

文章编号: 1001-0920(2007)07-0755-05

基于二分网格的支持向量预选取算法

唐耀华¹, 高静怀¹, 邱保志^{1,2}

(1. 西安交通大学 电子与信息工程学院, 西安 710049; 2. 郑州大学 信息工程学院, 郑州 450052)

摘要: 在 SVM 训练过程中, 二次规划问题的求解制约着 SVM 应用于大规模数据. SVM 的决策函数由邻近分类超平面的部分训练样本——支持向量决定. 基于减小训练样本数目、加快 SVM 训练过程的目的, 提出一种基于二分网格的边界样本提取方法. 数据仿真实验表明, 该方法具有边界样本提取准确、效率高、速度快、能够自适应样本分布的优点, 而且不会显著降低 SVM 分类器的性能.

关键词: 支持向量机; 二分网格; 分类; 边界样本

中图分类号: TP391.4 **文献标识码:** A

Pre-selecting support vectors based on bisection grid

TANG Yao-hua¹, GAO Jing-huai¹, QIU Bao-zhi^{1,2}

(1. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; 2. School of Information and Engineering, Zhengzhou University, Zhengzhou 450052, China. Correspondent: TANG Yao-hua, E-mail: tangsl@mail.xjtu.edu.cn)

Abstract: In training a SVM, the solution of a quadratic programming (QP) problem constrains the SVM in large scale problem. The decision function of SVM is determined by a subset of the training patterns, support vectors, which lies closer to the classification hyperplane. Motivated by reducing the training patterns, a grid-based algorithm is proposed to pre-select a subset of patterns near the decision boundary. Simulation experiments show that BS-grid efficiently extracts accurate boundary patterns without degrading the classification ability of SVM and has good adaptive capacity to pattern's density and outline.

Key words: Support vector machine; Bisection grid; Classification; Boundary pattern

1 引言

支持向量机(SVM)的训练,本质上是二次规划求解问题(QP),其算法复杂度随求解问题规模的增大呈指数增长.计算量大、训练速度慢、不适宜大规模数据问题的应用,已成为影响支持向量机发展的主要因素.为此,人们提出了许多加速支持向量机训练过程的算法,如 chunking, SMO 等 QP 问题分解算法^[1], RSVM^[2], 预提取训练样本方法等^[3-6].

分解算法解决了算法空间复杂度高的问题,但对于具有较多支持向量的大规模问题,仍存在收敛速度慢的缺陷. RSVM 方法由于采用随机提取训练样本的方式,使用的数据信息较少,但分类性能有所降低^[7]. 预提取训练样本方法是以某种距离特征为基础,提取位于分类边界附近的训练样本,对样本的分布密度、分布边界形状较为敏感,算法中的参数也

需人为设定,自适应能力较低.针对上述缺点,本文提出基于网格技术的二分网格边界向量提取算法(BS-grid),提取位于分类决策超平面附近的训练样本(称边界样本),用边界样本训练 SVM 来得到分类决策函数,加快了 SVM 的训练过程.

2 算法基本原理

支持向量机的判决函数为

$$f(x) = \text{sgn}(W \cdot X + b) = \text{sgn}\left(\sum_{i \in \text{SV}} y_i K(x_i, x) + b\right).$$

其中:SV 为支持向量, $K(\cdot, \cdot)$ 为核函数^[8].可以看出, SVM 判决函数仅与支持向量相关.但是,支持向量是对全部训练样本进行二次规划求解后得到的,因而在 SVM 训练过程中,大量时间都浪费在对非支持向量的计算上.根据 KKT 条件,支持向量分布在

收稿日期: 2006-04-08; 修回日期: 2006-06-20.

基金项目: 国家自然科学基金项目(40174032).

作者简介: 唐耀华(1977—),男,郑州人,博士生,从事模式识别、机器学习的研究;高静怀(1960—),男,陕西乾县人,教授,博士生导师,从事多维信号处理、波传播与成像等研究.

分类超平面的两侧附近^[9]. 若能够预选取一包含所有支持向量的样本子集, 则可用较少的训练样本, 在不降低 SVM 分类性能的条件下, 得到相同的分类判决函数, 并显著提高 SVM 的训练速度. SVM 的这一特征为边界样本提取方法的可行性提供了理论基础.

空间网格分割技术将矢量空间划分为相邻的多个网格, 所有训练样本均位于对应的某个网格之中. 基于网格的算法是以网格而不是样本为计算单位进行分析计算的, 因此它具有较高的计算效率^[10,11]. 同时, 变尺度的网格方法也对样本的分布密度、分布形状具有较好的自适应能力. 网格算法的这一特征为所提出算法的有效性提供了理论支持.

