上海交通大学

博士后学位论文

机器学习中若干特征选择算法研究

姓名：李云

申请学位级别：博士后

专业：计算机软件与理论

指导教师：吕宝粮

20070901

内 容 摘 要

本报告在分析特征选择相关性质的基础上，研究并设计了多种环境下特定的特征选择算法，而这些也是目前特征选择的研究热点。

本报告的主要结论是：

1. 利用指数熵作为特征排序准则，并结合改进的模糊特征评价指标，设计了一种新的非监督特征选择方法，效果很好。

2. 将维数约简中两种典型方法（特征抽取与特征选择）相结合，利用 K 近邻聚类，设计一类基于主成分分析的特征选择方法，将没有实际意义的主成分投影到原始特征空间，选择关键的原始特征。

3. 在深刻分析 K 近邻分类器的损失函数基础上，提出新的基于 K 近邻分类损失-间隔的特征选择评价准则和算法，并利用能量模型进行理论分析。实验结果表明该算法能获得比其它进行 K 近邻规则的特征选择算法（如 Simba、Mitra 和 Relief）更好的性能。

关键词：特征选择，主成分分析，损失间隔

Abstract

Based on the analysis of characteristics of feature selection, some special algorithms in different conditions are proposed in this report. The major works are introduced as follows:

1. The exponential entropy is utilized as the ranking index, and the fuzzy feature evaluation index is improved to design a new unsupervised feature selection method, the experimental results have shown its good performance.

2. The two major methods for dimensionality reduction, i.e., feature extraction and feature selection, are combined to design a feature selection algorithm based on PCA (Principal Component Analysis), which can map the principal component without practical meaning to original feature space, and identify critical original features.

3. Based on the deep analysis of loss function of KNN classification, a new feature selection criterion and algorithm based on the loss-margin of KNN classification are proposed. At the same time, the theoretic analysis from the energy model points is given. The experimental results have shown the algorithm can get higher performance than other algorithms based KNN rule, such as Simba, Mitra's and Relief.

Keywords: Feature Selection; Principal Component Analysis; Loss-Margin

1 绪 论

1.1 课题的意义

特征选择是统计模式识别、机器学习和数据挖掘等领域的一个热门研究课题，受到广泛的重视。

特征选择已广泛应用到文本分类、图像检索、客户关系管理、入侵检测和基因分析等方面。

因此，研究特征选择算法具有重要的学术意义和实用价值。

1.2 国内外研究现状简介

特征选择从其输出结果来说可以分为两类：① 连续的特征选择，保留所有的特征，只是为每个特征赋予不同的权值；② 二值特征选择，即从原始特征集中选择一个或几个特征子集来降低模式的维数，且满足一定的评价准则[1]。描述为“根据一定的评价准则从含有 n 个特征的集合中寻找出 n'（$n'<n$）个相关特征”。特征选择算法根据不同的评价准则，大体上可以分为三类：过滤器模型，封装器模型以及混合模型[2]。过滤器模型是将特征选择作为一个预处理过程，利用数据的内在特性对选取的特征子集进行评价，独立于学习算法，而封装器模型则将后续学习算法的结果作为特征子集评价准则的一部分。一般过滤器模型的时间复杂度较低，效果欠佳，而封装器模型的时间复杂度较高，效果较好。另外，混合模型试图利用上面两种模型的优点，在不同的搜索阶段利用不同的评价准则。

特征选择的具体作用体现在三个方面：提高泛化能力，即对未知样本的预测能力；决定相关特征，即与学习任务相关的特征；特征空间的维数约简[2]。

当训练样本的类别已知，即监督的特征选择来说，实际工作中有三种特征选择问题：① 是从原始特征集中选出固定数目的特征，使得分类器的错误率最小这是一个无约束的组合优化问题；② 是对于给定的允许错误率，求维数最小的特征子集，这是一种有约束的最优化问题；③ 是在错误率和特征子集的维数之间进行折中[1, 4]。上述三种特征选择都属于 NP 难的问题，除了穷尽搜索之外，不能保证得到最优解，在原始特征维数 n 较小时，尚可用穷举法求解，对稍大些的 n，如 $n>20$，穷尽搜索实际上已经不可行。在评价准则对应的目标函数满足单调性的前提下，*Narendra* 和 *Fukunaga*[5]引入分枝限界法（BB）来求解最优特征子集，但单调性前提在实际问题中往往不能满足。另外，即使 BB 算法减少了 99.9%的工作量，算法的复杂度与 n 仍是指数关系，当 n 较大时，BB 算法仍不可行。

由于求最优解的计算量太大，人们一直在致力于寻找能得到较好次优解的算法。60 年代早期的方法是，在特征间相互独立的假设下，单独研究每一特征的类可分性或熵（当时分类器错误率的研究也刚起步），然后取单独使用效果最好的组合在一起。这类方法没有考虑到特征之间的相互作用，结果自然不理想。*Cover*[6]指出即使满足相互独立的条件，两个单独使用最好的特征组合起来，也不能保证是最好的组合，极端情况下，甚至可能成为最差的组合。此后出现的顺序前进法（SFS）、顺序后退法（SBS），以及改进的广义顺序前进法（GSFS）、广义顺序后退法（GSBS）等实际上都属于贪心一类的算法。这些算法考虑到了特征的相互作用，但也存在明显的缺点，特征一旦被加入或者被剔除，以后将不再改变，即所谓“筑巢”（*nesting*）效应。为了克服这些缺陷，出现了增 *l* 减 *r* 法（PTA），另外由 *Backer* 和 *Sahlppe*[7]提出的极大一极小算法（MM）是一种速度较快的算法，但实验结果表明，当 *N* 较大时．这种算法的解的质量很差[8]。

到了上世纪九十年代，*Siedlecki* 和 *Sklansky*[9, 10]把遗传算法应用到特征选择中，获得了较好的结果，但遗传算法常出现过早收敛的问题。*Pudil*[11]等提出了顺序浮动前进法（SFFS）和顺序浮动后退法（SFBS）。这两种算法可以理解为增 1 减 *x* 法和减 1 增 *x* 法，*x* 根据搜索情况动态地变化。算法对增 *l* 减 *r* 法的改进是，变固定的 *l* 和 *r* 为浮动的，减少了不必要的回溯及在需要时增加回溯的深度，解决了参数 *l* 和 *r* 取值难于确定的问题。根据文献［12］的实验， SFFS 和 SFBS 算法的解接近于最优解，而计算速度要快于分枝限界法。

早期的研究主要集中于监督的特征选择研究，但是，最近的进展表明上面所提到的一些算法通过一定的改造后，可以有效地用于非监督特征的选择，即样本的类别是未知的情况[13]。

直到 1997 年，很少有领域所使用的特征维数超过 40，然而近几年，情况发生了很大的变化，许多领域所涉及的特征维数都非常高，如基因选择，文本检索等[14]。对于这种高维特征选择，主要采用排序、特征关联等，目前仍是研究热点，在国际知名刊物和会议上，每年都有相关文献出现。

我国特征选择的研究主要从九十年代开始。其中，具有代表意义的是陈彬、洪家荣等人于 1997 年在计算机学报上发表了《最优特征子集选择问题》[15]，该文证明了最优特征子集选择是 NP 难题，并给出了一个启发式算法，另外，张鸿宾[4]等人利用 Tabu 搜索来进行特征选择，在维数较高时，也能收到不错的效果。还有其它一些论文出现在“计算机学报”，“自动化学报”，“电子学报”等重要刊物上[16,17]。

1.3 主要研究内容和创新点

本报告研究的主要内容是对几种特定的选择算法进行分析，主要创新点在：

1）监督的特征选择算法较多，而非监督特征选择算法较少，本文提出了一种新的非监督高维特征选择算法，它通过指数熵对特征的聚类性能进行排序，然后利用一种扩展的模糊特征评价指标作为评价准则来获取特征子集，这是一种过滤器方法，时间复杂度较低，且效果较好。

2）特征维数约简主要有两种方法：特征选择和特征抽取。而目前大部分的研究都将二者独立开来。本文考虑将特征抽取与特征选择相结合，利用 K 近邻聚类，提出了一种基于主成分分析 PCA 的特征选择方法，并设计多种相似性度量方法来计算 PCA 变换矩阵中行成分之间的相似性。

3）分类间隔是目前机器学习研究领域的一个热点，本文提出了一种基于 K 近邻分类间隔的特征选择算法，在分析 K 近邻分类损失函数的基础上， 提出新的评价准则，并通过能量模型对其进行理论分析。在基准数据集和人脸图像上的实验表明该算法效果很好。

1.4 本报告的组织结构

本报告共分两部分六章，第一部分，说明特征选择算法的现状和相关特性（包括第 1、2 章）；第二部分，各种特定条件下的特征选择算法设计（第 3、4、5）。各章具体安排如下：

第一章，绪论

介绍特征选择算法的作用和意义，简要概述特征选择算法的国内外现状，本报告研究的主要内容和创新点。

第二章，特征选择算法的特性研究

介绍特征选择算法的基本结构、组成部分、算法形式，并分析了选用合适的特征选择算法所需要考虑的因素。

第三章，非监督高维特征选择算法

提出了一种新的基于排序的非监督高维特征选择算法，是一种过滤器方法，并分析其时间开销和局限性，通过实验验证其正确性和有效性。

第四章，基于主成分分析的特征选择算法研究

在报告中首先分析了基于主成分分析 PCA 的特征选择算法框架，并且提出了一种基于 K 近邻规则的新方法，同时提出了多个新的度量准则来计算 PCA 变换矩阵中行成分相似性。

第五章，基于分类间隔的特征选择算法研究

为了有效地结合封装器和过滤器的优点，本章在研究 K 近邻分类损失函数的基础上，提出了一种针对 K 近邻分类器的特征选择算法，它是基于分类间隔（Margin）的，而已有理论表明，大的分类间隔能保证分类器具有很好的推广能力。

第六章，总结

总结了作者在博士后期间的研究工作，并阐述了特征选择算法的研究方向。

2 特征选择算法的基本特性研究

特征选择是数据预处理的主要内容之一。本章介绍了常规特征选择算法的基本结构、详细分析各个组成部分、给出其形式化描述。并分析了选用合适的特征选择算法所需要考虑的因素。

2.1 特征选择算法的结构

特征选择是从一组特征中挑选出一些最有效的特征达到降低特征空间维数的目的，通常无法找到最优特征子集，并且许多与特征选择相关的问题都是 NP 难问题。

特征选择算法的基本结构图如下[13]：

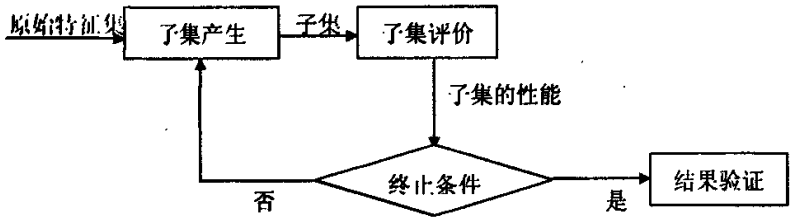


图 2.1 特征选择算法的基本结构图

Figure 2.1 The basic architecture of feature selection algorithm

一个典型的特征选择算法通常包括四个基本步骤[1, 2, 13]：

1）子集的产生（Subset Generation），这是一个搜索过程，通过一定的搜索 策略产生候选的特征子集。子集的产生是启发式搜索的必要过程，它包含两个方面的内容：首先，必须决定搜索开始点，如选择特征子集初始为空集或是整个特征集或者随机选择一个子集；其次，必须决定搜索策略，对于一个 n 维的数据集，存在 2^n 个候选子集，即使是较小的 n，穷尽搜索都是不可行的，因此需要采用一些搜索策略，包括完全搜索（Complete Search）、顺序搜索（Sequential Search）和随机搜索（Random Search）。

完全搜索：可以保证获得对于给定的评价准则是最优的特征子集，例如穷尽搜索就是一种完全搜索。但是，并不是说所有的完全搜索都是穷尽搜索，某些启发式评价函数可以用来减少搜索空间并能保证获得最优特征子集。相应的算法有分支限界法[5]和 BS（Beam Search）算法[18]。

顺序搜索：它不需要进行完全搜索，因此也不能保证获得的结果是最优的。它包括贪心爬山法的各种变化，如顺序前进法 SFS（Sequential Forward Selection）、顺序后退法 SBS（Sequential Backward Selection）以及双向选择。所有这些方法都只是每次增加或删除一个特征。另外，也可以在某一步增加（或删除）p 个特征，而在下一步删除（或增加）q 个特征（$p>q$）。顺序搜索算法相对比较简单且时间复杂度较低，一般为 $O(n^2)$ 或更低。

随机搜索：开始随机选择一个特征子集，紧跟着有两种不同的处理方式，一种是进行顺序搜索，将随机性与顺序性相结合，如随机开始的爬山法和模拟退火算法。另一种是接着再采用随机搜索获得下一个特征子集，如 Las Vegas 算法[1]。利用随机搜索可以避免局部最优，并能保证所选特征子集的最优性。

2）子集评价（Subset Evaluation），每一个候选的特征子集都根据一定的评价准则进行评价并与先前最优的特征子集进行比较。评价准则根据与学习算法的关联情况大体上可以分成两类：关联准则和独立准则。

关联准则通常应用在封装器模型的特征选择算法中，先确定一个学习算法并且利用学习算法的性能作为评价准则。对于特定的学习算法来说，通常可以找到比过滤器模型更好的特征子集，但是要多次调用学习算法，一般时间开销较大，并且可能不适合其它学习算法。例如在监督特征选择中，分类准确率是一个常用的关联准则，对于特定的分类器，利用分类准确率可以选择较好的特征子集。对于非监督特征选择来说，通常利用特定的聚类算法在所选择的子集上的聚类质量来评价特征子集，这也是一种关联准则。目前有很多启发式准则去度量聚类质量，如类的紧凑性、类内类间的距离和最大似然性等。聚类技术通常可以分成两类[19]：层次型（Hierarchical）聚类和分割型（Partitional）聚类。层次型聚类生成一个树型的聚类谱系图，根据需要可以在不同层次上选取类别个数。分割型聚类对原有数据集生成一个划分。层次型聚类方法包括基于最短距离、基于最长距离、基于均值距离的方法。分割型聚类又包括方差法（如 k-means 方法）和基于图论的方法等等；分类算法也有很多，如线性最小平方拟合、贝叶斯(Bayes)、K 近邻（K-Nearest Neighbor, KNN）、决策树、支持向量机（Support Vector Machine, SVM）、基于神经网络的分类等[20]。

独立准则通常应用在过滤器模型的特征选择算法中，试图通过训练数据的内在特性来对所选择的特征子集进行评价，独立于特定的学习算法。通常包括：距离度量、信息度量、关联性度量和一致性度量。

距离度量：常用的有可分性距离和散度等。对于两类问题和两个特征 F_1 和 F_2，如果特征 F_1 使得两类的条件概率的差别更大，则先选择 F_1，因为我们试图找到能使两类尽可能被分开的特征。如Bhattacharyya距离和Chernoff距离等[19]。

信息度量：主要是计算特征的信息增益。一个特征 F_1 的信息增益可以定义为：F_1

使得先验不确定性与期望的后验不确定性之间的差别。如果特征F_1的信息增益大于特征F_2，则F_1好于F_2。常用的有与熵相关的各种度量。

关联性度量：主要是度量以一个变量的取值去获取另一个变量值的能力，在监督特征选择中，主要关注特征与类的关联性，如果特征F_1与类C的关联性大于特征F_2与类C的关联性，则选择F_1。在非监督特征选择中，主要考虑特征之间的关联性。

一致性度量：是试图找到与全集相同分类能力的最小特征子集。而不一致性定义为如果两个样本在选定的特征子集上取值相同，却属于不同类[21]。

3）终止条件（Stopping Criterion），指算法结束所要满足的要求。它与子集的产生过程和选用的评价准则有关。经常采用的终止条件有：搜索完成；某种给定的界限如指定的特征数或循环次数等，已达到；再增加（或删除）任何特征都不能获得更好的结果；对于给定的评价准则，已获得足够好的特征子集。

4）结果验证（Result Validation），根据一定的先验知识或通过合成或现实数据集的测试来证明所选择的特征子集的性能，先验知识通常是指对进行特征选择的数据集的了解，而在实际应用中，这种先验知识是无法获得的，于是就通过特征子集对学习算法的性能影响来验证所选择特征子集的质量。

2.2 特征选择算法的伪代码

一个统一的特征选择算法模型的描述如下[1, 13]：

算法 2.1 特征选择算法

输入：*S*:训练数据集，每个样本用特征集 *FS* 表示，$|FS|=n$

J：评价准则

GS：特征子集产生方法

输出：fs_{sup} 最优特征子集

第一步：初始化：*fs*=Start-point（*FS*）；

fs_{sup}=｛根据 *J* 获取 *FS* 中最好的特征子集｝；

第二步：DO BEGIN

（a） *fs*=Search-strategy(*fs*,*GS*, *FS*)； %*生成特征子集集合*

（b） fs'=｛根据 *J* 获取 *FS* 中最好的特征子集｝；

（c）IF (*J*(fs')好于 *J*(fs_{sup})) %*评价并比较当前特征子集*

$fs_{sup}=fs'$；

END UNTIL stop(*J*, *fs*)； %*满足终止条件*

第三步：OUTPUT fs_{sup}

对于给定的数据集 *S*，每个样本用一个 *n* 维的向量描述，算法从一个选定的特征

子集 fs 开始搜索（对于初始的特征子集，可能是空集、全集和任意随机选择的特征子集，这时 fs 中只有一个特征子集$|fs|=1$，则$fs_{sup}=fs$，而在某些情况下，初始点可能是特征子集的集合$|fs|>1$，如遗传算法中，初始群体中包含了多个个体，就需要根据判断准则 J 从中选择若干个较优子集，用于以后的交叉、变异[9]），然后根据一定的搜索策略在特征空间进行搜寻。根据选定的评价准则对每一个产生的特征子集进行评价，并与以前最好的特征子集进行比较。如果它更好，则作为当前最好的子集。整个搜索过程一直持续到满足特定的终止条件。算法输出的是相对于选定评价准则的最优特征子集。

如果评价准则（J）采用关联准则就获得封装器模型的伪代码描述，如果采用独立准则就可以获得过滤器模型的伪代码描述。另外通过改变搜索策略（search-strategy）和选择关联准则或独立准则下具体的评价准则就可以设计相应模型下的许多具体的算法。而对于混合模型只是将过滤器与封装器进行组合，先通过过滤器获取不同维数的最优特征子集，然后利用封装器从中选择全局最优特征子集。将过滤器与封装器的伪代码进行一定的组合就可以得到混合模型的伪代码，这里不作赘述。

2.3 特征选择算法的选用

由于特征选择算法研究的不断深入，出现了大量的特征选择算法，以后还会更多。随着特征选择算法的增多，如何选用合适的特征选择算法便成为一个紧迫的问题。针对具体的应用，除了具体领域的知识外，还需要对特征选择算法的技术细节有所了解。下面从用户的角度研究特征选择算法的选用。

