高维数据的特征选择及
基于特征选择的集成学习研究

（申请清华大学工学博士学位论文）

培养单位：　清华大学计算机科学与技术系

学　　科：　计算机科学与技术

研 究 生：　张 丽 新

指导教师：　王 家 廞 教 授

二〇〇四年四月

Study on Feature Selection and Ensemble Learning Based on Feature Selection for High-Dimensional Datasets

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Engineering

by

Li-xin Zhang

(Computer Science and Technology)

Dissertation Supervisor: Professor Jia-xin Wang

April, 2004

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

图像处理、信息检索以及生物信息学等大规模机器学习问题的不断涌现，对已有的特征选择算法和机器学习算法提出了严峻的挑战，迫切需要适应大规模数据集的准确性和运行效率等综合性能较好的特征选择算法以及机器学习算法。本文在高维数据的特征选择以及基于特征选择的集成学习上开展了研究。主要工作包括以下方面：

一、设计了两种串联型组合式特征选择算法。针对 Relief 评估不能去除冗余特征的缺点，设计了两种串联型组合式特征选择算法：一种为 Filter-Filter 模式，另一种为 Filter-Wrapper 模式。在人工数据集上的实验表明，Filter-Filter 模式的组合式算法可以有效的克服 Relief 不能去除冗余特征的缺点，去掉全部或者近似全部的冗余特征，且运行效率高于 Filter-Wrapper 模式的组合算法；在人工数据集和实际数据集上的实验表明，Filter-Wrapper 模式的组合式算法取得了明显高于 Filter-Filter 模式的测试准确率。

二、基于 Relief 和遗传算法各自的优缺点，提出了 Relief 和遗传算法耦合的组合式特征选择算法。算法采用 Relief 指导遗传算法种群初始化，目的是提高遗传算法搜索近似最优解的速度，以便在较短时间内寻找到近似最优解。在 17 个维数较高的数据集上的实验结果表明，从分类准确率，特征子集大小以及运行时间等多角度考察，该算法具有良好的综合性能。

三、从个体分类器准确率和个体分类器间差异度两方面出发，提出了一种适于高维数据的基于两步式特征选择的集成学习算法 ReFeatEn。实验表明，在特征维数较高，特征间关系较复杂的数据集上，ReFeatEn 算法的测试准确率始终优于或相当于 Bagging、Boosting 和基于随机特征选择的集成学习算法 RandFeatEn，并且 ReFeatEn 的运行速度远高于 Bagging 和 Boosting 算法，而且适于并行运行，是一种适用于高维数据的基于特征选择的集成学习算法。

四、提出了将特征选择嵌入到 Boosting 算法中的思路，并设计了总体算法框架，据此分别针对朴素贝叶斯分类器和最近邻中心分类器设计了相应的集成学习算法，解决了 Boosting 算法对噪声特征较敏感的缺陷，得到的测试准确率显著高于对应的 Boosting 算法，是一种鲁棒性很强且具有推广性的集成学习算法。

关键词：特征选择，高维，集成学习，Relief，遗传算法

Abstract

The emergence of high-dimensional machine learning fields such as image processing, information retrieval and bioinformatics pose severe challenges to the existing feature selection and machine learning algorithms.

This dissertation mainly studies on feature selection and ensemble learning based on feature selection for high-dimensional datasets. Contributions in this dissertation mainly include:

(1). Two two-phase combined feature selection algorithms are designed based on Relief evaluation algorithm. One is with filter-filter model, and the other is with filter-wrapper model. For the filter-filter model, in the first phase, Relief algorithm is used to filter the irrelevant features; in the second phase, correlation analysis is utilized to remove the redundant features. For the filter-wrapper model, the first phase is the same with filter-filter model, while in the second phase, backward sequential search algorithm is used to remove the redundant features with the performance of the induction algorithm to be used after feature selection used as evaluation for the feature subsets. Experiments on artificial and real datasets illuminate that the filter-wrapper combined model outperforms filter-filter model with respect to accuracy while is much slower than filter-filter model, and experiments on artificial datasets illuminate that filter-filter combined model can remove all or equal all redundant features.

(2). Based on the merits and demerits of Relief and genetic algorithm in wrapper model, a coupling model of Relief and genetic algorithm is proposed, which uses the feature evaluation of Relief to instruct the initialization of genetic population, the coupling model aims to improve the efficiency of genetic algorithm which use the performance of the classifier as evaluation of feature subsets. Experiments on 17 relatively high-dimensional datasets show that, the algorithm has good comprehensive performance with respects to accuracy, size of feature subsets, and efficiency.

(3). Considered about the accuracy of individual classifier and diversity among individual classifiers, this dissertation proposes an ensemble learning algorithm

based on two-phase feature selection for high-dimensional datasets. Experiments validate that on high-dimensional datasets, accuracy of ReFeatEn is always higher or equally good as Bagging, Boosting and the random subspace ensemble algorithm RandFeatEn. The efficiency of ReFeatEn is much greater than Bagging and Boosting, and also can be run in parallel, so ReFeatEn is very fit for high-dimensional problems.

(4). Propose the hypothesis of embedding feature selection into Boosting algorithm, and design a general algorithm structure. Accordingly corresponding ensemble learning algorithms for naïve Bayesian classifier and nearest mean classifier are designed. Experiment results and analysis show that this novel coupling algorithm solve the problem that Boosting is sensitive to noise features and samples, and gain accuracy which is remarkably higher than the Boosting algorithm, and is robust and easy to be extended for other classifiers.

Key words: Feature selection, high-dimensional, ensemble learning, Relief, genetic algorithm

目 录

第一章 引 言

1.1 研究意义、目的及研究背景

1.1.1 特征选择

特征选择即从输入特征集合中选择使某种评估标准最优的特征子集。

特征选择是统计学领域的经典问题，自上个世纪 60 年代起就有学者对特征选择问题进行研究，但当时主要是从统计学以及信息处理的角度进行研究，而且所涉及的问题通常特征数目不多[Lewis 1962][Kittler 1978][Cover 1974]。

特征选择也是机器学习领域的重要问题，对机器学习领域的所有问题都有重大意义，包括文本分类，数据挖掘，生物信息学，计算机视觉，信息检索，时间序列预测等。在一个学习算法通过训练样本对未知样本进行预测之前，必须决定哪些特征应该采用，哪些特征应该忽略。在机器学习领域，学习算法方面已经开展了大量的研究，但特征选择方面的研究则相对较少。自 90 年代以来，特征选择方面的研究引起机器学习领域学者前所未有的重视，主要原因有以下两个方面：

1）许多学习算法的性能受到不相关或冗余特征的负面影响。已有的研究结果表明，大多数学习算法所需训练样本的数目随不相关特征的增多成指数性增长[Langley 1994][Jain 1997][Xing 2001]。Langley 等的研究表明最近邻法的样本复杂度随不相关特征成指数增长，其他归纳算法也基本具有这一属性。例如，决策树对于逻辑与概念的样本复杂度随不相关特征线性增加，但对于异或概念的样本却是呈现指数增长；贝叶斯分类器虽然对不相关特征的存在不敏感，但其性能却对冗余特征的存在很敏感[Langley 1993][Langley 1994]。因此，特征选择对不同情况下的学习算法都有不可忽视的作用。选择好的特征不仅可以减小计算复杂度，提高分类准确度，而且有助于寻找更精简更易理解的算法模型。

2）大规模数据处理问题的不断出现。所谓大规模，一方面指样本数目的庞大，另一方面指描述样本的特征维数高。数据挖掘的发展对大规模数据处理

的研究提出了迫切的要求，如信息检索，遗传基因分析等[Jain 1997][Xing 2001]。特征空间的维数不宜过高，这已经是机器学习领域中一条经验性的“公理”，如此就迫切需要特征选择算法对高维数据进行降维，而高维数据的特征选择也对已有的特征选择算法提出了严峻的挑战。

由于上述原因，特征选择成为机器学习领域重要的研究方向，引起越来越多的机器学习领域学者的兴趣。国内外的各大研究机构如 CMU，Stanford，Washington，南京大学，哈尔滨工业大学，北京工业大学等都开展了相关研究[Kohavi 1997][Huang 1999] [章新 1998][张鸿宾 1999]。

特征选择算法有两种主要框架，即 Filter 和 Wrapper。在研究早期，算法主要为 Filter 类，自 Kohavi 系统提出 Wrapper 框架后[Kohavi 1997]，两类算法研究都很多。这两类算法具有很强的互补性，表现在 Filter 运行速度快但相对于后续学习算法评估偏差较大，而 Wrapper 相对于后续学习算法评估准确但运行速度慢，关于两者组合的研究较少。

本文研究的重点之一就是大规模数据的组合式特征选择问题，目的是设计一种适用于大规模数据的综合 Filter 速度快和 Wrapper 准确率高优点的特征选择算法，以降低后续学习算法的时间和空间复杂度，同时保持甚至提高学习算法的泛化性能。

1.1.2 基于特征选择的集成学习

高维数据的分类是公认难度较大的模式识别问题，在数据分布复杂或者维数很高，样本数较少的情况下，很难找到一个最优的分类器。集成学习为解决此类问题提供了一条途径，集成学习是通过集成多个分类器得到的学习系统，在机器学习中效果明显，成为机器学习领域一大研究热点。

集成学习在构造方面研究较多的是基于改变样本分布的构造方法，近年来也出现了基于特征选择的集成学习方法，但相对基于样本的集成学习而言研究较少。

实际应用中的高维问题通常具有特征间关系复杂，无关特征和冗余特征都存在的特点，在此情况下，基于特征选择的集成学习是一种很有潜力的方法。

本文在特征选择的基础上，对高维数据的基于特征选择的集成学习算法进行了研究，并对如何采用特征选择改进经典的 Boosting 算法进行了研究。

1.2 研究内容与主要工作

本文作者在选题时，因发现许多研究人员以各种分类器为研究课题，但通常在研究过程中很重要的一部分工作就是特征选择，同时高维问题的不断出现对特征选择研究提出了严峻的挑战，因此选择了以研究高维数据的特征选择问题为博士课题。最初确定手写体汉字识别为实验平台，但研究过程中发现，手写体汉字的 256 维模糊方向线素特征并不是一个很好的特征选择平台，而且一种好的算法应具有一定普适性，故实验对象应尽量多样，因此又选择了 UCI 数据库中特征维数大于 20 的数据集作为实验数据，在最初选定公认较好的特征评估算法 Relief 作为初步的实验对象后，根据 Relief 不能去除冗余特征的特点设计了两种串联型组合式特征选择算法，即 Relief 和相关分析结合的特征选择算法，以及 Relief 和顺序后向搜索算法的 Wrapper 式结构结合的特征选择算法。但实验发现应用这两种算法，分类准确率和运行效率难以同时取得较好的效果，而且发现 Relief 去除无关特征的能力较弱，并且 Relief 有可能会把和类别相关的特征去除。在分析 Relief 和遗传算法用于特征选择优缺点基础上，本文提出了 Filter 和 Wrapper 耦合度更高的 Relief-GA-Wrapper 算法，实验表明效果很好。

Relief-GA-Wrapper 进化后会产生较多的特征选择结果，只是我们在进行特征选择时在众多特征选择方案中挑选了性能最好的一个，而别的就被抛弃了，但也许各个特征选择方案都对分类有不同的作用，这使本文作者产生了利用多个特征子集进行集成学习的想法。在验证 Relief-GA-Wrapper 的集成学习性能后，发现集成学习的性能在绝大多数情况下，都比单个最好的分类器性能要好。本文在集成学习分类器的理论分析基础上，设计了另一种效率更高的基于两步式特征选择的集成学习算法 ReFeatEn，实验结果表明该算法对高维数据的决策树分类器很有效。继而在实验过程中，本文发现经典的集成学习算法 Boosting 和 ReFeatEn 对朴素贝叶斯分类器和最近邻中心分类器效果不明显，尤其是 Boosting 算法对有噪声的数据集效果较差，ReFeatEn 算法对这两种分

类器效果也一般，本文作者提出了将特征选择嵌入到 Boosting 算法中的思路，并针对朴素贝叶斯分类器和最近邻中心分类器分别设计了两种新算法 FeatBoostNBC 和 FeatBoostNearMean，实验表明这两种算法对朴素贝叶斯分类器和最近邻中心分类器的性能提高很显著，尤其是很好地解决了 Boosting 对噪声特征较敏感的缺陷。

以上介绍的是本文作者博士研究的工作起源以及研究思路发展过程。下面就介绍一下本文的主要研究工作和创新之处。

1.2.1 本文的主要工作

本文针对高维数据的机器学习问题，重点研究了高维数据的组合式特征选择算法和基于特征选择的集成学习问题。

在特征选择算法的研究方面，本文在分析特征选择算法的 Filter 和 Wrapper 模式基础上，选择 Relief 算法作为基础，验证了 Relief 在手写体汉字数据集上的评估性能，接着提出了用分类器性能决定选择多少 Relief 排序后的特征的 Relief-Wrapper 算法，又设计了首先进行特征变换然后采用 Relief-Wrapper 算法进行特征选择的 PCA-Relief-Wrapper 算法。继而，提出了两种串联型组合式特征选择算法：一，Relief 和相关分析组合的 ReCorre 算法；二，Relief 和顺序后向搜索算法的 Wrapper 式结构组合的 ReSBSW 算法。随后，在对 Relief，ReCorre，ReSBSW 分析基础上，提出了 Relief 和遗传算法耦合的 Relief-GA-Wrapper 算法，并在 17 个特征数目大于 20 的数据集上进行了实验比较和分析。

在基于特征选择的集成学习方面，本文在基于特征选择算法的研究和分析前人提出的基于随机特征选择的集成学习算法基础上，通过在 Relief-GA-Wrapper 的多个特征选择结果上的分类器集成，得到了 ReGAWrapperEn 集成学习算法。继而，提出了一种基于两步式特征选择的集成学习算法 ReFeatEn，并将 ReFeatEn 算法和几种流行的集成学习算法进行了实验比较，并分析了集成学习算法和特征子集大小以及个体分类器个数之间的关系。在以上集成学习研究过程中，我们发现 Boosting 算法对朴素贝叶斯分类器和最近邻中心分类器效果不稳定，尤其是对无关特征或者信息丢失较多特征的存在较

为敏感，针对以上问题，本文提出了将特征选择嵌入到 Boosting 中的思路，并分别针对朴素贝叶斯分类器和最近邻中心分类器设计了 FeatBoostNBC 算法和 FeatBoostNearMean 算法，并进行了实验比较和分析。

1.2.2 本文创新之处

本文的创新之处主要在于：

- 基于 Relief 特征评估方法，设计了两种串联式组合特征选择算法：（1）Relief 和相关分析结合的 ReCorre 算法：（2）Relief 和顺序后向搜索的 Wrapper 式结构结合的 ReSBSW 算法。实验分析表明 ReCorre 可以快速而较有效的去除冗余特征， ReSBSW 运行效率较低但测试准确率更好。
- 提出了一种 Filter 和 Wrapper 耦合的特征选择算法，在 17 个高维数据上的实验结果表明，从分类准确率，特征子集大小以及运行时间等多角度考察，该方法具有良好的综合性能。
- 提出了一种适用于高维数据的基于两步式特征选择的集成学习算法，从理论和实验上分析和验证了该方法相对于典型集成学习算法的优点，拓展了基于特征选择的集成学习研究。
- 提出了将特征选择嵌入到 Boosting 算法的思路，并设计了总体算法框架，据此分别针对朴素贝叶斯分类器和最近邻中心分类器设计了相应的集成学习算法，解决了 Boosting 算法对噪声特征较敏感的缺陷，测试准确率比 Boosting 算法有明显提高，是一种鲁棒性很强且具有推广性的集成学习算法。

1.3 论文内容编排

本文第一章介绍高维数据特征选择的研究背景和意义，以及基于特征选择的集成学习在高维数据机器学习上的研究背景和意义。

第二章全面介绍特征选择研究的历史与现状，特征选择的基本概念和一般过程，典型的特征选择算法，及特征选择的主要研究方向等。

第三章重点描述了本文作者在高维数据的特征选择算法上的研究工作，并在人工数据集和实际数据集上通过实验分析了 Relief 特征评估方法，Relief 和相关分析集合的 Filter-Filter 特征选择算法，Relief 和顺序后向搜索算法结合的 Fiter-SBS-Wrapper 算法，Relief 和遗传算法耦合的 Relief-GA-Wrapper 算法，并和前人提出的几种方法进行了实验比较和分析。

第四章重点介绍本文作者在高维数据的集成学习方面的研究工作。本章首先对集成学习给予概述，然后提出了基于两步式特征选择的集成学习算法 ReFeatEn，并和流行的 Bagging，Boosting 以及基于随机特征选择的集成学习算法进行了实验比较和分析，并分析了 ReFeatEn 算法和特征子集大小以及个体分类器数目的关系。

第五章介绍了本文提出的嵌入特征选择到 Boosting 中的思路和算法总体框架，详细介绍了针对朴素贝叶斯分类器设计的 FeatBoostNBC 算法，以及针对最近邻中心分类器设计的 FeatBoostNearmean 算法，并分别和 Boosting 算法以及第四章提出的 ReFeatEn 算法等进行了实验比较。

最后一章总结论文中的主要工作和结论，并展望进一步工作方向。

第二章 机器学习中的特征选择概述

2.1 特征选择研究的历史及现状

机器学习问题可以如下定义：给定一个样本描述和目标函数值已知的样本集，我们利用学习算法逼近其真正的目标函数，并对新来的样本进行预测。描述样本的属性就是特征，也常被称为变量(Variable)、属性(Attribute)、特性(Characteristic)等。本文主要用分类问题来说明采用的特征选择算法，虽然本文提出的特征选择算法并不局限于分类问题。

数据集的大小可以从两方面衡量：特征的数目 n 和样本的数目 p，n 和 p 可能很大，而 n 的庞大经常会引起维数灾难（Curse of Dimensionality）等问题。特征空间的维数不宜过高，这已经是模式识别领域中一条经验性的“公理” [[Bishop 1995][Jain 2000]。从一般意义上讲，许多分类器都要对许多未知参数进行估计（如概率分布密度中的有关参数），而这些参数的个数往往与特征空间的维数是有关的（如协方差矩阵的阶就等于特征空间的维数）。如果特征空间维数过高，那么要准确估计这些参数必须具备大量的训练样本，或者说，在训练样本数目有限的实际情况下，过高的特征空间维数会导致参数估计的准确率下降，进而影响分类器的性能。因此，一些文献中提出了对于样本数据降维的方法[Jain 2000] [Robit 2000]，以得到对分类最有效的特征。

数据降维常用的两类方法是特征选择和特征变换。特征变换是指将原有的特征空间进行某种形式的变换，以得到新的特征。主成分分析是这类方法中最著名的算法，该算法对许多学习任务都可以较好降维，但是特征的理解性很差，因为即使简单的线性组合也会使构造出的特征难以理解，而在很多情况下，特征的可理解性是很重要的。另外，由于特征变换的新特征通常由全部原始特征变换得到，从数据收集角度看，并没有减少工作量。

特征选择是指从原始特征集中选择使某种评估标准最优的特征子集。其目的是根据一些准则选出最小的特征子集，使得任务如分类、回归等达到和特征选择前近似甚至更好的效果。通过特征选择，一些和任务无关或者冗余的特征被删除，简化的数据集常常会得到更精确的模型，也更容易理解[Liu 1998b]。

最早的特征选择研究是 60 年代初开始的 [Lewis 1962]，当时的研究通常集中于统计学及信号处理问题，而且一般涉及到的特征较少，并且通常假定特征间独立[Cover 1974][Natendra 1977][Kittler 1978]。

上个世纪 90 年代以来涌现的大规模机器学习问题[Xing 2001][Jain 1997][Yang 1997]，使得已有的特征选择算法受到严峻的挑战，迫切需要适应大规模数据的准确性和运行效率等综合性能较好的特征选择算法。特征选择引起机器学习领域学者广泛的研究兴趣[Kohavi 1997][Huang 1999][Ng 1998]。

近十年来，特征选择研究呈现出多样化和综合性的趋势。各种新搜索算法和评估标准都应用到特征选择算法中。如粗糙集算法 [Kohavi 1994][Zhong 2001]，神经网络剪枝法 [Setiono 1997]，支持向量机的评估标准 [Weston 2000][范劲松 2001]，特征集的模糊熵评价[Lee 2001]，马尔可夫算法等[Koller 1996]。并且除监督式学习的特征选择研究外，也开展了关于非监督式学习的特征选择研究[Dash 2000]。另外还出现了关于特征选择的算法融合性的研究，如关于 Filter 方法和 Wrapper 方法结合的研究[Huang 1999] [Das 2001]，以及特征选择和样本选择的组合研究[Liu 2002]。

下面章节将对特征选择的基本概念，特征选择和学习算法的关系，以及典型的特征选择算法等给予概述。

2.2 特征选择的基本概念及一般过程

2.2.1 机器学习中特征选择的定义

机器学习中的特征选择可定义为：已知一特征集，从中选择一个子集使评价标准最优[Langley 1994]。以上定义可表述为：

给定一个学习算法 L，一个数据集 S，数据集 S 来自一个具有 n 个特征 X_1，X_2，X_3，…，X_n，具有类别标记 Y，以及符合分布 *D* 的例子空间，则一个最优特征子集 X_{opt} 是使得某个评价准则 J=J(L,S) 最优的特征子集。

从特征选择的定义可见，在给定学习算法，数据集，以及特征集的前提下，各种评价准则的定义和优化技术的应用将构成特征选择的重要内容。

特征选择是机器学习领域的难题。在实际应用中，寻找最优的特征子集通

常是很难的，很多和特征选择相关的问题被发现是 NP-hard 问题[Hyafil 1976]。[Davies 1994]证明寻找满足要求的最小特征子集是 NP 完全问题。但通过采用启发式搜索算法，还是可以在运算效率和特征子集质量间寻找到一个好的平衡点的，这也是众多特征选择算法努力的目标。

2.2.2 特征选择和学习算法的关系

2.2.2.1 特征选择和学习算法的相互作用

图 2-1 显示了包含特征选择 的机器学习系统基本组成部分，可见，整体上来说，特征选择是机器学习算法的前处理阶段。

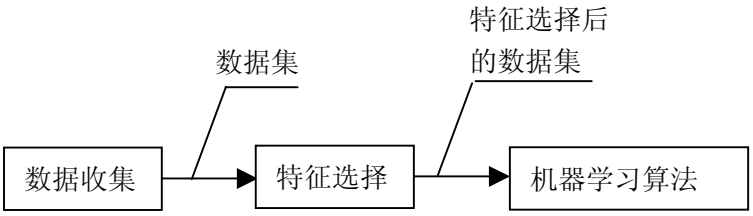

图2-1 机器学习系统的基本构成

特征选择和学习算法的关系可以从理论和实际应用两方面考虑。从理论上讲，对于分类问题，如果存在某个识别函数，那么只有那些在识别函数中出现的特征才是需要的。而在实际应用中，常常以某特征对学习算法是否有用来作为选择的依据。因为实际应用中事先并不知道识别函数，这个识别函数正是要学习的，学习总要选择一个学习算法来进行逼近，如决策树、神经网络、贝叶斯估计等，未知样本的预测也要通过某学习算法得到的分类器来进行，因此一个特征或者特征子集是否该选择，不应离开学习算法而进行。[Kohavi 1997]的实验和分析也支持了上面的论断。

对学习算法来说，有效的特征选择可以降低学习问题的复杂性，提高学习算法的泛化性能，简化学习模型。为了准确的估计参数或者对未知样本进行预测，训练样本的个数是随着特征数目指数增长的，这就是所谓的维数灾难(Curse of Dimensionality)。许多学习算法的性能受到无关特征和冗余特征的影响，比如朴素贝叶斯分类器，NBC 算法假设各特征独立，对具有 n 个特征

$(A_1，A_2，\cdots，A_n)$的样本 X，其类别为 y 的概率按如下贝叶斯公式估计：

$$P(y \mid X) = P(X \mid y) \cdot P(y)/P(X) \propto P(A1，\cdots，An \mid y) \cdot P(y) = \prod P(Aj \mid y) \cdot P(y) \quad \text{公式(2-1)}$$

采用 y_1， y_2，…，y_c 表示类别号，如类别为 y_k 的后验概率最大，则将样本 X 分到 y_k 类。如果特征 A_j 和目标 Y 无关，则理论上有 $P(A_j \mid y_1)=P(A_j \mid y_2)=\cdots=P(A_j \mid y_c)$，相当于用来比较的 $P(y \mid X)$都乘以相同的系数，因此理论上对分类没有影响。但如果特征 A_j 为冗余的相关特征，则公式（2-1）右侧会出现 $P(Aj \mid y)^2$ 项，从而影响 $P(y \mid X)$的值，进而影响 NBC 的分类性能。对于最近邻法[Atkeson 1997]，需要计算样本间的距离，这样不管是冗余特征还是无关特征，都会对距离的计算产生影响，势必会降低算法性能，而且研究者发现最近邻分类器容易陷入维数灾难中，尤其是当存在很多无关特征的情况下[Friedman 1997]。另外，决策树和神经网络的性能也同样受到无关或者冗余特征的影响[Langley 1994][Yang 1997]。

2.2.2.2 特征选择和学习算法结合的三种情况

特征选择和后续学习算法的结合方式可分为嵌入式(Embedded)、过滤式（Filter）和 Wrapper 式三种。

嵌入式特征选择

在嵌入式结构中，特征选择算法本身作为组成部分嵌入到学习算法里。如某些逻辑公式学习算法是通过向公式表达式中加减特征实现的[Blum 1992]。类似的加减特征操作也构成一些更复杂的逻辑概念推导的核心，只是通过不同特征组合形成更复杂的规则描述。最典型的即决策树算法，如 Quinlan 的 ID3 和 C4.5[Quinlan 1983][Quinlan 1993]以及 Breiman 的 CART 算法[Breiman 1984]等，算法在每一结点选择分类能力最强的特征，然后基于选中的特征进行子空间分割，继续此过程，直到满足终止条件，可见决策树生成的过程也就是特征选择的过程。

过滤式特征选择

如图 2-2 所示，过滤式特征选择的评估标准独立于学习算法，直接由数据

集求得。过滤式特征选择的评估依赖于数据集本身，通常是选择和目标函数相关度大的特征或者特征子集，一般认为相关度较大的特征或者特征子集会对应得到后续学习算法较高的准确率。过滤式特征选择的评估方法很多，如类间距离、信息增益、关联度(Correlation)以及不一致度等等。过滤式特征选择因为通常运行效率较高而适用于大规模数据集，但[Kohavi, 1997]指出寻找和类别相关的特征/特征子集和选择可最优化分类准确率的特征/特征子集是两个不同的任务。直接用分类器准确率作为特征子集评估标准的特征选择算法就是下面要介绍的 Wrapper 式特征选择。

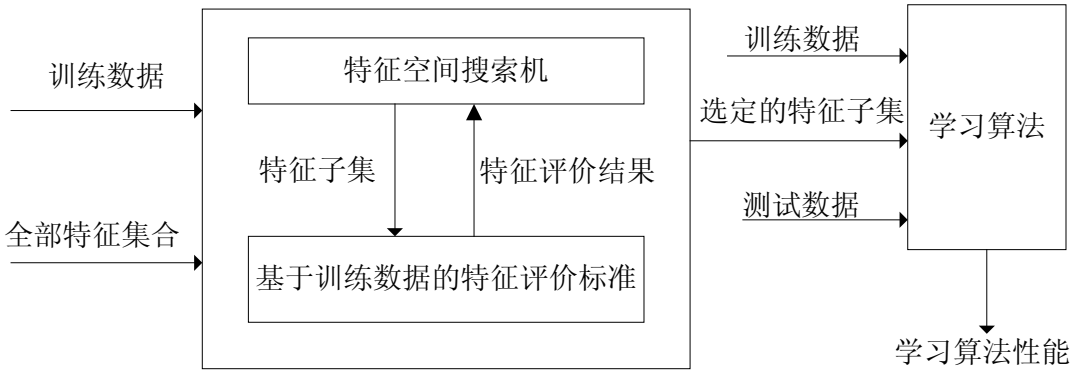

图2-2 过滤式特征选择一般流程

Wrapper 特征选择

Wrapper 特征选择算法最早由 John 等在 1994 年提出[John 1994]，如图 2-3 所示。该算法的核心思想是：和学习算法无关的过滤式特征评价会和后续的分类算法产生较大的偏差，而学习算法基于所选特征子集的性能是更好的特征评价标准。不同学习算法偏好不同的特征子集，既然特征选择后的特征子集最终将用于后续的学习算法，那么该学习算法的性能就是最好的评估标准。因此在 Wrapper 特征选择中将学习算法的性能作为特征选择的评估标准。

Wrapper 特征选择算法中用以评估特征的学习算法是没有限制的。如，John 等选用决策树[John 1994]，Aha 等将最近邻法 IB1 和特征选择算法相结合对云图进行分类研究[Aha 1994]，Provan，Inza 等则利用贝叶斯网络性能指导贪心的前向搜索算法[Provan 1995][Inza 2001]。

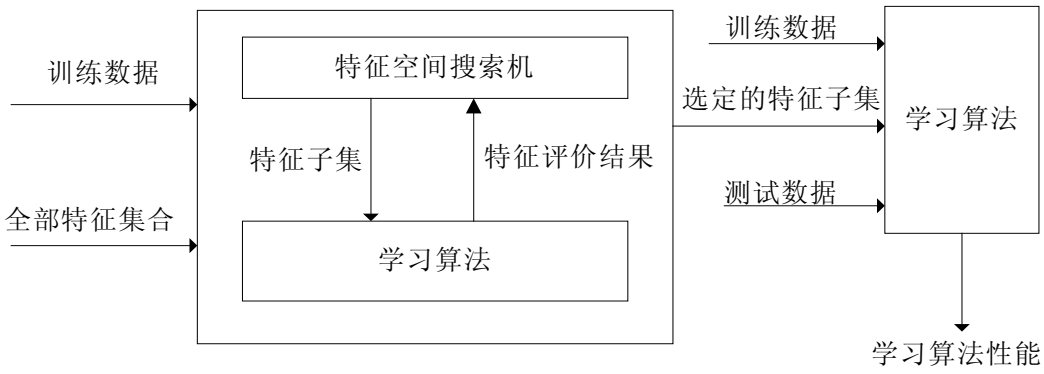

图2-3 Wrapper 特征选择一般流程

由于采用学习算法的性能作为特征评估标准，Wrapper 特征选择算法比过滤式特征选择算法准确率高，但算法效率较低。因此一些研究者努力寻找使评价过程加速的方法。Caruana 等提出一种加速决策树的方法[Caruana 1994a]，即在特征选择过程中大量减少决策树分支的数目。Moore 等通过减少评估特征阶段的分类器的训练样本来提高特征选择的速度[Moore 1994]。Wrapper 方法的另一个缺点是过适应问题，但该问题主要发生在训练数据规模较小的情况，而且研究者实验发现 Wrapper 算法的过适应并不多见[Kohavi 1997][Das 2001]。

2.2.3 特征选择作为搜索问题的四要素

一般而言，特征选择可以看作一个搜索寻优问题。对大小为 n 的特征集合，搜索空间由 2^n 种可能状态构成。显然，即使在特征数目不高的情况下，搜索空间也是庞大的，如当 n=20 时，搜索状态约 10^6 个，再考虑到特征评估过程的开销，在实际应用中，对特征数目较多的情况，穷尽式搜索通常是不可行的。因此，研究者们致力于用启发式搜索算法寻找最优解。最早的启发式特征选择算法是 Lewis 于 60 年代早期提出的[Lewis 1962]，Lewis 提出在假设特征间相互独立的前提下，可以逐个对单个特征进行评估，然后选择其中评价最好的 k 个特征组合在一起就可以组成特征数目为 K 的性能最好的特征子集。这个命题最有名的反例是[Cover 1974]设计的，该反例表明即使特征间独立，两个最优的特征组合不一定是最优的组合。[Toussaint 1971] 更确凿的证明，单个评价最优的特征不一定包含在最优的特征子集中。继而，[Davies 1994]等证

明最小特征子集的搜索是一个 NP 问题，即除了穷尽式搜索，不能保证找到最优解，因此人们开始致力于用启发式搜索算法寻找近似最优解。

如图 2-4 所示，一般而言，特征选择算法必须确定以下四个方面：1）搜索起点；2）搜索策略；3）特征评估函数；4）终止条件。图 2-4 中所示验证阶段不属于特征选择过程，该阶段用于最终验证特征选择的性能。本文重点介绍搜索策略和评估函数，其余两方面则概要介绍，感兴趣的读者可参阅[Liu 1998b]。有些特征选择算法只包含以上四个要素的部分内容，比如特征排序后选择前 m 个特征的特征选择算法，就只涉及特征评估和终止条件两方面。

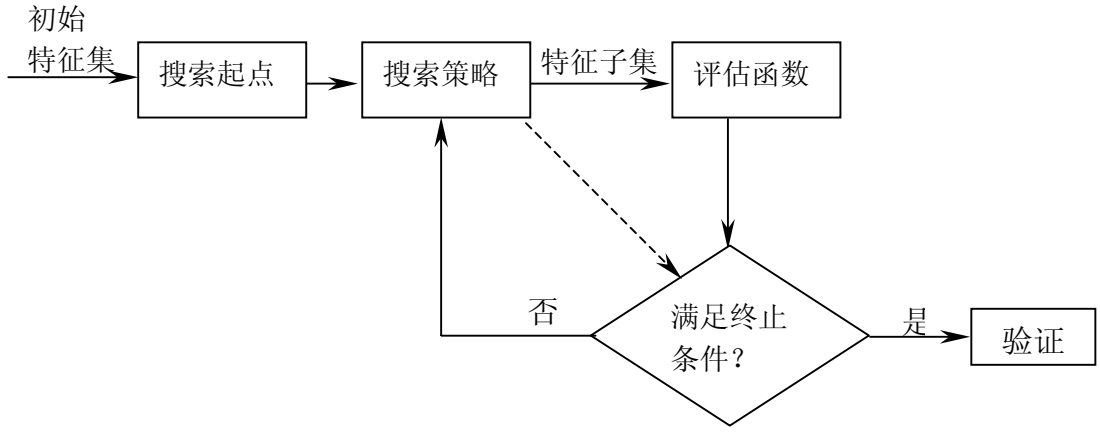

图2-4 特征选择过程

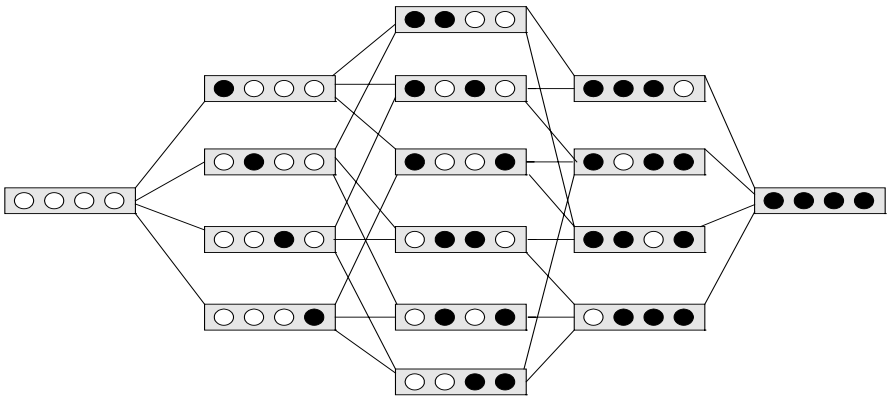

图2-5 特征空间的一种偏序状态图。 从左到右表示从空集到全集，每个状态指定了选择的特征，黑圈表示特征被选中，每个子结点比其父结点多选择一个特征

2.2.3.1 搜索起点

搜索起点是算法开始搜索的状态点，搜索起点的选择对搜索策略有重要影

响。图 2-5 为特征空间的一种偏序状态图，其中每个子结点比父结点多一个特征。如果搜索从空集开始，逐个地向集合里加入特征，即所谓的前向搜索；如果搜索起点为全集，然后不断地删减特征，即所谓的后向搜索；如果搜索从特征空间的中间结点开始，那么搜索策略通常就是随机的或者启发式的搜索。

2.2.3.2 搜索策略

根据搜索方向，搜索策略可以分为前向、后向和双向三种。对于小规模的特征集合，可采用穷尽式搜索求得最优子集。对于中等规模的特征集合，当评估函数对特征维数满足单调性时，可采用 Narendra 等提出的分支界限法（BB）求解最优特征子集[Narendro 1977]。但实际问题中，评估函数通常不具备单调性，同时 BB 算法的算法复杂度与特征个数之间是指数关系，在 n 较大时由于计算量太大而无法应用。因此人们一直致力于寻找能得到较好次优解的搜索算法。

非穷尽式特征子集搜索算法大致可分为顺序搜索和随机搜索两类。顺序搜索算法采用顺序向解集中加减特征逐步扩展搜索，如图 2-6 所示的顺序前向搜索(SFS)，图 2-7 所示的顺序后向搜索(SBS)，以及广义的顺序前向搜索(GSFS)和广义的顺序后向搜索(GSBS)等。该类算法的缺点是，特征一旦被加入或删除，以后便不会改变，因此容易陷入局部极值。为克服此缺点，出现了增 l 减 r 法，即搜索方向不再是单向加或者减，可以根据评估函数灵活的浮动，其问题在于 l 和 r 的大小难以确定[Liu 1998b]。Pudil 等提出了顺序浮动前向搜索(SFFS)和顺序浮动后向搜索算法(SFBS)[Pudil 1994]，算法变固定的增 l 减 r 法为浮动的，减少了不必要的回溯并在需要时增加回溯的深度，Somol 等进一步提出了自适应浮动搜索算法[Somol 1999]，根据当前特征子集的大小和目标特征子集的大小来控制搜索空间的大小，这种方法减小了陷入局部极值的可能性。随机搜索算法包括遗传算法、模拟退火和集束式搜索(Beam Search)等[Liu 1998b]。遗传算法在特征选择中的应用研究很多，并且显示出良好的性能[Yang 1998][Casillas 2001]。关于各搜索算法的优劣并没有一致的意见，但根据 Jain 和 Kudo 的实验，自适应浮动搜索算法和遗传算法是众多搜索算法中性能较好的算法[Jain 1997][Kudo 2000]。表 2-1 和表 2-2 列出了常见的特征选择搜索算法和算法时间复杂度。

