合式（4-1）中不同的参数，即选择不同的特征评估函数、计算公式和保留特征数，在训练集上建立多个不同的特征视图，每个特征视图包含足够多的信息训练生成一个基分类器，这样就可以产生多个基分类器。

为了改进 Co-training 算法，目标是寻找两个满足较高一致性和独立性特征视图，也就是寻找两个满足较高一致性和独立性的基分类器来进行协同训练。如何评估一对基分类器之间的差异性呢？文献[81-83]总结了一些差异评估方法用于集成分类器的研究，这里用来间接地评估两个基分类器之间的独立性。

用 $H=\{h_1, \cdots, h_M\}$ 表示一组基分类器，h_t 和 h_s 表示一对基分类器，$C=\{c_1, \cdots, c_L\}$ 表示一组类标记，$D=\{(x_1, y_1), \cdots, (x_N, y_N)\}$ 表示一组带类标签的样本。M、N、L 分别表示基分类器个数一样本数和类别数。基分类器 h_t 分类输出对应一个 N 维向量 $\boldsymbol{\mu}_t = [\mu_{t1}, \cdots, \mu_{ti}, \cdots, \mu_{tN}]^{\mathrm{T}}$，$t=1, \cdots, M$。

$$\mu_{ti} = \begin{cases} 1 & , h_t(x_i) = y_i \\ 0 & , \text{其他} \end{cases}$$

u_{ti} 表示基分类器对样本 $x_i \in D$ 的分类结果，为了评估一对基分类器 h_t 和 h_s 之间的差异，用 N^{11} 表示 h_t 和 h_s 都分类正确的样本数，N^{01} 表示 h_t 分类错误而 h_s 分类正确的样本数，N^{10} 表示 h_t 分类正确而 h_s 分类错误的样本数，N^{00} 表示 h_t 和 h_s 都分类错误的样本数，定义分别如下，且 $N = N^{11} + N^{00} + N^{01} + N^{10}$。

$$N^{11} = \sum_{i=1}^{N} (\mu_{ti} \wedge \mu_{si})$$

$$N^{01} = \sum_{i=1}^{N} (\overline{\mu}_{ti} \wedge \mu_{si})$$

$$N^{10} = \sum_{i=1}^{N} (\mu_{ti} \wedge \overline{\mu}_{si})$$

$$N^{00} = \sum_{i=1}^{N} (\overline{\mu}_{ti} \wedge \overline{\mu}_{si})$$

1．Q 统计

对两个基分类器 h_t 和 h_s，Q 统计（Q statistic）定义如下：

$$Q_{ts} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{4-2}$$

如果基分类器之间统计独立，那么 $Q_{t,s} = 0$，$Q_{t,s} \in [-1,1]$。两个基分类器的分类趋向一致则 $Q>0$，否则 $Q<0$。

2．相关系数

两个分类器之间的相关性用相关系统（Correlation Coefficient ρ）度量：

$$\rho_{t,s} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}} \tag{4-3}$$

如果 $\rho_{t,s} = 0$，那么两个基分类器 h_t 和 h_s 完全不相关，$\rho_{t,s}$ 越大二者的相关度越大，$\rho_{t,s} \in [0,1]$。

3．不一致评估法（Disagreement Measure）

不一致评估法评估的是两个基分类器 h_t 和 h_s 的不一致性，计算公式为

$$\mathrm{Dis}_{t,s} = \frac{N^{01} + N^{10}}{N^{11} + N^{01} + N^{10} + N^{00}} \tag{4-4}$$

4．双误评估法（Double Fault Measure）

$F_{t,\ s}$表示两个基分类器 h_t 和 h_s 一起出错的概率，计算公式为

$$F_{t,s}=\frac{N^{00}}{N^{11}+N^{01}+N^{10}+N^{00}} \tag{4-5}$$

5．综合评估法（integrate Diversity Measure，DM）

第 4.5 节的实验数据证明，这四种基分类器间的差异评估方法 $Q_{t,s}$、相关系数 $\rho_{t,s}$ 、$\mathrm{Dis}_{t,s}$ 和 $F_{t,s}$ 能够有效评估两个基分类器间的独立性。但是，从实验中也发现，单凭某一种差异评估法来给出判断，不是十分准确。因此，想到综合四种方法的评估结果，给出最后的评判。这里用 $\mathrm{DM}_{t,s}$ 表示综合四种评估法：

$$\mathrm{DM}_{t,s}=\alpha Q_{t,s}+\beta\rho_{t,s}+\gamma(1-\mathrm{Dis}_{t,s})+\delta F_{t,s} \tag{4-6}$$

其中，α、β、γ、δ 是权重调整因子，分别调整 $Q_{t,s}$、相关系数 $\rho_{t,s}$ 、$\mathrm{Dis}_{t,s}$ 和 $F_{t,s}$ 在综合评估指标 $\mathrm{DM}_{t,s}$ 中的作用，且 $\alpha+\beta+\gamma+\delta=1$，$\alpha$，$\beta$，$\gamma$，$\delta\in[0,1]$。

❑4.4　TV-SC 算法与 TV-DC 算法

分析 Co-training 算法，如果两个特征视图是独立的，那么在这两个视图上训练生成的两个基分类器在协同训练时，能够减少给同一个未标注文本都分类错误的可能性。也就是说，除了要考虑基分类器的分类正确性（accuracy），还要考虑基分类器之间的差异（diversity）。只有存在合理的差异，特别是二者是不相关的，或者说是独立的，才能较大程序地提高 Co-training 的分类精度。目标是寻找两个满足较高一致性和独立性的基分类器来进行协同训练。基于这种观点，采取以下改进措施。

① 通过组合式（4-1）中不同的参数，即选择不同的特征评估函数、计算公式和保留特征数，在训练集上建立多个不同的特征视图，每个特征视图包含足够的信息训练生成一个基分类器。不同的特征视图之间存在差异，即使选择同样的分类算法，训练产生的基分类器也会有所不同。

② 当每个视图上采用相同的分类算法时，称其为 TV-SC（Two Views-Same Classifiers algorithm）算法；每个视图上采用不同的分类算法时，

称其为 TV-DC（Two Views-Different Classifiers algorithm）算法，描述分别如表 4.2 和表 4.3 所示。

③ 按照 Q 统计、相关系数ρ、不一致性 Dis、双误法 DF 及综合评估法 DM 评估基分类器之间差异，选择一对独立性较强的基分类器相互协同训练学习。

令 D^l 表示标注样本集，D^u 表示未标注样本集，$C=\{c_1,\cdots,c_L\}$表示一组类标记，$V=\{V_1,\cdots,V_M\}$表示建立的多个特征视图，$H=\{h_1,\cdots,h_M\}$表示在 V 上建立的多个基分类器。f_1 和 f_2 表示两种分类算法，可以是 Naïve Bayesian（NB）、K 近邻（KNN）、质心向量法（CC）等。$DM(h_t, h_s)$表示两个基分类器 h_t 和 h_s 之间的综合差异性。r 表示迭代次数。

表 4.2　TV-SC 算法

1．Create M feature views $V_1,\cdots,V_M$ based TEF-WA(see Eq.3-3～3-12).
2．Use f and $V_t(D^l)$ to create classifiers h_t, $t=1,\cdots,M$.
3．Compute DM(h_t, h_s), $t,s=1,\cdots,M$. (See Eq. 4-2～4-6) .
4．Select two classifiers according to {DM(h_t,h_s)},associates with two views V_i and V_j.
5．Loop for r iterations
　(5.1) Create classifiers $h_1$and h_2 using f and $V_i(D^l)$, $V_j(D^l)$ respectively.
　(5.2) For each class c_j Do
　(5.2.1) Let b1 and b2 be unlabeled documents on which $h_1$and h_2 make most confident prediction for c_j.
　(5.2.2) Remove b1 and b2 from D^u, label them according to $h_1$and h_2, and add them to D^l respectively.
6．Combine the prediction of $h_1$and h_2

表 4.3　TV-DC 算法

1．Create M feature views $V_1,\cdots,V_M$ based TEF-WA(see Eq. 3-3～3-12).
2．Use f_1 and $V_t(D^l)$ to create classifiers h_{t1}, Use f_2 and $V_t(L)$ to create classifiers h_{t2}, $t=1,\cdots,M$.
3．Compute $DM(h_{t1},h_{s2})$, t, $s=1,\cdots,M$; (See Eq. 4-2～4-6).
4．Select two classifiers according to {DM(h_{t1},h_{s2})}, associates with two views V_i and V_j.
5．Loop for r iterations
　(5.1) Create classifiers $h_1$and h_2 using f_1 and $V_i(D^l)$, f_2 and $V_j(D^l)$ respectively
　(5.2) For each class c Do
　(5.2.1) Let b1 and b2 be unlabeled documents on which $h_1$and h_2 make most confident prediction for c_j.
　(5.2.2) Remove b1 and b2 from D^u, label them according to $h_1$and h_2, and add them to D^l respectively.
6．Combine the prediction of $h_1$and h_2

❑4.5 实验结果及其分析

实验数据采用从易宝中文下载的新闻文本作为数据集，包含了 20 341 篇分属于经济、政治、国际、文教和体育五大类别的文本。把整个数据集划分成多个不同的子集。可以看出，经济、政治、国际是比较难以区分的类别，文教和体育两个类别比较相近。实验结果评估采用宏平均精度 Macro-Precision、宏平均召回率 Macro-Recall、宏平均 F1 值 Macro-F1、微平均 F1 值 Micro-F1。

为了验证 TV-SC 和 TV-DC 的分类效果，与基于随机分割特征视图的 Co-training 算法（记为 Co-Rnd）进行了实验比较。表 4.4 和表 4.5 所示是 200 篇标注样本和 500 篇未标注样本上 TV-DC、TV-SC 及 Co-Rnd 算法的分类结果，计算 DM 时，$\alpha=\beta=\delta=0.333$，$\gamma=0$。

表 4.4 描述的是 TV-DC 算法的分类性能。TV-DC 算法依据式（4-1）选择不同的参数构建两个独立性较强的特征视图，每个视图上的基分类器分别选择 Naïve Bayesian（NB）和质心分类算法（CC）。表中第 2 列描述的是构建特征视图的参数和该视图上选择的分类算法；第 3 列描述的是基分类器的分类精度，反映的是基分类器的正确性；第 4～8 列描述的是使用不同的差异评估法 Q 统计、相关系数ρ、不一致性 Dis、双误法、综合法 DM 评价两个基分类器之间的差异性；第 9 列表示是否使用未标注文本；第 10～13 列是分别用 Macro-Precision、Macro-Recall、Macro-F1 和 Micro-F1 评估的 TV-DC 算法的分类结果。

表 4.4 TV-DC 算法分类结果及其两个基分类器的正确性和差异性比较

No.	Sub-View(Classifier)	Macro-Precision (%)	Diversity Measures					L+U	TV-DC Classifier(%)			
			Q	ρ	Dis	DF	**DM**		Macro-Precision	Macro-Recall	Macro-F1	Micro-F1
1	IG/TF/1200(NB)	77.77	0.4773	0.1700	0.5478	0.2000	0.28	yes	81.36	80.15	80.26	80.87
	Odds/TF/1000(CC)	73.52						No	75.95	76.06	75.74	76.52
2	MI/TF/1000(NB)	**79.77**	**0.3399**	**0.1305**	0.5391	**0.1913**	**0.22**	yes	**85.25**	**83.44**	**83.50**	**84.35**
	Odds/TF/1200(CC)	**78.19**						No	80.79	80.34	80.02	80.87

续表

No.	Sub-View(Classifier)	Macro-Precision (%)	Diversity Measures					L+U	TV-DC Classifier(%)			
			Q	ρ	Dis	DF	**DM**		Macro-Precision	Macro-Recall	Macro-F1	Micro-F1
3	ECE/DF/1300(NB)	**81.09**	**0.2874**	**0.1013**	0.5826	**0.1739**	**0.19**	yes	**83.86**	**82.89**	**82.89**	**83.48**
	Odds/TF/900(CC)	73.52						No	77.96	77.80	77.30	78.26
4	IDF/DF/900(NB)	78.92	0.3450	0.1246	0.5652	0.1913	0.22	yes	82.96	81.89	81.80	82.61
	Odds/TF/900(CC)	73.52						No	76.98	76.96	76.50	77.39

分析表 4.4 中的数据可以看出以下几点。

① TV-DC 的分类效果不仅与每个基分类器的精度有关，还与基分类器间的差异性有关。

a. 第 3 组数据表明由参数 ECE/DF/1300（NB）和 Odds/TF/900（CC）构建的一对基分类器间的差异性最大，Q 统计、相关系数ρ、双误法的值都最小，且 DM 值最小。

b. 第 1 组数据表明由参数 IG/TF/1200（NB）和 Odds/TF/1000（CC）生成的一对基分类器间的差异性最小。

c. 第 2 组的一对基分类器的正确性相对其他 3 组最好，差异性的评估、综合指标 DM 次于第 2 组。综合比较基分类器的正确性和差异性，第 2 组的一对基分类器最好，二者协同分类构成的 TV-DC 分类器的分类效果是最好的，Macro-Precision 达到了 85.25%。

d. 第 3 组数据中虽然 Odds/TF/900（CC）生成的基分类器的精度仅 73.52%，比较低，但是由于两个基分类器的差异性最大，使得它的 TV-DC 的分类效果仅次于第 2 组，Macro-Precision 达到 83.86%。

e. 比较第 1 组和第 4 组的数据，能够看出，两个基分类器的分类正确性差不多时，二者的差异性越大，构成的 TV-DC 分类器的分类结果越好。

② 观察表 4.4 中的数据，Q 统计、相关系数ρ、不一致性 Dis、双误法 DF 这几种差异评估方法都能比较好地反映基分类器间的差异性，但是不能仅根据一种差异评估法的值来判定基分类器间的差异性，需要综合考虑，因此 DM 评估指标最好。

③ 比较表 4.4 中每一组数据的上下两行，即 TV-DC 分类器是否使用未标注文本，可以看出，使用未标注文本能够明显提高分类效果，Macro-Precision 提高了 4.46%～5.97%，Micro-F1 提高了 3.48%～5.22%。这说明使用 TV-DC 算法结合标注文本能够明显地提高分类效果。

表 4.5 描述的是 TV-DC、TV-SC 和 Co-Rnd 算法的分类性能比较。TV-DC 算法与 TV-SC 算法的不同主要在于：在依据式（4-1）选择不同的参数构建两个独立性较强的特征视图之后，TV-DC 算法在每个视图上采用不同的分类算法建立基分类器，本次实验选择的分别是 Naïve Bayesian（NB）和质心分类算法（CC）；而 TV-SC 算法在两个视图上选择的分类算法是相同的，本次实验选择的是 Naïve Bayesian（NB）。Co-Rnd 算法基于随机分割法生成两个特征视图，且两个视图上分别采用 NB 和 CC 分类算法。

表 4.5 中第 3 列描述的是构建特征视图的参数和该视图上选择的分类算法；第 4 列描述的是基分类器的分类精度，反映的是基分类器的正确性；第 5～9 列描述的是使用不同的差异评估法 Q 统计、相关系数ρ、不一致性 Dis、双误法 DF、综合法 DM 评价两个基分类器之间的差异性；第 10～13 列描述的是 TV-DC 或 TV-SC 的分类结果，用 Macro-Precision、Macro-Recall、Macro-F1 和 Micro-F1 评估。

表 4.5　TV-DC、TV-SC 与 Co-Rnd 算法的分类结果比较及基分类器之间的正确性和差异性比较

No.	Classifier	Sub-View (Classifier)	Macro-Precision (%)	Diversity Measures					Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)	Micro-F1 (%)
				Q	ρ	Dis	DF	DM				
1	TV-DC	MI/TF/1000(NB)	79.77	**0.3399**	**0.1305**	0.5391	0.1913	**0.22**	**85.25**	**83.44**	**83.50**	**84.35**
		Odds/TF/1200(CC)	78.19									
	TV-SC	MI/TF/1000(NB)	79.77	0.4531	0.1582	0.5478	0.1652	0.26	78.71	77.06	77.21	78.26
		Odds/TF/1200(NB)	80.56									
	Co-Rnd	RNDV1(NB)	75.47	0.5464	0.2788	0.3304	0.1826	0.34	74.07	72.38	70.87	72.38
		RNDV2(CC)	73.75									
2	TV-DC	ECE/DF/1300(NB)	81.09	**0.2874**	**0.1013**	0.5826	0.1739	**0.19**	**83.86**	**82.89**	**82.89**	**83.48**
		Odds/TF/900(CC)	**73.52**									
	TV-SC	ECE/DF/1300(NB)	**81.09**	0.3419	0.1130	0.5913	0.1565	**0.20**	**82.23**	80.93	81.09	81.74
		Odds/TF/900(NB)	**80.25**									
	Co-Rnd	RNDV1(NB)	78.26	0.5392	0.2757	0.3304	0.1826	0.33	77.78	75.00	76.36	76.51
		RNDV2(CC)	75.74									
3	TV-DC	IDF/DF/900(NB)	78.92	0.3450	0.1246	0.5652	0.1913	**0.22**	**82.96**	81.89	81.80	**82.61**
		Odds/TF/900(CC)	73.52									
	TV-SC	IDF/DF/900(NB)	78.92	0.4010	0.1366	0.5739	0.1739	0.24	80.33	79.09	79.16	80.00
		Odds/TF/900(NB)	80.25									
	Co-Rnd	RNDV1(NB)	72.70	0.5172	0.2658	0.3391	0.1913	0.32	73.52	73.07	72.67	73.91
		RNDV2(CC)	71.43									

表 4.5 中的实验数据分析如下。

① 比较每一组数据，使用式（4-1）中相同的一对参数（评估函数、TForDF、保留特征数）构造的两个特征视图，选择不同的分类算法建立的两个基分类器间的差异性比选择相同的分类算法建立的基分类器的差异性要大，通过比较每组的 Q 统计、相关系数ρ、不一致性 Dis、双误法 DF 和综合法 DM 可以明显看到。因此，TV-DC 算法的分类效果要优于 TV-SC 算法，如 Macro-Precision 高出 1.63%～6.54%，Macro-Recall 高出 1.96%～6.38%。

② 第 2 组实验数据的 TV-SC 算法下的两个基分类器的 Q 统计、相关系数ρ、双误法 DF、综合法 DM 的值都比较小，所以二者的差异性比较大。这说明，使用 TEF-WA 权值调整技术，通过调整式（4-1）中的参数，可以找到一对独立性较强的特征视图 ECE/DF/1300 和 Odds/TF/900，即使在这对视图上使用相同的分类算法。另外二者的分类精度也都比较高，分别为 81.09% 和 80.25%。因此，TV-SC 算法也取得了比较好的分类效果，Macro-Precision 达到了 82.23%。

③ Co-Rnd 采用随机分割视图法建立的两个基分类器间的差异性较小，分类结果明显比 TV-SC 和 TV-DC 差。对比 Co-Rnd，TV-DC 的 Macro-Precision 提高了 6.08%～10.18%，TV-SC 的 Macro-Precision 提高了 4.45%～6.81%，Macro-Recall、Macro-F1 和 Micro-F1 的值也表明 TV-DC 和 TV-SC 的分类结果明显优于 Co-Rnd。

总的来说，通过实验和数据分析，可以得出以下结论。

① 通过 TEF-WA 权值调整技术，结合不同的评估函数、TF 或 DF 计算公式、保留特征数，可以构造多种不同的特征视图，可以从中得到一对独立性较强（差异性较大）的特征视图。

② Q 统计、相关系数ρ、不一致性 Dis、双误法 DF 四种差异评估方法能比较有效地评价一对基分类器间的差异性，综合法 DM 评估得更准确。

③ 结合未标注文本和标注文本的 Co-training 协同分类算法，在选择基分类器时，不仅要考虑基分类器的正确性，还要考虑二者的差异性，在满足一定正确性的前提下，差异性越大，最终的分类效果越好。TV-SC 和 TV-DC 正是从这一点出发对 Co-training 进行的改进，二者都优于 Co-Rnd 算法，而且 TV-DC 算法的效果要优于 TV-SC 算法。

❑4.6 本章小结

在 Co-training 算法中，通常假设两个特征视图具备一致性和独立性的要求，然而实际应用中同时满足上述条件且自然划分的视图往往不存在，且二者的独立性很难直接评判。首先基于 TEF-WA 技术利用特征评估函数及其他参数建立多个特征视图，每个特征视图包含足够多的信息训练生成一个基分类器，这样就可以产生多个基分类器。然后利用多种差异评估方法评价基分类器之间差异，选择一对独立性较强的基分类器相互协同训练学习。实验表明两种改进的 TV-SC 算法和 TV-DC 算法在半监督分类中都比较有效，优于基于随机分割的 Co-Rnd 算法，而且 TV-DC 算法的分类效果要优于 TV-SC 算法。

第5章

基于特征独立模型的 Co–training 改进算法

分析半监督学习的经典方法 Co-training 算法，它要求两个特征视图满足一致性和独立性的理论假设，但是许多应用中不存在自然划分且满足这种假设的两个视图，这就限制了 Co-training 算法的应用。

本章提出了通过条件互信息（Mutual Information，MI）[3][28][103]或条件 χ^2 统计量（CHI）[3][28][103]两种方法评估两个特征之间的相互独立性（Mutual Independence，MID）。在特征视图上建立一种相互独立性模型（Mutual Independence Model，MID-Model）。基于相互独立性模型，提出一种特征子集划分方法 PMID 算法，根据评估每对特征相互独立性的方法不同，分别称为 PMID-MI 算法和 PMID-CHI 算法，前者利用条件互信息计算特征的相互独立性，后者利用条件 χ^2 统计量（CHI）。PMID-MI 算法或 PMID-CHI 算法可以有效地将一个特征视图划分成两个条件独立性较强的特征子图。由于直接评估两个特征子图之间的独立性比较困难，然而评估由两个视图训练生成的分类器之间的差异性相对来说比较容易，从而实现间接地评估两个视图之间的独立性。

基于相互独立性模型提出了对 Co-training 的改进算法——SC-PMID 算法。实验表明，通过 PMID-MI 或 PMID-CHI 算法，可以有效地将一个特征集合划分成两个条件独立性较强的特征子集，明显优于随机划分特征子集的

方法。也正是因为特征子集之间存在的较强独立性，SC-PMID 算法结合未标注文本进行半监督分类是有效的。

❑5.1 特征独立模型

互信息（Mutual Information，MI）和χ^2统计量（CHI）在信息论中都可以用来衡量两个随机变量之间的统计独立关系[3][28][103]，在特征选择和权重调整时，利用互信息和χ^2统计量分别衡量特征词与某个类之间的独立关系[103]。受此启发，可以利用互信息、χ^2统计量评估两个特征词之间的独立关系，提出了两种估算特征之间的相互独立性的方法，从而建立特征独立模型。

❑5.1.1 基于条件互信息的相互独立性

两个特征之间的条件互信息定义如下（5.1）：

$$I(w_i, w_j \mid c_k) = \log \frac{P(w_i, w_j \mid c_k)}{P(w_i \mid c_k) P(w_j \mid c_k)} \tag{5-1}$$

其中，$P(w_i, w_j \mid c_k)$指特征词w_i与w_j在c_k类文本中共同出现的条件概率，$P(w_i \mid c_k)$指特征词 w_i在 c_k类文本中出现的条件概率。在实际计算中，这些概率可以用特征词在训练集中出现的相应频率予以近似。定义 w_i和 w_j在训练集中基于类 c_k同时出现的频数为 $F(w_i, w_j|c_k)$，N_{ck}为训练集中 c_k类文本的数目，$F(w_i|c_k)$为 w_i在训练集中出现的 c_k类文本频数，$F(w_j|c_k)$为 w_j在训练集中出现的 c_k类文本频数，那么$I(w_i, w_j \mid c_k)$可以近似为：

$$I(w_i, w_j \mid c_k) = \log \frac{F(w_i, w_j \mid c_k) \times N_{c_k}}{F(w_i \mid c_k) \times F(w_j \mid c_k)} \tag{5-2}$$

从概率上说，如果某两个词 w_i和 w_j在分布上统计独立，也就是说特征词 w_i的出现与 w_j的出现是完全独立的，那么，$P(w_i, w_j \mid c_k) = P(w_i \mid c_k) \times P(w_j \mid c_k)$，所以$I_{\min}(w_i, w_j \mid c_k) = 0$。当特征词 w_i与 w_j的出现完全不独立时，即要么同时出现要么都不出现，这时：

$$I_{\max}(w_i, w_j \mid c_k) = \log \frac{F(w_i, w_j \mid c_k) \times N_{c_k}}{F(w_i \mid c_k) \times F(w_j \mid c_k)}$$

$$= \log \frac{N_{c_k}}{F(w_i \mid c_k)} < \log N_{c_k}$$

定义 5.1　基于条件互信息的相互独立性（Mutual Independence based on MI，MID_MI），特征词 w_i 与 w_j 的相互独立性 $\mathrm{MID\_MI}(w_i, w_j)$ 用 w_i 与 w_j 的条件互信息 $I(w_i, w_j \mid c_k)$ 衡量，计算如式（5-3）或式（5-4）：

$$\mathrm{MID\_MI}(w_i, w_j) = \sum\nolimits_{c_k} I(w_i, w_j \mid c_k) = \sum\nolimits_{c_k} \log \frac{P(w_i, w_j \mid c_k)}{P(w_i \mid c_k) P(w_j \mid c_k)} \quad (5\text{-}3)$$

$$\mathrm{MID\_MI}(w_i, w_j) = \max_{k=1}^{m} I(w_i, w_j \mid c_k) = \max_{k=1}^{m} \log \frac{P(w_i, w_j \mid c_k)}{P(w_i \mid c_k) P(w_j \mid c_k)} \quad (5\text{-}4)$$

因此，按式（5-3）计算 MID_MI (w_i,w_j)的取值范围为[0,log(N/m))，按式（5-4）计算 MID_MI (w_i, w_j)的取值范围为 $[0, (\log(N/m))/m)$。MID_MI (w_i, w_j)的值越小，w_i 与 w_j 的相互独立性就越大，相互依赖性越小。

5.1.2　基于条件χ^2统计量的相互独立性

条件 χ^2 统计量的定义如式（5-5）：

$$\chi^2(w_i, w_j \mid c_k) = \frac{N_{ck}(ae - bd)^2}{(a+d)(b+e)(a+b)(d+e)} \quad (5\text{-}5)$$

这里，a 为训练集中包含 w_i 与 w_j 的 c_k 类文本数，b 为训练集中包含 w_i 但不包含 w_j 的 c_k 类文本数，d 为训练集中不包含 w_i 但包含 w_j 的 c_k 类文本数，e 为训练集中不包含 w_i 和 w_j 的 c_k 类文本数，N_{ck} 为训练集中的 c_k 类文本数。

定义 5.2　基于条件 χ^2 统计量的相互独立性（Mutual Independence based on CHI，MID_CHI），特征词 w_i 与 w_j 的相互独立性 MID_CHI(w_i, w_j）用 w_i 与 w_j 的条件 χ^2 统计量 $\chi^2(w_i, w_j \mid c_k)$ 衡量，如式（5-6）或式（5-7）：

$$\mathrm{MID\_CHI}(w_i, w_j) = \sum\nolimits_{c_k} P(c_k) \chi^2(w_i, w_j \mid c_k) \quad (5\text{-}6)$$

$$\mathrm{MID\_CHI}(w_i, w_j) = \max_{k=1}^{m} P(c_k) \chi^2(w_i, w_j \mid c_k) \quad (5\text{-}7)$$

当特征词 w_i 与 w_j 之间统计完全独立时，理想状态下 χ^2 统计量应该为 0，这种情况下，N_{ck} 个训练文本的数目应该在这四种文本中均匀分布，即 $a=b=e=d$，或者 $ad=be$，$\chi^2(w_i,w_j\,|\,c_k)=0$，$w_i$ 与 w_j 的相互独立性最大，相互依赖性最小。另一个极端是，当特征词 w_i 和 w_j 完全依赖时，也就是 w_i 和 w_j 要么同时出现要么都不出现。体现在 a、b、e、d 这四个数量上，$a+d=N_{ck}$，而 b=c=0，这时 $\chi^2(w_i,w_j\,|\,c_k)=1$，$w_i$ 与 w_j 的相互独立性最小，相互依赖性最大。

MID_CHI(w_i, w_j)与 MID_MI(w_i, w_j)的不同之处还有一点，MID_MI(w_i, w_j)是一个非规格化的值，其取值范围很大，特别是对于那些边缘概率分布很小的情况。而 MID_CHI(w_i, w_j)则是一个规格化的量，取值范围为[0,1]。$\text{MID\_CHI}(w_i,w_j)\in[0,1]$，MID_CHI($w_i$, w_j)的值越小，w_i 与 w_j 的相互独立性越大，相互依赖性就越小。

5.1.3 特征独立模型

为方便叙述，下面用 MID(w_i, w_j)代替 MID_MI(w_i, w_j)或 MID_CHI(w_i, w_j)，表示两个特征之间的相互独立性。

定义 5.3 相互独立性模型（Mutual Independence Model，MID-Model），给定特征集，建立加权无向图 $G=(V,E)$，$V=\{w_1, w_2,\cdots,w_m\}$表示经过特征选择保留的特征集，$E=\{(e_k, a_{ij})|a_{ij}=\text{MID}(w_i, w_j)$，$\forall w_i,w_j\in V$，$k=0,\cdots,m*(m-1)/2\}$，令集合 $S=\{\text{MID}(w_i, w_j)|\,\forall w_i,w_j\in V\}$，定义“≤”是 S 上的关系 R，那么满足：

① $\forall x\in S\Rightarrow (x,x)\in R$；

② $\forall x,y\in S\ \wedge(x,y)\in R\Rightarrow(y,x)\notin R$；

③ $\forall x,y,z\in S\wedge\ (x,y)\in R\wedge(\text{y,z})\in R\Rightarrow(x,\text{z})\in R$。

“≤”满足自反性、非对称性和传递性，是定义在 S 上的偏序集，所以特征独立模型具有以下性质。

性质 5.1 MID(w_i, w_j)的值越小，w_i 与 w_j 的相互独立性越大，相互依赖性就越小。当 MID (w_i, w_j)＜δ时，δ是阈值，认为 w_i 与 w_j 是相互独立的。

性质 5.2 如果 $\text{MID}(w_i,w_j)\leqslant\text{MID}(w_i,w_k)$，就说 w_i 与 w_j 的相互独立程度比 w_i 与 w_k 强，其依赖程度比 w_i 与 w_k 弱。