一般来说，需要考虑以下因素：特征选择的目的、时间要求、期望的输出结果、希望选择特征数与原始特征数的比例、类别的信息、特征的类型、数据的质量和特征数与样本数的比例。

1）特选择的目的，大体上分为可视化、数据理解、数据去噪、冗余和不相关特征的剔除、性能提高。而特征选择算法有三种模型，过滤器、封装器和混合模型。由于不同特征选择目的对应着不同的评价准则，可以相应地将不同的特征选择目的纳入这三种模型。例如冗余和不相关特征的剔除，可以采用过滤器模型，无偏差且较快；要提高学习算法的性能，可以采用封装器模型。

2）时间要求，特征选择过程对时间开销的关注程度，不同的时间限制影响着算法的选用，首先是搜索策略方面，如果对时间没有要求，可以采用完全搜索去获取更优解，否则，需采用顺序搜索或随机搜索。另外是算法模型，不同的模型有不同的计算复杂性。过滤器模型通常时间开销较少，在学习算法的时间开销很大，且不是必需时，就可以选用过滤器。

3）特征选择的输出形式，可分为两类：排序的列表和最小的子集。它们的区别是所选择的特征是否排序。

4）希望选择的特征数与原始特征数的比例，在决定合适的搜索策略时非常有用。如果希望选择的特征数很少，可以采用前向选择策略。如果希望选择的特征很多，则可以采用反向剔除策略。

5）类别信息，如果知道样本的类别信息，可以选用监督特征选择算法，若类别信息未知，就需要非监督的特征选择算法，结合上一节算法分类中的学习类型，就可以选用合适的特征选择算法。

6）不同的特征类型要求不同的处理机制。通常的特征类型有连续和离散特征，还有名词性特征。当一个数据集含有多种类型的特征时，问题就变得很复杂，需要考虑每个特征的影响，在选用特征选择算法时，应认识和考虑实际应用中的复杂性。后面将列表描述不同特征选择算法所处理的特征类型。

7）数据质量是指数据集中是否包含了缺值或噪声数据。不同的特征选择算法要求不同的数据质量。一些算法需要对数据进行预处理，如值的离散化和补缺，而其它的可能没有这方面的要求。

8）特征数与样本数的比例，通常样本数远大于特征数，但是，有时特征数可能很大而样本数却很小，如文本挖掘和基因分析。在这种情况下，需要关注那些在特征数上做更多强化工作的算法。

除了上面所列的因素，另外领域知识也可以帮助选用合适的特征选择算法，例如，有经验的医生大概知道那些特征在判断病情时更有效，便可以加速算法的选用过程。

综合上面所列的各种因素和上一节的算法分类，一般可以找到合适的特征选择算法。

2.4 本章小结

特征选择是机器学习和模式识别中的一个关键问题，也是一个棘手问题，许多与特征选择相关的问题都是 NP 难问题。本章系统地介绍了特征选择的四个基本步骤，并详细介绍每个步骤目前所包含的具体内容。为了便于用户选用合适的特征选择算法，分析了用户在选用特征选择算法时所考虑的一些因素与算法本身的联系。将二者结合起来就是一个理想的特征选择算法的选用平台。

3 分层模糊非监督特征选择过滤器算法

本章提出了一种非监督的特征选择方法，它是一种两层的过滤器方法，分别消除冗余特征和不相关特征。冗余特征可以采用任意的聚类方法，而作者提出了一种新的不相关特征过滤器：先根据特征聚类性能对其进行排序，然后利用修改后的模糊特征评价指标去获取最终的特征子集。实验结果显示本方法能有效处理高维数据。本章的主要工作有：①提出了一个新的非监督特征选择方法，属于过滤器方法；②给出了一个新的特征排序准则并验证其正确性；③使用并修改了模糊特征评价指标，并将其推广到离散数据。同时提出了一种计算特征权重的方法；④通过实验验证所提出算法的有效性。

3.1 问题分析和已有的工作

当数据的类别已知时，使用监督的特征选择方法，否则应该采用非监督的方法。在许多模式识别和数据挖掘的应用中，数据的类别是未知的，因此非监督特征选择是当前的一个研究热点，也是难点，具体表现为[22]：

1） 没有普遍认同的聚类性能评价准则；

2） 聚类数与特征子集的维数相关。

对于非监督特征选择，目前常用的方法有主成分分析PCA和局部线性嵌入LLE，这些又称为特征抽取，它们存在以下缺点[23]：

1）抽取的特征没有明确的实际含义，难以理解；

2）在特征抽取的过程仍然要用到所有的原始特征。

其它的非监督的特征选择方法，如文献[24]利用特征之间相似性来消除冗余特征，而文献[25, 26, 27]利用似然对数和聚类的判别性来评价不同特征子集的聚类性能。此外利用熵来评价每个特征聚类性能，再完成非监督特征选择[28]。在文献[29]利用遗传算法和C均值聚类实现特征选择。基于不相关特征与相关特征之间是非关联的假设，文献[30]中对标记数据提出了一种特征选择方法。在文献[31]中提出了一个概念“类别效用”，并用于概念聚类的特征选择中。另外最近提出了一种神经-模糊特征选择方法[32]，以及利用期望最大化EM方法来估计特征的重要性[33]。还有利用贝叶斯方法进行非监督特征选择[34]。但是这些方法都至少存在着下列问题之一：只能剔除冗余特征；只能消除不相关特征；对高维数据效果不佳；对高维数据计算开销较大；对噪声数据敏感等。

3.2 算法框架

作者所提出的方法是一个两层过滤系统，如图3.1所示。第一层剔除冗余特征，然后再消除不相关特征。当然两层过滤器的排列顺序需要考虑所采用过滤器时间开支，详细的讨论参见作者以前的论文[35]。

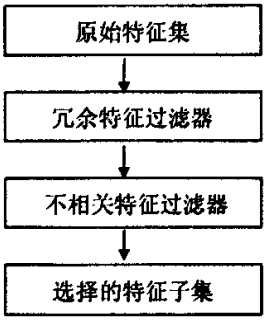


图3.1 非监督特征选择实现框架图

Figure3.1. Diagram of system for unsupervised feature selection

在此系统中，第一层过滤器用来从原始特征集中消除冗余特征，它可能通过任何聚类算法来实现，例如C均值。但是在这里作者采用最近提出的一种算法（Mitra's）[24]，它具有较高的性能和较小的时间开支。首先为每一个特征找到与其最近的 k 个特征，并且从中找到与其第 k 个特征最相似的特征，这也表明该特征与其 k 个近邻特征之间非常紧凑。然后选择出该特征，并抛弃与其相邻的 k 个特征。重复上面的过程直到剩下的所有特征都被考虑到。为了能够确定 k 值，需要设置一个阈值 ε，它的取值为前一次循环中所选择的特征与其第 k 个最近邻的相似度。在以后的循环中，将检查选择的特征与其第 k 个近邻特征之间的相似性，如果小于 ε，将减小 k 的值。因此算法在运行过程中，k 值是不断调整的，具有一定的自适应性。在算法中，因为 k 可以决定 ε，所以 k 控制聚类的程度。实验结果表明 $k+d\approx n$，其中 d 是选择特征子集的大小，n 为原始特征数。如果训练样本数为 N，则 Mitra's 的时间复杂度为 $O(n^2N)$。在原算法的基础上，基于下面的分析做了一些修改：原算法在流程上存在一些缺陷，在每次删除冗余特征后，原算法没有重新对剩余特征的关联性进行排序，重新寻找 r_i^k 最小的特征 F_i，从而使得算法流程不清晰，且存在误差。我们进行了一些修改（在第二步中先将所有特征两两之间的距离（相似性）都计算出来，构成一个矩阵，并增加了第六步），修改后的流程为：

算法 3.1 Mitra's

第一步：初始化，选取 $k\le n-1, FR\leftarrow F$。

第二步：对所有特征，两两之间计算非关联性，构建一个上三角矩阵，其行列都代表特征，

非零元素为特征之间的距离值，对每一个特征 $F_i \in FR$，计算 r_i^k 表示特征 F_i 与其在子集 FR 中第 k 个近邻特征的相似性。

第三步：保留 r_i^k 最小的特征 F_i，并抛弃 F_i 的 k 个最近邻特征，LET $\varepsilon = r_i^k$。

第四步：IF k>cardinality(*FR*)-1: k=cardinality(*FR*)-1 % *cardinality(FR)表示FR的势，也就是维数。*

第五步：IF k=1：GO TO *第九步*

第六步：对每一个特征 $F_i \in FR$，计算 r_i^k，并寻找 r_i^k 最小的特征 F_i

第七步：WHILE $r_i^k > \varepsilon$ DO:

{ k=k-1

$r_i^k = \inf_{F_i \in FR} r_i^k$

IF k=1：GO TO *第九步* }

END WHILE

第八步：GO TO *第三步*

第九步：输出 *FR*

系统中第二个过滤器，将用来消除不相关特征。已有一些算法可以消除不相关特征，如文献[28]中利用封装器方法来剔除不相关特征，时间开支较大。而EM算法用来选择相关特征，但无法处理高维数据[33]。文献[30]中介绍了一种用来选择相关标记特征，此外Relief算法[36]以及扩展[37]可以找到统计相关特征，但它们属于监督的特征选择方法。作者提出一种新的不相关过滤器AIF，并根据系统结构，将Mitra's算法与AIF算法进行组合，获得新得分层过滤方法MAIF，消除冗余特征和不相关特征，进行非监督的高维特征选择。Mitra's算法的输出是AIF的输入。实验证明该算法对高维数据是有效的且能处理噪声数据。由于提出的方法属于过滤器模型，时间复杂度比较低。

3.3 不相关过滤器

为了消除不相关特征，作者将先对特征根据聚类性能进行排序，然后采用独立于特定学习算法的评价准则来获取相关特征。利用指数熵来排序特征，并采用模糊特征评价指标 FFEI(Fuzzy Feature Evaluation Index)[32, 38, 39]作为评价准则从排序后的特征集中选择最终的特征子集。特征排序是一种过滤器方法，如果特征之间是相互独立的，那么排序方法就可以获得最优的特征子集。即使不满足独立性，排序方法在计算开支上也好于其它方法，因为它只需要计算 n 个特征的性能指标并进行排序，同时排序方法可以防止过学习[40]。

3.3.1 排序指标

在选择排序指标时，应当能刻画下面两个基本情况：在特征空间中

1）如果数据集中的样本分布是均匀的，则聚类的不确定性是最大的；

2）若样本分布有着良好的聚类，则相应的聚类不确定性是最小的。

基于以上两点，定义如下的排序指标：假设 $S_{p,q}$ 表示任意两个样本 x_p 和 x_q 的相似性，如果两个样本间很相似，则 $S_{p,q}$ 取值比较大，否则取值较小。N 表示样本数。特征排序指标定义为：

$$H=\sum_{p=1}^{N}\sum_{q=1}^{N}\left(S_{p,q}\times e^{(1-S_{p,q})}+(1-S_{p,q})\times e^{S_{p,q}}\right) \tag{3.1}$$

其中 $S_{p,q}$ 的取值范围是 [0.0-1.0]，当 $S_{p,q}\to 0(1)$，则H的取值将下降，但是，当 $S_{p,q}\to 0.5$ 时，H 的取值将上升。换句话说，若两个样本非常相似或不相似，则它们属于或不属于同一类的可能性就非常大，也就是具有良好的聚类，则使得H的取值很小，反之，若样本均匀分布，则H的取值就很大。因此该排序指标能有效地描述数据集在选定的特征空间下的聚类性能，它属于指数熵。

至于 $S_{p,q}$ 的计算，根据不同情况定义如下：

1) 连续数据，将使用欧氏距离来计算样本相似性：

$$S_{p,q}=e^{-\alpha(dis_{p,q})} \tag{3.2}$$

$$dis_{p,q}=\left[\sum_{k=1}^{n}\left(\frac{x_{pk}-x_{qk}}{\max_k-\min_k}\right)^2\right]^{1/2} \tag{3.3}$$

其中 α 是参数，n是特征维数，$(\max_k-\min_k)$为第k维特征在所有样本中的最大取值与最小取值之差，其用来对样本 x_p 和 x_q 的第k维特征 x_{pk} 和 x_{qk} 的取值之差进行归一化处理。另外在实验中 α 通过公式 $\overline{S}=e^{-\alpha\times\overline{dis}}$ 来自动获取，因为当该式左边等于0.5时，H的取值最大，可得到 $\alpha=-\ln 0.5/\overline{dis}$，其中 $\overline{dis}$ 为训练集中样本间的平均距离，$\overline{S}$ 为样本相似性的平均值。

2) 对于离散数据，采用海明距离。两个样本的相似性度量定义为

$$S_{p,q}=\frac{\sum_{k=1}^{n}|x_{pk}=x_{qk}|}{n} \tag{3.4}$$

其中当 x_{pk} 等于 x_{qk} 时，$|x_{pk}=x_{qk}|$ 等于1，否则为0。

3) 对于同时含有以上两种数据的数据集，可以先将连续数据离散化，再利用公式（3.4）。

3.3.2 排序算法

利用以下的方法来排序特征：轮流删除每一个特征，并计算在剩余特征构成的特征空间中数据集的H值，如果某个特征被删除后，所得的H值最小，则该特征是最不重要的。反之如果所得的H值最大，则该特征是最重要的。因为H值最小，则表明该特征对数据集中样本的分布特性影响最小，也就是聚类性能最差，也就是最不重要的。同理如果H最大，则表明该特征对数据集中样本的分布特性影响最大，也就是聚类性能最好、最重要。根据H值进行排序，就可获得排序后的特征集。排序算法（RANK）的伪代码如下：

算法3.2 RANK

```
RH =n 个特征的H值
FOR k =1,2, ..., n
  RH_k = CalH(F_k)
END
OUTPUT Rank(RH)
```

上述算法中，CalH(F_k)函数是利用前面介绍的方法来计算删除特征F_k后，数据集的H值。Rank函数是对获取的每个特征对应的H值进行排序，从而对相应的特征进行排序。另一方面，模式识别中经常会遇到样本数很多或者特征维数很高的数据集。至于维数高的数据集，后面将详细讨论。对于样本数很多的数据集，可以采用随机采样的方法。经过实验观察，在大部分情况下合理的随机样本能保留原始数据集的聚类信息[23]。需要注意的是，为了能有效地利用H值对特征进行排序，在采样过程中，应该确保数据集的聚类结构不变，且最大限度地与样本数无关。在含有大量样本的数据集中，采用的排序算法(SRANK)的伪代码如下：

算法3.3 SRANK

```
FOR k= 1, 2, ..., n
  特征 F_k 的最终排序 OR_k=0
END
FOR l=1, 2, ..., t   %t 是随机采样数
    选取样本 x_l
    运行 RANK，计算排序 R_l
    FOR k=1, 2, ..., n
      OR_k= OR_k + R_{l_k}
    END
END
```

OUTPUT *OR*

通常情况下，在处理大样本数据集时，35被认为是最少的采样数，即 $t \geq 35$。

3.4 评价准则

对排序后的特征集如何确定要选择特征的数量，也就是从排序的特征集中选择多少个特征的问题，是目前特征选择研究领域的一个难点问题。到目前为止，主要有以下方法（包括监督和非监督）：

1）如果知道所需要的重要特征数，就可以直接从最重要的特征开始选取，直到满足所需数目；

2）利用聚类算法[23]，选择使聚类性能最佳的特征子集，这是一种封装器法，一般计算开销较大，且目前没有普遍接受的聚类性能评价指标；

3）RFE方法[41]，这是一种反向剔除的方法，但不适合本方法选定的排序指标，因为H是随着特征数变化的；

4）过滤器法，一般计算开销较小，如[22]提出了一种特征子集的评价准则，它假设大部分的类间距离将大于类内距离，而这往往是不成立的。

此外还有探针（Probe）法[42]等。作者在这里计划采用过滤器方法，因为其时间开销较小且实现方便，比较适合高维数据。特征子集的评价准则采用模糊特征评价指标FFEI，这种指标已成功与神经网络相结合进行特征选择，获得了较好的性能。作者将它应用于对已排序的特征集进行特征选择，并在使用的过程中将对它进行改进，以推广到离散数据的情况。

3.4.1 特征子集评价准则

假设在 n 维原始特征空间里，第 p 个样本 x_p 和第 q 个样本 x_q 属于同一聚类的隶属度为 μ_{pq}^{O}，而 μ_{pq}^{T} 表示在所选择的 $n'(n' \leq n)$ 维特征子集中二者属于同一聚类的隶属度，N 为样本数，相应的评价准则 FFEI 定义为：

$$FFEI=\frac{2}{N(N-1)}\sum_{p}\sum_{p\neq q}\frac{1}{2}\left[\mu_{pq}^{T}\left(1-\mu_{pq}^{O}\right)+\mu_{pq}^{O}\left(1-\mu_{pq}^{T}\right)\right]$$

$$\mu_{pq}=\begin{cases}1-\dfrac{dis_{pq}}{D};\ \text{if}\ \ \text{dis}_{pq}\leq D\\ 0;\ \text{otherwise}\end{cases} \tag{3.5}$$

其中 dis_{pq} 表示两个样本 x_p 和 x_q 间的距离，也可以用来描述它们的相似性，D 为两个样本属于同一类的临界距离。

1）连续数据

$$D = \beta(dis_{\max})$$

$$dis_{\max} = \left[\sum_k (\max_k - \min_k)^2\right]^{1/2}$$

$$dis_{pq} = \left[\sum_k w_k^2 (x_{pk} - x_{qk})^2\right]^{1/2} \tag{3.6}$$

其中 $\beta \in [0,1]$ 是由用户指定的参数，$(\max_k - \min_k)$、x_{pk} 和 x_{qk} 的意思与前面一样。而 $w_k \in [0,1]$ 表示第 k 维特征的权重。

2）离散数据

D 是用户指定的最少特征数，若 D 个特征取值不同，则两个样本不属于同一类。而

$$dis_{pq} = n - \sum_k |x_{pk} = x_{qk}| \tag{3.7}$$

其中当 x_{pk} 等于 x_{qk} 时，$|x_{pk} = x_{qk}|$ 等于 1，否则为 0。

若两个样本非常相似或不相似，则评价指标的取值将下降，也就是说，如果在选择的特征子空间内，（类间）类内距离（增大）减小，则相应的特征子集的 FFEI 评价指标将降低，因此就是要选择使 FFEI 取最小值的特征子集。

3.4.2 权系数计算

在连续特征的 dis_{pq} 中，计算欧氏距离时的考虑了特征的权重，这样就突破了欧氏距离对球形聚类的假设，可以收到更好的效果。对于特征权重的计算，利用前面得到的每个特征对应的 H 值，提出了一种近似计算特征权重的算法(**CalWeight**)：

算法 3.4 CalWeight

假设特征集 $(RF_1, \cdots, RF_n)$ 中的各个特征已按对应的 H 值进行降序排列，其中 n 为该特征集中的特征个数。设 RH_k 为 RF_k 所对应的 H 值，当然 RH_n 的值最小。