2.2.3.3 特征选择的终止条件

特征选择的终止条件有最大迭代次数，性能不改进的循环迭代次数，找到满足评价函数的特征集合等[Langley 1994]。

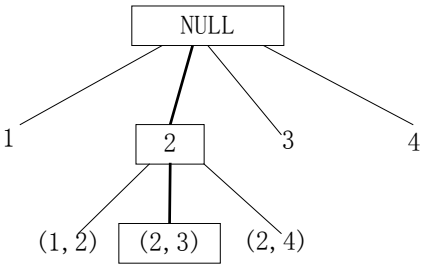

图2-6 顺序前向搜索（SFS）

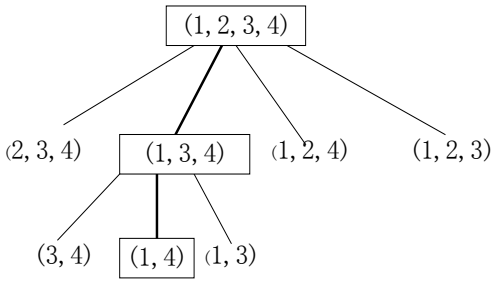

图2-7 顺序后向搜索(SBS)

表2-1 特征选择的搜索算法总结

算法名称	算法描述
SFS SBS	顺序前向搜索算法(Sequential Forward Search) 顺序后向搜索算法(Sequential Backward Search) 。SFS 从空集开始，先选择评估最好的一个特征 k，然后在选择包括 k 的评估最好的两个特征，如此继续，特点是选中的特征就不会被删掉。SBS 从全集开始，不断删除特征，特点是删掉了就不会再被选中
GSFS(*g*) GSBS(*g*)	广义的顺序前向搜索算法和顺序后向搜索算法（General Sequential Forward/Backward Search）每次评价大小为 *g* 的特征子集，每步操作为 *g* 个特征被加入(前向搜索)或者删除（后向搜索）
PTA(*l*,*r*) GPTA(*l*,*r*)	加 *l* 减 *r* 法 Plus *l* Take Away r，前行 *l* 步（通过 SFS 方法加入 *l* 个特征），后退 *r* 步（通过 SBS 方法减掉 *r* 个特征），GPTA(*l*,*r*)选用 GSFS(*l*)方法加特征，GSBS(*r*)方法减特征
SFFS SBFS	PTA(*l*,r)方法的浮动模式。和 PTA(*l*,r)不同，SFFS/SBFS 不限定前行/后退的步数，SFFS 和 SBFS 方法可以无限制的回溯，只要回溯可以找到笔目前的特征更好的特征。SFFS 和 SBFS 的不同之处在于算法的搜索空间的起点不同。SFFS 从空集开始，而 SBFS 从全集开始
BAB, BAB^{+} BAB^{++} 等	分支界限法及其扩展系列，一般需要评价函数单调，可以找到最优解
GA	遗传算法不能保证找到最优解但是搜索空间较大，速度较快，不易陷入局部极值
SA	模拟退火搜索算法，计算量较大，初始温度以及每个温度值下的迭代次数难以设定

表2-2 典型特征选择搜索算法的时间复杂度

搜索算法	时间复杂度 [a]	目标类型 [b]	搜索类型 [c]
SFS，SBS	$\Theta(n^2)$	A	S
GSFS(g) GSBS(g)	$\Theta(n^{g+1})$	A	S
PTA(l,r)	$\Theta(n^2)$	A	S
GPTA(l,r)	$\Theta(n^{\max(l+1,\ r+1)})$	A	S
SFFS，SBFS	$O(2^n)$	A	S
BAB，BAB$^+$，BAB^{++}	$O(2^n)$	A	S
GA	$\Theta(l)\ \Theta(n)$ [d]	C	P

[a] Θ(·)表示对时间复杂度的较精确估计，O(·) 表示时间复杂度的上界

[b] 目标类型：A—寻找最优的固定大小的特征子集； B—满足给定条件下的最小特征子集；C—寻找综合考虑特征集大小和评价函数的最优子集

[c] 搜索类型：S—顺序的 P—并行的

[d] 其中 l 为遗传代数，n 为初始特征维数

2.2.3.4 评价标准

特征评估函数和特征相关性分析关系密切，过滤式特征选择中的特征评估就是要评价特征或者特征子集和目标函数的相关性。下面首先给出有关特征相关性分析的概念，然后具体介绍并分析比较几种特征评估方法。

1.相关分析的概念

人类在面对多个特征时，可以迅速定位到那些和目标函数相关的特征而忽略其他无关或不重要的特征。在机器学习中，要选择和目标函数相关的特征，首先必需对特征相关性分析进行定义。John 等以及 Blum 等对特征的相关性分析进行了详细阐述[John 1994][Blum 1997]。下面给出几种典型的定义并对其进行分析。在以下定义中，A 和 B 表示样本集中的样本，I 表示实例空间，S 表示样本集合，C 表示目标函数，C(A)表示样本 A 所属的类别（或称目标函数的值），$X_i(A)$表示样本 A 的特征 i 的值，$R(X_k, c)$表示 X_k 和 c 相关。

- **定义 1** 特征和目标函数相关：如果实例空间内存在样本 A 和 B，A 和 B 仅有属性 X_i 不同而 C(A)≠C(B)，则称特征 X_i 和目标函数 C 相关。

$$\text{if } \exists(A,B)\in I\ \ \forall\ i\neq k\ X_i(A)=X_i(B)\ X_k(A)\neq X_k(B)\ \ c(A)\neq c(B)\ \text{then } r(X_k,c)$$

这个定义贡献在于首次从理论上给出一个严格定义，但实际应用中由于训练数据规模有限，很多情况下根本不存在只有一个属性值不同的样本对，因而无法根据这一定义确认相关特征。其次，如果存在两个特征冗余但和目标相关的情况，这个定义会将这两个特征均视为无关特征。

为了弥补上述定义的不足，John 等给出了关于样本集的强/弱相关的定义[John 1994]，该定义针对具体的数据集而不是全体实例空间。

- **定义 2** 关于样本集强相关和弱相关：如果在样本集合 S 中存在样本 A 和 B，仅有属性 X_i 和类别不同，则说特征 X_i 对于样本集强相关。类似地，如果在舍弃一些特征后，特征 X_i 变为强相关特征，则称特征 X_i 关于样本集弱相关。

  if $\exists(A,B)\in S\ \forall\ i\neq k\ X_i(A)=X_i(B)\ X_k(A)\neq X_k(B)\ \ c(A)\neq c(B)$ then $r(X_k,S)$

  or if $\exists(A,B)\in S\ \ p(\forall\ i\neq k\ X_i(A)=X_i(B)\ ,X_k(A)\neq X_k(B)\ ,c(A)\neq c(B))\neq 0$ then $r(X_k,S)$

关于样本集强相关的定义和定义 1 类似，只是用“样本集”代替了定义 1 中的“实例空间”，关于样本集弱相关定义扩充了原有的相关性定义，在实际应用中，可能某特征并不决定目标函数，但是强相关特征去掉时，该特征就起决定作用。

以上定义都是独立于学习算法且从单个特征和目标函数或者样本集的关系进行定义。但单个特征和目标函数或者样本集相关，并不能保证该特征就会对特定的学习算法有用，也不能保证多个相关的特征组成的集合就会使学习算法取得好的性能。Caruana 等考虑了特征和算法性能间的关系，给出了关于算法增益性相关的定义[Caruana 1994b]。

- **定义 3** 关于算法增益性相关： 给定一个数据集 S，一个学习算法 L，一个特征集合 F，如果该算法 L 使用$\{X_i\}\cup F$所得分类的准确率比使用 F 的准确率高，则特征 X_i 对于算法 L 相对于 F 是增益性相关的。

2.具体特征评估方法

实际应用中很少直接应用上述相关分析的概念进行特征选择，但关于目标函数或样本集相关的概念是过滤式特征评估标准的理论基础，而关于算法增益

性相关的概念可以看作 Wrapper 特征选择的一个理论诠释。在相关性分析的基础上，研究者提出各种更具有操作性的特征评估标准，特征的评价标准大致可分为：距离度量，不一致度，基于信息熵的评估标准，Relief 评估以及和学习算法性能相关的评估[Breiman 1984][John 1994][Blum 1997][Pfahringer 1995][Liu 1998b][Lee 2001][Dash 2003]。和学习算法性能相关的评估标准又分为：直接用学习算法性能作为特征的评价标准以及间接应用学习算法性能的评估标准[Weston 2000][范劲松 2001]。下面将分别予以介绍。

- **距离度量**

距离度量又包括类间距离度量和概率距离度量，类间距离度量中有欧式距离，马氏距离（Mahalanobis Distance）等。概率距离度量有 Bhattacharyya，Kolmogorov，Kullbak-Liebler 等。类似的还有概率相关度量(Probablistiv Dependent Measure)，由于概率相关度量和概率距离度量很相近，我们将其归到了一起。表 2-3 给出了部分距离公式。

表2-3 距离度量（选自[Doak 1992]）

类间距离	
名称	公式
欧式距离	$J(X_i, X_j, F) = \sqrt{\sum_{l \in F} (x_{i,l} - x_{j,l})^2}$
马氏距离	$J(F) = (u_1 - u_2)^T \sum^{-1} (u_1 - u_2)$
概率距离	
名称	公式
Bhattacharyya	$J(F) = -\log \sqrt{P(F \mid C_1) P(F \mid C_2)} dF$
Kullback-Liebler	$J(F) = \int [P(F \mid C_1) - P(F \mid C_2)] \log \frac{P(F \mid C_1)}{P(F \mid C_2)} dF$
Kolmogorov	$J(F) = \int \mid P(F \mid C_1) - P(F \mid C_2) \mid dF$

- **不一致度**

不一致度是 Almuallim 等在 FOCUS 特征选择算法中提出的特征评估标准

[Almuallim 1991]，并被广泛应用[Liu 1998a][Huang 1999] [Dash 2003]。

定义各特征的一种取值组合为一种模式（不包括类别标记），特征子集 F 包含|F|个特征 f_1，f_2，…，$f_{|F|}$，分别对应 $n_{f_1}, n_{f_2}, \ldots, n_{f_{|F|}}$ 种取值，则最多存在的模式为 $n_{f_1} * n_{f_2} * \ldots * n_{f_{|F|}}$ 种。不一致度的定义和计算如下：

（1） 不一致定义：某种模式下，如果两个样本除了类别外其余特征值都相同，则说这个模式不一致。如：样本(0 1 1)和(0 1 0)的前两个特征是一样的(0 1)，而类别却不同(分别为 1 和 0)。

（2） 某种模式下的不一致数目：在这种模式下所有样本个数减去不同类别中最多的样本个数。例如：在模式 p 下有 n_p 个样本，在他们当中 c_1 个属于类别 1，c_2 属于类别 2，c_3 个属于类别 3，满足 $c_1+c_2+c_3=n_p$;如果 $c_3=\max(c_1,c_2,c_3)$，则不一致数目为 $u_p=n_p-c_3$。需要说明的是，样本集的大小 $n=\Sigma_p n_p$。

（3） 特征子集 F 下的不一致度为 $U=(\Sigma_p u_p)/n$。

不一致度适用于离散特征，对连续特征，需要首先进行离散化。而且由于采用不一致度评估的特征选择算法需要考虑各种特征子集组合下，各种特征的值的组合形式，因此对于高维数据，其计算代价很高。另外不一致度的度量倾向于选择值变化较大的特征，如病人的身份证号码和病症无关，但不一致度评估则很可能会将身份证号码确认为相关特征。

- **基于信息熵的评估标准**

信息论中的熵是信息量的评估标准，信息熵的直观解释是描述信息所需要的二进制位数。对于离散的随机变量 x，Shannon 熵定义为：

$$H(x) = -\sum_{j=0}^{N-1} p(x_j)\log_2 p(x_j)$$

其中变量 x 的取值空间为$\{x_0, x_1, x_2, \ldots x_{N-1}\}$。基于信息熵的有代表性的评估标准有：信息增益(Information Gain)、最小描述长度(Minimum description

length)、互信息（Mutual Information）和关联度(Correlation)等。

信息增益是 Quinlan 在决策树算法中用来选择特征作为树的分支结点时采用的评价标准[Quinlan, 1993]。其思想是考察类别的先验熵和已知属性 X 的值 x 后的后验熵之间的差，并以之表示该属性所提供的类别可分性的信息。具体计算公式如下：

$$IG(X) = H(C) - H(X \mid C)$$
$$= -\sum_{c \in C} P(c) \log_2 P(c) - \sum_{x \in X} \left(- P(x) \times \sum_{C} P(c \mid x) \log_2 P(c \mid x) \right)$$

互信息在特征选择中常用来评价特征和目标属性间的紧密程度[Hamming 1986][Battiti 1994a]。特征 X 和类别 C 之间的互信息为：

$$MI(X,C) = \sum_{x \in X, c \in C} P(x,c) \log_2 \frac{P(x,c)}{P(x)P(c)}$$

从信息论角度考虑，互信息的意义是为了学习 C（或者 X），需要多个位(bit)来咨询 X(或者 C)。也可以理解为已知 C（或者 X），X（或者 C）的不确定度减少的数量。

根据贝叶斯公式 P(X, C)=P(C|X)P(X)，可推得互信息和信息增益是一致的，并且互信息是对称的度量，即 有

$$MI(X,C) = H(C) - H(C \mid X) = H(X) - H(X \mid C)$$

最小描述长度(MDL) [Pfahringer 1995]评估标准是基于信息理论中的 Occam 的剃刀准则（Entities should not be multiplied beyond necessity）提出的，该标准争取用最小的编码长度来对所有的信息进行编码，编码的长度由模型的编码长度和给定模型后数据的编码长度两部分组成，而具体模型以及给定模型后数据的编码长度则由相关信息熵表示。MDL 评估标准只适用于离散型特征，Domingos 指出对 Occam 剃刀准则的理解在理论上和实际应用中都不是绝对正确的[Domingos 1999]。MDL 评估标准和目标函数之间的关系不是直观的，其性能需要进一步进行实验验证和理论分析。

信息增益，最小描述长度，Mutual Informtion，以及类似的评价标准如Gini Index[Breiman 1984]以及 Lee 提出的模糊熵度量[Lee 2001]都假定特征间彼此独立，因此不适用于特征相关性较强的领域，且只适用于离散型特征，其优点是实现简单，适用于大规模数据集，具有较强的理论基础。

Hall 提出的 Correlation 评估给出了一种既考虑了特征和目标函数的相关度，也考虑特征之间相关度的特征子集评估标准[Hall 2000]。其准则为：好的特征子集含有的特征应该和目标函数有很高的相关度，但是和集合内其余特征不相关。但出于计算量限制，只能考虑两两特征之间的相关度，更复杂的特征之间的相互作用则不考虑。Correlation 评估将在下章加以介绍。

- **Relief 评估**

Kira 等 1992 年提出的 Relief 是公认的性能较好的特征评估方法[Kira 1992][Kononenko 1994][Yu 2003]，其特征评价借鉴了最近邻学习算法的思想，其核心思想是：一个好的特征应该使最近邻的同类样本之间特征值相同或相近，而使最近邻的不同类样本之间值不同或者差别很大。Relief 系列算法将在后面介绍典型算法时予以详细介绍。

- **和学习算法性能相关的评估标准**

和学习算法性能相关的评估标准又分为直接和间接利用学习算法性能作为特征评估标准两类。后者例如基于支持向量机的算法提出的最大正边缘距离，最大负边缘距离等特征评估标准等[Kohavi, 1997][Weston, 2000][范劲松, 2001][Kudo, 2000]。

3.特征选择的目标类型

特征选择通常只是学习系统的中间阶段，最终选择出来的特征子集对应的样本集要进行分类或者回归分析等处理，因此特征选择的最终结果要由机器学习算法评判。但实际应用中，特征选择研究者主要从以下几种角度来研究：

- 优化某目标函数（如一致度，在测试集上性能最优）的特征子集
- 固定特征子集大小前提下，寻找最优化目标函数（如一致度，在测试集上性能最优）的特征子集

- 设定优化目标的阈值，寻找满足阈值的最小特征子集
- 回答哪个特征有用，哪个特征无用
- 根据特征对目标函数的相关度或者特征对学习算法的有用程度进行排序

一般说来，第二类和第三类的目标中特征子集的大小和阈值是人为确定的，而且不容易确定。实际分类问题中，通常目的是寻找使分类准确率高同时特征集合尽量小的特征子集。实际应用中应根据需要确定选用何种研究方式。

2.3 典型特征选择算法介绍

下面将介绍几种典型的特征选择算法，其中对于下章将要用到的 Relief 系列算法和遗传算法用于特征选择的算法将较详细介绍。

2.3.1 Filter 类

在 Filter 类算法中，本文主要介绍 Focus 算法和 Relief 系列算法。

2.3.1.1 Focus

[Almuallim 1991]提出的 Focus 算法是机器学习领域最早的特征选择算法之一。Focus 以前面介绍的不一致度为评估标准，以空集状态为搜索起点，采用宽度优先的搜索算法进行搜索，直到找到满足不一致度为 0 的最小的特征子集。举例来说，如果有 3 个特征，则搜索的根结点为(0, 0, 0)，其子结点为(0, 0, 1)，(0, 1,0)和(1, 0,0)，其中’0’表示对应特征不在特征子集中，’1’表示对应特征在特征子集中。Focus 算法对有噪声的数据集很敏感，并且由于采用穷尽式搜索因而在某些情况下运行效率较低，限制了该算法在实际中的应用。后来[Almuallim 1992]提出的 Focus2 通过将特征集合分为不相交的特征子集使运行效率明显提高。[Dash 2003]中提到的 FocusM 算法在一定程度上解决了 Focus 和 Focus2 对噪声敏感的问题，但当应用于有噪声且特征维数较高的数据时，其运行效率以及算法性能都不理想，且该算法只能应用于特征为离散型的数据集。

图 2-8 给出了 Focus 算法，其中 inConCal(S', D)计算在数据集 D 上以及给定特征子集 S'后的不一致度。

Focus 算法

输入：数据集 D，特征全集 S

输出：一致的特征子集

1. For $size$=0 to $|S|$
2. for subsets S' with $|S'| = size$
3. if inConCal(S', D) = 0
4. return S'

图2-8 Focus 算法

2.3.1.2 Relief 系列算法

Relief 为一系列算法，包括最早提出的 Relief 以及后来拓展的 ReliefF 和 RReliefF，其中 RReliefF 算法是针对目标属性为连续值的回归问题提出的。本文没有涉及到回归问题，下面仅介绍后面应用较多的 Relief 和 ReliefF 算法。在以后章节中，如无特殊说明，Relief 系列算法笼统用 Relief 来表示。

1．Relief 算法

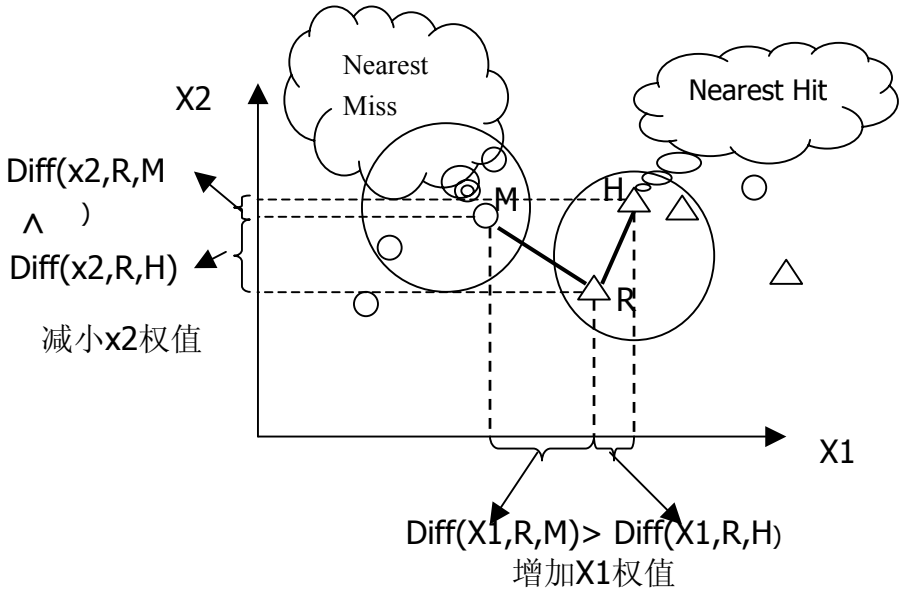

图2-9 Relief 系列方法示意图

Relief 评估最早由[Kira 1992a][Kira 1992b]提出，当时局限于解决两类数据的分类问题。Relief 系列算法的要点是根据特征对近距离样本的区分能力来评

估特征，其核心思想为：好的特征应该使同类的样本接近，而使不同类的样本之间远离。图 2-9 给出了二维空间（每维对应一个特征）中 Relief 算法的几何图示，其中圆和三角分别表示两类样本。算法从训练集 D 中随机选择一样 R，然后从和其同类的样本中寻找最近邻样本 H，称为 NearestHit，从和其不同类的样本中寻找最近邻样本 M，称为 NearestMiss，然后对于每维特征，如果 R 和 H 在其上的距离小于 R 和 M 上的距离，例如图 2-9 上的 Diff(x2,R,M)大于 Diff(x2,R,H)，则说明此维特征对区分同类和不同类的最近邻是有益的，则增加该特征权值；反之，如果 R 和 H 在其上的距离大于 R 和 M 上的距离，例如图 2-9 上的 Diff(x1,R,H)大于 Diff(x1,R,M)，说明特征 x1 对区分同类和不同类的最近邻是起反作用，则降低该特征权值，权值更新如公式(2-2)所示。重复以上过程 m 次，最后得到各特征的平均权值。特征的权值越大，表示该特征的分类能力越强，反之，表示该权值分类能力越弱。具体算法流程图如图 2-10 所示。

输入：训练数据集 D，迭代次数 m

输出：如为特征评估，则输出 W[A]，如为特征选择，则输出那些 W[A]大于指定阈值的特征。

1.置所有特征权值为 0，W[A]=0

2.for i:=1 to m do begin

1)随机选择一个样本 R;

2)从同类样本集中找到 R 的最近邻样本 H，从不同类样本集中找到最近邻样本 M;

3)for A:=1 to N do

W[A]:=W[A]-diff(A,R,H)/m+diff(A,R,M)/m; 公式(2-2)

3.End;

图2-10 Relief 算法

2．ReliefF 算法

Relief 算法提出时仅限于处理类别数为两类的数据的分类问题，并且没有对数据缺失的情况进行处理，后来 Kononenko[Kononenko 1994]扩展了 Relief

算法得到了 ReliefF 算法，ReliefF 可以解决多类问题以及回归问题，并补充了对数据缺失的情况的处理办法。本文仅涉及到分类问题，因此仅描述 Relief 在多类问题上的扩展算法。ReliefF 算法在处理多类问题时，不是从所有不同类样本集合中统一选择最近邻样本，而是从每个不同类别的样本集合中选择最近邻样本，并且不是选择一个最近邻样本，而是选择 k 个最近邻样本，如图 2-9 中圆圈分别圈中两个最近邻。（可类比于“最近邻”和“k 近邻”）。更新特征权值的公式也变为公式(2-3)。

ReliefF 算法

输入： /*D－训练样本集, N－特征数目, m－迭代次数, k－最近邻样本个数

输出: 特征评估向量 W

1)set all weighst W[A]:=0.0;

2) for i:=1 to m do begin

3) 从 D 中随机选择样本 R;

4) 找出 R 的 k 个 *nearHits* H_j;

5) for each class $C \neq$ class(R)

6) 从类 C 中找出 k nearest misses $M_j(C)$;

7) for A:=1 to N do

$$W[\mathrm{A}] = W[\mathrm{A}] - \sum_{j=1}^{k} diff(\mathrm{A}, R, H_j) \Big/ (mk) + \sum_{C \neq class(R)} \left[\frac{p(C)}{1 - p(class(R))} \sum_{j=1}^{k} diff(A, R, M_j(C)) \right] \Big/ (mk) \qquad \text{公式(2-3)}$$

8) end.

图2-11 ReliefF 算法

ReliefF 程序伪码如图 2-11 所示，ReliefF 每次随机从训练样本集中取出一个样本 R，然后从和 R 同类的样本集中找出样本 R 的 k 个近邻样本(nearHits), 从每个 R 的不同类的类样本中均找出 k 个近邻样本(nearMisses)，然后按照公式(2-3)更新每个特征的权值。以上过程重复 m 次，最后得到每个特征的权值，m 和 k 为人为设定的参数。

在公式(2-3)中，*diff(A,R1,R2)*表示样本 R_1 和样本 R_2 在特征 A 上的差，$M_j(C)$ 表示类 C 中的第 j 个最近邻样本。ReliefF 算法中需要计算样本间距离，两个样本之间的距离定义如下：

$$d(R_1,R_2)=\sum_{A=1}^{N} diff(A,R_1,R_2) \qquad \text{公式(2-4)}$$

其中

$$diff(A,R_1,R_2)=\begin{cases}|R_1[A]-R_2[A]|\ ;\text{if A is continuous}\\ 0\ ;\ \text{if A is discrete and } R_1[A]=R_2[A]\\ 1\ ;\ \text{if A is discrete and } R_1[A]=R_2[A]\end{cases} \qquad \text{公式(2-5)}$$

为了保证所有 *diff*(A,R_1,R_2)都在[0,1]之间，我们对所有连续属性进行了归一化。对于有数据缺失的情况，如果缺失属性为连续型，则用训练集中该特征的平均值代替；如果缺失属性为离散型，则用训练集中该属性出现频率最高的值代替。

Kononenko，Marko 等根据统计理论对 ReliefF 算法进行了分析[Kononenko 1994][Marko 2001]，得出当样本集充分大时，ReliefF 评估满足以下逼近：

$$\begin{aligned}W[\text{特征 i}] = {} & P(\text{特征 i 取值不同}\mid\text{不同类的最近邻样本})\\ & -P(\text{特征 i 取值不同}\mid\text{同类的最近邻样本})\end{aligned} \qquad \text{公式(2-6)}$$

其中 W[特征 i]表示特征 i 的权值，*P*(特征 i 取值不同｜不同类的最近邻样本)表示不同类的最近邻样本之间特征 i 取值不同的概率，*P*(特征 i 取值不同｜同类的最近邻样本)则表示同类的最近邻样本之间特征 i 取值不同的概率。

Relief 系列 算法的优点是：运行效率高，对数据类型没有限制，对特征间的关系不敏感（相对很多特征评估方法的特征间独立的假设）。Relief 算法的缺点是：和很多特征评估算法（如信息增益，Gini-Index 等）一样，Relief 算法不能去除冗余特征，算法会赋予所有和类别相关性高的特征较高的权值，而不管该特征是否和其余特征冗余[Kira 1992a][Kira 1992b][Kononenko 1994][Yu 2003]。

2.3.2 Wrapper 类—GA-Wrapper

遗传算法最早由 Holland [Holland 1975]提出，它是一种进化学习方法，通

过迭代搜索技术来实现。与传统的优化过程不同，遗传算法从多个初始解开始，经过演化，同时到达一个近似的最优解[Goldberg 1989]。这一过程是通过模拟自然界生物的选择和繁衍而进行随机的信息交换来完成的。待解决的问题通过编码由多个基因构成的染色体来描述。每个染色体都有一个适应值，由适应函数评价得到。这个适应函数是用来判断染色体的好坏及其生存概率的，适应值越高，则该染色体越有可能被选择进行繁殖。在遗传算法中，初始种群由较少的个体组成，然后反复地根据适应值繁衍出优化的后代。

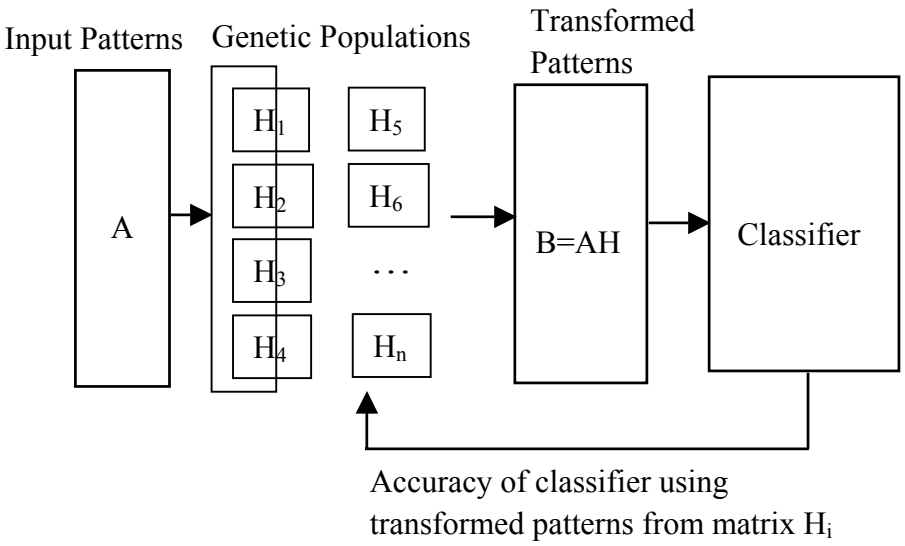

图2-12 GA-Wrapper 算法

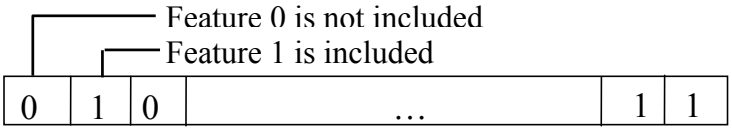

图2-13 遗传算法种群中的个体

最早采用遗传算法进行特征选择是 Siedlecki 和 Sklansky[Siedlecki 1988][Siedlecki 1989]。[Maria 1999][Yang 1998][Vafaie 1992]等采用遗传算法进行特征选择，取得了较好的结果。遗传算法对问题依赖性小，搜索能力强，适合大规模复杂问题的优化。但据 Jain[Jain 1997]实验，遗传算法容易过早收敛，需要就此采取措施，而且当应用于大规模数据，如果采用遗传算法的

Wrapper 式结构，运行效率较低[Martin-Bautista 1999]。

图 2-12 为 GA-Wrapper 算法的示意图。遗传算法的个体形式如图 2-13 所示，遗传算法的个体是长度为 n 的二进制串（n 为特征数目），第 i 位为 0 表示不选择特征 i，为 1 表示选择特征 i。这样遗传算法的每个个体表示一种特征选择方案。分类器在训练集上的分类准确率作为计算个体适配值的主要考虑因素，另一个影响适配值的参数为选择特征子集的大小。遗传算法通过选择、交叉、变异等算子得到下一代，如此迭代下去，直到满足终止条件而结束，然后输出适配值最高的个体作为最终的特征选择方案。

2.3.3 Filter 和 Wrapper 组合式算法

由前文可知，Filter 和 Wrapper 是两种互补的模式，因此一个很自然的想法就是：能否想到一种综合 Filter 和 Wrapper 优点的组合式特征选择算法。90 年代末期以来，有一些研究者对组合式特征选择开展研究。

Huang[Huang 1999]提出了一种两阶段组合式特征选择算法。在第一阶段，Huang 采用遗传算法搜索不一致度较低的特征子集；第二阶段，Huang 采用神经网络剪枝法进一步去除对神经网络来说无用的特征。由于不一致度的计算只能在离散型特征上进行，因此必须首先将所有连续特征离散化，这又增添了一道不确定因素。而且不一致度的计算，遗传算法和神经网络训练的运算效率较低，因此该两阶段组合式算法未解决 Wrapper 模式特征选择效率低的问题。

[Das 2001]中提出的组合算法的前提是假定数据集中特征对不同分类算法的作用是一致的，采用了前向搜索算法，特征评估采用的是信息增益，和一般算法不同的是，它利用 Boosting[Freund 1995]算法引导 Filter 式特征选择：Boosting 算法每运行一次，即对样本进行加权，然后在加权后的样本上计算每个特征的信息增益，选择不曾被选择的特征中信息增益最高的特征，随后在已选特征构成的特征子集上训练得到新的分类器，然后对样本进行加权，计算特征的信息增益，选择信息增益最高的特征，如此重复下去，直到选择了指定数目的特征，或者分类器的训练准确率开始下降。该组合算法存在问题：一、采用的一层决策树作为基分类器只能解决两类问题；二、不管后续分类器是何类

型，仅采用决策树指导特征选择，使其不能利用 Wrapper 的特性；三、该组合算法只适合采用顺序前向搜索算法，而顺序前向搜索算法很容易陷入局部极值。

2.4 特征选择的主要研究方向和应注意的问题

虽然特征选择领域的研究近年来取得了很大进展，但由于特征选择问题和实际对象的复杂性，还需要进行大量深入的研究，目前研究热点和需要解决的问题包括以下方面。

2.4.1 研究过滤式和 Wrapper 相结合的组合式特征选择算法

Filter 方法和 Wrapper 方法结合起来以得到综合两者优点的特征选择方法是一个很有潜力的研究方向。目前关于此方面研究仍较少，还需要借鉴样本选择以及集成学习等思想进行更系统的分析和研究。

2.4.2 研究特征选择和学习算法之间的关系

过滤式特征选择算法认为特征和后边的学习算法是独立的，即一个好的特征子集在任何学习算法上都应该取得较好的效果；Wrapper 算法认为特征选择不仅和数据集以及目标函数有关，还和后边的学习算法的性能密切相关，因此要用学习算法本身的性能作为选择特征子集的评估标准。目前对特征选择和学习算法之间的关系仍无定论，Kohavi 用几个实例证明过滤式算法的不足[Kohavi 1997]，但是 Das 认为 Kohavi 所举的都是很特殊的假想例子[Das 2001]，在实际情况中一个好的特征子集能使任何学习算法得到好的结果。进一步深入研究特征选择和学习算法之间的关系对于指导特征选择算法的构造有重要意义。

2.4.3 非监督式学习的特征选择问题

机器学习中的特征选择方面的研究主要集中于监督式学习，在非监督式学习方面的研究较少，但方向的研究是一个很有挑战性的研究方向。

2.4.4 研究特征选择需要注意的问题

- 特征和目标函数相关 VS 特征对学习算法有用

一个和目标函数相关的特征也许无益于提高学习算法的性能，比如冗余特征。反之，一个有助于提高学习算法性能的特征也许和目标函数是无关的，如神经网络中的常数输入项[Kohavi 1997]。

- 特征和目标函数相关 VS 特征决定目标函数

特征也许和目标函数相关但并不决定目标函数，如可以设计某特征和目标函数以 0.7 的概率相关，但目标函数由其余的特征决定。

2.5 本章小结

本章系统介绍了特征选择的理论，包括特征选择的定义，特征选择和学习算法的结合关系，特征选择的搜索算法和评估问题，特征选择的经典算法，以及特征选择的主要研究方向等。虽然综述没有穷尽特征选择的所有方面，但涵盖了特征选择领域的主要方面及经典方法。其中有很多概念和方法对本文后续工作很重要并将要加以深入研究和发展，例如：Relief 算法是本文特征选择研究的基础，这是由于该算法被公认为在辨别特征和类别相关性方面具有良好性能；SFS 和 SBS 搜索算法被本文用作比较的基准，因为这两种都是经典的特征选择搜索算法，其中 SBS 算法又被我们用作去除冗余特征的搜索算法；Correlation 分析在本文被用以去除冗余特征；遗传算法的 Wrapper 式特征选择在下一章给予了重要扩展和改进；组合式特征选择则是本文特征选择方面研究的重点；特征选择方面的研究又是本文第四章和第五章中基于特征选择的集成学习研究 的基础。

大规模数据的特征选择研究是对经典特征选择算法的挑战，本文在特征选择方面的研究目标即为研究适用于大规模数据的，能够综合 Filter 和 Wrapper 方法优点的特征选择算法，这些构成了下一章的主要内容。

第三章 基于 Relief 的组合式特征选择算法

3.1 引言

上一章对机器学习中的特征选择进行了概要综述，已经介绍了根据评估过程是否引入后续学习算法的性能，特征选择算法可分为 Filter 和 Wrapper 两大类。并较详细地介绍了 Filter 类中的 Relief 算法和 Wrapper 类中的利用遗传算法进行特征选择的 GA-Wrapper 算法。Relief 是公认较好的特征评估算法，可以很方便的以 Filter 方式进行特征选择；遗传算法的 Wrapper 式特征选择是研究较多的效果较好的特征选择算法。本章的工作主要是围绕 Relief 和 GA-Wrapper 算法展开的。

本章首先对 Relief 评估算法在手写体汉字数据集上的评估结果进行了分析，目的是考察 Relief 算法的评估效果。Relief 算法的评估效果是本章所有算法提出的出发点。并且本文不断用提出算法的性能来校验 Relief 算法的评估。

为利用 Relief 算法的优点，并克服 Relief 不能去除冗余特征的缺点，我们提出并实验了若干算法，将在本章一一介绍。这里，我们首先总体介绍一下各算法提出的思路。

1. Relief-Wrapper 算法。根据 Relief 评估可得到按相关度从大到小排序的特征序列，在假定 Relief 评估合理的前提下，要进行特征选择，只需决定删除多少排序靠后的特征即可，Relief-Wrapper 采用分类准确率作为特征子集的评估准则，进而决定删除的特征数目。

2. PCA-Relief-Wrapper 算法。主成分分析 (principle component analysis) 是一种数学变换的方法, 它可把给定的一组相关变量通过线性变换转成另一组不相关的变量。PCA-Relief-Wrapper 算法首先通过 PCA 将特征之间去相关，然后再运行 Relief-Wrapper 进行特征选择。

