文章编号:1001-506X(2011)11-2558-06

# 基于决策树的快速 SVM 分类方法

崔 建[1],李 强[1],刘 勇[2],宗大伟[3]
(1. 空军雷达学院预警监视情报系,湖北 武汉 430019;
2. 空军驻京津地区代表室,北京 100015;
3. 华中数控股份有限公司,湖北 武汉 430223)

**摘 要**:为提高支持向量机(support vector machine,SVM)算法对大规模数据的适应能力,加快 SVM 算法的分类速度,提出一种基于决策树的快速 SVM 分类方法。该方法的重点在于构建一棵决策树,将大规模问题分解为相对简单的子问题,树中节点由线性支持向量机组成,每个节点包含一个决策超平面,分类过程取决于节点的数量。此方法在分类复杂样本时避免了使用非线性核函数。并且由于使用线性核函数,则不用进行模型选择,进一步加快了样本的分类速度。实验表明,针对大规模多特征数据的非线性分类问题,该方法比传统方法具有更高的速度。

**关键词**:支持向量机;快速分类;决策树;大规模数据
**中图分类号**:TP 311 **文献标志码**:A **DOI**:10.3969/j.issn.1001-506X.2011.11.40

# Fast SVM classification method based on the decision tree

CUI Jian[1], LI Qiang[1], LIU Yong[2], ZONG Da-wei[3]
(*1. Department of Early Warning Surveillance Intelligence, Air Force Radar Institute, Wuhan 430019, China;*
*2. Air Force Representative Office in Beijing and Tianjin, Beijing 100015, China;*
*3. Huazhong Numerical Control CO. LTD, Wuhan 430223, China*)

**Abstract**: In order to improve the large-scale data adaptability of the support vector machine (SVM) algorithm, accelerate the classification speed of the SVM algorithm, one fast SVM classification method is proposed based on the decision tree. The focus of this method is to construct a decision tree and decompose the large-scale problem into relatively simple sub-problems, the tree nodes are composed by the linear SVMs, then each node contains a decision hyperplane, the classification process depends on the number of nodes. This method avoids using the nonlinear kernel function in classification of complex samples, and by using a linear kernel function, it needs not to undertake the model selection, thus accelerating the samples classification rate. Experiments show that for the nonlinear classification problem of large-scale data with multiple features, the method has higher speed than the traditional methods.

**Keywords**: support vector machine (SVM); fast classification; decision tree; large-scale data

## 0 引 言

支持向量机(support vector machine, SVM)是 Cortes & Vapnik 1995 年首先提出来的,是近年来机器学习研究的一项重大成果[1]。SVM 在解决分类、回归和密度估计等机器学习领域表现出许多独有的优势,被认为是目前针对小样本的分类、回归等问题的最佳方法[2]。但由于非线性的 SVM 训练过程中存在大量计算,对于大规模数据的分类问题,如果采用这种模型分类,分类速度将会非常缓慢[3]。

SVM 的训练需要求解一个二次规划的优化问题。首先训练的迭代过程需要多次计算和存储整个核函数赫森(Hessian)矩阵,其元素个数是 $m^2$($m$ 是样本数),更关键的是赫森矩阵不是稀疏的,这导致计算矩阵,尤其是非线性核函数矩阵需要消耗大量的时间。因此,SVM 对小规模训练集非常有效,但在实际中训练集规模常常比较大。国际上研究 SVM 最著名的学者,Vapnik、Smola、Scholköpf、Keerthi 等提出了一些关于 SVM 的开放性问题,其中将 SVM 快速算法列为最迫切需要解决的问题之一。