初始化：$ODH=0$，$DH_k=0$，$k=1,\ldots,n-1$

FOR $k=1, 2, \ldots, n-1$

$DH_k = RH_k - RH_n$

$ODH=ODH+DH_k$

END

$DH_n=1$

$ODH=ODH+DH_n$

FOR k=1, 2, ..., n

$w_k = DH_k / ODH$

END

3.4.3 不相关过滤算法 AIF

对于已经排序的原始特征集，采用类似的前向特征选择方法，从最重要的特征开始，逐步加入次重要的特征。这样就使得搜索空间大大缩小。不相关过滤器算法（AIF）的伪代码如下：

算法 3.5 AIF

运行 RANK 获取排序的特征 RF_k和相应的RH_k k=1,...,n

$LET\ fs=\{RF_1\}$

FOR k=1,2, ..., n

$fs' = fs \cup \{RF_k\}$

计算 $FFEI(fs')$

IF $FFEI(fs) - FFEI(fs') > \phi$

$fs = fs'$

CONTINUE

ELSE

OUTPUT fs

BREAK

END

END

如果样本数很多，则运行 SRANK 替换 RANK。而在没有先验知识的情况下，阈值ϕ是很难确定的。但根据前面的分析和后面的实验结果，随着重要特征的不断被选择，FFEI 的值将不断下降，一旦所有的重要特征都被选择了，若继续加入不重要的特征，则其值将上升或者保持相对稳定。因此特征选择就转换为寻找 FFEI 取最小值或者保持相对稳定的位置，而这在 FFEI 的变化曲线中是很容易观察到的，因此特征选择的终止条件可以很方便确定，而不需要事先给定ϕ值。

计算复杂性分析：对于 RANK，当某个特征被删除，则需要计算任意两个样本的距离，时间复杂度为$O(N^2(n-1))$。对每个特征而言，总的时间复杂度为$O(n^2N^2)$。而对特征进行排序的时间为$O(n)$，因此 RANK 算法的时间复杂度为$O(n^2N^2)+O(n)\approx O(n^2N^2)$。对于评价准则 FFEI，对某一个待评价的含有 f（f=1,2,⋯,n）个特征的特征子集，需要计算任意两个样本之间的距离，时间复杂

度为 $O(N^2 f)$，对所有的待评价子集，其总的时间复杂度为 $O(N^2)+O(2N^2)+\cdots+O(nN^2)\approx O(n^2N^2)$。最后算法 AIF 的时间复杂度为 $O(n^2N^2)+O(n^2N^2)=O(n^2N^2)$。通常情况下要选择的特征比较少，因此 f 的值会比较小，从而时间开支比较小。此外，样本之间的距离可以并行计算，这样也可以降低时间消耗。

3.5 实验

作者在多个数据集上验证所提出的特征选择分层过滤器系统和不相关过滤器算法。整个实验包括两个部分：一是测试不相关过滤器 AIF，另外是验证两层过滤器方法 MAIF（Mitra's+AIF）的性能。首先，通过一些基准和合成数据集来验证算法 RANK（SRANK）和 AIF 的正确性，这些数据集的重要特征都是已知。另外，还利用一些现实中的较高维数据集测试 MAIF 算法的性能。使用 MATLAB 中的随机函数来产生合成数据集，其中部分特征是重要的，它们的取值呈高斯分布。而其它不重要特征的取值是均匀分布的。如果没有指明，则每个聚类含有相同的样本数，且类间可以重叠。

3.5.1 数据集

低维的基准和合成数据集：四个合成的低维数据集(Syn2_6, Syn3_11, Syn4_15, Syn6_22)，它们的特征维数和类别数都不相同。基准数据集（包括连续和离散）来源于 UCI[43]。根据一定的先验知识，这些数据集的重要特征是已知的。虽然这些数据集的类别信息也是可以得到的，但在特征选择过程中将不考虑类别信息。Parity3+3 中包含有 3 个关联特征，3 个冗余特征和 6 个不相关特征。这些数据集的详细描述见表 3.1。

表 3.1 低维数据集及其排序结果

Table 3.1. The details of low dimensional data sets and the ranking results

数据集	特征数	类别数	重要特征	排序结果（降序）
Iris	4	3	3,4	{3,4},2,1
Corral	6	2	1,2,3,4	{6,3,1,4,2},5
Monk3	6	2	2,4,5	{5,4,2},1,6,3
Syn2_6	6	2	1,2,3	{1,2,3},6,4,5
Syn3_11	11	3	1-6	{2,3,6,1,4,5},8,11,10,7,9
Syn4_15	15	4	1-5	{1,3,4,2,5},12,9,11,…
Syn6_22	22	6	1-7	{5,6,1,2,7,3,4},9,10,…
Parity3+3	12	2	{1,7}, {2,8}, {3,9}	{9,3,8,2,7,1},4,10…

较高维数据集：一个合成的高维的数据集 Syn5_100，包含有 100 个特征（其中

前 20 个特征是重要的，而其它 80 个是不重要的），5 个类别，每一类呈高斯分布，不重要特征的取值是随机均匀的。每一类含有 20，000 个样本，且整个数据集中含有 5000 个噪声样本。另外从[43]选择三个现实的较高维数据集（Ionosphere, Sonar, Multi-features），它们的相关信息如表 3.2 所示。同样在特征选择过程中，将不考虑样本的类别信息。

表 3.2 高维现实数据集

Table 3.2. The details of high dimensional real-world data sets

数据集	样本数	特征数	类别数
Ionosphere	351	34	2
Sonar	208	61	2
Multi-feature	2000	649	10

3.5.2 AIF 性能

低维数据集的排序结果如表 3.1 中最后一列所示。本章提出的排序方法一般能将重要特征排在前面。只有 Corral 数据中的特征 F_6 的排位过高，但是，有 75%的数据样本的 F_6 与类是关联的[23]，因此对于现实数据集来说，这完全是可以接受的。对于 Parity3+3，排序也是正确的，只是冗余特征无法鉴别。

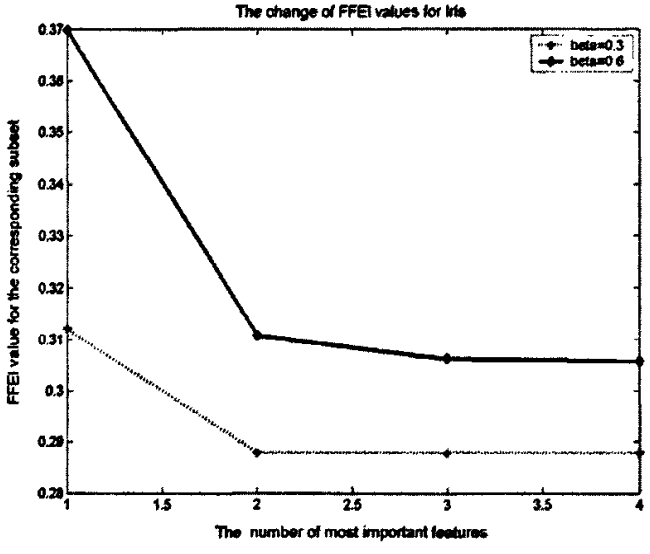
对高维合成数据的排序结果如表 3.3 所示，采样比例分别为 0.25%，0.50% 和 1.0%，这里只列出 5 次采样的排序结果，每次都能将重要特征排在最前面。

表 3.3 合成高维数据集 Syn5_100 的排序结果

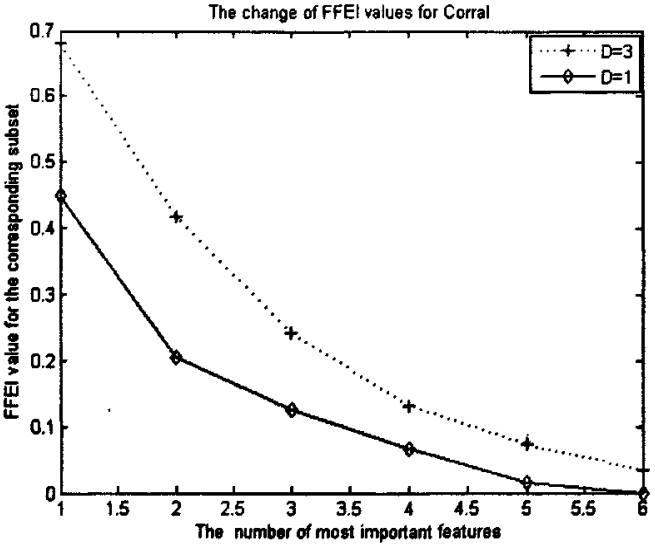
Table 3.3. The ranking results of high dimensional synthetic data set -Syn5-100

#	样本集尺寸		
次数	0.25%	0.50%	1.0%
1	{1,4,11,8,18,9,20,17,10,12,19,7,3,14,16,15,2,5,13,6},27,83,91,...	{12,17,19,11,3,4,9,1,5,16,7,10,20,13,15,6,14,18,8,2},44,43,94,...	{4,1,10,11,17,8,15,2,3,19,18,7,6,14,12,20,16,9,13,5},69,64,52,...
2	{10,1,14,12,7,8,20,3,6,11,9,5,18,17,4,13,19,16,15,2},60,25,99,...	{2,8,6,3,13,4,14,12,16,9,17,1,5,19,7,15,18,20,10,11},92,93,44,...	{16,8,10,9,12,18,20,14,4,5,3,11,2,15,1,17,19,7,13,6},91,65,51,...
3	{16,9,7,8,15,18,11,17,4,6,19,12,1,3,20,13,2,14,10,5},41,92,71,...	{6,2,15,19,4,9,13,3,18,12,14,8,16,7,17,10,20,1,11,5},21,64,48,...	{7,13,4,20,6,10,5,1,12,3,14,15,8,2,17,9,16,18,11,19},82,25,72,...
4	{3,11,2,7,14,10,5,17,6,1,13,20,8,16,19,18,9,15,12,4},80,49,94,...	{16,20,9,17,5,8,13,12,3,19,2,1,18,4,6,10,14,15,7,11},56,81,72,...	{16,8,10,9,12,18,20,14,4,5,3,11,2,15,1,17,19,7,13,6},91,65,51,...
5	{20,11,6,12,3,13,19,7,1,17,8,5,2,18,15,9,14,4,10,16},71,43,59,...	{11,6,3,20,15,5,10,19,2,16,1,8,12,9,4,14,13,18,7,17},41,77,...	{16,8,10,9,12,18,20,14,4,5,3,11,2,15,1,17,19,7,13,6},91,65,51,...

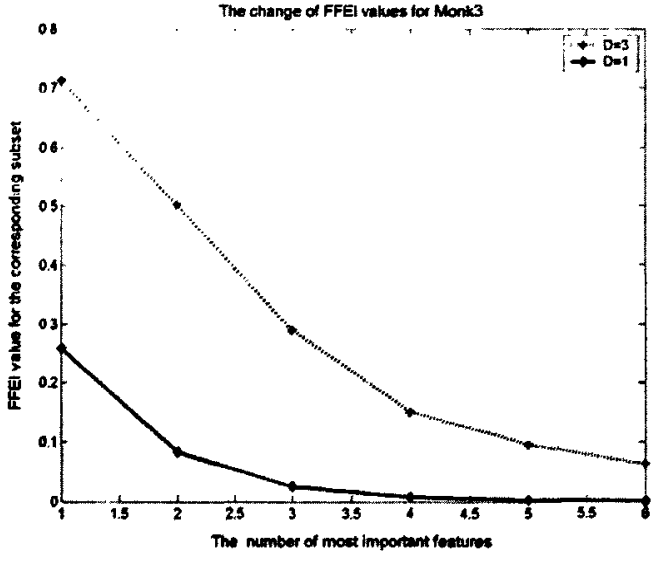
利用算法 AIF 对这些低维数据集进行特征选择，其 FFEI 的变化曲线如图 3.2 所示，分别对应数据集 Iris，Corral，Monk3 和 Syn3_11。每幅图中只显示了两种不同 D 值的实验结果曲线，D 的其它取值具有类似的曲线。图 3.3 描述了数据集 Syn5_100 在 5 次采样中的 FFEI 的变化曲线，分别对应两种不同的采样比例和 D 值。其中 X 轴表示重要特征数，Y 轴表示相应的重要特征构成的特征子集的 FFEI 值。从图中可知，FFEI 值将随着重要特征的加入而快速下降，但是当所有重要特征都已加入后，其值将保持相对稳定或者上升。对于实际应用来说，是很容易发现这种趋势的，因此可以通过寻找曲线中的拐点来选择特征子集。并且对于不同的 D 值，本算法得到的变化曲线很类似，因此对 D 值有一定的鲁棒性。



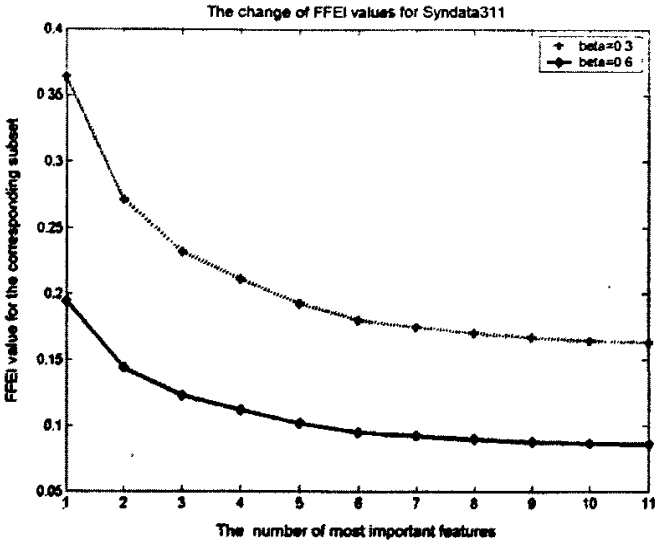
(a)


The change of FFEI values for Corral
D=3
D=1
FFEI value for the corresponding subset
The number of most important features
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

(b)


The change of FFEI values for Monk3
D=3
D=1
FFEI value for the corresponding subset
The number of most important features
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

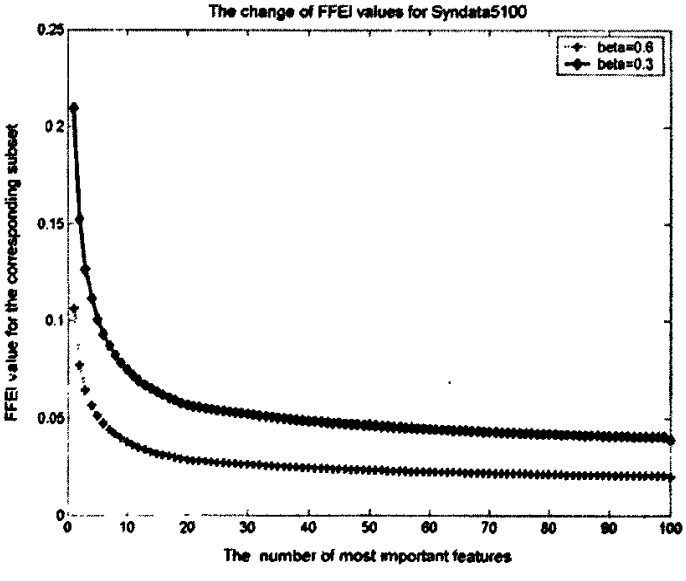
(c)



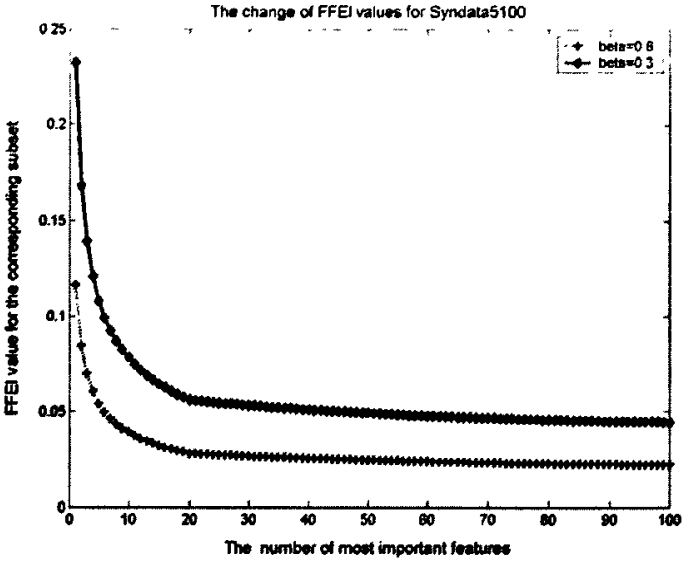
(d)

图 3.2 FFEI 值的变化趋势图 (a)Iris, (b)Corral, (c)Monk3, (d) Syn3_11

Figure 3.2. The change of FFEI values with the addition of the ranked features for (a)Iris, (b)Corral, (c)Monk3, (d) Syn3_11 data set



(a)



(b)

图 3.3 Syn5_100 数据集中 FFEI 值的变化趋势图采样比例为 (a) 0.25%，(b) 1%

Figure 3.3. The change of FFEI values with the addition of the ranked features for Syn5_100 data sets in five runs, the sample percentage is (a) 0.25%, (b) 1%

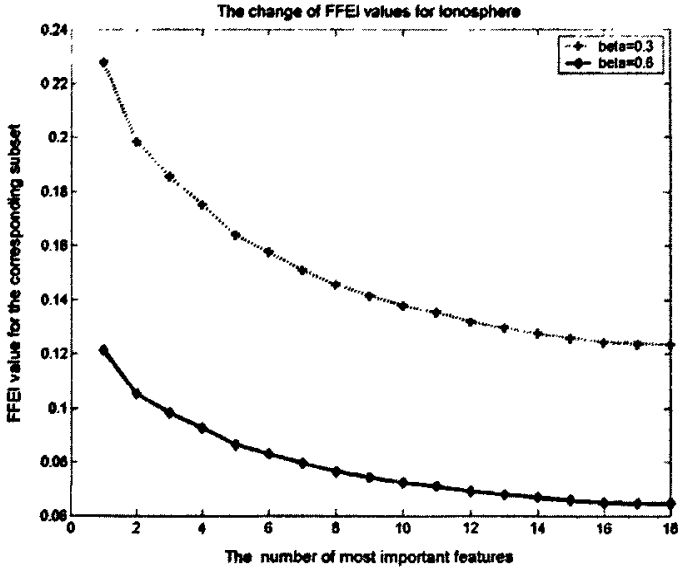
3.5.3 MAIF 性能

对于现实的高维数据集，利用 MAIF 去选择特征子集，并与 Mitra's 算法在分类性能和维数约简上进行比较。Mitra's 算法中 k 值的选择和特征关联性计算公式与文献[24]一样，由于不存在普遍接受的聚类性能评价指标，将利用监督学习中的分类准确率来验证所选择的特征子集的性能。采用 K（K=3）近邻分类器，通过以下的方式来实现交叉验证：首先随机选取 10%的数据作为 Mitra's 算法的训练集；然后合理选取适当的数据作为 SRANK 的训练集，对 Mitra's 算法的输出特征集进行特征排序；再从剩下的数据中选择部分数据，作为测试集，计算不同特征子集的 FFEI 值，选择特征子集；最后通过 K 近邻分类器对余下的数据（验证集）进行分类来验证，运行十次，求平均得到最后的分类准确率。所有结果都显示在表 3.4 和图 3.4。从图中很容易确定 MAIF 的终止条件，从而得到最后的特征子集。实验结果也表明本算法能获取更高维数约简率，且不以牺牲分类性能为代价，对某些数据集，还可以提高分类准确率。虽然对于 Multi-feature 来说，刚开始时 FFEI 的取值有些波动，这在实际应用中是完全可以接受的。当然 Mitra's 算法的运算时间开支肯定小于 MAIF。但是由于特征选择可以脱机处理，理应更关注所选择特征子集的性能。