性质 5.3　如果 $\mathrm{MID}(w_{i1},w_{i2}) \leqslant \mathrm{MID}(w_{j1},w_{j2}) \leqslant \mathrm{MID}(w_{k1},w_{k2})$，则有 $\mathrm{MID}(w_{i1}, w_{i2}) \leqslant \mathrm{MID}(w_{k1}, w_{k2})$。

为减少计算复杂度，一方面，根据特征评估函数，如互信息、信息增益、期望交叉熵、文本证据权、几率比、χ^2 统计量等进行特征选择，保留少量的对分类贡献比较大的特征，组成初始的特征集合 V；另一方面，用下三角矩阵 $\boldsymbol{A}$ 存储每对特征的相互独立性，既降低空间复杂度又减少计算量。

定义 5.4　特征与特征子集的相互独立性（Mutual Independence between Feature and SubSet，MID_FS），特征词 w_i 与特征子集 V_p 的独立性由 w_i 与特征子集 V_p 中所有特征的相互独立性的平均值描述，如式（5-8）：

$$\mathrm{MID\_FS}(w_i,V_p)=\frac{1}{|V_p|}\sum_{w_s\in V_p}\mathrm{MID}(w_i,w_s) \tag{5-8}$$

类似地，如果 $\mathrm{MID\_FS}(w_i, V_p) < \mathrm{MID\_FS}(w_i, V_q)$，表示 w_i 与 V_p 的相互独立程度强于 w_i 与 V_q，即相关度弱于 w_i 与 V_q。

定义 5.5　特征子集之间的相互独立性（Mutual Independence between two subsets，MID_SS），两个特征子集 V_p 与 V_q 的相互独立性计算如式（5-9）：

$$\mathrm{MID\_SS}(V_p,V_q)=\frac{1}{|V_p|+|V_q|}\left(\sum_{w_i\in V_p}\mathrm{MID\_FS}(w_i,V_q)+\sum_{w_j\in V_q}\mathrm{MID\_FS}(w_j,V_p)\right) \tag{5-9}$$

同样地，$\mathrm{MID\_SS}(V_p, V_q) < \mathrm{MID\_SS}(V_r, V_s)$ 表示 V_p 与 V_q 的相互独立性强于 V_r 与 V_s。

定义 5.5 给出的是基于特征独立模型（MID-Model）的两个特征子集之间的相互独立性的量化方法。5.2 节将给出理论证明，在第 5.3 节，作者利用基分类器之间的差异性评估，从另一个角度间接地评估两个特征子集的独立性。

❑5.2　特征子集划分算法 PMID

特征子集划分的目标是要把特征集合 V 分割成两个满足一致性和独立性比较强的特征子集 V_1 与 V_2。一致性通常容易满足，因此这里主要考虑独立性。

PMID 算法如表 5.1 所示。

表 5.1　PMID 算法

Input：V={ w_1,w_2 ,…,w_m } //特征选择后的保留特征集合

1. 计算每对特征词之间的相互独立性 MID(w_i,w_j)，计算如式（5-3）和式（5-4）或式（5-6）和式（5-7）所示。

2. 建立特征独立模型（MID-Model），建立加权无向图 G=(V,E),V={w_1,w_2,…,w_m}, E={(e_k,a_{ij})|a_{ij}=MID(w_i,w_j), $\forall w_i, w_j \in V$, k=0,…,m*(m−1)/2}，E 中的边按权重 MID 升序排列。

3. 令 V_1=φ，V_2=φ。

4. 选取 E 中当前最短边 (e_k,a_{ij})。

5. **while** (V≠φ and a_{ij}<δ)　**do**　　//按三种情况分别处理：

{ If　(w_i 与 w_j 同属于 V)

{ 计算 w_i,w_j 分别与 V_1 和 V_2 的独立程度；

$\arg\max_{k,p}\{\text{MID\_FS}(w_k, V_p),\ k=i,j;\ p=1,2\}$;

V_p= V_p ∪ {w_k};另一个点加入 V_q;

从集合 V 中删除 w_i 与 w_j; }

If (w_i 和 w_j 有且仅有一个属于 V)

{ 将属于 V 的点加入另一个点所不在的集合;

从 V 中删除该点;}

If　(w_i, w_j 同属于 V_1,V_2，或者分别属于 V_1 和 V_2)　舍弃该边;

选取下一条最短边;}

Output: V_1 和 V_2。

其中，步骤 5 按照三种情况分别处理。

（1）w_i 与 w_j 同属于 V：需要计算 w_i,w_j 分别与 V_1 和 V_2 的独立程度，求出与其中的一个子集独立性最小的特征，其加入该子集，另一个特征则加入另一个子集，并从 V 中删除 w_i, w_j。

（2）w_i 和 w_j 有且仅有一个属于 V：处理比较简单，只需将属于 V 的点加入另一个点所不在的集合，并从 V 中删除该点。

（3）w_i, w_j 同属于 V_1,V_2，或者分别属于 V_1 和 V_2：舍弃该边。

重复执行，直到 V=φ，这时，得到的 V_1 和 V_2 就是独立程度相对较高的两个特征子集。按照 PMID 算法，整个特征集合 V 可以分割成两个相对独立性较强的特征子集 V_1 和 V_2，可以推出满足下列两式：

$$V_1 \bigcap V_2 = \Phi \tag{5-10}$$

$$\forall w_i \in V_p \Rightarrow \text{MID\_FS}(w_i, V_q) \leqslant \text{MID\_FS}(w_i, V_p)\ \ p,q \in \{1,2\}, p \neq q \tag{5-11}$$

式（5-11）表示，对于 $\forall w_i \in \mathrm{V}_p$ ，w_i 与集合 V_q 的独立性要强于 w_i 与集

合 V_p 的独立程度。也就是说，同一个集合内的特征词相互依赖性较大，相互独立性较弱；而分别属于不同子集的特征词相互依赖性较小，相互独立性较强，从而使特征子集 V_1 与 V_2 之间有较强的相互独立性。

式（5-10）明显成立，为了证明式（5-11）成立，先证明式（5-12）成立。

$$\forall w_i, w_k \in V_p \Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_i, w_j) \leqslant \mathrm{MID}(w_i, w_k) \qquad (5\text{-}12)$$

证明：对任意 $w_i, w_k \in V_p$，存在两种情况：

① w_i 在 w_k 之前加入 V_p；

② w_i 在 w_k 之后加入 V_p。

下面对这两种情况分别证明。

① 如果 w_i 在 w_k 之前加入 V_p。

反证法：假设：$\forall w_i, w_k \in V_p \Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_i, w_k) < \mathrm{MID}(w_i, w_j)$

$\Rightarrow$ 边 (w_i, w_k) 必然排在边 (w_i, w_j) 之前

又 $\because w_i$ 在 w_k 之前加入 V_p

由 PMID 算法 $\Rightarrow w_i, w_k$ 被分别加入 V_p 和 V_q

$\Rightarrow w_k \in V_q \Rightarrow$ 矛盾

$\therefore \forall w_i, w_k \in V_p \Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_i, w_j) \leqslant \mathrm{MID}(w_i, w_k)$

② 如果 w_i 在 w_k 之后加入 V_p。

反证法：假设：$\forall w_i, w_k \in V_p \Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_i, w_k) < \mathrm{MID}(w_i, w_j)$

$\Rightarrow$ 边 (w_i, w_k) 排在边 (w_i, w_j) 之前

由 PMID 算法 $\Rightarrow$ 边 (w_i, w_k) 将被放弃

又 $\because w_i$ 在 w_k 之后加入 V_p，由 PMID 算法

$\Rightarrow \exists w_s \in V_q \wedge \mathrm{MID}(w_k, w_s) \leqslant \mathrm{MID}(w_i, w_k)$

$\therefore$ 当 w_i 在 w_k 之后加入 V_p 时，有

$$\forall w_i, w_k \in V_p \Rightarrow \exists w_s \in V_q \wedge \mathrm{MID}(w_k, w_s) \leqslant \mathrm{MID}(w_i, w_k)$$
$$\Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_k, w_j) \leqslant \mathrm{MID}(w_i, w_k)$$
$$\Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_k, w_j) \leqslant \mathrm{MID}(w_k, w_i)$$

由①和②，得 $\forall w_i, w_k \in V_p \Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_i, w_j) \leqslant \mathrm{MID}(w_i, w_k)$。

因此式（5-12）得证。

下面来证明式（5-11）成立。

证明：由式（5-12）有：

$$\forall w_i, w_k \in V_p \Rightarrow \exists w_j \in V_q \wedge \mathrm{MID}(w_i, w_j) \leqslant \mathrm{MID}(w_i, w_k)$$

$$\Rightarrow |V_p| \mathrm{MID}(w_i, w_j) \leqslant \sum_{w_k \in V_p} \mathrm{MID}(w_i, w_k)$$

$$\Rightarrow \sum_{w_j \in V_q} |V_p| \mathrm{MID}(w_i, w_j) \leqslant |V_q| \sum_{w_k \in V_p} \mathrm{MID}(w_i, w_k)$$

$$\Rightarrow \frac{1}{|V_p||V_q|} \sum_{w_j \in V_q} |V_p| \mathrm{MID}(w_i, w_j) \leqslant \frac{1}{|V_p||V_q|} |V_q| \sum_{w_k \in V_p} \mathrm{MID}(w_i, w_k)$$

$$\Rightarrow \frac{1}{|V_q|} \sum_{w_j \in V_q} \mathrm{MID}(w_i, w_j) \leqslant \frac{1}{|V_p|} \sum_{w_k \in V_p} \mathrm{MID}(w_i, w_k)$$

$$\Rightarrow \mathrm{MID\_FS}(w_i, V_q) \leqslant \mathrm{MID\_FS}(w_i, V_p)$$

因此，式（5-11）得证。

由此可证，基于特征相互独立模型 PMID 算法能够把特征集合 V 划分成两个有较强的相互独立性特征子集 V_1 与 V_2。同一个集合内的特征词相互依赖性较大，相互独立性较弱；而分属于不同子集的特征词相互依赖性较小，相互独立性较强，从而使特征子集 V_1 与 V_2 之间有较强的相互独立性。

根据评估每对特征的方法是基于条件互信息还是基于条件 χ^2 统计量，分别称其为 PMID-MI 算法和 PMID-CHI 算法。

为了进一步验证和评估由 PMID-MI 算法或 PMID-CHI 算法划分得到的两个特征子集的条件独立性，利用 $Q_{t,s}$、相关系数 $\rho_{t,s}$、$\mathrm{Dis}_{t,s}$、$F_{t,s}$ 和综合评估指标 $\mathrm{DM}_{t,s}$（定义见第 4 章 4.3 节）评估在这两个特征子集上训练生成的基分类器之间的差异性，来间接地评估两个特征子集的条件独立性。第 5.4 节的实验数据表明，这几种差异评估方法 $Q_{t,s}$、相关系数 $\rho_{t,s}$、$\mathrm{Dis}_{t,s}$、$F_{t,s}$ 和 $\mathrm{DM}_{t,s}$ 能够有效地评估两个基分类器间的独立性。

❑5.3 基于 MID-Model 的改进算法 SC-PMID

为了提高 Co-training 算法的性能，不仅要考虑两个分类器的分类正确性，还要考虑两个基分类器之间的差异性。因为，存在合理的差异，特别是二者是不相关的，或者说是独立的，能够减少两个基分类器给同一个未标注文本都标注错误的可能性。由第 5.1 节的特征独立模型，第 5.2 节的特征子集划分算法

PMID-MI 和 PMID-CHI，以及基分类器间的差异性评估验证，可以得到两个满足一致性和较高的独立性的特征子集，从而提高 Co-training 的分类性能。

改进的半监督分类算法 SC-PMID 的描述如表 5.2 所示。为了以示区别，采用 PMID-MI 特征子集划分算法的 SC-PMID 算法又称 SC-PMID-MI 算法；而使用 PMID-CHI 算法的则称为 SC-PMID-CHI 算法。

表 5.2　SC-PMID 算法

Input: labeled documents D^l, unlabeled documents D^u;
iteration k;
Classifier algorithm f_1 and f_2;
1. PMID(V,V_1,V_2); // PMID-MI or PMID-CHI
2. **For** t=1 to k **do**
(1) Use f_1 and $V_1(D^l)$ to create classifiers h_1,
Use f_2 and $V_1(D^l)$ to create classifiers h_2;
(2) Classify the unlabeled documents of D^u using h_1 and h_2 respectively.
(3) For each class c Do
(3.1) Let b1 and b2 be unlabeled documents on which $h_1$and h_2 make most confident prediction for c.
(3.2) Remove b1 and b2 from D^u, label them according to $h_1$and h_2, and add them to D^l respectively.
Output: Combine the prediction of $h_1$and h_2.

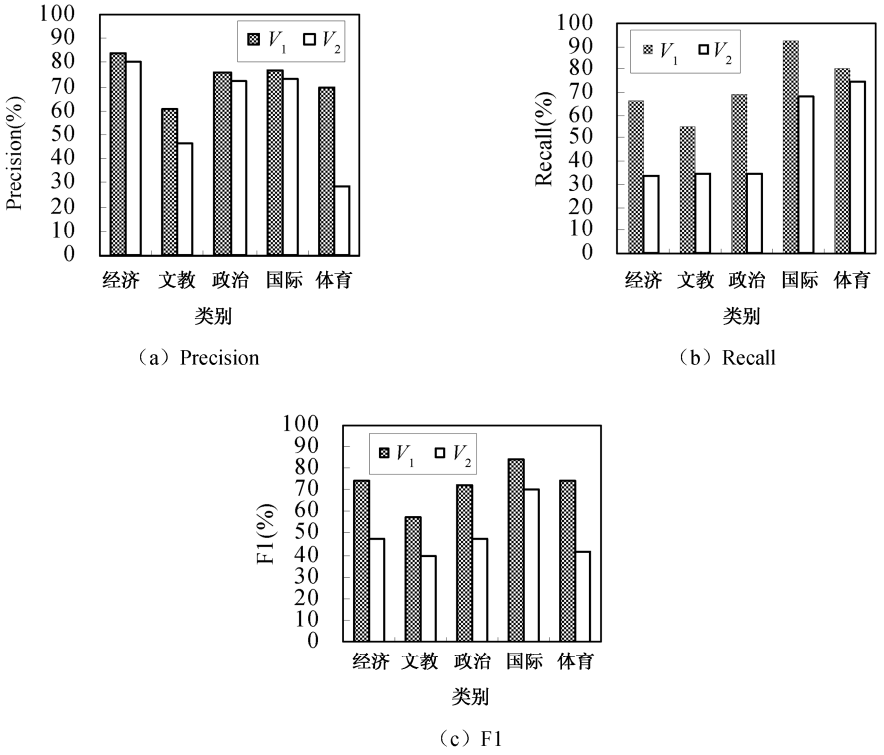
❑5.4　实验结果及其分析

实验数据采用从易宝中文下载的中文新闻文本作为数据集，包含了 20 341 篇分属于经济、政治、国际、文教和体育五大类别的新闻。随机选取包含不同数目的样本作为训练集和测试集，并且训练集与测试集之间没有交叉。选择一定数量的样本，去掉类别标签作为未标注样本集。用 *Ln* 表示包含 *n* 篇带类别标签的标注文本集，*Um* 表示包含 *m* 篇不带类别标签的未标注文本集，而 *Ts* 表示包含 *s* 篇的带类别标签的测试文本集。

分类结果评估采用宏平均精度 Macro-Precision，特征子集之间的条件独立性利用基分类器间的差异性间接评估，差异评估法选择 $Q_{t,s}$、相关系数 $\rho_{t,s}$、不一致性 $\mathrm{Dis}_{t,s}$、双误法 $F_{t,s}$ 评估法及综合法 $DM_{t,s}$。为了评估所提出的两种特征子集划分算法 PMID-MI 和 PMID-CHI 的性能，还与随机划分算法 PART-Rnd 进行了实验比较，这里仅列出部分实验结果。

5.4.1 PMID-MI 与 PART-Rnd 的实验比较

图 5.1 表示在使用望交叉熵 ECE、词频型计算公式 TF，进行权重调整和特征选择后保留了 1000 特征后，由 PMID-MI 算法划分的两个特征子集 V_1 和 V_2 上训练得到的两个 NB 分类器的分类结果比较。从图 5.1 可以看出二者的 Precision、Recall 和 F1 值是有差异的，这间接地说明 PMID-MI 划分算法能够比较有效地将一个特征集合划分成满足一定差异性的（或者说两个相对独立的）特征子集。



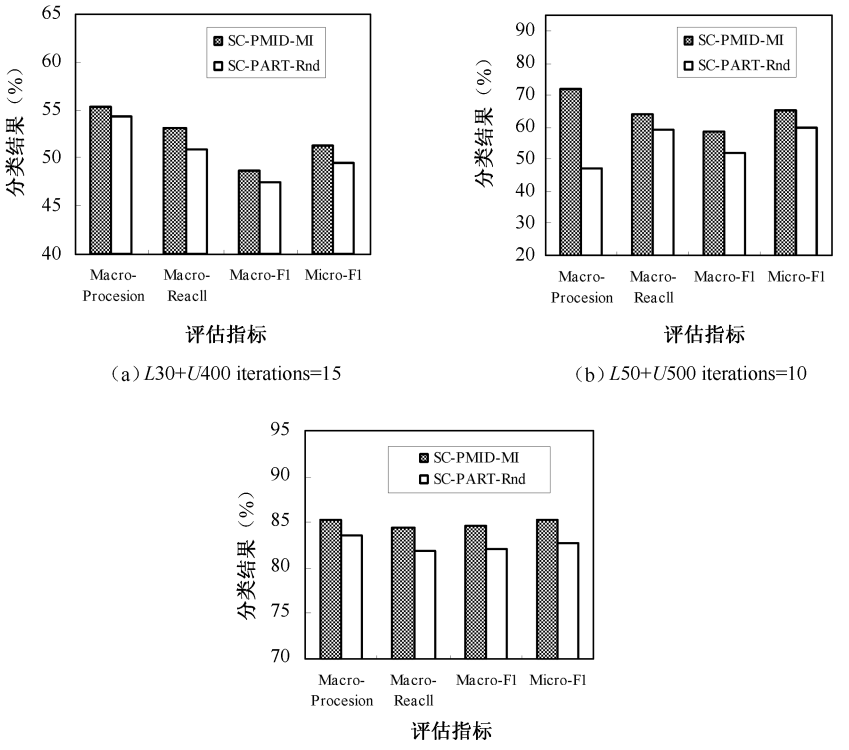
（a）Precision　　（b）Recall

（c）F1

图 5.1　由 PMID-MI 划分的特征子集 V_1 和 V_2 上的 NB 分类器的分类结果比较

图 5.2 给出的是基于 PMID-MI 和基于随机分割算法 PART-Rnd 的两种半

监督分类算法 SC-PMID-MI 和 SC-PART-Rnd 的分类结果的比较。可以看出，在 *L*30+*U*400、*L*50+*U*500 及 *L*200+*U*1000 上，SC-PMID-MI 的结果优于 SC-PART-Rnd。例如，在 *L*50+*U*500 子集上，SC-PMID-MI 的 Macro-Precision 比 SC-PART-Rnd 算法提高了 24.69%，Macro-F1 也提高了 6.61%，原因在于二者采用的特征子集分割方法不同。



（a）*L*30+*U*400 iterations=15

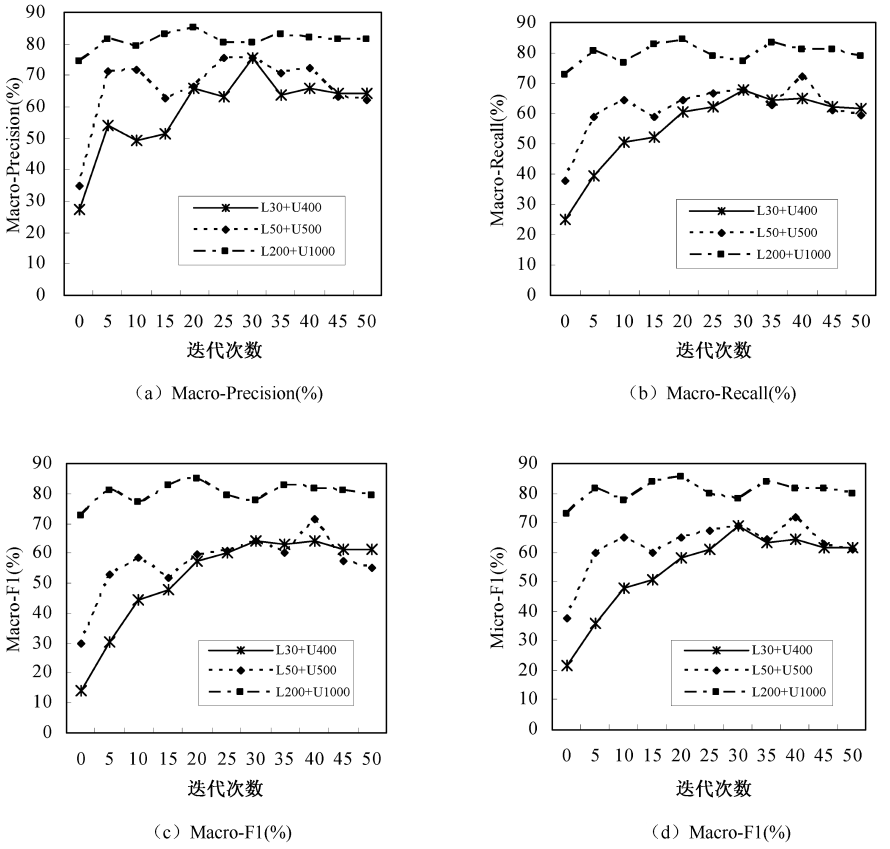
（b）*L*50+*U*500 iterations=10

（c）*L*200+*U*1000 iterations=25

图 5.2　SC-PMID-MI 算法与 SC-PART-Rnd 算法在不同的 D^l 和 D^u 上的分类结果比较

从图 5.3 可以看出初始时分别使用 30、50 和 200 个不同数量的标注文本，随着未标注文本数量的增加，未标注文本对 SC-PMID-MI 算法的分类结果的影响。例如，初始时使用 30 个已标注样本，随着未标注样本的加入，经过 30 次迭代，SC-PMID-MI 的 Macro-precision 和 Macro-F1 分别相对于 iterations=5

次时提高了 28.52% 和 33.24%。在 $L50+U500$ 上经过 40 次迭代，对比 iterations=5 时，SC-PMID-MI 算法的 Macro-F1 提高了 18.73%，而在使用 200 个标注文本和 1000 个未标注文本，经过 20 次迭代，对比 iterations=5 时，SC-PMID-MI 算法的 Macro-precision、Macro-F1 和 Micro-F1 分别提高了 4.16%、3.80%和 3.47%。



（a）Macro-Precision(%)　（b）Macro-Recall(%)

（c）Macro-F1(%)　（d）Macro-F1(%)

图 5.3　SC-PMID-MI 算法在不同的 D^l+D^u 上的分类结果比较

5.4.2　PMID-CHI 与 PART-Rnd 的实验比较

图 5.4 表明了由 PMID-CHI 算法划分的两个特征子集 V_1 和 V_2 上训练得到的两个 NB 分类器的 Precision、Recall 和 F1 值的差异，这间接地说明 PMID-CHI 划分算法能够有效地将一个特征集合划分成满足一定差异性的，或者说两个相对独立的特征子集。

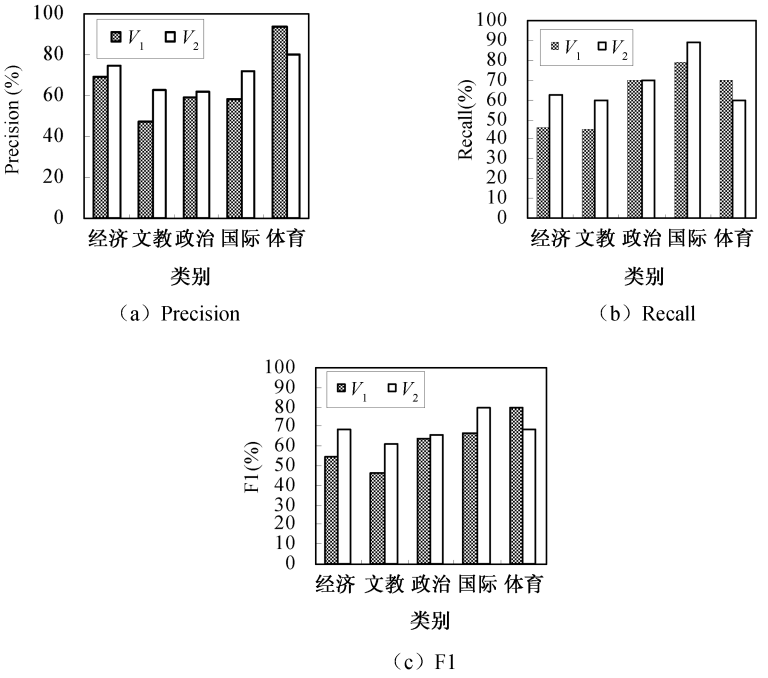
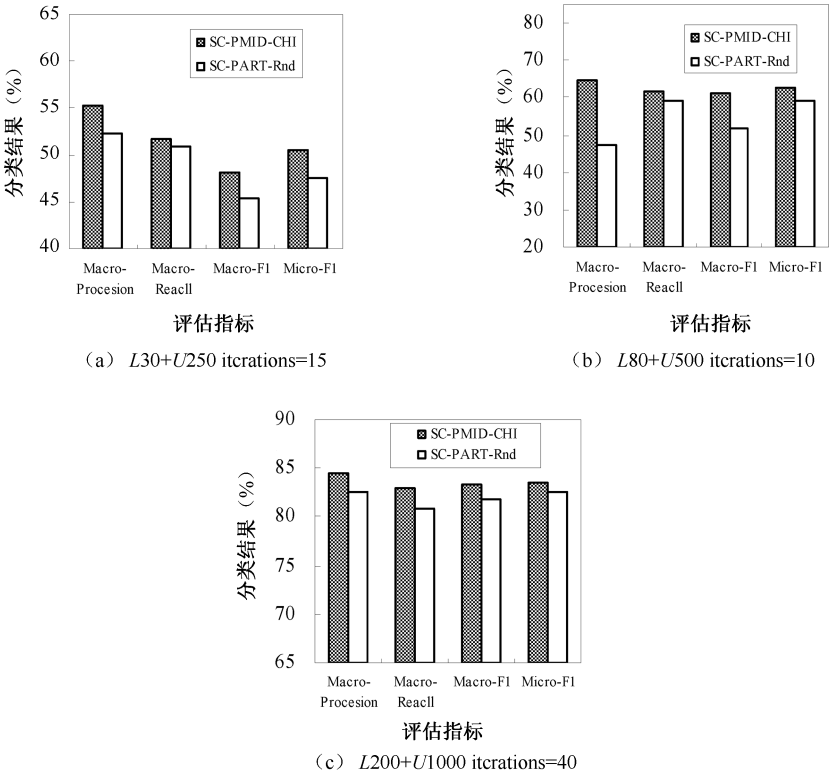


图 5.4　由 PMID-CHI 划分的特征子集 V_1 和 V_2 上的 NB 分类器的分类结果比较

图 5.5 给出的是基于 PMID-CHI 和基于随机分割算法 PART-Rnd 的两种半监督分类算法 SC-PMID-CHI 和 SC-PART-Rnd 的分类结果的比较。可以看出，在 $L30+U250$、$L80+U500$ 及 $L200+U1000$ 上，SC-PMID-CHI 的结果优于 SC-PART-Rnd。例如，在 $L80+U500$ 子集上，对比 SC-PART-Rnd 算法，SC-PMID-CHI 算法的 Macro-Precision 和 Macro-F1 分别提高了 17.07%和

3.58%。这是因为二者采用的特征子集分割方法不同，PMID-CHI 特征划分算法优于随机分割算法 PART-Rnd。



（a） $L30+U250$ itcrations=15　　（b） $L80+U500$ itcrations=10

（c） $L200+U1000$ itcrations=40

图 5.5　SC-PMID-CHI 算法与 SC-PART-Rnd 在不同的 D^l 和 D^u 上的分类结果比较

从图 5.6 可以看出初始时分别使用 30、50 和 200 个不同数量的标注文本，随着未标注文本的增加，未标注文本对 SC-PMID-CHI 算法的分类结果的影响。例如，初始时使用 30 个已标注样本，随着未标注样本的加入，经过 30 次迭代，SC-PMID-CHI 的 Macro-precision 和 Macro-F1 分别相对于 iterations=5 次时提高了 27.77%和 37.56%。在 $L50+U500$ 上经过 40 次迭代，比 iterations=0 时（即只使用 50 个标注文本），SC-PMID-CHI 算法的 Macro-Precision 和 Macro-F1 分别提高了 20.92%和 18.53%。