3. ReCorre 算法。ReCorre 为两阶段特征选择算法：第一阶段用 Relief 去除无关特征；第二阶段采用相关分析方法分析特征间的冗余度，并据此去除冗余特征。

4. ReSBSW 算法。ReSBSW 为两阶段特征选择算法：第一阶段用 Relief 去除无关特征；第二阶段采用顺序后向搜索方法，以分类器本身的分类准确率为评估标准，依次考察去除特征对分类器准确率的影响，藉此去除冗余特征。

5. Relief-GA-Wrapper 算法。根据前面各算法实验分析，发现 Relief 评估去除无关特征比例不大，且有时会将相关特征错认为无关特征去除。Relief-GA-Wrapper 用 Relief 评估引导遗传算法种群初始化，采用分类器本身的准确率评估特征子集，用遗传算法进行特征选择。

3.2 Relief 特征评估方法对手写体汉字数据集的特征评估

本节考察 Relief 算法对手写体汉字特征的评估结果，并根据所掌握的先验知识对 Relief 算法的评估性能进行验证。

3.2.1 手写体汉字数据库

我们所使用的手写体特征是基于模糊方向线素的 256 维向量[马少平 1997][李凌 2000]，向量的每维分量（一维分量即一个特征）为一个 0-255 的整数。特征的抽取是通过将汉字的点阵图划分为 8×8 的方格后分别统计每个方格中横、竖、撇、捺这四种不同笔划（不同方向）的个数得到的（具体说明请参见附录）。[李凌 2000]采用统计分析方法，发现此 256 特征具有复杂的相关性和不规则性。我们采用的脱机手写体汉字数据库共 800 类，每类 100 个样本。从每类中随机选取 50 个样本组成训练样本集，剩余的组成测试样本集。

本小节用 Relief 对手写体汉字特征进行评估，观察 Relief 评估和以前的统计分析结果[李凌 2000]是否符合。我们也通过实验考察了 Relief 算法对类别数目 C，迭代次数 m 以及最近邻样本个数 k 等参数的敏感程度。如无特别说明，最近邻样本的个数取 10，迭代次数为 10*C。

3.2.2 Relief 对汉字横竖撇捺评估

Relief 在 800 类汉字，每类 50 个样本的训练集上得到了横、竖、撇、捺四种笔划的特征权值。如图 3-1 所示，水平方向特征和竖直方向特征的权值比

撇和捺方向特征的权值大，即意味着横和竖笔划和类别的相关度大于撇和捺笔划和类别的相关度。从图 3-1 中，我们可以进一步看出四种笔划的具体分布情况，四种笔划的权值虽然不同，四条曲线的弯曲起伏也互不相同，但曲线上的“波谷”却普遍呈现出一定的周期性。仔细地观察发现曲线上有规律的“波谷”发生在 8n 处（n＝1,…,8）。这是由于汉字的特征是将汉字图像分成 8×8 来取的，那些分布在汉字图像左右两侧的特征（即附录所示的标号为 8n 的特征）一般值较小，其和类别的相关性也较小。Relief 对不同笔划评估的趋势通统计分析的结果是一致的[李凌 2000]，这也在一定程度上佐证了 Relief 评估的有效性。

3.2.3 手写体汉字类别数 C 对 Relief 评估的影响

如图 3-2 所示，总体说来，汉字的类别数对特征权值影响不大，但可以看出，随着类别数目的增多，特征权值评估的偏差逐渐增大。这可以从手写体汉字本身的复杂性角度来解释：当类别数目增多时，样本的分布更加复杂，不同类别的样本间分布交叉更多，使得特征和类别的相关性变得不那么明显，和类别数少的时候相比，特征的相关性发生变化，从实验结果看，特征的权值趋向于平均化、无序化，给汉字的特征选择增添了困难。

3.2.4 迭代次数 m 对 Relief 评估的影响

如图 3-3 所示，可见迭代次数取 1250，2500，5000，10000 和迭代次数为 250 时得到的评估差别不大，这显示了 Relief 一个突出的优点，即当样本规模较大时，迭代次数在远小于总样本数时，就可以取得较准确的评估结果，这使得 Relief 可以很容易地应用于大规模数据集。

3.2.5 最近邻样本个数 K 对 Relief 评估的影响

上面所列数据图示都是在 *K* 取 10 时得到的评估结果，为考察 *K* 对 Relief 评估的影响，我们做了 *K* 取 1 时的相关实验，*K* 取 1 是 Kira[Kira 1992a]最初提出 Relief 算法时采用的，*K* 取 10 则是 Knonenko[Knonenko 1994]建议设定的值，*K* 取 1 时得到的结果如图 3-4 和 3-5 所示，和 *K* 取 10 时的对应图 3-2 和 3-3 比较，可见当 *K* 取 1 时，评估结果随类别数目和迭代次数变化很大，说明此

时得到的评估很不稳定，和 K 取 10 时对比，我们可以得出 Relief 算法对最近邻样本个数 K 的选取较为敏感。而且，我们实验还发现，每类包含样本的个数对 Relief 评估也有较大影响，当我们将每类包含样本数减到 15 时，即使最近邻样本个数取 10，Relief 评估的结果也不稳定。

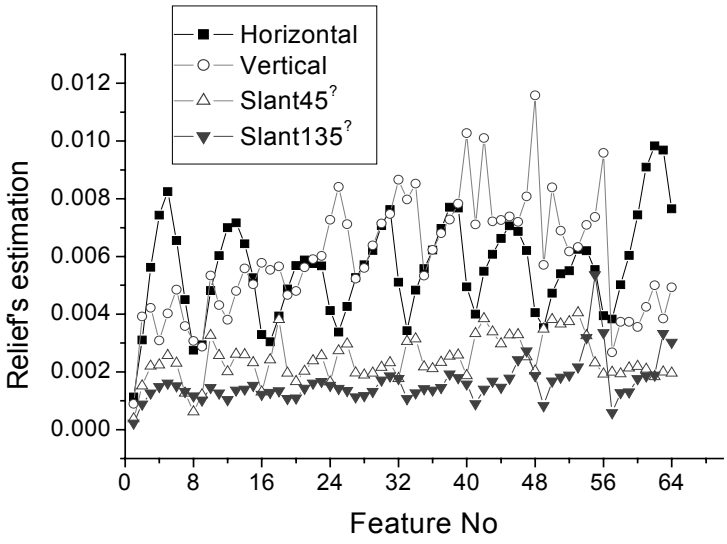

图3-1 Relief 分别对横、竖、撇、捺四种笔划的评估结果

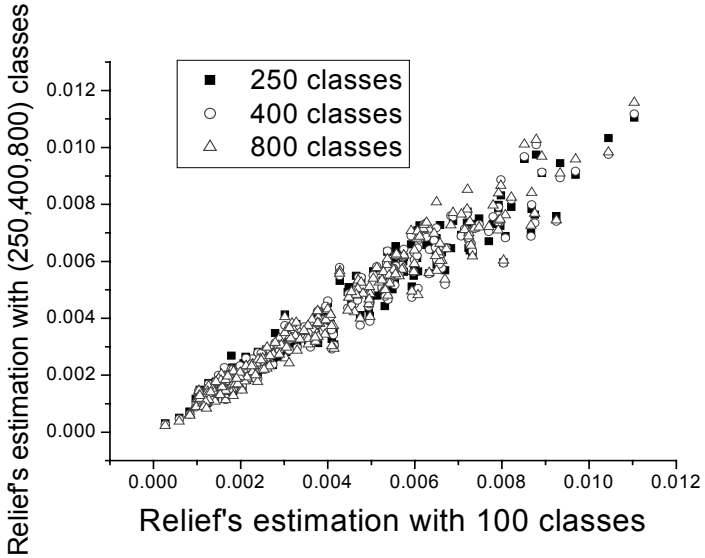

图3-2 类别数目 C 对 Relief 评估的影响

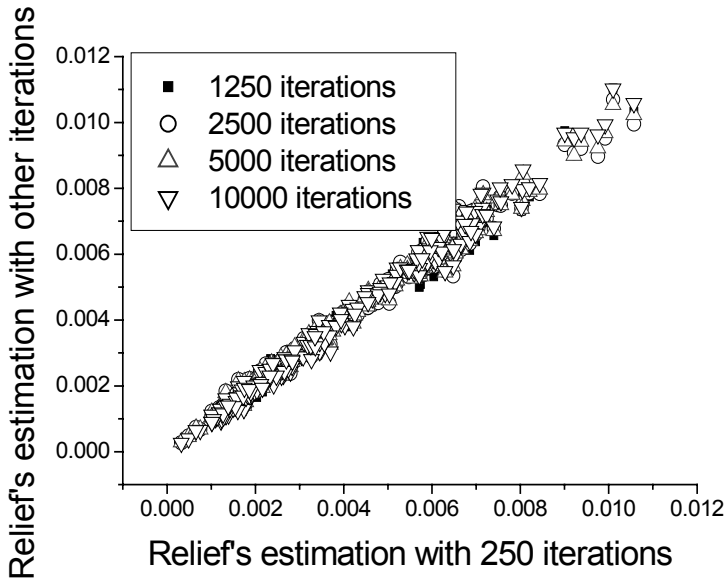

图3-3 迭代次数 m 对 Relief 评估的影响

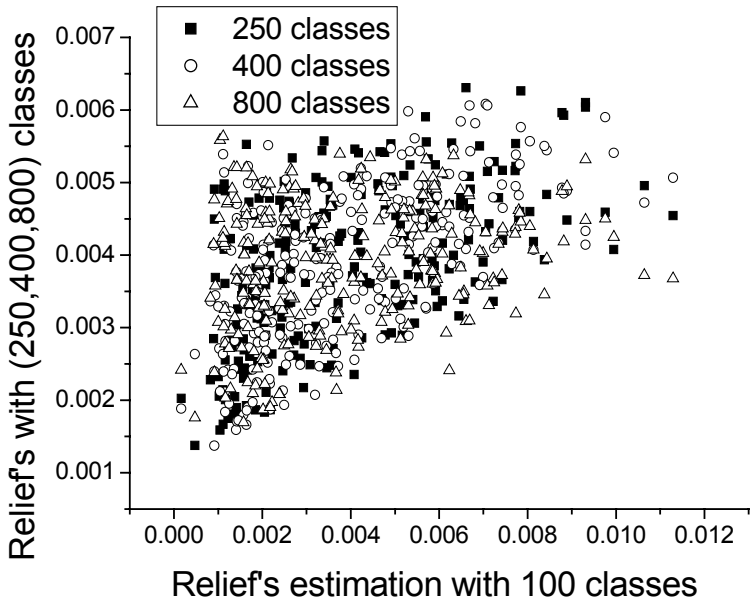

图3-4 Relief 中参数 *K* 取 1 时不同类别数目下的特征评估对比

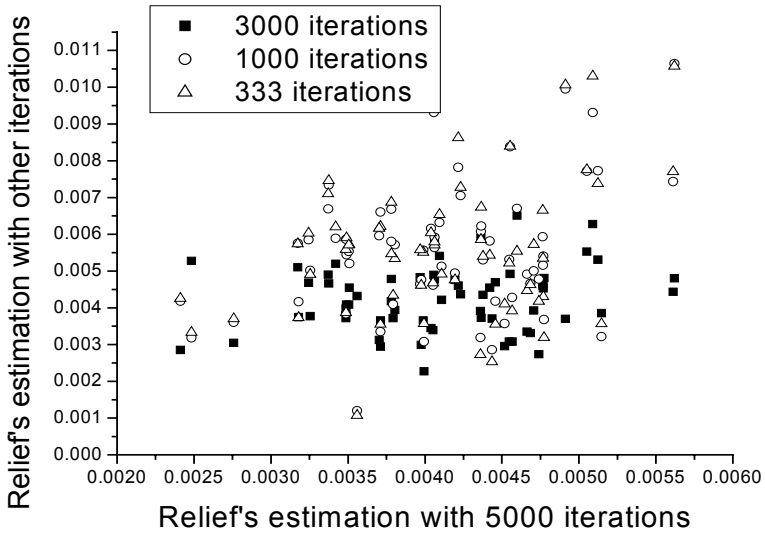

图3-5 Relief 中参数 *K* 取 1 时不同迭代次数下的特征评估对比

3.2.6 Relief 在手写体汉字上的评估总结

总结以上 Relief 在手写体汉字上的评估分析，可以得出如下结论：

- 相对而言，汉字特征中横、竖的重要性较大，而撇、捺较不重要，汉字边界的特征和类别的相关性一般较小。
- 当 k 取值较大，即取较多的最近邻样本时，算法的性能更稳定。迭代次数 m 在一定程度上代表 Relief 算法对样本空间的覆盖程度，m 和样本的分布有关，m 越大，评估的准确性越高
- Relief 评估和关于手写体汉字的先验知识以及统计分析结果一致，说明 Relief 评估较合理。

3.3 Relief-Wrapper 和 PCA-Relief-Wrapper 算法

3.3.1 Relief-Wrapper 算法

Relief 给出了对每个特征的评估—权值，但要进行特征选择，我们需要进一步决定如何进行特征选择。以前的研究者采用 Relief 进行特征选择通常选择那些权值大于 0 的特征组成特征子集[Kira 1992a][Yu 2003]，但是对于手写体

汉字数据库，实验发现所有特征的权值均大于 0，即理论上说，所有特征均和类别相关，图 3-6 为选择不同数目的排序靠前的特征得到的特征子集大小和对应的训练集准确率的关系，可见去除特征总是使得分类准确率下降。

在某些情况下，不但要求保证准确率，也要求一定运行效率，这时可以通过特征选择得到运行效率和准确率的较好折中：即再降维的前提下，尽量保证分类准确率。为此，我们设计了如公式(3-1)所示的评估函数，其中 F_s 表示特征子集的大小，A_s 表示采用 F_s 后的训练准确率，F_0 表示初始特征集的大小，A_0 表示未采用特征选择处理前得到的训练准确率。公式(3-1)要求训练准确率 A_s 相对于 A_0 的降低百分比小于 β（本文中 β 设为 0.02，表示训练准确率降低百分比不能超过 2%），否则将评估值置为一设定的最小值。公式(3-1)评估中涉及两项，前一项表示特征减少的百分比，后一项表示训练准确率降低的百分比，α 用来平衡这两项对评估函数的贡献，在本文中，我们设置 α 为 0.05，可以理解为分类准确率比特征大小的重要性高 20 倍。

$$f(F_s, A_s) = \begin{cases} \dfrac{F_0 - F_s}{F_0} * \alpha - \dfrac{A_0 - A_s}{A_0} & \text{if } \dfrac{A_0 - A_s}{A_0} < \beta \\ Min & \text{else} \end{cases} \qquad \text{公式(3-1)}$$

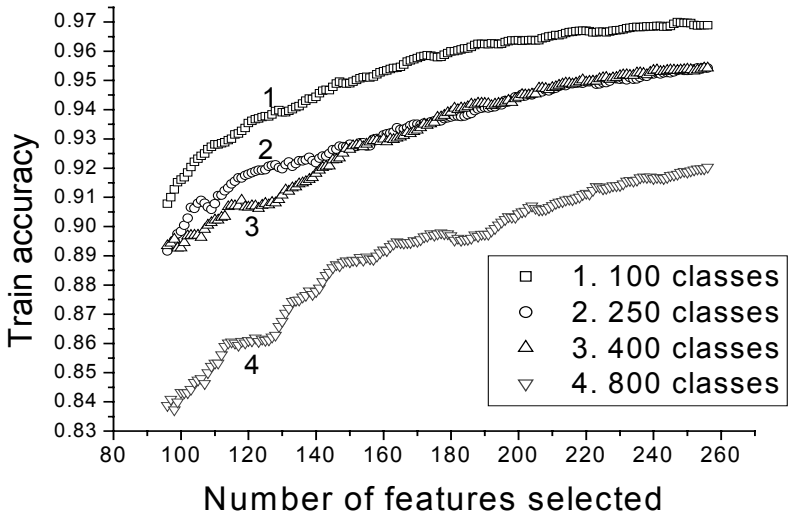

图3-6 特征子集大小和训练准确率

3.3.2 Relief-Wrapper 和 GA-Wrapper 在手写体汉字上的实验比较

我们在第二章介绍过遗传算法的 Wrapper 式特征选择算法 GA-Wrapper，下面我们将对 Relief-Wrapper 和 GA-Wrapper 在手写体汉字数据集上的性能进行实验比较。

我们分别在 100 类，200 类和 400 类的手写体汉字数据集上，实验了两种特征选择算法的性能。采用的分类器为最近邻中心法，最近邻中心分类器实现非常简单：即从训练集中算得各类的中心向量，计算待识别样本和各类的中心向量的距离，待识别样本的类别即距离最近的类中心向量所在的类。实验中设定遗传算法的迭代次数 g 为 50，种群大小 p 为 30。

表3-1 Relief-Wrapper 和 GA-Wrapper 在手写体汉字上选择特征子集的大小，训练准确率和测试准确率

	100			250			400		
	NoFs	RW	GW	NoFs	RW	GW	NoFs	RW	GW
F_s	256	188	152	256	201	149	256	206	139
A_{tr}	96.88	96.26	96.70	95.42	94.53	94.80	95.42	94.73	94.80
A_{te}	96.60	95.68	96.26	94.73	93.82	93.93	94.73	93.29	94.07

F_s—选择特征的数目 A_{tr}—训练准确率； A_{te}— 测试准确率；No F_s— 未经特征选择处理得到的有关性能； RW — Relief-Wrapper 算法；GW — GA-Wrapper 算法

表3-2 算法效率分析

算法名称	算法复杂度
Relief	$m \times n \times f$
Relief-Wrapper	$l \times f \times n \times c$
GA-Wrapper	$g \times p \times f \times n \times c$

c — 类别数，m — Relief 算法中的迭代次数，n—训练集中样本个数，f—初始特征集大小，l—Relief-Wrapper 中的迭代次数，g—遗传算法的迭代次数，p—遗传算法种群的个体数目， 一般 g× p> l

表 3-1 列出了 Relief-Wrapper 和 GA-Wrapper 在手写体汉字上的实验结果，可见 GA-Wrapper 算法无论在特征子集的大小还是对应的测试准确率上，都取得了更好的性能。表 3-2 列出了各算法的时间复杂度，对于特征数目为 256 的手写体汉字数据集（则 l 最大值为 256）， g×p 远大于 l，Relief-Wrapper

的时间复杂度中没有算入 Relief 的运算复杂度，因为 Relief 评估可以事先算出来给 Relief-Wrapper 使用，这样 GA-Wrapper 的运算要高于 Relief-Wrapper，但时间复杂度仍然是同一个量级的。

这个实验也说明 Relief 得到的特征评估对分类并不是最有效的，否则理论上 Relief-Wrapper 算法应该得到优于 GA-Wrapper 算法的性能。

3.3.3 PCA-Relief-Wrapper 算法

根据上节实验，可知 Relief-Wrapper 性能不如 GA-Wrapper。而以前研究者指出 Relief 的缺点是不能辨别冗余的特征[Kira 1992a]，尤其是对手写体汉字这种特征间相关度较高的情况，Relief-Wrapper 选取了权值较高的特征，而未考虑特征间冗余度的问题。主成分分析方法（Principle Component Analysis，PCA）是一种可以去掉特征之间的相关性的特征变换方法，因此设想经过 PCA 转换后得到新的特征结合再采用 Relief-Wrapper 方法进行特征选择，可能会得到更好的性能，这就是 PCA-Relief-Wrapper 方法的思路。

主成分分析 (principle component analysis，简称 PCA)，是一种数学变换的方法。常见的方法是线性 PCA，它可把给定的一组相关变量通过线性变换转成另一组不相关的变量，这些新的变量按照方差依次递减的顺序排列。经过此变换后，使得：1.特征之间线性无关(uncorrelated)，表现在变换后的协方差矩阵为对角阵；2.PCA 的变换矩阵为原数据协方差矩阵的特征向量，如果选择 m(m<n)个特征值最大的特征向量作为坐标轴，对于所有可能线性变换的 m 个特征来说，PCA 变换丢失的信息最少。此方法和与之相关的方法得到了广泛应用，如文献[Leonardis 2000] [Torrieri 2000]中的工作就是以此为基础，取得了很好的效果。

主成分分析的实现过程如下：

设原样本用矢量 X 描述 $X=[x_1, \ldots, x_n]$假设变换后用矢量 Y 描述 $Y=[y_1, \ldots, y_n]$，变换矩阵为$\Phi=[\phi_1, \ldots,\phi_n]$，则有

$$X = \sum_{i=1}^{n} y_i \phi_i \qquad \text{公式(3-2)}$$

根据线性无关的条件最后可以推得，Φ 的各组分 $\phi_1, \ldots, \phi_n$ 为 $\sum_X$ 的特征向量，$\sum_X$ 为 X 的协方差矩阵。

$$\sum_X = E\left[(X - E(X)(X - E(X)^T\right] \quad \text{公式(3-3)}$$

$\sum_X$ 的具体计算如公式(3-4)所示。

$$\begin{Bmatrix} E(x_1-\bar{x}_1)^2 & E(x_1-\bar{x}_1)(x_2-\bar{x}_2) & \ldots & E(x_1-\bar{x}_1)(x_n-\bar{x}_n) \\ E(x_1-\bar{x}_1)(x_n-\bar{x}_n) & E(x_1-\bar{x}_1)^2 & \ldots & E(x_1-\bar{x}_1)(x_n-\bar{x}_n) \\ \vdots & \vdots & & \vdots \\ E(x_n-\bar{x}_n)(x_1-\bar{x}_1) & E(x_n-\bar{x}_n)(x_2-\bar{x}_2) & \ldots & E(x_n-\bar{x}_n)^2 \end{Bmatrix} \quad \text{公式(3-4)}$$

3.3.4 PCA-Relief-Wrapper 和 Relief-Wrapper 在手写体汉字上的实验

我们在类别数为 1000，每类样本数为 100 的手写体汉字数据集上进行了实验，比较了在选择相同数目特征时 PCA-Relief-Wrapper 和 Relief-Wrapper 的性能。如表 3-3 所示，当选择特征数目低于 200 时，PCA-Relief-Wrapper 算法的性能明显优于 Relief-Wrapper。虽然 PCA-Relief-Wrapper 在一定程度上克服了 Relief-Wrapper 的缺点，但是也相应的带来一个明显的局限，即当经过 PCA 转换后，特征不再具有任何意义，失去了在某些情况下很重要的理解性。而且，虽然 PCA 算法在手写体汉字数据集上取得了较好的效果，但由于 PCA 是针对整体数据集进行变换的，变换中没有涉及分类的信息，因此变换后保留的特征可能对分类是贡献不大的，如图 3-7 所示二维空间两类样本集分布下，原坐标轴为轴 1 和轴 2，经 PCA 变换后新坐标轴为轴 1'和轴 2'，如想把特征空间降为 1 维，则该一维空间的坐标轴为轴 1'，但样本映射到该维上后的类别可分性很差，相反被去除的一维轴 2'虽然方差小，但类别可分性却很好。这种情况下可以考虑采用 LDA(Linear Discriminant Analysis)变换方法[Fukunaga 1990]。特征变换不是本文研究内容，本文后面将不对 PCA-Relief-Wrapper 进行阐述，但通过此对比实验，我们可以得出某些情况下完全取和类别相关度高的特征并不会取得很好的性能，后续算法应考虑如何去除彼此相关度很高的特征。

表3-3 PCA-Relief-Wrapper 和 Relief-Wrapper 在手写体汉字上的实验结果

Fs	Tr-PRW	Tr-RW	Te-PRW	Te-RW
140	***94.3***	90.2	***92.7***	89.3
160	***94.8***	91.9	***92.8***	90.5
180	***94.9***	92.9	***92.8***	91.6
200	***94.9***	93.8	***92.8***	91.6
220	94.9	94.5	92.9	92.4
240	94.9	94.9	92.9	92.8
256	95.1	95.1	93.0	93.0

Fs－特征子集，Tr－PCA-Relief-Wrapper 的训练准确率，Tr-RW—Relief-Wrapper 的训练准确率，Te-PRW—PCA-Relief-Wrapper 的测试准确率，Te-RW—Relief-Wrapper 的测试准确率

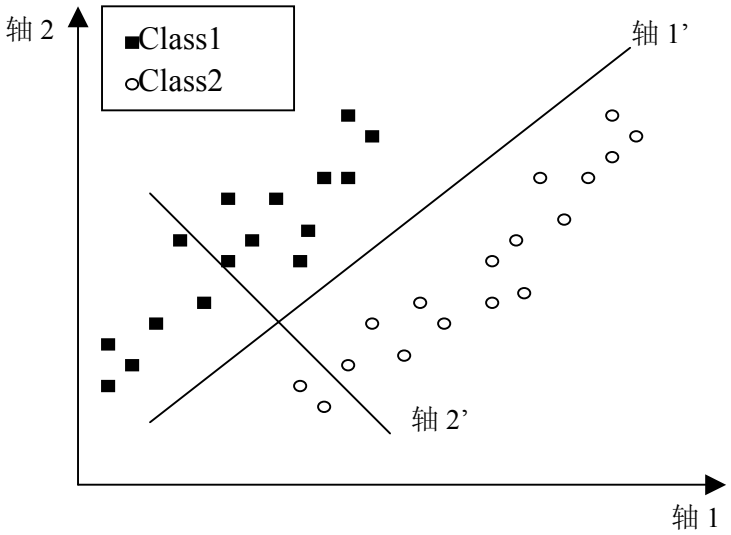

图3-7 PCA 变换和降维

3.4 串联式组合特征选择算法

为克服 Relief 算法不能去除冗余特征的缺点，我们设计了串联式组合特征选择算法 ReCorre(Relief-Correlation)和 ReSBSW(Relief-SBS-Wrapper)，两种算法分别采用相关分析以及后向搜索算法去除冗余特征。

3.4.1 Filter-Filter 组合式特征选择算法 ReCorre

ReCorre 方法为两阶段特征选择算法，第一阶段采用 Relief 去除无关特征，第二阶段采用特征间相关度分析的方法，去除冗余特征，算法中不涉及分类算法，因为可以该算法看为 Filter-Filter 模式。

3.4.1.1 特征间冗余度度量

本文借鉴了 Hall 的度量特征间冗余度的方法[Hall 2000]，采用相关分析的方法来衡量特征之间的冗余度。但在处理连续型和离散型混合的数据集时，Hall 采取首先将连续特征转化为离散型的方法，而我们认为根据特征类型区别对待更合理。特征根据其值的类型可以分为数值型连续特征和离散型特征。我们的处理规则为：如果两特征均为数值型，我们采用泊松相关系数(Poission Correlation)衡量相互间的冗余度；如果两特征均为离散型特征，我们采用信息论中的互信息的一种变形来度量。对于两特征分别为连续型和离散型的，我们首先将连续型特征转化为离散型特征，然后再采用衡量离散型特征间冗余度的方法度量互相的冗余度。我们采用了 Fayyad 的基于信息熵的离散方法对连续特征进行离散化[Fayyad 1993]。

(1).连续型特征间的相关度分析

如公式(3-5)所示，其中 x_i，y_i 分别表示特征 x 和特征 y 在第 i 个样本中的取值，$\bar{x}$ 和 $\bar{y}$ 表示特征 x 和特征 y 的均值，r 表示特征间的相关度，r 值越大，表示特征 x,y 间的相关度越大，也即意味着两者间的冗余度越大。

$$r=\frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_i (x_i-\bar{x})^2}\sqrt{\sum_i (y_i-\bar{y})}} \qquad \text{公式(3-5)}$$

(2).离散型特征间的相关度分析

对于离散属性，我们采用信息论中熵的概念，信息论中熵是不确定性的度量，我们这里采用 Shannon 熵，如公式(3-6)所示，其中 $P(x_i)$表示特征 x 取第 i 个值 x_i 的概率。已知变量 y 后 x 的不确定性用条件熵来衡量，如公式(3-7)所示，其中 $P(x_i \mid y_j)$表示特征 y 取值为 y_j 时特征 x 取值为 x_i 的概率。

$$H(x)=-\sum_i P(x_i)\log_2 P(x_i) \qquad \text{公式(3-6)}$$

$$H(x\mid y)=\sum_j P(y_j)\sum_i P(x_i\mid y_j)\log_2 P(x_i\mid y_j) \qquad \text{公式(3-7)}$$

则变量x，y之间的互信息（也称为信息增益）可按公式(3-8)进行度量[Battiti 1994a]。

$$\begin{aligned} MI(x,y) &= H(x) - H(x \mid y) \\ &= H(y) - H(y \mid x) \\ &= \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned} \quad \text{公式(3-8)}$$

相关度评估取值应在[0,1]区间，相关度为 0 表示两个特征间独立，相关度为 1，则表示两特征完全冗余。且相关度应关于变量 x,y 对称，为此，我们采用公式（3-9）进行度量。

$$SU(x,y) = 2\left[\frac{MI(x,y)}{H(x)+H(y)}\right] \quad \text{公式(3-9)}$$

[Hall 2000]的度量公式为

$$SU(x,y) = 2\left[\frac{H(x)+H(y)-H(x,y)}{H(x)+H(y)}\right] \quad \text{公式(3-10)}$$

由 $H(x,y) = H(x) + H(y \mid x) = H(y) + H(x \mid y)$，可推出

$$\begin{aligned} SU(x,y) &= 2\left[\frac{H(x)+H(y)-H(x,y)}{H(x)+H(y)}\right] \\ &= 2\left[\frac{H(x)+H(y)-(H(x)+H(y \mid x))}{H(x)+H(y)}\right] \\ &= 2\left[\frac{H(y)-H(x \mid y)}{H(x)+H(y)}\right] \\ &= 2\left[\frac{MI(x,y)}{H(x)+H(y)}\right] \end{aligned} \quad \text{公式(3-11)}$$

可见相关度定义公式（3-9）和[Hall 2000]中的离散属性的相关度公式(3-10)等同。可见，利用归一化的互信息来评估特征间相关度和 Hall 利用对称不确定度度量取得了同样的公式。

3.4.1.2 ReCorre 方法

如图 3-8 所示，ReCorre 分两阶段：第一阶段运行 Relief 算法，得到每个特征的权重 Wi，将权值大于设定阈值δ的特征加入到初始状态为空的集合 S’（图 3-8 中第 3 行）。第二阶段将集合 S’中冗余度大于阈值 *Ct* 的两特征中 Relief 权值较小的特征删除（图 3-8 中第 4-7 行）。最后输出特征选择的结

果—特征子集 S'。

输入：训练数据集Tr，测试数据集Te，Relief过滤阈值δ，相关度阈值*Ct*，
初始特征集S={F_i, i=1,…,n }

```
1.   S'=NULL
2.   Relief(Tr)  //得到特征权值W={Wi,i=1,2,…,n}  i=1, 2,  …,n
3.   If(Wi>δ)  S'= S'∪{i}  //得到新的特征集合S',  假设含有m个特征
4.   将S'特征按照权值从大到小排序，依次为1,2,  …, m
5.   For i=1,2, …,m-1
6.       For j=i+1,i+2, …,m
7.           If correlation(i,j)≥Ct   S'={S'-Fj}
                  //correlation(i,j)表示序号为i和j的特征间的相关系数
8.  Return S'
9.  输出：特征子集S'
```

图3-8 ReCorre 算法

输入： 训练数据集 Tr，测试数据集 Te，Relief 过滤阈值δ，初始特征集 S={F_i, i=1,…,n }

```
1.   Relief(Tr)得到特征权值 W={Wi,i=1,2,…,n}
2.   保留权值大于δ的特征构成新的特征集合 S'={Fi | Wi>δ, i=1,…,n }
3.   ε=C(Tr, S')
4.   εbest=ε, Changed=True
5.   while |S'|>0 & Changed {
6.      Changed=False;
7.       For i=1,2, …, |S'|{
8.          S''={S'-Fi};
9.          ε'=C(Tr,S");
10.         if(ε'<εbest){
11.           εbest=ε' ; Changed=True;
12.         Fworst=Fi ;}
13.      }//end For
14.     S'={ S'-Fworst};
15.  }//end while
输出：特征子集
```

图3-9 ReSBSW 算法

3.4.2 Filter-Wrapper 组合式特征选择算法 ReSBSW

ReSBSW 算法结构同 ReCorre 类似，为两阶段串联式组合特征选择算法，不同的是，在第一阶段采用 Relief 得到特征权值，滤掉权值小于阈值的特征得

到特征子集 S'后；ReSBSW 以分类器准确率为特征子集的评估标准，采用顺序后向搜索(Sequential Backward Search，SBS)算法逐步去除冗余特征，具体算法如图 3-9 所示。ReSBSW 算法结构为 Filter-Wrapper 式。

3.4.3 Relief, ReCorre 和 ReSBSW 算法实验比较和分析

我们对 Relief, ReCorre 和 ReSBSW 算法进行了实验比较和分析，实验中采用了人工数据集和实际数据集，人工数据集可以控制有用特征、无用特征、冗余特征和部分相关特征的比例，从而可以更清楚的分析特征选择算法的性能。实际数据由于特征间关系和目标函数均未知，常常更加复杂，通过验证特征选择算法在实际数据集上的性能，可以考察特征选择对分类器本身的性能。

3.4.3.1 实验数据集

1. 人工数据集

我们选用了两类人工数据集，如表 3-4 所示的前 6 个数据集。

（1） waveform-40 数据是 Breiman 在[Breiman 1984]中构造的数据，是从为从 waveform-21 数据集加入 19 维均值为 0 方差为 1 的符合高斯分布的随机特征得到的。Waveform-40 数据描述的是三个三角形波，前 21 维数据为加入白噪声后的 21 个等间隔点上的波高。

（2） Art 类数据采用 Isabelle Guyon 的程序产生，该程序源代码可以从 NIPS2001 的关于特征选择的主页上下载[NIPS 2001]。数据集按如下规则产生：首先产生 a2 个彼此独立的特征，由这 a2 个特征随机线性组合产生 a3 个特征，从已经产生的(a2+a3)个特征中随机选择某个特征进行复制得到 a4 个特征，接着在这 a2+a3+a4 特征中加入白噪声。然后前 a1(a1<a2)个特征的线性函数的符号决定对应样本的类别，最后按固定比例（本文为 0.01）对每个样本的类别进行随机变异。数据集名为 Art(a1,p,a2,a3,a4)格式，其中 a1 表示用以产生类别属性的特征个数，p 表示全部样本集包含的样本个数，a2 表示独立特征的个数，a3 表示无关特征的个数，a4 表示冗余特征的个数。可见这个数据集既含有固定的有用特征，又含有完全无关的特征，同时也含有和目标函数部分相关的

特征，特征值和样本类别又含有噪声，因此这是一个较复杂的分类问题。我们用这类数据集考察 Relief 算法去除无关特征的比例，以及 Relief 将多少个有用特征（这里有用特征表示直接用来产生目标函数的特征，包括冗余特征）错认为是无关特征。

2. 实际数据集

实际数据集除包括前几节使用的 100 类每类 100 样本的手写体汉字数据集外，其余均选自 UCI 数据集[UCI]，如表 3-4 下部分所示，这些数据集的特征维数从数十到数百不等，数据类型不同（只包含离散型或连续型特征的单一类型以及同时包含离散型和连续型特征的混合类型），也有的数据集的数据不完全，以上这些数据有较广泛的代表性，可以较好地验证特征选择方法在大规模实际数据集上的性能。有关数据集的意义介绍可参见附录。采用的分类器为 C4.5 决策树[Quinlan 1993]和朴素贝叶斯分类器(NBC, Naive Bayes Classifier)。

表3-4 实验数据集

数据集	Nom	Num	#Train / #test	#C
waveform-40	0	40	300/4700	3
Art(3,300,50,10,10)	0	70	200/100	2
Art(3,3000,50,10,10)	0	70	2000/1000	2
Art(3,3000,50,50,50)	0	150	2000/1000	2
Art(25,3000,50,50,50)	0	150	2000/1000	2
Art(40,3000,50,50,50)	0	150	2000/1000	2
Ionosphere	0	34	234/117	2
Hanzi	0	256	5000/5000	100
Isolet	0	617	1039/520	26
Satimage	0	36	4435/2000	6
Auto	11	15	136/69	7
Chess	36	0	2130/1066	2
DNA-nominal	60	0	2000/1186	3
German	13	7	666/334	2
horse-colic	15	7	300/68	2
Hypothyroid	7	18	2109/1054	2
Sonar	0	60	215/108	2

[a] Num-数值型特征的个数. [b] Nom-离散型特征的个数. [c] #Train/#Test-训练样本的数目/测试样本的数目.[d] #C－类别个数

3.4.3.2 实验方法

(1) 实验中采用的分类器

我们实验中用到的分类器有两种，分别为：1) c4.5 决策树；2) 朴素贝叶斯分类器（NBC），NBC 算法设定各特征独立，对具有 n 个特征(A1，A2，…，An)的样本 X，其类别为 y 的概率按如下贝叶斯公式估计：P(y | X) =P(X | y)·P(y)/P(X) ∝P(A1，…，An | y) · P(y)=∏P(Aj | y) ·P(y)。其中对于离散特征，P(y)以及 P(Aj | y)通过计数进行估计；对于连续特征，采用高斯分布进行逼近，高斯分布的参数期望值和标准偏差由训练集中统计得到。NBC 算法实现简单，而且即使在特征间独立的前提不成立的情况下也能得到很好的效果，并且对无关特征不敏感[Langley 1997]。

(2)实验步骤

在每个数据集上，我们运行 Relief，ReCorre，ReSBSW 特征选择算法，记录下特征选择后的特征子集的大小，然后在各特征选择算法处理后的训练数据集上训练得到 C4.5 决策树或者朴素贝叶斯分类器，接着得到 C4.5 或者朴素贝叶斯分类器在测试集上的分类错误率并记录下来。每个实验重复 5 次，每次均对数据集进行随机打乱以得到独立训练集和测试集。采用训练集上的 3 倍交叉验证法（3-fold Cross Valication）法[Blum 1999]得到 Wrapper 式特征选择过程中作为特征子集评估标准的分类器性能。表 3-5，3-6 所列结果为 5 次重复实验的平均值。