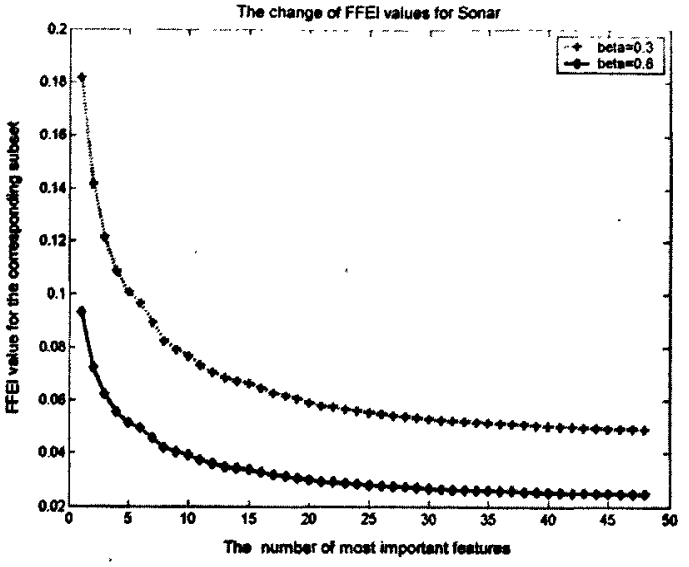
表 3.4 Mitra's 和 MAIF 的分类准确率和维数约简率（OFS 表示利用原始特征集进行分类）

Table 3.4 The classification accuracy and Dimensionality Reduction Rate (DRR) of Mitra's and MAIF for three real-world data sets (OFS: Original Feature Set)

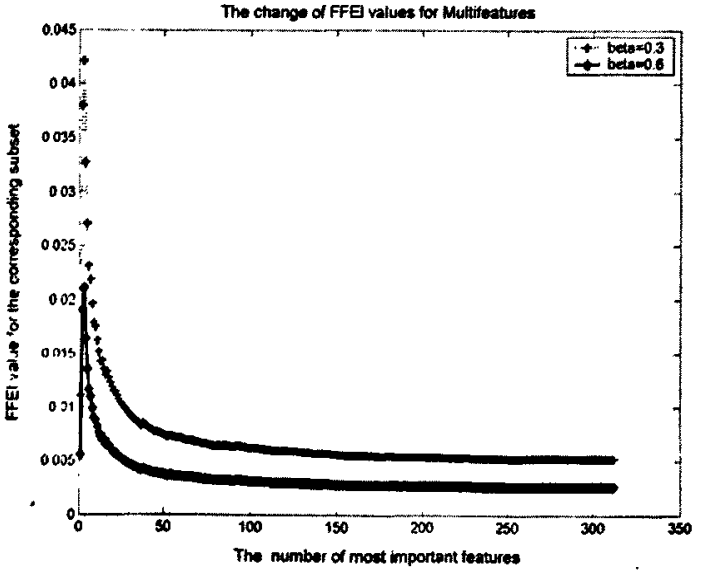
数据集	算法	准确率% (k=3)		维数约简率%
		均值	均方差	
Ionosphere	Mitra's	84.27	0.30	52.9
	MAIF	**84.87**	0.30	**70.6**
	OFS	83.94	0.31	0
Sonar	Mitra's	75.92	0.30	40.3
	MAIF	77.43	0.30	**50.1**
	OFS	**78.62**	0.30	0
Multi-feature	Mitra's	**93.43**	0.20	49.9
	MAIF	92.39	0.22	**77.0**
	OFS	93.35	0.21	0



(a)



(b)



(c)

图 3.4 FFEI 值的变化趋势图 (a) Ionosphere，(b) Sonar，(c) Multi-feature

Figure 3.4. The change of FFEI values with the addition of the ranked features for (a)Ionosphere, (b) Sonar, (c) Multi-feature

3.6 本章小结

本章提出了一种两层非监督过滤器方法 MAIF，其时间开支较小，能有效消除冗余和不相关特征。其中冗余特征过滤器是 Mitra's 算法，而提出了一种新的不相关过滤器 AIF 算法，实验结果表明这种分层方法能获得很好的性能，且能有效处理高维和噪声数据。

在 AIF 算法中，利用指数熵对特征进行排序，同时利用随机采样处理大样本数据集，而根据特征的聚类性能对特征进行排序，在多次随机采样中能获得高度一致的结果。采用过滤器方法对排序后的特征集进行特征选择，评价准则是模糊特征评价指标 FFEI，同时将其扩展到离散特征。在计算评价指标时，利用特征的聚类性能，提出了一种新的计算特征权重的方法，取得很好的效果。并使得特征子集评价指标对不同的临界值 D 具有一定的鲁棒性，这极大方便了用户的使用。

4 基于主成分分析的特征选择算法

特征维数约简主要有两种方法：特征选择和特征抽取。而目前大部分的研究都将二者独立开来。本章以特征抽取的典型方法主成分分析 PCA 为例，将主成分分析与特征选择相结合进行研究。在文中首先分析了基于主成分分析的特征选择算法框架，并且提出了一种基于K近邻规则的新方法，同时提出了多个新的度量准则来计算PCA变换矩阵中行成分相似性。在大量基准数据集上的实验和基于人脸图像的性别识别结果表明，所提算法具有较低的时间开支，且能获得很好的性能。

4.1 问题分析和已有的工作

在许多现实问题中，如人脸识别、文本分类、图像检索等，维数约简是一个不可缺少的步骤[13]。而特征抽取和特征选择是两种常用维数约简方法。特征选择是指从原始空间中挑选特征，而特征抽取是指进行特征空间的变换，将原始特征空间映射到低维空间，正如第三章所介绍的，低维空间中的特征没有实际意义，很难理解。常用的特征抽取方法有主成分分析 PCA，独立成分分析 ICA 和 Fisher 判别分析 LDA 等[44]。

主成分分析 PCA 是一种常用特征提取方法，它将原始特征通过线性变换映射到新的低维特征空间，而获得的主成分可以看着是所有原始特征的线性组合[45]。PCA 有很多优点，已经成功应用于许多领域，如人脸识别和信号处理等。此外由于特征选择是直接寻找有实际意义的特征，且能减少计算开支，因此作者考虑将 PCA 与特征选择相结合，利用它们各自的长处去进行维数约简，也就是研究基于 PCA 的特征选择方法，试图通过找到与主成分相关的关键特征或者删除冗余、不相关以及没有意义的特征将主成分又重新映射到原始空间，来理解主成分的实际意义。

已经存在的基于 PCA 的特征选择方法可以分为两类：

一类是采用一致性评价准则，比较所选择的特征子集与原始特征全集在 PCA 变换中的性能差异。此类方法主要包括主变量方法[46]、数据结构保存算法 DSP[47]以及最少均方估计方法 LSE[45]。其中主变量方法主要是寻找与特征全集具有类似性能的子集，这些性能包括在 PCA 子空间中能最大化样本的分布间隔、保留样本在原始空间中的差异性以及最小化样本在主成分上的投影与在原始空间中的投影之间的均方差。而 DSP 算法的主要特性是选择与特征全集有着相似数据结构的子集。LSE 算法主要是选择使得样本在主成分上的投影与全集一致的特征子集，此外如果搜索策略

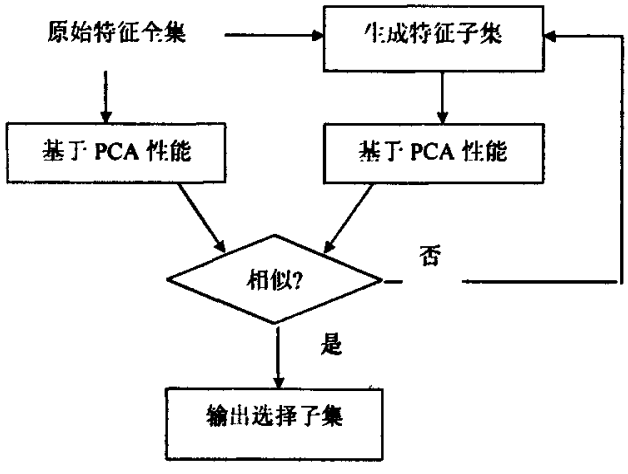
采用前向搜索时（Forward-LSE），寻找关键特征就包括两个主要步骤：第一步是利用PCA 得到在全集下的主成分，以及样本在主成分上的投影；第二步是采用前向顺序选择找到与全集具有相似性能的特征子集。LSE 算法与 DSP 算法相类似，但简化了评价过程，并具有较低的时间复杂度。但是总体来说 DSP 和 LSE 算法的时间开支都很大，不太适合高维数据（大样本和高维特征）。

另一类基于 PCA 的特征选择方法是对 PCA 变换得到的主成分进行分析。其中 B2 和 B4 是最有名的方法[48, 49]。先对主成分按照对应的特征值进行降序排序，B2 抛弃与最后几个主成分最相关的特征，而 B4 是选择与前几个主成分最相关的特征。由于独立地评价各个特征的意义，B2 和 B4 不能消除冗余特征。此外 PFA 算法[50]通过分析 PCA 变换矩阵的元素来获得最优特征子集，但是与 B2 和 B4 不同，PFA 算法考虑了所有的主成分，因此可以有效地剔除冗余特征。

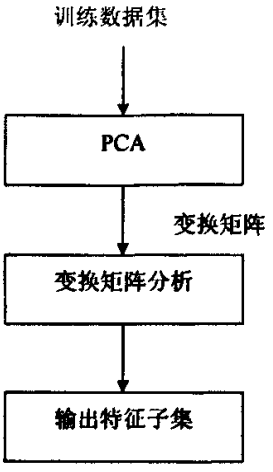
不过以上所提到的各种方法至少存在以下弱点之一：性能较差、计算开支较大、保留了冗余信息以及无法处理高维数据等。本章提出了一种有效的新方法来实现基于 PCA 的特征选择，它属于前面提到的第二类方法。

4.2 基于 PCA 的特征选择算法框架

在 4.1 节中已经简单介绍了两类基于 PCA 的特征选择方法，它们的算法框架分别如图 4.1 所示：



(a)



(b)

图 4.2 基于 PCA 特征选择方法示意图 (a) 第一类，(b) 第二类

Figure 4.2. Diagram of feature selection based on PCA (a) first, (b) second

第一类方法是先使用一些搜索策略（如前向选择，反向删除等）产生候选的特征子集并与全集在 PCA 变换的某些性能上进行比较，其中包括最大化 PCA 子空间上样本的散布、保留分散性、最小化均方差、数据结构相似性以及相同的数据投影能力等。

第二类方法是在训练样本集中先进行 PCA 变换，然后直接分析选取的主成分，得到关键的原始特征。

第一种方法非常吸引人，因为它是基于 PCA 的一些特性，但是由于一般要使用组合特征搜索策略，使得时间开支很大，不适合高维数据。而第二种方法非常直观且计算开支较少，非常适用于高维特征选择。

4.3 算法设计

4.3.1 PCA

假设一个给定的训练数据集含有 N 个样本 $\{x_i\}_{i=1}^{N}$，每个样本由 n 维特征矢量描述 $x_i=[x_{i1},x_{i2},\cdots,x_{in}]^T\in R^n$。训练集的协方差矩阵 Σ 定义为：

$$\Sigma=\sum_{i=1}^{N}(x_i-m)(x_i-m)^T \tag{4.1}$$

其中 m 为所有样本的均值，特征矢量 f_i 以及特征矩阵 F 可以表示为：

$$\begin{aligned} f_i&=[x_{1i},x_{2i},\cdots,x_{Ni}]\\ F&=\{f_1,f_2,\cdots,f_n\}^T \end{aligned} \tag{4.2}$$

在 PCA 中，特征矢量根据特征值进行排序，而特征值表示样本在特征矢量上的分布方差。假设 PCA 子空间由 d 个最大的特征矢量构成 $Q=\{q_1,q_2,\cdots,q_d\}$，它们是基于最小重构误差来上重构样本。这样 PCA 的线性变换可以用变换矩阵 Q 来表示，模式 x_i 在新的子空间就表示为：

$$y_i = Q^T x_i \tag{4.3}$$

其中 $y_i=\{y_{1i},y_{2i},\cdots,y_{di}\}^T$，并且

$$Q=\{q_1,q_2,\cdots,q_d\},\ q_j=\{q_{1j},q_{2j},\cdots,q_{nj}\}^T,\ j=1,2,\cdots,d \tag{4.4}$$

其中 n 是原始特征数，$d\le n$ 并且 $d<<n$。采用这种线性变换通常来说有十种最优特性，其中最重要的特性是使得样本在低维空间中尽量分散、保留样本在原始空间中的差异性和在低维空间中的投影数据与原始数据之间的均方差最小。

4.3.2 特征选择

考虑样本 x_i 在第 j 个主成分上的投影：

$$y_{ji} = q_j^T x_i = \sum_{l=1}^{n} q_{lj} x_{li} \tag{4.5}$$

如上式所示，样本在主成分上的投影是所有原始特征的线性组合。但是有些特征可能是冗余的、不相关的以及没有意义的，因此需要进行特征选择。这也表明可以通过寻找那些在数据投影中起重要作用的关键特征来实现基于 PCA 的特征选择。

特征 $f_i=[x_{1i},x_{2i},\cdots,x_{Ni}]$ 的意义可以通过变换矩阵中与其相应的参数 q_{ij} 来评价，也就是我们可以根据变换矩阵中的元素来确定与主成分关系密切的关键原始特征。另一方面，变换矩阵可以表示为：

$$Q=\{v_1,v_2,\cdots,v_n\}^T, v_i=\{v_{i1},v_{i2},\cdots,v_{id}\}=\{q_{i1},q_{i2},\cdots,q_{id}\}, i=1,2,\cdots,n \tag{4.6}$$

这里矢量 v_i 被称为行成分，它表示第 i 个原始特征 f_i 在低维 PCA 子空间上的投影，也就是说 v_i 中的 d 个元素为 f_i 在 PCA 子空间的各个坐标（主成分）上的投影权重[51]，原始特征与行成分是一一对应的。我们发现如果原始特征关联性很强，那么它们在子空间上的投影权重也会非常相近，也就是具有类似的行成分，并且投影权重的符号没有统计意义[51]。在极端情况下，对于两个相互独立的特征，它们的投影权重将极大不同；而两个完全关联的特征，它们有着相同的投影权重（不考虑符号因素）。基于以上的观察，我们可以通过选择行成分，与所选择的行成分对应的原始特征就是最终所选择的特征子集。

为了寻找特征子集，PFA 方法是利用行成分的结构特性去寻找一些子集，通过聚类使得子集中的行成分高度关联，然后从每一个子集中选出一个代表性的行成分。所

选出的行成分能很好地代表其所在子集中的所有行成分。而与代表性行成分对应的原始特征就是选择出来的特征。所选出的特征数量与聚类（子集）的数目一致。PFA 算法的简单流程如下：

a) 利用 PCA 获得变换矩阵 Q;

b) 利用 C 均值聚类将行成分 $v_1, v_2, \cdots, v_n \in \Re^d$ 聚成 $p \geq d$ 个类，距离度量为欧氏距离；

c) 从每个聚类中找到与类中心最近的行成分 v_i；

d) 相应的原始特征 f_i 就是关键特征，最终将选出 p 个关键特征。

基于前面对变换矩阵和行成分的特性分析，作者将提出一种基于 K 近邻规则的聚类新方法[52]来进行基于 PCA 的特征选择。首先为每一个行成分找到与其最近的 k 个行成分，并且从中找到与其第 k 个行成分最相似的行成分，这也表明该行成分与其 k 个近邻行成分之间非常紧凑。然后选择出该行成分，并抛弃掉与其相邻的 k 个行成分。重复上面的过程直到剩下的所有行成分都被考虑到。为了能够确定 k 值，需要设置一个阈值 τ，它的取值为前一次循环中所选择的行成分与其第 k 个最近邻的相似度。在以后的循环中，将检查选择的行成分与其第 k 个近邻行成分之间的相似性，如果小于 τ，将减少 k 的值。因此算法在运行过程中，k 值是不断调整的，具有一定的自适应性。算法的具体步骤如下：

算法4.1 基于K近邻规则和PCA的行成分选择

第一步：进行 PCA 变换获得变换矩阵 $Q = \{v_1, v_2, \cdots, v_n\}^T$，其中行成分的个数为 n，R 是约减的行成分子集，s_i^k 表示行成分 v_i 与其在子集 R 中第 k 个近邻的相似性，选取 $k \leq n-1, R \leftarrow Q$。

第二步：对所有行成分，两两之间计算相似性，构建矩阵 M，其元素 $M_{ii(i\neq i)} = sim(v_i, v_i)$ 表示两个行成分之间的相似性，并且 $M_{ii} = 0$，M 是一个对称矩阵。

第三步：对每一个行成分 $v_i \in R$，从 M 中获取 s_i^k。

第四步：保留 s_i^k 最大的行成分 v_i，并抛弃 v_i 的 k 个最近邻行成分，以及在 M 中对应的行和列，LET $\tau = s_i^k$。

第五步：IF k+1 大于 R 的尺寸：k = sizeof(R)-1。% sizeof 表示 R 的尺寸。

第六步：IF k=1：GO TO *第十步*

第七步：对每一个行成分 $v_i \in R$，从 M 中得到计算 s_i^k，并寻找 s_i^k 最大的行成分 v_i

第八步：WHILE $r_i^k < \tau$ DO:

{ k=k-1

$s_i^k = \sup_{v_{ii} \in R} s_i^k$

IF k=1：GO TO *第十步* }

END WHILE

第九步：GO TO *第四步*

第十步：输出 R

与 R 中行成分相对应的原始特征就是特征选择的结果。

4.3.3 行成分相似性度量

在算法 4.1 中需要解决的关键问题是如何计算行成分之间的相似性。常用的计算变量之间相似性的方法有关联系数、最少回归方差以及最大信息压缩（MICI）等[52]。当然它们也可以用来计算行成分的相似性。Mitra 的分析和实验已经表明 MICI 优于其它方法。现简单介绍如下：

MICI：假设 Σ_{12} 表示变量 x_1 和 x_2 的协方差矩阵，而将 MICI 定义为 $\lambda_2(x_1,x_2)$ 等于 Σ_{12} 的最小特征值。

$$2\lambda_2(x_1,x_2)=\left(D(x_1)+D(x_2)-\sqrt{(D(x_1)+D(x_2))^2-4D(x_1)D(x_2)(1-\rho^2_{x_1,x_2})}\right) \tag{4.7}$$