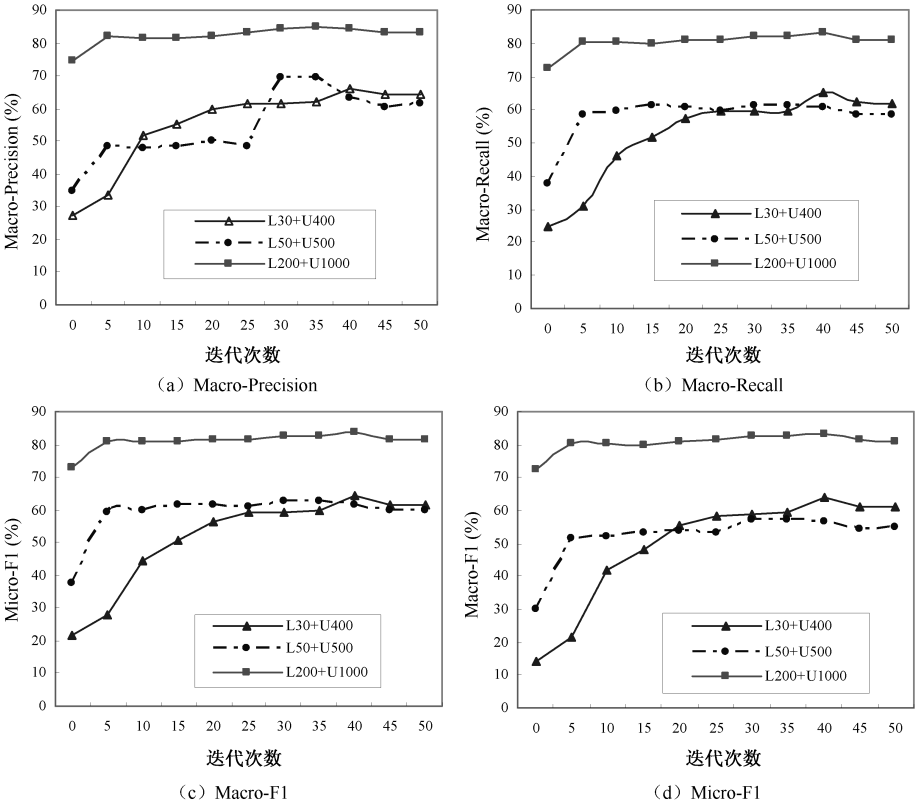


图 5.6　SC-PMID-CHI 算法在不同的 D^l 和 D^u 上的分类结果比较

5.4.3　PMID-MI、PMID-CHI 和 PART-Rnd 的实验比较

表 5.3 为在 *L*150+*U*500+*T*115 上，分别采用 PMID-MI、PMID-CHI 和 PART-Rnd 三种特征子集划分方法的性能比较。表 5.3 中第 1 列表示特征选择方法，其中“ECE/TF/1000”表示使用期望交叉熵 ECE、词频型计算公式 TF 进行权重调整和特征选择，保留 1000 个特征；“IG”表示信息增益，“WET”表示文本证据权。第 2 列对应三种特征子集划分方法。第 3 列表示特征子集

V_1 和 V_2 上分别使用 Naïve Bayesian（NB）和质心向量分类（CC）算法。第 4～8 列是每对特征子集上训练生成的基分类器间的差异性评估，间接地评估两个特征子集的独立性。计算 DM 时，$\alpha = \beta = \delta = 0.333, \gamma = 0$ 。

表 5.3　PMID-MI、PMID-CHI 和 PART-Rnd 划分算法下特征子集的独立性

特征选择	划分算法	基分类器	差异性评估方法				
			Q	ρ	Dis	DF	DM
ECE/TF/1000	PMID-MI	V1(NB)-V2(CC)	0.2896	0.1334	0.3913	0.1391	**0.1874**
	PMID-CHI	V1(NB)-V2(CC)	**0.1174**	**0.0493**	0.4000	**0.0957**	**0.0875**
	PART-Rnd	V1(NB)-V2(CC)	0.5392	0.2757	0.3304	0.1826	0.3325
IG/TF/1000	PMID-MI	V1(NB)-V2(CC)	0.3125	0.1375	0.4696	0.1826	**0.2109**
	PMID-CHI	V1(NB)-V2(CC)	**0.1885**	**0.0833**	**0.3913**	**0.1130**	**0.1283**
	PART-Rnd	V1(NB)-V2(CC)	0.5172	0.2658	0.3391	0.1913	0.3248
WET/TF/1200	PMID-MI	V1(NB)-V2(CC)	0.5670	0.2870	0.3130	0.1652	0.3327
	PMID-CHI	V1(NB)-V2(CC)	**0.2941**	**0.1323**	**0.3913**	**0.1304**	**0.1856**
	PART-Rnd	V1(NB)-V2(CC)	0.5464	0.2788	0.3304	0.1826	0.3359

5.4.4　SC-PMID-MI、SC-PMID-CHI 和 SC-PART-Rnd 的实验比较

图 5.7 和图 5.8 描述的分别是在 *L*150+*U*500+*T*115 上和 *L*30+*U*400+*T*115 数据子集上，使用望交叉熵 ECE、词频型计算公式 TF 进行权重调整和特征选择并保留了 1000 个特征后，SC-PMID-MI、SC-PMID-CHI 和 SC-PART-Rnd 三种算法的分类性能比较。

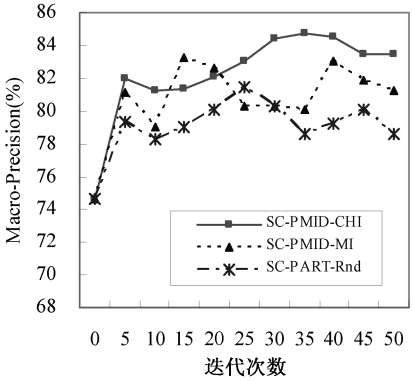


图 5.7　SC-PMIID-CHI、SC-PMID-MI 和 SC-PART-Rnd
在 *L*150+*U*500+*T*115 上的分类结果比较

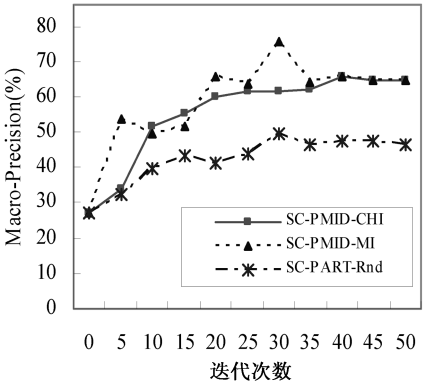


图 5.8　SC-PMIID-CHI、SC-PMID-MI 和 SC-PART-Rnd
在 *L*30+*U*400+*T*115 上的分类结果比较

从图 5.7 可以明显看出，在 *L*150+*U*500+*T*115 数据子集上，SC-PMID-CHI 的效果最好，其次是 SC-PMID-MI，SC-PART-Rnd 相对较差。例如，在迭代 40 次时，SC-PMID-CHI 和 SC-PMID-MI 的 Macro-Precision 分别是 84.56%和 83.05%，对比 SC-PART-Rnd 的 79.23%，分别提高了 5.33%和 3.82%。如图 5.8 所示，在 *L*30+*U*400+*T*115 数据子集上，SC-PMID-CHI 和 SC-PMID-MI 的分类精度相差不是很大，但是二者都优于 SC-PART-Rnd 算法。例如，迭代

25 次之后，SC-PMID-CHI 和 SC-PMID-MI 的分类精度比 SC-PART-Rnd 分别提高了 18.08% 和 19.87%。

分析原因，三种半监督分类算法采用的特征子集划分算法不同，所以分类效果不同。表 5.3 中的数据已经表明，在 $L150+U500+T115$ 上由 PMID-CHI 划分算法得到的一对特征子集之间的独立性最强，其次是 PMID-MI，最弱的是 PART-Rnd。特征子集之间存在较强的独立性，能够减少两个基分类器给同一个未标注文本都标注错误的可能性，帮助对方协同训练，从而提高半监督分类的精度。所以对 Co-training 的两个改进算法 SC-PMID-MI、SC-PMID-CHI 都是有效的，优于基于随机分割特征子集算法的 SC-PART-Rnd 算法，而且随着标注样本的增加，SC-PMID-CHI 算法优于 SC-PMID-MI 算法。

❑5.5 本章小结

本章提出了使用条件互信息、条件 χ^2 统计量两种方法评估两个特征之间的相互独立性，在特征集合上建立一种特征独立模型。基于该模型的特征子集划分方法——PMID 算法，可以有效地将一个特征视图划分成两个条件独立性较强的特征子图，从而改进了 Co-training 算法的分类效果。为了进一步验证和评估由此产生的两个特征子图之间的独立性，利用 Q 统计等多种方法计算两个基分类器之间的差异性，间接地评估二者的独立性。实验表明，在此基础上对 Co-training 的改进算法 SC-PMID-MI 和 SC-PMID-CHI，在结合未标注文本进行半监督分类时是有效的，而且随着标注样本的增加，SC-PMID-CHI 优于 SC-PMID-MI，二者都优于基于随机分割特征子集算法的 SC-PART-Rnd 算法。

基于特征独立模型的特征子集划分算法 PMID 存在的不足是，计算复杂度比较高，因为需要计算每对特征之间的相互独立性。虽然在算法中通过特征选择保留一定数量的初始特征、采用下三角矩阵存储相互独立性矩阵，减少了计算量，但效率还有待于提高。

第 6 章

基于投票信息熵和多视图的 AdaBoost 改进算法

文本分类技术的发展过程中出现了许多经典的分类方法，如决策树、贝叶斯方法、神经网络、K 近邻、支持向量机等[3][23][26][27][33]。然而在现有的经典方法中，没有一种分类方法总是优于其他分类方法。如何提高分类器的分类性能是分类器研究的主要目标之一，因此采用集成学习（ensemble learning）方法提高分类性能的研究受到了学术界的广泛关注。

最早由 Schapire 提出的 Boosting 算法[34][36]是一种有效的多分类器集成学习方法，目标是提高任何给定的学习算法的分类准确率。Boosting 算法有许多变形，Freund 和 Schapire 提出的 AdaBoost（Adaptive Boosting）[37,38]是比较常用的一种。AdaBoost 利用某种学习算法迭代产生一系列的基分类器，每个基分类器的训练依赖于在其之前产生的分类器的分类结果，每次迭代，按照一定的准则增加被错误分类样本的权重、减少被正确分类样本的权重、使得下一次迭代产生基分类器能够集中力量对这些错误样本进行学习，最终分类器通过单个基分类器的加权投票建立起来[37,38]。AdaBoost 在文本分类、图像分类、人脸检测等领域得到了广泛的应用[37,38] [85- 102]。

❑6.1 AdaBoost 算法

6.1.1 AdaBoost 算法描述

令训练样本集 $D=\{(x_1,y_1),\cdots,(x_N,y_N)\}$，$x_i\in X$，$X$ 表示训练样本集空间，$y_i\in Y=\{1,\cdots,L\}$ 是某一类别集。每次迭代的索引为 $t=1,2,\cdots,T$，AdaBoost 算法在训练样本上维护一套权重分布 W，每个训练样本 x_i 都对应一个权重 w_i^t，初始时，对所有 i 都有 $w_i^1=1/N$。AdaBoost 算法如表 6.1 所示。

最终分类器 h^* 由每个基分类器 $\{h_t\}_{t=1}^T$ 的投票来获得。如果 h_t 把某样本 x 决策为属于类 k，则在 k 上的投票就增加了 $\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$，$h^*$ 于是把 x 决策为总投票最多的类。

表 6.1 AdaBoost 算法

1. Input：N 个训练样本 $\{(x_1,y_1),\cdots,(x_N,y_N)\}$，其中 $x_i\in X$，类标签 $y_i\in Y=\{1,\cdots,L\}$；

迭代次数 T；

弱分类算法 Weaklearn。

2. 初始化：赋予每个样本相等的权重：$w_i^1=1/N$。

3. For $t=1$ to T **Do**

（1）在训练样本集 D 上，利用样本权重 w^t 和 Weaklearn 学习得到弱分类器 $h_t:X\to Y$；

（2）计算弱分类器 h_t 的错误率：$\varepsilon_t=\sum_{i=1}^N w_i^t I(h_t(x_i)\neq y_i)$，

$$I(h_t(x_i)\neq y_i)=\begin{cases}1 & ,h_t(x_i)\neq y_i\\ 0 & ,\text{其他}\end{cases}$$

if $\varepsilon_t>0.5$ then

重新初始化每个样本的权重 1/N;转向(1)。

（3）计算：$\alpha_t=\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$。

（4）根据错误率 ε_t 更新样本的权重：

$$w_i^{t+1}=\frac{w_i^t\exp(-\alpha_t I(h_t(x_i)=y_i))}{\sum_{i=1}^N w_i^t\exp(-\alpha_t I(h_t(x_i)=y_i))},\ i=1,\cdots,N$$

4. Output：最终分类器 $h^*(x)=\arg\max_{y\in Y}\sum_{t=1}^T\alpha_t I(h_t(x)=y)$

6.1.2　AdaBoost 提升 NB 文本分类器的问题

将 AdaBoost 算法直接用于 Naïve Bayesian（简称 NB）算法时，在每次迭代过程中训练样本 x_i 若被错误分类，权重 w_i^{t+1} 将增加，否则 w_i^{t+1} 将减少。但是在这种改变下，NB 分类器仍然非常稳定，每次迭代产生的基分类器没有很大的不同，因此通过投票决策也不能纠正彼此之间的分类错误，失去了组合的意义，使得 AdaBoost 不能有效地减少 NB 的分类错误。

学习算法的不稳定性是 AdaBoost 能否发挥作用的关键因素。所谓不稳定性，是指训练样本发生小的变动会明显影响分类结果[35][76]。AdaBoost [36-38] 作为常用的 Boosting 集成学习方法，能明显提高不稳定学习算法（如决策树、神经网络等）的分类正确率[36-38] [92,93]，但对稳定的学习算法（如 NB）效果不理想，甚至使其性能下降 [97,99]。NB 分类方法由于具有坚实的数学理论基础，且模型简单、效率较高，成为一种比较广泛使用的分类方法。但是 NB 分类算法的属性独立性假设限制了其分类精度的提高[3][96]，因此如何修正和改进 NB 分类算法引起了人们的持续关注。文献[97]将 NB 与决策树方法相结合，然后用 AdaBoost 技术提高其分类性能。文献[98]在若干属性子集上建立多个 NB 分类器，并由这些分类器形成集成分类器，进而提高分类精度。文献[99]用 AdaBoost 方法对 Friedman 等人提出的树状贝叶斯网络（TAN）进行组合，得到的 Boosting-MultiTAN 分类器对比标准的 TAN 分类器显现出较高的分类性能。不难看出，现有 NB 提升方法大多从改变 NB 分类器的结构出发，结合决策树或引入树状贝叶斯网络，使简洁的 NB 分类器变得较为复杂。

鉴于此，本章提出了基于特征多视图和投票信息熵的 BoostVE（Boosting algorithm based on multiple Views and vote Entropy）算法，在保持 NB 分类器简洁性的基础上，还具有如下特点。

1．利用特征多视图训练有差异的 NB 文本分类器

结合多种特征评估函数调整特征权重，在同一训练文本集上建立多个特征视图，每次迭代，在不同的特征视图上训练生成不同的 Naïve Bayesian 文

本分类器，作为 AdaBoost 的基分类器，增加 NB 基分类器之间的差异性。

2．基于投票信息熵改进传统的样本权重维护策略

传统 AdaBoost 每次迭代仅根据样本在上一轮是否被分错来调整权重，这种策略难以改变 NB 的稳定性。基于投票信息熵的样本权重维护新策略，不仅考虑样本在当前基分类器上是否被分错，还考虑该样本在前几轮基分类器上的投票分歧，分类差异较大的训练样本称为“有争议”（informative）样本或“不确定”（uncertain）样本，增加其权重。并选择适当的函数将样本权重引入 NB 分类器的参数中，对 NB 产生扰动，增加基分类器间的差异性。另外，NB 基分类器的置信度除了与基分类器错误率有关，还与其对基分类器间的差异性（diversity）的贡献有关。

❑6.2 利用特征评估函数构造多视图

特征权重调整是构造 NB 文本分类器的核心问题，传统的文本分类算法常采用 TF-IDF 函数[3]来计算特征的权重，但该函数难以从文本数据中区分出有用词条和噪声词条，故本书采用 TEF-WA[103]技术来调整特征的权重。TEF-WA 权值调整技术利用信息论中常用的评估函数代替逆文本频率给每个特征独立打分，评估分的高低能够很好地代表特征的重要性，根据评估分调整特征的权重。特征权重计算如式（6.1）所示。

$$ws_{ik} = \mathrm{TF}(s_{ik}) \times \mathrm{TEF}(s_{ik}) \qquad (6\text{-}1)$$

其中，$\mathrm{TF}(s_{ik})$ 表示特征 s_{ik} 在文本 x_i 中出现的词频。$\mathrm{TEF}(s_{ik})$ 表示常用的评估函数，用于给各个特征打分，反映特征与各类之间的相关程度。常用的评估函数有：文本频率、信息增益、期望交叉熵、互信息、文本证据权、几率比、χ^2 统计量[3] [23] [28][103]等，具体计算公式见第 3.2.2 节。

这里使用评估函数不仅是为了调整权重，更重要的是利用不同的评估函数，在同一训练集上构造不同的特征视图，由不同的特征视图可以训练生成多个有差异的 NB 文本基分类器，进而提高 NB 文本分类器的不稳定性。

令 V_t 表示根据某个评估函数建立的特征视图：

$$V_t = \mathrm{Create\_view}(\mathrm{TEF}, \mathrm{TForDF}, m) \qquad (6\text{-}2)$$

其中 TEF 表示某个评估函数，TForDF 表示特征权重计算时使用的是词频型（TF）还是文本型（DF）公式[3]，m 表示保留特征数量或比率。这里使用评估函数不仅是为了调整特征的权重，更重要的是利用不同的评估函数创建多个不同的特征视图，以便训练生成多个有差异的 NB 文本基分类器。这是因为，即使保留同样的特征数，当评估函数或权重计算公式不同时，对应的特征视图也是不同的，导致每个训练文本的特征向量表示有所不同，所以即便训练集中仍然是原来那些训练文本，但实际上文本的表示已经发生了很大的变化，训练生成的 NB 基分类器当然会有很大的区别。因此，可以通过调整式（6-2）中的参数，在同样的训练集 D 上建立多个不同的特征视图，从而使训练集每个文本的 VSM 向量在每次迭代中有更大的不同，生成有差异的 NB 基分类器，从而使 NB 变得不稳定。

在不同的特征视图上建立不同基分类器的目的，一方面是破坏 NB 的稳定性，使其符合 AdaBoost 算法对基分类器的要求；另一方面，无论是 Boosting 系列算法还是其他集成分类器，在选择基分类器时，除了要考虑基分类器的分类正确性外，还要考虑基分类器之间的差异。只有存在合理的差异，特别是负相关分类器，才能有效地、较大地提高集成分类器的分类精度[78] [80-83] [93]。

❑6.3　基于投票信息熵的样本权重维护新策略

AdaBoost 的核心思想是为训练样本集 D 维护一套权重分布，每次迭代，根据前一次基分类器产生的分类结果，增加被错误分类样本的权重、减少被正确分类样本的权重，目的是使得下一次迭代的基分类器更加关注被错误分类的训练样本。但是这种样本权重更新策略，一方面使 AdaBoost 对噪声非常敏感，导致 AdaBoost 退化；另一方面，按照这种策略调整样本权重，NB 基分类器依然趋于稳定，基分类器之间的差异性很小，使得集成的最终分类器不能纠正基分类器彼此之间的分类错误，提高了失效程序。

分析 AdaBoost 的权重调整策略，评判样本“最富信息”的标准是问题的关键。AdaBoost 的评判标准是训练样本在本轮迭代中是否分类正确，被分错的训练样本就是“最富信息”的样本。这种评判过于简单，没有充分利用前几轮基分类器对该样本的分类差异。

关于训练样本“最富信息”，即“价值”的评判标准有多种。在主动学习（Active Learning）方法中，Lewis 等人选取当前分类器最不能确定其类别的样本进行分类，一般称为 Uncertainty 采样[105]。Seung 等人提出的 QBC（Query By Committee）方法[106]，如果各分类器成员对某样本的分类不一致最大时，提交该样本进行类别标注。Dagan 等人用投票信息熵衡量分类委员会（Classification Committee）中的成员对某个单词词性标注的分歧[107]。受此启发，AdaBoost 在迭代过程中也会生成多个基分类器，多个基分类器对训练样本的分类分歧蕴含了样本的“价值”。因此，可以利用前几轮迭代的多个基分类器对每个训练样本给出的类别投票分布，计算每个样本的投票信息熵（Vote Entropy），度量其“价值”。

6.3.1 投票信息熵

令训练样本集 $D=\{(x_i,y_i)\}_{i=1}^{N}$，类别集合 $C=\{c_j\}_{j=1}^{L}$，N、L 为样本数和类别数。$\boldsymbol{y}_i=(y_i^1,\cdots,y_i^L)\in\{0,+1\}^L$，如果 x_i 属于第 j 个类别 c_j，$y_i^j=+1$，否则 $y_i^j=0$，且 $\sum_{j=1}^{L}y_i^j=1$，即每个样本 x_i 只能属于唯一的一个类别。$H=\{h_1,\cdots,h_T\}$ 为 Boosting 迭代中的一组基分类器，基分类器 h_t 对 x_i 的分类 $h_t(x_i)$，输出对应 $\hat{\boldsymbol{y}}_i=(\hat{y}_i^1,\cdots,\hat{y}_i^L)\in\{0,+1\}^L$，令 $h_{ti}^j\equiv\hat{y}_i^j$，表示 h_t 把样本 x_i 预测为 c_j 类，同样 $\sum_{j=1}^{L}\hat{y}_i^j=1$。

定义 6.1 令 l_{ji} 表示前 t 次迭代中把样本 x_i 分为 c_j 类的基分类器数，计算如下：

$$l_{ji}=\sum_{s=1}^{t}h_{si}^j\ ,\ i=1,\cdots,N;j=1,\cdots,L;t=3,\cdots,T \tag{6-3}$$

定义 6.2 第 t 次迭代，前 t 个基分类器对样本 x_i 的分类分歧由投票熵（Vote Entropy）评估：

$$\mathrm{VE}_{ti}=-\sum_{j=1}^{L}\frac{l_{ji}+\vartheta}{t}\log\frac{l_{ji}+\vartheta}{t},\ (t>2) \tag{6-4}$$

式中，$0<\vartheta\ll1$，是平滑因子，避免 log0，实验中取 $\vartheta=0.0001$。

VE_{ti}=0，表示 t 个基分类器对样本 x_i 的分类意见一致。当 t 个基分类器

分歧最大时，在两类问题中表现为半数基分类器将样本 x_i 标注为正类，另外半数基分类器将 x_i 标注为负类。在多类问题中，表现为对样本 x_i 的分类，多个基分类器在 L 个类别上的投票均匀分布，即

$$\mathrm{VE}_{ti\ \max}=\begin{cases}\log t & ,\ t<L\\ \log L & ,\text{其他}\end{cases}$$

所以当 $t<L$ 时，归一化因子 $Ze=\log t$，否则 $Ze=\log L$，式（6-4）修改如下：

$$\mathrm{VE}_{ti}=-\frac{1}{Ze}\sum\nolimits_{j=1}^{L}\frac{l_{ji}+\vartheta}{t}\log\frac{l_{ji}+\vartheta}{t},\ (t>2) \tag{6-5}$$

$0\leqslant \mathrm{VE}_{ti}\leqslant 1$，显然，投票信息熵 VE_{ti} 衡量的是 t 个基分类器对样本 x_i 分类的分歧，VE_{ti} 越大，t 个基分类器对样本 x_i 的分类分歧越大，x_i 就越“富有信息”，对构建分类器越有“价值”。

6.3.2　基于投票信息熵的样本权重维护新策略

训练样本的权重维护（re-weighting）新策略综合考虑以下三个因素。

① 第 t 次迭代，训练样本 x_i 被分类正确还是错误 $h_t(x_i)\boldsymbol{y_i}^{\mathrm{T}}$；

② 与基分类器 h_t 的分类错误率 ε_t 有关的 α_t；

③ 前 t 个基分类器对每个训练样本 x_i 的投票信息熵 VE_{ti}。这是与 AdaBoost 的不同之处，AdaBoost 只考虑了①和②。

如果 h_t 对样本 x_i 分类正确，$h_t(x_i)\boldsymbol{y_i}^{\mathrm{T}}=1$，否则 $h_t(x_i)\boldsymbol{y_i}^{\mathrm{T}}=0$。为叙述简便，令 $u_{ti}\equiv h_t(x_i)\boldsymbol{y_i}^{\mathrm{T}}$。加权训练错误（training error）$\varepsilon_t=\sum_{i=1}^{N}w_i^t(1-u_{ti})$。这样，第 t+1 次迭代，训练样本 x_i 的权重 w_i^{t+1} 计算如下：

$$w_i^{t+1}=w_i^t\mathrm{e}^{(1-2u_{ti})(\alpha_t+\eta\mathrm{VE}_{ti})}/Z_t \tag{6-6}$$

其中，$0\leqslant \mathrm{VE}_{ti}\leqslant 1$，如果 $t\leqslant 2$，令 $\mathrm{VE}_{ti}=0,\ i=1,\cdots,N$，$Z_t$ 是归一化因子。如果 h_t 对样本 x_i 分类错误 $1-2u_{ti}=1$，否则 $1-2u_{ti}=-1$。

分析式（6-6），w_i^{t+1} 的更新除了与 $(1-2u_{ti})$ 有关，还由下面两项决定。

① α_t：$\alpha_t = \frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$，与 h_t 的分类错误率有关，但是 α_t 对每个训练样本 x_i 来说是相同的。

② VE_{ti}：VE_{ti} 与具体的训练样本 x_i 有关，蕴含了前 t 个基分类器对样本 x_i 的投票分歧。VE_{ti} 越大，x_i 就越“富有信息”，对构建基分类器就越有“价值”。可见 VE_{ti} 对识别“富有信息”的训练样本非常有用。

上述两项在 w_i^{t+1} 中的作用由因子 η 调节。当 $\eta = 0$ 时，式（6-6）与经典的 AdaBoost 一致。迭代过程中，η 的取值与最小化训练错误边界有关，将在 6.4 节详细讨论。

令 $\beta_{ti} = \alpha_t + \eta \mathrm{VE}_{ti}$，$w_i^{t+1}$ 的更新修改如下：

$$w_i^{t+1} = w_i^t \mathrm{e}^{(1-2u_{ti})\beta_{ti}} / Z_t \qquad (6\text{-}7)$$

可以看出，新策略评判训练样本“价值”的标准与传统 AdaBoost 不同，不仅考虑训练样本在当前基分类器上是否分错，还考虑了前几轮基分类器对每个训练样本的投票分歧。通过引入投票分歧 VE_{ti}，不仅增加了基分类器间的差异性（diversity），而且通过加强对有争议的训练样本的学习，减少了训练错误（training error），以提高基分类器的分类正确性（accuracy）。

6.3.3 样本权重对 NB 文本分类器的扰动

这里的目标是改进 AdaBoost 算法对 NB 分类算法的提升效果。NB 分类算法可用于文本分类、图像分类、图像识别等多个领域，这里主要考虑文本分类问题。文本分类中，文本表示通常采用向量空间模型（Vector Space Model, VSM）法[10]，每个训练文本 x_i 用一个特征向量 $x_i = (ws_{i1}, ws_{i2}, \cdots, ws_{ik}, \cdots, ws_{im})$ 描述，ws_{ik} 表示从文本 x_i 中提取的特征 s_{ik} 的权重。

NB 分类算法作为 Boosting 的基分类器，训练时要计算两个参数：①每个类的先验概率 $P(c_j)$；②每个特征 s_{ik} 基于每个类的条件概率 $P(s_{ik} \mid c_j)$。

NB 文本分类器对未知类别的文本 x_i 的类别预测就是求 $c = \arg\max_j P(c_j \mid x_i)$。根据独立假设和 Bayesian 公式有

$$P(c_j \mid x_i)=\frac{P(c_j)\prod_{k=1}^{|x_i|}P(s_{ik} \mid c_j)}{\sum_{r=1}^{L}P(c_r)\prod_{k=1}^{|x_i|}P(s_{ik} \mid c_r)} \tag{6-8}$$

先验类概率 $P(c_j)$ 为

$$P(c_j)=\frac{1+\sum_{i=1}^{N}P(y_i^j=+1 \mid x_i)}{L+N} \tag{6-9}$$

每个特征 s_{ik} 基于类的条件概率 $P(s_{ik} \mid c_j)$ 为

$$P(s_{ik} \mid c_j)=\frac{1+\sum_{i=1}^{N}\delta(s_{ik},x_i)P(y_i^j=+1 \mid x_i)}{|V|+\sum_{b=1}^{|V|}\sum_{i=1}^{N}\delta(s_{ib},x_i)P(y_i^j=+1 \mid x_i)} \tag{6-10}$$

其中，$\delta(s_{ik},x_i)$ 表示特征 s_{ik} 在文本 x_i 中出现的次数。当文本 x_i 属于类别 c_j 时，$P(y_i^j=+1 \mid x_i)=1$，否则 $P(y_i^j=+1 \mid x_i)=0$。V 代表训练文本集的特征集合，N 和 L 分别是训练文本数和类别数。

对文本 x_i 分类时，比较文本 x_i 属于每个类的后验概率 $P(c_j \mid x_i)$，将其分到后验概率最大的那个类。

$$\begin{aligned}h(x_i)&=\arg\max_j P(c_j \mid x_i)\\&=\arg\max_j P(c_j)\prod_{k=1}^{|x_i|}P(s_{ik} \mid c_j)\end{aligned}$$

计算时取对数：

$$h(x_i)=\arg\max_j\{\log[P(c_j)]+\sum_{k=1}^{|x_i|}\log P(s_{ik} \mid c_j)\} \tag{6-11}$$

Boosting 迭代过程中为每个训练样本维护的权重为 w_i^t，引入 NB 分类器的参数 $P(c_j)$ 和 $P(s_{ik} \mid c_j)$，则式（6-9）和式（6-10）分别修改如下：

$$P(c_j)=\sum_i^N w_i^t P(y_i^j=+1 \mid x_i) \tag{6-12}$$

$$P(s_{ik} \mid c_j)=\frac{1+\sum_{i=1}^{N}\delta(s_{ik},x_i)\mathrm{e}^{w_i^t}P(y_i^j=+1 \mid x_i)}{|V|+\sum_{b=1}^{|V|}\sum_{i=1}^{N}\delta(s_{ib},x_i)\mathrm{e}^{w_i^t}P(y_i^j=+1 \mid x_i)} \tag{6-13}$$

这样，每次迭代，随着样本权重 w_i^t 的更新，NB 分类器的参数 $P(c_j)$ 和 $P(s_{ik}|c_j)$ 也随之变化，对 NB 文本分类器产生扰动，增加了 NB 基分类器之间的差异性。

❑6.4 BoostVE 算法

6.4.1 BoostVE 算法

结合 6.2 节的特征多视图与 6.3 节提出的基于投票信息熵的权重维护新策略，改进的 BoostVE 算法如表 6.2 所示。对比 AdaBoost 算法，其不同之处在于步骤（3.1*）、（3.2*）、（3.5*）～（3.7*）和（4*）。

表 6.2 BoostVE 算法

1. Input：$D=\{(x_1,y_1),\cdots,(x_N,y_N)\}$，$C=\{c_1,\cdots,c_L\}$，$\boldsymbol{y_i}=(y_i^1,\cdots,y_i^L)\in\{0,+1\}^L$

迭代次数 T。

2. Initialize: $w_i^1=1/N$，$i=1,\cdots,N$。

3. For t = 1 to T Do

（3.1*）在训练文本集 D 上选择不同的特征评估函数、TF/DF、特征数，建立不同的特征视图 V_t;