为了提高 SVM 的分类速度,研究人员已经做了大量的工作。文献[4]提出一种直接缩减支持向量(support vec-

**收稿日期**:2010-12-16;**修回日期**:2011-05-04。
**基金项目**:国家自然科学基金(60736009)资助课题
**作者简介**:崔建(1981-),男,博士研究生,主要研究方向为数据挖掘和情报分析。E-mail:ldxy-cj@163.com

tors, SVs)数量的方法,但这种方法在决定缩减集的时候计算量非常大;后来,文献[5]改进了该方法,确定及删除不必要的 SVs,约简率最高能达到 40.96%;文献[6]提出利用主成分分析法(principal components analysis, PCA)缩减特征空间,来减少向量分量的数量,从文献实验结果来看,该方法在分类时间上比传统 SVM 提高了 10 倍左右;文献[7]通过实验证明将大的边界引入到使用线性决策函数的决策树中,可以提高 SVM 的泛化能力,但此方法要依赖并调节一些参数才能得到满意的结果;文献[8]对 SVM 处理大规模数据集的训练时间建立了评估模型;文献[9]成功地利用 LIBSVM 包建立了应用于生物信息学领域的大规模预测模型;文献[10]提出一种新的方法利用新增加的数据集用于提高分类精度,降低数据集的时间,仿真结果表明此方法得到较低的误分率;文献[11]给出了一种基于网格的分布式 SVM 算法,用于实现大规模数据的处理,目前已经用 C 语言加以实现,关于多类分类和参数优化还需进一步研究;国内方面有利用特征空间的向量投影预选 SVs[12],和在模糊支持向量机中引入类中心思想实现样本集缩减[13]。这些方法的实质是通过预选取支持向量而缩减训练样本集,并且不影响 SVM 的分类性能。但这些方法在样本缩减过程中需要完全计算或存储核矩阵$(\boldsymbol{K}_{ij})_{i,j=1}^{m}$,其计算复杂度仍然依赖于样本数 $m$,在样本规模很大时导致缩减样本的时间过长。

为提高大规模数据的分类速度,本文给出了一种新的 SVM 分类方法 HardTreeSVM。该方法借鉴决策树的多分类 SVM 算法思想,将原始问题分解为若干个线性子问题,每个子问题对应于决策树一个节点,得到一个分类超平面,并通过对 SVM 目标函数和约束条件的修改,使获得的超平面能够对硬类(hard class, HC)完全分类、对非硬类(non hard class, NHC)进行最大分类间隔的划分。由于 HardTreeSVM 算法在分类过程中使用线性核函数,并且对测试样本的分类大多不用遍历完整个决策树,同时该方法避免了非线性 SVM 的调节参数问题,所以使得分类速度大大加快。实验表明了该方法的有效性。具体过程如下。

## 1　相关问题定义

**定义 1**　(两类 SVM 问题)令 $m_1$ 和 $m_2$ 为两个自然数,满足 $m=m_1+m_2$,$m_1>0$,$m_2>0$,并令 $l=\{1,\cdots,m\}$,则对正类和负类有如下定义:

**正类(类 1)**　训练样本中的正类由集合$\{x_i\}$组成,其中 $i\in l_1$,$l_1=\{1,\cdots,m_1\}$,$m_1$ 为正类样本的个数,对于所有 $i\in l_1$,有 $y_i=1$;并定义 $C_i$ 为单个样本的惩罚值,$i\in l_1$。

**负类(类 2)**　训练样本中负类由集合$\{x_i\}$组成,($i\in l_2$,$l_2=\{m_1+1,\cdots,m_1+m_2\}$),$m_2$ 为负类样本的个数,对于所有 $i\in l_2$,有 $y_i=-1$;并定义 $C_i$ 为单个样本的惩罚值,$i\in l_2$。

如果存在一个线性判别函数(即最优分类超平面)$P$ 的方程为 $P:\{\boldsymbol{x}\in H|\langle \boldsymbol{w},\boldsymbol{x}\rangle+b=0\}$($\boldsymbol{w}\in H$,$b\in\mathbf{R}$)能够无误差地划分正负两类样本,则称两类样本时线性可分的。向量 $\boldsymbol{w}$ 是与超平面 $P$ 正交的向量,并且$\langle \boldsymbol{w},\boldsymbol{x}\rangle$是 $\boldsymbol{x}$ 在 $\boldsymbol{w}$ 方向上的投影的长度。

分类学习问题的最优目标函数和约束条件可以归纳为定义 2 的形式。

**定义 2**　(SVM 初始最优问题)类 1 和类 2 分别为定义 1 中所定义,最优边界超平面的初始问题可以按照如下定义

$$\min_{\boldsymbol{w}\in H,b\in\mathbf{R}} \text{imize}\ \ \tau(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}\|^2 \tag{1}$$

$$\text{s.t.}\ \ y_i(\langle \boldsymbol{x}_i,\boldsymbol{w}\rangle+b)\geqslant 1,\ i=1,\cdots,m \tag{2}$$