3.4.3.3 实验结果及讨论

表 3-5，表 3-6 分别列出了 C4.5 和 NBC 在各特征选择算法处理后的测试准确率，为比较起见，每个表的最后一列列出了分类器在原始数据集上的测试准确率。表 3-7 列出了 Relief，ReCorre，ReSBSW(C4.5)以及 ReSBSW(NBC)算法得到的特征子集大小，ReCorre(0.8)和 ReCorre(0.7)分别表示图 3-8 中的相关度阈值 *Ct* 分别取 0.8 和 0.7，ReSBSW(C4.5)和 ReSBSW(NBC)表示涉及分类器分别为 C4.5 和 NBC。

由表 3-5 可见，采用 ReSBSW 进行特征选择后，C4.5 在 8 个数据集上取

得了最高测试准确率；采用 ReCorre(0.8)和 ReCorre(0.7)进行特征选择后，C4.5 在 5 个数据集上取得了最高测试准确率；采用 Relief 进行特征选择后，C4.5 在 4 个数据集上取得了最高的测试准确率；而采用全部特征的 C4.5 在 2 个数据集上取得了最高测试准确率，总的说来 ReSBSW 算法对应的准确率最高。另外，经各特征选择算法处理后得到的测试准确率大多数情况下比未进行特征选择得到的测试准确率高，如 Relief 特征选择后 C4.5 在 7 个数据集上测试准确率高于采用全部特征得到的测试准确率，而仅在 2 个数据集上低于采用全部特征得到的测试准确率，说明采用特征选择不仅可以降维而且很多情况下可以提高算法的准确率。由表 3-6 可见，经 ReSBSW 特征选择后，NBC 在 7 个数据集上取得了最高的测试准确率；而经 Relief，ReCorre(0.8)和 ReCorre(0.7)特征选择后，NBC 算法分别在 4 个数据集上取得了最高的测试准确率；而采用全部特征的 NBC 算法在 6 个数据集上得到了最高的测试准确率，总的说来 ReSBSW 特征选择算法对应的准确率最高。表 3-5 和表 3-6 说明采用 Wrapper 模式的特征选择算法 ReSBSW 更有效，但 ReSBSW 算法运行效率低，在高维数据上运行时间太长使其应用上受到限制，如在 Hanzi 和 Isolet 数据集上 24 小时内 ReSBSW 算法没有运行完。

图 3-10，3-11，3-12，3-13 分别表示 Relief 算法去除特征数目的比例、去除无关特征的比例、去除因果特征的比例以及去除相关特征的比例，因果特征表示直接决定目标函数的特征，相关特征指不直接决定目标函数但是和目标函数有一定相关度的特征。可见，对人工数据集，Relief 可去除约 50%的无关特征，但也会将约 20%的决定目标函数的因果特征去掉，同时大约 40%的相关特征（Dependent features，由决定目标函数的特征和无关特征线性组合得到）被去除。但是在实际数据集上，如表 3-7 所示，Relief 去除的特征很有限。

表 3-8 列出了运行 ReCorre 算法后冗余特征数目以及运行 Relief 后的冗余特征数目，可见 ReCorre 在去除冗余特征上非常有效。而且从表 3-5，3-6 可见，在人工数据集上，经过 ReCorre 处理后，和 Relief 相比，在大部分人工数据集上，学习算法的性能得到了提高；但在大部分实际数据集上，ReCorre 的准确率要低于 Relief 算法，说明 ReCorre 经常会去除对某种分类器来说有用的特征。

表3-5 C4.5 上各特征选择算法的测试错误率

	Relief	ReCorre(0.8)	ReCorre(0.7)	ReSBSW	All
Waveform-40	**71.00** [a]	**71.00**	**71.00**	70.72	69.50
Art(3,300,50,10,10)	**75.00**	**75.00**	**75.00**	**75.00**	73.00
Art(3,3000,50,10,10)	84.50	84.50	84.50	**86.1**	81.4
Art(3,3000,50,50,50)	81.70	**83.20**	**83.20**	81.00	80.00
Art(25,3000,50,50,50)	61.50	**61.60**	**61.60**	60.40	61.20
Art(40,3000,50,50,50)	**62.60**	61.00	61.00	60.50	60.90
Auto	71.01	66.67	66.67	**76.81**	71.01
Chess	97.84	97.84	97.84	97.84	**99.00**
German	71.56	71.56	71.56	**71.86**	71.56
horse-colic	80.88	80.88	80.88	**83.82**	80.88
hypothyroid	98.80	98.80	98.40	**99.00**	98.90
sonar	75.72	62.86	68.57	**75.74**	74.29
hanzi	**59.96**	58.02	57.92	—[b]	**59.96**
Isolet	73.08	**74.24**	70.20	—	73.08
Satimage	84.9	84.4	**88.04**	83.25	84.90
Ionosphere	88.04	88.04	79.55	**88.89**	88.04

[a] 黑体表示每行的最大值 [b] —表示 48 小时内未运行完

表3-6 NBC 上各特征选择算法的测试错误率 [a]

	Relief	ReCorre(0.8)	Recorre(0.7)	ReSBSW	All
Waveform-40	80.53	80.53	**81.13**	80.17	80.53
Art(3,300,50,10,10)	74	**79**	**79**	75	74
Art(3,3000,50,10,10)	79.5	80.6	80.6	**82**	76.6
Art(3,3000,50,50,50)	75.8	81.4	81.4	**81.5**	77.7
Art(25,3000,50,50,50)	73.7	74.1	74.1	**76.5**	73.6
Art(40,3000,50,50,50)	76.2	75.7	75.7	74.5	**76.9**
auto	50.72	52.17	52.17	**53.62**	50.72
chess	88.36	88.36	88.36	**93.71**	87.14
German	**77.54**	**77.54**	**77.54**	73.95	**77.54**
Horse-colic	79.41	79.41	79.41	77.94	**80.88**
hypothyroid	**97.73**	**97.73**	**97.73**	97.54	**97.73**
sonar	**70**	68.57	68.57	68.57	**70**
hanzi	94.12	**94.42**	94.02	—[b]	94.12
Isolet	**85.96**	85.57	85.00	—	**85.96**
Satimage	79.65	79.4	76.55	**79.75**	79.65
Ionosphere	83.76	83.76	83.76	**84.62**	83.76

[a] 黑体表示每行的最大值 [b] —表示 48 小时内未运行完

一个有趣的发现是，虽然 ReCorre 算法仅仅是在 Relief 基础上去除冗余特征，但其在 Art(3,3000,50,50,50)和 Art(25,3000,50,50,50)数据上 C4.5 和 NBC 的性能得到提高，而在 Art(40,3000,50,50,50)上的 C4.5 和 NBC 上性能却下降了。在实际数据集上的结果也出现或者提高或者下降的现象，这从某种程度上说明特征选择和学习算法间关系的复杂性，也说明不应脱离学习算法直接进行特征选择。另外，值得说明的是，ReSBSW 算法在采用 C4.5 决策树和 NBC 时所去除的特征差别很大，这也表明同样特征对不同分类器作用不同，某个对 C4.5 决策树有用的特征，对 NBC 却可能无用，反之亦然，这再一次验证了 Kohavi 提出的 Wrapper 思想[Kohavi 1997]。

表3-7 各特征选择算法得到的特征子集大小

Datasets	All	Relief	ReCorr(0.8)	ReCor(0.7)	ReSBSW (C45)	ReSBSW (NBC)
Waveform-40	40	29	29	28	**25**	27
Art(3,300,50,10,10)	70	39	**34**	**34**	38	37
Art(3,3000,50,10,10)	70	36	30	30	**25**	28
Art(3,3000,50,50,50)	150	91	**61**	**61**	81	81
Art(25,3000,50,50,50)	150	92	**61**	**61**	84	84
Art(40,3000,50,50,50)	150	95	**65**	**65**	84	91
Auto	26	25	21	**19**	22	21
Chess	36	28	28	28	23	**17**
German	20	20	20	20	18	**17**
horse-colic	22	19	19	19	16	**14**
hypothyroid	25	24	23	19	22	**17**
sonar	60	56	34	**28**	55	55
Hanzi_256	256	256	246	208	—	—
Isolet	617	617	314	187	—	—
satimage	36	36	13	**3**	29	34
ionosphere	34	33	32	**26**	30	30

表3-8 相关分析算法去除冗余特征, 特征集合中的冗余特征数目

Datasets	Relief	ReCorre
Art(3,300,50,10,10)	5	0
Art(3,3000,50,10,10)	6	0
Art(3,3000,50,50,50)	32	2
Art(25,3000,50,50,50)	31	0
Art(40,3000,50,50,50)	31	1

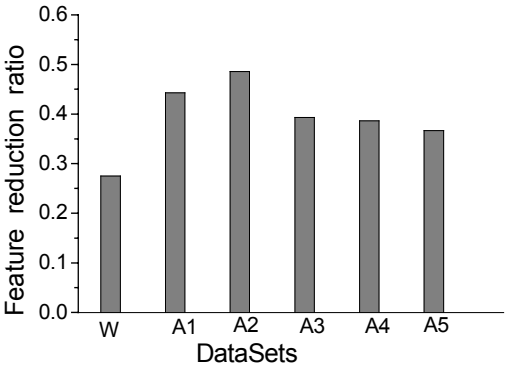

图3-10 Relief 去除特征的比例

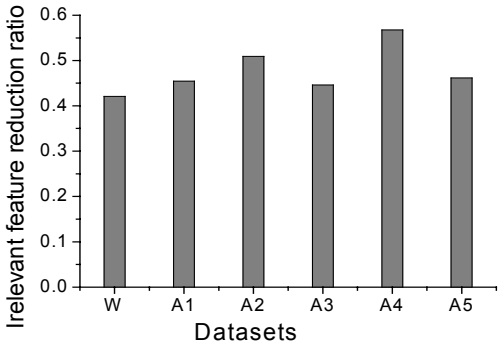

图3-11 Relief 去除无关特征的比例

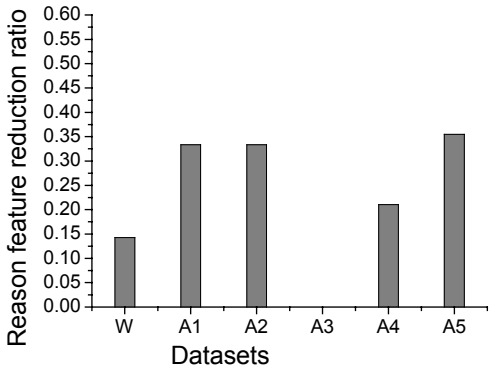

图3-12 Relief 去除因果特征的比例

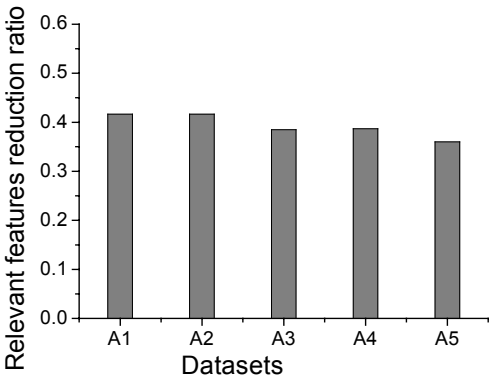

图3-13 Relief 去除相关特征的比例

3.5 Relief-GA-Wrapper 耦合式特征选择算法

3.5.1 算法提出的动机

前面提出的 Relief-Wrapper，ReCorre 以及 ReSBSW 算法有一个共同点，即默认 Relief 对特征和目标函数间的相关度评估是正确的，只是不能去除冗余特征。理论上是这样的，但如上节实验结果可见，Relief 评估有不容忽视的偏差，会出现本来和类别相关的特征被错认为是无关特征的情况，也会出现本来和类别无关的特征被错认为相关的情况。因此采用 Relief 直接过滤“无关”特征过于乐观了。Relief-Wrapper 根据 Relief 给出的特征排序决定选择多少排序靠前的特征，ReCorre 和 ReSBSW 方法首先采用 Relief 去掉”无关特征”。实验结果表明 Relief-Wrapper 的准确率低于 GA-Wrapper，ReCorre 的准确率低于

ReSBSW，这说明了以下两点：一，特征选择算法中不应该用 Relief 去除无关特征；二，特征选择算法应考虑学习算法的性能。根据 Kohavi[Kohave 1997]的研究结果以及上节实验可知：对不同的分类算法，不存在一个公共的最优的特征子集，比如决策树分类器偏好的特征子集，最近邻法却可能得到较差的性能。本文作者认为特征子集的评估不能离开学习算法性能的反馈，但是可以利用 Filter 得到的统计信息来加快 Wrapper 类算法的搜索速度，进而得到综合 Filter 和 Wrapper 优点的组合式特征选择算法。

第二章介绍采用遗传算法进行特征选择时提到，遗传算法用于特征选择性能很好，但是当应用于 Wrapper 式特征选择结构时，算法效率较低，那么是否可采用 Relief 评估的信息加速遗传算法搜索的速度呢，这正是将要介绍的 Relief-GA-Wrapper 算法提出的动机。

3.5.2 Relief-GA-Wrapper 算法描述

Relief 评估用来过滤无关特征是对 Relief 评估能力过分依赖了，但是 Relief 的特征评估应该可以为 Wrapper 结构的搜索算法提供启发信息。

在上节分析 Relief 和 GA-Wrapper 两种算法优缺点的基础上，本着充分利用 Filter 阶段信息以加速 Wrapper 阶段搜索的思想，本文有机地将两种算法接合起来，提出了 Relief-GA-Wrapper（简称 RGW）算法。RGW 算法采用对特征进行加权的 Filter 算法和 Wrapper 耦合的模型，其中 Filter 的作用是进行特征加权，而不是特征过滤，此特征加权为 Wrapper 阶段的遗传算法种群初始化提供启发信息，目的是使得遗传算法的初始种群中含有较好的初始点，从而使遗传算法可采用小的进化代数和小规模种群搜寻到较优的特征子集。采用这种模式的目的是为了避免前面所提组合模式中 Filter 过滤掉重要特征的危险，减小 Wrapper 方法过适应的可能性，同时提高遗传算法用于 Wrapper 式特征选择的运行效率。

图 3-14 为 RGW 方法的示意图，RGW 算法在采用 Relief 得到特征评估的基础上，根据 Relief 得到特征的排序（排序靠前表示该特征对近邻样本的区分能力强）引导遗传算法种群初始化，然后用第二章介绍的 GA-Wrapper（简称 GW）方法进行特征选择。

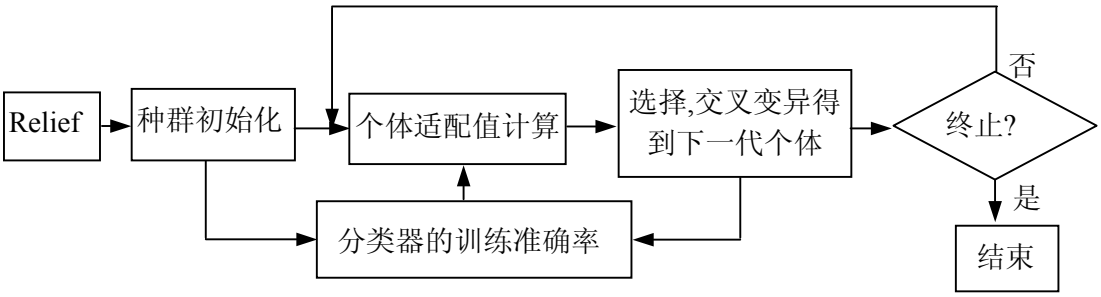

图3-14 Relief-GA-Wrapper 算法

3.5.3 为什么选用 Relief 和遗传算法

特征选择算法涵盖众多的 Filter 式算法和搜索算法，本文选择 Relief 和遗传算法主要出于以下考虑：

(1) Filter 所得结果的作用是为 Wrapper 的过程提供引导，因此要求 Filter 方法得到每个特征权值，而不是特征子集，在 Filter 方法中满足此条件的常用的有信息增益和 Relief 评估，信息增益的计算需要首先将连续的特征离散化，而 Relief 的算法则不受数据类型限制，而且对噪声和特征相互作用不敏感。

(2) 对于大规模数据，Relief 算法效率高。Filter-Wrapper 组合式算法要想运行效率高，那么 Filter 阶段的效率一定要高，对于大规模数据，Relief 算法由于迭代次数远远少于样本数，因而运行效率较高。

(3) 搜索算法要适合大规模数据，且对于特征间复杂的关系不敏感。顺序搜索算法容易陷入局部极值，加入一定随机性的模拟退火算法具有较强的局部搜索能力，并能使搜索过程避免陷入局部最优解，但模拟退火算法对整个搜索空间的状况了解不多，不便于使搜索过程进入最有希望的搜索区域，从而使得模拟退火算法的运算效率不高。相对而言，遗传算法对问题依赖性小，全局搜索能力强，适合大规模复杂问题的优化。另外遗传算法可以采用涉及多目标的评估函数，如既考虑分类准确率又兼顾特征子集大小的目标函数。 [Maria 1999][Yang 1998][Vafaie 1992]等采用遗传算法进行特征选择，取得了较好的结果。但据 Jain[Jain, 1997]实验，遗传算法容易过早收敛，需要就此采取措施，而且当应用于大规模数据

时，如果采用分类器的准确率作为特征子集的评估，运行效率较低，所以考虑用 Relief 的评估信息引导遗传算法的 Wrapper 式搜索，以提高遗传算法的运行效率。

综合以上分析，我们选择了 Relief 算法和遗传算法，并试图通过两者的结合提高 Wrapper 的速度，同时保持较高的测试准确率。

3.5.4 遗传算法种群初始化

遗传算法是一种不确定搜索算法，其性能受初始种群影响很大，一个好的初始种群可以为遗传算法提供一批好的搜索起点，Relief 评估相当于为遗传算法提供了问题的先验知识。Relief 引导遗传算法种群初始化的具体步骤为：

(1) 不同特征根据 Relief 评估结果进行排序；排序越靠前表示该特征和类别相关性越大。

(2) 产生特征的被选概率；根据特征排序先后具有不同的选择概率。如果某特征排序在前边，那么就使得遗传算法个体中该特征对应位为 1 的概率大些，反之如果该特征排序靠后，则使对应位为 1 的概率小些。具体操作如下：设定排序第 1 的特征选中概率为 p1 (p1>0.5)，排序最后的特征选择概率为 p2 (p2<0.5)，然后按照等差数列或等比数列求出其余特征的选中概率，其中 p1 和 p2 是人为设定的参数。另外，还可以参考遗传算法里的轮盘赌选择法决定各特征被选的概率，不过这时需要特殊处理某特征权值为负的情况，否则权值为负的特征将永远不会被选择，这就退化到了用 Relief 去除无关特征的情况。本文中 p1 和 p2 取值为 0.8 和 0.4，并选用等差数列产生各特征的选择概率。

(3) 种群初始化：设初始种群大小为 g，依以上所得特征的选择概率产生个(g-2)个个体。同时，为充分利用已知信息，初始种群包含 Relief 本身进行特征选择得到的特征选择方案（即权值大于 0 的特征子集）对应的个体编码。并且，由于决策树自身具有特征选择的功能（树的每一个非叶子结点都要选择特定的特征），初始种群也包含了决策树自身选用的特征子集对应的个体编码，这样就得到了大小为 g 的初始种群。

3.5.5 遗传算法的个体评估函数

影响遗传算法性能的另一个重要因素是个体的评估函数，对于特征选择算法，个体即特征子集的评估应满足：

1）分类准确率越高，评估函数值越大；

2）特征子集越小，评估函数值越大。

在参考文献[Kudo 2000] 中遗传算法评估函数选择的基础上，本文设计了如公式(3-12)所示的评估函数：评估标准中起决定作用的是分类准确率，当分类准确率相近时，特征数目少的选择方案优先。

$$f(X)=\alpha\exp\left(-\frac{|X|}{n}\right)+\exp\left(\frac{A(X)-(1-\beta)A_0}{\beta A_0}\right) \qquad \text{公式(3-12)}$$

其中$|X|$表示特征子集 X 所包含的特征数目，$A(X)$表示选用特征子集 X 时得到的训练准确率，A_0 表示包含全部特征时的训练准确率，n 为所有特征的数目。公式(3-12)等式右侧的第一项表示特征数目越少，评估函数取值越大；第二项表示特征子集 X 对应的分类准确率越大，则评估函数越大。 参数α用来控制特征数目的减少和分类准确率的提高对评估函数的贡献，本文中取值为0.5，即认为分类准确率对评估函数更重要，而特征数目的减少次之。等式右侧第二项中的β用来设置能够接受的选用特征子集 X 后分类器准确率下降的百分比。由于当 $A(X)<(1-\beta)A_0$ 时指数函数值降得很快，因此相当于设定了特征子集分类准确率的阈值为$(1-\beta)A_0$。本文中β取值为 0.01，表示当选择特征子集 X 对应分类器的准确率较使用全部特征集合的分类器准确率降低 1%时，此特征子集的评估值迅速下降，因而不会被选中。用户可以根据不同问题和需要设置参数α和β的值，如当希望降维幅度大些时，则增加α的值，否则减少α的值；如果可容忍较大幅度的准确率降低，则增加β的值，否则减少β的值。

3.5.6 Relief-GA-Wrapper 具体参数设置

Relief 算法中，对于样本数大于 1000 的训练数据集，迭代次数 M 为训练集的三分之一，否则 M 取值为训练集的大小，K 取值为 1。

遗传算法的参数设定如下：种群大小 30，最大迭代次数 20，交叉概率

0.8，变异概率为 0.1。为了使在进化过程中产生出的优良个体能尽可能的保留到下一代群体中，采用最优保存策略进化模型(Elitist Model)来进行优胜劣汰操作，即当前群体中适应度最高的个体不参与交叉运算和变异运算，而是用它来替换掉本代群体中经过交叉、变异等遗传操作后所产生的适应度最低的个体。交叉算法是产生新个体的主要方法，它决定了遗传算法的全局搜索能力。为了增加群体的多样性，引入了均匀交叉(Uniform Crossover)算子，均匀交叉操作通过设置一屏蔽字来确定新个体的各个基因由哪一个父代个体来提供。采用轮盘赌选择法，终止条件为连续 5 次最优解不变或者达到最大迭代次数。

3.5.7 实验结果和分析

3.5.7.1 实验数据

表3-9 实验数据集

数据集编号	数据集名称	Nom[a]	Num[b]	#Train / #test[c]	#C[d]
1	Anneal	32	6	598/300	6
2	Auto	11	15	136/69	7
3	Chess	36	0	2130/1066	2
4	DNA-NOMINAL	60	0	2000/1186	3
5	German	13	7	666/334	2
6	German-org	12	12	666/334	2
7	Horse-colic	15	7	300/68	2
8	Hypothyroid	7	18	2109/1054	2
9	Ionosphere	0	34	234/117	2
10	Isolet	0	617	1039/520	26
11	Nettalk	203	0	7229/7242	324
12	Satimage	36	36	4435/2000	6
13	sick-euthyroid	18	7	2108/1055	2
14	soybean-large	35	0	455/228	19
15	Sonar	0	60	215/108	2
16	waveform-21	0	21	300/4700	3
17	waveform-40	0	40	300/4700	3

[a] Num-数值型特征的个数. [b] Nom-离散型特征的个数. [c] #Train/#Test-训练样本的数目/测试样本的数目.[d] #C－类别个数.

实验数据集选自 UCI 数据集[UCI]，如表 3-9 所示，这些数据集的特征维数从数十到数百不等，数据类型不同（离散型，连续型，同时包含离散型和连

续型特征的混合类型），也有的数据集的数据不完全，以上这些数据有较广泛的代表性，可以较好地验证 Relief-GA-Wrapper(简称 RGW)方法在大规模数据集上的性能。

3.5.7.2 实验方法

实验中采用的分类器为 C4.5 决策树[Quinlan 1993]。为验证 RGW 算法的性能，我们将其和下面几种已有的 Filter 算法，Wrapper 算法以前面介绍的 ReSBSW 组合算法进行了实验比较：Relief 算法，选择权值大于零的特征构成的特征子集作为特征选择结果；SFSW（SFS-Wrapper）和 SBSW(SBS-Wrapper)分别采用顺序前向和顺序后向搜索算法，采用分类准确率作为特征子集评估标准，搜索终止条件为找不到使评估改进的特征进行添加(SFSW)或删除(SBSW)。GW 即第二章介绍的遗传算法的 Wrapper 式特征选择算法。

在每个数据集上，我们运行 Relief，RGW，GW，SFSW，SBSW 和 ReSBSW 特征选择算法，记录下运行时间，特征子集的大小以及 C4.5 决策树在各特征选择算法处理后得到的训练集上训练后，然后在测试集上得到的分类准确率。为比较起见，也记录并列出了原始特征集合的大小，不进行特征选择的 C4.5 决策树的运行时间，以及在原始数据集上 C4.5 的测试准确率，即有关表格中 Full-Set 对应的值。每个实验重复 5 次，每次均对数据集进行随机打乱以得到独立训练集和测试集。采用训练集上的 3 倍交叉验证方法得到 Wrapper 类特征选择算法中分类器的性能评估。表 3-10，3-11，3-12 所列结果为 5 次重复实验的平均值(在 48 小时内未得到实验结果的 Isolet 没有参与 SBSW 和 ReSBSW 的平均值计算)。

3.5.7.3 实验结果及讨论

评价一个特征选择算法，主要考察以下三方面：经该特征选择算法处理后的数据集上的分类准确率，特征子集的大小，以及算法的运行效率。表 3-10，3-11，3-12 分别从以上三方面对各特征选择算法的性能进行了比较。

- 分类准确率

考察分类准确率，由表 3-10 可见，在全部 17 个数据集中，RGW 在 8 个

测试数据集上取得了最高分类准确率，平均测试准确率也明显高于其他特征选择算法，图 3-15~3-20 显示了 RGW 和其他特征选择算法以及不进行特征选择的准确率对比关系，图中直线为 y=x 的函数，每一个点对应一个数据集，如果该点在 y=x 直线的上方，说明 RGW 在该数据集上的测试准确率比参与比较的特征选择算法高，从图 3-15~3-20 可以更直观的看出 RGW 和特征选择算法相比在测试准确率上的优势。RGW 的分类准确率明显高于 GW 和其他几种特征选择算法，这可以从以下方面理解：初始种群对搜索代数不高的遗传算法性能有重要影响，RGW 初始种群引入的 Relief 启发信息和决策树本身的特征选择信息有利于遗传算法快速确定最优特征子集；遗传算法代数较低，从某种程度了降低了过配的概率；SFS，SBS 搜索算法容易导致局部极值，对于特征数目较多的特征选择问题，搜索空间较大，这两种方法更容易陷入局部极值，因此性能并不理想。

- 特征子集的大小

从特征子集的大小来看，SFS 算法得到的特征子集最小，GW 次之，RGW 居中。Relief 降维的作用很小，这验证了 Relief 对冗余特征无能为力的弱点。SFSW 算法得到的特征子集小，分类准确率较低，SBSW 和 ReSBSW 算法的降维作用较弱，说明顺序搜索算法容易很快陷入局部极值的情况。

- 算法的运行效率

运行时间由数据集本身的特征维数、样本个数以及各算法的时间复杂度决定。RGW 运行时间对特征维数的升高不敏感，因为 RGW 中遗传算法的种群大小和搜索代数为常数，特征维数的影响是由对分类器速度的影响而间接体现在对 RGW 运行效率的影响上。由表 3-12 可见，随着特征维数的升高，RGW 方法运行时间明显低于 SFS 和 BFS 方法，但高于 Relief 方法，RGW 算法的运行时间高于 GW 算法，因为 RGW 多了 Relief 算法的运行时间，但实际应用中，Relief 算法可以事先离线运行得到，然后 RGW 算法直接引用 Relief 的评估结果，这样 RGW 和 GW 的运行时间就近似相等。在 Isolet 数据集上，SBS-W 和 ReSBSW 在 48 小时内没有运行完，计算效率上的劣势更为明显。

表3-10 各种特征选择算法得到的分类准确率

No	Full-Set	RGW	GW	SFSW	SBSW	ReSBSW	Relief
1	89.4±0.64	91.1±1.82	90.1±0.33	89.8±0.25	**91.4±1.27**	**91.4±1.27**	88.0±0.33
2	67.4±5.2	**76.8±1.13**	67.4±4.26	62.3±7.23	73.9±4.44	74.3±2.76	67.4±5.06
3	**99.3±0.21**	98.9±0.15	98.5±0.01	94.55±0.65	97.1±2.00	98.8±0.35	98.5±0.38
4	**93.2±0.89**	92.7±1.21	91.4±1.05	**93.2±1.47**	91.4±2.44	91.4±2.44	**93.2±0.89**
5	72.0±1.45	**72.2±1.4**	72.0±1.16	69.4±0.72	69.9±1.05	71.1±0.23	71.5±0.85
6	73.5±1.35	**74.1±0.91**	70.9±0.67	71.7±2.13	72.5±0.16	70.7±1.36	73.5±0.45
7	83.1±2.29	83.1±0.83	**83.8±0.21**	82.8±2.05	83.1±2.28	83.8±1.55	83.0±2.12
8	98.9±0,33	99.1±0.19	**99.2±0.11**	98.8±0.33	98.8±0.23	98.8±0.23	98.8±0.23
9	88.4±0.4	**88.2±2.09**	86.3±3.62	87.2±2.15	87.6±0.54	87.6±0.54	87.2±0.88
10	74.8±0.54	**76.2±0.89**	74.4±1.54	68.3±1.25	—	—	75.2±0.97
11	72.1±0.14	72.4±0.29	63.8±0.46	72.0±0.65	72.2±0.23	**72.5±0.12**	72.3±0.32
12	86.1±0.55	**86.1±0.89**	85.2±0.48	85.3±1.44	85.0±0.70	85.0±0.70	86.0±0.65
13	**98.0±0.11**	97.8±0.11	96.8±0.14	97.7±0.12	97.9±0.11	97.8±0.10	97.8±0.10
14	74.3±0.02	**75.9±1.67**	65.7±0.45	72.1±2.16	62.4±1.59	74.5±0.43	74.5±0.43
15	89.7±0.26	90.1±0.61	88.8±1.02	88.0±1.88	**90.7±0.35**	90.4±0.05	89.7±0.26
16	70.1±0.13	**71.2±0.45**	70.6±0.38	69.6±0.1	69.1±0.58	69.1±0.58	70.1±0.13
17	69.5±0.89	70.8±1.32	70.4±0.53	65.5±2.15	67.8±0.44	68.2±0.67	**72.0±0.50**
AV	82.8±1.45	**83.8±0.620**	81.3±0.73	81.2±0.71	81.9±0.71	82.8±0.66	82.7±0.65

表3-11 各种特征选择算法得到的特征子集大小

No	Full-Set	RGW	GW	SFSW	SBSW	ReSBSW	Relief
1	38	23	19.5	8	37	15	16
2	26	14	13	4	24	24	25
3	36	23	25	5	35	26.5	28
4l	60	34	30	11	59	59	60
5	20	12	11	11	13	15	20
6	24	16	15	13	21	14	19
7	22	12	11	6	20	18	19
8	25	12	11	3	24	23	24
9	34	19	18	7	33	32	33
10	617	355	351	26	—	—	617
11	203	161	121	74	196	157	166
12	36	24	22	12	32	32	36
13	25	14	10	5	24	22	23
14	60	30	29	5	58	55	56
15	35	24	21	12	33	32	35
16	21	12	12	9	18	17	21
17	40	22	17	8	38	25	29
AV	44.1	28.3	24.1	12.1	41.6	35.4	38.1

表3-12 各种特征选择算法的运行时间（秒）

No	Full-Set	GW	RGW	SFSW	SBSW	ReSBSW	Relief
1	1	34	46	497	12	19	12
2	1	22	23	17	4	5	1
3	1	45	114	35	11	80	69
4	1	65	130	1197	748	826	65
5	1	38	47	54	24	20	9
6	1	47	58	899	21	20	11
7	1	17	18	44	5	5	1
8	1	41	151	22	11	120	110
9	1	26	27	41	9	10	1
10	29	6696	6748	16767	——	——	52
11	50	4118	6332	5397333	33031	30501	2214
12	2	422	750	1258	487	819	328
13	1	53	154	141	14	116	101
14	1	27	28	363	23	16	1
15	1	26	31	454	12	19	5
16	1	27	29	44	12	13	2
17	1	47	51	78	30	27	4

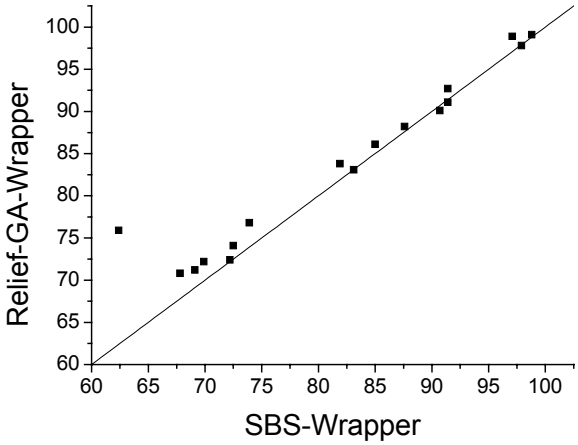

图3-15 Relief-GA-Wrapper 和 SBS-Wrapper 的准确率比较

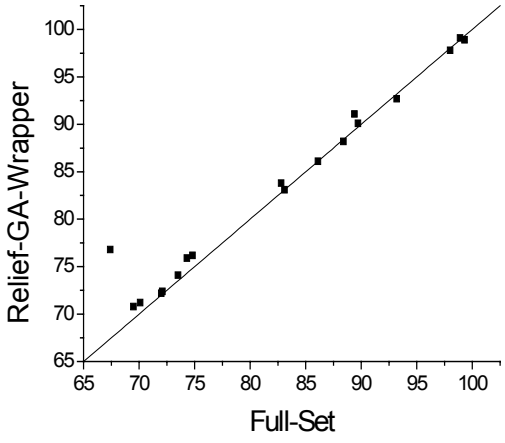

图3-16 Relief-GA-Wrapper 和 Full-Set 的准确率比较

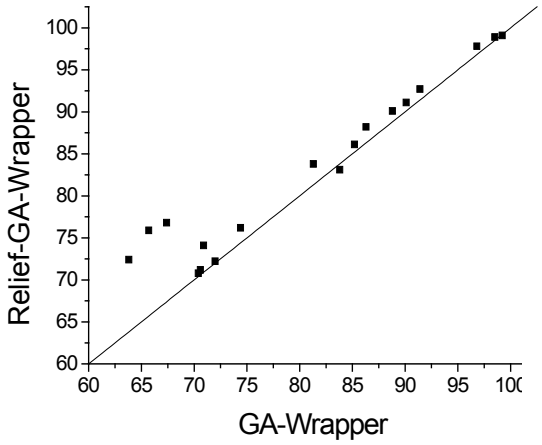

图3-17 Relief-GA-Wrapper 和 GA-Wrapper 的准确率比较

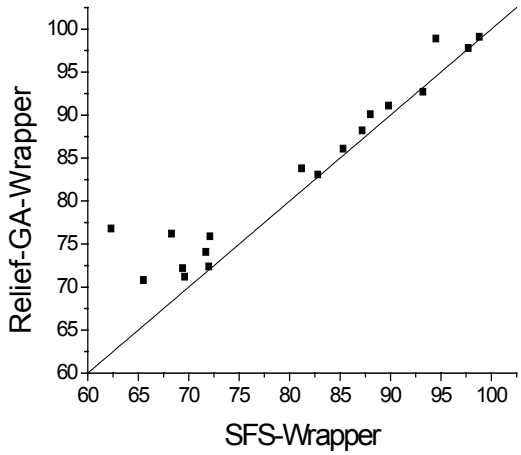

图3-18 Relief-GA-Wrapper 和 SFS-Wrapper 的准确率比较

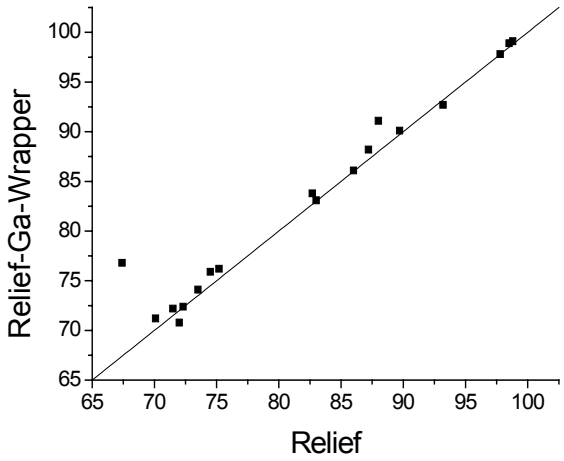

图3-19 Relief-GA-Wrapper 和 Relief 的准确率比较

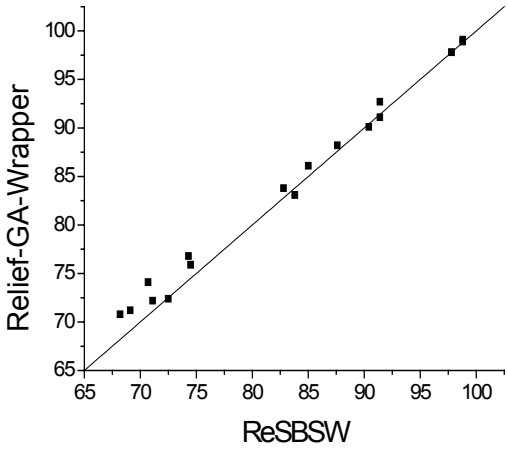

图3-20 Relief-GA-Wrapper 和 ReSBSW 的准确率比较

3.5.7.4 总结

从实验结果比较可以看出，Relief-GA-Wrapper 算法在分类准确率，选择特征子集大小和运行时间方面均优于 ReSBSW 算法和 SBSW 算法，在分类准确率和特征子集大小方面优于 Relief 算法，对高维数据在分类准确率和运行时间方面优于 SFSW 算法。综合而言，Relief-GA-Wrapper 算法既可得到较高的