（3.2*）利用权重分布 W^t 和 V_t，生成新的 NB 基分类器 h_t;

（3.3）用 h_t 对每个训练文本 x_i 分类，输出 $h_t(x_i)=\hat{\boldsymbol{y}}_{\boldsymbol{i}}=(\hat{y}_1,\cdots,\hat{y}_L)\in\{0,+1\}^L$；

$$\hat{y}_i^j=\begin{cases}1, & j=\arg\max_{j'}\{\log[P(c_{j'})]+\sum_{k=1}^{|x_i|}\log P\left(s_{ik}\mid c_{j'}\right)\}\\ 0, & \text{其他}\end{cases}$$

（3.4）计算基分类器 h_t 的错误率：$\varepsilon_t=\sum_{i=1}^{N}(1-u_{ti})w_i^t$；

if（$\varepsilon_t>0.5$） $w_i^t=1/N$，转向（3.1*）;

（3.5*）if（t>2）

{计算每个文本 x_i 在前 t 次迭代的投票信息熵 VE_{ti}，如式（6.4）所示;

计算加入 h_t 的后的平均投票信息熵 $\overline{\mathrm{VE}_t}=\frac{1}{N}\sum_{i=1}^{N}\mathrm{VE}_{ti}$ }

（3.6*）计算 h_t 的置信度 β_t：$\beta_t=\alpha_t+\eta\overline{\mathrm{VE}_t}=\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)+\eta\overline{\mathrm{VE}_t}$，

调整 η 的取值使最小错误率的上界下降 ；

（3.7*）更新权重：

if（t>2） $w_i^{t+1}=w_i^t\mathrm{e}^{(1-2u_{ti})(\alpha_t+\eta\mathrm{VE}_{ti})}/Z_t$

else $w_i^{t+1}=w_i^t\mathrm{e}^{(1-2u_{ti})\alpha_t}/Z_t$

4*.Output: 集成分类器：

$H(x_i)=\sum_{t=1}^{T}\beta_t h_t(x_i)$，$x_i$ 被决策为 $c^*=\arg\max_j\{\sum_t^T\beta_t h_{ti}^j\}$

步骤（3.1*）通过组合选择不同的评估函数、TF/DF 公式、特征数，建立不同的特征视图。步骤（3.2*）在不同的特征视图 V_t 上，结合训练文本的权重分布 W^t 训练生成不同的 NB 文本基分类器，并且训练文本的权重嵌入 NB 分类器的 $P(c_j)$ 和 $P(s_{ik}\mid c_j)$，对 NB 产生扰动，增大基分类器之间的差异性。

步骤（3.5*）计算每个文本 x_i 在前 t 次迭代的投票信息熵 VE_{ti}，即 t 个基分类器对 x_i 的投票分歧，以及加入 h_t 后的平均投票信息熵 $\overline{\mathrm{VE}_t}$。

步骤（3.6*）中基分类器 h_t 的置信度 β_t 的计算采用了与 AdaBoost 不同的方法：

$$\beta_t=\alpha_t+\eta\overline{\mathrm{VE}_t}=\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)+\eta\overline{\mathrm{VE}_t} \tag{6-14}$$

其中，前一项与错误率 ε_t 有关，代表基分类器 h_t 的正确性（accuracy）；后一项 $\overline{\mathrm{VE}_t}=\frac{1}{N}\sum_{i=1}^{N}\mathrm{VE}_{ti}$，是 t 个基分类器在所有训练文本上的平均投票分歧，代表加入 h_t 后 t 个基分类器间的差异性（diversity）。这样，在错误率 ε_t 相同的情况下，那些能够增大基分类器间差异性的基分类器会获得更大的权重。η 的取值将在 6.4.2 节详细讨论。

最终分类器 H 由 $\{h_t\}_{t=1}^{T}$ 加权投票获得：

$$H(x_i)=\sum_{t=1}^{T}\beta_t h_t(x_i) \tag{6-15}$$

输出 $\hat{y}_i=(\hat{y}_i^1,\cdots,\hat{y}_i^L)\in\mathbb{R}^L$，$H_i^j\equiv\hat{y}_i^j=\sum_t^T\beta_t h_{ti}^j$，$j=1,\cdots,L$。$x_i$ 被决策为 $c^*=\arg\max_j\{\sum_t^T\beta_t h_{ti}^j\}$。

6.4.2　BoostVE 算法的最小训练错误上界

Freund 和 Schapire 在文献[37，38]已经证明 AdaBoost 算法的最小训练错误上边界（Upper Bounds for Training Errors）满足：

$$\varepsilon\leqslant 2^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)} \tag{6-16}$$

下面将论证 BoostVE 算法的最小训练错误上边界，论证方法受文献[94]启发。

引理 6.1 令 $r \in [0,1]$，那么

$$e^r \leqslant 1+(e-1)r \tag{6-17}$$

证明：不等式（6-17）的右边代表过点（0，1）和（1，e）之间的线段上的点，而 e^r 是关于 r 的凸函数，所以对 $r \in [0,1]$，式（6-17）成立。

定理 6.1 令 ε 为由 BoostVE 算法生成的集成分类器的训练错误，令 ε_t（$t=1,\cdots,T$）为执行 BoostVE 算法时第 t 次迭代的基分类器 h_t 的加权训练错误，β_t 表示 h_t 的置信度，那么下面的不等式成立。

$$\varepsilon \leqslant \prod_{t=1}^{T}(1-(e-1)\beta_t(1-2\varepsilon_t)) \tag{6-18}$$

证明：根据式（6-7），BoostVE 算法初始化后，有

$$w_i^2 = \frac{w_i^1 e^{(1-2u_{ti})\beta_{ti}}}{\sum_{j=1}^{N} w_j^1 e^{(1-2u_{1j})\beta_{1j}}} \tag{6-19}$$

令式（6-20）表示第 t 次迭代的系数：

$$C_t = \sum_{j=1}^{N} w_j^t e^{(1-2u_{tj})\beta_{tj}} \tag{6-20}$$

那么，训练样本权重的一般公式为

$$w_i^{t+1} = w_i^1 \prod_{s=1}^{t} \frac{e^{(1-2u_{si})\beta_{si}}}{C_s} \tag{6-21}$$

令 $D^{(-)}$ 表示训练集中被分错的样本子集，集成分类器的的分类错误用样本集的初始 w_i^1 表示如下：

$$\varepsilon = \sum_{x_i \in D^{(-)}} w_i^1 \tag{6-22}$$

因为在 BoostVE 算法中每次迭代所有权重的和为 1，即

$$1 = \sum_{i=1}^{N} w_i^{T+1} \geqslant \sum_{x_i \in D^{(-)}} w_i^{T+1} = \sum_{x_i \in D^{(-)}} w_i^1 \prod_{t=1}^{T} \frac{e^{(1-2u_{ti})\beta_{ti}}}{C_t} \tag{6-23}$$

所有基分类器集合 $H=\{h_1,\cdots,h_T\}$，根据对某个样本 $x_i \in D$ 的分类输出，可以被分成 3 个子集：

① $H^w \subset H$，其输出是获胜的类别标签的基分类器的子集；

② $H^+ \subset H$，其输出是样本真正的类别标签的基分类器的子集;

③ $H^{-} \subset H$，其输出是另外某个错误的类别标签的基分类器的子集。

当样本 x_i 被集成分类器分错时，错误类别标签对应的分数（也就是基分类器的置信度 β_t 的和）必然大于其他类别标签对应的分数，包括正确的类别，即

$$\sum_{h_t \in H^w} \beta_t \geqslant \sum_{h_t \in H^+} \beta_t \tag{6-24}$$

两边同时加上 $\sum_{h_t \in H^w}(.)+\sum_{h_t \in H^-}(.)$，得

$$2\sum_{h_t \in H^w} \beta_t + \sum_{h_t \in H^-} \beta_t \geqslant \sum_{t=1}^{T} \beta_t \tag{6-25}$$

注意式（6-25）的左边，是所有错误类的投票权重的两倍，即

$$2\sum_{t=1}^{T}(1-u_{ti})\beta_t \geqslant \sum_{t=1}^{T} \beta_t \tag{6-26}$$

$$\sum_{t=1}^{T}(1-u_{ti})\beta_t \geqslant \frac{1}{2}\sum_{t=1}^{T} \beta_t \tag{6-27}$$

式（6-27）两边同时加上 $\sum_{t=1}^{T}(-u_{ti}\beta_t)$，得

$$\sum_{t=1}^{T}(1-2u_{ti})\beta_t \geqslant 0 \tag{6-28}$$

那么

$$\mathrm{e}^{\sum_{t=1}^{T}(1-2u_{ti})\beta_t} \geqslant 1$$

$$\prod_{t=1}^{T}\mathrm{e}^{(1-2u_{ti})\beta_t} \geqslant 1 \tag{6-29}$$

由式（6-22）、式（6-23）和式（6-29）得

$$1 \geqslant \sum_{x_i \in D^{(-)}} w_i^1 \prod_{t=1}^{T} \frac{\mathrm{e}^{(1-2u_{ti})\beta_{ti}}}{C_t} \doteq \sum_{x_i \in D^{(-)}} w_i^1 \prod_{t=1}^{T} \frac{\mathrm{e}^{(1-2u_{ti})\beta_t}}{C_t}$$

$$\geqslant \left(\sum_{x_i \in D^{(-)}} w_i^1\right) \prod_{t=1}^{T} \frac{1}{C_t} = \varepsilon \prod_{t=1}^{T} \frac{1}{C_t}$$

所以

$$\varepsilon \leqslant \prod_{t=1}^{T} C_t \tag{6-30}$$

由引理 6.1 可得

$$C_t = \sum_{j=1}^{N} w_j^t \mathrm{e}^{(1-2u_{tj})\beta_{tj}} \doteq \sum_{j=1}^{N} w_j^t \mathrm{e}^{(1-2u_{tj})\beta_t}$$

$$\leqslant \sum_{j=1}^{N} w_j^t (1+(\mathrm{e}-1)(1-2u_{tj})\beta_t)$$

$$
\begin{aligned}
&= \sum_{j=1}^{N} w_j^t + (\mathrm{e}-1)\beta_t \sum_{j=1}^{N} w_j^t (1-2u_{tj}) \\
&= 1 + (\mathrm{e}-1)\beta_t (\varepsilon_t - (1-\varepsilon_t)) \\
&= 1 - (\mathrm{e}-1)\beta_t (1-2\varepsilon_t)
\end{aligned} \tag{6-31}
$$

合并式（6-30）和式（6-31），得

$$
\varepsilon \leqslant \prod_{t=1}^{T} (1-(\mathrm{e}-1)\beta_t (1-2\varepsilon_t))
$$

因此，式（6-18）成立，证明完毕。

与 AdaBoost 算法的错误边界理论类似，定理 6.1 表明，如果基分类器比随机猜测好，所提出的 BoostVE 算法能够呈指数减少训练错误。

注意，训练错误 ε 的上界是 β_t 的线性函数，假设 $\varepsilon_t < 0.5$，β_t 越大，最小训练错误的上界就越小，训练错误将越小。

下面来比较 BoostVE 算法和 AdaBoos 算法的训练错误最小上界。

定理 6.2 令 h_t 表示运行 BoostVE 算法第 t 次迭代的基分类器，ε_t 表示 h_t 的加权训练错误，令 ε 表示训练错误的上界，那么当 $\varepsilon_t < 0.5$ 时，下面的不等式成立。

$$
\varepsilon < (\mathrm{e}-1)^T \prod_{t=1}^{T} \sqrt{\varepsilon_t (1-\varepsilon_t)} \tag{6-32}
$$

证明：当 $\varepsilon_t \in (0, 0.5)$ 时，如果如下不等式成立，

$$
\frac{1-(\mathrm{e}-1)\sqrt{\varepsilon_t(1-\varepsilon_t)}}{(\mathrm{e}-1)(1-2\varepsilon_t)} - \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \leqslant 0 \tag{6-33}
$$

又因为 $\overline{VE_t} > 0$，那么 $\forall \eta > 0$，得

$$
\eta\overline{\mathrm{VE}_t} > \frac{1-(\mathrm{e}-1)\sqrt{\varepsilon_t(1-\varepsilon_t)}}{(\mathrm{e}-1)(1-2\varepsilon_t)} - \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{6-34}
$$

当 $\varepsilon_t \in (0, 0.5)$ 时，如果不等式（6-33）不成立，即

$$
\frac{1-(\mathrm{e}-1)\sqrt{\varepsilon_t(1-\varepsilon_t)}}{(\mathrm{e}-1)(1-2\varepsilon_t)} - \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0
$$

又因为 $\overline{\mathrm{VE}_t} > 0$，那么 $\exists \eta > 0$，不等式（6-34）成立。

因此，当 $\varepsilon_t \in (0, 0.5)$ 时，因为 $\overline{\mathrm{VE}_t} > 0$，那么 $\exists \eta > 0$，不等式（6-34）成立。

即　　$1-(\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)+\eta\overline{\mathrm{VE}_t})(\mathrm{e}-1)(1-2\varepsilon_t)<(\mathrm{e}-1)\sqrt{\varepsilon_t(1-\varepsilon_t)}$　　（6-35）

由式（6-35）、式（6-18）和式（6-14）得，下面的不等式成立：

$$\varepsilon<(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

因此定理 6.2 得证。

注意，定理 6.2 表明 BoostVE 算法的训练错误上界为 $(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}$。换句话说，如果基分类器优于随机猜测，通过选择适当的 β_t，也就是根据 $\overline{\mathrm{VE}_t}$ 选择适当的 η 的值，能够保证 BoostVE 算法的训练错误上界满足 $\varepsilon<(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}$。在实验中，只要限制 $\varepsilon_t\in(0,0.375)$，不等式（6-33）必然成立，那么 $\forall\eta>0$，可以确定 $\varepsilon<(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}$ 成立。

定理 6.3　令 h_t 表示运行 BoostVE 算法第 t 次迭代的基分类器，ε_t 表示 h_t 的加权训练错误，令 ε 表示训练错误的上界，那么当 $\varepsilon_t<0.5$ 时，BoostVE 算法的训练错误上界优于 AdaBoost 算法。

证明：因为

$$(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}<2^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}\qquad(6\text{-}36)$$

那么，由定理 6.2 和式（6-36）知下面的不等式成立。

$$\varepsilon<(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}<2^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}\qquad(6\text{-}37)$$

由以上的理论分析得到的重要结论是，BoostVE 算法的训练错误上界可以降低为 $(\mathrm{e}-1)^T\prod_{t=1}^{T}\sqrt{\varepsilon_t(1-\varepsilon_t)}$，优于 AdaBoost 算法。

❑6.5　实验结果及其分析

为了验证所提 BoostVE 算法对 NB 分类算法提升的有效性，在通用数据

集 20-newsgroups dataset[①]上比较了三种 Boosting 算法对 NB 文本分类器的提升效果，验证了 BoostVE 算法对 NB 文本分类器的提升效果。

① AdaBoost 算法使用传统的 re-weighting 策略和单个特征视图；

② AdaBoost-MV 算法使用传统的 re-weighting 策略和多个特征视图；

③ BoostVE 算法使用基于投票熵的 re-weighting 新策略和多个特征视图。

以上三种算法分别称为 AdaBoost、AdaBoost-MV 和 BoostVE（如表 6.3 所示）。

表 6.3 AdaBoost、AdaBoost-MV 和 BoostVE 算法比较

算 法	w_i^{t+1} 更新策略	h_t 的信任度	特征视图
AdaBoost	$w_i^{t+1}=w_i^t e^{(1-2u_{ti})\alpha_t}/Z_t$	$\alpha_t=\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$	单个特征视图
AdaBoost-MV	$w_i^{t+1}=w_i^t e^{(1-2u_{ti})\alpha_t}/Z_t$	$\alpha_t=\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$	多个特征视图
BoostVE	If （$t>2$） $w_i^{t+1}=w_i^t e^{(1-2u_{ti})(\alpha_t+\eta VE_{ti})}/Z_t$ Else $w_i^{t+1}=w_i^t e^{(1-2u_{ti})\alpha_t}/Z_t$	$\beta_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)+\eta\overline{\mathrm{VE}}_t$	多个特征视图

因为原始的 20-newsgroups 数据集规模比较庞大，所以选取了其中的三个子集，如表 6.4 所示。在下面的叙述中，令 *L*n 表示包含 *n* 篇文本的训练子集，*Tm* 表示包含 *m* 篇文本的测试子集。实验结果如图 6.1～图 6.6 和表 6.5～表 6.6 所示，评估方法采用 Macro-Precision、Macro-Recall、Macro-F1、Precision、Recall 及 F1。

表 6.4 20-newsgroups 数据集子集

数 据 子 集	类 别	训练样本数	测试样本数
subset-Sci （L400+T624）	sci.crypt sci.electronics sci.med sci.space	400（100 per class）	624（156 per class）
subset-Rec （L464+T592）	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	464（116 per class）	592（148 per class）

① http://people.csail.mit.edu/jrennie/20Newsgroups/

续表

数 据 子 集	类　　别	训练样本数	测试样本数
subset-Talk （L300+T96）	talk.politics.misc talk.politics.guns talk.politics.mideast	300（100 per class）	96（32 per class）

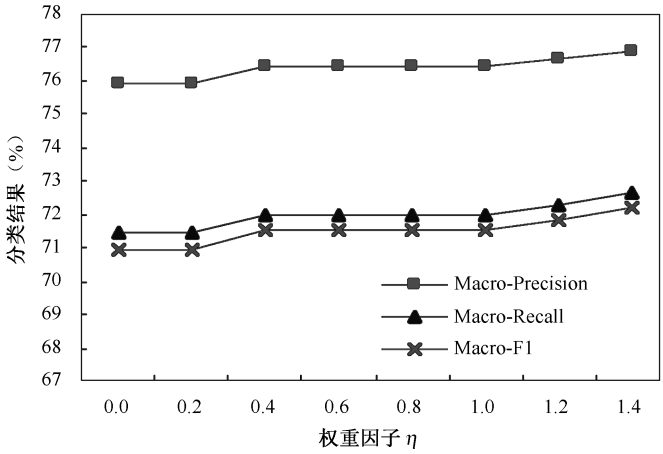
6.5.1　参数η对 BoostVE 算法性能的影响

对基分类器h_t的信任权重β_t的计算，BoostVE 算法采用了与 AdaBoost 算法不同的方法，遵循了集成分类器的原则，不仅考虑基分类器的正确性，还要考虑基分类器间的差异性。即β_t不仅与基分类器的错误率ε_t有关，还与其对增加基分类器间差异性的贡献有关。在错误率ε_t相同的情况下，能够增大基分类器间的差异性的基分类器会获得更大的权重。因而能够有效提升集成分类器的分类性能。

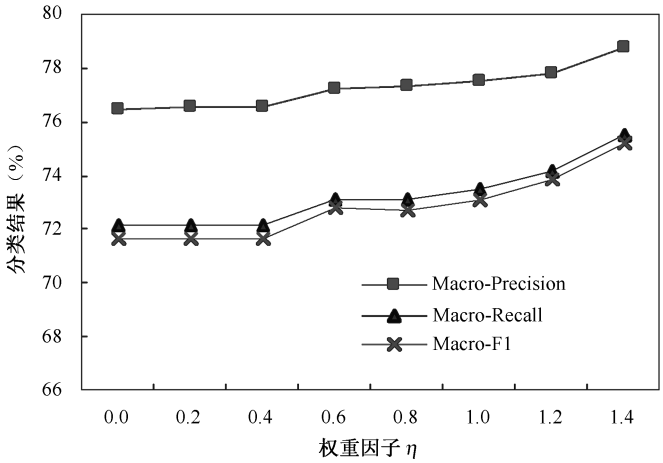
这里引入权重因子η来平衡正确性和差异性的关系。实验中令η取 0～1.6 之间的不同的值，实验结果如图 6.1～图 6.3 所示，揭示了η对 BoostVE 算法的影响，也就是平均投票熵$\overline{\mathrm{VE}_t}$对 BoostVE 算法提升 NB 文本分类器的影响。

通过比较采用平均投票熵$\overline{\mathrm{VE}_t}$（即$\eta>0$）和不采用$\overline{\mathrm{VE}_t}$（即$\eta=0$）的情况，从图 6.1～图 6.3 可以看出，只要选择适当的η，总的来说采用$\overline{\mathrm{VE}_t}$有助于 BoostVE 提升 NB 分类器的结果。在数据集 subset-Rec 和 subset-Sci 上，随着η值的增加，BoostVE 的结果得到了提高。例如，在 subset-Rec 上，$\eta=1.4$时，BoostVE 算法迭代 15 次后其 Macro-F1 比$\eta=0$时提高了 3.57%；在 subset-Sci 上，$\eta=1.6$时，BoostVE 算法迭代 11 次后其 Macro- Precision 比$\eta=0$时提高了 4.46%。

这说明，通过引入平均投票熵$\overline{\mathrm{VE}_t}$，BoostVE 算法的基分类器置信度的计算公式优于 AdaBoost 传统的计算公式。依据是定理 6.2 和定理 6.3，选择适当的η值，可以减小 BoostVE 的训练错误的上界。

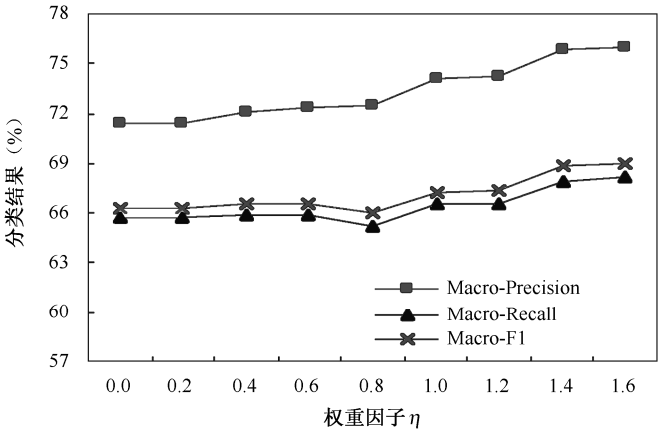


（a）Iteration 11（Rec）

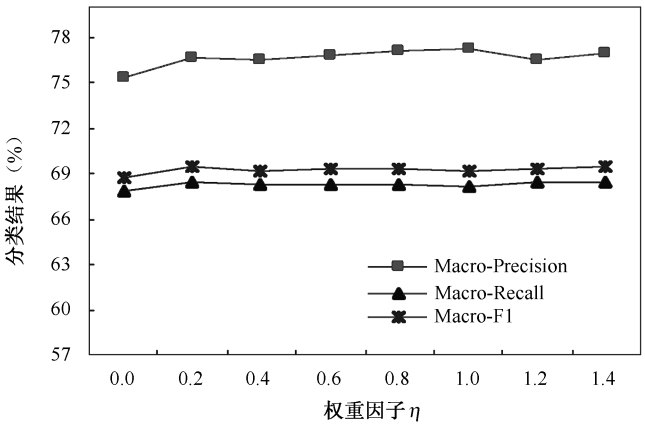


（b）Iteration 15（Rec）

图 6.1　参数 η 对 BoostVE 算法提升 NB 文本分类器的影响
（20-newsgroups 的子集 subset-Rec）



（a）Iteration 11(Sci)



（b）Iteration 15(Sci)

图 6.2　参数η对 BoostVE 算法提升 NB 文本分类器的影响
（20-newsgroups 的子集 subset-Sci）

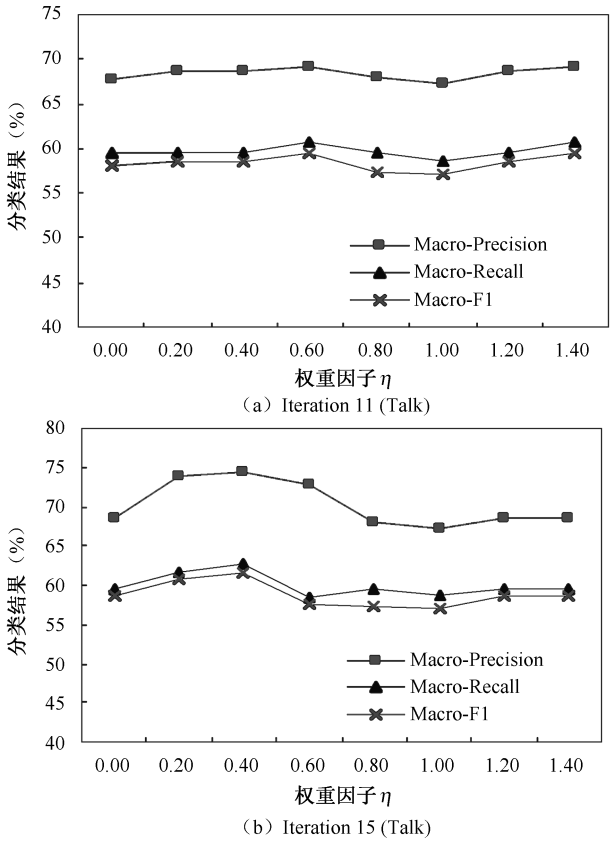


图 6.3　参数η 对 BoostVE 算法提升 NB 文本分类器的影响
（20-newsgroups 的子集 subset-Talk）

6.5.2　Boost VE 算法与 AdaBoost-MV 算法、AdaBoost 算法的实验比较

从图 6.4 和图 6.5 可以看出，在 subset-Rec 和 subset-Sci 上 BoostVE 算法明显优于 AdaBoost-MV 和 AdaBoost 算法，如图 6.6 所示，在 subset-Talk 上，经过 15～21 迭代，BoostVE 算法也取得了比其他两种算法好的结果。

1．特征多视图对 BoostVE 和 AdBoost-MV 的影响

观察图 6.4～图 6.6，AdaBoost 算法的结果明显比 BoostVE、AdaBoost-MV 算法差。例如，对比 AdaBoost，BoostVE 算法在 subset-Rec 数据集上迭代 33 次后，Macro-Precision 提高了 7.78%（见图 6.4（c）），Macro-Recal 提高了 12.16%（见图 6.4（b））；图 6.5 表明，采用数据集 subset-Sci，BoostVE 取得了明显优于 AdaBoost 的类似结果；采用数据集 subset-Talk，BoostVE 算法的结果也优于 AdaBoost，如图 6.6 所示。AdaBoost-MV 对比 AdaBoost，也取得了比较好结果。如图 6.4 所示，在 sunset-Rec 上，经过 33 次迭代，AdaBoost-MV 的 Macro-Precision、Macro- Recall、Macro-F1 分别提高了 3.64%、2.7%和 2.38%。如图 6.5 所示，在 subset-Sci 数据集上，AdaBoost-MV 也优于 AdaBoost。

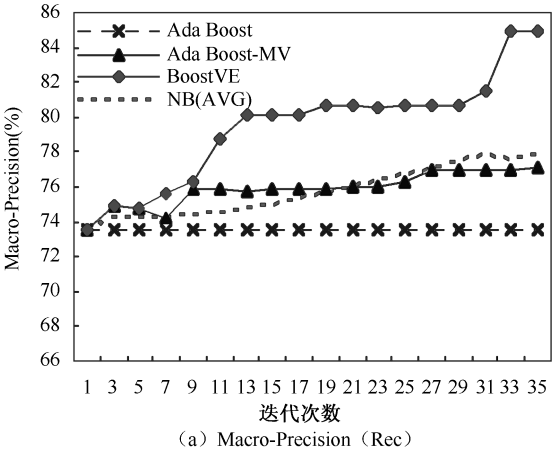
分析原因，AdaBoost 每次迭代使用的是同一个特征视图，而 BoostVE 和 AdaBoost-MV 采用的是特征多视图。Adaboost 的性能不能随着迭代次数而改变，这是因为 AdaBoost 算法每次迭代使用的是同一个特征视图，仅依据训练文本被分错时增加权重、否则减少权重，NB 基分类器仍然非常稳定，被分错的样本在下一次迭代时仍然分错，其权重增长很快，致使 ε_t 很快超过 0.5，AdaBoost 重新初始化每个文本的权重为 1/N，转向步骤（3.1），因此 AdaBoost 的性能在图 6.4～图 6.6 中没有随迭代次数而改变。而 BoostVE 和 AdaBoost-MV 通过选择不同的评估函数、TF/DF 公式、保留特征数创建不同的特征视图，从而训练产生不同的 NB 基分类器，多个有差异的 NB 基分类器通过投票组合能够纠正彼此的错误，提高集成的 NB 分类器的精度。

2．基于投票熵的 re-weighting 新策略对 BoostVE 的影响

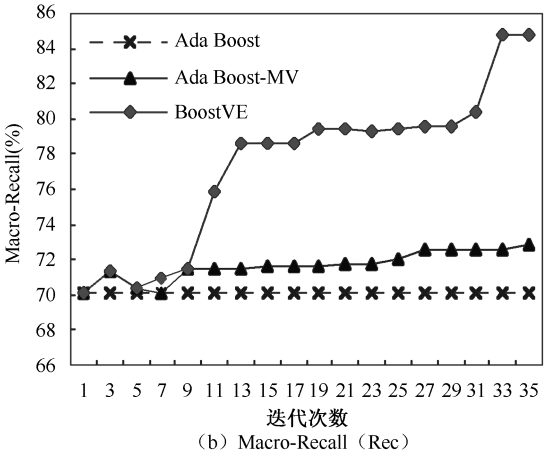
对比图 6.4～图 6.6 中 BoostVE 算法和 AdaBoost-MV 算法的运行结果，无论是 Macro-Precision、Macro-Recall 还是 Macro-F1 评价指标，BoostVE 算法的结果明显优于 AdaBoost-MV 算法。

例如，对比 AdaBoost-MV，在 subset-Rec 上，迭代 33 次后，BoostVE 的 Macro-Precision 提高了 11.45%，如图 6.4（a）所示；Macro-Recall 提高了 14.70%，如图 6.4（b）所示；Macro-F1 提高了 14.03%，如图 6.4（c）所

示。在 subset-Sci 上，如图 6.5 所示，BoostVE 也明显优于 AdaBoost-MV。在 subset-Talk 上，如图 6.6 所示，BoostVE 比 AdaBoost-MV 的改善更温和。