相应的决策函数为

$$f(x)=\text{sign}(\langle \boldsymbol{w},\boldsymbol{x}\rangle+b) \tag{3}$$

对于训练样本集近似线性可分的情况,即没有线性解能够将两类样本完全分离。引入松弛变量 $\xi=\{\xi_1,\cdots,\xi_m\}$,允许分类面对训练样本有一定的错分。对于此类问题,通过在目标函数中增加松弛变量和惩罚值,将得到下面边界最优问题形式:

**定义 3**　(C-SV 分类初始问题)对于两类问题,带松弛变量的初始问题定义如下

$$\min_{\boldsymbol{w}\in H,b\in\mathbf{R}\xi\in\mathbf{R}^m} \text{imize}\ \ \tau(\boldsymbol{w},\boldsymbol{\xi})=\frac{1}{2}\|\boldsymbol{w}\|^2+\sum_{i=1}^{m}C_i\xi_i \tag{4}$$

$$\text{s.t.}\ \ y_i(\langle \boldsymbol{x}_i,\boldsymbol{w}\rangle+b)\geqslant 1-\xi_i,\ i=1,\cdots,m \tag{5}$$

$$\xi_i\geqslant 0,\ i=1,\cdots,m \tag{6}$$

该优化问题称为软间隔支持向量机,如果不允许有任何训练误差,即 $\boldsymbol{\xi}=0$,则算法就退化成初始定义 2,称为硬间隔支持向量机。

下面通过修改定义 2 中的约束条件和目标函数,得到一个新的 SVM 最优问题。

**定义 4**　(HartTreeSVM 硬边界最优问题)正类和负类分别为定义 1 中所定义,HartTreeSVM 的最优边界超平面问题定义如下

$$\min_{\boldsymbol{w}\in H,b\in\mathbf{R}} \text{imize}\ \ \tau(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}\|^2-\sum_{i\in l_k}y_i(\langle \boldsymbol{x},\boldsymbol{w}\rangle+b) \tag{7}$$

$$\text{s.t.}\ \ y_i(\langle \boldsymbol{x}_i,\boldsymbol{w}\rangle+b)\geqslant 1,\ i\in l_k \tag{8}$$

式中,$k=1$,$\bar{k}=2$ 或者 $k=2$,$\bar{k}=1$。

这里本文将类 $k$ 定义为 HC,将类 $\bar{k}$ 定义为 NHC,则由定义 4 定义与 SVM 初始最优问题的区别可以看出,前者将约束条件该为 $y_i(\langle \boldsymbol{x}_i,\boldsymbol{w}\rangle+b)\geqslant 1$,$i\in l_k$,则其最优问题的可行解求解能够在不利用松弛变量的前提下,以最大的边界准确划分所有属于硬类 $k$ 的样本。另一方面,在目标函数中增加一个分量,表示如果非硬类 $\bar{k}$ 的样本距离所求超平面越远,则目标函数值越小,确保能找到的超平面以尽量小的误差划分类 $\bar{k}$ 的样本。

## 2　HardTreeSVM 算法描述

该算法的目的是构造一棵带有 SVM 节点的树。算法的每一步都有一个由超平面划分出的区域被标记为非硬类,直到整个空间被标记完毕。下面通过对实例的分析来进一步描述该算法。

图 1 中的样本为 LIBSVM c++库中的 Forclass 数据

集[14]，其中每个样本包含两个特征，所以方便以二维平面的形式表现数据。

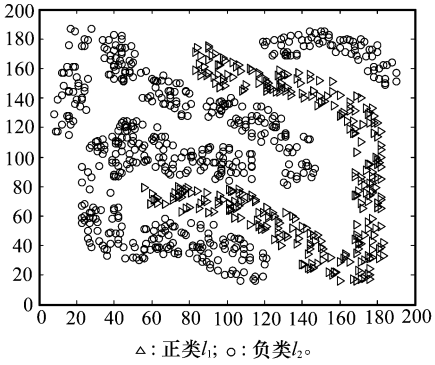



图 1 数据集 Fourclass 样本

从图 2 中可以看出，利用基于径向基核函数(radial basis function，RBF)的非线性 SVM 方法对该样本进行分类，可以得到一个较清晰的非线性解。在转换空间得到的分类超平面用实线表示，得到的两类边界用虚线表示。边界上的点为分类所需的支持向量。很明显，训练样本的数据量越大，分类样本点所需的时间就会越长。