二分网格算法以网格为处理对象, 根据网格密度特征判定是否存在边界向量. 以两类分类问题为例, 给定样本集 $\{(x_1, y_1), \dots, (x_n, y_n)\}$, $x_i \in R^d$, $y_i \in \{-1, 1\}$. 网格密度定义为

$$\text{Den} = (m - n_{-1}) / (m + n_{-1}).$$

其中: m 表示网格中属于类别 1 的样本数目, n_{-1} 表示网格中属于类别 -1 的样本数目.

边界样本存在于以下两种情况:

1) 若两相邻网格 A 与 B 有 $\text{Den}_A \times \text{Den}_B = -1$, 即网格内样本均同类(但网格间相比较, 样本异类), 则边界样本位于两网格间的边界附近, 如图 1(a) 中网格 A 和 B 所示.

2) 若 $|\text{Den}| = 1$, 即 m 和 n_{-1} 均不为零, 网格内存在不同类别样本, 则网格内存在边界样本, 如图 1(b) 中网格 A 所示.

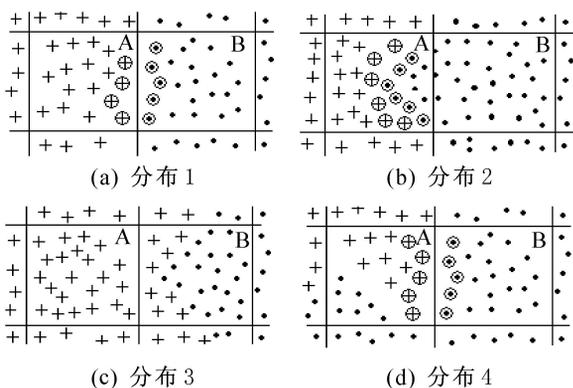


图 1 二分网格后子网格 A 的 4 种形式(子网格 B 为 A 的相邻网络)

基于上述两种边界样本存在情况, 二分网格法通过逐级二分含有异类样本的网格来生成新的子网格, 依据子网格密度特征在不同网格尺度下提取边界样本. 算法基本步骤如下:

Step1: 样本数据归一化. 整个样本空间视为一个超立方体网格, 并作为当前网格, 将训练样本投影

到网格内的相应位置.

Step2: 针对当前网格, 二分每个坐标轴, 生成 2^d (d 为样本维数) 个等体积、两两相邻的子网格, 并计算子网格密度特征.

Step3: 依据网格密度, 判定并提取每个子网格内的边界样本. 以二维样本为例作图, 详细说明如下:

1) 依次提取两个子网格 A 和 B, 若其网格密度满足 $\text{Den}_A \times \text{Den}_B = -1$, 即网格 A 中均为同类样本, 其相邻网格 B 中均为另一类样本, 如图 1(a) 所示. 边界样本位于网格边界附近, 基于 Euclidean 距离提取网格 A 和 B 中距离最近的多个 (K -near) 样本作为边界样本.

2) 对于剩余的子网格, 若某个子网格 A 满足 $|\text{Den}_A| = 1$, 即网格 A 内部样本是同类的, 如图 1(c) 所示, 则此时网格 A 中不含边界样本.

3) 剩余的子网格内部均含有异类样本, 如图 1(b) 所示. 此时边界样本位于其内部, 不能直接提取. 令网格 A 为异类网格, 等待参与下一次二分网格操作, 在新生成的子网格中继续进行边界样本的判断和提取. 作为补充, 设定网格二分的停止标准: 当网格内含有异类样本且样本数目小于阈值 N -least 时, 网格内所有样本均为边界样本.

Step4: 更新并排列异类样本网格. 令第 1 个网格为当前网格, 转至 Step2, 继续二分网格操作, 提取该网格中的边界样本, 直至没有网格需继续被二分时停止.

Step5: 旋转坐标系, 将原始样本投影至新的坐标系中. 重复 Step2 ~ Step4, 再次提取边界样本. 两次提取的所有边界样本即为所求.

注 1 实际提取过程中会遇到一类特殊情形(见图 1(d)), 网格 A 中含有异类样本, 如果仍直接对网格 A 二分, 则图 1(d) 中所标示出的边界样本会被漏选. 为此增加了 Step5, 用旋转坐标系二次提取边界向量, 使得提取效果有较大改善. 限于篇幅, 这里不对旋转坐标系的构造进行讨论.