（a）Macro-Precision（Rec）



（b）Macro-Recall（Rec）

图 6.4　AdaBoost、AdaBoost-MV、BoostVE 对 NB 文本分类器的提升效果比较

（20-newsgroups 的子集 subset-Rec（BoostVE 中取 η=0.8））

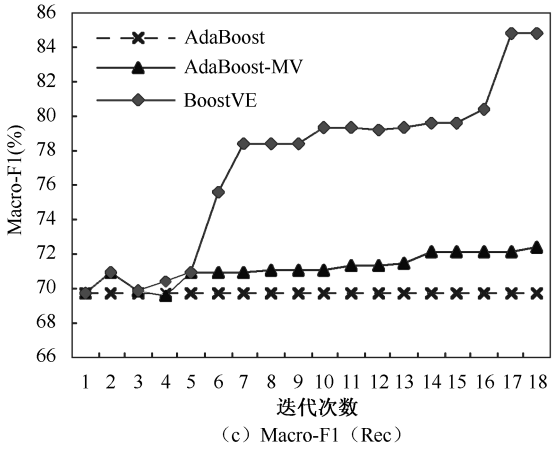


图 6.4　AdaBoost、AdaBoost-MV、BoostVE 对 NB 文本分类器的提升效果比较（20-newsgroups 的子集 subset-Rec（BoostVE 中取 η=0.8））（续）

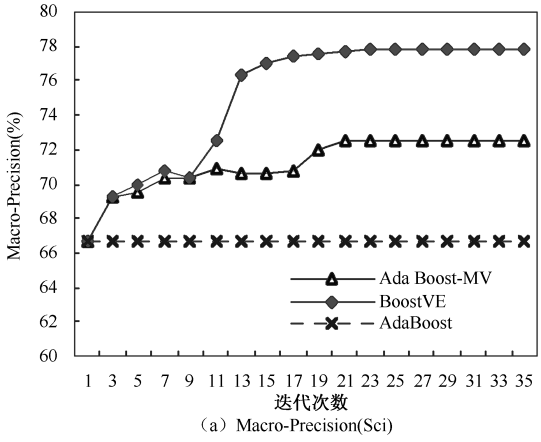
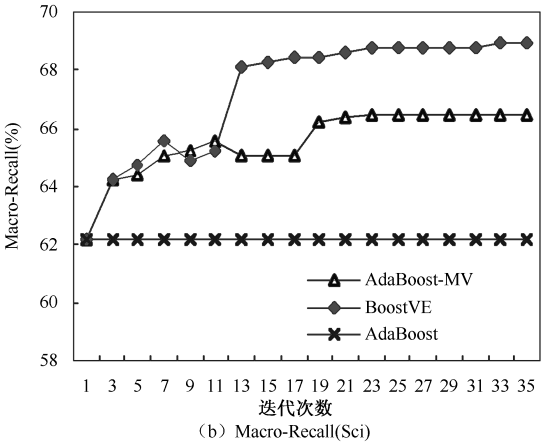
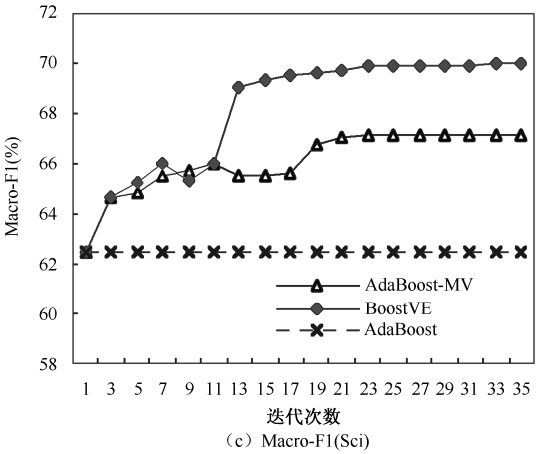


图 6.5　AdaBoost、AdaBoost-MV、BoostVE 对 NB 文本分类器的提升效果比较（20-newsgroups 的子集 subset-Sci（BoostVE 中取 η=0.8））

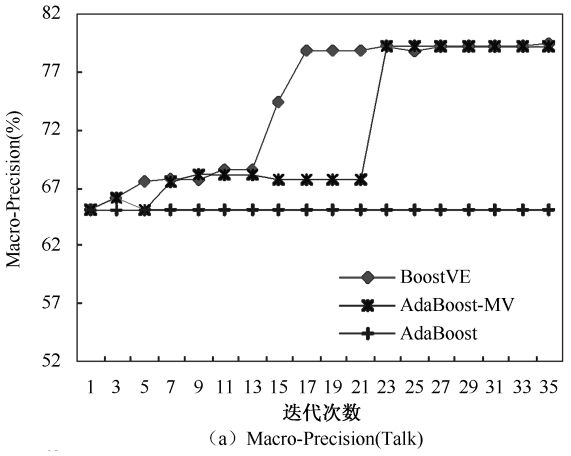


（b）Macro-Recall(Sci)

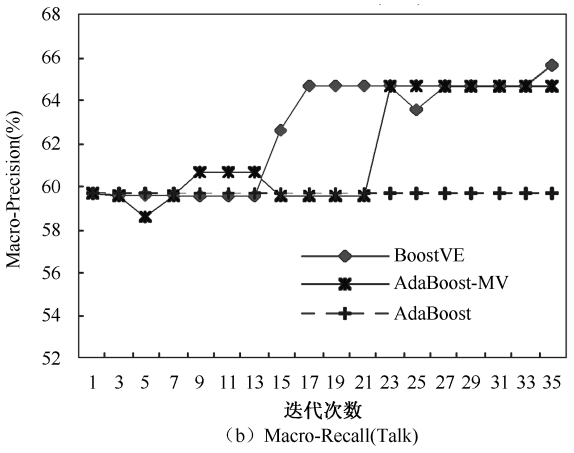


（c）Macro-F1(Sci)

图 6.5　AdaBoost、AdaBoost-MV、BoostVE 对 NB 文本分类器的提升效果比较（20-newsgroups 的子集 subset-Sci（BoostVE 中取 η=0.8））（续）

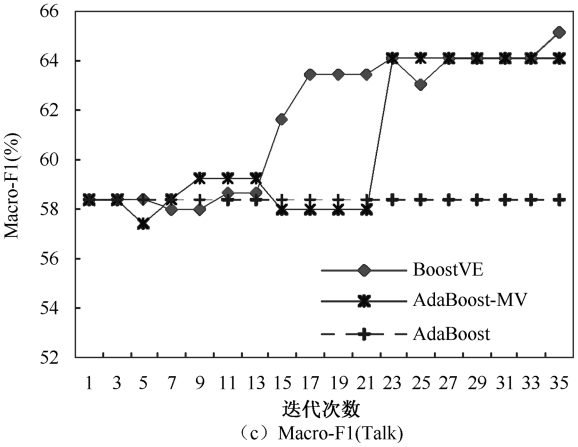


（a）Macro-Precision(Talk)



（b）Macro-Recall(Talk)

图 6.6　AdaBoost、AdaBoost-MV、BoostVE 对 NB 文本分类器的提升效果比较
（20-newsgroups 的子集 subset-Talk（BoostVE 中取 η=0.4））



（c）Macro-F1(Talk)

图 6.6　AdaBoost、AdaBoost-MV、BoostVE 对 NB 文本分类器的提升效果比较（20-newsgroups 的子集 subset-Talk（BoostVE 中取 η=0.4））（续）

BoostVE 和 AdaBoost-MV 的唯一区别是训练文本的权重更新策略不同，BoostVE 采用本章提出的基于投票熵的 re-weighting 策略，而 AdaBoost-MV 采用传统 re-weighting 策略，而且在这两种策略下，基分类器置信度的计算公式是不同的。

基于投票熵的 re-weighting 策略中，文本的权重不是简单地根据分类正确与否调整的，还要考虑前 t 个基分类器在每个训练文本的投票分歧 VE_{ti}。能够通过减少“最富信息”样本的分类错误，提高基分类器的正确性；利用投票分歧，增加基分类器间的差异性。新策略下基分类器的置信度 β_t 的计算同时考虑分类错误率 ε_t 和基分类器的平均投票熵 $\overline{\mathrm{VE}_t}$，在 ε_t 相同的情况下，那些能够增大基分类器间差异性的基分类器会获得更大的权重，从而提高 NB 分类器的泛化能力。这说明采用基于投票信息熵的样本权重调整新策略能够有效提升 NB 分类器的分类结果。

6.5.3　BoostVE 算法提升 NB 文本分类器的有效性

从图 6.4～图 6.6 可以看出，BoostVE 算法对 NB 文本分类器的提升效果明显优于 AdaBoost 和 AdBoost-MV 算法。特别应该注意图 6.4（a）中，虚

线（NB（AVG））代表在 t 个视图上构造的 t 个 NB 分类器的 Macro-Precision 指标的平均值，对比图 6.4（a）中的 NB（AVG）和 BoostVE 的轨迹，可以说明 BoostVE 算法能够有效地提升 NB 文本分类器。

表 6.5～表 6.7 分别描述的是数据集 subset-Rec、subset-Sci 和 subset-Talk 上，BoostVE 算法经过不同次数的迭代后，对 NB 分类器的提升结果。实验结果的评估不仅采用了 Macro-Precision、Macro-Recall 和 Macro-F1，而且包括每个类的 Precision、Recall 和 F1 评估指标。表 6.5～表 6.7 中粗体表示的数据揭示了集成的 NB 分类器的分类结果得到了 BoostVE 的有效提升。例如，如表 6.5，在数据集 subset-Rec 上，经过 33 次迭代后，rec.autos 类的 Recall 和 F1 分别提高到了 86.49%和 87.37%，尽管其 Precision 下降了；在数据集 subset-Sci 上，迭代 25 次后，类 sci.crypt 的 Precision、Recall 和 F1 分别提高了 23.41%、11.64% 和 15.79%，如表 6.6 所示。

分析 BoostVE 算法采用的基于投票熵的样本权重维护新策略，该策略下基分类器信任权重的计算方法都遵循集成分类器的原则，既考虑了通过减少“最富信息”样本的分类错误，提高基分类器的正确性，利用引入投票信息熵来增加 NB 基分类器间的差异性，因而能够有效提高 NB 分类器的泛化能力。

表 6.5　BoostVE 算法在 subset-Rec 子集上对 NB TC 的提升效果（η=0.8）

T	classname	Precision(%)	Recall(%)	F1 (%)	Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)
1	rec.autos	93.62	59.46	72.73	73.48	70.10	69.79
	rec.motorcycles	63.93	94.59	76.29			
	rec.sport.baseball	61.59	68.24	64.74			
	rec.sport.hockey	74.78	58.11	65.40			
15	rec.autos	93.58	**68.92**	**79.38**	**80.06**	**78.55**	**78.47**
	rec.motorcycles	**78.16**	91.89	**84.47**			
	rec.sport.baseball	**69.10**	**83.11**	**75.46**			
	rec.sport.hockey	**79.39**	**70.27**	**74.55**			
33	rec.autos	88.28	**86.49**	**87.37**	**84.93**	**84.80**	**84.82**
	rec.motorcycles	**88.44**	**87.84**	**88.14**			
	rec.sport.baseball	**78.62**	**84.46**	**81.43**			
	rec.sport.hockey	**84.40**	**80.41**	**82.35**			

表 6.6　BoostVE 算法在 subset-Sci 子集上对 NB TC 的提升效果(η=0.8)

T	classname	Precision(%)	Recall(%)	F1(%)	Macro-Precision (%)	Macro-Recal (%)	Macro-F1 (%)
1	sci.electronics	79.66	60.26	68.61	66.65	62.18	62.50
	sci.crypt	67.59	46.79	55.30			
	sci.med	46.77	78.85	58.71			
	sci.space	72.59	62.82	67.35			
11	sci.electronics	**92.86**	58.33	**71.65**	**72.52**	**65.22**	**65.98**
	sci.crypt	**75.44**	**55.13**	**63.70**			
	sci.med	**47.16**	**85.26**	**60.73**			
	sci.space	**74.62**	**62.18**	**67.83**			
25	sci.electronics	89.81	**62.18**	**73.48**	**77.81**	**68.75**	**69.87**
	sci.crypt	**91.00**	**58.33**	**71.09**			
	sci.med	**47.62**	**89.74**	**62.22**			
	sci.space	**82.79**	**64.74**	**72.66**			

表 6.7　BoostVE 算法在 subset-Talk 子集上对 NB TC 的提升效果(η=0.4)

T	classname	Precision (%)	Recall (%)	F1(%)	Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)
1	talk.politics.guns	76.92	31.25	44.44	65.17	59.66	58.33
	talk.politics.mideast	70.59	75.00	72.73			
	talk.politics.misc	48.00	72.73	57.83			
15	talk.politics.guns	83.33	31.25	45.45	**74.34**	**62.59**	**61.66**
	talk.politics.mideast	**91.30**	65.63	**76.36**			
	talk.politics.misc	**48.39**	**90.91**	**63.16**			
25	talk.politics.guns	**92.31**	**37.50**	**53.33**	**78.76**	**63.57**	**63.08**
	talk.politics.mideast	**94.74**	56.25	**70.59**			
	talk.politics.misc	**49.23**	**96.97**	**65.31**			

❑6.6　本章小结

本章提出了一种基于投票信息熵和特征多视图的 AdaBoost 改进算法——BoostVE 算法，用于提升 NB 文本分类器。除了利用特征视图构建多个有

差异性的 NB 基分类器外，重点提出了基于投票信息熵的样本权重调整策略。对比 AdaBoost，新策略不仅考虑样本是否被当前基分类器分错，还要考虑该样本在前几轮基分类器上的投票分歧。通过引入投票熵，减少最富信息样本的分类错误，提高基分类器的正确性。新策略下的基分类器置信度的计算不仅与分类错误率有关，而且与基分类器间的差异性有关。在分类错误率 ε_t 相同的情况下，对增大基分类器间的差异性贡献大的基分类器会获得更大的置信度。这样，正确性和差异性较高的 NB 基分类器能够被集成为泛化能力更好的强分类器。理论分析证明 BoostVE 算法的最小训练错误上界优于 AdaBoost 算法。在 20-newsgroups 上的对比实验表明 BoostVE 算法能够有效提高 NB 文本器的泛化能力。

BoostVE 算法引入了权重参数 η 以平衡正确性和差异性，参数 η 在实验中采用的是经验值，这是存在的不足。在后续的工作中，将根据训练错误率和平均投票熵动态调整 η 的取值，进一步改进和完善 BoostVE 算法。基于投票信息熵的样本权重调整策略在图像分类、图像识别等领域用来提升其他学习算法，也有潜在的应用价值。

第 7 章

结合半监督学习的 SemiBoost–CR 分类模型

半监督学习（semi-supervised learning）和集成学习（ensemble learning）作为机器学习（machine learning）的两大主流，在各自的领域已经取得了不少研究成果，但是也存在一些问题：① 结合少量 labeled 样本和大量 unlabeled 样本的半监督学习中，当迭代到一定的次数时，随着未标注样本的增加，分类精度不再提高或下降；② AdaBoost 由于过分追求分类器样本间隔（margin）的最大化，从而存在过学习（overfitting）问题，在训练样本数目少且存在噪声的情况下更是如此。另外，AdaBoost 算法对不稳定的学习算法（如决策树、神经网络等）的提升效果比较好[36-38][92, 93]，但对稳定的学习算法（如 Naïve Bayesian，NB）的提升效果不理想，有时甚至使其分类性能下降[97-99]。这些都是值得继续研究的问题。

最近的研究文献表明，结合半监督学习和集成学习的研究是一个有趣的方向和研究热点[109-116]。D'Alche-Buc F.和 Grandvalet Y.在文献[109]中提出了将 MarginBoost 算法应用在半监督图像分类中。Mallapragada 和 Rong Jin 在文献[112]中提出了 SemiBoost 算法，在 AdaBoost 算法中引入了未标注（unlabeled）样本，在给未标注样本分类时遵循两个主要标准：相似度高的未标注（unlabeled）样本一定属于相同的类别；未标注样本的类别一定与其相似度高的标注（labeled）样本的类别一致；并以未标注样本与标注样本之间

的不一致性、未标注样本之间的不一致性为依据定义目标函数[112]。

本章结合半监督学习和 Boosting 技术，基于置信度对未标注样本重取样，提出了一种 SemiBoost-CR（Semi-supervised Boosting based on Confidence Resampling）分类模型。每次迭代，按照置信度重采样，选取一定数量置信度比较高和置信度比较低的未标注样本，分别以不同的策略加入到已标注的训练样本集中，对训练集进行扰动，目的是生成多个有差异的基分类器。关于置信度的度量，提出了基于最大差距和 K 近邻（包括标注近邻样本和未标注近邻样本）的两种计算方法。实验表明 SemiBoost-CR 分类模型能够有效提升 NB 的分类性能。

❑7.1　SemiBoost-CR 模型的目标函数

令训练样本集 $D = D^l \bigcup D^u$，$D^l = \{(x_i^l, \boldsymbol{y}_i^l)\}_{i=1}^{N_l}$ 表示包含 N_l 个标注样本的集合，$D^u = \{x_i^u\}_{i=1}^{N_u}$ 表示包含 N_u 个未标注样本的集合。类别集合 $C = \{c_j\}_{j=1}^{L}$，L 为类别数。$\boldsymbol{y}_i^l = (y_i^1, \cdots, y_i^L) \in Y,\ Y = \{0,+1\}^L$，如果 x_i 属于第 j 个类别 c_j，那么 $y_i^j = +1$，否则 $y_i^j = 0$，且 $\sum_{j=1}^{L} y_i^j = 1$，即每个样本 x_i 只能属于唯一的一个类别。

每次迭代建立的基分类器 $h_t : D \to Y$，基分类器 h_t 对样本 x_i^l 和 x_i^u 的分类结果 $h_t(x_i^l)$ 和 $h_t(x_i^u)$，都对应一个 L 维向量 $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = (\hat{y}_i^1, \cdots, \hat{y}_i^L) \in Y$，$Y = \{0,+1\}^L$。令 $h_{ti}^j \equiv \hat{y}_i^j$，$j = 1, \cdots, L$，表示 h_t 把样本 x_i^l 或 x_i^u 预测为 c_j 类，同样 $\sum_{j=1}^{L} \hat{y}_i^j = 1$。

Boosting 方法每次迭代的目标是训练错误（training error）最小化[37, 38]。只使用标注样本集训练学习的有监督 AdaBoost 算法，因为每个训练样本 $x_i^l \in D^l$ 都有标注类别 y_i^l，所以其目标函数（或损失函数）比较好确定：

$$F_l = \sum_{i=1}^{N_l} \mathrm{e}^{(1-2u_{ii}^t)} \tag{7-1}$$

其中 $u_{ii}^t \equiv h_t(x_i^u)\boldsymbol{y}_{\boldsymbol{i}}^{\mathrm{T}} = \hat{\boldsymbol{y}}_{\boldsymbol{i}}\, \boldsymbol{y}_{\boldsymbol{i}}^{\mathrm{T}}$，如果第 t 次迭代的基分类器 h_t 对样本 x_i^l 分类正确，$\hat{\boldsymbol{y}}_{\boldsymbol{i}}\, \boldsymbol{y}_{\boldsymbol{i}}^{\mathrm{T}} = 1$，即 $u_{ii}^t = 1$，否则 $\hat{\boldsymbol{y}}_{\boldsymbol{i}}\, \boldsymbol{y}_{\boldsymbol{i}}^{\mathrm{T}} = 0$，即 $u_{ii}^t = 0$。

这里要构造的 SemiBoost-CR 分类模型，每次迭代需要按照某种策略选

取一定数量的未标注样本加入标注训练集 D^l，训练生成新的基分类器。由于未标注样本 x_i^u 没有已知的标注类别，计算它的损失时就没有评判依据。已有的一种解决方法是，认为 unlabeled 样本总是分对的，即没有损失；另一种方法是，以它最大可能的标注类别为依据判断对错。受文献[112]的启发，给未标注样本分类时必须遵循以下两个主要标准：①相似度高的未标注样本一定属于相同的类别；②未标注样本的类别一定与其相似度高的标注样本的类别一致[112]。文献[115, 116]在半监督图像分类中也遵循了类似的标准。这样，未标注样本的损失函数就由两部分组成。

① 未标注样本与相似的标注样本之间分类的不一致性，定义如下：

$$F_{lu}=\sum_{i=1}^{N_u}\sum_{j=1}^{K}S(x_i^u,x_j^l)\mathrm{e}^{(1-2u_{ij}^t)} \tag{7-2}$$

② 相似的未标注样本之间的分类的不一致性，定义如下：

$$F_{uu}=\sum_{i=1}^{N_u}\sum_{j=1}^{K}S(x_i^u,x_j^u)\mathrm{e}^{(1-2v_{ij}^t)} \tag{7-3}$$

其中，$u_{ij}^t\equiv h_t(x_i^u)\boldsymbol{y}_j^{\mathrm{T}}=\hat{\boldsymbol{y}}_i\,\boldsymbol{y}_j^{\mathrm{T}}$ 表示未标注样本 x_i^u 与标注样本 x_j^l 的分类结果是否一致。如果 x_i^u 与 x_j^l 的分类结果一致，$\hat{\boldsymbol{y}}_i\,\boldsymbol{y}_j^{\mathrm{T}}=1$ 即 $u_{ij}^t=1$；否则，$\hat{\boldsymbol{y}}_i\,\boldsymbol{y}_j^{\mathrm{T}}=0$ 即 $u_{ij}^t=0$。类似地，$v_{ij}^t\equiv h_t(x_i^u)h_t(x_j^u)^{\mathrm{T}}=\hat{\boldsymbol{y}}_i\,\hat{\boldsymbol{y}}_j^{\mathrm{T}}$ 表示两个未标注样本 x_i^u 与 x_j^u 的分类结果是否一致。如果 x_i^u 与 x_j^u 的分类结果一致，$\hat{\boldsymbol{y}}_i\,\hat{\boldsymbol{y}}_j^{\mathrm{T}}=1$ 即 $v_{ij}^t=1$；否则，$\hat{\boldsymbol{y}}_i\,\hat{\boldsymbol{y}}_j^{\mathrm{T}}=0$ 即 $v_{ij}^t=0$。$S(x_i,x_j)$ 指样本 x_i 和 x_j 的相似度。文本 x_i 和 x_j 的相似度可以通过计算两个文本向量之间的距离来度量，如欧式距离、余弦距离或内积，这里采用夹角余弦公式[327]。

$$S(x_i,x_j)=\cos(x_i,x_j)=\frac{\sum_{k=1}^{|V|}ws_{ik}ws_{jk}}{\sqrt{\sum_{k=1}^{|V|}ws_{ik}^2\sum_{k=1}^{|V|}ws_{jk}^2}} \tag{7-4}$$

由式（7-2）和式（7-3），未标注样本的损失计算如下：

$$\begin{aligned}F_u&=F_{lu}+\gamma F_{uu}\\&=\sum_{i=1}^{N_u}\sum_{j=1}^{K}S(x_i^u,x_j^l)\mathrm{e}^{(1-2u_{ij}^t)}+\gamma\sum_{i=1}^{N_u}\sum_{j=1}^{K}S(x_i^u,x_j^u)\mathrm{e}^{(1-2v_{ij}^t)}\end{aligned} \tag{7-5}$$

其中，$\gamma \in [0,1]$，调节与未标注样本 x_i^u 相似的标注样本和未标注样本在计算 x_i^u 的训练错误时的作用。

这样，由式（7-1）和式（7-5），结合未标注样本训练的 SemiBoost-CR 分类模型的目标函数定义如下：

$$F = F_l + CF_u$$
$$= \sum_{i=1}^{N_s} e^{(1-2u_{ii}^l)} + C\left(\sum_{i=1}^{N_u}\sum_{j=1}^{K} S(x_i^u, x_j^l) e^{(1-2u_{ij}^l)} + \gamma \sum_{i=1}^{N_u}\sum_{j=1}^{K} S(x_i^u, x_j^u) e^{(1-2v_{ij}^l)}\right) \quad (7\text{-}6)$$

其中，F_l 表示标注样本的训练错误；F_u 表示未标注样本的训练错误；C 是常数，调节标注样本和未标注样本在目标函数中的作用。

❑7.2　未标注样本的置信度

结合半监督学习的 SemiBoost-CR 分类模型，每次迭代需要选取一定数量的未标注加入标注训练集 D^l，以便训练生成新的有差异的基分类器。通常选择置信度高的未标注样本，这就涉及置信度的度量问题。假设 $P(c_r \mid x_i^u)$ 表示未标注样本 x_i^u 被基分类器 h_t 分为类 c_r 的后验概率，半监督学习中已有的方法是：按照 $P(c_r \mid x_i^u)$ 大小排序，$P(c_r \mid x_i^u)$ 大的样本，其置信度就越高。这种度量有一定的道理，但是未必是最合理、最好的度量方法。本节提出两种新的置信度计算方法。

7.2.1　基于 K 近邻的置信度

如 7.1 节讨论目标函数时所述，给未标注样本分类时必须遵循以下两个主要标准：①相似度高的未标注样本一定属于相同的类别；②未标注样本的类别一定与其相似度高的标注样本的类别一致[112]。因此，可以通过计算未标注样本 x_i^u 与它最相似的 K 个标注近邻的样本的分类一致性，以及 x_i^u 与它最相似的 K 个未标注近邻的样本的分类一致性，来度量 x_i^u 的置信度。

定义 7.1　令 $S(x_i^u, x_j^l)$ 表示未标注样本 x_i^u 与它的标注近邻 x_j^l 的相似度，$S(x_i^u, x_j^u)$ 表示未标注样本 x_i^u 与它的未标注近邻 x_j^u 的相似度，样本 x_i^u 被基分

类器 h_t 标记为某个类 c_r 的置信度，由 x_i^u 与它的 K 个标注近邻的一致性 A_i^r 及 x_i^u 与它的 K 个未标注近邻的一致性 B_i^r 决定，用 $\mathrm{conf}_1(x_i^u)$ 表示，计算如下：

$$\mathrm{conf}_1(x_i^u) = \lambda A_i^r + (1-\lambda) B_i^r \tag{7-7}$$

$$A_i^r = \sum_{j=1}^{K} S(x_i^u, x_j^l) \hat{y}_i^r \, y_j^r \tag{7-8}$$

$$B_i^r = \sum_{j=1}^{K} S(x_i^u, x_j^u) \hat{y}_i^r \, \hat{y}_j^r \tag{7-9}$$

其中，如果 h_t 对 x_i^u 与 x_j^l 分类一致，那么 $\hat{y}_i^r \, y_j^r = 1$，否则 $\hat{y}_i^r \, y_j^r = 0$。同样道理，如果 h_t 对 x_i^u 与 x_j^u 的分类一致，那么 $\hat{y}_i^r \, \hat{y}_j^r = 1$，否则 $\hat{y}_i^r \, \hat{y}_j^r = 0$。$A_i^r$ 可以看成将 x_i^u 分为 c_r 类时 K 个标注近邻给出的置信度，而 B_i^r 则表示 K 个未标注近邻给出的置信度。$\lambda \in [0,1]$，调节 A_i^r 和 B_i^r 在 $\mathrm{conf}_1(x_i^u)$ 中的权重，当 $\lambda = 1$ 时，表示只看重 K 个标注近邻给出的置信度 A_i^r。

由式（7-7）、式（7-8）和式（7-9）得

$$\mathrm{conf}_1(x_i^u) = \lambda \sum_{j=1}^{K} S(x_i^u, x_j^l) \hat{y}_i^r \, y_j^r + (1-\lambda) \sum_{j=1}^{K} S(x_i^u, x_j^u) \hat{y}_i^r \, \hat{y}_j^r \tag{7-10}$$

$\mathrm{conf}_1(x_i^u)$ 越大，表明样本 x_i^u 与最相似的 K 个已标注、未标注近邻的分类结果越一致，x_i^u 的置信度越高；反之，$\mathrm{conf}_1(x_i^u)$ 越小，x_i^u 与它最相似的 K 个已标注、未标注近邻的分类结果越不一致，也就是说，x_i^u 的置信度越低。

这里提出的 $\mathrm{conf}_1(x_i^u)$ 与文献[112]中的置信度计算方法类似，但有所不同。① 文献[112]讨论的是两类问题，这里讨论的是多类问题；② 本书的相似度计算方法与文献[112]不同，本书只选取最相似的 K 个近邻，而不是如文献[112]那样选择所有的样本，减少了计算量；③ 文献[112]只关心置信度高的样本，本书对置信度较低的样本也加以利用（将在 7.3 节讨论）。实验中，K 取值为 15、25 和 35。

7.2.2 基于最大差距的置信度

从直观上分析，在两类问题中，如果某个未标注样本 x_i^u 被基分类器 h_t 分

为正类的后验概率 $P(+1|x_i^u)$ 或负类的后验概率 $P(-1|x_i^u)$ 越接近 0.5，说明基分类器 h_t 对 x_i^u 的类别标注越“不自信”；相反，如果 $P(+1|x_i^u)$ 和 $P(-1|x_i^u)$ 的差越大，则说明 h_t 对 x_i^u 的类别标注越“自信”。在多类问题中（假设 L 个类别），如果 $P(c_j|x_i^u)$ 越接近 $1/L$，说明基分类器 h_t 对 x_i^u 的类别标注越“不自信”；而 $P(c_j|x_i^u)-P(\overline{c_j}|x_i^u)$ 的值越大，即 x_i^u 被标注为 c_j 类和非 c_j 类的距离越大，说明 h_t 对 x_i^u 的类别标注越“自信”。由此提出基于最大差距的置信度定义。

定义7.2　令 $P(c_j|x_i^u)$ 表示样本 x_i^u 被基分类器 h_t 分为类 c_j 的后验概率，且 $\sum_{j=1}^{L}P(c_j|x_i^u)=1$。如果未标注样本 x_i^u 被基分类器 h_t 分为 c_r 类，其置信度由 x_i^u 被分为 c_r 类的后验概率 $P(c_r|x_i^u)$ 与其他非 c_r 类的平均后验概率的距离度量，用 $\mathrm{conf}_2(x_i^u)$ 表示，计算如下：

$$\mathrm{conf}_2(x_i^u)=P(c_r|x_i^u)-\frac{1}{L-1}\sum_{c_{r'}\neq c_r}P(c_{r'}|x_i^u) \tag{7-11}$$

$\mathrm{conf}_2(x_i^u)$ 越大，表明样本 x_i^u 被 h_t 越“自信”地分为 c_r 类；反之，$\mathrm{conf}_2(x_i^u)$ 越小，表明 h_t 对样本 x_i^u 的分类结果越“不自信”，或者说，x_i^u 越不确定（uncertain）。