分类准确率，在计算效率上也明显高于一般的顺序搜索算法，在大规模数据的特征选择上具有明显的竞争力。

有一点需要指出的是，本文 Relief-GA-Wrapper 算法中遗传算法的最大迭代代数为 20，如果增加迭代次数，Relief-GA-Wrapper 相比于 GA-Wrapper 的优势会逐渐减少。实验表明，随着迭代次数的增加，Relief-GA-Wrapper 和 GA-Wrapper 的性能确实在慢慢接近。但遗传算法和 Wrapper 结构的时间开销较大，对于较大规模的数据集，采用很高的迭代次数在实际应用中是不适宜的，而 Relief 和 GA-Wrapper 的组合恰好可以弥补遗传算法时间开销大的缺陷，同时在较低的计算量下得到较高的分类准确率。

3.6 本章小结

本章通过实验测试了 Relief 特征评估方法，提出基于 Relief 的几种特征选择算法，并进行了实验比较和分析。本章主要工作和结论如下

1. 通过 Relief 对手写体汉字和人工数据集上的评估表明，Relief 是一种性能较好的特征评估算法，但是 Relief 的评估也有不容忽视的偏差，而且无法辨别冗余特征，因此直接采用 Relief 进行特征选择的风险较大。

2. 设计了两阶段串联型组合算法 ReCorre 和 ReSBSW，ReCorre 通过相关分析可以有效的去除大部分冗余特征，但是采用分类器准确率为评估标准的 ReSBSW 算法得到的测试准确率高于 ReCorre 算法，再参考 Relief-Wrapper 和 GA-Wrapper 实验比较的结果，一方面说明 Relief 过滤无关特征的步骤误差较大，另一方面说明特征选择算法中特征子集的评估应考虑分类器算法的性能。

3. 考虑到 Relief 评估性能较好但有偏差，以及 GA-Wrapper 算法准确率高但在维数高的数据集上运行效率低的特点，本章提出了 Relief 和遗传算法耦合的特征选择算法，算法中 Relief 的评估结果为遗传算法的种群初始化提供启发信息，为遗传算法提供好的搜索起点，以加快遗传算法搜索到近似最优解的效率。实验分析表明，从分类准确率，特征子集大小以及运行时间等多角度考察，Relief_GA-Wrapper 取得了较好的综合性

能。

本章所有工作都是基于 Relief 算法展开的，但是事实上，本文只是提出了几种 Filter 和 Wrapper 的组合模式，本文在利用 Filter 信息加速 Wrapper 搜索的思路基础上提出了 Relief-GA-Wrapper 算法，这是因为 Relief 是公认的一种性能较好的特征评估算法，遗传算法进行特征选择也是一种公认较好的算法，在一个高的起点上开展工作更容易取得成效。当然，研究者也可以考虑将其他的特征评估方法(如信息增益、Gini-index 等)与某种搜索算法按本文思路进行结合。

Relief-GA-Wrapper 算法终止时得到遗传算法的最后一个种群，只是本文选择了其中评估函数最高的个体作为特征选择的最终结果，设想如果分别利用多个特征子集训练得到不同的分类器，然后用这些分类器的集成对未知样本进行分类，也许会得到更好的性能。基于此想法，本文开展了在基于特征选择的集成算法方面的研究，具体将在下一章详细介绍。

第四章 一种基于特征选择的适于高维数据的集成学习算法

4.1 引言

在上一章，我们对高维数据的特征选择进行了讨论，提出并实验了几种基于 Relief 的组合式特征选择方法，其中采用 Relief 特征评估信息指导遗传算法种群初始化，从而加速搜索较优解过程的 Relief-GA-Wrapper 算法从测试准确率，降维性能和运行效率各方面取得了较好的综合性能。Relief-GA-Wrapper 算法中最后遗传算法给出的是一个种群结果，只是我们从中选择了适配值最高的个体作为特征选择的结果，而抛弃了其他个体。我们设想，是否其他个体也会提供有用的信息，如果在不同的个体（即特征子集）上训练得到不同的个体分类器，然后将这些分类器组合起来，是否会得到很好的集成分类结果。以上想法，正是本章研究的最初动机，即研究基于特征选择的集成学习算法。

本章将首先介绍集成学习算法的定义、典型集成学习算法以及集成学习的理论分析。然后介绍在 Relief-GA-Wrapper 的结果种群上的集成学习结果，最后重点介绍我们提出的基于两步式特征选择的集成学习算法 ReFeatEn(Relief Feature based Ensembles)。

4.2 集成学习算法综述

4.2.1 集成学习算法的定义

集成学习是近年来机器学习领域的研究热点之一，通过训练多个学习系统并将其结果按一定方式进行集成，可以显著提高学习系统的泛化能力，被视为具有广泛应用前景的计算技术[Dietterich 1995][Dietterich 2000a][Dietterich 2000b][Bauer 1999][周志华 2002]。

高维数据的分类是模式识别领域公认的难题，高维数据集通常具有特征维数高，特征间关系复杂，满足的求解函数空间大，计算开销大，易出现过学习的特点，要构造单一的性能较好的分类器很困难[Jain 1987a]。面对高维问题，研究者通常首先利用特征选择算法降维，从而降低问题复杂度。但正如前两章所述，高维数据的降维本身就是一个很困难的问题，而且降维可能会丢失

有用的特征。集成学习方法为解决高维数据的学习问题提供了一种有效的解决办法。

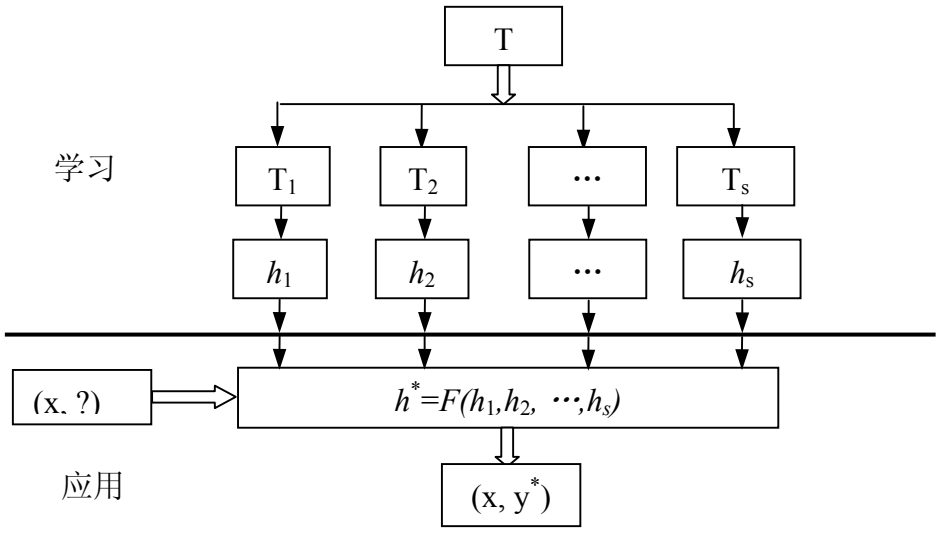

图4-1 集成学习分类器

图 4-1 为包含 S 个个体分类器 h_1, h_2，…，h_s 的集成分类器的基本框架，该框架包含两部分：（1）学习部分；（2）应用部分。在学习部分，生成个体分类器 h_1,h_2，…，h_s，常用方法如从训练集 T 中产生训练集 T_k，k=1,…,S, 利用每个 T_k 训练得到 h_k。在应用部分，个体分类器通过某种方式组成 $h^*=h^*=F(h_1,h_2, \cdots,h_s)$，测试样本 x 由 h^*进行分类。个体分类器的集成主要有两种方式:一、组合法，即每个个体分类器都对测试样本进行分类，最后结果由所有个体分类器的分类结果组合而成；二、选择法，即从所有个体分类器中选择一个，并将选择的个体分类器的输出作为集成分类器的输出。本文研究集中于个体分类器的产生方法，关于个体分类器的集成则采用了使用较多的组合法。

集成学习算法根据个体产生办法和集成手段的不同，大体可分为四类：

1. 对训练样本分布进行操作，可称为 Pattern-Level，如 Bagging，Boosting 以及交叉验证法(Cross-Validation)等[Breiman 1996][Freund 1995][Hansen 1990]。
2. 对特征空间进行操作，可称为 Feature-Level，如 Ho[Ho 1998]提出的随机子空间集成学习方法等。

3. 对类别进行操作，可称为 Class-Level，如 Ditterich [Dietterich 1995]提出的输出编码错误纠正法，该方法适合于类别数大于 2 的数据集。

4. 对训练过程进行随机扰动。如神经网络的随机产生多轮初始权值法，以及决策树的各结点特征随机选择法[Kwok 1990] [Dietterich 2000a]等。

下面将对本文涉及较多的典型集成学习算法进行介绍。

4.2.2 典型的集成学习算法介绍

4.2.2.1 Bagging

Bagging[Breiman 1996]的基础是可重复取样(Bootstrap Sampling)[Efron 1993]，Bootstrap 方法通常用于估计小样本集的错误率[Efron 1982][Jain 1987b][Weiss 1991]。Breiman 最先将 Bootstrap 方法用于集成学习中，称为 Bootstrap aggregation，简称为 Bagging。 图 4-2 为 Bagging 算法的流程图。产生 T 个 Bootstrap 的样本集 B_1，B_2,…,B_T，在每个样本集 B_i 上训练得到分类器 C_i，最后通过分类器 C_1，C_2，…，C_T 的相对多数投票法集成得到分类器 C^*。

输入：训练集 S，基分类器 I，循环代数 T

1. for i =1 to T\{
2. 从 S 中得到一个 Bootstrap 样本集 S'
3. $C_i = I(S')$
4. \}
5. $C^*(x) = \arg\max_{y} \sum_{i:C_i(x)=y} 1$ //返回得票最多的类别

输出：分类器 C^*

图4-2 Bagging 算法

在等概率的重复采样产生的 Bootstrap 样本集中，原始训练集中某些样本可能在新的训练集中出现多次，而另外一些样本则可能一次也不出现。假设有

m 个样本，则当可重复采样时，一个样本至少被选中一次的概率为 $1-(1-\frac{1}{m})^m$，当 m 较大时，其值接近于 1-1/e = 63.2%,也就是说每轮重复取样平均包含原训练样本集的 63.2%样本[Ditterich 2000]。Bagging 方法通过重新选取训练集增加了集成个体间的差异度，从而提高了泛化能力。Breiman[Breiman 1998]将此类算法称为 P&C (Perturb and Combine) 族算法。[Breiman 1996]指出，稳定性是 Bagging 能否发挥作用的关键因素，Bagging 能提高不稳定学习算法的预测精度，而对稳定的学习算法效果不明显，有时甚至使预测精度降低。学习算法的稳定性是指如果训练集有较小的变化，学习结果不会发生较大变化，例如，k 最近邻方法、Naive Bayes 等是稳定的分类算法，而决策树、神经网络等方法是不稳定的分类算法。目前 Bagging 也有很多种变体，例如在扰动训练集时不进行重取样，而是对各示例的权加入零均值高斯噪音的 Wagging(Weight Aggregation)［Bauer 1999］。

4.2.2.2 Boosting

Kearns 和 Valiant[Kearns 1988]指出，在 PAC 学习模型[Anthony 1997]中，若存在一个多项式级学习算法来识别一组概念，并且识别正确率很高，那么这组概念是强可学习的；而如果学习算法辨别一组概念的正确率仅比随机猜测略好，那么这组概念是弱可学习的。Kearns 和 Valiant 提出了弱学习算法与强学习算法的等价性问题，即是否可以将弱学习算法提升成强学习算法。如果两者等价，那么在学习概念时，只需找到一个比随机猜测略好的弱学习算法，就可以将其提升为强学习算法，而不必直接去找通常情况下很难获得的强学习算法。

上述等价性问题可视为 Boosting 系列算法的出发点。1990 年，Schapire[Schapire 1990] 通过一个构造性方法对该问题作出了肯定的证明，其构造过程即称为 Boosting，并由 Freund 进行了改进[Freund 1995]，和 Bagging 方法不同，Boosting 方法的各个体分类器是串行式产生的，即第 n 个分类器赖以进行训练的训练集合依赖于前 n-1 个分类器在训练样本集上的性能。但是 Schapire[Schapire 1990] 和 Freund[Freund 1995]的算法在解决实际问题时有一个缺陷，即它们都要求事先知道弱学习算法学习正确率的下限，这在实际问题中

很难做到。1997 年，Freund 和 Schapire[Freund 1997] 提出了 AdaBoost (Adaptive Boost) 算法， 该算法的效率与 Freund 算法[Freund 1995]很接近，却可以非常容易地应用到实际问题中。因此，该算法已成为目前最流行的 Boosting 算法。如图 4-3 所示，Adaboost 维持训练样本集的一个权值分布，训练样本的初始权值均为 1，然后训练得到分类器，根据分类器对训练样本分类的正误以及本轮的训练集上的加权错误率更新样本权值，使得被错分的样本权值增加，从而使下一轮的分类器训练时努力使分类错误的样本分类正确。最后集成分类器通过分类器集合的加权投票得到，训练错误率低的个体分类器在最后投票中占较高的权重。

输入：S—训练样本集，大小为 m；I— 分类器；T—循环代数

1. 拷贝 S 得到新训练样本集 S' 设定 S' 中每个样本权重为 1
2. 初始化 i=0
3. 在加权后的样本集 S' 上训练得到分类器 C_i
4. 利用下面公式得到训练集上的加权错误率 ε_i

$$\varepsilon_i = \frac{1}{m} \sum_{X_j \in S' : C_i(X_j) \neq Y_j} weight(X_j)$$

若 $\varepsilon_i > 1/2$ 或者 $\varepsilon_{i=0}$，则用可重复采样方法从 S 中产生新 S'，初始化每个样本权重为 1，返回(3)

5. $\beta_i = \varepsilon_i / (1-\varepsilon_i)$
6. 对 S' 中每个样本 X_j，若 C_i 对之分类正确，则更新权值 weight(X_j) = weight(X_j). β_i
7. 归一化样本权值，使所有样本的权值和为 m
8. i++，当 i 小于循环代数 T 时，重复执行(3)到(8)的操作
9. 按下式得到集成分类器 C^* $C^*(X) = \arg\max_{y \in Y} \sum_{i:C_i(X)=y} \log \frac{1}{\beta_i}$

结束

图4-3 Adaboost 算法

值得注意的是，虽然 Boosting 方法能够增强集成算法的泛化能力，但是同时也有可能使集成过分偏向于某几个特别困难的样本。因此，该方法不太稳

定，有时能起到很好的作用，有时却没有效果[Schapire 1990]，甚至会发生加入新的个体分类器，集成分类器准确率下降的情况[Wickramaratna 2001]。

Boosting 方法有两种不同的使用方式, 即使用带权的样本和按概率重取样本，Quinlan[Quinlan 1996] 通过实验发现, 前者效果优于后者。1996 年，Breiman[Breiman 1998]提出了 Arcing (Adaptive Resample and Combine) 的概念，认为 Boosting 是 Arcing 算法族的一个特例。在此基础上，他设计出 Arc-x4 算法，该算法在产生新的个体分类器时，样本的权的变化与已有的所有个体分类器都有关。有趣的是，Bauer 和 Kohavi[Bauer 1999]通过实验发现，与 AdaBoost 相反，按概率重取样本的 Arc-x4 优于使用带权样本的 Arc-x4。

Bagging 类算法与 Boost ing 类算法的主要区别在于：

- Bagging 的训练集的选择是随机的，各轮训练集之间相互独立，而 Boosting 的训练集的选择不是独立的，各轮训练集的选择与前面各轮的学习结果有关。
- Bagging 的各个预测函数没有权重，而 Boosting 是有权重的。
- Bagging 的各个预测函数可以并行生成，而 Boosting 的各个预测函数只能顺序生成。对于像神经网络这样极为耗时的学习方法，Bagging 可通过并行训练节省大量时间开销。
- 另外，一些研究者[Opitz 1999a] [Bauer 1999]发现，一般情况下，Bagging 方法总是可以改善学习系统的性能；而 Boosting 方法在不同数据集上性能很不同，由于 Boosting 方法训练中努力将边界的样本分对，因此对噪声较敏感。在有效时效果比 Bagging 还好，但在无效时却可能使学习系统的性能恶化。

值得注意的是，Boosting 和 Bagging 的轮数并非越多越好，实验[Opitz 1999a]表明, 学习系统性能的改善主要发生在最初的若干轮中。

4.2.2.3 基于特征选择的集成学习方法

构造集成学习分类器的另一类技术是通过对选择不同的特征子集实现的，也有研究者称为子空间法[Ho 1998]，不同的特征子集即构成不同的子空间。

如图 4-4 所示，基于特征选择的集成学习分类器通过选择不同的特征子集，训练得到不同的个体分类器，进而得到集成学习分类器。同基于样本处理的集成学习比较而言，基于特征处理的集成学习研究及应用都较少。这是由于，研究者认为这种方法只适合特征间高度冗余的情况。Tumer 在[Tumer 1996]文中采取保留那些和某类别相关度较大的特征的方法进行集成，该方法在 Sonar 数据上的实验表明即使只去掉很少的特征也会导致单个分类器性能的下降从而使得集成分类器的性能降低。Tumer 采用的基分类器为神经网络，我们在 Sonar 数据上的实验表明，通过随机特征选择和 Relief 特征选择得到的决策树集成分类器的测试错误率可以比采用全部特征的决策树降低 36%，NBC 的集成分类器的错误率可以比 NBC 降低 4.8%，最近邻中心法的集成分类器的错误率可以比最近邻中心法降低 10.0%（参看表 4-3，4-4，4-5）。同样本文在其他数据上的实验结果也验证了基于特征选择的集成学习的有效性。当然，考虑特征集合中只存在一个相关特征而其余特征均为无关特征的极端情况，那么基于特征选择的集成学习的性能一定不乐观。但实际应用尤其是高维问题中，特征间的关系很复杂，特征间相互作用也极为常见，常常难以准确辨识出特征是否和目标函数相关 ，寻找一个单一的好的假设函数很困难，这时基于特征选择进行集成学习对高维复杂特征的模式识别问题是一种很有潜力的方法。

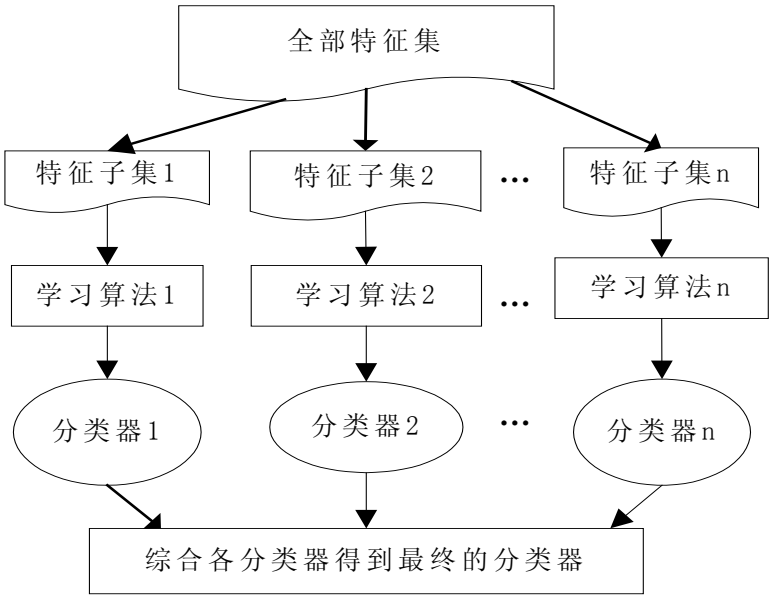

图4-4 基于特征选择的集成学习分类器

[Ho 1998]年研究了随机特征选择的集成决策树分类器，即随机地选择一定数目的特征得到不同的特征子集，在经过不同特征子集过滤过的数据上训练得到不同的个体分类器，通过某种个体分类器的投票法得到集成学习分类器。随机特征选择算法需要确定选择的特征子集的大小。[Ho 1998]根据经验确定特征子集大小为原特征集合的一半。实验表明该方法可以取得和 Bagging 和 Boosting 相当的性能。几个研究者提出采用遗传算法搜索特征子集的特征选择集成学习算法[Optiz 1996b] [Tsymbal 2001] [Guerra-Salcedo 1999]，在[Optiz 1996b]中，采用遗传算法寻找最优的集成个体神经网络，遗传算法的个体评估函数既考虑了个体分类器的分类准确率(在训练集(Train Set)上或者验证集(Validation Set))也考虑个体分类器和已生成的集成算法间的差异度，Optiz 通过实验声明该个体生成算法有效。Optiz 和 Guerra-Salcedo 通过观察各个体分类器在训练集或者校验集(Validation Set)上的准确率和差异度来选择特征子集，而[Ho 1998]是随机地特征特征子集。本文提出的 ReFeatEn 算法介于两者之间，在特征子集产生过程中未涉及分类器算法，但利用特征和目标函数的相关信息引导特征选择，本章第四小节将对该算法予以详细介绍。

4.2.2.4 其他个体生成方法

此外还存在多种个体生成方法，出于篇幅以及不是本文研究重点，仅在此处列出而不加以介绍，有兴趣的读者可以参考相关文献。

- 交叉验证法[Krogh 1995]。
- 超类/子类法：专家系统集成，输出编码校正法[Dietterich 1995] [Ricci 1997]。
- 对训练过程进行扰动：如神经网络的随机产生多轮初始权值法[Maclin 1995]，Parmanto 等比较了这种方法和 Bagging 以及交叉验证法，结论是交叉验证法性能最优，Bagging 其次，随机初始权值法最差[Parmanto 1995]。

4.2.3 结论生成方法

结论生成方法通常随分类器输出类型不同而不同。一般的，分类器的输出

信息有以下三类：

1）单一的决策值：分类器只输出一个标记，如类别。

2）排序序列：分类器输出为类别标记的一个序列，排在最前的为第一选择，例如 NBC 可以按照各类别的后验概率对类别标记进行排序。

3）度量：分类器对每个类别标记赋予一个度量值，表示输入样本属于该类别的程度。如：NBC 中各类别的后验概率，最近邻中心法中的距离值等。

在以上三类输出信息中，度量含有的信息最多，单一的决策值含的信息量最少。从度量信息，我们可以得到排序序列，进而可以得到单一的决策。单一的决策是最具有普遍性的输出，本文只涉及到分类器只输出单一决策值的结论生成方法。集成分类器的输出通常由个体分类器的输出投票产生。通常采用绝对多数投票法(某类别为最终决策结果当且仅当有超过半数的神经网络输出结果为该类别) 或相对多数投票法(某类别成为最终结果当且仅当输出结果为该类别的神经网络的数目最多)。理论分析和大量试验表明[Hansen 1990]，后者优于前者。因此，在对分类器进行集成时，目前大多采用相对多数投票法。研究人员发现，简单的平均(对回归问题)或者相对多数投票法（对分类问题）是一种有效的组合机制[Alpaydin 1993] [Breiman 1998] [Krogh 1995] [Battiti 1994b]。平均或者投票法又涉及到各个体分类器的结果是否加权。Perrone 等人[Perrone 1993]认为, 采用加权平均可以得到比简单平均更好的泛化能力。但是，也有一些研究者[Optiz 1996a] 认为，对权值进行优化将会导致过配(Overfitting)，从而使集成的泛化能力降低，因此，他们建议使用简单平均。Breiman 的实验结论是加权与否对集成学习结果影响并不大[Breiman 1998]。本文的结论生成方法采用了相对多数投票法。

4.2.4 集成学习的理论分析

Hansen 和 Salamon 1990 年以神经网络集成学习算法为例，分析指出集成学习比个体分类器性能优越的充分必要条件是：1.个体分类器准确；2.个体分类器间存在较大的差异[Hansen 1990]。所谓个体分类器准确，意思是个体分类

器可以取得比随机猜测好的性能。个体分类器存在较大的差异，即个体分类器间在样本空间的错误分布不同，显然，如果个体分类器在样本空间的错误分布相同，那么其集成结果还是得到相同的错误分布，根本不可能改善性能。1995 年 Krogh 和 Vedelsby 给出了集成学习泛化误差计算公式[Krogh 1995]。2000 年 Dietterich 从统计、计算和表示三方面定性分析了集成学习算法的有效的原因[Dietterich 2000a]。下面将首先对以上理论分析予以概要介绍，然后介绍几种衡量集成学习个体分类器的差异度的表示方法。

4.2.5 结论生成方法分析

1990 年， Hansen 和 Salamon 证明， 对神经网络分类器来说， 采用集成方法能够有效提高系统的泛化能力。 假设集成由 N 个独立的分类器构成， 采用绝对多数投票法， 再假设每个分类器以 1- p 的概率给出正确的分类结果，并且分类器之间错误不相关, 则该集成学习发生错误的概率为：

$$p_e = \sum_{k>N/2}^{N} \binom{N}{K} p^k (1-p)^{N-k} \qquad \text{公式(4-1)}$$

在 p < 1/2 时， p_e 随 N 的增大而单调递减。 因此， 如果每个分类器的预测精度都高于 50%， 并且各分类器之间错误不相关， 则分类器集成中的分类器数目越多， 集成的精度就越高。 当 N 趋向于无穷成的错误率趋向于 0。 在采用相对多数投票法时， 分类器集成的错误率比公式(4-1) 复杂得多， 但是 Hansen 和 Salamon 的分析表明，采用相对多数投票法在多数情况下能够得到比绝对多数投票法更好的结果。

在实际应用中，由于各个独立的分类器并不能保证错误不相关， 因此，分类器集成的效果与理想值相比有一定的差距， 但其提高泛化能力的作用仍相当明显。1993 年, Perrone 和 Cooper 证明， 在将神经网络集成用于回归估计时，如果采用简单平均，且各网络的误差是期望为 0 且互相独立的随机变量，则集成的泛化误差为各网络泛化误差平均值的 1/N [Perrone 1993]。其中 N 为集成中网络的数目；如果采用加权平均，通过适当选取各网络的权值，能够得到比采用简单平均法更好的泛化能力。

1996 年，Sollich 和 Krogh 指出，在神经网络集成的规模较大，即个体网络较多时，对结论的权进行优化没有好处，适于使用简单平均等结论合成方法；而在神经网络集成的规模较小，即个体网络较少，或者数据集中噪音较多时，对结论的权进行优化将提高学习系统的泛化能力[Sollich 1996]。

1995 年，Krogh 和 Vedelsby 给出了神经网络集成泛化误差计算公式[Krogh 1995]。假设学习任务是利用 N 个神经网络组成的集成对 f：Rn→R 进行近似，集成采用加权平均，各网络分别被赋以权值 w_α，并满足公式(4-2) 和(4-3)。

$$w_\alpha > 0 \qquad \text{公式(4-2)}$$

$$\sum_\alpha w_\alpha = 1 \qquad \text{公式(4-3)}$$

再假设训练集按分布 p(X) 随机抽取，网络 α 对输入 X 的输出为 $V^\alpha(X)$，则神经网络集成的输出为

$$\bar{V}(X) = \sum_\alpha w_\alpha V^\alpha(X) \qquad \text{公式(4-4)}$$

神经网络 α 的泛化误差 E^α 和神经网络集成的泛化误差 E 分别为

$$E^\alpha = \int (f(x) - V^\alpha(X))^2 p(X) dX \qquad \text{公式(4-5)}$$

$$E = \int (f(X) - \bar{V}(X))^2 p(X) dX \qquad \text{公式(4-6)}$$

各网络泛化误差的加权平均为

$$\bar{E} = \sum_\alpha w_\alpha E^\alpha \qquad \text{公式(4-7)}$$

神经网络 α 的差异度 A^α 和神经网络集成的差异度 $\bar{A}$ 分别为

$$A^\alpha = \int (V(X) - \bar{V}(X))^2 p(X) dX \qquad \text{公式(4-8)}$$

$$\bar{A} = \sum_\varepsilon w_\alpha A^\alpha \qquad \text{公式(4-9)}$$

则神经网络集成的泛化误差为

$$E = \overline{E} - \overline{A} \qquad \text{公式(4-10)}$$

公式（4-10）中的 $\overline{A}$ 度量了神经网络集成中各网络的相关程度，若集成是高度偏向(biased)的，即对于相同的输入，集成中所有网络都给出相同或相近的输出，此时集成的差异度接近于 0，其泛化误差接近于各网络泛化误差的加权平均。反之，若集成中各网络是相互独立的，则集成的差异度较大，其泛化误差将远小于各网络泛化误差的加权平均。因此，要加强分类器集成的泛化能力，就应该尽可能地使集成中各个体分类器尽可能不相关。

近似的，Tumer 和 Ghosh 推导出集成分类器的泛化误差如公式(4-11)所示[Tumer 1996]，其中 E 为集成分类器的泛化误差，$\overline{E}$ 为个体分类器的平均泛化误差，S 为个体分类器的个数，δ_i 为个体分类器间在预测类别 i 上的相关度(Correlation)。可见当个体分类器的相关度为 0 时，集成分类器的泛化误差为个体分类器平均泛化误差的 1/S。

$$E = \frac{1 + (S-1)\sum_{i=1}^{L} P_i \delta_i}{S} \cdot \overline{E} \qquad \text{公式(4-11)}$$

4.2.6 个体生成方法分析

1997 年，Freund 和 Schapire 以 AdaBoost 为代表，对 Boosting 类方法进行了分析[Freund 1997]，并证明该类方法产生的最终预测函数 H 的训练误差满足公式(4-12)，其中 ε_t 为预测函数 h_t 的训练误差，$\gamma_t = \frac{1}{2} - \varepsilon_t$。

$$\varepsilon \le 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t (1-\varepsilon_t)} = \prod_{t=1}^{T} \sqrt{1-4\gamma_t^2} \le \exp(-2\sum_t \gamma_t^2) \qquad \text{公式(4-12)}$$

从公式(4-12) 可以看出，只要学习算法略好于随机猜测， 训练误差将随 t 以指数级下降。在此基础上，Freund 和 Schapire[Freund 1997]用 VC 维[Vapnik 1995]对 Boosting 的泛化误差进行了分析。设训练例为 m 个，学习算法的 VC 维为 d，训练轮数为 T，则其泛化误差上限如公式(4-13) 所示，其中 $\hat{P}(\cdot)$ 表示对训练集的经验概率。

$$\hat{P}(H(x)\neq y)+\hat{O}\left(\sqrt{\frac{Td}{m}}\right) \qquad \text{公式(4-13)}$$

公式(4－13) 表明， 若训练轮数过多，Boosting 将发生过配。 但大量试验表明，Boosting 即使训练几千轮后仍不会发生过配现象，而且其泛化误差在训练误差已降到零后仍会继续降低。为解释这一现象，1998 年 Schapire 等人从边际(margin)的角度对泛化误差进行了分析[Schapire 1998]。边际 margin (x , y) 定义为

$$m\arg in(x,y)=y\sum_{i=1}^{n}\alpha_i h_i(x) \qquad \text{公式(4-14)}$$

正边际表示正确预测，负边际表示错误预测，较大的边际可信度较高，较小的边际可信度较低。如图 4-5 所示，假设存在两个不同的类别的数据点，若以 h_1 为划分超平面，则两个分类的最小边际为 d_1，若以 h_2 为划分超平面，则两个分类的最小边际为 d_2。显然，如果 $d_2 > d_1$，则 h_2 是比 h_1 更好的划分超平面，因为其分类鲁棒性更好。

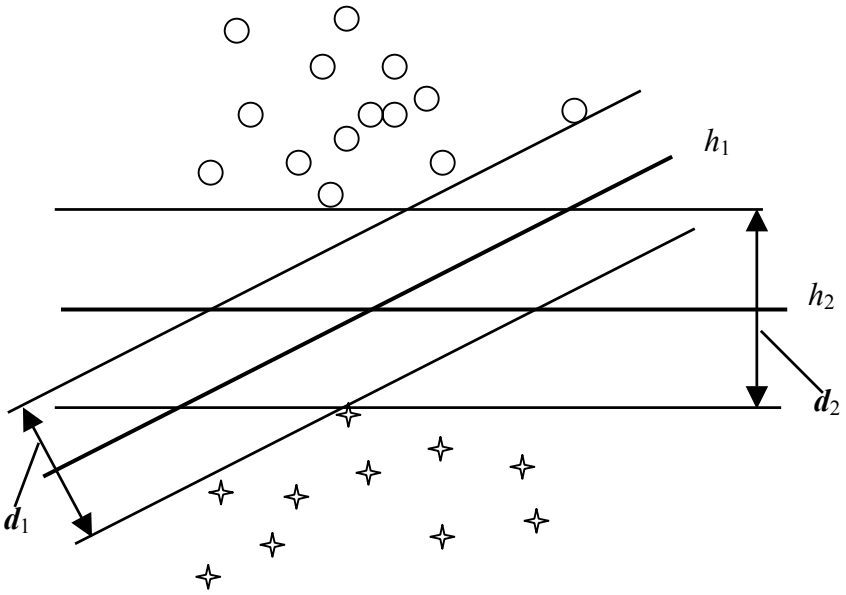

图4-5 不同的划分平面及其边际

Schapire 等人[Schapire 1998] 认为，在训练误差降为零后，Boosting 仍会改善边际，即继续寻找边际更大的划分超平面，这就使得分类可靠性得到提

高，从而使泛化误差得以继续降低。进一步，Schapire 等人还具体地给出了泛化误差的上限：

$$\hat{P}_r[m\arg in(x,y)\le\theta]+\hat{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) \qquad \text{公式(4-15)}$$

从公式(4-15) 可以看出，Boosting 的泛化误差上限与训练轮数无关，[Schapire 1999]的一些实验也证实了这一点。然而，1998 年 Grove 和 Schuurmans [Grove 1998] 指出，Schapire 等人的边际假说并不能真正解释 Boosting 成功的原因。为证明这一点，他们在 AdaBoost 的基础上设计了 LPBoost 算法，通过线性规划来调整各预测函数的权重，从而增大最小边际。他们指出，如果边际假说成立，那么 LPBoost 算法产生的学习系统泛化误差应比较小。然而实验表明，该学习系统的泛化误差并不小，也就是说，边际的增大并不必然导致泛化误差的减小，有时甚至造成泛化误差增大。因此，关于 Boosting 为什么有效，目前仍然没有一个被广泛接受的理论解释。

1996 年，Breiman[Breiman 1996a] 对 Bagging 进行了理论分析。他指出，分类问题可达到的最高正确率以及利用 Bagging 可达到的正确率分别如公式(4-16) 和(4-17) 所示，其中 C 表示序正确(order correct)的输入集，C' 为 C 的补集，I(*)为指示函数(indicator function)。

$$r^* = \int mas_j\, p(j\mid x)p_x(x) \qquad \text{公式(4-16)}$$

$$r_A = \int_{x\in C} \max_j p(j\mid x)p_x(dx) + \int_{x\in C'}\left[\sum_j I(\phi_A(x)=j)p(j\mid x)\right]p_x(x) \qquad \text{公式(4-17)}$$

显然，Bagging 可使序正确集的分类正确率达到最优，单独的预测函数则无法做到这一点。对回归问题,Breiman 推出公式(4-18) , 不等号左边

$$|E_L\,\varphi(x,L)|^2 \le E_L\varphi^2(x,L) \qquad \text{公式(4-18)}$$

除此之外，Breiman[Breiman 1996b] 还从偏向(bias) 和差异(variance) 的角度对泛化误差进行了分析。他指出，不稳定预测函数的偏向较小，差异较大，Bagging 正是通过减小差异来减小泛化误差的。在此之后，Wolpert 和 Macready[Wolpert 1999]具体地给出了泛化误差、偏向和差异之间的关系：

$$E(C \mid f,m,q) = \sum_{d} P(d \mid f,m)(h_d(q) - f^*(q))^2 \qquad \text{公式(4-19)}$$
$$= (h^*(q) - f^*(q))^2 + \sum_{d} P(d \mid f,m)(h_d(q) - h^*(q))^2$$

公式(4-19) 左边为泛化误差，右边第一项为偏向的平方，第二项为差异。Bagging 就是对 $h^*(q)$ 进行模拟，使得在偏差相同的情况下差异尽量趋向于零。值得注意的是，虽然利用偏向和差异来解释 Bagging 获得了一定的成功，但 Freund 和 Schapire[Freund 1998]通过一系列基于 Stumps 和 C4.5 的实验指出，偏向和差异并不能很好地解释 Boosting。

4.2.7 个体分类器间的差异度量

由前面的分析可知，个体分类器间差异度对集成学习性能影响很大。本节介绍几种个体分类器的差异度量方法。

常用的个体分类器的差异度有以下几种度量方法：Q 统计法[Yule 1990]，不一致度度量法[Opitz 1999b][Ho 1998]，熵度量法[Cunningham 2000]。

- Q 统计

  对两个分类器 C_i 和 C_j，Q 统计值定义如下：

$$Q_{i,j} = \frac{ad - bc}{ad + bc} \qquad \text{公式(4-20)}$$

其中 a 为两个分类器同时分类正确的概率，d 为两个分类器同时分类错误的概率，b 为分类器 C_i 分类正确而 C_j 分类错误的概率，c 为分类器 C_i 分类错误而 C_j 分类正确的概率。$\overline{Q}$ 为分类器间 Q 统计的均值。Q 的取值范围为-1 到 1 之间，如果两分类器独立，则 Q 为 0。Q 的绝对值越大，分类器间的差异度越小。

分类器间的差异度为两两分类器间的 Q 统计的均值，如公式 4-21 所示。

$$\overline{Q} = \frac{2}{n(n-1)} \sum_{i \neq j} Q_{i,j} \qquad \text{公式(4-21)}$$