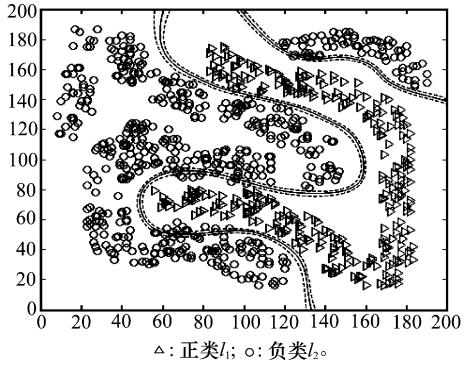



图 2 非线性 SVM 分类结果

与非线性 SVM 分类方法不同的是，本文通过构建一个决策树来实现对一个数据集的快速分类，如图 3 所示。决策树中的节点是由线性 SVM 训练得到的超平面。

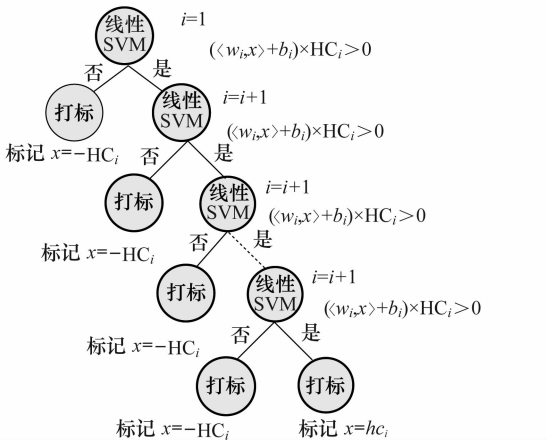



图 3 线性 SVM 决策树

在构建树的过程中，每一步都对事先定义的硬类 $l_k$ 进行选择，训练一个 SVM 直到获得的超平面可以正确地区分所有属于类 $l_k$ 的样本点，因此所有样本 $x_i$，$i\in l_k$ 都将位于超平面的一侧，并且在超平面另一侧的样本全部属于非硬类 $l_{\bar{k}}$。这样在每一步中可以通过删除属于类 $l_{\bar{k}}$ 的训练样本(已被正确分类)来减少下一步需要划分的样本数量。重复此过程，直到剩余的所有样本都属于同一个类别。

以 Fourclass 样本为例，假设正类 $l_1$(三角形)为硬类，使用该决策树的方法对其进行训练，图 4 为第一步中(对应决策树第一个节点)通过解决定义 4 得到的超平面。正如定义 4 的约束条件和目标函数所描述，在超平面右侧硬类 $l_1$ 被完全分类，而超平面左侧的所有样本都属于非硬类 $l_2$。在第一个节点划分之后，那些被打标为非硬类 $l_2$ 的样本被删除，将剩下没有完全分类的训练样本作为下一节点的输入，如图 5 所示。逐步重复该过程，不断地对新约简后的样本空间进行非硬类样本的标记。

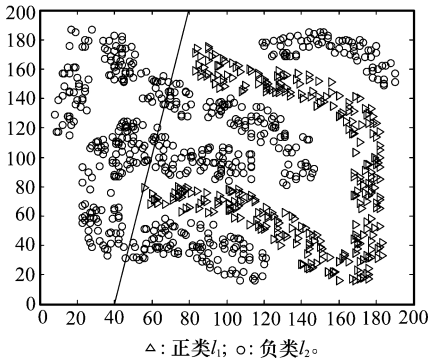



图 4 Fourclass 的第一个超平面(三角形样本为硬类)