与 $\mathrm{conf}_1(x_i^u)$ 比较，$\mathrm{conf}_2(x_i^u)$ 比较简单，不需要相似矩阵 $\boldsymbol{S}$，降低了时间复杂度。

❑7.3　基于置信度的重取样策略

每次迭代，按照定义 7.1 或定义 7.2 计算出 h_t 对每个未标注样本 x_i^u 分类的置信度，由高到低排序后，如何基于置信度重取样呢？在半监督学习中，基于置信度重取样的策略通常是：选择一定数量置信度高的未标注样本加入到标注样本集 D^l。文献[112]中每次迭代只选择 10%的置信度高的样本。那么置信度低的样本有没有利用价值呢？

无论是 Boosting 系列算法还是其他集成分类器，在选择基分类器时，除

了要考虑基分类器的分类正确性外，还要考虑基分类器之间的差异。只有存在一定的差异，基分类器之间才能够纠正彼此的分类错误，提高最终分类器的分类精度[78][81-83][93]。

在“主动学习”（active learning），也称为“基于查询的学习”中，通过重采样技术提高分类器的性能，也就是选择“最富信息”的样本作为“查询项”提交给一个“神谕”（oracle）——即一个永远无错的标注样本的教师[117]。“最富信息”的样本连同根据“神谕”提供的类别标注添加到训练样本集，训练产生新的分类器。在“基于置信度的查询选择方法”中，分类器计算样本属于每一类的判别函数 $g_r(x), r = c_1, \cdots, c_l$，“最富信息”的样本就是属于每个类的判别函数值基本相等的样本。受此启发，Boosting 每次迭代时，根据未标注样本的置信度重采样，不仅应该选择置信度最高未标注样本，而置信度最低的样本往往就是“最富信息”的样本，可以把它提交给“神谕”——这里选择交互方式下的“用户”，由“用户”反馈一个恰当的类别。

因此，SemiBoost-CR 分类模型中，每次迭代，不仅选取置信度较高的top*N*%未标注样本，还选择置信度最低的 bottom*N*%样本未标注，但是按照不同的策略添加到标注样本集 D^l 中。这样不仅提高了基分类器的正确性，而且增加了基分类器间的差异性。

① 置信度较高的 top*N*%样本：选择 top*N*%未标注样本 x_i^u 连同它们的预测类别 $\hat{\boldsymbol{y}}_i$ 一起，即将 $(x_i^u, \hat{\boldsymbol{y}}_i)$ 添加到已标注训练集 D^l 中，同时从 D^u 中删除。

② 置信度较低的 bottom*N*%样本：

bottom*N*%样本的置信度较低，由 7.2 节的定义 7.1 和定义 7.2 可知，置信度低说明其不确定性较高，即“富有信息”（more informative），它们的类别可由“神谕”给出。“神谕”可以是交互界面下的“用户”，即由用户给出一个类别，或者由该样本的 K 个相似近邻给出一个伪类标记（pseudo-label）。

定义 7.3 令 $S(x_i^u, x_j^l)$ 表示未标注样本 x_i^u 和标注样本 x_j^l 的相似度，$S(x_i^u, x_j^u)$ 表示未标注样本 x_i^u 和未标注样本 x_j^u 的相似度，x_i^u 的伪类标记（pseudo-label）由与它最相似的 K 个标注样本和 K 个未标注样本给出，用 $\hat{\boldsymbol{y}}_i^*$ 表示：

$$\hat{\boldsymbol{y}}_i^* = \sum_{j=1}^{K} S(x_i^u, x_j^l)\boldsymbol{y}_j + \gamma \sum_{j=1}^{K} S(x_i^u, x_j^u)\hat{\boldsymbol{y}}_j, \quad \gamma \in [0,1] \qquad (7\text{-}12)$$

$$\hat{\boldsymbol{y}}_i^{*}=(\hat{y}_i^{1*},\cdots,\hat{y}_i^{r*},\cdots,\hat{y}_i^{L*})\in\{0,+1\}^L$$

$$\hat{y}_i^{r*}=\begin{cases}1, & r^*=\arg\max_{r'}\ \{\sum_{r'}^{L}(\sum_{j=1}^{K}S(x_i^u,x_j^l)y_i^{r'}+\gamma\sum_{j=1}^{K}S(x_i^u,x_j^u)\hat{y}_i^{r'})\}\\ 0, & \text{其他}\end{cases}$$

即 x_i^u 的伪类标记 $\hat{\boldsymbol{y}}_i^{*}$ 由与 x_i^u 最相似的 K 个标记样本的真实类别和 K 个未标记样本的分类结果决定。$\gamma\in[0,1]$，调节与 x_i^u 相似的未标注样本在决定 $\hat{\boldsymbol{y}}_i^{*}$ 时的作用。对于置信度较低的 bottom$N$%的每个样本 x_i^u，将 $(x_i^u,\hat{\boldsymbol{y}}_i^{*})$ 添加到 D^l 中，并从 D^u 中删除 x_i^u。

7.6 节的实验表明，交互界面下的“用户”作为“神谕”时，能给出较为准确的类别标注。伪类标记由 K 个标注近邻和 K 个未标注近邻给出，不需要人工干预，但是准确性相对于“用户”标注的较低。实验中，top$N$%和 bottom$N$%分别选取（5%，0%）、（2.5%，2.5%）、（5%，0%）或（5%，5%），用来验证基于置信度选择不同比例的未标注样本对集成分类器的分类性能的影响。

❑7.4　样本权重维护策略

1．标注样本 x_i^l 的权重

第t次迭代结束时，计算标注样本 x_i^l 在第t+1次迭代的权重 $w_{t+1}(x_i^l)$ 如下：

$$w_i^{t+1}(x_i^l)=w_i^t(x_i^l)\mathrm{e}^{(1-2u_{ii}^t)\alpha_t}/Z_t \qquad (7\text{-}13)$$

其中，$u_{ii}^t\equiv h_t(x_i)\boldsymbol{y}_i^{\mathrm{T}}$，如果 x_i^l 被分对，$u_{ii}^t=1$，否则为 $u_{ii}^t=0$。$\varepsilon_t=\sum_{i=1}^{N_l}w_t(x_i^l)\mathrm{e}^{(1-2u_{ii}^t)}$ 为加权错误率，文献[112]中，为最小化目标函数 F 的上界，推导出 $\alpha_t=\frac{1}{4}\ln((1-\varepsilon_t)/\varepsilon_t)$。SemiBoost-CR 分类模型中采用 $\alpha_t=\frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$。

2．未标注样本 x_i^u 的权重

按照 7.2 节基于置信度重取样机制，被选中的前 top$N$%或后 bottom$N$%的未标注样本，权重按式（7-14）计算，取所有已标注样本的权重的平均值。

$$w_{t+1}(x_i^u)=\frac{1}{N_l}\sum_{j=1}^{N_l}w_{t+1}(x_j^l)/Z_t \tag{7-14}$$

式（7-13）和式（7-14）中的 Z_t 是归一化因子，为了满足 $\sum_{i=1}^{N_l}w_{t+1}(x_i^l)+\sum_{i=1}^{N_u}w_{t+1}(x_i^u)=1$。在下一次迭代中，已经加入到 D^l 的样本都按标注样本对待，按照式（7-13）更新权重。

❑7.5 SemiBoost-CR 分类算法

结合半监督学习和 Boosting 技术，根据 7.1～7.4 节所提出的目标函数、置信度计算方法、基于置信度的重取样策略及样本权重的维护策略，下面将给出 SemiBoost-CR（Semi-supervised Boosting based on Confidence Re-sampling）分类算法描述。对比 AdaBoost 算法，其不同之处在于步骤（3.1*）、（3.3*）～（3.5*）、（3.8*）～（3.9*）和（4*）。

表 7.1 SemiBoost-CR 算法

1. Input：训练集 $D=D^l\cup D^u$, $D^l=\{(x_i^l,\boldsymbol{y}_i^l)\}_{i=1}^{N_l}$, $D^u=\{x_i^u\}_{i=1}^{N_u}$, $\boldsymbol{y}_i^l=(y_i^1,\cdots,y_i^L)\in Y$, $Y=\{0,+1\}^L$, $C=\{c_j\}_{j=1}^L$,如果 $\boldsymbol{x}_i^l$ 属于第 j 个类别 c_j，那么 $y_i^j=+1$，否则 $y_i^j=0$，且 $\sum_{j=1}^L y_i^j=1$。

近邻数 K, 迭代次数 T, top$N$, bottom$N$, 基分类算法 ϕ_{NB}。

2. Initialize：为每个样本赋予相等的权重：$w_i^1=1/N$。

3. For $t=1$ **to** T **Do**

（3.1*）分别计算 D^u 中每个样本 x_i^u 与 D^l 中每个样本的相似度 $S(x_i^u,x_j^l)$，与 D^u 中每个样本的相似度 $S(x_i^u,x_j^u)$。

（3.2）在训练文本集 D^l 和权重分布 W^t 上，训练生成新的基分类器 ϕ_{NB}，即 h_t。

（3.3*）利用 ϕ_{NB} 对 D^l 中每个标注训练样本 x_i^l 中分类，输出：

$$h_t(x_i^l)=\hat{\boldsymbol{y}}_i=(\hat{y}_1,\cdots,\hat{y}_L)\in\{0,+1\}^L,\quad \hat{y}_i^j=\begin{cases}1, & j=\arg\max_{j'}\{\phi_{NB}(x_i^l)\}\\ 0, & \text{其他}\end{cases}$$

（3.4*）用 ϕ_{NB} 对 D^u 中每个未标注样本 $\boldsymbol{x}_i^u$ 分类，输出：

$$h_t(x_i^u)=\hat{\boldsymbol{y}}_i=(\hat{y}_1,\cdots,\hat{y}_L)\in\{0,+1\}^L,\quad \hat{y}_i^j=\begin{cases}1, & j=\arg\max_{j'}\{\phi_{NB}(x_i^u)\}\\ 0, & \text{其他}\end{cases}$$

（3.5*）计算每个 D^u 中每个未标注样本 x_i^u 的置信度 $\text{conf}(x_i^u)$，如式（7-10）或式（7-11），并降序排列。

（3.6）计算基分类器 h_t 的错误率：$\varepsilon_t=\sum_{i=1}^{N_l}w_t(x_i^l)\mathrm{e}^{(1-2u_{it}^l)}$。

续表

if（ε_t>0.5）置每个文本的权重为 1/N，转向（3.2*）。

（3.7）计算 $\alpha_t = \ln((1-\varepsilon_t)/\varepsilon_t)/2$ 。

（3.8*）选择 D^u 中置信度较高的 top$N$%样本 x_i^u，将 $(x_i^u, \hat{y}_i)$ 添加到 D^l 中，并从 D^u 中删除。

选择置信度较低的 bottom$N$%样本 x_i^u，计算其伪类标记 $\hat{y}_i^*$，将 $(x_i^u, \hat{y}_i^*)$ 添加到 D^l 中，并从 D^u 中删除。

（3.9*）更新原来 D^l 中的样本权重：$w_i^{t+1}(x_i^l) = w_i^t(x_i^l)\mathrm{e}^{(1-2u_{ti}^l)\alpha_t} / Z_t$。

更新 D^l 中新加入的样本权重：$w_{t+1}(x_i^u) = \frac{1}{N_l}\sum_{j=1}^{N_l} w_{t+1}(x_j^l)/Z_t$ 。

4*. Output: 最终分类器 $H(x_i) = \sum_{t=1}^{T}\alpha_t h_t(x_i)$，$x_i$ 被决策为 $c^* = \arg\max_j\{\sum_t^T \alpha_t h_{ti}^j\}$

令 $h_t : D \to Y$ 表示第 t 次迭代训练产生的基分类器，另 $H(x): D \to \mathbb{R}$ 表示经过 T 次迭代后集成的分类器，计算如下：

$$H(x_i) = \sum_{t=1}^{T}\alpha_t h_t(x_i) \tag{7-15}$$

其中 α_t 是 h_t 的信任权重，集成分类器对待分类样 x_i 的分类，输出 $\hat{\boldsymbol{y}}_i = (\hat{y}_i^1, \cdots, \hat{y}_i^L) \in \mathbb{R}^L$，$H_i^j \equiv \hat{y}_i^j = \sum_t^T \alpha_t h_{ti}^j$，$j = 1, \cdots, L$。则 x_i 将被标注为类 $c^* = \arg\max_j\{\sum_t^T \alpha_t h_{ti}^j\}$。

这里选择 Naïve Bayesian（NB）算法作为 SemiBoost-CR 分类模型的基分类器。需要说明的是，SemiBoost-CR 分类模型不仅可以提升 NB 分类器，也可以用来提升其他学习算法，而且其应用也不只限于文本分类，也可以用于图像分类，只需对如样本的表示、相似度计算加以修改就可以实现。

❑7.6　实验结果及其分析

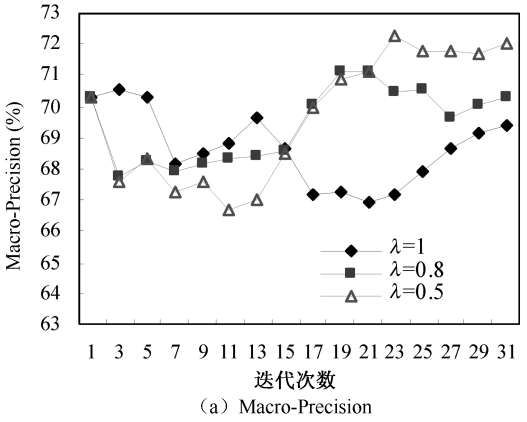
为了验证所提 SemiBoost-CR 算法对 NB 文本分类算法提升的有效性，在通用数据集 20-newsgroups dataset[①]上验证了 SemiBoost-CR 算法对 NB 文本分类器的提升效果，包括：使用式（7-10）的 conf_1 置信度计算方法的 SemiBoost-CR 算法；使用式（7-11）的 conf_2 置信度计算方法的 SemiBoost-CR 算法。

① http://people.csail.mit.edu/jrennie/20Newsgroups/

表 7.2　20-news newsgroups 数据子集

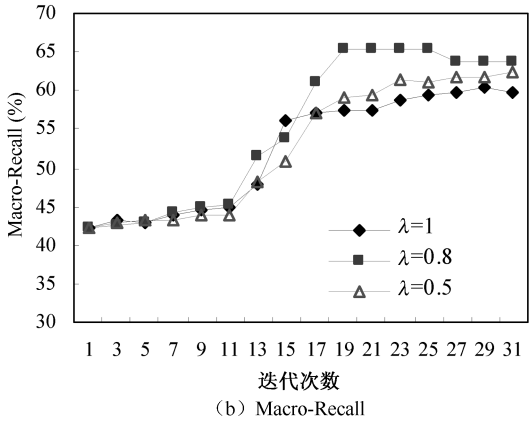
数 据 子 集	类　　别	标注样本数	未标注样本数	测试样本数
subset-Rec（*L*60+*U*500+*T*592）	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	60（15 per class）	592	592（148 per class）
subset-Talk（*L*54+ *U*375+*T*96）	talk.politics.misc talk.politics.guns talk.politics.mideast	54（18 per class）	375	96（32 per class）
subset-Sci（*L*100+*U*500+*T*624）	sci.crypt sci.electronics sci.med sci.space	100（25 per class）	500	624（156 per class）

因为原始的 20-newsgroups 数据集规模比较庞大，这里选取了其中的三个子集，详细信息如表 7.2 所示。下面的叙述中，令 *Ln* 表示包含 *n* 篇文本的训练子集，*Tm* 表示包含 *m* 篇文本的测试子集。实验结果如图 7.1～图 7.12 所示，评价采用 Macro-Precision、Macro-Recall、Macro-F1。

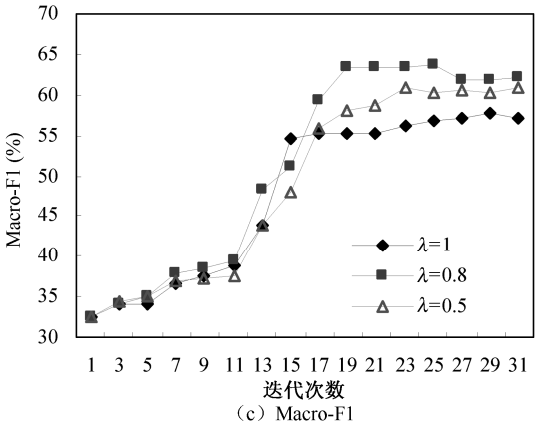


（a）Macro-Precision

图 7.1　置信度 conf_1 中 λ 取不同值时 SemiBoost-CR 的分类结果
（subset-Sci: *L*100 +*U*500+*T*624,top*N*=2.5%,bottom*N*=2.5%,*K*=15）



（b）Macro-Recall



（c）Macro-F1

图 7.1　置信度 conf_1 中 λ 取不同值时 SemiBoost-CR 的分类结果

（subset-Sci: *L*100 +*U*500+*T*624,top*N*=2.5%,bottom*N*=2.5%,*K*=15）（续）

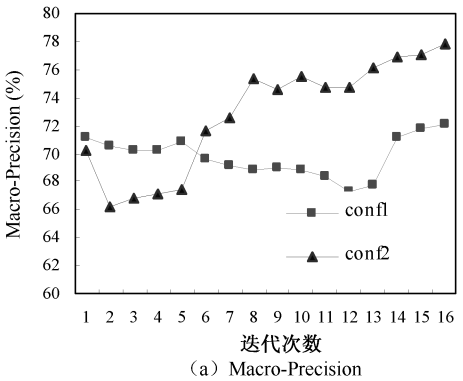
7.6.1　未标注近邻样本对置信度 conf_1 的影响

第 7.2 节所提出的基于 K 近邻的置信度 conf_1，计算公式如式（7-10），其中，$\lambda \in [0,1]$，调节 A_i^r 和 B_i^r 在 $\text{conf}_1(x_i^u)$ 中的权重，即标注近邻和未标注近邻样本在计算 $\text{conf}_1(x_i^u)$ 时的作用。当 $\lambda = 1$ 时，只有 K 个标注近邻给出的置信度 A_i^r。λ 的取值表示 x_i^u 的相似标注近邻和未标注近邻样本在计算 x_i^u 置

信度时的作用。如图 7.1（b）和图 7.1（c）所示，每次迭代按置信度排序选择前 2.5%和后 2.5%的未标注样本的加入 D^l，从指标 Macro-Recall 和 Macro-F1 看，λ=0.8 最好，其次是 λ=0.5，最后是 λ=1。图 7.1（a）所示的 Macro-Precision 指标表明，λ=0.5 最好，其次是 λ=0.8，最后是 λ=1。这说明在采用基于近邻的置信度计算方法中，未标注近邻样本对置信度的计算是有价值的。

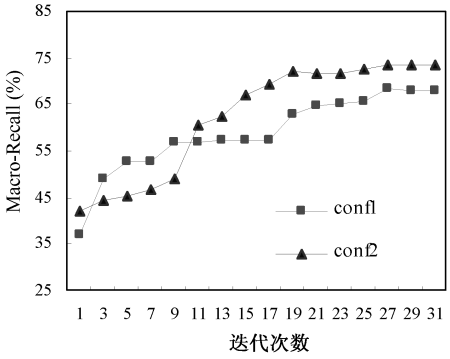
7.6.2 两种置信度方法 $conf_1$和 $conf_2$的实验比较

第 7.2 节提出了两种置信度计算方法：基于 K 近邻的 $conf_1$（见式（6-10））和基于最大差距的 $conf_2$（见式（6-11））。图 7.2～图 7.4 是在 20-newsgroups 的三个数据子集上，对基于 K 近邻的 $conf_1$ 和基于最大差距的 $conf_2$ 两种置信度计算方法的实验比较。

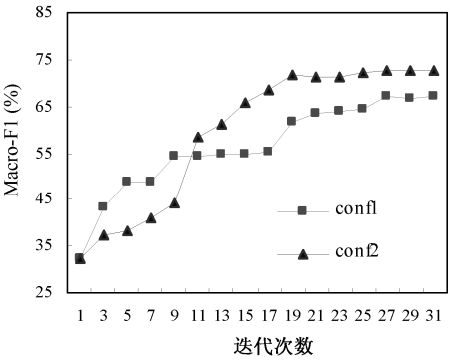


（a）Macro-Precision

图 7.2　两种置信度方法 $conf_1$ 和 $conf_2$ 下 SemiBoost-CR 分类结果比较
（subset-Sci :$L$100+$U$500+$T$624，top$N$=5%,bottom$N$=5%, λ=0.5, K=15）

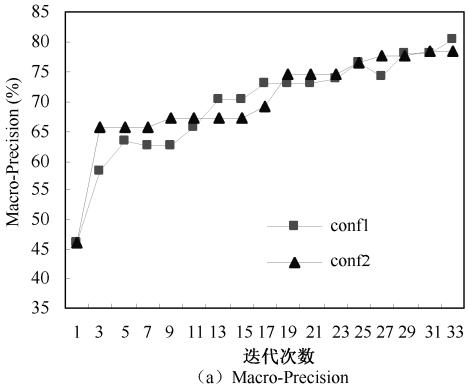


（b）Macro-Recall



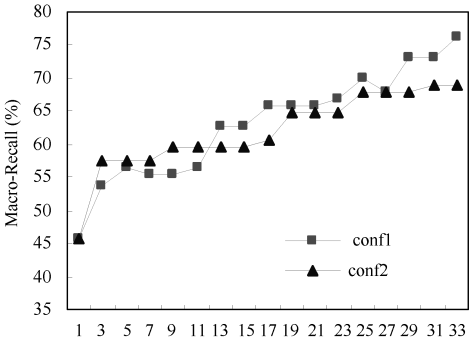
（c）Macro-F1

图 7.2　两种置信度方法 $conf_1$ 和 $conf_2$ 下 SemiBoost-CR 分类结果比较（subset-Sci :$L$100+$U$500+$T$624，top$N$=5%,bottom$N$=5%, λ=0.5, K=15）（续）

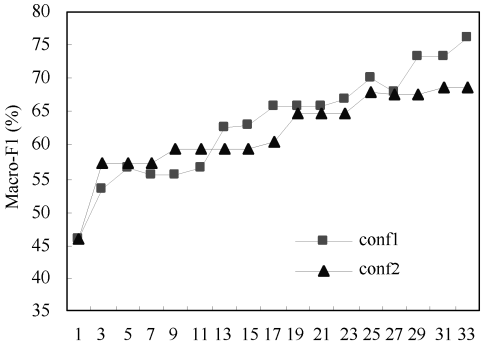


（a）Macro-Precision

图 7.3　两种置信度方法 $conf_1$ 和 $conf_2$ 下 SemiBoost-CR 分类结果比较（subset-Talk $L$54+ $U$375+ $T$96, top$N$=2.5%,bottom$N$=2.5%, K=15, λ=0.5）



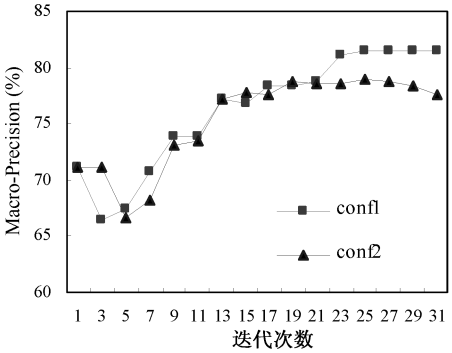
（b）Macro-Recall



（c）Macro-F1

图 7.3　两种置信度方法 $conf_1$ 和 $conf_2$ 下 SemiBoost-CR 分类结果比较

（subset-Talk $L$54+ $U$375+ $T$96, top$N$=2.5%,bottom$N$=2.5%, K=15, λ=0.5）（续）



（a）Macro-Precision

图 7.4　两种置信度方法 $conf_1$ 和 $conf_2$ 下 SemiBoost-CR 分类结果比较

（subset-Rec: $L$60+$U$500+$T$592, top$N$=5%,bottom$N$=5%, K=15, λ=0.5）

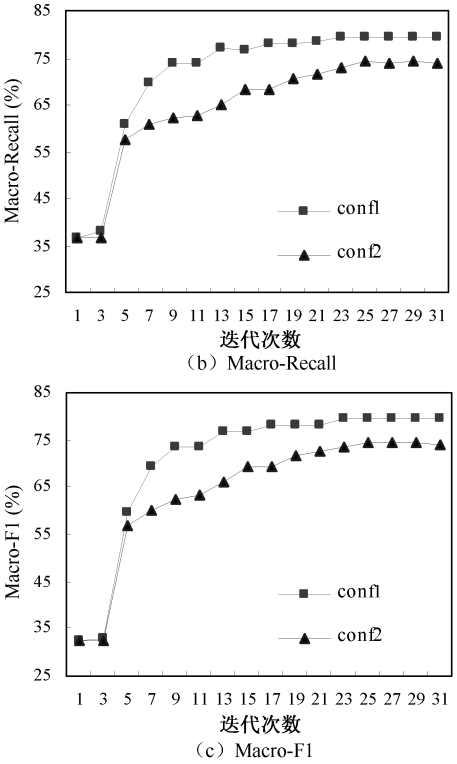


图 7.4　两种置信度方法 $conf_1$ 和 $conf_2$ 下 SemiBoost-CR 分类结果比较
（subset-Rec: *L*60+*U*500+*T*592, top*N*=5%,bottom*N*=5%, *K*=15, λ=0.5）（续）

从图 7.2～图 7.4 中可以看出，随着迭代次数的增加，按照 $conf_1$ 和 $conf_2$ 两种方法计算置信度，选择 top*N*%和 bottom*N*%的未标注样本加入标注样本集，继续训练生成新的基分类器，对比只使用少量的标注样本，SemiBoost-CR 分类结果的 Macro-Precision、Macro-Recall 和 Macro-F1 都得到了显著的提高。例如图 7.4 所示的，初始只使用 60 个标注样本时，NB 分类器的 Macro-F1 是 32.28%，随着迭代的进行，按照 $conf_1$ 或 $conf_2$ 两种方法计算置信度，选择前 5%置信度高的和后 5%置信度低的未标注样本，迭代到 21 次时，NB 的 Macro-F1 分别提升到 78.19%和 72.48%。

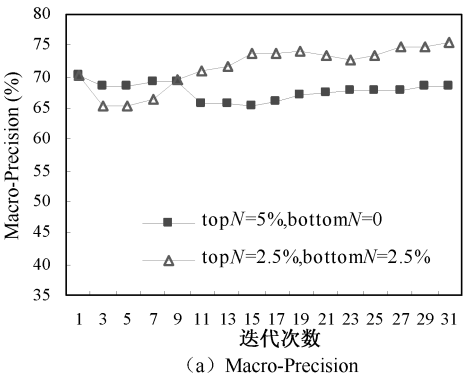
比较 $conf_1$ 和 $conf_2$，在 subset-Sci（*L*100+ *U*500+*T*624）上，top*N*=5%，bottom*N*=5%，λ=0.5，*K*=15 时（如图 7.2 所示），$conf_2$ 的效果要优于 $conf_1$。

在 subset-Talk（$L$54+ $U$375+ $T$96 上），top$N$=2.5%，bottom$N$=2.5%，K=15，λ=0.5 时（如图 7.3 所示），subset-Rec（$L$60 +$U$500+$T$592）上，top$N$=5%，bottom$N$=5%，K=15，λ=0.5 时（如图 7.4 所示），$conf_1$ 的效果要优于 $conf_2$。其他实验的比较结果类似，即 $conf_1$ 和 $conf_2$ 比较，不能说哪一种方法总是比另一种强，这与具体的数据集和参数选择有关。但有一点是肯定的，就是 $conf_1$ 和 $conf_2$ 两种计算方法都能比较准确地度量未标注样本的置信度，对 SemiBoost-CR 分类模型是有效的。

7.6.3 top$N$和 bottom$N$对 SemiBoost-CR 模型的影响

1．bottom$N$=0 与 bottom$N$≠0 对 SemiBoost-CR 模型的影响

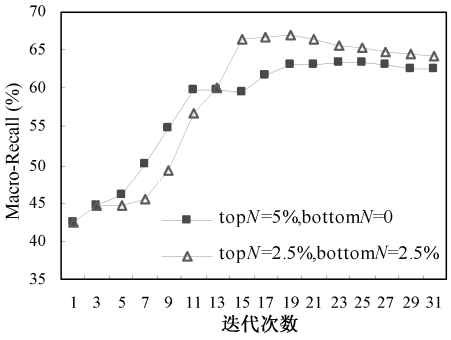
bottom$N$=0 表示不选择置信度较低的未标注样本，只选择置信度较高的 top$N$%未标注样本。bottom$N$≠0 表示不仅选择置信度高的 top$N$%未标注样本，而且选择置信度较低的 bottom$N$%的未标注样本。两种情况下的实验比较如图 7.5～图 7.7 所示。



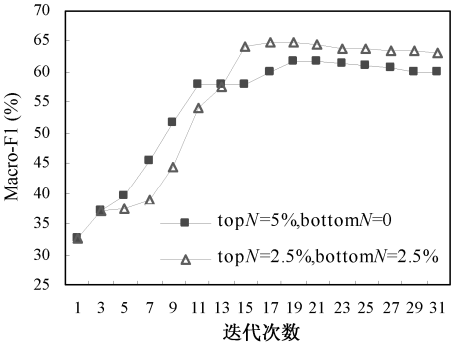
（a）Macro-Precision

图 7.5　bottom$N$=2.5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较

（subset-Sci：$L$100+$U$500+$T$624, $conf_1$：K=25, λ=1）



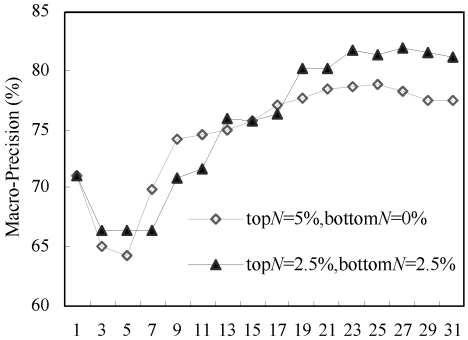
（b）Macro-Recall



（c）Macro-F1

图 7.5　bottom$N$=2.5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较

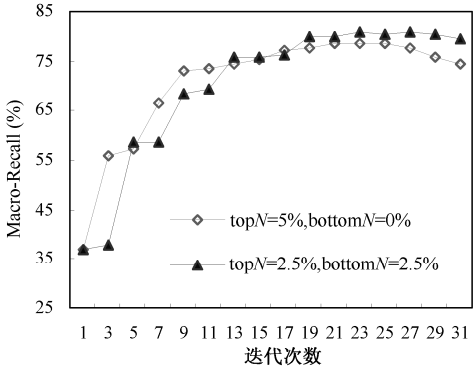
（subset-Sci：$L$100+$U$500+$T$624, conf$_1$：K=25, λ=1）（续）