3 仿真实验

为验证所提出方法的有效性, 本文针对人工合成数据和真实数据集作仿真实验. 实验在 Pentium 2.4 MHz, 512 MB RAM PC 机上进行, SVM 训练结果是基于 Steve Gunn's SVM Matlab Toolbox^[12] 得到的.

3.1 高斯分布合成数据

算法参数设置如下: BS-grid 算法参数: K -near = 10, N -least = 5; 支持向量机参数: 松弛因子 $C = 0.2$, 核函数选用高斯核函数, 其宽度参数 $\gamma = 1$.

第 1 个实验是曾在文献[5]中使用过的异或问题(XOR). 训练样本是随机产生的服从高斯分布的两类样本,其分布形式为

$$C_1 = \{x / x \quad C_{1A} \quad C_{1B}\}, \quad (1)$$

$$C_2 = \{x / x \quad C_{2A} \quad C_{2B}\}. \quad (2)$$

其中

$$C_{1A} = \left\{ x / N \left[\begin{matrix} 1 \\ 1 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\},$$

$$C_{1B} = \left\{ x / N \left[\begin{matrix} -1 \\ -1 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\},$$

$$C_{2A} = \left\{ x / N \left[\begin{matrix} -1 \\ 1 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\},$$

$$C_{2B} = \left\{ x / N \left[\begin{matrix} 1 \\ -1 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\}.$$

图 2 是针对 XOR 数据集的仿真实验结果. 其中:图 2(a) 是原始两类训练样本分布,样本数为 600;图 2(b) 是应用 BS-grid 算法提取的边界样本分布(边界样本数为 152(25.3%) ,BS-grid 运行时间为 0.309 s);图 2(c) 是由原始训练样本训练得到的 SVM(简称 NSVM) 分类超平面;图 2(d) 是由边界样本训练得到的 SVM(简称 SSVM) 分类超平面. 直接应用训练样本训练 SVM,38.7% 的样本成为支持向量, SVM 训练时间为 60.3 s. 而应用 BS-grid 算法后仅 25.3% 的训练样本被提取为边界样本,其中 99.3% 的边界样本在训练 SVM 后成为支持向量, SVM 的训练时间缩短至 1.1 s. 从实验结果看,应用 BS-grid 算法提取边界样本后,几乎所有的边界样本均成为支持向量,从而提高了 SVM 训练中的数据利用率,训练速度也由于训练样本的减少得以大幅度提高.

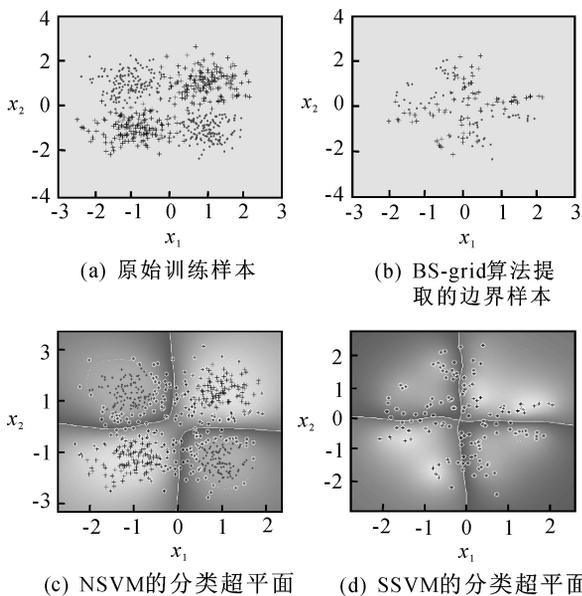


图 2 XOR数据集实验结果

从图 2(c) 和图 2(d) 可以看出, SSVM 与 NSVM 具有相同的分类效果,甚至 SSVM 分类超平面在原点附近对异或问题的表征更为准确. 但是, SSVM 分类超平面的平滑性不如 NSVM 分类超平面. 这是由于 SVM 是基于数据驱动的分类方法,尽管 SVM 适宜于小样本分类问题,但减少训练样本数量仍会对 SVM 的局部泛化性能产生影响,在分类边界区域依赖于少量训练样本的特征而变得不平滑. 通过调整松弛因子 C 可在一定程度内改善 SVM 的泛化性能. 其次, NSVM 中的支持向量来源于靠近分类超平面的样本和四周边界位置的部分样本,而 SSVM 仅使用了最靠近分类边界的部分样本. 因此, SSVM 中由于缺少部分支持向量而使得 SVM 局部分类性能受到影响.