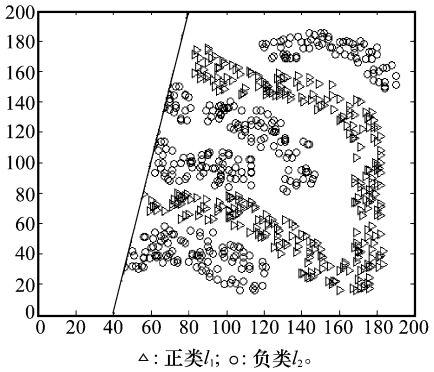
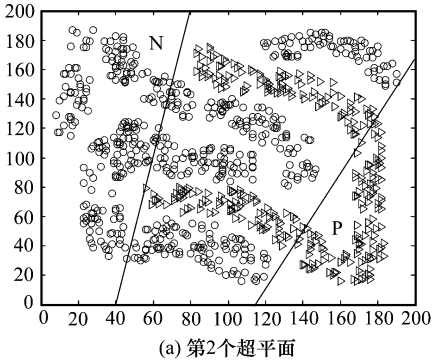


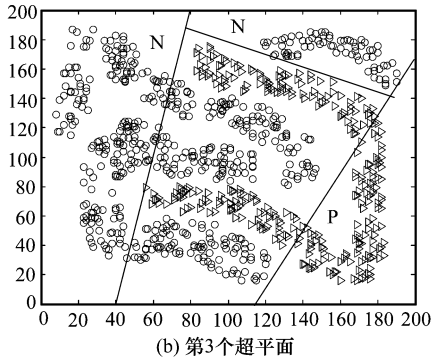

图 5 缩减过的用于下一分类步骤的样本

图 6～图 8 显示了该方法在树的每一步得到的超平面。图中的区域 N 表示将正类(三角形)作为硬类时的二次规划的解，因此所有在区域 N 中的样本都被标记为负类样本。类似地，区域 P 表示负类(圆形)作为硬类时二次规划问题的解，则所有在 P 区域的样本被标记为正样本。一个树节点得到对应的一个超平面，然后将被划分成非硬类的样本移除，并在下一个节点求解剩余样本的二次规划问题。每一步找到一个合适的超平面，在该超平面的一侧区域中