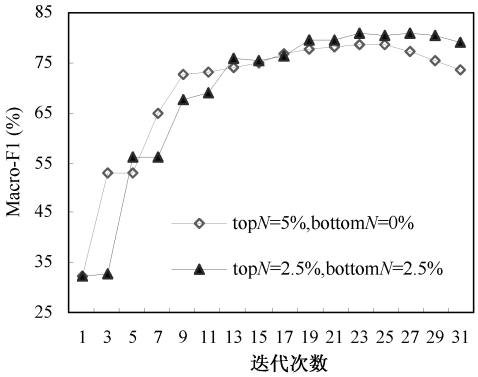
（a）Macro-Precision

图 7.6　bottom$N$=2.5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较

（subset-Rec: $L$60 +$U$500+$T$592,conf$_1$:K=15, λ=0.5）

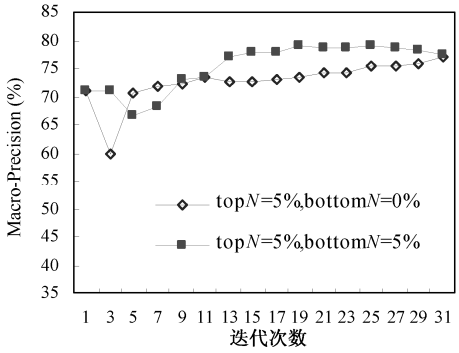


（b）Macro-Recall



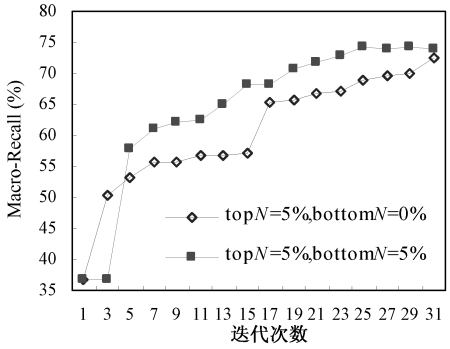
（c）Macro-F1

图 7.6　bottom$N$=2.5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较（subset-Rec: $L$60 +$U$500+$T$592, conf_1 :K=15, λ=0.5）（续）

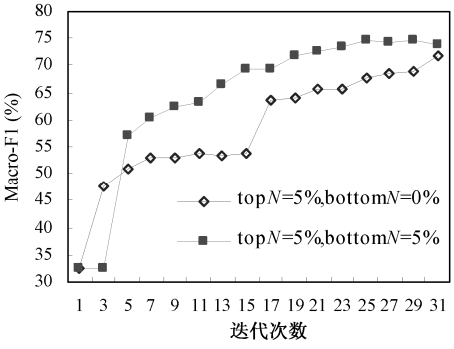


（a）Macro-Precision

图 7.7　bottom$N$=5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较（subset-Rec: $L$60+$U$500+$T$592, conf_1：K=25, λ=1）

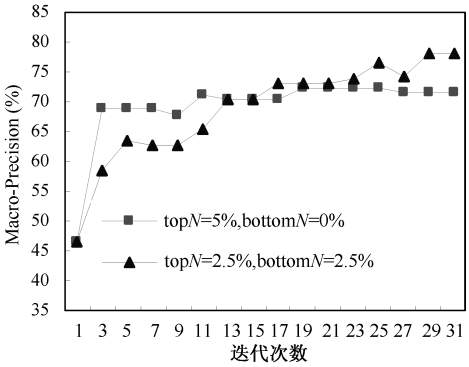


（b）Macro-Recall



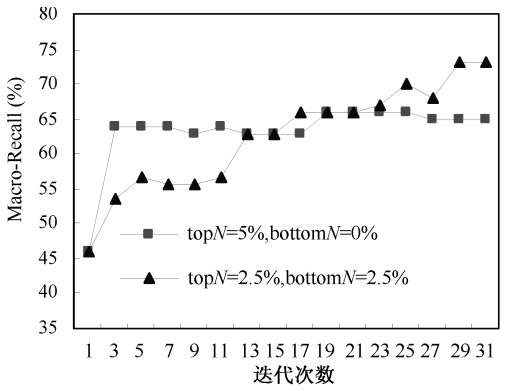
（c）Macro-F1

图 7.7　bottom$N$=5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较
（subset-Rec: $L$60+$U$500+$T$592, conf_1：K=25, λ=1）（续）

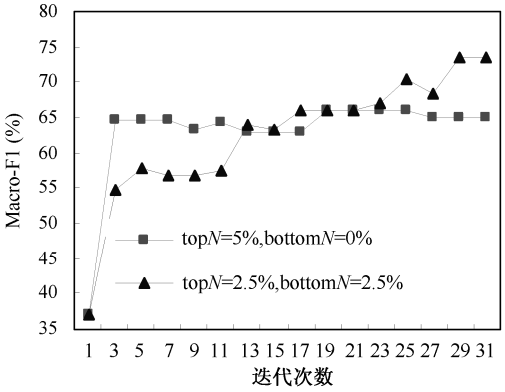


（a）Macro-Precision

图 7.8　bottom$N$=2.5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较
（subset-Talk: $L$54+$U$375+$T$96, conf_1：K=15, λ=0.5）



（b）Macro-Recall



（c）Macro-F1

图 7.8　bottom$N$=2.5%与 bottom$N$=0 的 SemiBoost-CR 分类结果比较
（subset-Talk: $L$54+$U$375+$T$96, conf_1：K=15, λ=0.5）（续）

从图 7.5～图 7.8 可以看出，除了选择置信度高的未标注样本，还选择一部分置信度较低的未标注样本，交给“神谕”——交互状态下的用户，由用户给出一个比较恰当的类别标注，加入标注样本集，这种策略比只选择置信度高的未标注样本要好。如图 7.5 所示，在 subset-Sci（$L$100+$U$500+$T$624）数据集上，每次迭代选择前 2.5%和后 2.5%未标注样本与只选择前 5%的未标注样本的 SemiBoost-CR 分类结果比较，迭代 9 次后，前者的 Macro-Precision 就超过了后者；迭代 13 次后，前者的 Macro-Recall 和 Macro-F1 也超过了后者。如图 7.6 所示，在 subset-Rec（$L$60 +$U$500+$T$592）上，每次迭代选择前 2.5%和后 2.5%与只选择前 5%的未标注样本的 SemiBoost-CR 分类结果比较，

迭代 17 次以后，前者比后者（即 bottom$N$≠0 比 bottom$N$=0）的效果要好。图 7.7 和图 7.8 也表现出类似的比较结果。

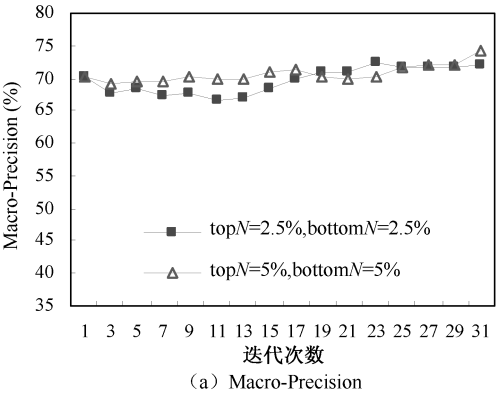
分析基于 K 近邻的置信度计算方法 conf_1，$\text{conf}_1(x_i^u)$ 越小，说明 x_i^u 与它最相似的 K 个已标注、未标注近邻的分类结果越不一致，也就是说，x_i^u 的置信度越低，x_i^u 越“富有信息”，由用户给出恰当的类别标注加入标注样本集，有助于提高基分类器的正确性和基分类器间的差异性。因此，每次迭代，除了选择置信度高的未标注样本外，同时选择一部分置信度较低的未标注样本，由用户类别标注后加入标注样本集，这种策略比只选择置信度高的未标注样本要好。

2．不同的 top$N$、bottom$N$ 对 SemiBoost-CR 分类模型的影响

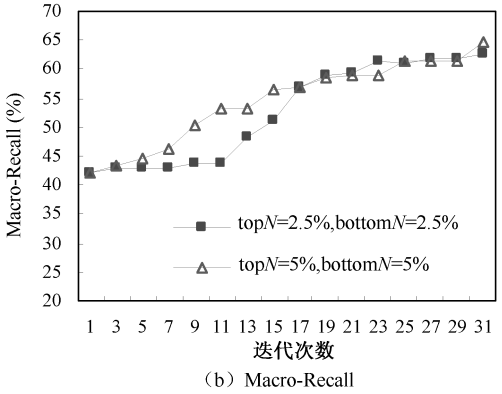
图 7.9～图 7.12 展示了在 20-newsgroups 标准数据集的三个子集上，选择不同比例的 top$N$%和 bottom$N$%的未标注样本加入标注训练集，对 SemiBoost-CR 分类模型提升 NB 文本分类器的影响。

如图 7.11 所示，在 subset-Rec（$L$60+$U$500+$T$592）上，置信度 conf_1，K=15，λ=0.5，当迭代次数小于 19 时，top$N$=5%、bottom$N$=5%的情况比不上 top$N$=2.5%、bottom$N$=2.5%的情况，但是随着迭代次数增加，top$N$=5%、bottom$N$=5%的情况优于 top$N$=2.5%，bottom$N$=2.5%的情况。如图 7.9 所示，在 subset-Rec（$L$60+$U$500+$T$592）上，置信度 conf_1，K=25，λ=1 时比较结果类似。从图 7.10 可以看出，在 subset-Talk（$L$54+$U$375+$T$96）上，置信度 conf_1，K=25，λ=1 时，top$N$=5%、bottom$N$=5%明显优于 top$N$=2.5%，bottom$N$=2.5%。如图 7.12 所示，在 subset-Rec（$L$60+$U$500+$T$592）上，置信度 conf_1，K=25，λ=1 时，top$N$=5%、bottom$N$=5%优于 top$N$=2.5%、bottom$N$=2.5%，top$N$=2.5%、bottom$N$=2.5%优于 top$N$=5%、bottom$N$=0。实验表明，通常选择 top$N$=5%、bottom$N$=5%比较合适。

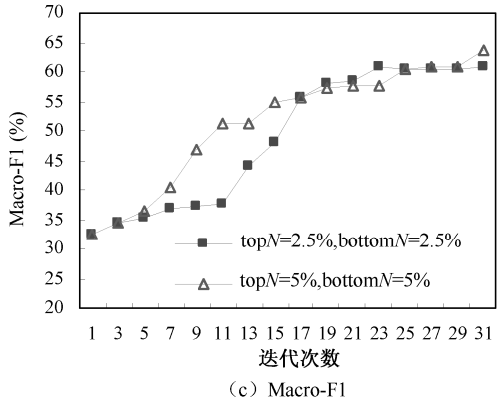
上述对比实验表明，每次迭代按照置信度排序，选择置信度高的未标注样本加入，目的是提高下一次迭代的基分类器的正确性。选择置信度低的未标注样本加入，目的是增加基分类器间的差异性。因为无论是基于 K 近邻（conf_1）还是基于最大差距 conf_2 的置信度计算方法，置信度越低，表示其“不确定性”越大，越“富有信息”。因此，SemiBoost-CR 分类模型能够有效地提升基分类器——NB 文本分类器的性能。



（a）Macro-Precision



（b）Macro-Recall



（c）Macro-F1

图 7.9　选择不同比例的 unlabeled 样本的 SemiBoost-CR 分类结果比较

（subset-Sci: $L$100+$U$500+$T$624, conf_1: K=25, λ=1）

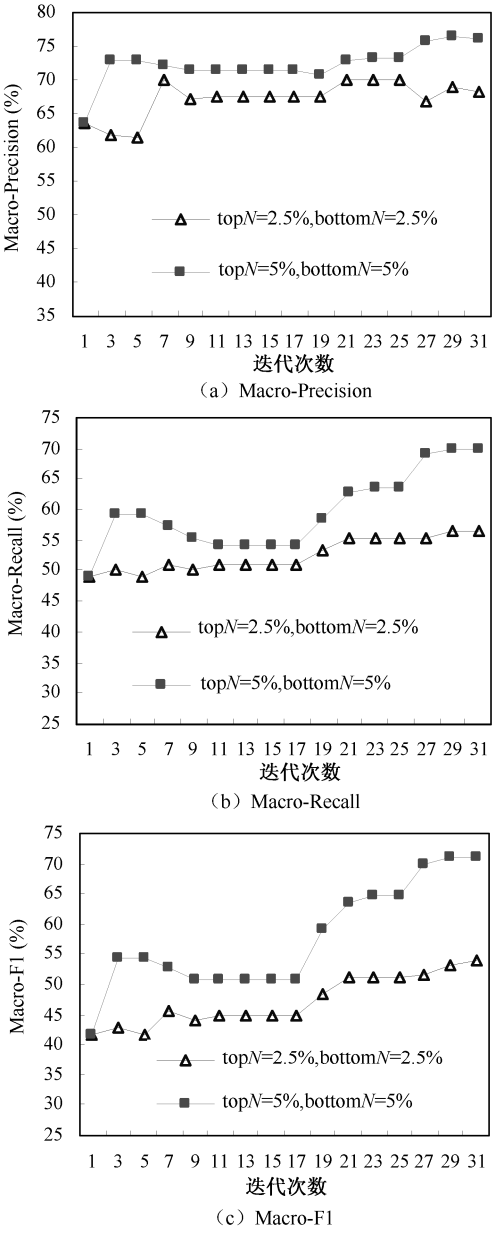
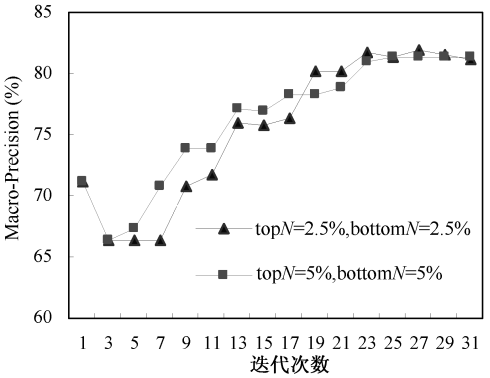
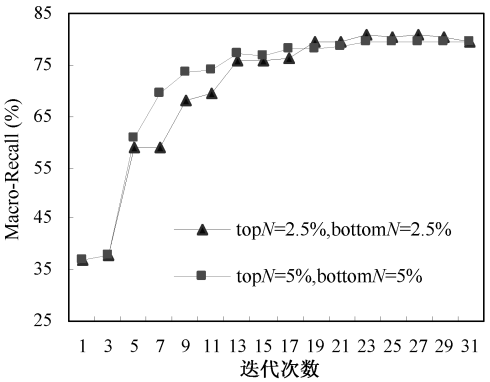


图 7.10　选择不同比例的 unlabeled 样本的 SemiBoost-CR 分类结果比较

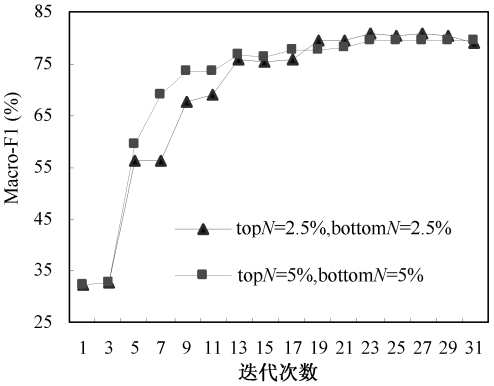
（subset-Talk: $L54+U375+T96$, conf_1: $K=25$, $\lambda=1$）



（a）Macro-Precision



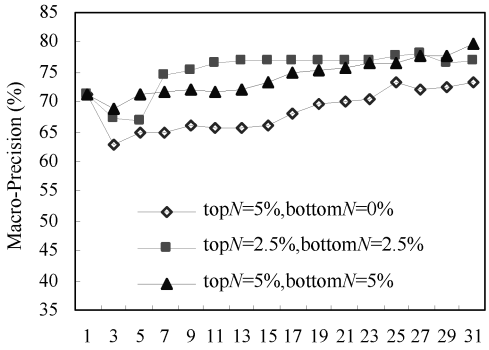
（b）Macro-Recall



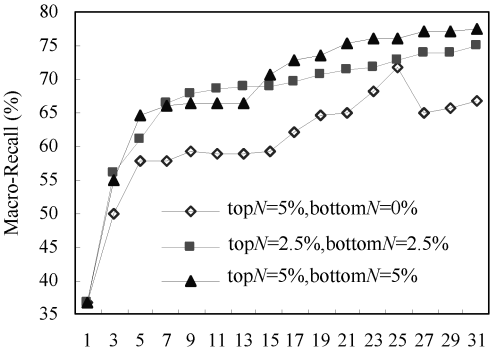
（c）Macro-F1

图 7.11　选择不同比例的 unlabeled 样本的 SemiBoost-CR 分类结果比较

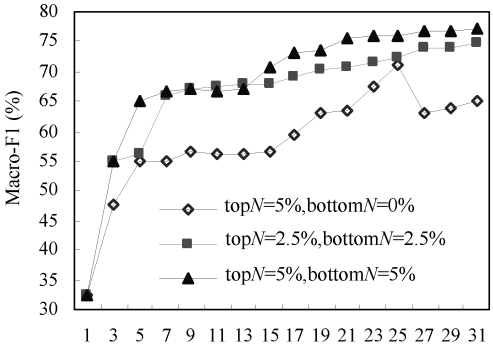
（subset- Rec: $L60+U500+T592$, conf_1：$K=15$，$\lambda=0.5$）



（a）Macro-Precision



（b）Macro-Recall



（c）Macro-F1

图 7.12　选择不同比例的 unlabeled 样本的 SemiBoost-CR 分类结果比较

（subset- Rec: $L60+U500+T592$, conf_1：$K=25$, $\lambda=1$）

❑7.7 本章小结

结合半监督学习和集成学习方法，提出了一种基于置信度重取样的SemiBoost-CR 分类模型。提出了基于最大差距和基于 K 近邻的两种置信度计算公式，按照置信度重采样，选取一定比例置信度较高和置信度较低的未标注样本，分别以不同的策略加入到已标注的训练样本集。对比实验表明，使用少量的标注样本和大量的未标注样本，SemiBoost-CR 分类模型能够有效提升 NB 文本分类器的性能。需要改进的是根据目标函数进一步优化 α_t 的取值。

第8章

文本自动分类系统 SECTCS

❑8.1 系统简介

中英文文本分类系统 SECTCS（the Smart English and Chinese Text Categorization System）是前期使用 VC++ 6.0 开发的一个实验平台，原有系统针对的是有监督分类方法的研究，只实现了质心分类、Naïve Bayesian 分类、K 近邻分类和支持向量机（SVM）等有监督的分类算法。通过对国内外有关半监督学习和集成学习的研究，对原来的 SECTCS 系统进行了修改和功能扩展，进一步开发实现了半监督分类方法和集成分类方法中的经典算法，以及前几章提出的基于半监督学习与集成学习的各种改进算法。

1．SECTCS 系统原有的功能

① 训练文本的处理。实现文本的切词、英文 stemming 处理、词频统计、停用词过滤、转换成初始文本向量、各主题类的初始向量、各特征词的向量，生成语料表和类别表。

② 特征提取和权值调整。系统中实现了使用文档频数（Document Frequency）、IDF、信息增益（Information Gain）、期望交叉熵（Expected cross Entropy）、互信息（Mutual Information）、文本证据权（the Weight of Evidence for Text）、几率比（Odds Ratios）、χ^2 统计（CHI）8 种评估函数进行特征权值调整，权值计算基于词频型、文档型两种公式。

③ 实现了多种有监督学习的分类模型。包括基于向量空间模型的质心

分类、基于实例的 K 近邻（KNN）分类、基于概率的朴素贝叶斯（Naïve Bayesian，NB）分类和支持向量机（SVM）。

④ 分类结果评价。采用 Precision、Recall、F1 三种评估方法对分类结果进行评价。

2．SECTCS 系统新扩展的功能

① 修改了训练文本处理部分，增加了对未标注文本的处理。

主要修改了与文本向量、各主题类向量、各特征词的向量、语料表和类别表等相关的数据结构，以适应半监督分类算法和集成分类算法的处理。

② 设计实现了基于半监督学习的分类模型，包括：

- 经典的 Co-training 分类算法；
- 基于随机分割特征子集的 Co-Rnd 算法（也称 SC-PART-Rnd 算法）；
- 基于 TEF-WA 技术对 Co-training 的改进算法：TV-SC 算法和 TV-DC 算法；
- 基于特征独立模型（MID-model）的 SC-PMID-MI 算法和 SC-PMID-CHI 算法；
- EM 算法、EM-λ算法。

③ 设计实现了集成分类模型，包括：

- 经典的 AdaBoost 算法；
- 基于特征多视图的 AdaBoost-MV 算法；
- 基于投票信息熵和多视图的 BoostVE 算法。

④ 结合半监督学习与集成学习，设计实现了基于置信度重取样的 SemiBoost-CR 分类模型。

⑤ 扩充了分类结果评价功能：增加了 Macro-Precision、Macro-Recall、Macro-F1、Micro-F1 四种评估方法，从多个角度对分类结果进行评价，给出分类错误信息，并输出到指定的文本文件中，以便综合比较和分析各种分类模型的性能优劣。

❑8.2 系统总体结构

文本自动分类系统/SECTCS 的总体结构如图 8-1 所示。

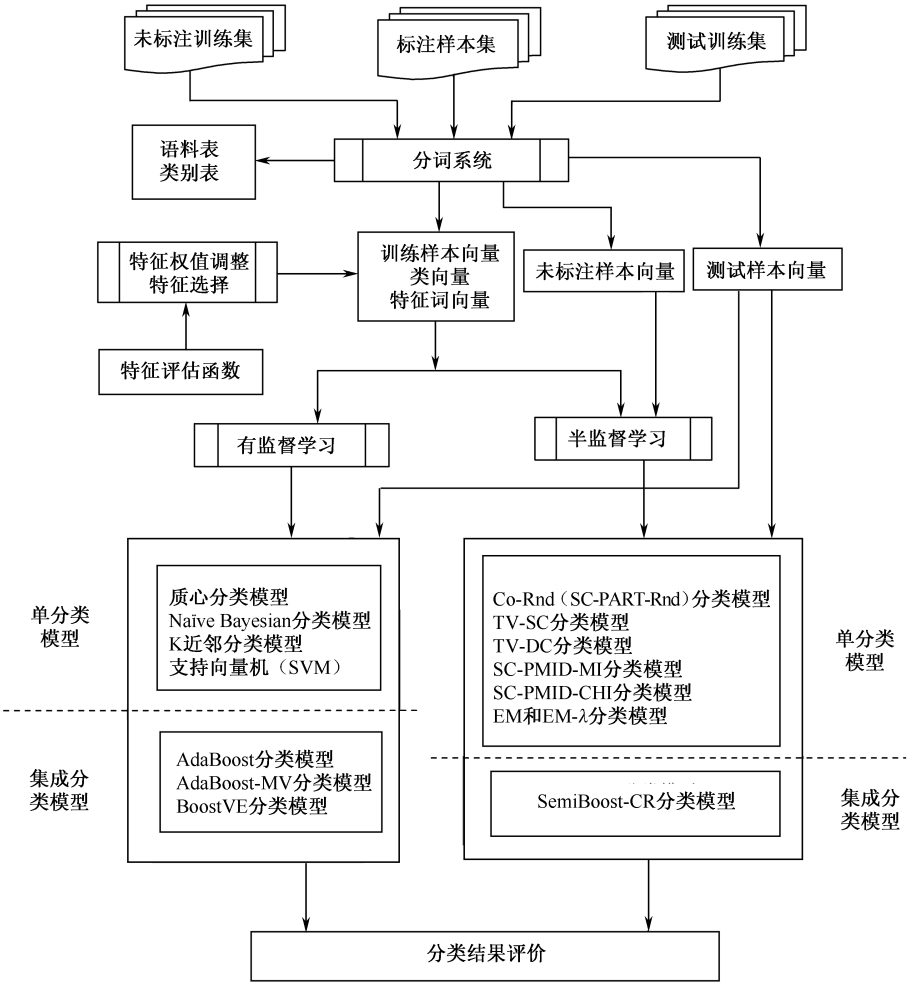


图 8.1　系统总体结构

❑8.3　系统的用户界面

1．系统主窗口

运行 SECTCS 文本分类系统，出现如图 8.2 所示的主窗口，包括样本处理、特征处理、分类模型、分类结果评价、帮助等菜单。其中“分类模型”

菜单包含下列子菜单。

图 8.2　系统主窗口

① 有监督学习。

a．单分类器：Naïve Bayesian 、KNN、质心向量、支持向量机（SVM）。

b．集成分类器：AdaBoost、AdaBoost-MV、BoostVE 算法。

② 半监督学习。

a．Co-training 系列：Co-training、SC-PMID、TV-SC、TV-DC 算法。

b．SemiBoosting：SemiBoost-CR 分类算法（半监督与集成学习的结合）。

c．EM 算法：EM 算法、EM-λ算法。

d．主动学习：有待后续扩展。

2．样本处理界面

图 8.3 描述的是样本处理界面，单击“修改”按钮，出现浏览窗口，选择标注样本集所在的路径，单击“读入”按钮，从磁盘分别读入标注训练样本集，经过分词、词频统计、停用词过滤等转换为文本向量，同时系统还生成“特征词表”、“特征词向量”、“类向量”等必要的数据结构。类似地，选择测试样本集、未标注样本集所在路径，读入测试样本集和未标注样本集。

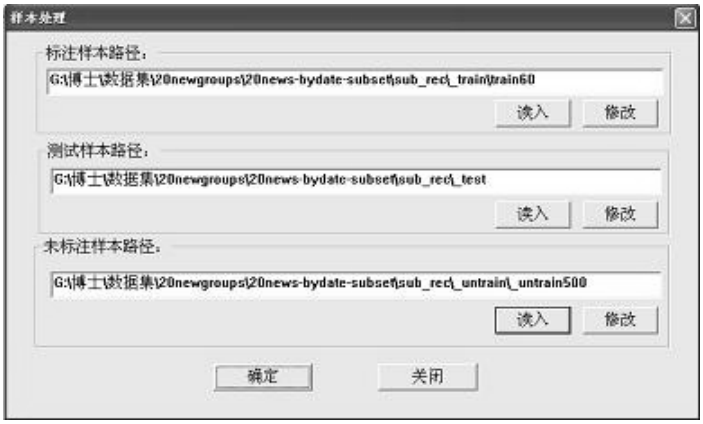


图 8.3　样本处理界面

3．特征处理界面

图 8.4 所示的是权值调整的评估函数和权值计算公式的选择对话框。对话框上方列出了 8 种评估函数，左下部分列出了权值计算时可采用的公式：词频型、文档型。右下方“保留特征的方式”下拉列表中有“按评估分”和“按保留特征数”两种选择，每种方式又可结合是否选中“是否比率”复选框。设置好参数后，单击“特征选择和权重调整”按钮，进行特征选择和权值调整。

图 8.4　特征选择参数设置

4．基于有监督学习的单分类器模型

SECTCS 系统实现了基于有监督学习的单分类器模型：质心分类模型、朴素贝叶斯分类模型、K 近邻分类模型、支持向量机分类模型，如图 8.5 所示。调用该对话框之前，需要先调用图 8.4 所示的特征选择与权值调整对话框。

图 8.5　基于有监督学习的单分类器模型

5．Co-training 的改进算法：SC-PMID 分类算法

第 5 章对半监督分类的代表算法 Co-training 进行了改进，建立了特征相互独立模型（MID-model），基于该模型提出一种特征子集划分方法 PMID 算法，根据评估每对特征相互独立性的方法不同，分别称为 PMID-MI 算法和 PMID-CHI 算法。基于 PMID-MI 算法和 PMID-CHI 算法，提出了改进的 SC-PMID-MI 算法和 SC-PMID-CHI 算法。为了验证所提算法的有效性，与基于随机特征子集分割法的 SC-PART-Rnd 算法进行了对比实验。如图 8.6 所示，设定保留特征数、迭代次数、基分类器参数后，单击“确定”按钮，训练生成相应的分类器，然后对测试样本进行分类，结果输出到指定的 txt 文件中。

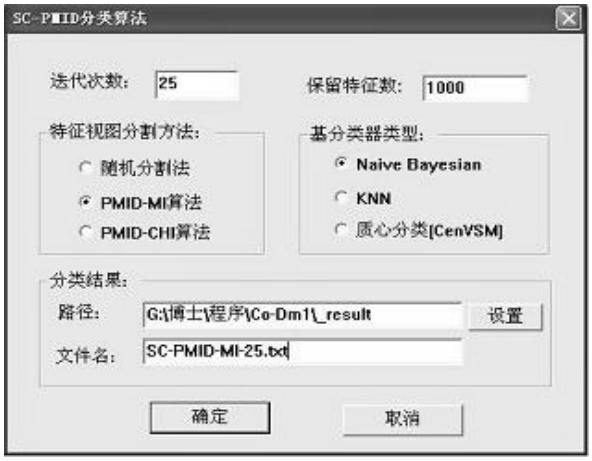


图 8.6　SC-PMID 分类算法界面

6．AdaBoost、AdaBoostMV、BoostVE 分类模型界面

第 6 章对集成学习的代表算法 AdaBoost 进行了改进，实现了基于特征多视图的 AdaBoost-MV 算法、基于投票信息熵与特征多视图的 BoostVE 算法。图 8.7 所示是 AdaBoost 算法及其改进算法 AdaBoostMV、BoostVE 算法的参数设置对话框，其中参数 Lumda 代表第 6.4.1 节公式（6-14）中的参数 η，表示平均投票熵 $\overline{\mathrm{VE}_t}$ 在计算基分类器信任权重时的作用。BoostVE 算法的提出是为了改进 AdaBoost 对 Naïve Bayesian 分类器的提升效果，所以实验中选择的基分类器是 Naïve Bayesian，但是 BoostVE 算法，特别是它的基于投票信息熵的 re-weighting 策略，对提升其他分类算法也有潜在的应用价值。AdaBoost、AdaBoostMV、BoostVE 分类算法的实验结果和分析详见第 6.5 节。