其中 $D(\ )$ 表示变量的方差，ρ 表示关联系数。

但是基于 K 近邻规则的分类，理想的情况其距离度量应该适应于特定的问题[53]。当然对于基于 K 近邻规则的聚类，其相似性度量也应该充分考虑所要处理问题的特性。在变换矩阵 Q 中，可以发现行成分的各个元素 $v_{ij}(i=1,2,\cdots,n\quad j=1,2,\cdots,d)$ 的取值范围是[-1, 1]，由于符号没有统计意义，并且特征矢量是相互独立的。因此在充分考虑这些特性的基础上，我们提出了三种新方法来计算行成分之间的相似性，并命名为成分相似指标 CSI，定义如下：

定义： 对于两个行成分 $v_i=\{v_{i1},v_{i2},\cdots,v_{id}\}$，$v_t=\{v_{t1},v_{t2},\cdots,v_{td}\}$，$v_{ij},v_{tj}\in[-1,1]$，$i,t=1,2,\cdots,n$ $j=1,2,\cdots,d$，其相似性计算公式如下：

$$\text{CSI1: } sim(v_i,v_t)=1-\max\left(\left|v_{ij}\right|-\left|v_{tj}\right|\right) \tag{4.8}$$

$$\text{CSI2: } sim(v_i,v_t)=\left(\sum_{j=1}^{d}\min\left(\left|v_{ij}\right|,\left|v_{tj}\right|\right)\right)\Big/\left(\frac{1}{2}\sum_{j=1}^{d}\left(\left|v_{ij}\right|+\left|v_{tj}\right|\right)\right) \tag{4.9}$$

$$\text{CSI3: } sim(v_i,v_t)=\left(\sum_{j=1}^{d}\min\left(\left|v_{ij}\right|,\left|v_{tj}\right|\right)\right)\Big/\left(\sum_{j=1}^{d}\sqrt{\left|v_{ij}\right|*\left|v_{tj}\right|}\right) \tag{4.10}$$

它们的属性总结如下：

1）$0\le sim(v_i,v_t)\le 1$

2）$sim(v_i,v_t)=1$，当且仅当 v_i 等于 v_t

3）$sim(v_i,v_t)=sim(v_t,v_i)$

CSI1 是基于 Chebychev 距离，同时以上的定义也可以看着是最小最大学习规则的应用，这种规则是集成学习和分类器组合的常用方法[54, 55, 56]，并且这种规则也应用到 EEG 信息的分类[57]和性别识别[58]等。

作者将利用相似性度量方法 CSIs (s=1, 2, 3)和 MICI 所构成的基于 PCA 的特征选择算法分别命名为 PCSIs (s=1, 2, 3) 和 PMICI。

4.3.4 算法分析

算法 PCSIs (s=1, 2, 3) 的相关特性可以总结如下：

1）CSIs (s=1, 2, 3) 非常简单并且没有参数需要估计。由于特征矢量的数量 $d \le n$，且通常 $d << n$，因此如果使用复杂的模型去计算行成分的相似性，在主成分数量较少时，参数很难估计。

2）CSIs (s=1, 2, 3) 充分考虑了变换矩阵中元素的取值范围以及特征矢量的独立性，因此所提算法 PCSIs (s=1, 2, 3) 对基于 PCA 的特征选择来说，是一种专业算法，能获得很好的效果。

3）在 PCSIs (s=1, 2, 3) 中，因为 k 决定阈值 τ，所以 k 控制所选择的行成分子集 R 的大小，也就控制了所选择特征子集的大小，这样可以利用 k 来多尺度表示关键特征子集。

4）基于 K 近邻规则的聚类，是一种动态的，部分的以及非分层的聚类。

计算复杂性分析：算法 PCSIs (s=1, 2, 3)，PMICI 和 PFA 都是在进行 PCA 变换以后进行特征选择，因此它们的计算开支差异在于如何对变换矩阵进行处理。就行成分的数量 n 来说，算法 PCSIs (s=1, 2, 3) 的复杂度为 $O(n^2)$，其它的基于搜索的方法，如前向选择、反向删除、顺序浮动搜索、正交前向选择和反向删除[45, 59]，在它们之中只有前向选择和反向删除的时间复杂度为 $O(n^2)$，其它搜索策略的复杂度都高于二次方。另一方面，在计算行成分的相似性时，PMICI 和 PCSIs (s=1, 2, 3) 的复杂度较低。如果特征矢量的个数为 d，那么计算行成分相似性的复杂度为 $O(d)$，因此 PMICI 和 PCSIs (s=1, 2, 3) 的时间复杂度 $O(n^2 d)$。

4.4 实验结果

在缺少先验知识，无法预先知道数据集中各个特征的重要性，为了验证所提出算法的性能，作者是通过与其它方法在分类性能以及时间开支上进行对比实验而不是列出具体选择的特征。

整个实验包括两个部分：首先在基准数据集上比较 PCSIs (s=1, 2, 3)、PMICI、PFA、Mitra's 和 Forward-LSE 的性能和时间开支；另外在性别数据集上比较 PCSIs (s=1, 2, 3)、PMICI、Mitra's 和 PFA 的性别分类性能和时间开支。由于 Forward-LSE 在高维数据集上时间开支较大，不太适用于性别分类。所有算法都是用 MatLab 编译器实现，并运行在 P4 2.8G PC 机上。此外还在基准数据集上分析了 k 值与 R 的尺寸的关系。

4.4.1 数据集

基准数据集来自 UCI 机器学习知识库[43]，详细的描述见表 4.1。

Table 4.1. 基准数据集描述
Table 4.1. Description of benchmark data sets

数据集	样本大小 (测试样本)	特征维数	类别数
Sonar	208 (10 交叉验证)	60	2
Spectf	535 (267)	44	2
Waveform	5000 (10 交叉验证)	21	3

对于性别分类问题，用于训练的数据集包括 786 张男性脸部图像和 1269 张女性图像，特征维数是 1584 个 gabor 小波过滤器[58]。

表 4.2 性别分类的训练集描述
Table 4.2. Description of gallery set for gender classification

性别	样本数
男性	786
女性	1 269

测试集包括 15 类性别数据，它们分别对应不同的姿态、表情、背景以及遮挡物。具体的描述参见表 4.3 以及图 4.3。相关数据集已经被[58]使用过。

表 4.3 性别分类的测试集描述
Table 4.3. Description of probe sets for gender classification

编号	人脸描述	样本数
1	正面 1	1278
2	正面 2	1066
3	向下 10 度	820
4	向下 20 度	819
5	向下 30 度	816
6	笑	805
7	张嘴	815
8	闭眼	805
9	正面戴眼镜	813
10	向右 10 度	814
11	向右 20 度	815
12	向右 30 度	805
13	向上 10 度	819
14	向上 20 度	816
15	向上 30 度	816

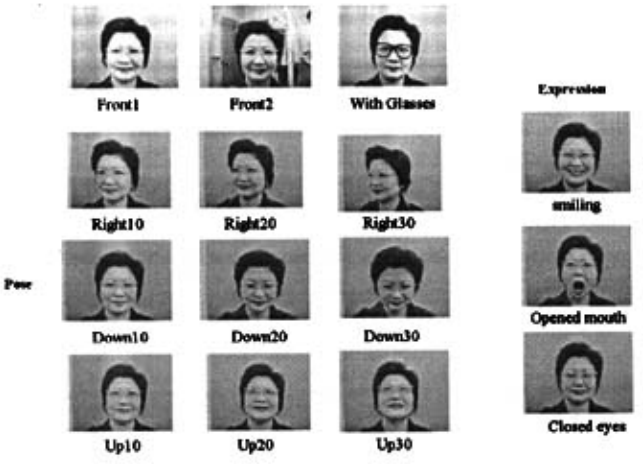


图 4.2 人脸样例图像

Figure 4.2. The examples from probe sets

4.4.2 基准数据集实验结果

对于基准数据集，作者使用 K 近邻分类器（K=3）的分类准确率去评价所选择的特征子集，采用 10 次交叉验证来获取训练集和测试集。但对于 Spectf 数据集，直接使用原数据集中已经划分好的训练样本和测试样本。在进行 PCA 变换时，需要事先确定特征矢量的个数，也就是 d 的取值。在实验中，将特征值进行降序排序，然后从头开始选取特征值，使得特征值之和占到所有特征值之和的 90%，并选取它们对应的特征向量，构成变换矩阵 Q。对于 PFA 算法来说，生成的聚类数等于 d，也就是 $p=d$。实验结果如表 4.4 所示。

表 4.4 基准数据集的实验结果

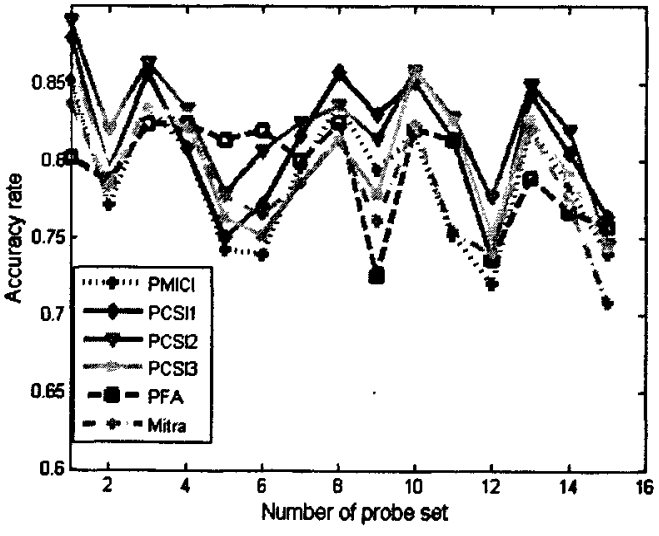
Table 4.4. The experimental results for Sonar, Spectf and Waveform

数据集	算法	KNN 准确率 (K=3)		时间
		均值%	方差	
Sonar	PMICI(k=20)	60.34	0.22	0.9844
	Forward-LSE	61.99	0.22	7.7969
	PFA	62.17	0.20	0.1094
	Mitra's(k=20)	58.74	0.23	1.1406
	PCSI1(k=30)	61.48	0.23	0.7656
	PCSI2(k=30)	62.62	0.22	0.7344
	PCSI3(k=30)	**63.18**	0.21	0.7344

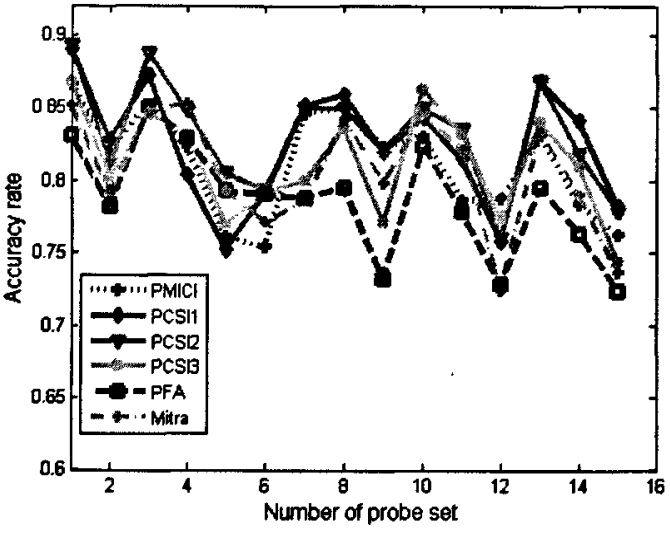
Spectf	PMICI(k=33)	73.98	0.19	0.5781
	Forward-LSE	77.70	0.17	3.1563
	PFA	79.93	0.16	0.1406
	Mitra's(k=33)	75.46	0.19	0.6875
	PCSI1(k=30)	79.18	0.17	0.4219
	PCSI2(k=30)	79.84	0.18	0.4688
	PCSI3(k=30)	**80.84**	0.15	0.4063
Waveform	PMICI(k=6)	77.14	0.18	0.2344
	Forward-LSE	77.62	0.19	28.8438
	PFA	75.29	0.19	0.0781
	Mitra's(k=6)	77.46	0.22	0.8594
	PCSI1(k=8)	**80.90**	0.16	0.2031
	PCSI2(k=8)	78.67	0.18	0.2188
	PCSI3(k=8)	78.38	0.18	0.1875

4.4.3 性别分类

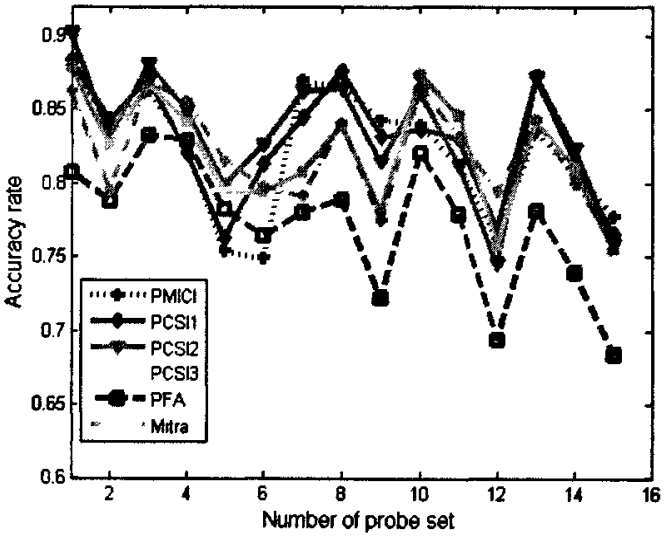
在本节中，将所设计的算法应用到性别识别上，并与 PFA，PMICI 以及 Mitra's 算法进行比较。分类器采用传统的支持向量机 SVM[60]，参数 C 设为 1。特征矢量个数的选取与前面一致。实验结果如图 4.4 所示。



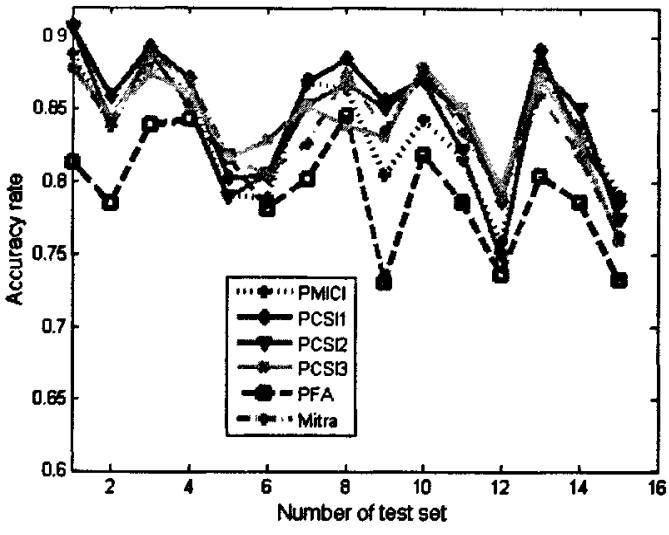
(a)


Accuracy rate
Number of probe set
PMICI
PCSI1
PCSI2
PCSI3
PFA
Mitra

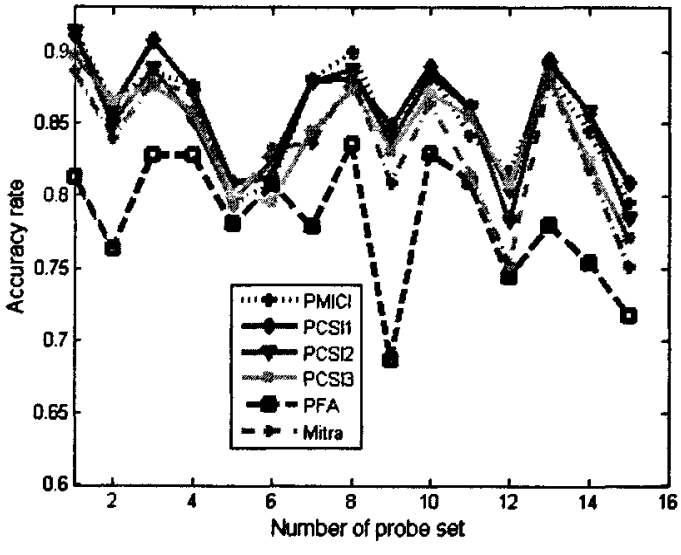
(b)


Accuracy rate
Number of probe set
PMICI
PCSI1
PCSI2
PCSI3
PFA
Mitra

(c)



(d)



(e)

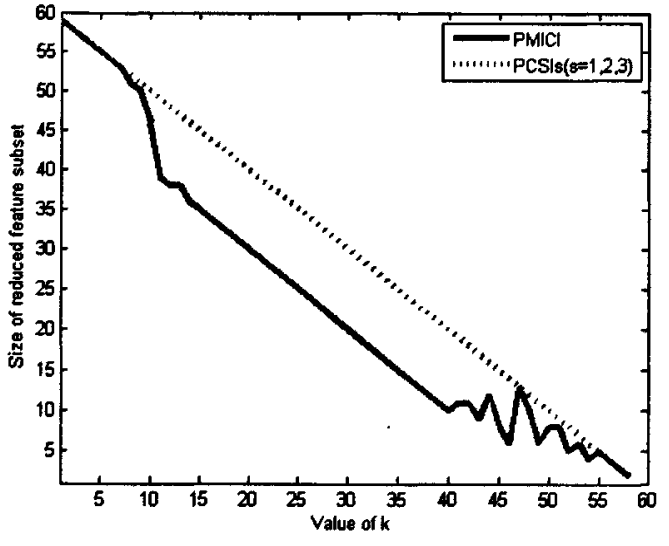
图 4.3 选择不同数量特征的性别分类结果

Figure 4.3. Experimental results of gender classification for different number of selected feature: (a) 484, (b) 584, (c) 684, (d) 884 and (e) 1084