的所有样本都属于非硬类,通过此方法来逐步对特征空间进行标记。重复此过程,直到剩余样本都属于同一类别。



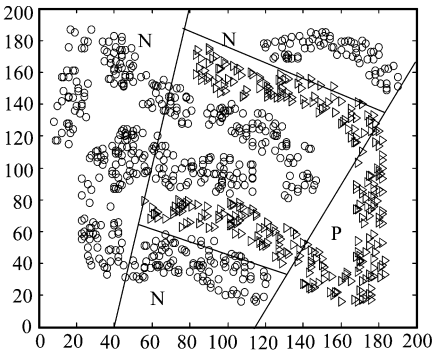

(a) 第2个超平面



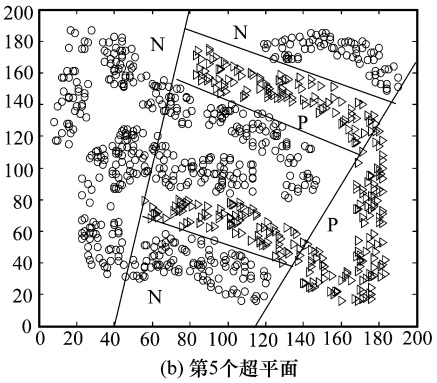

(b) 第3个超平面

△: 正类$l_1$; ○: 负类$l_2$。

图 6　第 2 个和第 3 个超平面



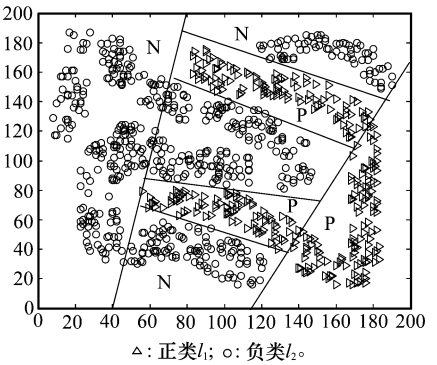

(a) 第4个超平面




(b) 第5个超平面

△: 正类$l_1$; ○: 负类$l_2$。

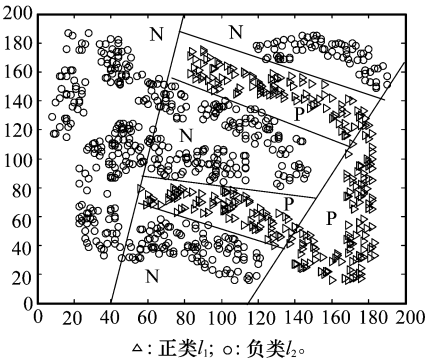
图 7　第 4 个和第 5 个超平面




△: 正类$l_1$; ○: 负类$l_2$。

图 8　第 6 个超平面

图 9 表示 Fourclass 数据集作为该算法的最终解,以及整个特征空间根据决策树所划分的情况。




△: 正类$l_1$; ○: 负类$l_2$。

图 9　Fourclass 样本的最终解

## 3　增强 HardTreeSVM 算法收敛的方法

为确保算法的收敛,以及针对训练中可能出现的超平面失效和冗余的问题,提出几点解决方法。

### 3.1　增加树节点超平面的选择

从上述描述可知,定义 4 的目的是使得到的超平面能够完全划分定义的硬类样本,并以尽量大的边界划分非硬类样本。同样,如果将定义 3(C-SV 分类器对偶问题)中定义的属于硬类样本的惩罚值 $C_k$,($k$=H C)设置足够大,增加目标函数对硬类样本误分的损失,则求解的超平面也可以起到类似定义 4 的作用。

如果想进一步确保算法的收敛,可以在训练每一个树节点时,同时用定义 4 和定义 3 两种方法求解当 H C=1 或 H C=2 时的解,这样一次可以得到 4 个超平面,然后选择能够最大程度约简非硬类样本的超平面。

### 3.2　改变正交向量 $w$ 的符号

有时给定的超平面不能起到约简问题的作用,如图 10 所示。在这种情况下,没有非硬类样本被删除,那么这时可以改变 $w$ 的方向,以 $-w$ 代替重新进行求解。图 10 中显示向量 $w$ 的方向指向正类(事先定义,三角形表示),使用该参数得到的超平面(虚线表示)没能够划分出任何非硬类的样本(正类,三角形表示)。通过改变 $w$ 的方向后得到的超平面能够实现对问题的约简,从而进入下一步迭代。

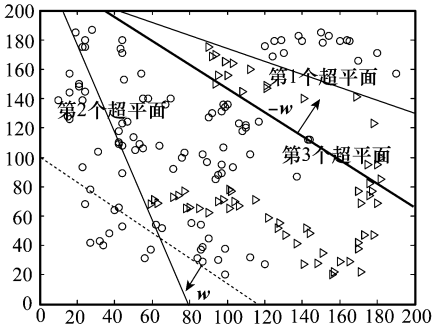



图 10 $w$ 方向改变示意图

### 3.3 垂直超平面

如果改变正交向量的方向后，仍然无法划分出非硬类样本，如图 11 所示，向量 $w$ 的方向指向正类，而利用参数 $w$ 得到的超平面（虚线表示）不能正确地划分任何属于非硬类的样本。在这种情况下，可以通过求解原超平面的垂直超平面（矢量 $w'$）对训练样本进行划分。通过这种方式可以提高算法的分类效率和泛化能力。由于在实际应用中，通常只有在特殊情况才需要改变超平面的方向，所以并不会对算法的时间复杂度产生很大影响。

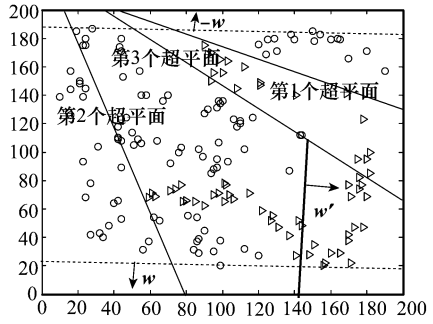



图 11 垂直超平面示意图

另一种思路是：在每次求解节点 SVM 超平面时，同时得到其正交超平面，然后选择能够划分非硬类样本数量最多的一个超平面。

### 3.4 修剪冗余超平面

HardTreeSVM 算法在没有剩余样本后终止。在算法结束后，由于后面产生节点的超平面比前面的超平面的泛化性能更强，则这些超平面就是冗余的，应该删除；或者如果在决策树中更深层的超平面与之前被使用过的超平面相同，则这些超平面的对应节点也应该予以修剪。由此可以减小决策树的规模，相应地减少了分类时间。具体情况如图 12 所示，将训练后的所有超平面中的无用超平面从树中移除，提高分类方法的泛化能力。