- 不一致度量

不一致度量是所有差异度计算里边应用最多的，也是本文采用的分类器差

异度度量方法。

Ho 采用如下公式度量分类问题中两个分类器差异度：

$$D_{i,j}=\frac{1}{m}\sum_{k=1}^{m}f(X^k) \qquad \text{公式(4-22)}$$

其中

$$f(X^k)=\begin{cases}1 & if\ \ C_i(X^k)\neq C_j(X^k)\\ 0 & else\end{cases} \qquad \text{公式(4-23)}$$

则分类器间差异度为

$$\overline{D}=\frac{2}{n(n-1)}\sum_{i\neq j}D_{i,j} \qquad \text{公式(4-24)}$$

其中 $\overline{D}$ 的取值范围为 0 到 1， $\overline{D}$ 越大，分类器间差异越大。

- 熵度量方法

熵度量方法，即首先度量各分类器在一个样本上分类结果的离散度，然后得到所有样本离散度的均值，定义如下：

$$\overline{A}=\frac{1}{m}\sum_{x=1}^{m}\sum_{k=1}^{C}-P_k^x\log(P_k^x)$$

其中 P_k^x 表示样本 x 被分到类 k 的概率，m 为样本个数，C 为类别个数。

Kuncheva 和 Shipp 等对差异度的各种度量进行了分析和实验[Kuncheva 2003][Shipp 2002]，发现各种度量方法间相关度很大，和集成分类器准确率的关系也近似，我们在后续的算法分析中运用了广为采用的不一致度度量。

集成学习的个体分类器差异度对集成学习的性能非常重要[Cunningham 2000][Kuncheva 2003]，显然，彼此分类相同的分类器即使错分率低也不能综合得到比单个分类器好的性能。产生不同的个体分类器通常有下面三种方法：一.改变训练数据集的样本分布，如 Bagging，Boosting 以及 Cross Validation 方法[Breiman 1996a] [Freund 1995] [Hansen 1990]；二，改变训练数据的特征分布[Ho 1998]；三，改变分类器训练参数或者分类器结构，如 Opitz 的神经网络集成通过设置不同的神经元初始权值得到不同的神经网络[Opitz 1999a]。同时个

体分类器的平均准确率也很重要，取极端情况，性能都低于随机猜测的个体分类器的集成通常会取得比单个个体分类器更差的性能。而差异度和平均准确率又存在一定的矛盾，当个体分类器的个数增多时，个体分类器间的差异度常常会变低，如何取得两者间较好的平衡，正是本文努力实现的目标。

4.3 Relief-GA-Wrapper 的副产品—ReGAWrapperEn

上一章提出的 Relief-Ga-Wrapper 特征选择算法最后给出的是一个种群，即多个特征选择方案，我们只选择了其中适配值最高的个体作为最终的特征选择结果。那么我们设想，其余的特征选择方案也含有有用信息，是否可以利用所有这些特征选择方案得到更好的分类结果？这就得到了 Relief-GA-Wrapper 算法的副产品—基于 Relief-GA-Wrapper 特征选择的集成学习算法 ReGAWrapperEn(Relief-GA-Wrapper Ensemble)的思路。

ReGAWrapperEn 算法是 Relief-GA-Wrapper 特征选择算法的副产品：在 Relief-GA-Wrapper 遗传算法最后一代的种群上，也即不同的特征子集上，训练得到不同的个体分类器，这些个体分类器利用相对多数投票法得到集成分类器。表 4-1 列出了 C4.5 决策树上 25 个个体分类器的集成分类器和标准 C4.5 决策树的测试准确率，最后一列为 ReGAWrapperEn 和 Standard 相比的错误率降低百分比。可见，和标准 C4.5 决策树相比，ReGAWrapperEn 算法的测试准确率有明显提高，错误率最多可降低 44.85%。

表4-1 ReGAWrapperEn 和标准的 C4.5 决策树的测试准确率比较

Dataset	Standard	ReGAWrapperEn	错误率降低百分比
German	71.56	73.70	7.52%
Horse-colic	80.98	85.30	22.71%
Ionosphere	88.04	92.30	35.62%
Isolet	73.18	85.21	44.85%
Satimage	84.9	90.2	35.10%
Sonar	74.19	75.70	5.85%
waveform-40	70.49	73.41	9.89%

4.4 基于两步式特征选择的集成学习算法 ReFeatEn

上文介绍的 ReGAWrapper 算法是 Relief-GA-Wrapper 算法的副产品，我们并没有为集成分类特意产生不同的特征子集，那么如果专门为集成分类产生不同的特征子集也许会取得更好的集成分类效果。Opitz 和 Guerra-Salcedo 研究提出采用遗传算法搜索用以产生不同个体分类器的特征子集[Opitz 1999b] [Opitz 1999c] [Guerra-Salcedo 1999]。[Ho 1998]提出的随机子空间集成学习算法 RandFeatEn 的特征子集完全随机产生，而采用遗传算法搜索特征子集的集成学习算法则严格控制个体分类起的产生，本文试图设计一种介于这两种方法之间的算法，该算法既考虑到保证集成学习性能的因素，同时避免串行产生个体分类器，可以并行运行，以便适合大规模数据，基于上述考虑，本文作者提出了 ReFeatEn 算法。

4.4.1 算法设计思路

如前文介绍，构造好的集成学习算法，可从保证以下两方面着手：一，个体分类器的分类准确率较高；二，个体分类器间在对数据的分类上存在较大的差异[Optiz 1996] [Dietterich 2000a]。对于基于特征选择的集成学习算法，针对第一方面，应该使特征子集和目标函数尽量相关，从而训练得到准确率较高的个体分类器；对第二方面，则应使各特征子集间重合度尽量小，才可能得到差异度较大的个体分类器。基于以上考虑，我们设计了如下两步式特征子集产生方法：第一步，采用 Relief 特征评估方法得到特征和目标函数的相关度，根据相关度依概率产生特征子集，这样产生的特征子集会和目标函数有较大的相关性；第二步，调整各特征子集使特征子集间差异较大，同时尽量保留和目标函数相关度较大的特征。

4.4.2 算法具体介绍

算法的具体步骤如下：

第一步，运行 Relief 算法，然后采用类似遗传算法中的轮盘赌法产生不同的特征子集。这里的不同特征即对应遗传算法中不同的个体，不同的是，在遗传算法的轮盘赌选择中，个体可以重复选择，而我们这里设定，特征只能被选

一次，不可以复制。当然也可以研究特征可以重复选择的情况，本文没有对此进行研究。由于 Relief 得到的相关度可能为负，这样采用轮盘赌法可能会导致某些特征永远无法被选中，为避免这种情况对于相关度为负的值，我们首先将其值改为文中设定的最小值。另外，特征子集的大小设为原特征子集的一半。

第二步，调整各特征子集使特征子集间差异较大。第一步带来的风险是：个体分类器间差异度减小导致集成性能下降。因此我们引入第二步，当新产生的特征子集和以前产生的某特征子集间差异度小于设定的阈值时，对新产生的特征子集特征进行变异，直到满足设定的差异度阈值，同时控制使两特征子集同时选择的特征变异的概率较大，而两特征子集均未选择的特征变异的概率较小，目的是较少的添加无关特征。由于第二步可能进行变异处理，因此，所产生的特征子集的大小是不固定的，而且由于上述根据情况不同对变异概率的控制，越是后面产生的特征子集的大小越小。

输入：S—训练样本集，大小为 m；I— 预设函数；T—循环代数

1．运行 Relief 算法，得到特征权值 W={w_i ,i=1,2,…,n}

2．For i=0; i<=T; i++ GenerateFeatSet(F_i) //产生 T 个大小为 n/2 的特征子集,

3．For i=0; i<=T; i++

4. For j=0;j<=(i-1);j++{

5. While DifferRate(F_i , F_j)<δ , mutate(F_i, F_j);} //控制特征子集间的差异度

6. For i=0; i<=T; i++ C_i=H(F_i, S) //训练得到 T 个个体分类器

7. 按下式得到集成分类器 C^* $C^*(X)=\arg\max_{y\in Y}\sum_{i:C_i(X)=y}1$

结束

图4-6 ReFeatEn 算法

ReFeatEn 算法的整体流程如图 4-6 所示，首先运行 Relief 算法得到特征权值（第 1 行），然后调用 GenerateFeatSet()产生 T 个特征子集 F_i（第 2 行）。接着检查特征子集 F_i 和 F_1，F_2，…，F_{i-1} 的差异度（第 3-5 行），如果 F_i 和 F_j 之间的差异小于给定阈值δ，则调用 mutate(F_i, F_j)对 F_i 进行变异，直到 F_i 和 F_1，F_2，…，F_{i-1} 的差异度大于给定阈值δ。在每个 F_i 映射后得到训练集上训练得到个体分类器 C_i（第 6 行），最后得到集成分类器 C*（第 7 行）。本文中δ

设为 0.5，表示要求 F_i 和 F_j 不同的特征数目要不小于全部特征数目的一半。图 4-7 为依据 Relief 特征评估采用轮盘赌法产生特征子集的子程序。图 4-8 为对特征子集 F_i 进行变异的子程序，我们对 F_i 中 F_i 和 F_j 都包含和都不包含的特征变异概率设置不同，对都包含的特征被去除的特征大些，而对都不包含的特征被选择的概率小些，目的是更多的去除 F_i 和 F_j 都选择了的特征，而较少的向 F_i 中添加 F_i 和 F_j 中都未选择的特征，使得特征子集中包含的无关特征不多，同时后面产生的特征子集更容易满足和前面特征子集间的差异度。

输入：Relief 特征评估结果 W，n –原特征全集的大小

1．For(i=1; i<=n;i++) if (W_i < =0) W_i=MINIMUM;

2. For(i=1;i<=n;i++) $prob[i] = \sum_{j=1}^{i} W_j \Big/ \sum_{j=1}^{n} W_j$; //产生各特征选择概率

3. For(count=0, count<=n/2;count ++)

4. { Prob=RandomFloat(0,1);

5. For(i=1;i<=n;i++)

6. If(Prob>Prob[i-1] && Prob<Prob[i] && i∉F) //采用轮盘赌方法，选择特征

7. F=F∪{i};}

输出：特征集合 F

图4-7 GenerateFeatSet(F)子程序

输入：特征子集 F_i 和 F_j 输出：返回特征子集 F_i

1．While(DifferRate(F_i , F_j)<δ){

2． For(k=0;k<=n;k++){

3． If(F_i[k]= =F_j[k] && F_i[k]= =1){

4． If(RandomFloat(0,1)<PMUTATE1) F_i[k]=0;}

5． else if(F_i[k]= =F_j[k] && F_i[k]= =0) {

6． If(RandomFloat(0,1)< PMUTATE2) F_i[k]=1;}

7． }//end for

8. }//end while

图4-8 Mutate(F_i, F_j)算法 其中 PMUTATE1=0.5，PMUTATE2=0.3，目的是更多的去除 F_i 和 F_j 都选择了的特征，而较少的向 F_i 中添加 F_i 和 F_j 中都未选择的特征

4.4.3 实验结果与分析

4.4.3.1 分类器介绍

本文采用的基分类器有 c4.5 决策树[quinlan 1993]，最近邻中心分类器以及朴素贝叶斯分类器(NBC)。最近邻中心分类器方法从训练集中算得各类的中心向量，待识别样本的类别即同该样本距离最近的中心向量所在的类。朴素贝叶斯分类器是在设定各特征独立前提下得到的贝叶斯分类器。对具有 n 个特征(A1, A2, … , An)的样本 X，其类别为 y 的概率按如下贝叶斯公式估计；P(y丨X)=P(X丨y)×P(y)/P(X)∝P(A1, A2, …, An丨y)P(y)=∏P(Aj丨y)×P(y)。对于离散特征，P(y)以及 P(Aj丨y)通过计数进行估计，对于连续特征，则采用一个高斯分布进行逼近，高斯分布的参数均值和标准偏差从训练集中统计得到。NBC 算法实现简单，而且即使在不满足特征间独立的前提时也能得到很好的效果，并且对无关特征不敏感[Langley 1997]。

4.4.3.2 实验数据

表4-2 实验数据集

No	DataSet	Nom[a]	Num[b]	#Train / #test[c]	#C[d]	Majority error
1	Hanzi_256	0	256	5000/5000	100	99
2	German	13	7	666/334	2	30
3	Horse-colic	15	7	300/68	2	36.91
4	Ionosphere	0	34	234/117	2	35.87
5	Isolet	0	617	1039/520	26	96.15
6	Satimage	0	36	4435/2000	6	76.60
7	Sonar	0	60	215/108	2	33.65
8	waveform-40	0	40	300/4700	3	66.67
9	Sick-euthyroid	18	7	2108/1055	2	90.43

[a] Num-数值型特征的个数，[b] Nom-离散型特征的个数

[c] #Train/#Test-训练样本的数目/测试样本的数目，[d] #C－类别个数

实验数据如表格 4-2 所示。我们选择数据的原则为：

1） 特征数目大于 20，因为我们主要是验证算法在高维数据上的性能。

2） 数据集多样性：特征离散型和连续型共存，样本数目从几百到几

千，特征维数从几十到几百，使得实验结果具有普遍性，类别的数目既有几个的也有上百的，有的数据集没有数据缺失，但有的数据集有大量数据缺失。

4.4.3.3 实验方法

我们的实验目的是：

1. 比较分别采用 C4.5，NBC 和最近邻中心法为基分类器时，ReFeatEn，RandFeatEn，Bagging 和 Boosting 方法的性能。
2. 比较分析 ReFeatEn 和 RandFeatEn 的性能差别和原因。
3. 分析 ReFeatEn 和 RandFeatEn 算法性能同特征子集大小和构成集成系统的个体分类器数目的关系。

下面将分别就以上几方面进行实验比较和分析。在同 Bagging，Boosting 进行比较实验中，我们设定 RandFeatEn 特征子集大小以及 ReFeatEn 方法的第一步特征子集大小均为原始特征集得 50%。Hansen 实验表明集成学习的性能改善最大在最初几个个体分类器集成时最明显[Hansen 1990]，Optiz 在实验里根据经验设定个体分类器的数目 25[Optiz 1999a]，Freund 设定个体分类器数目为 100[Freund, 1996]，本文取折衷设定个体分类器的个数为 50。

4.4.3.4 ReFeatEn,Bagging, Boosting, RandFeatEn 性能比较

为比较以上四种集成学习算法的性能，我们在上述 9 个数据集上，分别以 C4.5，NBC 和最近邻中心为基分类器，设定个体分类器个数为 50，对 Bagging，Boosting，RandFeatEn 和 ReFeatEn 算法进行实验，所得结果为 5 次独立运行的平均值。实验结果如表 4-3，表 4-4，表 4-5 所示，图 4-9，图 4-10，图 4-11 分别是基分类器为 C4.5 决策树、NBC 和最近邻中心分类器时各集成学习算法同单个学习算法的相比的错误率降低百分比。

4.4.3.5 集成学习算法测试准确率比较

- 基分类器为 C4.5 决策树的集成学习算法测试准确率对比

从表 4-3 以及图 4-9 可以看出，以决策树为基分类器的各种集成学习算法

的测试准确率均明显高于单个学习算法。以 ReFeatEn 集成学习算法为例，和标准 C4.5 决策树相比，ReFeatEn 在 hanzi_256 数据集上错误率下降最多，为 88.0%，在 Horse-colic 上错误率下降最少，为 14.9%，在所有数据集上，错误率平均下降了 44.55%，说明集成学习算法对提高决策树的准确率总是很有效的。在全部的 9 个数据集中，ReFeatEn 算法在 6 个数据集上取得了最高的测试准确率， Boosting 算法在 3 个测试集上取得了最高测试准确率，在所有数据集上，ReFeatEn 比 Boosting 算法的平均错误率下降了 9.7%，以上比较说明基于决策树的 ReFeatEn 算法在本文实验的数据集上的性能优于 Boosting 算法。ReFeatEn 算法在 German, Satimage 和 Waveform-40 数据集上准确率低于 Booosting 算法，但在 Staimage 数据集上，两者的准确率非常接近，现在重点考察 Waveform-40 和 German 数据集，Waveform-40 含有 19 个无关特征，German 是所有数据集中特征数目最小的，这在一定程度上说明基于特征选择的集成学习算法 ReFeatEn 在特征较少或无关特征较多的数据上性能一般。

基于特征选择的决策树集成学习在高维数据上取得了显著优于单个分类器的良好效果。这和高维数据特征间关系复杂以及决策树分类器本身的特性都有关系。以脱机手写体汉字识别为例，决策树实质上是一种空间分割法，它不断选取使得信息增益最大的特征进行分割，直到不能通过分割降低熵值为止。手写体汉字的特征间具有相关性，且同一类汉字的不同样本间的特征值差异较大，维数又很高，通过一个决策树的空间分割即使很复杂（如本文标准 C4.5 构造的决策树用到 217 个特征，分割了 664 次）仍很难得到较好的边界。但通过基于不同特征子集的空间分割及加权投票法，每个分类器仅需使用几十个特征就可以使得分类界面更好的逼近实际分类界面，可参看表 4-3 和表 4-12。

- 基分类器为 NBC 的集成学习算法测试准确率比较

如表 4-4 以及图 4-10 所示，以 NBC 为基分类器的各集成学习算法性能在不同数据集上差异很大，并且集成学习算法并不总是能取得高于单个学习算法的测试准确率。在 German，Horse-colic 和 Waveform-40 数据集上，单个学习算法保持着最优性能。Boosting 算法在 Inosphere、Isole、Satimage 和 Sonar 数据集上，和标准 NBC 相比，准确率取得了明显的提高。ReFeatEn 算法在

Ionosphere、Isolet、Satimage、Sonar 和 Sick-euthyroid 数据集取得了高于标准 NBC 的准确率。Bagging 算法由于对原数据分布改变不大，而且为随机抽取，因此对 NBC 这样非常稳定的算法作用很小，Bagging 算法在 7 个数据集上准确率相当于或者低于标准 NBC 算法，仅在 Isolet 和 Sonar 数据集上准确率有较大提高。RandFeatEn 算法由于随机从原特征集合中其抽取 50％的特征，因此它取得的结果不稳定，如果取到好的特征集合，则测试准确率可能较高，但如果取到差的特征集合，则测试准确率会很低，因此平均下来性能一般，除了在 Sick-euthyroid 数据集上准确率显著优于标准 NBC 算法外，在其他数据集上准确率均相当于或者低于标准 NBC。ReFeatEn 平均错误率比 Boosting 降低了 25.7%，这主要由于在 Sick-euthyroid 数据集上，ReFeatEn 取得了远高于 Boosting 的准确率，经分析，发现这是由于 Sick-euthyroid 中最后一个特征在不同类别中分布的严重不平衡导致 NBC 在最后一个特征上的估计差距很大，而 Boosting 算法由于仅仅对样本进行加权，因此对“坏”特征无能为力，而 ReFeatEn 算法则由于刻意选择对分类有用的特征，因此“坏”特征以较小的概率被选择，因此较好的处理了 Sick-euthyroid 数据集。

- 基分类器为最近邻中心法的集成学习测试准确率比较

从表 4-5 和图 1-11 可以看出，总体来说，基于最近邻中心的各种集成学习算法在提高分类器准确率方面效果不明显，和标准最近邻中心分类器比较，除 Horse-colic、Isolet、Sonar 和 Sick-euthyroid 数据集上 Boosting 或者 ReFeatEn 算法和标准最近邻中心分类器相比错误率降低大于 10%外，在其余数据集上，所有集成学习算法的错误率降低均小于 10%甚至错误率高于标准最近邻中心分类器。ReFeatEn 在 Hanzi_256、Horse-colic、Sonar 以及 Sick-euthyroid 数据集上的分类准确率最高，和标准最近邻中心分类器比较，在以上 4 个数据集中，ReFeatEn 在 Sick-euthyroid 数据集上错误率降低最多，为 44.2%，在 Hanzi_256 数据集上错误率降低最少，为 9.42%。ReFeatEn 在所有数据集上的平均错误率比标准最近邻中心法降低了 11.6%。Boosting 算法在 Isolet 和 Satimage 数据上测试准确率最高，和标准最近邻中心分类器比较，在 Isolet 上，错误率降低了 28.2%，在 Satimage 上，错误率降低了 3.4%。ReFeatEn 算法和 Boosting 算法相比，平均错误率降低了 3.0%。

从实验结果可见，采取 C4.5，NBC 和最近邻中心法作为基分类器，ReFeatEn 方法均可取得高于或者相当于 Bagging 和 Boosting 的在测试准确率。另外，本文作者通过在手写体汉字上的实验发现，Bagging 的标准偏差最小，Adaboost 算法的标准偏差明显高于其余三种集成算法。

表4-3 以 C4.5 决策树为基分类器的集成学习算法测试准确率比较

Dataset	Bagging	Boosting	RandFeatEn	ReFeatEn	Standard [a]
Hanzi_256	92.21	94.70	94.66	**95.20**[b]	59.96
German	77.25	**77.84**	76.35	76.95	71.56
Horse-colic	82.35	79.41	82.35	**83.82**	80.98
Ionosphere	94.02	94.87	95.72	**96.58**	88.04
Isolet	87.11	88.85	86.15	**89.23**	73.18
Satimage	90.10	**91.45**	90.80	91.25	84.9
Sonar	80.00	78.15	82.29	**83.49**	74.19
waveform-40	81.19	**81.98**	79.91	80.62	70.49
Sick-euthyroid	98.01	97.85	97.06	**98.10**	97.06
Average [c]	86.92	87.23	87.25	**88.36**	75.41

[a] 为单独采用标准 C4.5 决策树时的测试准确率 [b] 黑体表示所有集成学习算法中的最高值

[c] 各集成学习方法在所有数据集上测试准确率的平均值

表4-4 以 NBC 为基分类器的集成学习算法测试准确率

Dataset	Bagging	Boosting	RandFeatEn	ReFeatEn	Standard [a]
Hanzi_256	93.76	93.84	**94.68** [b]	94.06	97.14
German	76.95	74.55	76.65	**77.25**	77.54
Horse-colic	**80.88**	77.00	79.03	79.94	80.88
Ionosphere	83.76	**91.45**	83.76	85.98	83.76
Isolet	86.92	**88.65**	86.15	87.23	85.96
Satimage	79.55	**81.15**	80.05	80.05	79.65
Sonar	71.43	**81.42**	68.57	71.43	70.00
waveform-40	**80.49**	79.62	80.30	79.47	80.53
Sick-euthyroid	18.58	25.87	73.74	**91.18**	18.58
Average [c]	74.70	77.06	80.33	**82.95**	74.89

[a] 该列为单独采用标准 NBC 时的测试准确率 [b] 黑体表示所有集成学习算法中的最高值

[b] 该行为各集成学习方法在所有数据集上测试准确率的平均值

表4-5 在最近邻中心分类器（NearMean）上的集成学习算法测试准确率

Dataset	Bagging	Boosting	RandFeatEn	ReFeatEn	Standard[a]
Hanzi_256	97.20	97.24	97.24	**97.50** [b]	97.24
German	**68.26**	67.96	66.48	67.96	69.16
Horse-colic	66.18	70.59	70.59	**72.06**	67.65
Ionosphere	**82.05**	80.34	81.19	81.20	80.34
Isolet	83.65	**88.26**	82.88	85.00	83.65
Satimage	78.10	**79.00**	77.80	78.55	78.25
Sonar	72.86	70.00	71.43	**74.29**	71.43
waveform-40	79.66	77.09	**80.19**	79.49	79.66
Sick-euthyroid	62.65	76.97	67.35	**78.58**	61.62
Average [b]	76.73	78.64	77.23	**79.29**	76.56

[a] 为单独采用最近邻中心分类器时的测试准确率 [b] 黑体表示所有集成学习算法中的最高值

[c] 各集成学习方法在所有数据集上测试准确率的平均值

表4-6 基于 NBC 的 Adaboost 和 ReFeatEn 运行时间比较，表中单位为秒

	Ionosphere	Waveform-40	isolet
Adaboost	1.01	3.27	121.33
ReFeatEn	0.43	1.17	15.86

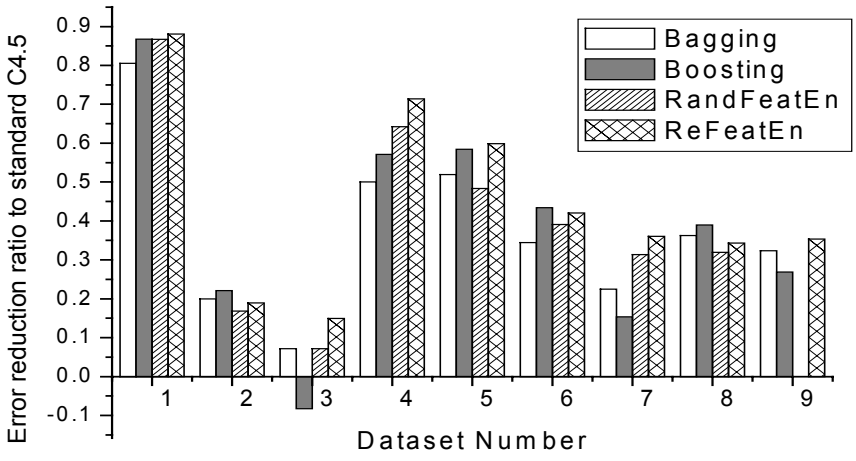

图4-9 基分类器为 C4.5 决策树的集成分类器和标准 C4.5 比较的错误率降低比率

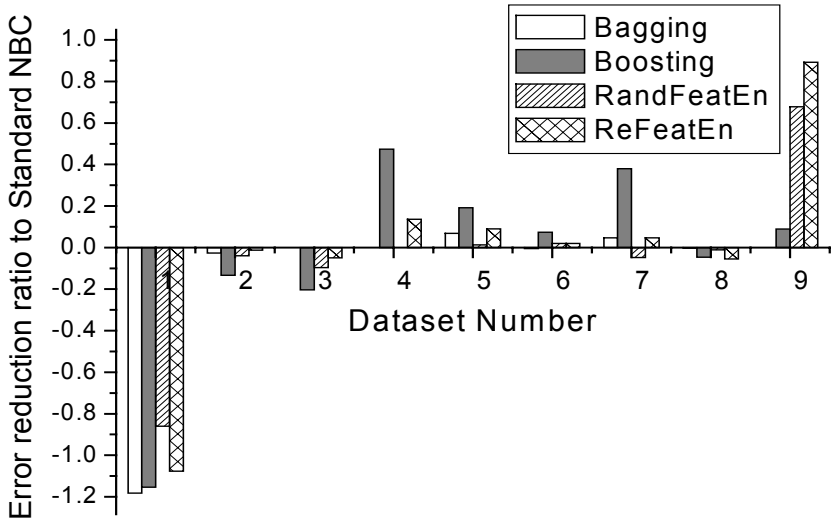

图4-10 基分类器为 NBC 的集成分类器和标准 NBC 比较的错误率降低比率

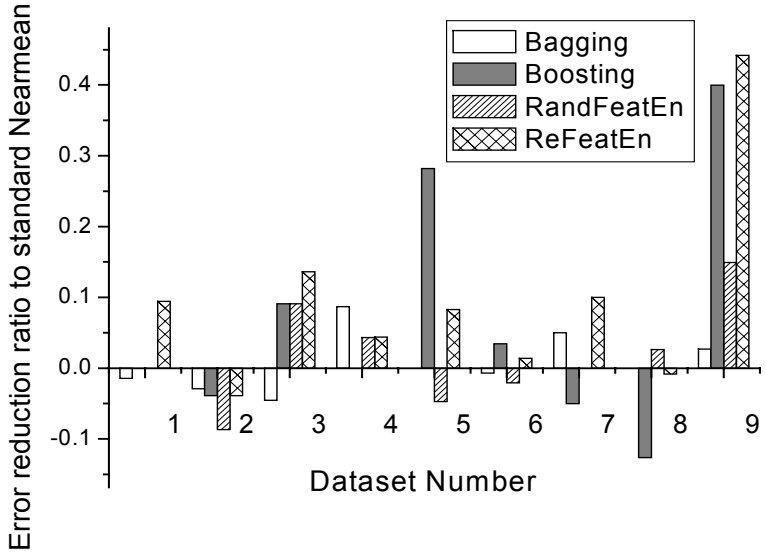

图4-11 基分类器为 NearMean 的集成分类器和标准 NearMean 比较的错误率降低比率

4.4.3.6 集成学习算法的运行效率比较

如表 4-6 所示，基于特征选择的集成学习算法的运算速度明显高于 Bagging 和 Boosting。原因在于，对于 C4.5，由于降维，使得决策树选择构造树结点时需要评估的特征大大减少，因此大大加快了运算速度。对于 NBC，

采用 Bagging 或者 Boosting 由于每训练一个个体分类器，都要重新产生训练数据集，因此要重新估计概率 P(y)和 P(Aj｜y)（参见公式(4-1)）。而 ReFeatEn 和 RandFeatEn 算法只要程序初始计算好 P(y)和 P(Aj｜y)，那么对不同特征只要选择相应 P(Aj｜y)进行计算即可，因而 ReFeatEn 和 RandFeatEn 的算法运行效率远远高于 Bagging 和 Boosting 算法。同理，对于最近邻中心法，Bagging 和 Boosting 对于每个重新产生的训练数据集，都要重新计算各类中心向量。而对于 ReFeatEn 和 RandFeatEn 则只需要在程序初始计算好各类中心向量即可。因此，总的来说，基于特征选择的集成学习算法运行效率要高于 Bagging 和 Boosting。再加上基于特征选择的集成学习方便于并行运算，这使基于特征选择的集成学习在处理高维数据时具有明显的优势。因为本文没有采用并行计算，所以 Bagging 和 Boosting 运行时间相近，而 ReFeatEn 和 RandFeatEn 运行时间相近。表 4-6 列出了 ReFeatEn 和 Boosting 的运行时间比较，从表 4-6 可知，基于特征选择的集成学习算法 RandFeatEn 和 ReFeatEn 的运行效率要高于 Bagging 和 Boosting 算法。

4.4.3.7 集成学习算法泛化性能和个体分类器个数的关系

下面以各集成学习算法在 Hanzi_256 和 Satimage 数据集上的决策树集成结果为例，简要说明各集成学习算法和迭代次数的关系，并比较各集成学习算法一般情况下个体分类器间差异度和个体分类器测试准确率。本文采用的差异度公式为公式(4-24)。下节将重点对 RandFeatEn 和 ReFeatEn 进行分析比较。

由表 4-7，4-8 可见，一般而言集成算法的测试准确率会随个体分类器个数的增多而提高。在个体分类器差异度上，Boosting 算法的个体差异度最大，ReFeatEn 次之，其次是 Bagging 算法，最后是 RandFeatEn 算法。关于个体分类器的平均准确率，则一般是 Boosting 算法最低，Bagging 算法最高。

[Breiman, 1996a]实验发现集成算法准确率提高主要发生在前 10 个个体分类器集成时。本文实验发现，集成算法准确率何时收敛和数据本身性质、集成算法本身性质以及基分类器都有很大关系。如表 4-7 中，个体分类器个数为 50 时的集成算法准确率和个体分类器个数为 25 时的准确率相比有明显提高。基于决策树的集成学习算法通常都是个体分类器越多集成算法准确率越高的。

但是基于 NBC 的 Boosting 算法却偶尔会发现个体分类器个数增多时集成算法准确率下降的情况，如表 4-9 所示，这个现象也曾经被 Wickramaratna 和 Elkan 注意到[Wickramaratna 2001] [Elkan 1997]，但没有提到解决方案，本文下一章提出的嵌入特征选择的 Boosting 算法在一定程度上解决了这个问题。

表4-7 基分类器为 C4.5 决策树，手写体汉字数据集上，不同迭代次数下，集成学习算法的测试准确率，个体分类器差异度和个体分类器平均测试准确率

Iterates	Bagging	Boosting	RandFeatEn	ReFeatEn
5[a]	74.72 0.6299 53.60	74.7 0.6828 51.47	77.04 0.5970 56.20	77.86 0.6141 56.40
13	85.84 0.6257 53.96	87.81 0.7600 46.46	88.50 0.6023 56.64	89.13 0.6259 54.97
25	90.01 0.6279 53.91	92.53 0.7617 44.10	92.50 0.6117 56.15	92.52 0.6323 54.54
50	92.21 0.6269 53.98	94.70 0.7768 42.80	94.66 0.6131 56.09	95.20 0.6353 54.34

[a] 不同迭代次数下，每个集成算法有三个数据，分别为测试准确率、个体分类器间差异度、个体分类器平均测试准确率，如 Bagging 算法在迭代次数为 5 的情况下的三者值分别为 74.7，0.6828，51.47

表4-8 基分类器为 C4.5 决策树，Satimage 数据集上，不同迭代次数下，集成学习算法的测试准确率，个体分类器差异度和个体分类器平均测试准确率 [a]

Iterates	Bagging	Boosting	RandFeatEn	ReFeatEn
5	88.15 0.1814 83.72	87.80 0.2239 81.97	88.75 0.1863 83.68	89.45 0.1902 84.15
13	89.30 0.1797 83.77	90.05 0.2671 79.68	90.40 0.1823 84.22	90.65 0.1847 84.15
25	89.70 0.1783 83.83	90.75 3014 77.54	90.85 0.1796 0.8430	90.95 0.1885 0.8389
50	90.10 0.1796 83.81	91.45 0.3364 75.41	90.80 0.1750 0.8454	91.25 0.1865 83.99

[a] 有关说明同表 4-7

表4-9 Ionosphere 数据集上，基分类器为 NBC 的 Adaboost 算法在不同迭代次数下的测试准确率

迭代次数	5	9	13	17	21	30	40	50
测试准确率	88.89	90.60	92.30	93.16	89.74	92.31	91.45	91.45

表4-10 个体分类器个数为 50，RandFeatEn 和 ReFeatEn 的各项指数对比 [a]

Dataset	C4.5		NBC		NearM	
	Rand	Relief	Rand	Relief	Rand	Relief
Hanzi	94.66	**95.20**[b]	94.51	94.26	97.24	96.50
	0.6131	**0.6353**	0.0739	0.1014	0.0448	**0.0762**
	56.09	54.34	92.48	91.64	95.61	93.54
Ionosphere	94.70	**96.58**	83.76	**85.98**	81.19	81.20
	0.1329	**0.1566**	0.0918	**0.1346**	0.1275	**0.1575**
	88.15	87.03	83.30	82.30	77.35	76.09
Satimage	90.80	**91.25**	80.05	80.05	77.80	**78.55**
	0.1750	**0.1865**	0.0741	**0.0744**	0.0926	**0.1112**
	0.8454	83.99	79.03	79.04	76.75	76.64
Isolet	87.88	**90.01**	86.15	**87.23**	82.88	**85.00**
	0.3348	**0.3424**	0.0835	**0.0956**	0.0905	**0.1009**
	73.15	73.33	84.58	**84.99**	81.77	**82.08**

[a] 有关数值意义说明同表 4-7。[b] 黑体表示对应项的值 ReFeatEn 高于 RandFeatEn

表4-11 个体分类器个数为 5，RandFeatEn 和 ReFeatEn 的各项指数对比 [a]

Dataset	C45		NBC		NearM	
	Rand	Relief	Rand	Relief	Rand	Relief
Hanzi	77.04	**77.86**	93.76	**93.96**	96.60	96.54
	0.5970	**0.6141**	0.06708	**0.0856**	0.0445	0.0659
	56.20	**56.40**	92.30	92.17	95.57	95.01
Ionosphere	90.60	**92.92**	82.90	**84.61**	79.34	**80.20**
	0.1003	**0.1373**	0.1043	**0.1379**	0.1224	**0.1465**
	87.90	**88.69**	81.20	**82.24**	77.12	**79.01**
Satimage	88.75	**89.45**	80.05	80.05	77.20	**78.45**
	0.1863	**0.1902**	0.0661	0.0661	0.1453	0.0922
	83.68	**84.15**	79.35	**79.40**	75.42	**77.39**
Isolet	82.57	**84.09**	85.76	**87.12**	82.11	**84.81**
	0.3275	**0.3486**	0.0857	0.0821	0.0958	**0.1010**
	72.75	**73.01**	85.04	**86.23**	81.5	**82.92**

[a] 有关说明同表 4-10

4.4.3.8 RandFeatEn 和 ReFeatEn 方法比较

由上文实验可知，ReFeatEn 算法性能明显优于 RandFeatEn 算法的性能，两种算法不同之处在于：ReFeatEn 的特征子集是刻意控制下产生的；而 RandFeatEn 的特征子集是完全随机产生。下面我们通过实验从个体分类器间差异度和个体分类器的平均测试准确率两方面分析 ReFeatEn 性能比 RandFeatEn 优越的原因。

表 4-10 列出了在 Hanzi_256，Ionosphere，Satimage 以及 Isolet 数据集上，个体分类器个数为 50，分别以 C4.5，NBC 和最近邻中心分类器为基分类器的 ReFeatEn 和 RandFeatEn 的测试准确率、个体分类器间的差异度以及个体分类器的平均测试准确率。可见，在全部 12 种情况中，ReFeatEn 有 9 种测试准确率高于 RandFeatEn，而在全部情况中，ReFeatEn 的个体分类器间差异度均大于 RandFeatEn，而个体分类器的平均测试准确率有 9 种情况均低于 RandFeatEn。这说明在个体分类器个数较高的情况下，个体分类器的差异度对提高集成学习的性能起了关键作用。

表 4-11 列出了 RandFeatEn 和 ReFeatEn 在 5 个个体分类器情况下的集成学习准确率、个体分类器间差异度以及个体分类器的平均测试准确率。ReFeatEn 在 12 种情况中，有 10 种平均测试准确率高于 RandFeatEn，而在其余 2 种情况也仅略低于 RandFeatEn。并且在 12 种情况中，仅在 Isolet 上的 NBC 集成以及在 Satimage 的 NearM 集成中，ReFeatEn 的平均差异度低于 RandFeatEn。