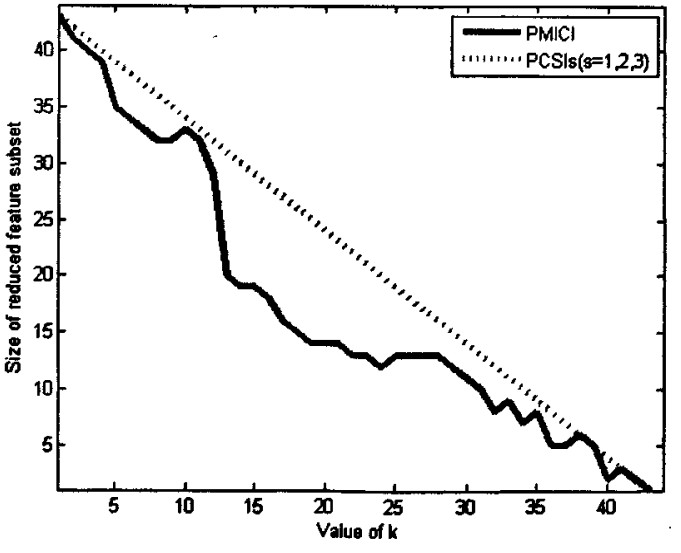
4.4.4 参数 k 的特性

对于算法 PCSIs (s=1, 2, 3) 和 PMICI，作者在三个基准数据集上进行实验，以探

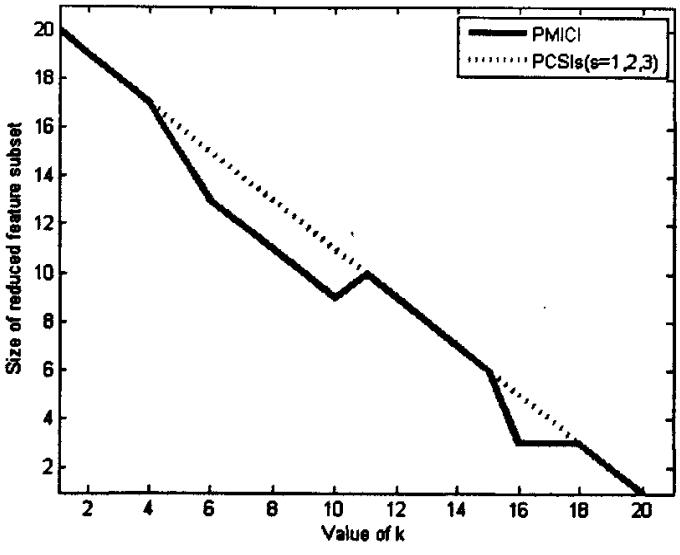
索参数 k 的值与所选择的特征子集维数的关系。与预料的一样，所选择的特征子集维数随着 k 值的增大而减少。这也表明 k 可以控制子集的大小。实验结果同时表明，在算法 PCSIs (s=1, 2, 3)中，k 值与特征子集大小之和严格等于原始的特征维数。但是对于 PMICI，k 值与特征子集大小之和是近似等于原始特征维数，没有严格的等式。这样如果用户事先知道要选择的特征个数，那么对于算法 PCSIs (s=1, 2, 3)来说，就很容易确定 k 值。实验结果如图 4.4 所示。



(a)



(b)



(c)

图 4.5 k 值与选择特征数的关系图(a) Sonar, (b) Spectf, (c) Waveform

Figure 4.5 The relation of k and the size of reduced feature subset R for (a) Sonar, (b) Spectf, (c) Waveform

4.4.5 观察评论

1）PCSIs (s=1, 2, 3) 能获得很好的性能，这也表明所提出的相似性度量方法相对于 MICI 来说，能够更好地捕捉到变换矩阵中元素的特性，且能有效计算行成分之间的相似性。

2）PCSIs (s=1, 2, 3) 的时间开支小于 PMICI，Mitra's 和 Forward-LSE。对于基准数据集，PCSIs (s=1, 2, 3) 中至少有一个算法获得比 PMICI、PFA、Mitra's 和 Forward-LSE 更高的分类准确率。对于性别分类，在不同维数的特征子集上，PCSIs (s=1, 2, 3) 中至少有一个算法在大部分的测试集上获得比 PMICI、PFA 和 Mitra's 更高的性能。

3）PFA 算法的时间开支是最小的。但是在性别分类中 PCSIs (s=1, 2, 3) 和 PMICI 的性能好于 PFA。且当选择的特征维数增加时，PFA 算法的性能没有提高甚至变坏，但 PCSIs (s=1, 2, 3) 和 PMICI 的分类准确率在提高。这个事实表明所提出的方法更适合于高维数据。此外特征选择可以脱机处理，我们应该更关注所选择特征的性能。

4）从图 4.5 和表 4.5 中可以得到如下结论：对于 PCSIs (s=1, 2, 3)来说，其 k

值与所选择的特征个数之和严格等于原始特征数。这一特性可以方便用户选定 k 值，而这一性质并没有在 PMICI 算法中得到严格的体现。

4.5 本章小节

基于 PCA 的特征选择是一种有效的、具有实际意义的维数约简方法。在本章中，作者提出了一类新方法，它都是基于 K 近邻原理的，只是采用了不同的相似性度量方法。这些相似性度量方法能有效利用变换矩阵中元素的特性以及特征矢量的相互独立性。除了理论上的分析，还在基准数据集和性别分类上比较了所提出的算法 PCSIs (s=1, 2, 3)与其它方法的性能。这些方法包括基于 PCA 的特征选择方法 PMICI、PFA、和 Forward-LSE 以及其它基于 K 近邻原理的特征选择方法 Mitra's。实验结果表明 PCSIs (s=1, 2, 3) 能够获得较高性能且时间开支较小。此外 k 值和特征子集的大小之和等于原始特征空间的维数。

5 基于 K 近邻分类损失-间隔的特征选择算法

在前面的章节中，已经说明特征选择算法主要包括两类：封装器和过滤器。封装器对特定的分类器效果较好，但开支较大。而过滤器的适用面比较广，开支较小但效果一般。为了有效地结合封装器和过滤器的优点，本章提出了一种针对 K 近邻分类器的特征选择算法，它是基于 K 近邻分类损失函数和分类间隔（Margin），众所周知，大的分类间隔能保证分类器具有很好的推广能力。

5.1 引言

分类间隔是目前机器学习领域的一个研究热点，它经常用来描述分类器性能的置信度，并且它已经成为算法设计的理论基础和依据。目前由于对支持向量机（SVM）的分类间隔研究比较深入和透彻，因此基于SVM分类间隔的特征选择方法也比较多，如SVM-RFE [41]，R^2W^2 [62]等。而K近邻分类器[63]是一种最古老的、最简单的分类器，应用非常广，但对其分类间隔研究相对比较少，而基于K近邻分类间隔的特征选择算法也较少。本章将从损失函数的角度对K近邻分类器的分类间隔进行分析，并介绍一种新的基于K近邻分类间隔的特征选择算法。

本章所介绍的算法的创新点在于利用最大分类间隔原理并结合损失函数来对特征进行排序。众所周知，特征排序是一种过滤器方法，它是一种预处理过程，独立于特定的学习器。如果特征之间是相互独立的，那么排序方法就可以获得最优的特征子集。即使不满足独立性，排序方法在计算开支上也好于其它方法，因为它只需要计算 n 个特征的性能指标并进行排序，同时从统计的观点来看，排序方法可以防止过学习[14]。

5.2 评价准则

假设存在训练集 S 含有 N 个样本 $\{x_i, y_i\}_{i=1}^{N}$，并且每个样本由 n 维特征矢量描述 $x_i = (x_{i1}, x_{i2}, \cdots, x_{in}) \in \Re^n$，其标记 y_i 是离散的。我们可以构建二值矩阵 **B**，其元素 $b_{ij} \in \{0,1\}$ 表示标记 y_i 和 y_j 是否相同。此外 k 个与 x_i 标记相同的最近邻同类样本命名为目标近邻，也就是说它们与 x_i 的距离最短且标记相同，这样可以定义二值矩阵 **T**，其元素 $t_{ij} \in \{0,1\}$ 表示 x_j 是否是 x_i 的目标近邻。矩阵 **B** 和 **T** 都是固定的，在特征选择的过程中是不变的，而距离度量采用的是欧氏距离。

5.2.1 基于损失的评价函数

损失函数是机器学习中一个常用来寻找分类误差与分类间隔大小之间的平衡的方法。一旦选定损失函数，学习算法就该考虑如何最小化损失函数以便得到最大的分类间隔[63]。对于输入样本x_i，其在K近邻分类中的损失函数可以定义如下：

定义 5.1 *设S为训练集，x_i为样本，则x_i的损失函数为：*

$$L_S(x_i)=\sum_j t_{ij}\|x_i-x_j\|^2+c\sum_{jp}t_{ij}(1-b_{ip})h_{jp}(x_i)$$

$$h_{jp}(x_i)=\left[\theta_i+\|x_i-x_j\|^2-\|x_i-x_p\|^2\right]_+ \tag{5.1}$$

其中$[z]_+=\max(z,0)$表示hinge损失，c为正常数，通常通过交叉验证得到。

值得注意的是在损失函数的第一项中惩罚那些与x_i较远的目标近邻样本，而不是所有与x_i具有相同标记的样本。而第二项则惩罚那些与x_i的目标近邻距离较近且与x_i标记不同的样本[53]。而作者尤其关注那些位于特定区域的具有与x_i不同标记的样本，这些样本与x_i的距离小于x_i到其所有目标近邻的距离再加上一个间隔θ_i，定义为：

$$\theta_i=\left\|x_i-nearmiss(x_i)\right\|^2-\left\|x_i-nearhit(x_i)\right\|^2 \tag{5.2}$$

其中$nearhit(x_i)$和$nearmiss(x_i)$分别表示与x_i最近邻的具有相同标记和不同标记的样本。它们很容易从矩阵B和T中得到。由于选择不同的特征子空间可以影响样本间的距离，从而影响K近邻分类的损失函数。因此可以通过选择特征子空间使得损失函数最小，从而可以将损失函数作为特征选择的评价准则。

如果许多样本都有着低损失和大的分类间隔，那么就可以保证算法有着好的推广性能。作者在介绍基于最大分类间隔的K近邻分类的特征选择评价准则之前，首先将K近邻分类的损失函数转化为所选择的特征子集的函数。

定义 5.2 *假设S为训练集，x_i为样本，w为特征集中每个特征权重构成的权重矢量，则样本x_i的基于特征权重的损失函数定义为：*

$$L_S(w,x_i)=\sum_j t_{ij}\|x_i-x_j\|_w^2+c\sum_{jp}t_{ij}(1-b_{ip})h_{jp}(w,x_i)$$

$$h_{jp}(w,x_i)=\left[\theta_i+\|x_i-x_j\|_w^2-\|x_i-x_p\|_w^2\right]_+ \tag{5.3}$$

其中$\|z\|_w=\sqrt{\sum_f w_f^2 z_f^2}$，$w_f\in[0,1]$。定义5.2在计算样本距离时考虑了特征的权重。我们可以通过特征的权重来对特征进行排序，再选择重要特征。

这样特征选择的评价准则就可以定义为所有样本的损失函数之和：

定义 5.3 *训练集S，权重矢量w，则评价函数为：*

$$e(w)=\sum_i L_S(w,x_i) \tag{5.4}$$

5.2.2 分类间隔

损失函数中融入了分类间隔的思想。尤其是定义 5.1 中的第二项，hinge 损失是由那些与 x_i 的标记不同且与 x_i 的距离小于 x_i 与其所有目标近邻的距离再加上一个预先定义的间隔 θ_i 的样本产生的。这样评价函数就会选择特征子集，使得与 x_i 标记不同的样本在所选择的特征空间里距离 x_i 和其目标近邻比较远，从而使得它们不会影响到 x_i 的目标近邻，从而获得大的分类间隔，保证分类的准确性。整个学习过程如图 5.1 所示。

此外，间隔 θ_i 的定义包含了最近邻分类（1-NN）中样本 x_i 的 Hypothesis 分类间隔的思想[63]。在文献[63]中，将分类间隔分为两种：Sample 间隔和 Hypothesis 间隔。Sample 间隔是指样本与分类界面的距离，如 SVM 中就采用这种间隔。而 Hypothesis 间隔是指某一个分界面与能使得样本类别保持不变的最临界分界面的距离，如 Adaboost 中就使用了此类间隔。1-NN 中样本 x_i 的 Hypothesis 间隔定义为：

$$\theta_{i,NN}(x_i)=\frac{1}{2}\left(\|x_i-nearmiss(x_i)\|-\|x_i-nearhit(x_i)\|\right) \tag{5.5}$$

最后，如在 SVM 中 hinge 损失是由决策面附近的样本产生的一样，本章所提出的评价函数中的 hinge 损失也只是由那些会影响到目标近邻的不同标记样本产生。

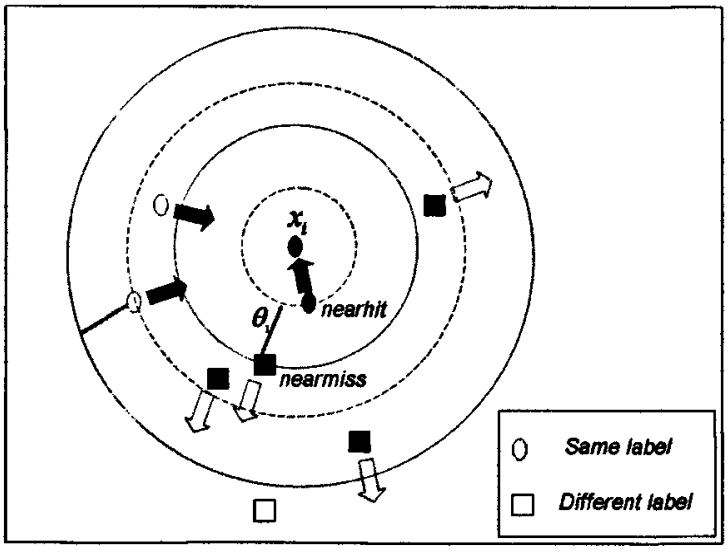


Figure 5.1. Illustration of optimization of evaluation function for one sample x_i: The target neighbors is represented by circle and the solid square denotes the differently labeled samples whose distance to

x_i is not exceed the distance from x_i to any of its target neighbors plus the predefined margin θ_i represented by green lines. Arrows indicate the gradients on distance arising from feature selection for the optimization of evaluation function.

图 5.1 评价函数的优化过程描述。圆形表示目标近邻，实心方块表示与 x_i 标记不同的样本，且它们与 x_i 的距离小于 x_i 与其任何目标近邻的距离加上间隔 θ_i，分类间隔用绿线表示。箭头方向表示通过特征选择使得样本间的距离变化方向。

5.2.3 能量模型

现在剩下的问题是“最小化所提出的损失函数是否可以使得样本之间发生如图 5.1 所示的行为？”。作者将从能量模型的角度深刻分析这个问题。能量模型为每一对样本 x, y 赋予一定的能量 $E(w,y,x)$，而能量函数是以矢量 w 为参数的，其是需要学习的[64, 65]。整个推导过程就是最小化能量函数寻找与 x 最接近的配置 y（例如图像分类）。而且，如文献[64]所讲，利用损失函数来训练能量模型可以规避评价部分函数及其求导。基于能量模型，损失函数将使得系统能够做出最好的决定。能量模型已经应用到相似性度量学习[66]、维数约简[67]以及人脸检测[68]等。在监督学习中，训练集 $\{x_i, y_i\}_{i=1}^N$，x_i 为输入，y_i 为理想的输出。实际上每个学习算法都可以描述为寻找参数矢量 w，使得精心设计的损失函数的值最小。损失函数可以用来描述在训练集中学习算法的实际行为与理想行为的差异。好的基于能量模型的损失函数应该能得到理想的输出使得 $E(w,y,x)$ 最小。因此，好的损失函数可以使得理想输出的能量 $E(w,y_i,x_i)$ 小于所有其它可能的输出。这样可以定义要正确分类样本 x_i，需要满足的条件为[64, 66]：

条件 1 $E(w,y_i,x_i) < E(w,y_{i'},x_i),\ \forall y_{i'} \neq y_i$。*为了确保分类结果更加稳定和鲁棒，我们可以考虑使得理想输出的能量小于其它输出的能量并减去间隔 m：*

$$E(w,y_i,x_i) < E(w,y_{i'},x_i) - m,\ \forall y_{i'} \neq y_i \tag{5.6}$$

现在剩下的问题是如何刻画损失函数的形式，满足最小化该损失函数可以最终使得学习过程满足条件 1，当 $E(w,y_{i'},x_i) < E(w,y_i,x_i) + m$ 时，最小化损失函数可以增加 $E(w,y_{i'},x_i) - E(w,y_i,x_i)$ 差值。换句话说，找到具有最小能量的非理想输出，如果其能量与理想输出的能量差小于给定的间隔，则学习过程就应该增大它们的差值。损失函数的统一形式如下：

$$L_S(w,x_i) = Q_{[E_y]}\left(E(w,y_i,x_i), E(w,y_{i'},x_i)\right) \tag{5.7}$$

其中参数 $[E_y]$ 包含除 y_i 和 $y_{i'}$ 以外的其它输出的能量。基于文献[64, 66]中相关结论，最小化 $L_S(w,x_i)$ 可以最终使得条件 1 得到满足的充足条件为：

条件 2 存在一个训练样本，可以找到一个 w 使得条件 1 得到满足。

条件 3 $E(w,y_i,x_i)$ 和 $E(w,y_i,x_i)$ 都是凸的

条件 4 在间隔面 $E(w,y_i,x_i)=E(w,y_i,x_i)+m$ 上，$Q(E(w,y_i,x_i),E(w,y_i,x_i))$ 的梯度负值在方向[-1，1]上存在的点积。

现在将就上面的条件对所提出的损失函数进行分析。

所提出的损失函数是基于 K 近邻分类的，样本 x_i 的标记是由目标近邻 x_j 的距离来决定的。因此可以用 x_j 和 x_p 来定义能量函数 $E(w,y_i,x_i)$ 和 $E(w,y_i,x_i)$，其中 x_p 为与 x_i 不同标记的样本。这样 $E(w,y_i,x_i)$ 和 $E(w,y_i,x_i)$ 就转化成 $E(w,x_j,x_i)$ 和 $E(w,x_p,x_i)$，并被定义为：

$$E(w,x_j,x_i)=\|x_i-x_j\|_w^2$$
$$E(w,x_p,x_i)=\|x_i-x_p\|_w^2 \tag{5.8}$$

对于条件 2，在所提出的损失函数（5.1）中，m 等于 θ_i。我们肯定可以找到一个样本 x_p，其与样本 x_i 的距离大于 $nearmiss(x_i)$ 与 x_i 的距离，肯定可以找到一个 w 使得 $E(w,nearhit(x_i),x_i)<E(w,x_p,x_i)-m$。条件 2 就得到了满足。

对于条件 3，很明显我们的能量函数是凸的。

基于在 2.2 小节中的分析，我们的损失函数属于推广（Generalization）的间隔损失。因此为了讨论条件 4，我们仅分析此类间隔损失，它直接使用最可能影响到理想输出的其它输出，并放置在对比项，一般形式如下：

$$Q(E(w,y_i,x_i),E(w,y_i,x_i))=Q^+(E(w,y_i,x_i))+Q^-(E(w,y_i,x_i)) \tag{5.9}$$

根据文献[64, 66]中的分析，如果 Q^+ 是单调递增的凸函数，而 Q^- 是单调递减的凸函数，则条件 4 将得到满足。在所提出的损失函数中，$Q^+=E(w,x_j,x_i)$ 是加权欧氏距离的平方函数，因此肯定是一个单调递增的凸函数。而 $Q^-=c[\theta_i+E(w,x_j,x_i)-E(w,x_p,x_i)]_+$，它是一个 hinge 损失，$c$ 为正常数。对于 Q^-，它关注的是特定区域内与 x_i 不同标记样本的能量，因此对于给定的目标近邻 x_j，$\theta_i+E(w,x_j,x_i)$ 可以被看作是一个常量。这样 Q^- 就等于：

$$Q^-=\begin{cases}0 & \text{if } (\theta_i+E(w,x_j,x_i)\le E(w,x_p,x_i)) \\ c.value & \text{otherwise}\end{cases}$$
$$value=\theta_i+E(w,x_j,x_i)-E(w,x_p,x_i) \tag{5.10}$$