为检验 BS-grid 算法对训练样本分布密度的自适应能力,构造一变形异或问题(CXOR)数据集,将 XOR 数据分布区域稍作修改,产生数据交叠和相离并存的分布形式,样本分布如图 3(a) 所示(样本数为 800). C_{1A} , C_{1B} , C_{2A} 和 C_{2B} 修改为

$$C_{1A} = \left\{ x / N \left[\begin{matrix} 1 \\ 2 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\},$$

$$C_{1B} = \left\{ x / N \left[\begin{matrix} -1 \\ -0.8 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\},$$

$$C_{2A} = \left\{ x / N \left[\begin{matrix} -1 \\ 0.8 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\},$$

$$C_{2B} = \left\{ x / N \left[\begin{matrix} 1 \\ -2 \end{matrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 0.5^2 \end{bmatrix} \right] \right\}.$$

图 3 是针对 CXOR 数据集的实验结果. 其中:图 3(a) 是原始训练样本分布,样本数增至 800;图 3(b) 是应用 BS-grid 算法提取的边界样本分布(边界样本

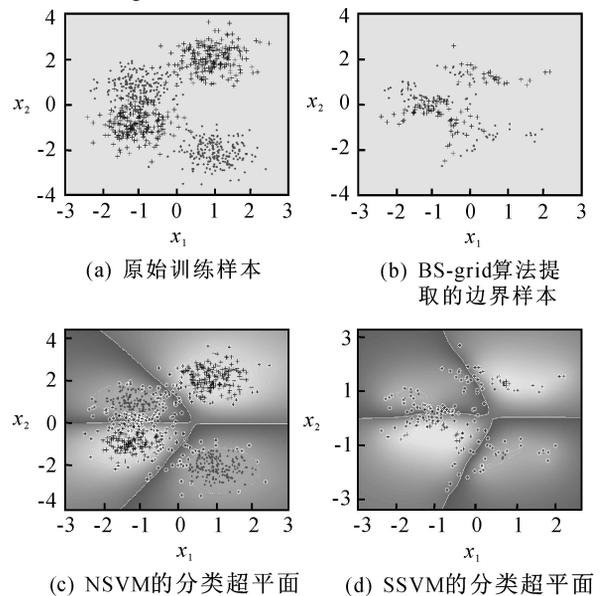


图 3 CXOR数据集实验结果

数为 199 (24.9%) ,BS-grid 运行时间为 0.463 s) .可以看出,BS-grid 算法能够根据样本的分布密度特征自适应地提取训练样本:在样本交叠区域提取较多的训练样本作为边界样本,有助于增强 SVM 的泛化性能,达到较好的分类效果;在样本相离区域提取最靠近边界的少量样本,以尽可能地加快 SVM 的训练速度.从边界样本提取和加速 SVM 训练过程看,应用 BS-grid 算法,199 个原始样本被提取为边界样本.其中 175 个边界样本在训练 SVM 后成为支持向量,少于应用原始样本直接训练 SVM 所得到的支持向量数(241 个).SVM 训练时间也由 198.6 s 缩短至 2.3 s.从图 3(c) (支持向量数为 241 (30.1%) ,SVM 运行时间为 198.6 s) 和图 3(d) (支持向量数为 175 (87.9%) ,SVM 运行时间为 2.3 s) 可以看出,NSVM 与 SSVM 分类效果完全相同.因此,BS-grid 算法对样本分布具有较高的适应能力,能够在不降低分类性能的基础上加快 SVM 训练速度.然而,存在的问题仍然是 SSVM 分类超平面的平滑性有所降低.

为检验 BS-grid 算法对 SVM 分类性能的影响,采用 10-重交叉验证法分别对上述两个问题进行实验.随机各生成 10 组数据集,每次利用一组数据集作为训练样本生成 SVM 分类决策函数,其他 9 组数据集作为检验样本计算分类正确率,检验 SSVM 的分类性能.检验正确率的平均值及相关数据如表 1 所示.实验表明,BS-grid 算法未明显降低 SVM 分类性能.