表 4-10 和表 4-11 与我们的设想一致，即在个体分类器个数少的情况，由于 ReFeatEn 中的个体分类器是在采用 Relief 评估引导的特征子集上产生的，而且控制了特征子集间的差异，因此其个体分类器的准确率较高，彼此间的差异度也较大。而当个体分类器的个数较多时，这时就要更经常的通过对新产生的特征子集进行变异来维持特征子集间的差异度，因此其相应的个体分类器间差异度高于 RandFeatEn，而由于变异过程中更倾向于从特征子集中删除特征，因此 ReFeatEn 的特征子集平均大小低于 RandFeatEn，个体分类器的平均准确率则由于这种有意的调整，而相当于或者略低于 RandFeatEn，但在大多数情况下，ReFeatEn 的分类准确率高于 RandFeatEn，这从一个侧面验证我们

的设想，即当个体分类器的个数较多时，在保证个体分类器的准确率性能不下降太多的情况下，保持个体分类器间的差异度对提高集成系统的性能非常重要。

4.4.3.9 特征子集大小和迭代次数对基于特征选择的集成学习分类器的影响

表 4-12，表 4-13 和图 4-12 列出了 RandFeatEn 和 ReFeatEn 在 Hanzi_256 数据集上的测试准确率的均值及标准偏差。在表中所列 4 种特征子集大小中，特征子集大小为 64 的 ReFeatEn 在迭代次数为 50 时取得了最高的测试准确率。从表 4-12 及 4-13 可见，对大小为 26 的特征子集，在迭代次数为 5，9，13，25 的集成学习性能均明显低于其余含有特征数目较多的特征子集对应的集成学习性能。而在大小为 192 的特征子集对应的集成学习算法，则在迭代次数为 13 时，其集成学习性能已经明显低于选择大小为 64 和 128 的特征子集对应的集成学习算法。这说明并不是特征子集越大，基于特征选择的集成分类器性能越好，因为特征子集越大，则基于其上的各个体分类器性能越相近，从而使得集成分类器性能下降。而特征子集过小，则由于个体分类器的分类性能下降过多使得集成学习的性能受到影响。在实际应用中，应根据数据集性质的不同，选择合适大小的特征集合。

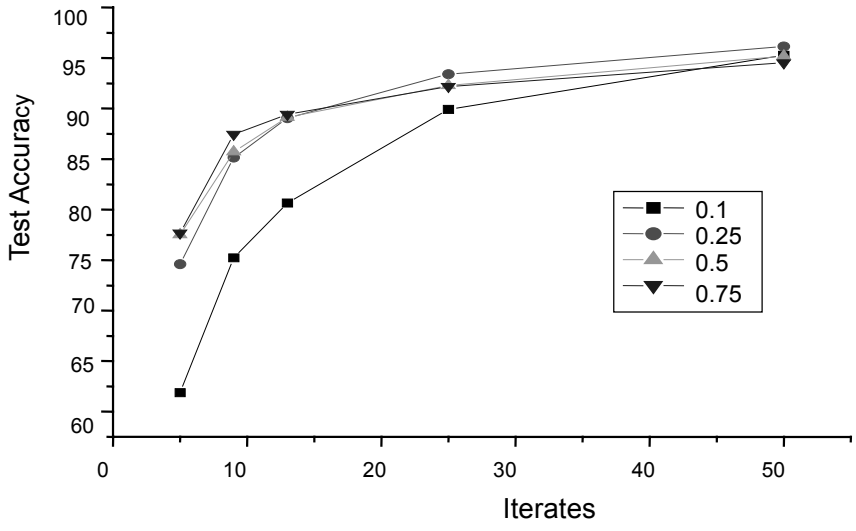

图4-12 ReFeatEn 算法测试准确率和特征子集大小和迭代次数

表4-12 不同迭代次数以及特征子集大小时，Hanzi_256 数据集上基于 C4.5 的 RandFeatEn 测试准确率的均值和标准偏差

迭代次数	特征子集大小			
	25	64	128	192
5	55.10±3.496	73.15±0.937	77.04±0.536	76.83±1.046
9	72.97±1.227	84.35±0.613	85.83±0.770	84.10±1.035
13	80.15±1.190	89.82±0.620	89.30±0.546	86.77±0.635
25	89.87±0.803	93.74±0.170	93.24±0.477	90.41±1.006
55	95.18±0.082	95.52±0.280	94.9±0.372	91.63±0.699

表4-13 不同迭代次数以及特征子集大小时，Hanzi_256 数据集上基于 C4.5 的 ReFeatEn 算法的测试准确率的均值和标准偏差

迭代次数	特征子集大小			
	25	64	128	192
5	**61.89±2.104** [a]	**74.61±0.513**	**77.53±0.272**	**77.70±1.440**
9	**75.22±1.626**	**85.154±0.644**	85.72±0.331	87.46±2.271
13	**80.66±0.605**	89.07±0.512	89.13±0.447	**89.44±0.700**
25	**89.94±0.443**	93.42±0.310	92.28±0.683	**92.18±1.450**
55	**95.27±0.101**	**96.15± 0.287**	**95.20±0.652**	**94.54±0.356**

[a] 黑体表示 ReFeatEn 的测试准确率高于表 4-12 所示对应的 RandFeatEn

4.5 本章小结

高维数据的分类由于特征维数高，特征间关系复杂，是模式识别领域公认的难度较大的分类问题。以往针对高维问题通常首先通过特征选择算法进行降维，降低问题复杂度后再进行分类，但正如第三章所述，高维数据的特征选择本身就是很具挑战性的研究课题。

本章充分利用高维数据集的特征间通常信息关联度较高的特点，提出一种基于特征选择的集成学习算法 ReFeatEn。ReFeatEn 算法分两阶段产生用于构建不同个体分类器的特征子集：第一阶段利用 Relief 特征评估结果依概率从原始特征集合中选择特征组成特征子集，第二阶段通过精调使各特征子集间满足

一定的差异度。分别以决策树、Naive Bayes、最近邻中心法为基分类器，在多个实际数据集和人工数据集上，对 ReFeatEn、Bagging、 Boosting，以及基于随机特征子集的集成学习算法 RandFeatEn 进行了实验。实验表明，在特征维数较高，特征间关系较复杂的数据集上，ReFeatEn 算法的测试准确率始终优于或相当于 Bagging、Boosting，以及 RandFeatEn。并且 ReFeatEn 的运行速度远高于 Bagging 和 Boosting 算法，而且适于并行运行，因此是一种适用于高维数据的基于特征选择的集成学习算法。经实验分析，我们发现和 RandFeatEn 相比：在个体分类器个数较少时，ReFeatEn 性能较好是由于个体分类器的分类性能较高以及个体分类器间的差异度较大；当个体分类器个数较多时，ReFeatEn 性能更好主要是由于个体分类器间的差异度较大。

我们在实验中发现，基分类器为朴素贝叶斯和最近邻中心分类器的 Boosting 算法对于含有噪声特征的数据集效果较差，而基于特征选择的 ReFeatEn 算法则对于含有噪声特征的数据集具有较好的性能。能否将特征选择嵌入 Boosting 算法中，提高 Boosting 算法对噪声数据的适应能力？这正是下一章提出的 FeatBoostNBC 和 FeatBoostNearmean 算法的初衷，具体将在下一章详细介绍。

第五章 一种嵌入特征选择的 Boosting 集成学习算法

5.1 引言

集成学习算法的研究主要集中于决策树和神经网络，在贝叶斯分类器以及最近邻法的研究上要少得多。这是由于决策树和神经网络被认为是不稳定的学习算法，即输入样本集的微小变化会引起决策树和神经网络性能的较大改变，而贝叶斯和最近邻分类算法则是很稳定的算法。对于贝叶斯分类器，输入样本集的微小变化对决策中使用的各参数的统计估计影响很小，而对于最近邻法，基于样本的集成学习算法并不能改变样本之间的最近邻关系，因此性能提高有限，甚至会导致集成学习性能不如单个分类器的性能。

上一章实验发现，对于贝叶斯和最近邻中心分类器，Boosting 算法对含有噪声的数据集性能较差，而且出现集成学习算法的性能低于单个分类器的现象。而本文提出的基于两步式特征选择的集成学习算法 ReFeatEn 算法虽然对有噪声的数据性能很好，但在大多数数据集上，ReFeatEn 性能提高则有限。

本章在 Boosting 算法的基础上，提出了将特征选择嵌入到 Boosting 算法中的思路，目的是通过特征选择减小个体分类器间的错误样本区域的重叠，同时消除噪声特征对分类器的影响。在上述思路指导下，根据贝叶斯分类器和最近邻分类器的特点，分别设计了 FeatBoostNBC 和 FeatBoostNearMean 算法，在 UCI 数据上的实验结果充分验证了这两种算法的有效性。

5.2 算法设计思路

集成分类器的设计要解决两个问题，即个体分类器的生成以及结论生成。本文集中于研究个体分类器的生成，为得到较好的集成分类器，个体分类器生成主要考虑保证以下两个因素：个体分类器的分类错误的样本分布区域不同，或者说个体分类器间差异较大；个体分类器的分类准确率较高。

Boosting 算法在贝叶斯和最近邻中心分类器上经常效果不明显，尤其是在有噪声的数据上性能不如第四章介绍的 ReFeatEn。本文作者认为，造成上述现象可能有两个原因：一是由于 Boosting 算法是基于样本的集成学习，无法

消除噪声特征对于分类器的影响；二是由于贝叶斯和最近邻分类器都是很稳定的算法，基于样本的 Boosting 算法在个体分类器间的差异度不足。

第四章提出的基于两步式特征选择的集成学习算法(ReFeatEn)在有噪声特征的数据集上集成学习准确率明显高于 Boosting 算法，这是因为 ReFeatEn 算法在征子集产生过程中的第一步偏向于选择类别的相关度高的特征，第二步虽然进行了变异，但倾向于删除已有特征而不是增加新特征，以上操作使得无关特征在特征子集中出现概率较低，因此对噪声特征具有很强的鲁棒性。

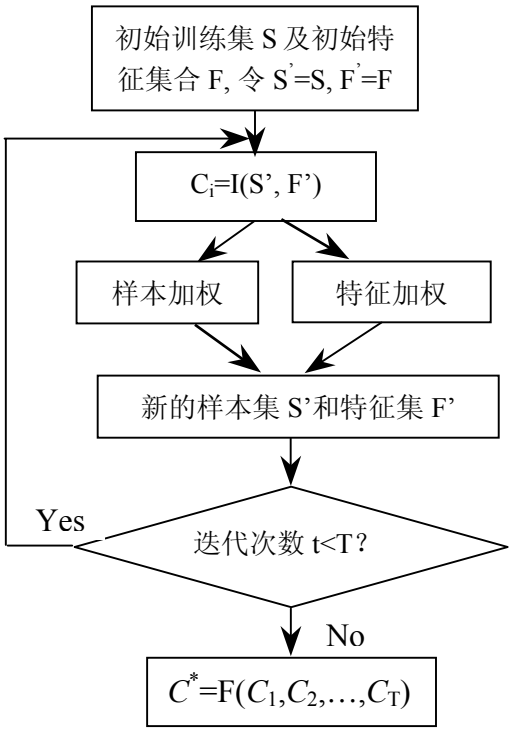

图5-1 嵌入特征选择的 Boosting 算法结构图

分析 Boosting 和 ReFeatEn 两算法的机理， Boosting 算法迭代中努力在当前轮将上一轮被错分的样本分正确，而 ReFeatEn 算法则努力消除无关特征的影响。那么可否设计一种算法，克服 Boosting 算法易受噪声特征影响的缺点，同时保留 Boosting 将注意力集中于被错分样本的特性。

基于上述想法，我们提出了将特征选择嵌入到 Boosting 算法的思路，并设计了总体算法框架，如图 5-1 所示，算法整体结构和 Boosting 算法结构类似，但是在每一轮循环中，除了 Boosting 自身的样本加权外，根据分类器本

身分类机制以及错分样本的情况，对每个特征进行加权，然后通过样本取样和特征选择得到新的样本集和特征集，新的样本集和特征集用于训练下一轮个体分类器，如此循环直至指定代数，然后采用加权投票法组合得到集成分类器。

在以上总体算法结构图指导下，依据贝叶斯分类器和最近邻中心分类器的自身特点，本文分别设计了 FeatBoostNBC 和 FeatBoostNearMean 集成算法。

5.3 嵌入特征选择的贝叶斯集成学习算法 FeatBoostNBC

为说明嵌入特征选择的出发点，本文首先回顾一下朴素贝叶斯分类器。

5.3.1 朴素贝叶斯分类器

不失一般性，设 A_1 到 A_f 为离散型特征（连续型特征可以化为离散型特征），这些特征用来判断类别 C。给定一个特征值为 a_1 到 a_f 的样本，最优决策 $c=\arg_c \max P_r(C=c \mid A_1=a_1 \wedge \ldots \wedge A_f=a_f)$。根据贝叶斯理论，$P_r(C=c \mid A_1=a_1 \wedge \ldots \wedge A_f=a_f)$ 为

$$\frac{P_r(A_1=a_1 \wedge \ldots \wedge A_f=a_f \mid C=c)}{P_r(A_1=a_1 \wedge \ldots \wedge A_f=a_f)} P_r(C=c) \qquad \text{公式(5-1)}$$

其中先验概率 $P_r(C=c)$ 很容易从训练集中按下式估计得到。

$$\hat{P}_r(C=c)=\frac{count(C=c)}{count(trainset)} \qquad \text{公式(5-2)}$$

由于 $P_r(A_1=a_1 \wedge \ldots \wedge A_f=a_f)$ 的值对所有类都一样，所以和决策无关。因此问题集中于根据训练样本集估计 $P_r(A_1=a_1 \wedge \ldots \wedge A_f=a_f \mid C=c)$ 的值。根据贝叶斯原理可得出公式(5-3)：

$$P_r(A_1=a_1 \wedge \ldots \wedge A_f=a_f \mid C=c)=$$
$$P_r(A_1=a_1 \mid A_2=a_2 \wedge \ldots \wedge A_f=a_f, C=c).P_r(A_2=a_2 \wedge \ldots \wedge A_f=a_f \mid C=c) \qquad \text{公式(5-3)}$$

递归推得公式(5-4)：

$$P_r(A_1 = a_1 \wedge \ldots \wedge A_f = a_f \mid C = c) =$$
$$P_r(A_1 = a_1 \mid A_2 = a_2 \wedge \ldots \wedge A_f = a_f, C = c) \times$$
$$P_r(A_2 = a_2 \mid A_3 = a_3 \wedge \ldots \wedge A_f = a_f, C = c) \times \ldots \times P_r(A_f = a_f \mid C = c)$$

公式(5-4)

根据特征间独立的假设可得公式(5-5)：

$$P_r(A_1 = a_1 \mid A_2 = a_2 \wedge \ldots \wedge A_f = a_f, C = c) = P_r(A_1 = a_1 \mid C = c)$$ 公式(5-5)

同样 A_2，…，A_f也可如此简化，则有公式(5-6)：

$$P_r(A_1 = a_1 \wedge \ldots \wedge A_f = a_f \mid C = c) =$$
$$P_r(A_1 = a_1 \mid C = c) P_r(A_2 = a_2 \mid C = c) \ldots P_r(A_f = a_f \mid C = c)$$ 公式(5-6)

乘积里的每一项可从训练集中按公式(5-7)估计得到，此估计为最大似然估计，即给出的估计值将使样本出现的概率最大。

$$\hat{P}_r(A_j = a_j \mid C = c) = \frac{count(A_j = a_j \wedge C = c)}{count(C = c)}$$ 公式(5-7)

5.3.2 朴素贝叶斯分类器，特征和先验概率

本节将通过贝叶斯分类器的变形，说明以 $P_r(A_j = a_j \mid C = c)$ 形式表现的单个特征在分类中的作用，并进而在下一节提出嵌入特征选择的贝叶斯集成分类算法。

设某数据集类别数为 2，分别用 0 和 1 表示，$a_1,\ldots a_f$为一个测试样本的特征值，$b_0=P_r(C=0)$，$b_1=P_r(C=1)=1-b_0$，$p_{j0}=P_r(A_j=a_j|C=0)$，$p_{j1}=P_r(A_j=a_j|C=1)$，则根据上节推理可知两类别的后验概率满足公式(5-8)和公式(5-9)。

$$p = P_r(C = 1 \mid A_1 = a_1 \wedge \ldots \wedge A_f = a_f) = (\prod_{j=1}^{f} p_{j1}) b_1 / z$$ 公式(5-8)

$$q = P_r(C = 0 \mid A_1 = a_1 \wedge \ldots \wedge A_f = a_f) = (\prod_{j=1}^{f} p_{j0}) b_0 / z$$ 公式(5-9)

公式(5-8)和(5-9)中 z 是常数，其作用是进行归一化。对这两公式取对数相减可得

$$\log p - \log q = \sum_{j=1}^{f}(\log p_{j1} - \log p_{j0}) + (\log b_1 - \log b_0) \qquad \text{公式(5-10)}$$

设 $w_j = \log p_{j1} - \log p_{j0}$， $b = \log b_1 - \log b_0$ 则

$$\log \frac{p}{q} = \sum_{j=1}^{f} w_j + b \qquad \text{公式(5-11)}$$

由公式(5-11)可见，一个测试样本的分类，由 b 以及 w_j 决定。当一个样本分错时，Boosting 会增加该样本的权值，因而改变了 b 以及 w_j 的值，使得该样本在下一轮个体分类器设计中有可能得到正确分类。

5.3.3 FeatBoostNBC 集成学习算法

Boosting 算法是基于样本的集成学习算法。观察公式(5-11)，从特征的角度考虑，假设一个样本 X 类别为 1，但朴素贝叶斯分类器得出 p 小于 q（即 $\log p/q < 0$），则样本 X 被错分为类别 0。考察 w_j，j=1，…，f，如果 w_j<0，则说明特征 j 对测试样本 X 的正确分类起了负面作用，如果将特征 j 删除，则 $\log p/q$ 增加，样本 X 被正确分类的可能性增大。本文算法 FeatBoostNBC 正是从特征角度出发，将特征选择嵌入到 Boosting 算法中，使得 Boosting 算法内的每一次迭代过程不仅使用有偏向的样本集，也选用有偏向的特征子集，从而使稳定的朴素贝叶斯分类器倾向于不稳定，在每一轮都努力将上一轮错分的样本进行正确分类。

FeatBoostNBC 算法的关键在于在每一轮迭代过程中产生下一轮选用的特征子集，本文借鉴了 Relief 系列算法的思想，采取对特征加权的策略，Relief 的核心是根据每个特征区分最近邻样本的能力对特征进行评估，而本文的目的是评估特征对样本错分的作用，特征选择的目的是选择的特征子集在下一轮尽力使当前被错分的样本被正确分类。具体做法如图 5-2 所示，设在迭代次数为 i 时，采用 i-1 轮时决定的特征子集构造分类器 C_i，对每个被分类器 C_i 错分的样本 X_j，设 X_j 类别为 Y_j，设 C_i 对 X_j 分类结果为 $C_i(X_j)$，考虑每个特征 F[k]，如果 $Pr(X_{jk} \mid Y_j)$ — $Pr(X_{jk} \mid C_i(X_j))$<0，则说明在概率 $Pr(X_j, C)$ 计算中，特征 F[k]

起了负作用，所以应该降低 F[k]的权值，反之如果 $Pr(X_{jk} \mid Y_j) - Pr(X_{jk} \mid C_i(X_j)) > 0$，则特征 F[k]起了正作用，应该增加 F[k]的权值。特征权值按公式(5-12)进行更新。

$$Weight(F[k]) = Weight(F[k]) + P_r(X_{jk} \mid Y_j) - P_r(X_{jk} \mid C_i(X_j)) \qquad \text{公式(5-12)}$$

输入：S－训练样本集，大小为 m；I－ 分类算法；T－循环代数；F－特征全集

1．S'=copy(S)，F'=copy(F) ,i=0, weight(X_j)=1;

2．C_i=I(S', F');

3．$\varepsilon_i = \frac{1}{m} \sum_{X_j \in S: C_i(X_j) \neq Y_j} \text{weight}(X_j)$

4．Foreach F[K] ∈F　Weight(F[k])=0; //初始化特征权值

5．ForEach $X_j \in S: C_i(X_j) \neq Y_j$)

6．　Weight(F[k])= Weight(F[k])＋ $P_r(X_{jk} \mid Y_j) - P_r(X_{jk} \mid C_i(X_j))$;

7．若$\varepsilon_i \geq 1/2$ 或者$\varepsilon_{i=0}$，且 ResampleTime<25 则 S'=bootstrap(S)，返回(2)

8．$\beta_i = \varepsilon_i / (1-\varepsilon_i)$

9．ForEach $X_j \in S$　//更新样本权值

10．　if($C_i(X_j) = Y_j$)　weight(X_j) = weight(X_j). β_i

11．F'=copy(F),　ForEach F[k] ∈ F　//产生下一轮的特征子集

12．　　if (Weight(F[k])<0)　F'={F'-F[K]};

13．i++, if i<T, goto 2

14．按下式得到集成分类器 C^*　C^*（X）＝ $\arg\max_{y \in Y} \sum_{i:C_i(X)=y} \log \frac{1}{\beta_i}$

结束

图5-2 FeatBoostNBC 算法

当特征权值确定之后，则认为特征权值为负的特征对于错分的样本起了负作用，因此将其删除，这样就得到了下一轮所要使用的特征子集 F’。下一轮个体分类器在新样本集 S'和新特征集 F'上训练得到（图 5-2 第 2 行），如此重复，直到指定的迭代次数，然后按照加权的相对多数投票法的集成分类器（图 5-2 第 14 行）。

集成学习性能另一个重要影响因素是个体分类器的准确率，本文采用

Bauer 和 Kohavi 中采用的方案，当$\varepsilon_i \geq 0.5$ 时，则利用可重复取样方法取得新的训练集以及采用全部特征集合循环训练，直到$\varepsilon_i < 0.5$，为保证程序在可承受的时间内结束，本文设置循环代数为 25（图 5-2 第 6 行）。

FeatBoostNBC 算法另一个需要注意的问题是每一轮特征子集的大小问题，虽然运行程序中还未发现特征子集会出现空集的情况，但不能排除这种可能，而这是需要避免的。为此，如果删除的特征数目大于原特征集合的四分之三，那么我们将随机添加四分之一的特征到特征子集中去。这些参数是人为定的，有关实验表明实验结果对参数并不敏感。

5.3.4 实验结果与讨论

5.3.4.1 实验目的和方法

实验目的有以下两个：

1. 验证 FeatBoostNBC 算法性能是否优于 NBC 的 Boosting 算法。
2. 一个很自然的问题就是，如果 FeatBoostNBC 比 Boosting 性能优越，那是由于其引进了特征选择机制。那么如果不引入 Boosting，而直接利用 FeatBoostNBC 中采用的特征选择机制每轮得到不同的特征子集，而不进行 Boosting 中的加权以及重复采样产生新样本集的操作，那么效果如何呢，我们称此算法为 FeatNBC 算法，直觉，这样操作容易产生来回振荡的情形，效果不稳定，我们通过实验进行了测试和分析。

我们在表 5-1 第一列所示的数据集上进行了实验。选择个体分类器个数为 50，数据集为从 MLC 上下载的 UCI 数据。除列出 FeatBoostNBC， Boosting 以及 FeatNBC 的测试准确率外，为比较起见，也列出了不进行集成的单个朴素贝叶斯分类器的测试准确率，即 Standard 对应的列。同时列出了第四章提出的经实验验证性能较好 ReFeatEn 算法在朴素贝叶斯分类器上的测试准确率。为了更清楚的对比 FeatBoostNBC 和 Boosting 间的性能，我们在最后一列(标注为ε_r)列出了 FeatBoostNBC 相对于 Boosting 算法的错误率降低百分比。错误率降低百分比是这样定义的，比如两个进行比较的方法为 A 和 B，A 的错误率

为ε_A，B 的错误率为ε_B，则 A 相对于 B 的错误率百分比为$(\varepsilon_B-\varepsilon_A)/\varepsilon_B$。采用错误率降低百分比而不是正确率提高百分比，是由于对某些正确率已经很高的情况下如 98%，如果改进后的正确率为 99%，那么正确率提高百分比为(99%-98%)/%98=1%，而错误率降低百分比为(2%-1%)/2%=50%，显然这种情况下错误率降低百分比作为算法衡量更合理。

如表 5-1 所示，FeatBoostNBC 算法在所有实验的数据集上均取得了高于 Boosting 的性能，由ε_r列的值可以看出，在本文实验的数据集上，和 Boosting 算法相比，FeatBoostNBC 错误率最多降低了 87.08%，最少降低了 7.75%，平均降低了 26.59%。需要说明的是，原 sick-euthyroid 数据集有 25 个特征，其中有 7 个连续特征，数据集中多个特征存在值丢失的情况，最后一个特征的值丢失情况尤其严重，并且丢失情况在两类间严重不平衡，这种情况对决策树分类器影响不大，但是对 NBC 有很大的影响。为此，我们分别验证了在原 sick-euthyroid 数据集上以及去除最后一个特征后的数据集上（分别称为 sick-euthyroid 和 sick-euthyroid(1)）各算法的性能，在 sick-euthyroid 数据集上，Boosting 算法测试准确率为 25.87%。但在 sick-euthyroid(1)数据集上，Boosting 算法测试准确率为 92.51%，可见 Boosting 对此类噪声特征很敏感。类似的情况也发生在 Breast-cancer 数据集上，Breast-cancer 的第一个特征是病人编号，对分类毫无帮助。Boosting 算法在 Breast-cancer 上的错误率比标准 NBC 增加了 19.98%。而其他的集成学习算法因为每一轮个体分类器都是基于特征选择得到的特征子集上训练生成的，所以对于无关特征不敏感，在 sick-euthyroid 和 Breast-cancer 数据集上，都取得了较好的性能。

如表 5-2 所示，在经过几次（大多数情况小于 5）循环后，除 Ionosphere 和 Isolet 数据外，FeatNBC 的个体分类器在其余数据集上出现循环重复的现象，基本上是在两种分类器之间振荡。即正如最初直觉得出的结论那样，FeatNBC 选择的特征子集常常只是在两种方案间振荡。如图 5-3 所示，由于训练数据集的样本不变，那么在 t 次循环时，采用特征子集 F_t 训练得到个体分类器 C_t，被 C_t 错分的样本集 S_t 决定了下一轮特征子集 F_{t+1}，在第 t+1 次循环时，采用特征子集 F_{t+1} 训练得到个体分类器 C_{t+1}，被错分的样本 S_{t+1} 决定了下一轮训练使用

的特征子集 F_{t+2}，在第 t+2 次循环时，采用特征子集 F_{t+2} 训练得到个体分类器 C_{t+2}，被错分的样本集为 S_{t+2}。经常出现下面情况：F_{t+2} 和 F_t 相同，F_{t+1} 和 F_{t+3} 相同，如此即导致个体分类器 C_t 和 C_{t+2} 相同，C_{t+1} 和 C_{t+3} 相同，如此即出现振荡情况。出现这种情况的结果可能会导致集成分类器结果变好，也可能会导致集成分类器结果变坏，这和参与振荡的两种分类器性能有关系，也和何时开始重复有关系。如表 5-1 所示，在本文实验的数据集上，和其他集成算法相比，FeatNBC 仅在 Horse-colic 数据集取得最好的集成性能，但常常得到最差的集成性能。例如：在 sonar，Soybean-large 和 German 数据集上的测试准确率远低于单个朴素贝叶斯分类器的测试准确率，在 Soybean-large 数据集上，错误率增加了 200%，而在 sonar 上，错误率增加了 29%，在 German 数据集上，错误率增加了 33%。但是对于 Sick-euthyroid 和 Breast-cancer 数据集都存在”坏”特征对 NBC 分类器影响较大的情况，FeatNBC 算法由于可以去除这种“坏”特征，因而取得了较高的测试准确率。在存在“坏”特征的数据集 Sick-euthyroid 和 Breast-cancer 数据集上，由于 ReFeatEn 算法的特征子集通常会剔除“坏”的特征，因而也总是取得较好的性能。

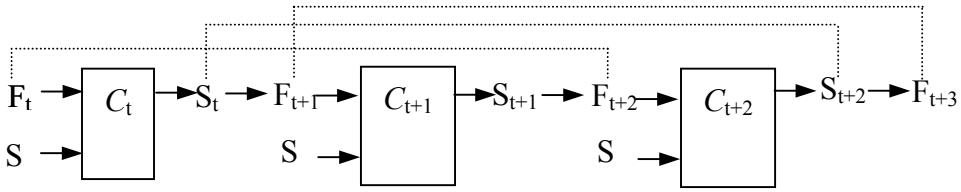

图5-3 FeatNBC 算法的在两种分类器间振荡的示意图

总体看来，FeatBoostNBC 在稳定性和准确性上显著优于 Boosting、FeatNBC 和 ReFeatEn 算法。从算法时间复杂度考虑，由于 FeatBoostNBC 比 Boosting 算法每一轮多了特征评估一项，比 FeatNBC 算法多了样本加权一项，因此运行时间比 Boosting 和 FeatNBC 算法长，但由于无论是样本加权还是特征评估的运行时间和个体分类器构造比起来要少得多，因此 FeatBoostNBC 的运行效率和 Boosting 以及 FeatNBC 是同等数量级的。和上述三种集成算法比较，ReFeatEn 算法运行效率最高，且适于并行运行，最适于应用于大规模数据集。

表5-1 NBC 上各集成学习算法的测试准确率比较

	FeatBoostNBC	Boosting	FeatNBC	ReFeatEn	Standard	ε_r [c]
Ionosphere	**94.02**[a]	91.45	90.59	85.98	83.76	30.06
Isolet	**90.19**	88.65	88.07	87.23	85.96	13.57
Horse-colic	82.35	77.00	**83.82**	79.94	80.88	23.26
Sonar	**87.14**	81.42	61.43	71.43	70.00	30.79
Waveform-40	**81.20**	79.62	75.45	79.47	80.53	7.75
Soybean-large	**94.30**	92.54	80.26	90.78	93.42	23.59
hypothyroid	**98.10**	97.06	97.06	97.32	97.72	35.37
Sick-euthyroid	90.42	25.87	91.65	**91.18**	18.58	87.08
Sick-euthyroid(1)[b]	**94.21**	92.51	91.65	92.75	85.40	22.70
Breast-cancer	74.73	68.42	73.68	**75.79**	73.68	19.98
German	76.94	74.55	70.06	77.25	**77.54**	9.39
Average	**87.60**	79.01	82.16	84.47	77.04	26.59

[a] 黑体表示该值为表中所有集成学习算法中测试准确率的最大值

[b] Sick-euthyroid（1）为去掉 Sick-euthyroid 数据集中的最后一个属性得到的数据集

[c] ε_r , FeatBoostNBC 和 Boosting 算法相比的错误率降低百分比

表5-2 FeatNBC 个体分类器的两种测试准确率

Waveform	German	Breast[a]	hypothyroid	Sick	Soybean	Horse	Sonar
75.45	75.15	78.95	97.25	91.66	80.26	83.82	77.14
64.51	68.56	65.26	97.06	84.46	69.69	69.11	61.43

[a] 在不会引起混淆的前提下，某些数据集的名称用前几个字母表示，如 Breast 表示 Breast-cancer 数据集

5.3.4.2 FeatBoostNBC 和 Boosting 算法比较

一般认为，个体分类器间的差异度和个体分类器的平均准确率对集成学习分类器影响很大。本节希望从以上两方面探讨 FeatBoostNBC 比 NBC 的 Boosting 算法性能优越的原因。同时，本文也观察了 FeatBoostNBC 和 NBC 的

Boosting 算法的各有关参数随迭代次数变化的情况。

表 5-3，5-4，5-5 显示了在数据集 Ionosphere，Horse-colic 和 Sick-euthyroid 上，不同迭代次数下，Boosting 和 FeatBoostNBC 算法的集成准确率、个体分类器间平均差异度以及个体分类器的平均准确率。本文发现：

1． FeatBoostNBC 的个体分类器间的平均差异度一般大于 Boosting，个体分类器的平均准确率一般小于 Boosting，而 FeatBoostNBC 的准确率一般高于 Boosting，这说明 FeatBoostNBC 的准确率提高主要应归功于个体分类器间的差异度较高，但 FeatBoostNBC 和 Boosting 的个体分类器间的平均差异度间差异并不特别明显，因此目前还无法肯定的下论断。

2．FeatBoostNBC 和 Boosting 算法的集成准确率提高主要发生最初的几轮循环内，从表 5-3，5-4，5-5 可以看出，一般在个体分类器个数为 13 时，集成算法的测试准确率和个体分类器个数为 50 时的测试准确率相差不多。这个发现是鼓舞人心的，这说明我们要通过集成学习提高某种分类学习算法的性能，并不需要运行很多次，这对大规模数据的集成学习尤为重要。

3．实验中本文作者发现，NBC 的 Boosting 算法经常出现迭代次数多，集成学习准确率下降的现象。如表 5-3 所示，在 Ionosphere 即呈现出这种现象，但很难总结规律。以前研究者通常报告的在其余基分类器上的实验结果都是迭代次数越多，集成学习算法准确率越高[Schapire 1997]。[Elkan 19997] [Wickramaratna 2001]曾报告过这个现象，但[Elkan 1997]没有给予解释，[Wickramaratna 2001]主要是针对性能很好的强分类器情况。观察 Ionosphere 实验，会发现 Boosting 算法的个体分类器间的平均差异度在个体分类器个数超过 17 后开始下降（本文没有依次记录不同个体分类器数目下的各集成后性能参数，但不影响总体趋势），而个体分类器的平均准确率通常总是随着个体分类器数目的增多而下降。虽然可以用这两点来解释集成算法准确率下降，但是集成学习算法是很复杂的，有时会出现同样的个体分类器平均差异度，以及同样的个体分类器平均准确率，但是集成学习的性能却差别很大的情况。本文作者认为要想详细解释集成学习的性能，需要考察个体分类器间的关联情况，个体分类器间的平均差异度和平均准确率不足以完全解释集成学习算法的性能。

表5-3 Ionosphere 数据集上，Boosting 和 FeatBoostNBC 对比

T[a]	Boosting			FeatBoostNBC		
	En_acc[b]	Div[c]	Av_acc[d]	En_acc	Div	Av_acc
5	88.89	0.3829	74.02	89.71	0.3487	76.41
9	90.60	0.4008	72.27	94.01	0.3970	71.51
13	92.30	0.3778	72.65	94.01	0.4258	67.85
17	93.16	0.4496	65.71	93.16	0.4583	64.25
21	89.74	0.4347	66.67	92.31	0.4433	65.97
25	92.31	0.4271	67.04	92.31	0.4270	67.32
30	92.31	0.4154	67.86	93.16	0.4473	64.73
35	91.45	0.4210	67.42	93.16	0.4509	64.10
40	91.45	0.4093	68.31	93.16	0.4498	64.08
45	91.45	0.4084	68.40	94.02	0.4482	64.41
50	91.45	0.4081	68.32	94.02	0.4531	63.68

[a] T：迭代次数 , [b] En_accu： 集成学习算法的测试准确率，[c] Div： 个体分类器间平均差异度，[d] Av_acc：个体分类器的平均测试准确率

表5-4 Horse-colic 数据集上， Boosting 和 FeatBoostNBC 算法对比[a]

T	Boosting			FeatBoostNBC		
	En_acc	Div	Av_acc	En_acc	Div	Av_acc
5	75.57	0.37647	67.65	83.82	0.4911	63.24
9	75.53	0.4314	63.40	83.82	0.4714	59.64
13	76.53	0.4204	62.90	82.35	0.4611	59.95
17	76.70	0.4211	60.47	80.88	0.4610	59.43
21	76.47	0.4250	59.89	85.29	0.4668	59.73
25	74.53	0.4268	59.87	82.35	0.4557	61.00
30	76.01	0.4413	59.76	83.82	0.4625	59.56
35	76.30	0.4515	58.19	83.82	0.4585	60.00
40	76.53	0.4542	57.98	83.82	0.4631	59.26
45	75.05	0.4646	56.86	82.35	0.4631	58.79
50	77.00	0.4635	57.09	82.35	0.4706	57.74

[a] 有关符号说明同表 5-3

表5-5 Sick-euthyroid(2)数据集上， Boosting 和 FeatBoostNBC 对比 [a,b]

T	Boosting			FeatBoostNBC		
	En_acc	Div	Av_acc	En_acc	Div	Av_acc
5	91.85	0.2877	78.09	93.01	0.3522	76.74
9	92.41	0.2866	75.45	93.31	0.3558	72.51
13	92.41	0.2890	74.90	93.80	0.4279	66.80
17	92.43	0.2900	73.66	93.83	0.4824	56.73
21	92.51	0.3066	72.90	93.74	0.4744	59.68
25	—	—	—	94.21	0.4655	60.24
30	—	—	—	94.60	0.4641	61.00
35	—	—	—	94.12	0.4539	62.35
40	—	—	—	94.50	0.4435	63.47
45	—	—	—	94.50	0.4330	64.49
50	—	—	—	94.78	0.4219	65.62

[a] 有关符号说明同表 5-3, [b] “—”表示迭代中途中断，即图 5-2 中第 7 行的 ResampleTime 大于 25

5.4 嵌入特征选择的最近邻中心集成学习算法 FeatBoostNearMean

最近邻中心分类器也是一种很稳定的分类器，从第四章研究可以看出，Boosting 算法应用于最近邻中心分类器上的效果不明显。本文提出的 FeatBoostNearMean 算法结构和 FeatBoostNBC 算法非常类似，不同之处在于每一轮特征评估的方法。特征评估方法在本文是根据基分类算法的本身性质设计的，因此本文首先分析一下最近邻中心分类器，然后介绍 FeatBoostNearMean 算法。