对于 $E(w,x_p,x_i)$ 来说，Q^- 是一个单调递减的凸函数。

另外，我们考虑 k 个目标近邻以及所有到 x_i 的距离小于 x_i 到其任意目标近邻的距离加上间隔 θ_i 且与 x_i 标记不同的样本。基于公式（5.7）和（5.9），所有这些样本的能量相加就可以得到定义 5.2 中给出的损失函数。

综上所述，我们可以得到相关结论：最小化所提出的损失函数将能够得到一个权重矢量 w，使得学习过程如图 5.1 所示。

5.3 特征选择

5.3.1 基于损失间隔的算法 Lmba

为了寻找使得评价函数最小化的特征子集，许多搜索策略可以使用，如顺序前向和反向搜索、增 l 减 r、顺序浮动搜索、遗传算法和分支定界等[13, 70]。但是，它们只是将特征权重赋为 1 或者 0，分别表示特征被选或者没被选，且它们的时间复杂度至少为 $O(N^2n^2)$，其中 N 为训练集的大小，n 为特征数。因此我们想寻找一种更柔性的策略且时间开支较少。由于 $e(w)$ 几乎是平滑的，我们就考虑利用梯度下降去寻找权重矢量 w，使得评价函数最小，并利用阈值来确定最后的特征子集。评价准则的梯度定义如下：

$$\frac{\partial e(w)}{\partial w_f}=\sum_i \frac{\partial L_S(w,x_i)}{\partial w_f}=\sum_i\left(2w_f\sum_j t_{ij}(x_{if}-x_{jf})^2+c\sum_{jp}t_{ij}(1-b_{ip})\frac{\partial h_{jp}(w,x_i)}{\partial w_f}\right) \tag{5.11}$$

而 hinge 损失的梯度定义如下：

$$\frac{\partial h_{jp}(w,x_i)}{\partial w_f}=\begin{cases}0;\ \text{if}\left(\theta_i+\|x_i-x_j\|^2\right)\le\|x_i-x_p\|^2\\ g(w_f);\ \text{otherwise}\end{cases}$$
$$g(w_f)=2w_f\left((x_{if}-x_{jf})^2-(x_{if}-x_{pf})^2\right) \tag{5.12}$$

算法 Lmba 的步骤如下：

算法 5.1 Lmba

第一步：初始化 $w=(1,1,\cdots,1)$

第二步：计算矩阵 B 和 T

第三步：FOR i=1, 2, ..., N

(a) 随机选择样本 x_i

(b) 找到 $nearmiss(x_i)$ 和 $nearhit(x_i)$，并得到 θ_i 的值

(c) FOR f=1, 2, ..., n 计算

$$\nabla_f = 2w_f \sum_j t_{ij}\left(x_{if} - x_{jf}\right)^2 + c\sum_{jp} t_{ij}\left(1 - b_{ip}\right)\frac{\partial h_{jp}(w, x_i)}{\partial w_f}$$

(d) $w = w - \dfrac{\nabla}{\|\nabla\|}$

第四步： 根据特征权重 w 排序

在每次循环中，我们通过一个样本 x_i 来修改 w。由于特征的权重在不断增加，因此∇的作用在相对减少，这样算法就会典型收敛。Lmba 的时间开支主要是计算 B 和 T 以及 w，它们各自的复杂度为 $O(N^2)$，$O(N^2)$和 $O(N^2kn)$。而由于 k 值通常比较小，因此总的时间复杂度为 $2O(N^2)+O(N^2n)\approx O(N^2n)$。

5.3.2 其它相关算法比较

Simba[70, 71]也是一个基于分类间隔的特征选择算法，它能很好地评价特征的性能。它的评价准则是直接最大化 1-NN 分类器的 Hypothesis 间隔。Simba 算法流程如下：

算法 5.2 Simba

第一步：初始化 $w = (1,1,\cdots,1)$

第二步：FOR i=1, 2, ..., N

（a）随机选择样本 x_i

（b）找到 $nearmiss(x_i)$ 和 $nearhit(x_i)$

（c）FOR f=1, 2, ..., n 计算

$$\Delta_f = \frac{1}{2}\left(\frac{\left(x_{if} - nearmiss(x_i)_f\right)}{\|x_i - nearmiss(x_i)\|} - \frac{\left(x_{if} - nearhit(x_i)_f\right)}{\|x_i - nearhit(x_i)\|}\right)w_f$$

$$w = w + \Delta$$

第三步：$w \leftarrow \dfrac{w^2}{\|w^2\|_\infty}$

其时间复杂度为 $O(N^2n)$。搜索方法与 Lmba 类似，只是梯度的方向不一样。但是 Lmba 是通过最小化 K-NN 分类器的损失函数来间接优化分类间隔，而 Simba 是直接优化 1-NN 分类器的 Hypothesis 分类间隔。此外，Lmba 中 k 值是大于 1 的，因此它能更好地处理噪声数据。而在 θ_i 的定义中也融入了 1-NN 的 Hypothesis 分类间隔的思想。因此 Lmba 比 Simba 更鲁棒且可以看作是 Simba 的扩展。

Mitra's[24]是基于 K 近邻规则寻找特征高度相关的特征子集。时间复杂度是 $O(n^2N)$。虽然 Mitra's 也是基于 K 近邻规则，但它是一种非监督特征选择方法，在训练过程中没有用到样本的标记信息，没有融入分类间隔的思想。它仅仅是使用 K 近

邻规则将特征进行聚类并找到某个特征，其所在的类最紧凑，也就是该特征与其第 k 个近邻特征的相似性最大。然后选择该特征，并抛弃与其相邻的 k 个特征。对所有剩下的特征再重复这个过程。同时 k 在循环的过程会自动变化并且 k 可以控制所选择特征子集的大小。所有这些都与 Lmba 和 Simba 不同。Mitra's 算法流程详见第三章。

此外，Relief[36]是另外一个非常有名的基于最近邻规则的特征选择算法。该算法也是为每个特征估计权重，并通过训练对权重进行修改。Relief 算法在文献[37]中被扩展来处理多类问题以及噪声和缺失数据。Relief 中的权重修改规则与 Simba 相似，其时间复杂度是 $O(N^2n)$。但是以前的工作表明 Simba 是优于 Relief，并且 Simba 可以看着是 Relief 的提高版。因此在下面的实验中，我们将仅比较 Lmba、Simba 和 Mitra's 的性能，而不考虑 Relief。Relief 算法流程如下：

算法 5.4 Relief

输入：每个训练实例（样本）的特征矢量和类标记

输出：特征的关联程度矢量 w

第一步：初始化：$w[F]=0.0$　　*% F 为特征集;*

第二步：FOR $i=1,2,\ldots,m$　　*% m 为选取的训练实例数*

（a） 随机选择实例 x_i；

（b） 寻找与 x_i 同类的最近实例 $nearhit(x_i)$ 和不同类的最近实例 $nearmiss(x_i)$；

（c） FOR $j=1,2,\ldots,n$

$$w[f_j]=w[f_j]-diff(f_j,nearhit(x_i),x_i)/m+diff(f_j,nearmiss(x_i),x_i)/m$$

其中 n 表示所有特征，函数 *diff(特征，实例 1，实例 2)* 是计算两个实例在某个特征上的差异，对于离散特征，可以定义为：

$$diff(f_j,I_1,I_2)=\begin{cases}0; value(f_j,I_1)=value(f_j,I_2)\\ 1; otherwise\end{cases} \tag{5.13}$$

对于连续特征，可以定义为：

$$diff(f_j,I_1,I_2)=\frac{|value(f_j,I_1)-value(f_j,I_2)|}{\max(f_j)-\min(f_j)} \tag{5.14}$$

其中 I_1，I_2 为两个实例，$value(f_j,I_1)$ 表示 I_1 在特征 f_j 上的取值，$\max(f_j)$, $\min(f_j)$ 分别表示特征在 f_j 所有样本上的最大取值和最小取值。通常 m 取值越大，效果越好，一般等于训练样本数。

5.4 实验

我们将在不同的数据集上验证所提出的特征选择方法。实验分成三个部分：首先在基准和合成数据集上验证评价函数的正确性，看是否能将重要特征排在前面。使用 Matlab 中的 random 函数去产生合成数据。两个合成数据 S1 和 Multi-class 具有不同的类别数和特征数，但是都含有 100 个样本。对于 S1，重要特征呈高斯分布，不重要特征的取值是随机的。对于 Multi-class 数据集，样本的取值是随机的 $X=\{x_1,x_2,\cdots,x_{100}\}$，而样本的标记是由下面的 Matlab 函数产生的 $Y=bin2dec(num2str(X(:,1:2)>0))$。基准和现实数据集都来自 UCI 机器学习资料库。这些数据集的描述如表 5.1 所示。对于 Iris 和 Monk 数据集，基于一定的先验知识，它们的重要特征都是已知的，重要特征的标号如表 5.1 中最后一列（从左到右）所示。

表 5.1 基准和合成数据集描述

Table 5.1. Description of benchmark and synthesis data sets

数据集	特征数	类别数	重要特征
Iris	4	3	3,4
Monk	6	2	2,4,5
Multi-class	10	4	1,2
S1	22	6	1-6

此外 Multi-features 数据集也被用来评价特征选择算法 Lmba，Simba 和 Mitra's 的性能。它是荷兰公共事业地图上的手写字母（0-9）的特征构成，含有 2000 个样本，649 个特征以及 10 类。

最后在基于人脸图像的性别分类中进行实验。在性别识别中，用来训练的样本包括男女人脸图像各 500 张，特征是 1584 维的 gabor 过滤器。测试集包括 15 类性别数据，它们对应不同的姿态、表情、背景以及遮挡物等。相关描述见第四章。

5.4.1 实验结果

使用 Lmba 对基准数据集中的特征进行排序，排序后利用第三章中的所提出的方法设置阈值，并选择最后的特征子集。排序与选择结果如表 5.2 所示。从表中可以看到，Lmba 能够将重要特征排在最前面。对于 Multi-features 数据集，由于缺乏一定的先验知识，我们只比较所选择的不同大小特征子集的性能，而不列出所有选出的特征。采用 1-NN 分类器的分类性能去评价特征子集的性能，并采用 5 次交叉验证。由于 Mitra's 算法很难确定其参数 k 的确切值，因此在实验中只是确定其近似值，使得所选择的特征子集的维数近似等于或稍微大于 Lmba 和 Simba 所选择的特征数。在 Lmba 算法中，参数 c 和 k 被设为 1 和 3，后面的实验中也采用相同的设置。实验结果如表 5.4 所示。

表 5.2 基准和合成数据集的排序及选择结果

Table 5.2. Ranking and feature selection results for benchmark and synthesis data sets

数据集	排序	选择特征
Iris	{3,4},2,1	3,4
Monk	{5,4,2},1,6,3	5,4,2
S1	{2,3,6,1,4,5},8,...	2,3,6,1,4,5
Multi-class	{2,1},9,6,...	2,1

表 5.3 Multi-features 数据集实验结果

Table 5.3. Experimental results for Multi-features

选择特征数	算法	准确率/方差
13	Lmba	86.93/0.13
	Simba	89.96/0.10
	Mitra's	85.47/0.14
26	Lmba	90.15/0.09
	Simba	92.60/0.07
	Mitra's	87.51/0.11
52	Lmba	93.81/0.06
	Simba	92.92/0.07
	Mitra's	90.19/0.09
104	Lmba	95.14/0.04
	Simba	94.12/0.05
	Mitra's	92.86/0.07
208	Lmba	95.68/0.04
	Simba	94.32/0.05
	Mitra's	93.41/0.06
416	Lmba	95.76/0.04
	Simba	94.40/0.05
	Mitra's	94.43/0.05

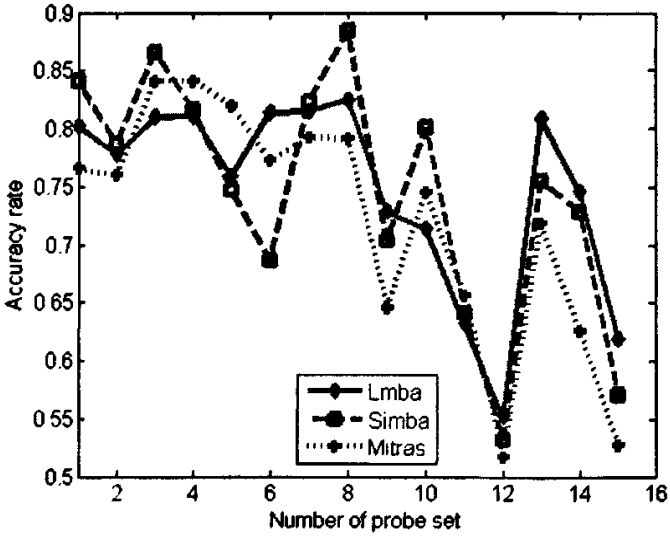
5.4.2 性别分类

在本小节中，将给出不同特征选择算法 Lmba、Simba 和 Mitra's 在不同大小的特征子集上的性别分类结果。所选择的特征数为 127、254、508、1016。分类器采用的是支持向量机[60]，其中参数 C 设为 1。不同算法在 15 个测试集上的结果如图 5.2 所示。X 轴为测试集的编号，而 Y 轴为性别分类准确率。在所有测试集上的平均准确率如表 5.4 所示。

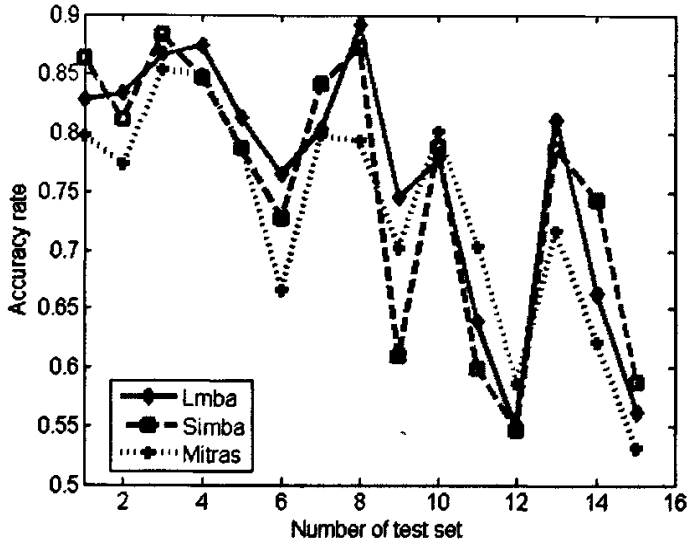
表 5.4 性别分类在测试集上的平均准确率

Table 5.4. Average accuracy rate for gender classification

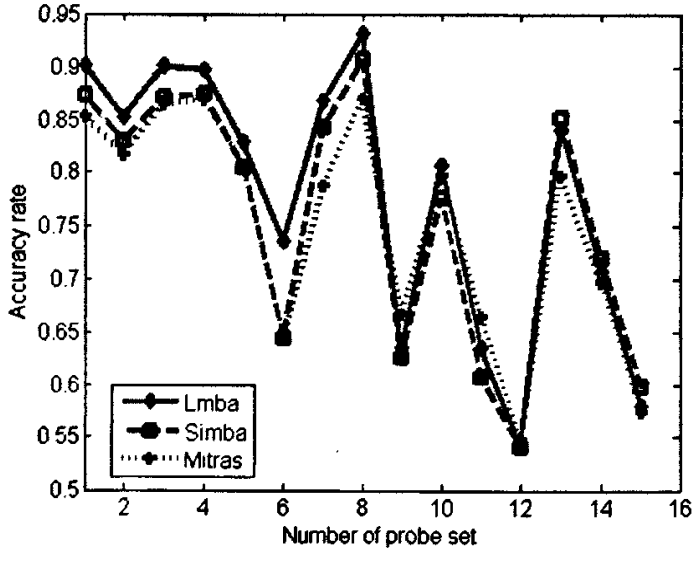
选择特征数	算法	平均准确率
127	Lmba	74.73
	Simba	74.55
	Mitra's	72.13
254	Lmba	76.23
	Simba	75.26
	Mitra's	73.19
508	Lmba	77.72
	Simba	75.78
	Mitra's	75.00
1016	Lmba	79.08
	Simba	77.58
	Mitra's	76.59



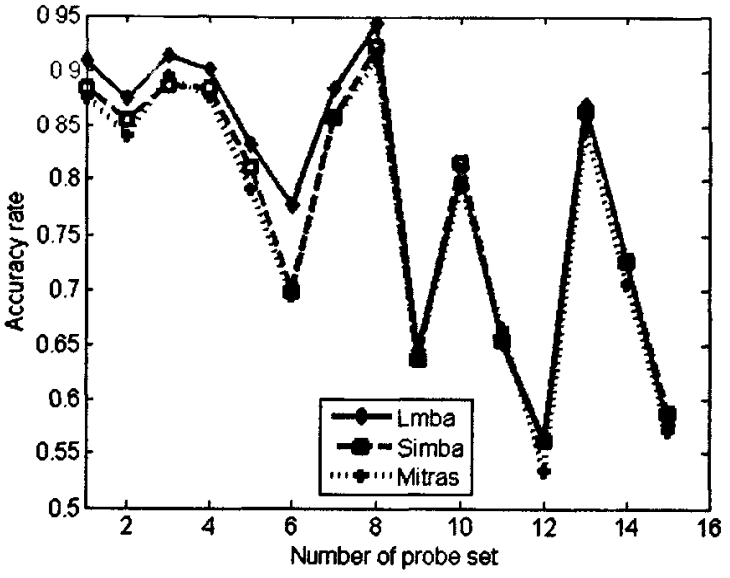
(a)


Accuracy rate
Number of test set
Lmba
Simba
Mitras

(b)


Accuracy rate
Number of probe set
Lmba
Simba
Mitras

(c)



(d)

图 5.2 在 15 个测试集上的准确率，分别对应不同特征数（a）127，（b）254，（c）508，（d）1016

Figure 5.2. The accuracy rates of fifteen probe sets for the different number of selected features (a)127, (b)254, (c)508 and (d)1016

5.4.3 观察评论

1）所提出的评价准则能够准确地排序重要特征；

2）对不同的数据集，Lmba 和 Simba 通常能在不同分类器上获得比 Mitra's 更高的性能。同时 Lmba 在大部分情况下分类性能最高，且对于性别分类，Lmba 获得最高性能的测试集数目比 Simba 多。

3）随着所选择特征数量的增加，Lmba 能获得更好的性能，且在性别识别中 Lmba 获得最高性能的测试集数量也在增加。

对于实验结果，Simba 是基于 1-NN 分类的 Hypothesis 间隔，它在没有噪声的数据集上以及采用 1-NN 分类器时效果最好。因此我们只需要在这样的环境下对 Lmba 和 Simba 进行比较。当然如果采用其它分类器，如 K-NN（K>1），并且数据集里含有噪声，则 Lmba 肯定能获得比 Simba 更好的性能。这主要是因为 Lmba 中所使用的 K 近邻规则能更好的处理噪声数据[63]，而且评价准则是基于 K 近邻分类间隔的。众所周知，如果一个算法能够选择特征空间以增加分类间隔，则该特征子集肯定有好的推广能力[70]。