表 1 BS-grid 算法对 SVM 分类性能的影响

数据集类型	是否使用 BS-grid 算法 (占训练样本百分比)	支持向量数	运行时 /s	检验正确率 / %
异或数据	使用	201.9 (98.84 %)	3.47	94.74
异或数据	未使用	162.8 (20.38 %)	191.02	95.51
变形异或数据	使用	187.1 (97.89 %)	3.03	95.21
变形异或数据	未使用	304.8 (38.12 %)	199.73	96.33

为检验 SVM 分类性能对 BS-grid 算法中参数设置的敏感性,基于上述异或问题和变形异或问题,采用 10-重交叉验证法,对 BS-grid 算法中参数 K -near 和 N -least 的取值与提取边界样本数目和 SSVM 分类正确率的关系进行实验,结果如表 2 所示.表中数据为 20 组数据在不同参数下运算得到的平均值,上方数据是提取的边界样本数,下方数据是分类检验正确率.实验表明,当参数 K -near 和 N -least 的取值较小时,提取的边界向量少,对应的分类正确率相对较低.随着 K -near 和 N -least 取值的增大,提取的边界向量随之增多,对应的分类正确

率也有所提高.参数 K -near 对提取边界向量的数量和分类正确率的影响较 N -least 大.从总体上看,SVM 分类性能对 BS-grid 算法中参数设置不敏感.

表 2 BS-grid 算法参数设置对 SVM 分类性能的影响

N -least	K -near			
	3	5	10	15
3	144.2	161.6	188.7	205.9
	91.75 %	94.3 %	94.82 %	94.83 %
5	147.1	164.1	191.3	205.5
	92.33 %	94.51 %	95.21 %	95.26 %
10	170.5	183.3	209.5	226.5
	91.69 %	93.99 %	95.40 %	95.57 %
15	195.0	204.0	227.1	243.9
	93.43 %	94.71 %	95.49 %	95.64 %

3.2 真实数据集

fourclass 是人工合成用于 4 类分类问题的二维数据集^[13],经变换后转换为 2 类分类问题数据,样本数为 862,分布形式如图 4(a) 所示.本实验的目的是检验 BS-grid 算法对训练样本分布形状的适应能力.实验参数设置如下: K -near = 3, N -least = 5, SVM 核函数为高斯核函数, $\gamma = 1$, 松弛因子 $C = 10$.图 4(c) (边界样本数为 127 (15%)) 和图 4(d) (支持向量数为 97 (77.6%) ,SVM 运行时间为 0.8 s) 分别为 BS-grid 算法提取的边界样本分布和 SSVM 的分类超平面.可以看出,由边界样本刻画的分界形状清晰正确,训练得到的 SSVM 分类超平面完全正确,表明 BS-grid 算法对具有复杂分布形状的训练样本仍能较好地提取边界样本,在不降低 SVM 分类性能的情况下加快了训练过程.

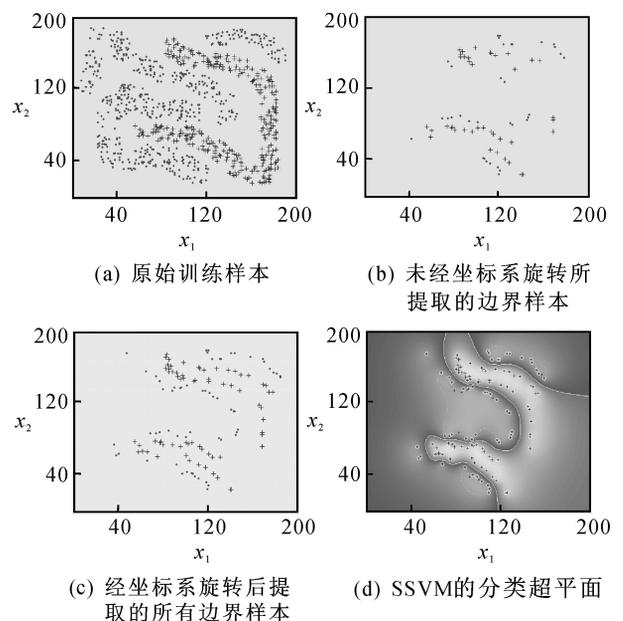


图 4 Fourclass 数据集实验结果

表 3 BS-grid 算法对 SVM 分类性能的影响

训练数据集	检验数据集	是否使用 BS-grid	支持向量数 (占训练样本百分比)	运行时间 / s	检验正确率 / %
svmguide	svmguide_t	使用	343 (36.9 %)	305.1	95.65
svmguide	svmguide_t	未使用	386 (12.5 %)	9 151.4	95.63
svmguide_t	Ssvmguide	使用	431 (34.7 %)	692.4	95.47
svmguide_t	Ssvmguide	未使用	463 (11.6 %)	19 677.7	95.53

为检验 BS-grid 算法在不同规模训练样本下运算对 SSVM 分类正确率的影响,应用 XOR 数据、CXOR 数据以及 Fourclass 数据集,通过随机提取不同规模的训练样本进行实验.从原始数据集中随机提取 500 个训练样本作为检验样本,实验结果如图 5 所示.