5.4.1 最近邻中心分类器

不失一般性，假设一个类别数为 l 的数据集，训练集 S 由 m 个样本组成 $\{X_1,X_2,\ldots,X_m\}$，每个样本由 f 个特征变量描述，$X_j=\{X_{j,1},X_{j,2},\ldots,X_{j,f}\}$，每个类 i 可得到一个中心向量 $M_i=\{M_{i,1},M_{i,2},\ldots,M_{i,f}\}$，则

$$M_i = \sum_{X_j \in S:Y_j=i} X_j \qquad \text{公式(5-13)}$$

样本和类中心向量间距离采用欧式距离，计算公式为

$$d(M_i, X_j) = \sum_{k=1}^{f} (M_{i,k} - X_{j,k})^2 \qquad \text{公式(5-14)}$$

最近邻中心分类器的训练和对样本 X_j 分类过程如下：

- step one： 从训练数据集获得各类的中心向量 $M_1, M_2, \ldots, M_l$；
- step two： 对测试样本 X_j，根据公式(5-13)计算 X_j 到所有类中心向量的距离 $d(M_i, X_j)$；
- step three：得到 X_j 的类别： $C(X_j) = \arg\min_k d(M_k, X_j)$

5.4.2 FeatBoostNearMean 算法

由公式(5-14)可见，如果一个样本 X_j 被错分到类 l 中，则有 $d(M_l, X_j) < d(M_{Y_j}, X_j)$，即

$$\sum_{k=1}^{f} (M_{l,k} - X_{j,k})^2 - \sum_{k=1}^{f} (M_{Y_j,k} - X_{j,k})^2 < 0 \qquad \text{公式(5-15)}$$

则有 $\sum_{k=1}^{f} [(M_{l,k} - X_{j,k})^2 - (M_{Y_j,k} - X_{j,k})^2] < 0$，如果在特征 k 上 $(M_{l,k} - X_{j,k})^2 - (M_{Y_j,k} - X_{j,k})^2 > 0$，则特征 K 为样本 X_j 的错分做了贡献，应该降低特征 K 的权值。反之，则应增加特征 K 的权值。特征的权值越高，说明该特征对分类起的正面作用越大。我们采用公式(5-16)更新特征 K 的权值

$$Weight(F[k]) = Weight(F[k]) + |M_{j,k} - X_{j,k}| - |M_{Y_j,k} - X_{j,k}| \qquad \text{公式(5-16)}$$

算法流程和图 5-2 所示 FeatBoostNBC 不同之处即在于用公式(5-15)代替了公式(5-11)，因此本文不再赘述 FeatBoostNearMean 的算法。

需要说明的是，虽然 FeatBoostNearMean 算法是针对最近邻中心分类器

的，但是该算法可以很容易扩展而应用到最近邻或者 k-近邻分类器上。

5.4.3 实验结果与讨论

表 5-6 列出了基于最近邻中心分类器的各集成学习算法的测试准确率，和 FeatBoostNBC 的实验设计类似。Standard 列表示最近邻中心分类器的测试准确率，ε_r 列表示 FeatBoostNearMean 和 Boosting 相比的错误率降低百分比。

从表 5-6 可以看出，FeatBoostNearMean 在所有数据集上，测试准确率均高于或者相当于 Boosting 算法，在 Sick-euthyroid 数据集上，错误率降低最多，为 54.71%。在 Breast-cancer 数据集上，错误率降低了 17.23%。

表5-6 最近邻中心分类器上各集成学习算法的测试准确率比较

	FeatBoostNearMean	Boosting	FeatNearMean	ReFeatEn	Standard	ε_r (%)
Ionosphere	**83.76**[a]	80.34	76.92	81.20	82.06	13.62
Isolet	**90.19**	88.26	88.27	85.00	83.65	16.44
Horse-colic	**83.82**	70.59	76.47	72.06	67.65	44.98
Sonar	**75.71**	70.00	67.14	74.29	71.43	19.03
Waveform-40	**81.34**	77.09	79.02	79.49	79.66	18.55
Soybean-large	**72.03**	71.05	62.28	68.42	71.06	3.39
hypothyroid	94.02	94.02	**97.15**	92.04	94.03	3.34
Sick-euthyroid	**89.57**	76.97	81.14	78.58	61.62	54.71
Breast-cancer	**74.73**	69.47	73.68	63.16	53.68	17.23
German	69.16	67.96	**70.06**	67.96	69.16	3.75
Average	**81.45**	76.58	77.21	76.22	73.40	19.50

[a] 黑体表示该值为表中所有集成学习算法中测试准确率的最大值

FeatBoost 算法在最近邻中心分类器上的表现也和其在 NBC 上类似。即个体分类器最后在两类分类器上振荡，因此其集成和个体分类器个数关系不大，多数情况等价于两个个体分类器的集成，因此其集成性能也就完全由两个个体分类器决定。表现在集成后的性能在各数据集上不总是高于单个个体分类器，如在 Sonar 和 Soybean-large 数据集上，其准确率低于最近邻中心分类器，但是在 hypothyroid 数据集上，取得了最高的测试准确率。尤其值得一提的是，和单个最近邻中心分类器相比。FeatNearMean 算法在 German 数据集上测试准确

率有所提高，而所有其他的集成学习算法都没有取得这样的效果。

5.5 本章小结

本章在 Boosting 算法的基础上，根据贝叶斯分类器和最近邻中心分类器的特点，在 Boosting 算法的每一轮嵌入了特征选择算法，以减小个体分类器间的错误样本区域的重叠，分别得到了 FeatBoostNBC 和 FeatBoostNearMean 算法，在 UCI 数据上的实验结果充分验证了这两种算法的有效性，性能提高很显著。尤其对存在无用特征的数据集上，得到的测试准确率显著高于对应的 Boosting 算法，分析和实验结果表明 FeatBoostNBC 和 FeatBoostNearMean 算法改进了 Boosting 算法的性能，很好的解决了 Boosting 算法对噪声特征敏感的问题，是一种鲁棒性很强的集成学习算法。

虽然本章设计的嵌入特征选择的集成学习算法是在贝叶斯和最近邻中心分类器上实现的，但该思路同样可以应用于其他分类算法，如决策树和神经网络分类器，只要可以从分类器得出特征对分类的作用，就可以应用该思路设计相应的集成学习算法。

结 论

图像处理、信息检索以及生物信息学等大规模机器学习问题的不断涌现，对已有的特征选择算法和机器学习算法提出了严峻的挑战，迫切需要适应大规模数据的准确性和运行效率等综合性能较好的特征选择算法以及机器学习算法。本文在大规模数据的特征选择以及基于特征选择的集成学习上开展了一些研究工作。

在高维数据的特征选择方面，我们在分析特征选择算法的 Filter 和 Wrapper 模式优缺点及互补性基础上，基于 Relief 特征评估算法，主要做了以下两部分工作：

1. 设计了两阶段串联型组合式特征选择算法，分别为 Filter-Filter 模式的 ReCorre 算法和 Filter-Wrapper 模式的 ReSBSW 算法，其中 ReCorre 算法第一步采取 Relief 去除无关特征，第二步利用相关分析方法去除冗余特征；ReSBSW 算法第一步采取 Relief 去除无关特征，第二步采用分类器的准确率作为评估标准，利用顺序后向搜索法去除冗余特征。在人工数据集和实际数据集上的实验表明，ReCorre 方法能够较好的去除冗余特征，且运行效率较高，但总体比较，ReSBSW 算法的测试准确率高于 ReCorre 算法。

2. 提出了 Relief 和遗传算法的耦合式特征选择算法，算法采用 Relief 特征评估指导遗传算法种群初始化，目的是提高遗传算法搜索近似最优解的速度。在 17 个高维数据上的实验结果表明，从分类准确率，特征子集大小以及运行时间多角度考察，该方法取得了良好的综合性能。

集成学习是近年来机器学习领域的研究热点之一，通常可以显著提高学习算法的泛化性能。在集成学习领域，基于样本的各种集成学习算法研究较广泛，而基于特征选择的集成学习研究则相对较少，但对高维数据的模式识别问题，基于特征选择的集成学习具有运行效率高的显著优势。本文在基于特征选择的集成学习方面开展了研究，主要工作有以下两方面：

1. 提出一种适于高维数据的基于两步式特征选择的集成学习算法

ReFeatEn。ReFeatEn 算法从个体分类器准确率和个体分类器间差异度两方面出发，分两阶段产生用于构建不同个体分类器的特征子集：第一阶段利用 Relief 特征评估结果依概率从原始特征集合选择特征组成特征子集，第二阶段通过精调使各特征子集间满足一定的差异度。实验表明，在特征维数较高，特征间关系较复杂的数据集上，ReFeatEn 算法的测试准确率始终优于或相当于 Bagging、Boosting，以及 RandFeatEn。ReFeatEn 的运行速度远高于 Bagging 和 Boosting 算法，而且适于并行运行，是一种适用于高维数据的基于特征选择的集成学习算法。

2. 提出了将特征选择嵌入到 Boosting 算法的思路，并设计了总体算法框架。据此，分别针对朴素贝叶斯分类器和最近邻中心分类器设计了相应的集成学习算法，解决了 Boosting 算法对噪声特征较敏感的缺陷，得到的测试准确率显著高于对应的 Boosting 算法，是一种鲁棒性很强的集成学习算法。

在高维数据的特征选择以及基于特征选择的集成学习方面，还有很多研究工作需要开展。特征选择和样本选择的结合是一个可能的研究方向，不同的样本聚集区域，也许应该选用不同的特征进行学习，本文提出的将特征选择嵌入到 Boosting 算法中就是这个方向的一个尝试。另外，本文研究的为有监督的学习，在无监督的学习领域，如何进行特征选择是未来一个挑战性的研究方向。另外，和基于样本的集成学习的理论分析较多相比，在基于特征选择的集成学习方面理论研究较弱，开展基于特征选择的集成学习理论研究也是一个很有挑战性的研究方向。

参考文献

[Aha 1994] Aha D W, Bankert R L. Feature selection for case-based classification of cloud types An empirical comparison. In: Aha D W Eds. In Working Notes of the AAAI94 Workshop on Case-based Reasoning. Menlo Park, CA.: AAAI Press, 1994, 106-112

[Almuallim 1991] Almuallim H, Dietterich T G. Learning with many irrelevant features. In: Fisher D, Lenz J H, Eds. Proceeding of the 9th National Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 1991, 547-552

[Almuallim 1992] Almuallim H, Dietterich T G. Efficient algorithms for identifying relevant features. Proceedings of the 9th Canadian Conference on Artificial Intelligence. San Fransisco: Morgan Kaufmann, 1992, 38-45

[Alpaydin 1993] Alpaydin, E. Multiple networks for function learning. IEEE International Conference on Neural Networks. NJ: IEEE press, 1993, 1: 9-14

[Anthony 1997] Anthony M. Probabilistic analysis of learning in artificial neural networks: the PAC model and its variants. Neural Computing Surveys, 1997, 1: 1- 47

[Atkeson 1997] Atkeson C, Moore A, Schaal S. Locally weighted learning. Artificial Intelligence Review, 1997, 11(1-5): 11-73

[Battiti 1994a] Battiti R. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Network, 1994, 5(4): 537-550

[Battiti 1994b] Battiti R, Colla A M. Democracy in neural nets: voting schemes for classification. Neural Networks, 1994, 7(4): 691-707

[Bauer 1999] Bauer E, Kohavi R. An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Machine Learning, 1999, 36 (1- 2): 105- 139

[Bishop 1995] Bishop C M. Neural Networks for Pattern Recognition. Oxford: Clarendon Press, 1995

[Blum 1992] Blum A L. Learning boolean functions in an infinite attribute space. Machine Learning, 1992, 9(4): 373-386

[Blum 1997] Blum A L, Langley P. Selection of relevant features and examples in machine learning. Artificial Intelligence, 1997, 97(2): 245-271

[Blum 1999] Blum A L, Kalai A, Langford J. Beating the hold-out: bounds for K-fold and progressive cross-validation. Proceedings of the 12th Annual Conference on Computational Learning Theory, 1999, 203-208

[Breiman 1984] Breiman L, Friedman J H, et al. Classification and Regression Trees. Wadsforth International Group, 1984.

[Breiman 1996a] Breiman L. Bagging predictors. Machine Learning, 1996, 24(2): 123- 140

[Breiman 1996b] Breiman L. Bias, variance, and arcing classifiers. Department of Statistics, University of California at Berkeley: Technical Report 460, 1996

[Breiman 1998] Breiman L. Arcing classifiers. Annals of Statistics, 1998, 26(3): 801-849

[Caruana 1994a] Caruana R A, Freitag D. Greedy attribute selection. In: Cohen W W, Hirsh H, Eds. Proceedings of the 11th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 1994, 28-36

[Caruana 1994b] Caruana R A, Freitag D. How useful is relevance? Working notes of the AAAI Fall Symposium on Relevance. Menlo Park, CA.: AAAI Press, 1994, 25-29

[Casillas 2001] Casillas J, Cordon O, et al. Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems. Information Sciences, 2001, 136(1-4): 135-157

[Cover 1974] Cover T M. The best two independent measurements are not the two best. IEEE Transactions on System, Man and Cybernetics, 1974, 4(1): 116-117

[Cunningham 2000] Cunningham P, Carney J. Diversity versus quality in classification ensembles based on feature selection. In: López de Mántaras R, Plaza E, Eds. The 11th European Conference on Machine Learning. LNCS, Springer-Verlag Heidelberg, 2000, 1810: 109-116

[Das 2001] Das S. Filters, wrappers and a boosting-based hybrid for feature selection. In: Brodley C E, Danyluk A P, Eds. Proceeding of the 18th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 2001, 74-81

[Dash 2000] Dash M, Liu H. Feature selection for clustering. 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, 110-121

[Dash 2003] Dash M, Liu H, Consistency-based search in feature selection, Artificial Intelligence, 2003, 151(1-2): 155-176

[Davies 1994] Davies S, Russl S. Np-completeness of searches for smallest possible feature sets. Proceedings of the AAAI Fall 94 Symposium on Relevance. Menlo Park, CA.: AAAI Press, 1994, 37-39

[Dietterich 1995] Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 1995, 2: 263-286

[Dietterich 2000a] Dietterich T G. Ensemble methods in machine learning. Lecture Notes in Computer Science, 2000, 1857: 1-15

[Dietterich 2000b] Dietterich T G. An experimental comparision of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 2000, 40(2): 139-157

[Doak 1992] Doak J. An evaluation of feature selection methods on their application to computer security. Technical Report CSE-92-18, Davis, Ca: University of California, Department of Computer Science, 1992

[Domingos 1999] Domingos P. The role of Occam's razor in knowledge discovery. Data Mining and Knowledge Discovery, 1999, 3(4): 409-425

[Efron 1993] Efron B, Tibshirani R. An Introduction to the Bootstrap. NewYork: Chapman & Hall, 1993

[Elkan 1997] Elkan C. Boosting and Naive Bayesian learning, Tech. Report CS97-557, Department of CS and Engineering, University of California, San Diego, USA, 1997.

[Freund 1995] Freund Y. Boosting a weak algorithm by majority. Information and Computation, 1995, 121(2): 256- 285

[Freund 1997] 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

[Freund 1998] Freund Y, Schapire R E. Discussion of the paper "Arcing classifiers" by Leo Breiman. Annals of Statistics, 1998, 26(3): 824-832

[Friedman 1997] Friedman J. On bias, variance, 0/1-loss and the curse of dimensionality. Data Mining and Knowledge Discovery, 1997, 1(1): 55-77

[Fukunaga 1990] Fukunaga K. Introduction to Statistical Pattern Recognition. San Diego, California: Academic Press, 1990

[Goldberg 1989] Goldberg D. Genetic Algorithm as Search Optimization and Machine Learning. Addison-Wesley, 1989

[Grove 1998] Grove A J, Schuurmans D. Boosting in the limit: maximizing the margin of learned ensembles. Proceedings of the 15th National Conference on Artificial Intelligence. Menlo Park, CA.: AAAI Press, 1998, 692- 699

[Guerra-Salcedo 1999] Guerra-Salcedo C, Whitley D. Genetic approach to feature selection for ensemble creation. In: Banzhaf W, Daida J, Eiben A E, eds. Proceedings of the Genetic and Evolutionary Computation Conference. Orlando, Florida, USA: Morgan Kaufmann, 1999, 1: 236-243

[Hall 2000] Hall M A. Correlation-based feature selection for discrete and numeric class machine learning. In: Langley P, Eds. Proceedings of the 17th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 2000, 359-366

[Hamming 1986] Hamming R W. Coding and information theory. Englewood Cliffs, NJ: Prentice-Hall, 1986

[Hansen 1990] Hansen, L K, Salamon P. Neural network ensembles. IEEE Transations on Pattern Analysis and Machine Intelligence, 1990, 12(10): 993-1001

[Ho 1998] Ho T H. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 832-844

[Holland 1975] Holland J. Adaption in natural and artificial systems. Ann Arbor, Michigan: University of Michigan Press, 1975

[Huang 1999] Huang Y, Shian-Shyong T, Wu G, et al. A two-phase feature selection methods using both filter and wrapper. Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics. 1999, 2: 132 –136

[Hyafil 1976] Hyafil L, Rivest R L. Constructing optimal binary decision trees is NP-complete. Information Processing Letters, 1976, 5(1): 15-17

[Inza 2001] Inza I, Larra$\widetilde{n}$aga P, Sierra B. Feature subset selection by Bayesian networks based on optimization. Artificial Intelligence, 2001, 123(1-2): 157-184

[Jain 1987a] Jain A K, Chandrasekaran B. Dimensionality and Sample Size Considerations in Pattern Recognition Practice. Handbook of Statistics, 1987, 2: 835-955

[Jain 1987b] Jain A K, Dubes R., Chen C. Bootstrap techniques for error estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1987, 9(5):628-633

[Jain 1997] Jain A k, Zongker D. Feature selection: evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(2): 153-158

[Jain 2000] Jain A k, Duin R, Mao J C. Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22 (1): 4~37

[John 1994] John G, Kohavi R, Pfleger K. Irrelevant features and the subset selection problem. In: Cohen W W, Hirsh H, Eds. The Eleventh International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 1994, 121-129

[Kearns 1988] Kearns M, Valiant L G. Learning Boolean formulae or factoring. Aiken Computation Laboratory, Harvard University, Cambridge, MA, Technical Report: TR21488，1988

[Kira 1992a] Kira K, Rendell L A. The feature selection problem: traditional methods and a new algorithm. Proceedings of the Ninth National conference on Artificial Intelligence, 1992: 129-134

[Kittler 1978] Kittler J. Feature set search algorithms. Pattern Recognition and Signal Processing. 1978, 41-60

[Kohavi 1994] Kohavi R, Frasca B. Useful feature subsets and rough set reducts. The Third International Workshop on Rough Sets and Soft Computing, 1994: 310-317

[Kohavi 1997] Kohavi R, John G H. Wrappers for feature subset selection. In Artificial Intelligence journal, special issue on relevance, 1997, 97(1-2): 273-324

[Koller 1996] Koller D, Sahami M. Toward optimal feature selection. In: Saitta L, Eds. Proceedings of the Thirteenth International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 1996, 284-292

[Kononenko 1994] Kononenko I. Estimation attributes: analysis and extensions of RELIEF. Proceedings of the 1994 European Conference on Machine Learning, 1994, 171-182

[Krogh 1995] Krogh A, Vedelsby J. Neural network ensembles, crossvalidation, and active learning. In: Tesauro G, Touretzky D, Leen T eds. Advances in Neural Information Processing Systems 7. Cambridge, MA: MIT P ress, 1995, 231- 238

[Kudo 2000] Kudo M, Jack S. Comparison of algorithms that select features for pattern classifiers. Pattern Recogniton, 2000, 33: 25-41

[Kuncheva 2003] Kuncheva L I, Whitaker C J, Measures of diversity in classifier ensembles. Machine Learning, 2003, 51: 181-207

[Kwok 1990] Kwok S W, Carter C. Multiple decision trees. Uncertainty in Artificial Intelligence, 1990, 4: 327-335

[Langley 1993] Langley P, Iba W. Average-case analysis of a nearest neighbour algorithm. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, 1993, 2: 889-894

[Langley 1994] Langley P. Selection of relevant features in machine learning. Proceedings of the AAAI Fall Symposium on Relevance. Menlo Park, CA.: AAAI Press, 1994, 140-144

[Langley 1997] Langley P, Sage S. Scaling to domains with many irrelevant features. Computational Learning Theory and Natural Learning Systems. Cambridge, MA: MIT Press, 1997, vol. 4

[Lee 2001] Lee H M, Chen C M, et al. An efficient fuzzy classifier with feature selection based on fuzzy entropy. IEEE Transactions on Systems and Cybernetics-Part B: Cybernetics, 2001, 31(3): 426-432

[Leonardis 2000] Leonardis A, Bischof H. Robust recognition using eigenimages. Computer Vision and Image Understanding. 2000, 78 (1): 99~118

[Lewis 1962] Lewis P M, The characteristic selection problem in recognition system. IRE Transaction on Information Theory, 1962, 8:171-178

[Liu 1998a] Liu H, Motoda H, Dash M. A monotonic measure for optimal feature selection. Proceedings of the 10th European Conference on Machine Learning. Berlin: Springer, 1998, 101-106

[Liu 1998b] Liu H, Motoda H. Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers, 1998

[Liu 2002] Liu H, Motoda H, Yu L. Feature selection with selective sampling. In: Sammut C, Hoffmann A G, Eds. Proceedings of the 19th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 2002, 395-402.

[Maclin 1995] Maclin R, Shavlik J W. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. Proceedings of the 14th International Joint Conference on Artificial Intelligence. San Fransisco: Morgan Kaufmann, 1995, 524-530

[Marko 2001] Marko R S, Kononenko I. Comprehensible interpretation of relief estimates. In: Brodley C E, Danyluk A P, Eds. Proceedings of the 18th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 2001, 443-440

[Martin-Bautista 1999] Martin-Bautista M J, Vila M-A. A survey of genetic feature selection in mining issues. Proceedings of the 1999 Congress on Evolutionary Computaion. NJ: IEEE Press, 1999, 13-23

[Moore 1994] Moore A W, Lee M S. Efficient algorithms for minimizing cross validation error. In: Cohen W W, Hirsh H, Eds. Proceedings of the 17th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 1994, 190-198

[Natendra 1977] Natendra P, Fakunaga K. A branch and bound algorithm for feature subset selection. IEEE Transations on Computers, 1977, 26(9):917-922

[Ng 1998] Ng A Y. On feature selection: learning with exponentially many irrelevant features as training examples. In: Shavlik J W, Eds. Proceedings of the 15th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 1998, 404-412.

[Opitz 1996] Opitz D, Shavlik J W. Generating accurate and diverse members of a neural network ensemble. In: Touretzky D S, Mozer M C, Hasselmo M E, Eds. Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 1996, 8: 535-541

[Opitz 1999a] Opitz D, Maclin R. Popular Ensemble Methods: An empirical study. Journal of Artificial Intelligence Research, 1999, 11: 169-198

[Opitz 1999b] Opitz D. Feature selection for ensembles. In Proceedings of the 16th National Conference on Artificial Intelligence, 1999, 379-384

[Opitz 1999c] Opitz D, Shavlik J. A genetic algorithm approach for creating neural network ensembles. Combining Arificial Neural Nets, 1999, 79-99

[Parmanto 1995] Parmanto B, Munro P W, Doyle H R. Improving committee diagnosis with resampling techniques. In Tourezky D S, Mozer MC, Hesselmo M E Eds. Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 1995, 8: 882-888

[Perrone 1993] Perrone M P, Cooper L N. When networks disagree: Ensemble method for neural networks. Artificial Neural Networks for Speech and Vision, 1993, 126-142

[Pfahringer 1995] Pfahringer B. Compression-based feature subset selection. IJCAI-95 Workshop on Data Engineering for Inductive Learning, 1995, 101-106

[Provan 1995] Provan G M, Singh M. Learning bayesian networks using feature selection. Proceedings of the 5th International Workshop on AI and Statistics, 1995, 450-456

[Pudil 1994] Pudil P, Novovicova J, Kittler J. Floating search methods in feature selection. Pattern Recognition Letters, 1994, 15(11): 1119-1125

[Quinlan 1983] Quinlan J R. Learning efficient classification procedures and their application to chess end games. Machine Learning: An artificial intelligence approach, San Francisco, CA: Morgan Kaufmann, 1983, 463-482

[Quinlan 1993] Quinlan J R. C4.5: programs for machine learning. San Francisco: Morgan Kaufmann, 1993

[Quinlan 1996] Quinlan J R. Bagging, boosting, and C4.5. In the 13th National Conference on Artificial Intelligence. Menlo Park, CA : AAAI press, 1996, 725-730

[Ricci 1997] Ricci F, Aha D W. Extending local leaners with error-correcting output codes. Tech. rep., Naval Center for Applied Research in Artificial Intelligence, Washington D.C. 1997.

[Robit 2000] Rohit L, Ravi K. Adaptive linear dimensionality reduction for classification. Pattern Recognition, 2000, 33 (2): 185 ~194

[Schapire 1990] Schapire R E. The strength of weak learnability. Machine Learning, 1990, 5(2): 197-227

[Schapire 1998] Schapire R E, Freund Y, Bartlett Y, et al. Boosting the margin: a new explanation for the effectiveness of voting methods. Annals of Statistics, 1998, 26(5): 1651-1686

[Schapire 1999] Schapire R E. A brief introduction of boosting. In the16th International Joint Conference on Artificial Intelligence, Stockho lm, Sweden, 1999, 1401-1405

[Setiono 1997] Setiono R, Liu H. Neural-Network feature selector. IEEE Transactions on Neural Networks, 1997, 8(3): 654-662

[Shipp 2002] Shipp C A, Kuncheva L I. Relationships between combination methods and measures of diversity in combining classifiers. Information fusion, 2002, 3(2): 135-148

[Siedlecki 1988] Siedlecki w, Sklansky J. On automatic feature selection. International Journal of Pattern Recognition and Artificial Intelligence, 1988, 2(2): 197-220

[Siedlecki 1989] Siedlecki w, Sklansky J. A note on genetic algorithms for large-sclae feature selection. Pattern Recognition Letters, 1989, 10:335-347

[Sollich 1996] Sollich P, Krogh A. Learning with ensembles: How over-fitting can be useful. In: TouretzkyD, MozerM, HasselmoM eds. Advances in Neural Information Processing Systems 8. Cambridge, MA: MIT Press, 1996, 190- 196

[Somol 1999] Somol P, Pudil P, Novovicova J, et al. Adaptive floating search methods in feature selection. Pattern Recogniton Letters, 1999, 20(11-13): 1157-1163

[Torrieri 2000] Torrieri D. The eigenspace separation transform for neural-network classifiers. Neural Networks, 2000, 12 (3): 419~427

[Tsymbal 2001] Tsymbal A, Puuronen S, Skrypnyk I. Ensemble feature selection with dynamic integration of classifiers. Proc. Int. ICSC Congress on Computational Intelligence Methods and Applications CIMA'2001. NAISO Academic Press, 2001, 558-564

[Tumer 1996] Tumer K, Ghosh J. Error correlation and error reduction in ensemble classifiers. Connection Science, 1996, 8(3-4): 385-404

[UCI] Urphy P, Aha D. UCI ML Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[Vafaie 1992] Vafaie h, De J K. Genetic algorithm as a tool for feature selection in machine learning. Proceedings of the 4th International Conference on Tools with Artificial Intelligence. IEEE Computer Society Press, 1992, 200-204

[Vapnik 1995] Vapnik V N. The Nature of Statistical Learning Theory. New York：Springer-Verlag, 1995

[Weiss 1991] Weiss S M, Kulikowski C. Computer Systems That Learn. Morgan Kaufmann, 1991

[Weston 2000] Weston J, Mukherjee S, et al. Feature selection for SVMS. In Advances in Neural Information Processing Systems. 2000, 13: 668-674

[Wickramaratna 2001] Wickramaratna J, Holden S, Buxton B. Performance degradation in boosting. In: Kittler J, Roli F, Eds. Proceedings of the 2nd International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science. Berlin: Springer, 2001, 2096: 11－21

[Wolpert 1999] Wolpert D H, Macready W G. An efficient method to estimate bagging's generalization error. Machine Learning, 1999, 35(1): 41- 55

[Xing 2001] Xing E, Jordan M, Karp R. Feature selection for high-dimensional genomic microarray data. In: Brodley C E, Danyluk A P, Eds. Proceedings of the 18th International Conference on Machine Learning. San Fransisco: Morgan Kaufmann,2001, 601-608

[Yang 1997] Yang Y, Pederson J O. A comparative study on feature selection in text categorization. In: Fisher D H, Eds. Proceeding of the Fourteenth International Conference on Machine Learning. San Fransisco: Morgan Kaufmann, 1997, 412-420

[Yang 1998] Yang J, Vasant H. Feature subset selection using a genetic algorithm. IEEE Intelligent Systems, 1998, 13: 44-49

[Yu 2003] Yu L, Liu H. Feature selection for high-dimensional data: a fast correlatin-based filter solution. In: Fawcett T, Mishra N, Eds. Proceedings of the 20th International Conference on Machine Learning. Menlo Park, CA.: AAAI press, 2003, 856-863

[Zhong 2001] Zhong n, Dong J. Using Rough Sets with Heuristics for Feature Selection. Journal of Intelligence Information Systems, 2001, 16: 199-214

[章新 1998] 章新，一种特征选择的动态规划方法．自动化学报，1998，25(4): 675-680

[张鸿宾 1999] 张鸿宾，孙广煜．Tabu 搜索在特征选择中的应用．自动化学报，1999, 25(4): 457-466

[范劲松 2001] 范劲松，方廷建．特征选择和提取要素的分析及其评价．计算机工程与应用，2001, 13: 95-99

[马少平 1997] 马少平，夏莹，朱小燕. 基于模糊方向线素的手写体汉字识别. 清华大学学报（自然科学版）, 1997, 37(3): 42-45

[李凌 2000] 李凌. 大规模数据的粗分类及其在脱机手写体汉字识别中的应用：[硕士学位论文]. 北京：清华大学计算机系，2000

[周志华 2002] 周志华，陈世福. 神经网络集成. 计算机学报，2002, 25(1): 1-8

致谢

衷心感谢我的导师王家歧教授。从选题、研究工作的开展到最后论文工作的完成都是在他的悉心指导下完成的。他为我创造了宽松自由的研究环境，并对我的研究工作给予极大的信任、支持和鼓励。同时他严谨的治学精神、务实的研究作风、敏锐的洞察力都对我产生了重要的影响。

感谢赵雁南老师对我学习和生活的关怀。感谢张钹，马少平和吴文虎老师在开题和预答辩过程中给予我的指导和建议。

感谢实验室的张剑、朱红梅、张妤、左锦宇等师兄师弟师妹们对我的关心、鼓励和支持。与大家在一起的日子是我在清华最难忘的时光。

最后我不能忘记的是我挚爱的家人。感谢他们为我带来的信心和力量，感谢他们给予我的爱。

==========

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____日 期：_____

附录　本文所用实验数据集意义说明

一、手写体汉字数据集

手写体汉字数据集的特征是基于模糊方向线素的 256 维向量[马少平 1997]，向量的每维分量为 0-255 的 8 位整数。特征的抽取是通过汉字的点阵图划分为如上图的 8×8 的方格后分别统计每个方格中横、竖、撇、捺这四种不同笔划（不同方向）的个数得到的。从方格中统计出特征分量后，使用以下方式形成 256 维特征：每格内的先后顺序为横、竖、撇、捺，而格子的顺序为先从左到右，后从上到下[李凌 2000]。

横、竖、撇、捺是在走向上不同的四种笔划特征，论文中有时为讨论方便将这四种特征抽取出来分别显示，这样就得到了 4 个 64 维的子特征集合，在论文第三章的图 3-1 就是用这种方法绘制的，在该图中，横坐标表示的序号恰好是 8×8 划分中方格的序号。

二、UCI 数据集

1． Ionosphere

描述：Ionosphere 为判断雷达信号是否正常的数据集，特征为天线收集到的电磁波信号

数据缺失情况：无

2. Isolet

描述：Isolet 为 26 个英文字母的语音识别数据库，由 617 维声学特征进行描述

数据缺失情况：无

3. Horse-colic

描述：Horse-colic 为诊断马腹绞痛是否由外科手术引起的数据集。特征包括：是否进行过外科治疗，年龄，病历号，直肠温度，心率，呼吸频率，环境温度，黏膜，内脏蠕动情况，腹胀程度等

数据缺失：近 30%的数据丢失

4. Waveform-40

描述：waveform-40 数据是 Breiman 在[Breiman 1984]书中构造的数据，为从 waveform-21 数据加入 19 维均值为 0 方差为 1 的符合高斯分布的随机特征得到的。描述三个三角形波，前 21 维数据为加入噪声后的 21 个等间隔点上的波高

数据缺失：无

5. Satimage

描述：这个数据集用来判断卫星返回图片的土壤类别，样本用 9 个区域的 4 个波段的波长来描述

数据缺失：无

6. Sonar

描述：声纳信号数据，根据从海底传回的声波信号判断是金属还是矿石。特征空间为 60 维，每维表示在固定波长范围内的能量

数据缺失：无

7. German

描述：German 为评价顾客信用度的数据集，类别分别为 good 和 bad。特征包括：支票帐户的状态，信用卡历史记录状态，消费用途，信用卡余额，就业状况，分期付款站收入的比例，个人性别及婚姻状

况，财产情况，年龄，居住房屋所有权归属，是否留有电话，是否为外籍员工等

数据缺失：无

8. Sick-euthyroid

描述：甲状腺机能是否正常的诊断数据集，根据患者个人信息和检测结果判断患者甲状腺机能是否正常

数据缺失：部分特征的值大量缺失，部分特征的值少量缺失

9. Breast-cancer

描述：乳癌的是否存在复发情况的诊断数据集。特征信息包括年龄、肿瘤大小、位置，是否做过放射治疗等

数据缺失：少量缺失

10. Hypothyroid

描述：甲状腺机能是否正常的诊断数据集

数据缺失：最后一个特征的值大量缺失，少部分特征的值有较少缺失，大部分特征的值无缺失

11. DNA_nominal

描述：生物信号识别，根据长为 60 的 DNA 序列，判断其内是否含有生物信息学中的“Acceptors”“Donars”信号

数据缺失：无

12. Chess

描述：根据棋局当前信息判断终局胜负

数据缺失：无

13. Soybean-large

描述：大豆情况分类，根据大豆种植及生长的一些信息判断大豆种类（如是否生霉，豆荚是否有腐烂等）

数据缺失：部分缺失

14. Tokyo1

描述：根据 cpu, io,disk, 内存的信息判断服务器性能好坏的数据集

数据缺失：无

15. Anneal

描述：根据退火过程各种信息进行退火分类的数据集

数据缺失：部分特征值大量丢失

16. Nettalk

描述：324 类，从 7 个顺序字符中产生中间字符的正确发音的数据集。

缺失情况：无

17. Auto

描述：对汽车的性价比进行分类。特征包括汽车本身的机械性能指标，价格，品牌等

缺失情况：部分特征值缺失严重，部分特征值少量缺失

个人简历、在学期间的研究成果及发表的论文

基本情况:

姓名：张丽新　　**性别：女**　　**出生年月：**1975 年 11 月

学习经历:

1999-现在 博士（直读），清华大学计算机系，预计 2004 年获工学博士学位

1994-1999 本科，清华大学机械工程系，获工学学士学位

1996-1999 辅修，清华大学计算机系，获计算机辅修学位

作者完成的论文

1. 张丽新, 赵雁南, 蔡少青, 杨泽红, 刘弘宇, 王家廞. 参差比较和遗传算法的中药识别分类器. 计算机科学, 2002, 6: 102-105.
2. Zhang L X, Zhao Y N, Yang Z H, and Wang J X. Classifier for Chinese Traditional Medicine with High-Dimensional and Small Sample-Size Data. Proceeding of 4th world congress on control intelligence and automation (WCICA'02). Shanghai, China, 2002, 330-334(**EI：03047336464, ISTP**).
3. Zhang L X, Zhao Y N, Yang Z H, and Wang J X. Classification of Traditional Chinese Medicine by Nearest-Neighbor Classifier and Genetic Algorithm. Proceeding of the 5th international conference on information fusion (Fusion'2002), Maryland, USA, 2002, 1596-1601(**ISTP，INSPEC: 7412259**)
4. Zhang L X, Zhao Y N, Yang Z H, and Wang J X. Feature Selection in Recognition of Handwritten Chinese Characters. Proceeding of the 1st international conference on machine learning and cybernetics, Beijing, China, 2002. 3: 1158-1162(**EI: 03127405749, ISTP**).
5. Zhang L X, Zhao Y N, Yang Z H, and Wang J X. A Novel Hybrid Feature Selection Algorithm: using ReliefEstimation for GA-Wrapper Search. Proceeding of the 2nd international conference on machine learning and cybernetics, Xi'an, China, 2003, 380-386(**EI: 04128071045, ISTP)**.
6. 张丽新，王家廞，赵雁南，杨泽红. 机器学习中的特征选择，计算机科学, 2004(已接收).

7．张丽新，王家廞，赵雁南，杨泽红. 基于 Relief 的组合式特征选择，复旦大学学报，2004（已接收）.

8．张丽新，王家廞，赵雁南，杨泽红. 一种适用于高维数据的基于特征选择的集成学习方法，同济大学学报增刊，2004（已接收）

在学期间获奖情况：

2002	清华大学“一二.九优秀辅导员奖”
2001	清华大学研究生社会工作一等奖学金
	清华之友—IS-ONE 安氏一等奖学金
2000	清华大学优良奖学金
1999	清华大学优良毕业生
1998	清华之友—高田一等奖学金
1996~1998	清华大学“优秀学生”
1997	清华之友—通用电气一等奖学金
1996	清华大学优秀奖学金
1995	清华大学优良奖学金

攻读博士期间参加科研项目情况

1. “863”重点项目“基于 DSP 的控制器整机开发”，荣获“2001 年中国高校科学技术二等奖
2. 国家“863”项目“大型复杂曲面钢板水火成型智能机器人控制系统产品化样机”
3. 北京市科学技术基金项目“虚拟现实实验系统”研究，荣获 2002 北京市科技进步三等奖
4. 清华大学基础研究基金项目“基于生数据的海量信息检索”