5.5 本章小节

在本章中，作者使用最大间隔原理结合损失函数来设计特征选择算法。提出了一种基于 K 近邻分类损失函数的评价准则，并且从分类间隔和能量模型角度对评价准则进行了理论分析。通过使用梯度下降来最小化所提出的评价准则，从而实现特征选择。从实验中可以看出所提出的算法 Lmba 能够获得比 Simba 和 Mitra's 更好的性能。虽然评价准则是基于 K 近邻分类，但在其它分类器上，如 SVM，仍能获得比 Simba 和 Mitra's 更好的性能。此外，由于 K 近邻规则能更好地过滤噪声数据，因此 Simba 对噪声数据具有一定的鲁棒性。

在今后的研究可以从以下几个方面深化，其中之一是使用更好的优化方法来最小化评价准则。也可以考虑在评价准则中使用其它距离度量以及将特征选择与距离度量的学习相结合。

6 总结

特征选择作为数据预处理的一个必要步骤，是模式识别中一个关键问题，同时又是一个棘手的问题。对特征选择的研究有着几十年的历史。特征选择算法目前有着广泛的应用，如人脸识别、文本分类、图像检索、客户关系管理、入侵检测和基因分析等。由于与特征选择相关的许多问题都是NP难问题，研究人员总是努力地寻找着各种方法来尽量提高特征选择的性能。作者对特征选择的理论和算法进行了深入研究，取得了一定的研究成果。本报告就是对这些研究成果的一个全面总结。

报告全面介绍了特征选择算法的特性，并详细描述作者在特征选择算法的设计和应用上的研究成果心得。

其中，在特征选择算法的特性分析中，介绍了特征选择算法的理论模型。并讨论了合适特征选择算法的选用问题。这必将为特征选择算法的应用打开方便之门，也为对特征选择算法的研究提供基石。

设计新的特征选择算法是作者的主要研究内容，而在特征选择算法的设计中，主要包括在非监督高维特征选择、基于主成分分析 PCA 的特征选择和基于分类间隔的特征选择的研究。其中在非监督高维特征选择中，提出了一种基于排序的过滤器方法，它是非监督的且能处理高维数据和噪声数据，时间复杂度也比较低，相对于目前的其它非监督特征选择算法，它的功能更全面，应用面更广；基于 PCA 的特征选择是将特征抽取中典型方法 PCA 与特征选择相结合，从而将没有实际意义的主成分映射到原始空间，找到关键的原始特征；分类间隔是目前机器学习领域的研究热点，在分析 K 近邻分类的损失函数和分类间隔的基础上，提出了基于 K 近邻分类间隔的特征选择算法，并从能量模型的角度进行了深刻的理论分析。

特征选择算法还有许多问题亟待解决，如对高维的特征选择，特别训练样本较少，而特征是极高维（大于万）的特征选择，如何设计有效算法；在样本的类别数未知的情况下，时间复杂度较低的非监督特征选择仍然是值得关注的问题；如何在进行特征选择的同时，选择有意义的训练样本；此外将维数约简与距离度量学习相结合，也是一个重要的研究方向。由于特征选择的应用领域不断扩大，当出现新的数据类型时，如何设计新的特征选择算法。另外，在不同的应用领域设计具体的、专业的、高效的特征选择算法，也是一个值得重视的研究方向，同时将设计的算法应用到具体的领域也是非常重要的。

参考文献

[01] L. C. Molina, L. Belanche and A. Nebot, Feature selection algorithm: a survey and experimental evaluation, In Proc. 2002 IEEE International Conference on Data mining, pp.306-313, 9-12 Dec. 2002

[02] M. Dash and H. Liu, Feature selection for classification, Intelligent Data Analysis, pp.131-152, 1997

[03] O. Chapelle, V. Vanpnik, *et al*, Choosing multiple parameters for support vector machines, Machine Learning, vol.46, pp.131-159, 2002

[04] 张鸿宾，孙广煜，TABU 搜索在特征选择中的应用，自动化学报，vol.7（4），pp.457-466, 1999

[05] P. M. Narendra and K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. on Computer, vol.26(9) , pp.917-922, 1977

[06] T. M. Cover, The best two independent measurements are not the two best, IEEE Trans. System Man Cybernetic, vol.4(2), pp.116-117, 1974

[07] E. Backer and J. A. Shipper, On the max-min approach for feature ordering and selection, Proc. Seminaron Pattern Recognition, Liege, 1977

[08] J. Kittler, *et al*, Feature set search algorithms, Pattern Recognition and Signal Process, pp.41-60, 1978

[09] W. Siedlecki and J. Sklansky, On automatic feature selection, Int'l J. Pattern Recognition and Artificial Intelligence, vol.2(2), pp. 197-220, 1988

[10] W. Siedlecki and J. Sklansky, A note on genetic algorithm for large-scale feature selection, Pattern Recognition Letters, vol.10(11), pp. 335-347, 1989

[11] P. Pudil, Novovicova, *et al*, Floating search methods for feature selection, Pattern Recognition Letters, vol.15(11), pp.1119-1125, 1994

[12] A. Jain and D. Zongker, Feature selection: evaluation, application and small sample performance, IEEE Trans. Pattern Analysis and Machine Intelligent, vol. 19(2), pp.153-158, 1997

[13] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. on Knowledge and Data Engineering, vol.17(3), pp.1-12, 2005

[14] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, J. of

Machine Learning Research, vol. 3, pp.1157-1182, 2003

[15] 陈彬，洪家荣，王亚东，最优特征子集选择问题，计算机学报，vol. 20（2），pp.133-138, 1997

[16] 宋明，王俊普，蔡庆生，一种最优特征集的选择算法，计算机研究与发展，vol.35(9), pp.803-805, 1998

[17] 章新华，一种特征选择的动态规划方法，自动化学报，vol.24(5), pp.675-680, 1998

[18] J. Doak, An evaluation of feature selection methods and their application to computer security. Technical report, Davis CA: University of California, Department of Computer Science, 1992.

[19] P. Berkhin, Survey of clustering data mining techniques, Technical Report, Accrue software, 2002

[20] 边肇祺，张学工，模式识别，第二版，北京：清华大学出版社, 2001

[21] H. Liu and R. Setiono. A probabilistic approach to feature selection - a Filter Solution. In Pro. of the Thirteenth International Conf. on Machine Learning , pp. 319-327, 1996

[22] M. Dash, K. Choi, P. Scheuermann and H. Liu, Feature selection for clustering – a filter solution, Proc. of the Second International Conf. on Data Mining, pp.115-122, 2002.

[23] M. Dash and H. Liu, Feature selection for clustering, Proc. of Fourth Pacific Asia Conf. on Knowledge Discovery and Data Mining, (PAKDD-2000) , pp.110-121, 2000.

[24] P. Mitra, C. A. Murthy and S. K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Analysis and Machine Intelligence vol.24(3), pp.301-312 , 2002.

[25] J. G. Dy and C. E. Brodley, Feature subset selection and order identification for unsupervised learning, Proc. seventeenth Int'l Conf. Machine Learning, pp.247-254, 2000.

[26] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick and A. M. Aisen, Unsupervised feature selection applied to content-based retrieval of lung images, IEEE Trans. Pattern Analysis and Machine Intelligence , vol. 3(25), pp. 373-378, 2003.

[27] M. Dash, K. Choi, P. Scheuermann and H. Liu, Feature selection for clustering-A filter solution, IEEE Int'l Conf. on Data Mining, pp. 115-122, 2002.

[28] M. Dash and H. Liu, Feature selection for clustering, Proc. Pacific Asia Conf. on

Knowledge Discovery and Data Mining, pp.110-121, 2000.

[29] Y. Kim, W. Street, and F. Menczer, Feature selection in unsupervised learning via evolutionary search, Proc. 6th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp.365-369, 2000.

[30] L. Talavera, Dependency-based feature selection for clustering symbolic data, Intelligent Data Analysis, vol. 4, pp.19-28 , 2000.

[31] M. Devaney and A. Ram, Efficient feature selection in conceptual clustering, Proc. Fourteenth Int'l Conf. Machine Learning, pp.92-97, 1997.

[32] S. K. Pal, R. K. De and J. Basak, Unsupervised feature evaluation: a neuro-fuzzy approach, IEEE Trans. Neural Network, vol.11(3), pp. 366-376, 2000.

[33] H. C. Mart, A. T. Mario, Figueiredo and A. K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 9(26), 1-13, 2004.

[34] S. Chang, N. Dasgupta and L. Carin, A Bayesian approach to unsupervised feature selection and density estimation using expectation propagation, IEEE Conf. Computer Vision and Pattern Recognition, vol. 7, pp.1043 –1050, 2005.

[35] Y. Li, Z. F. Wu and J. M. Liu, Efficient feature selection for high-dimensional data using two-level filter, Proc. of the third Int'l Conf. Machine Learning and Cybernetics, vol.8, pp.1711-1716, 2004.

[36] K. Kira and L. Rendell, A Practical approach to feature selection, Proc. ninth Int'l Workshop Machine Learning, pp. 249-256, 1992

[37] I. Kononerko, Estimating attributes: analysis and extension of RELIEF, Proc. of European Conf. on Machine Learning, pp.171-182, 1994.

[38] J. Basak, R. K. De and S. K. Pal, Unsupervised feature selection using a neuro-fuzzy approach, Pattern Recognition Letters, vol.19, pp.997-1006, 1998.

[39] J. Basak, R. K. De and S. K. Pal, Unsupervised neuro-fuzzy feature selection, Proc. of IEEE Int'l Joint Conf. On Neural Network, vol.1, pp. 18-23, 1998.

[40] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research, vol.3, pp.1157-1182, 2003.

[41] I. Guyon, J. Weston, S. Barnhill and V. Vapnik. Gene selection for cancer classification using support vector machines, Machine Learning, vol.46(1-3), pp.389-422, 2002

[42] J. Bi, K. Bennett, M. Embrechts, C. Breneman and M. Song. Dimensionality reduction via sparse support vector machines. J. of Machine Learning Research, vol.

3, pp.1229-1243, 2003

[43] C. L. Blake and C. J. Merz, UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998

[44] A. K. Jain, R. P. Dulin and J. Mao, Statistical pattern recognition: a review, IEEE Trans. on Pattern Recognition and Machine Intelligence, vol.22(1), pp.4-37, 2000.

[45] K. Z. Mao, Identifying critical variables of principal components for unsupervised feature selection. IEEE Trans. Systems, Man, and Cybernetics-part B: Cybernetics, vol.35(2), pp. 339-344, 2005

[46] G. P. McCabe, Principal variables, Technometrics, vol.26, pp.127-134, 1984.

[47] W. J. Krzanowski, Selection of variables to preserve multivariate data structure using principal components. Appl. Statist. Vol.36(1), pp.22-33, 1987

[48] I. T. Jolliffe, Discarding variables in a principal component analysis I: Artificial data. Appl. Statist. Vol.21(2), pp.160-173, 1972.

[49] I. T. Jolliffe, Discarding variables in a principal component analysis II: Real data. Appl. Statist. Vol.22(1), pp.21-31, 1973

[50] I. Cohen, Q.Tian, X. S. Zhou and T. S. Huang, Feature selection using principal feature analysis, Proc. of IEEE Int'l Conf. on Image Processing. 2002

[51] I. T. Jolliffe, Principal component analysis, New York: Springer-Verlag, 2002

[52] P. Mitra, C. A. Murthy and S. K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Recognition and Machine Intelligence, vol.24(3), pp. 301-312, 2002

[53] K. Q. Weinberger, J. Blitzer and L. K. Saul, Distance metric learning for large margin nearest neighbor classification, Advances in Neural Information Processing Systems, vol.18, 2006.

[54] J. Kittler, M. Hatef, R. P. W. Duin and J. Matas, On combing classifiers, IEEE Trans. on Pattern Recognition and Machine Intelligence, vol.20(3), pp. 226-239, 1998

[55] B. L. Lu and M. Ito, Task decomposition and module combination based on class relations: a modular neural network for pattern classification, IEEE Trans. on Neural Networks, vol.10, pp. 1244-1256, 1999

[56] B. L. Lu, K. A. Wang, M. Utiyama and H. Isahara, A part-versus-part method for massively parallel training of support vector machines, Proc. of Int'l Joint Conf. Neural Networks'04, Budapast, pp.735-740, July 25-29, 2004

[57] B. L. Lu, J. Shin and M. Ichikawa, Massively parallel classification of single-trial EEG signals using a min-max modular neural network, IEEE Trans. on Biomedical Engineering, vol.51(3), pp. 551-558, 2004

[58] H. C. Lian and B. L. Lu, Gender recognition using a min-max modular SVM, LNCS 3611, pp.433-436, 2005

[59] K. Z. Mao, Orthogonal forward selection and backward elimination algorithms for feature subset selection, IEEE Trans. Systems, Man, and Cybernetics-part B: Cybernetics, vol.34(1), pp.629-634, 2004

[60] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/»cjlin/papers/libsvm.ps.gz

[61] J. Weston, S. Mukherjee, O. Chapelle, *et al.* Feature selection for SVMs, In Advances in Neural Information Processing Systems, vol.13, 2001b.

[62] K. Crammer, R. G. Bachrach, A. Navot and N. Tishby. Margin analysis of the lvq algorithm, Proc. of Advances in Neural Information Processing System, La Jolla CA, 2002

[63] T. Cover and P. Hart. Nearest neighbor pattern recognition, IEEE Trans. Information Theory, vol.13, pp.21-27, 1967

[64] Y. LeCun and F. J. Huang. Loss functions for discriminative training of energy-based models, Proc. of International Workshop on Artificial Intelligence and Statistics, 2005

[65] Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang and M. A. Ranzato, A tutorial on energy-based learning, Predicting Structured Outputs, Bakir et al. (eds), MIT Press, 2006.

[66] S. Chopra, R. Hadsell and Y. LeCun, Learning a similarity metric discriminatively, with application to face verification. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA, pp.539-546, 2005

[67] R. Hadsell, S. Chopra and Y. LeCun. Dimensionality reduction by learning an invariant mapping, Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, vol.2, pp.1735-1742, 2006

[68] M.Osadchy, M. L. Miller and Y. LeCun. Synergistic face detection and pose estimation with energy-based model, Proc. of Advances in Neural Information Processing Systems, 2004

[69] P. A. Devijver and J. Kittler, Pattern recognition: a statistical approach, Prentice Hall, Englewood Cliffs, 1982.

[70] R. G. Bachrach, A. Navot and N. Tishby. Margin based feature selection-theory and algorithm, Proc. of the 21th Int'l Conf. on Machine Learning, Banff Canada, 2004

[71] I. Guyon, S. Gunn, M. Nikravesh and L. Zadeh, Feature extraction, foundations and applications, Springer Physica-Verlag, New York, 2006.

附 A：攻读博士学位期间发表论文及参加科研情况

1、论文发表情况

[1] Y. Li, Z. F. Wu, et al. "Efficient feature selection for high-dimensional data using two-level filter", Proceedings of the third International conference on machine learning and cybernetics (ICMLC), Vol.8, pp. 1711-1716, 2004 (EI)

[2] Y. Li, J. M. Liu, J. Li, Z. F. Wu, et al. "The fuzzy similarity measures for content-based image retrieval", Proceedings of the second International conference on machine learning and cybernetics (ICMLC), Vol.11, pp.3224-3228, 2003 (EI)

[3] 李云，吴中福，刘嘉敏."基于扩张矩阵的模糊特征选择算法"，模式识别与人工智能, Vol. 4(17), pp. 417-423, 2004 (EI)

[4] 李云，刘嘉敏，强保华，吴中福等."图像检索中相关反馈技术的特性研究"，计算机工程, Vol.30(7), pp.128-139, 168, 2004

[5] 李云，叶春晓，吴中福. "基于特征关联性的特征选择算法研究"，微型机与应用, Vol.6, pp. 58-60, 2004

2、参加科研情况

[1] 重庆自然科学基金（编号：2004BB2226）：基于信息融合的人耳特征综合识别技术研究。

[2] 重庆大学骨干教师资助计划项目（2003A13）：基于几何特征的人脸图像检索算法研究

[3] 教育部项目：现代远程教育资源建设

[4] 博士点基金：基于用户和角色属性的访问控制研究

附 B：博士后期间发表论文及参加科研情况

1、论文发表情况

[1] Y. Li and Z. F. Wu, "Fuzzy feature selection based on min-max learning rule and extension matrix", Pattern Recognition, vol.41, pp. 217-226, 2008 (SCI, EI)

[2] Y. Li, B. L. Lu and Z. F. Wu, "Hierarchical fuzzy filter method for unsupervised feature selection", Journal of Intelligent & Fuzzy Systems, vol.18, pp.157-169, 2007 (SCI, EI)

[3] Y. Li, B. L. Lu and Z. F. Wu, "A hybrid method of unsupervised feature selection based on ranking", Proceedings of International conference on Pattern Recognition (ICPR), vol.2, pp.687-690, 2006 (EI)

[4] Y. Li and B. L. Lu, "Feature selection for identifying critical variables of principal components based on k-nearest neighbor Rule". International Conference Series on Visual Information Systems (VISUAL), LNCS 4781, 2007 (In press)

[5] Y. Li and B. L. Lu, "Feature selection based on loss-margin nearest neighbor classification", (In preparation)

2、参加科研情况

[1] 中日合作项目：性别分类和年龄估计数据库；

[2] 国家自然科学基金项目(编号：60375022)：增量学习模型研究；

[3] 国家自然科学基金项目(编号：60473040)：超并列模式分类器的问题分解与模块集成研究；

致 谢

首先，我要衷心感谢我的合作导师吕宝粮教授。吕老师对我的工作自始至终给予了无私的指导和关注。吕老师严格谨慎、一丝不苟的治学作风和对科学事业孜孜不倦、不懈追求的工作精神一直深深地感染着我，影响着我，更激励着我。在吕老师的指导和帮助下，两年来我受益匪浅，眼界不断拓宽，业务水平也得到很大的提高。

此外，我要感谢仿脑计算与机器智能研究中心的博士、硕士生们，特别是图像组中两年来跟我一起并肩战斗的“战友”们。他们是范志刚、连惠城、李敬、罗俊、李季檩、周峰、刘爽以及我所指导的本科生谢金融和连晓晨。谢谢他们对我工作的支持，与他们在学术上的交流，使我能集思广益、受益良多。

最后，我要感谢我的父母，在这两年博士后研究期间，他们一直给我以极大的支持和鼓励。用言辞无法表达我对他们的感激之情。离开了他们，本文的一切都无从谈起。

本报告的研究获得国家自然科学基金 60375022 和 60473040 资助。

李 云

2007.8.21

个人简历

李云，男，安徽安庆，1995 年毕业于安徽大学，2002 年获得重庆邮电大学计算机应用专业硕士学位，2005 年获得重庆大学计算机软件与理论博士学位，目前在上海交通大学计算机科学与技术流动站从事博士后研究，主要研究兴趣包括机器学习和计算机视觉。

电子邮件：liyun_mail@sjtu.edu.cn，yuncloudlee@gmail.com；

个人主页：http://bcmi.sjtu.edu.cn/~liyun