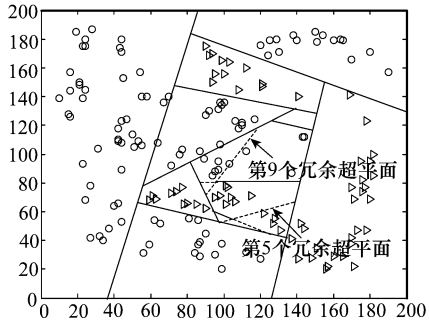



图 12 修剪冗余超平面示意图

具体的修剪过程是：从最后一个节点开始，先去掉一个指定节点，然后通过对训练集进行分类，计算分类器的分类精度；如果精度没有下降，则移除该节点。以此类推，完成整个树的冗余超平面的删减。

## 4 实验分析

为了验证本章算法对较大规模数据集和多特征属性数据的分类效率和准确率，选择了美国邮政服务（United States postal service, USPS）和 Isolet 数据集[15-16]进行训练和分类，同样对所使用的 3 种方法的实验结果进行横向比较。训练样本和测试样本随机抽取产生。

实验结果如表 1 和表 2 所示，实验使用 min-max 标准化方法对两组训练样本进行标准化，然后分别使用经典的基于 RBF 核函数的非线性 SVM、基于定义 4 的 HardTreeSVM 方法以及加入增强收敛性方法后的 HardTreeSVM 共 3 种方法对训练样本进行分类，表中最后两个属性分别是非线性 SVM 实验结果与后两个方法结果的比值，如表所示。

**表 1 min-max 标准化的 USPS 样本分类结果**

| USPS(min-max) | SVM(RBF 核) | HardTreeSVM | HardTreeSVM(增强型) | RBF/HardTreeSVM | RBF/增强型 HardTreeSVM |
|---|---|---|---|---|---|
| 样本特征数 | 256 | 256 | 256 | — | — |
| 训练样本数 | 18 060 | 18 060 | 18 060 | — | — |
| SVs 或超平面数 | 3 574 | 47 | 42 | 76.04 | 85.09 |
| 训练时间/s | 44.56 | 22.31 | 108.3 | 1.99 | 0.41 |
| 测试样本数 | 7 294 | 7 294 | 7 294 | — | — |
| 分类准确度 | 6 891 | 6 860 | 6 926 | 1 | 0.99 |
| 分类时间/s | 118.59 | 13.38 | 15.07 | 8.86 | 7.87 |
| 分类精度 | 94.47% | 94.05% | 94.95% | 1 | 0.99 |

表 2　min-max 标准化的 Isolet 样本分类结果

| Isolet(min-max) | SVM(RBF 核) | HardTreeSVM | HardTreeSVM(增强型) | RBF/HardTreeSVM | RBF/增强型 HardTreeSVM |
|---|---|---|---|---|---|
| 样本特征数 | 617 | 617 | 617 | — | — |
| 训练样本数 | 154 647 | 154 647 | 154 647 | — | — |
| SVs 或超平面数 | 23 952 | 325 | 311 | 73.70 | 77.02 |
| 训练时间/s | 689.45 | 340.28 | 2 539.74 | 2.03 | 0.27 |
| 测试样本数 | 2 862 | 2 862 | 2 862 | — | — |
| 分类准确度 | 2 726 | 2 734 | 2 748 | 1 | 0.99 |
| 分类时间/s | 199.85 | 39.48 | 33.14 | 5.06 | 6.03 |
| 分类精度 | 95.25% | 95.53% | 96.02% | 1 | 0.99 |