7．SemiBoost-CR 分类模型界面

结合半监督学习和集成学习两种方法，设计实现了基于置信度重取样的 SemiBoost-CR 分类模型，运行界面如图 8.8 所示。对话框中的单选按钮“由 K 个近邻的分类”和“最大差距”分别对应第 7 章提出的两种置信度计算方法：第 7.2 节中式（7-10）的 conf_1 置信度计算方法和式（7-11）的 conf_2 置信度计算方法。其中 Lumda 代表公式（7-10）中的λ，用来调节标注近邻和未

标注近邻在计算 $conf_l$ 置信度时的作用，K 表示近邻数。参数 topN 和 bottomN 表示比例，例如，topN=0.025，bottomN=0.025，表示每次迭代按置信度排序后，选择 2.5%置信度高和 2.5%置信度低的未标注样本加入标注样本集。选择不同的置信度计算公式、不同的 topN、bottomN 对 SemiBoost-CR 的分类结果有不同的影响，实验结果及其分析详见第 7.6 节。

图 8.7　AdaBoost 及其改进算法界面

图 8.8　SemiBoost-CR 分类模型界面[①]

①“SemiBoost-CD”在程序中最初的命名是“SemiBoost-CS”。

8．分类结果输出

为了对比分类模型的性能优劣，在 SECTCS 系统中采用了 Macro-Precision、Macro-Recall、Macro-F1、Micro-F1 及每个类的 Precision、Recall、F1 等多种评估方法，对每个分类模型的分类结果进行评价，给出分类错误信息，并输出到指定的文件中，以便综合分析，如图 8.9 所示。

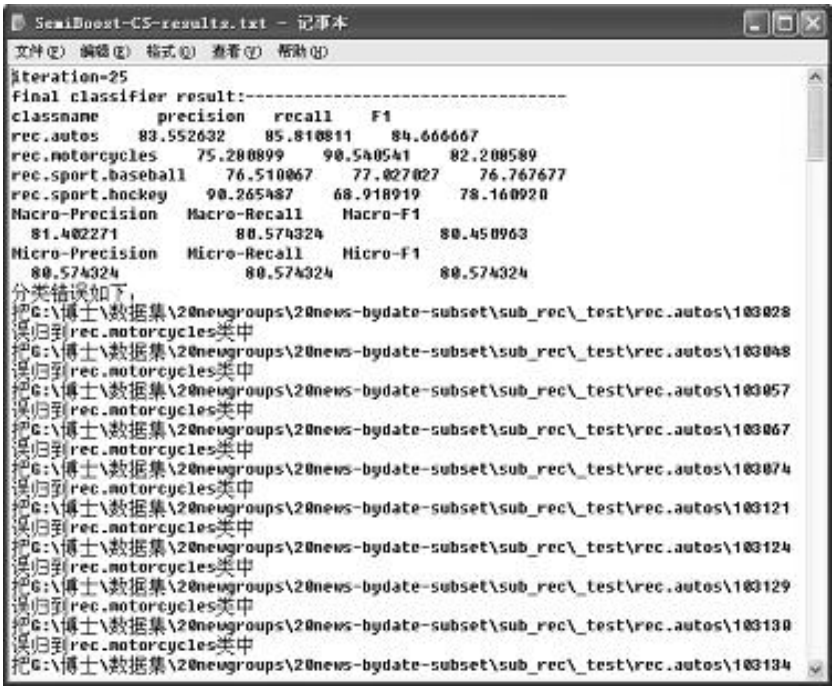


图 8.9　分类结果评价

❑8.4　实验数据集

各种文本分类算法的分类性能优劣的比较，需要一个共同的训练数据集和测试数据集，这里采用了国际上通用的英文文本分类标准数据集 20-newsgroups，以及从易宝中文下载的中文新闻数据集。

1．英文文本分类标准数据集 20-newsgroups

20-newsgroups 数据集[①]包含了大约 20 000 个新闻文本，按照主题划分成 20 个新闻组。最初由 Ken Lang 收集整理，是机器学习中文本分类、文本聚类等研究领域广泛使用的通用实验数据集。20-newsgroups 按照类别主题划分成 20 个新闻组，其中的一些新闻组是比较相近的，如 comp.sys.ibm.pc.hardware/comp.sys.mac.hardware，而有的是高度不相关的，如 misc.forsale/soc.religion.christian。20 newsgroups 数据集主题类别描述如表 8.1 所示。

表 8.1　20-newsgroups 数据集的主题类别

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast talk.religion.misc	alt.atheism soc.religion.christian

2．中文文本分类数据集

从易宝中文下载的中文新闻数据集包括国际、经济、体育、文教、政治 5 个主题类别的 20 341 篇新闻文本，可以看出，经济、政治、国际是比较难以区分的类别，文教和体育两个类别比较相近。实验中，将训练集划分为由不同数量的文本组成的训练文本集和测试文本集，然后做交叉验证，按照 Lewis 划分的要求，训练文本集与测试文本集没有交集。令 *L*n、*U*m、*T*k 分别表示包含 *n* 篇的训练文本集、包含 *m* 篇没有类标签的未标注文本集和包含 *k* 篇的测试文本集。

① http://people.csail.mit.edu/jrennie/20Newsgroups/.

❑8.5　本章小结

SECTCS 是前期针对有监督分类算法开发的中英文文本分类系统，系统原有的功能只实现了几种有监督的分类算法。通过对半监督学习和集成学习的研究，对原来的 SECTCS 系统进行了修改和功能扩展，进一步开发实现了半监督分类方法和集成分类方法中的经典算法，以及前几章所提出的基于半监督学习与集成学习的各种改进算法。本章阐述了 SECTCS 系统原有的功能与新扩展的功能、总体结构、系统用户界面及操作，描述了分类模型的多种评估方法和实验数据集。SECTCS 系统还需要不断修改和完善，以适应下一步的研究工作。

结束语

1．总结

文本分类是机器学习、数据挖掘、网络挖掘、自然语言处理等领域的研究热点，在信息组织和管理、网络信息过滤等领域都有着广泛的应用。但目前面临着缺少标注样本、分类精度难以进一步提高等诸多挑战。鉴于此，本书采用了机器学习领域的半监督学习和集成学习机制，重点对半监督学习的代表算法 Co-training 方法和集成学习的 AdaBoost 方法进行了深入研究，提出了一些比较有效的解决方案，主要完成了以下研究工作。

① 在 Co-training 算法中，通常假设两个特征视图具备一致性和独立性的要求，然而实际应用中同时满足上述条件且自然划分的视图往往不存在，且二者的独立性很难直接评判。结合 TEF-WA 技术，利用多种特征评估函数建立特征多视图，通过评估两个基分类器之间的差异性，可间接评估两个特征视图的独立性，提出了对 Co-training 的改进算法 TV-SC 和 TV-DC，并通过实验证明了所提方法的有效性，而且 TV-DC 算法的分类效果要优于 TV-SC 算法。

② 针对 Co-training 方法的独立性假设问题，提出了利用互信息或 CHI 统计量评估特征之间的相互独立性，建立了一种特征独立模型。基于该模型的特征子集划分方法 PMID 算法，根据评估相互独立性方法的不同，分别称为 PMID-MI 算法和 PMID-CHI 算法，能有效地将一个特征集合划分成两个独立性较强的子集。进而提出了改进的半监督分类算法：SC-PMID-MI 算法和 SC-PMID-CHI 算法。理论分析和对比实验结果表明，SC-PMID-MI 算法和 SC-PMID-CHI 算法优于结合随机分割法的 SC-PART-Rnd 算法。

③ AdaBoost 算法能够有效地提升决策树、神经网络等分类算法，但是对 Naïve Bayesian 算法的提升效果很差。本书分析了 AdaBoost 算法不能有效提升 Naïve Bayesian 分类器的原因，提出了一种基于投票信息熵的样本权重维护新策略，即样本权重的调整不仅考虑是否被当前基分类器分错，还考虑该样本在前几轮基分类器上的投票分歧，而且在错误率相同的情况下，对基分类器间的差异性贡献大的基分类器将会获得更大的信任度。理论分析证明了改进的 BoostVE 算法的最小训练错误上界优于 AdaBoost 算法，对比实验表明 BoostVE 算法能够有效提高 Naïve Bayesian 文本分类器的泛化能力。

④ 结合半监督学习和集成学习方法，提出了一种基于置信度重取样的 SemiBoost-CR 分类模型。采用基于相似近邻和基于最大差距两种置信度计算公式，每次迭代选取一定比例的置信度较高和置信度较低的未标注样本，分别以不同的策略加入已标注的训练样本集。在 20-newsgroups 标准数据集上的大量对比实验表明，使用少量的标注样本和大量的未标注样本，SemiBoost-CR 分类模型能够明显提升 NB 文本分类器的性能。

⑤ 对前期采用 VC++ 6.0 实现的中英文文本分类系统 SECTCS 进行了修改和功能扩展，进一步开发实现了半监督分类和集成分类方法中的经典算法，以及上述研究所提出的各种改进算法，并在 20-newsgroup 数据集和中文新闻数据集上进行了大量的对比实验和分析，表明了所提方法的有效性。

2．进一步的工作

本书的研究工作还存在一些缺陷和不足，还有一些设想没来得及实现，需要在以后的研究中不断改进、拓展和完善。

① 比较两种 Co-training 的改进方法：方法一，结合 TEF-WA 技术改进方法的不足是，两个特征子集有部分重叠，但是基于 MID-Model 的改进方法不存在特征重叠的问题，因而划分得到的特征子集的独立性要优于前者；方法二，基于 MID-Model 的方法存在的缺点是时间复杂度高，因为需要计算特征之间的条件互信息或 CHI 统计量，时间效率低，虽然在算法中采取了减少时间复杂度的措施。相对而言，结合 TEF-WA 技术的方法效率较高。

② BoostVE 算法在基分类器置信度的计算中引入了权重参数 η 以平衡 accuray 和 diversity，参数 η 在实验中采用的是经验值，这是存在的不足。在后续的工作中，将根据训练错误率和平均投票熵动态调整 η 的取值，进一步

改进和完善 BoostVE 算法。另外，BoostVE 算法的核心思想——基于投票信息熵的样本权重维护新策略，在图像分类、图像识别等领域用来提升其他学习算法（包括稳定的和不稳定的学习算法），也有潜在的应用价值。下一步的工作是改进BoostVE算法，尝试用于提升其他学习算法并应用到其他领域。

③ 半监督学习和集成学习相辅相成，集成学习要求基分类器在满足一定正确性的基础上，尽量增加基分类器之间的差异性。而半监督学习中未标注样本的引入能够对基分类器进行扰动，增加差异性。下一步研究的重点是结合半监督学习和集成学习，进一步优化 SemiBoost-CR 分类模型的目标函数，动态调整基分类器信任权重，构建新的 Semi-Boosting 分类模型，一些新的设想将在今后的工作中陆续展开。

参考文献

[1] Jiawei Han, Micheline Kamber. 数据挖掘概念与技术．北京：机械工业出版社，2002．

[2] 史忠植．知识发现．北京：清华大学出版社，2002．

[3] Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 2002, 34(1):1-47．

[4] Fayyad U. M. , Piatetsky-Shapiro G．, Smyth P．. From Data Mining to Knowledge Discovery: An Overview. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996:1-34．

[5] Raymond K．, Hendrik B．. Web Mining Research: A Survey. ACM SIGKDD, 2000(2-1):1-15.

[6] Witten I．H．, Bray Z．, Mahoui M．et al. Text mining: a new frontier for lossless compression. Proceedings Data Compression Conference, Snowbird, Utah: IEEE Pr., 1999:198-207.

[7] Ahonen H．, Klemettinen M, Heinonen A．O．et al. Applying Data Mining Techniques in Text Analysis. Department of Computer Science P.O.Box 26, FIN-00014 University of Iielsinki, Finland, 1997．

[8] Maron M．. Automatic indexing: an experimental inquiry. Journal of the Association for Computing Machinery, 1961, 8(3):404-417．

[9] Salton G．. The SMART Information Retrieval System. Prentic Hall, Englewood Cliffs, NJ, 1971．

[10] Salton G．, Wong A．, Yang C．S．. A Vector Space Model for Automatic Indexing. Communication of the ACM, 1995(18):613-620．

[11] Philip J．, Hayes, Steven P．et al. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. Second Annual Conference on Innovative Applications of Artificial Intelligence, Menlo Park, California: AAAI Pr., 1990．

[12] Sahami M．. Using Machine Learning to Improve Information Access. [PhD dissertation],Computer Science Department, Stanford University, USA, 1999．

[13] Deerwester S．, Dumais S．T．, Furnas G．W．, Landauer T．K．, Harshman R．. Indexing by latent semantic indexing. Journal of the American Society for Information Science. 1990, 41(6)：391-407．

[14] Belkin N．J．, Croft W．B．. Information filtering and information retrieval: two sides of the same coin. Communications of the ACM.1992,35(12): 29-38．

[15] Mladenic D．,Grobelnik M．. Feature selection for unbalanced class distribution and Naive Bayesian . Proceeding of the 16th International Conference on Machine Learning ICMl-99, Morgan Kaufmann Publishers, San Francisco, CA., 1999：258-267．

[16] Yang Y．, Pedersen J．P．. A Comparative Study on Feature Selection in Text Categorization Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), 1997：412-420. (http://www.cs.cmu.edu/~yiming/papers.yy/icml97.ps.gz．)

[17] Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Proc of the 14th International Conference on Machine Learning ICML97,1997. 143-151．

[18] Shankar S, Karypis G. A feature weight adjustment algorithm for document categorization. KDD-2000, Boston, USA, August 2000．

[19] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. Proc of ACM SIGMOD Conf. on Management of Data Washington, 1993：207-216．

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

[21] Vladimir N.Vapnil. 统计学习理论的本质. 北京:清华大学出版社,2000.
[22] 边肇祺，张学工，等. 模式识别. 北京：清华大学出版社, 2000.
[23] Yiming Y. An evaluation of statistical approach to text categorization. In Technical Report CMU-CS-97-127, Computer Science Department, Carnegie Mellon Univ., 1997.
[24] Yiming Y, Sean S, Rayid G. A Study of Approaches to Hypertext Categorization http://www.cs.cmu.edu/~yiming/papers.yy/.
[25] Li Y H, Jain A K. Classification of Text Documents. The Computer Journal, 1998, 41(8):537-546.
[26] 鲁明羽．Web Mining 技术及其应用研究（博士学位论文）．北京：清华大学，2002.
[27] 唐焕玲．文本自动分类方法的研究（硕士学位论文），北京：清华大学, 2004.
[28] 陆玉昌，鲁明羽，李凡．向量空间法中单词权重函数的分析和构造．计算机研究与发展，2002, 39(10)：1205-1210.
[29] 鲁明羽，李凡，庞淑英，陆玉昌，周立柱．基于权值调整的文本分类改进方法．清华大学学报（自然科学版），2003，43(4)：513-515.
[30] 杨绪兵，陈松灿，杨益民．局部化的广义特征值最接近支持向量机．计算机学报，2007，30(08)：1227-1234.
[31] 尹学松，胡思良，陈松灿．基于成对约束的判别型半监督聚类分析.软件学报，2008，19(11)：2791-2802.
[32] DAI Qun, CHEN Song-Can, WANG Zhe. Hybrid Neural Network Architecture Based on Self-Organizing Feature Maps. Journal of Software, 2009, (05)：1329-1336.
[33] 苏金树，张博锋，徐昕．基于机器学习的文本分类技术研究进展．软件学报，2006，17(9)：1848-1859.
[34] Schapire R E. The strength of weak learnability. Machine Learning, 1990, 5(2)：197-227.
[35] Breiman L. Bagging predictors. Machine learning, 1996, 26(2):123-140.
[36] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997, 55(1):119-139.

[37] Robert E, Schapire R E, Yoram Singer. Improved boosting algorithms using confidence-related predictions. Proceedings of the eleventh Annual Conference on Computational Learning Theory, Madison, Wisconsin, USA: ACM Pr., 1998:80-91.

[38] Schapire R E, Singer Y. BoosTexter: A Boosting-based system for text categorization. Machine Learning, 2000, 39(2-3): 135-168.

[39] Seeger M. Learning with labeled and unlabeled data: (Technical Report). Edinburgh: Univ. of Edinburgh, 2001.

[40] Dempster A, Laird N and Rubin D. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 1977(1):1-37.

[41] Nigam K, McCallum A K, Thrun S et al. Text classification from labeled and unlabeled documents using EM. Machine Learning, 2000(39): 103-134.

[42] Nigam K. Using unlabeled data to improve text classification: (Ph.D. dissertation). Pittsburgh: Carnegie Mellon Univ., 2001.

[43] Nigam K, Lafferty J, McCallum A. Using maximum entropy for text classification. IJCAI-99 Workshop on Machine Learning for Information Filtering, Stockholm, Sweden: Morgan Kaufmann Pr., 1999:61-67.

[44] Riloff E, Wiebe J, Wilson T. Learning subjective nouns using extraction pattern bootstrapping. Proceedings of the 7th Conference on Natural Language Learning, Edmonton, Canada: ACM Pr., 2003:25-32.

[45] Maeireizo B, Litman D, Hwa R. Co-training for predicting emotions with spoken dialogue data. The Companion Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Barcelona, Spain: ACM Pr., 2004.

[46] Rosenberg C, Hebert M, Schneiderman H. Semi-supervised selftraining of object detection models. In Seventh IEEE Workshop on Applications of Computer Vision, WACV/MOTIONS '05 (1), Breckenridge, Colorado: IEEE Pr., 2005:29-36.

[47] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. Proceedings of the eleventh Annual Conference on Computational Learning Theory, Madison, Wisconsin: ACM Pr., 1998: 92-100.

[48] Mitchell T. The role of unlabeled data in supervised learning. Proceedings of the Sixth International Colloquium on Cognitive Science. San Sebastian, Spain, 1999.

[49] Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training. Ninth International Conference on Information and Knowledge Management, Washington, DC: ACM Pr., 2000: 86–93.

[50] Goldman S, Zhou Y. Enhancing supervised learning with unlabeled data. Proceedings of 17th International Conf. on Machine Learning, San Francisco, CA: Morgan Kaufmann Pr., 2000: 327–334.

[51] Zhou Y, Goldman S. Democratic co-learning. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL: IEEE Pr., 2004:594-602.

[52] Zhou Z-H, Chen K-J, Jiang Y. Exploiting unlabeled data in content-based image retrieval. Proceedings of ECML-04, 15th European Conference on Machine Learning, Pisa, Italy: Springer Pr., 2004: 525-536.

[53] Zhou Z-H, Li M. Semi-supervised regression with co-training. Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI’05), Edinburgh, Scotland, 2005:908-913.

[54] Zhou Z-H, Li M. Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering, 2005(17):1529-1541.

[55] Zhou Z-H, Zhan D-C, Q Yang. Semi-supervised learning with very few labeled training examples. Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI’07), Vancouver, Canada: AAAI Pr., 2007: 675-680.

[56] Muslea I, Minton S, Knoblock C. Active + semi-supervised learning = robust multi-view learning. Proceedings of ICML-02, 19th International Conference on Machine Learning, Sydney, Australia: MIT Pr., 2002:

435-442.

[57] Chapelle O, Chi M, Zien A. A continuation method for semi-supervised SVMs. Proceedings of ICML06, 23rd International Conference on Machine Learning, Pittsburgh, USA: MIT Pr., 2006:185-192.

[58] Chapelle O, Sindhwani V, Keerthi S S. Branch and bound for semisupervised support vector machines. Advances in Neural Information Processing Systems, Vancouver, B.C., Canada: MIT Pr., 2006:217-224.

[59] Brefeld U, Scheffer T. Semi-supervised learning for structured output variables. Proceedings of the Twenty-Third International Conference（ICML 2006）, Pittsburgh, Pennsylvania, USA, 2006:145-152.

[60] Chapelle O, Weston J, Scholkopf B. Cluster kernels for semi-supervised learning. Advances in Neural Information Processing Systems 15, British Columbia: MIT Pr., 2002: 585-592.

[61] Chapelle O, Zien A. Semi-supervised classification by low density separation. Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005), Barbados: MIT Pr., 2005: 57-64.

[62] Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts. Proceedings of 18th International Conf. on Machine Learning, Williamstown, MA, USA: Morgan Kaufmann 2001:19-26.

[63] Blum A, Lafferty J, Rwebangira M et al. Semi-supervised learning using randomized mincuts. Carla EB, ed. Proceedings of the 21st Int'l Conf. on Machine Learning (ICML 2004), Banff, Alberta, Canada: ACM Pr., 2004: 934-947.

[64] Zhu X J. Semi-supervised learning with graphs: (Ph.D. dissertation). Pittsburgh, PA: Carnegie Mellon University, CMU-LTI-05-192, USA, 2005.

[65] Lawrence N D, Jordan M I. Semi-supervised learning via Gaussian processes. In Saul L K, Weiss Y and Bottou L (Eds.), Advances in neural information processing systems 17, Cambridge, MA: MIT Pr., 2005: 753-760.

[66] Chu W, Ghahramani Z. Gaussian processes for ordinal regression. Journal of Machine Learning Research, 2005, 6(7):1019-1041.

[67] Chu W, Sindhwani V, Ghahramani Z et al. Relational learning with gaussian processes. Neural Information Processing Systems (NIPS-19), Vancouver, B.C., Canada: MIT Pr., 2006: 289-296.

[68] Chapelle O, Zien A, Schölkopf B. Semi-supervised learning. Cambridge, MA: MIT Pr., 2006.

[69] Xiaojin Zhu. Semi-Supervised Learning Literature Survey, 2006. http://www.cs.wisc.edu/～jerryzhu/pub/ssl survey.pdf.

[70] Xiaoli Li, Bing Liu, See-Kiong Ng. Learning to Identify Unexpected Instances in the Test Set. Proceedings of Twenth International Joint Conference on Artificial Intelligence (IJCAI-07), Hyderabad, India: AAAI Pr., 2007: 2802-2807.

[71] Xiaoli Li, Bing Liu. Learning from Positive and Unlabeled Examples with Different Data Distributions. Proceedings of European Conference on Machine Learning (ECML-05), Porto, Portugal: Springer Pr., 2005: 218-229.

[72] Gao Cong, Wee Sun Lee, Haoran Wu, Bing Liu. Semi-supervised Text Classification Using Partitioned EM. Jeju Island: Springer Pr., 2004: 482-493.

[73] Bing Liu, Yang Dai, Xiaoli Li et al. Building Text Classifiers Using Positive and Unlabeled Examples. Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, Florida: IEEE Pr., 2003:19-22.

[74] Xiaoli Li, Bing Liu. Learning to classify text using positive and unlabeled data. Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico: AAAI Pr., 2003.

[75] 宫秀军,史忠植. 基于 Bayesian 潜在语义模型的半监督 Web 挖掘. 软件学报, 2002,13(8):1508-1514.

[76] Quinlan J R. Bagging, Boosting and C4.5. Ben-Eliyahu, Rachel eds. Proceedings of the 13th National Conf. on Artificial Intelligence, Menlo Park, CA: AAAI Press/MIT Press, 1996: 725-730.

[77] Dietterich T A. Experimental Comparison of Three Mothods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization. Machine Learning, 2000, 40(2):139-157.

[78] Sun Y, Wang Y, Wong A K C. Boosting an Associative Classifier. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(7):988-992.

[79] 周志华. 选择性集成（Selective Ensemble）. 第九届中国机器学习会议, 上海：复旦大学，2004.

[80] Tumer K, Ghosh J. Error correlation and error reduction in ensemble classifiers. Connection Sci.8, 3-4, 1996: 385-403.

[81] Ruta D, Gabrys B. A Theoretical Analysis of the Limits of Majority Voting in Multiple Classifier Systems. Pattern Analysis & Applications, 2002, 5(4): 333-350.

[82] Kuncheva L I, Whitaker C J. Measures of diversity in classifier ensembles. Machine Learning, 2003, 51(2):181–207.

[83] Kuncheva L I, Whitaker C J, Shipp C A et al. Limits on the Majority Vote Accuracy in Classifier Fusion. Pattern Analysis & Applications, 2003, 6(1): 22-31.

[84] Valiant L G. A theory of the learnable. Communications of the ACM, 1984, 27(11):1134-1142.

[85] Gomez-Verdejo V, Ortega-Moral M, Arenas-Garcıa J et al. Boosting by weighting critical and erroneous samples. Neurocomputing, 2006 (69): 679-685.

[86] Ji Zhu. Multi-class AdaBoost. 2006. http://www-stat.stanford.edu/~hastie/Papers/samme.pdf.

[87] Yoonkyung Lee, Yi Lin, Grace Wahba. Multicategory Suport Vector Machines: Theory and Application to the classification of Microarray Data and Satellite Radiance Data: (Technical Report No. 1064), Madison, WI: Univ. of Wisconsin, 2002.

[88] Ismet Yalabık, Fatos T, Yarman-Vural A. Pattern Classification Approach for Boosting with Genetic Algorithms.The 22nd International Symposium on Computer and Information Sciences (ISCIS 2007), Ankara,Turkey:

IEEE Pr., 2007:1-6.

[89] Li Stan Z, Zhang Z, Shum H, Zhang H. Floatboost learning for classification, NIPS 15, British Columbia: MIT Pr., 2002.

[90] Domingo C, Watanabe O. Madaboost: A modification of adaboost. Proceedings of 13th Annu. Conference on Compute. Learning Theory, San Francisco: Morgan Kaufmann, 2000:180-189.

[91] 刁力力，胡可云，陆玉昌，石纯一．用 Boosting 方法组合增强 Stumps 进行文本分类(英文). 软件学报，2002，13(08)：1361-1367.

[92] Z H Zhou, Y Jiang. NeC4.5: Neural ensemble based C4.5. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(6): 770-773.

[93] N Li and Z-H Zhou. Selective ensemble under regularization framework. Proceedings of the 8th International Workshop on Multiple Classifier Systems (MCS'09), Reykjavik, Iceland: MIT Pr., 2009: 293-303.

[94] Kuncheva L I. Error Bounds for Aggressive and Conservative AdaBoost. Proceedings of 4th Int'l Workshop on Multiple Classifier Systems, Guilford, UK: Springer Pr., 2003: 25-34.

[95] 李闯，丁晓青，吴佑寿．一种改进的 AdaBoost 算法——AD AdaBoost. 计算机学报，2007，30(1)：103-109.

[96] Lewis D D. Naive (Bayesian) at forty: The independence assumption in information retrieval. Proceedings of ECML-98, Chemnitz, Germany, Springer Pr., 1998: 4-15.

[97] Ting K M, Zheng Z. Improving the performance of boosting for naive Bayesian classification. Ning Zhong, Lizhu Zhou eds. Proceedings of the 3rd PAKDD, Beijing, China: Springer Pr., 1999: 296-305.

[98] Zheng Z. Naive Bayesian classifier committees. Chaire Nedellec, Celine Rouveirol eds, Proceedings of the 10th European Conf. on Machine Learning, Chemnitz, Berlin Germany: Springer Pr., 1998: 196-207.

[99] 石洪波，黄厚宽，王志海．基于 Boosting 的 TAN 组合分类器. 计算机研究与发展，2004，42(2)：340-345.

[100] 唐旭晟，欧宗瑛，苏铁明等．基于 AdaBoost 和遗传算法的快速人脸定位算法. 华南理工大学学报（自然科学版），2007, 35(1)：64-69.

[101] 雷云，丁晓青，王生进．嵌入粒子滤波中的 AdaBoost 跟踪器. 清华大学学报（自然科学版）, 2007, 47(7): 1141-1143．

[102] 王海川，张立明. 一种新的 AdaBoost 快速训练算法. 复旦学报（自然科学版）, 2004, 43(2):27-33．

[103] 唐焕玲，孙建涛，陆玉昌．文本分类中结合评估函数的 TEF-WA 权值调整. 计算机研究与发展，2005，42(1): 47-53．

[104] 朱靖波，王会珍，张希娟. 面向文本分类的混淆类判别技术. 软件学报, 2008, 19(3): 630-639．

[105] Lewis D D, Gale W A. A sequential algorithm for training text classification. Proceedings of the 17th ACM Int'l Conference on Research and Development in Information Retrieval, Berlin: ACM Pr., 1994．

[106] Seung H S, Opper M, Sompolinsky H. Query by committee. Proceedings of the Fifth Workshop on Computational Learning Theory, San Mateo, CA: Morgan Kaufmann, 1992: 287-294.

[107] Dagan I, Engelson S. Committee-based sampling for training probabilistic Classifiers. Proceedings of the 12th Int'l Conf on Machine Learning, Lake Tahoe, Calif: Morgan Kaufmann, 1995: 150-157．

[108] Leskes B, Torenvliet L.The value of agreement a new boosting algorithm. Journal of Computer and System Sciences, 2008, 74(4): 557-586．

[109] D'Alche-Buc F, Grandvalet Y, Ambroise C. Semi-supervised marginboost. Neural Information Processing Systems Foundation (NIPS 2002), British Columbia: Springer Pr., 2002: 553-560．

[110] Roli F. Semi-supervised Multiple Classifier Systems: Background and Research Directions. 6th Int. Workshops on Multiple Classifier Systems (MCS 2005), Seaside, CA: Springer Pr., LNCS 3541, 2005:1-11．

[111] Rong Jin, Jian Zhang. Multi-class learning by smoothed Boosting. Machine Learning, 2007, 67(3): 207-227．

[112] Mallapragada P k, Jin Rong, AK Jain et al. SemiBoost: Boosting for Semi-supervised Learning. IEEE Transactions on Pattern Analysis and Machine Intelegence, 2008, 31 (11):2000-2014．

[113] Forrest S, Javornik B, Smith R et al. Using Genetic algorithms to explore pattern recognization in the immune system. Evaluationary Computation, 1993:191-211.

[114] Didaci L, Roli F. Using Co-training and Self-training in semi-supervised Multiple Classifier Systems. Lecture Notes in Computer Science, Springer Pr., 2006 (4109): 522-530.

[115] Fei Wang, Jingdong Wang, Changshui Zhang et al. Semi-supervised Classification using linear neighborhood propagation. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, USA: IEEE Pr., 2006(1):160-167.

[116] Fei Wang, Changshui Zhang. Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2008, 20(1): 55-67.

[117] Richard O Duda, Peter E Hart, David G Stork. 模式分类（Pattern Classification）. 北京：机械工业出版社，2007.