从上面实验结果可以看出,在大规模多特征数据集中,采用本章的 HardTreeSVM 和增强型 HardTreeSVM 算法对两组样本进行训练得到的超平面数要远小于基于 RBF 核函数的 SVM 方法得到的支持向量数。虽然增强型算法增加了启发式方法来避免出现分类无效的情况,减少了决策树超平面数量,但也使得训练时间有所增加。不过在上述实验结果中,非线性 SVM(RBF 核)方法的训练时间中也并未包含参数调整的时间。从整体上看,前两者的分类精度比传统 SVM 方法略有提高,特别是分类速度要快 5～6 倍。同时,从算法在两个数据集上的分类性能表现来看,该方法具有较好的泛化能力。

## 5　结　论

本文提出一种基于决策树的快速 SVM 分类方法。相对于传统的非线性 SVM 方法,该方法对于大规模的数据具有更快的分类速度,并得到较好的分类结果;该方法能够将非线性优化问题分解成多个线性 SVM 问题求解,避免了传统方法调整核函数参数的过程,方法简单,易于实现。进一步要研究的问题是改进决策树模型,提高算法的训练速度和分类精度,使该方法最终能更好的适用于实时性要求较高的工程应用中。

## 参考文献:


[1] Vapnik V N. *Statistical learning theory*[M]. New York: Wiley,1998.

[2] Zheng L G, Zhou H, Wang C L, et al. Combing support vector regression and ant colony optimization to reduce $NO_x$ emissions in coal-fired utility[J]. *Energy and Fuels*,2008,22(2):1034 - 1040.

[3] 文益民,王耀南,吕宝粮,等. 支持向量机处理大规模问题算法综述[J]. 计算机科学,2009,36(7):20 - 25. (Wen Y M, Wang Y N, Lu B L, et al. Survey of applying support vector machines to handle large-scale problems[J]. *Computer Science*, 2009, 36(7):20 - 25.)

[4] Burges C J C, Schölkopfand B. Improving speed and accuracy of support vector learning machines[C]// *Proc. of the Advances in Neural Information Processing Systems*,1997:375 - 381.

[5] Downs T, Gates K E, Masters A. Exact simplification of support vector solutions[J]. *Machine Learning*,2001,42(2):293 - 297.

[6] Stine R, Lin H, Auslender L. Speeding up multi-class SVM evaluation by pca and feature selection[C]// *Proc. of the Society for Industry and Applied Mathematics Workshop*,2005:72 - 79.

[7] Kristin P, Bennett K P, Cristianini N, et al. Enlarging the margins in perceptron decision trees[J]. *Machine Learning*, 2000, 41(3):295 - 313.

[8] Segata N, Blanzieri E. Fast and scalable local kernel machines[J]. *Machine Learning Research*,2010,11(6):1883 - 1926.

[9] Dorff K C, Chambwe N, Srdanovic M, et al. BDVal: reproducible large-scale predictive model development and validation in high-throughput datasets[J]. *Bioinformatics*, 2010, 26(19): 2472 - 2473.

[10] Shalev S S, Srebro N. SVM optimization: inverse dependence on training set size[C]// *Proc. of the 25th Conference on Machine Learning*,2008:928 - 935.

[11] Meligy A, Al-khatib M. A grid-based distributed SVM data mining algorithm[J]. *European Journal of Scientific Research*, 2009, 27(3):313 - 321.

[12] 李青,焦李成,周伟达. 基于向量投影的支撑向量预选取[J]. 计算机学报,2005,28(2):145 - 152. (Li Q, Jiao L C, Zhou W D. Pre-extracting support vector for support vector machine based on vector projection[J]. *Chinese Journal of Computers*, 2005, 28(2):145 - 152.)

[13] 曹淑娟,刘小茂,张钧,等. 基于类中心思想的去边缘模糊支持向量机[J]. 计算机工程与应用,2006,42(22):146 - 149. (Cao S J, Liu X M, Zhang J, et al. Fuzzy support vector machine of dismissing margin based on the method of class-center[J]. *Computer Engineering and Applications*,2006,42(22):146 - 149.)

[14] Chih C C, Chih J L. A library for support vector machines[EB/OL]. [2010 - 12 - 8]. http://www.csie.ntu.edu.tw/～cjlin/libsvm/,2005.

[15] Hull J J. A database for handwritten text recognition research[J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1994,16(5):550 - 554.

[16] UCI Repository, Uci machine learning repository[EB/OL]. [2010 - 12 - 8]. http://www.ics.uci.edu/ mlearn/MLRepository.html.