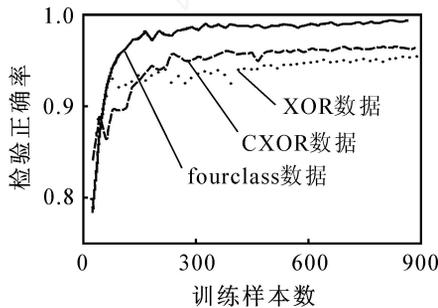


图 5 训练样本规模与检验正确率的关系

在小规模训练样本下提取边界样本来训练 SVM 依然能获得较高的检验正确率,从而印证了以下事实:

- 1) BS-grid 算法对小样本边界向量提取准确有效;
- 2) SVM 具有良好的泛化性能,适用于小样本分类问题.

第 2 个真实数据集为 4 维样本数据集 svmguide^[14],它包含两部分数据:svmguide 中含有 3 089 个样本,svmguide_t 中含有 4 000 个样本.本实验用于验证 BS-grid 算法针对 4 维数据的有效性.利用这两部分数据集进行交叉验证,实验结果如表 3 所示.实验表明,当样本维数增加时,BS-grid 算法对样本的提取效率有所降低,但边界样本的提取仍较为准确,SVM 的分类性能无明显降低.

3.3 算法时间复杂性分析

BS-grid 算法的运行时间主要由以下 3 部分组成:1) 二分网格,计算网格密度的时间;2) 二分网格过程中出现二相邻异类网格的次数;3) 针对二相邻异类网格,提取 K -near 个距离最近的异类样本的时间.BS-grid 算法时间复杂度为 $O(dN + dlmn)$.其中: N 为样本数, d 为样本维数, l 为需要计算二相邻

异类网格中最近邻样本次数, m 和 n 分别为二相邻异类网格中的样本数.一般情况下, l 比较小, $m \ll N, n \ll N$,实际 BS-grid 算法的时间复杂度稍大于 $O(dN)$.鉴于网格算法所具备的高速性能,BS-grid 算法适用于大规模数据集.

4 结 论

本文基于 SVM 分类超平面的支持向量位于分类边界附近这一事实,结合网格算法速度快的优点,提出将二分网格应用于边界样本提取的 BS-grid 算法,加快了 SVM 训练速度.该算法具有以下优点:

- 1) 能够在不显著降低 SVM 分类性能的前提下,提取较少的边界样本,加快 SVM 的训练速度;
- 2) 能够自适应具有不同分布密度、不同分布形状的训练样本;
- 3) 算法时间复杂度稍大于 $O(dN)$,适用于大规模数据集;
- 4) SVM 分类性能对算法中的参数选择敏感性较低.

然而,由于二分向量空间所生成的子网格数随样本维数的增大呈指数增长,对于高维数据,BS-grid 算法将失去快速、高效率提取边界样本的优点.因此 BS-grid 算法主要适用于低维数据分类问题,尤其对于样本规模大、分布形式复杂的问题具有较强的优势.

参考文献(References)

- [1] Platt J C. Advances in kernel methods —Support vector learning[M]. Cambridge: MIT Press, 1998:185-208.
- [2] Lee Y J, Mangasarian O L. RSVM: Reduced support vector machines[C]. Proc of 1st SIAM Int Conf on Data Mining. Chicago:SIAM, 2001:350-366.
- [3] Wang J G, Neskovic P, Cooper L N. Training data selection for support vector machines[C]. Proc of 1st Int Conf in Natural Computation. Berlin: Springer, 2005: 554-564.
- [4] Abe S, Inoue T. Fast training of support vector machines by extracting boundary data[J]. Lecture Notes in Computer Science, 2001:308-313.

(下转第 764 页)

由 Step2 得到下面的解:

$$A_{c1} = \begin{bmatrix} -5.6579 & 2.8343 \\ -10.3763 & -0.6574 \end{bmatrix},$$

$$A_{c2} = \begin{bmatrix} -4.2006 & 1.9203 \\ -6.8530 & -1.6040 \end{bmatrix},$$

$$C_{c1} = [-3.0349 \quad 0.3778],$$

$$C_{c2} = [-3.0693 \quad 1.2813],$$

$$B_{c1} = \begin{bmatrix} -2.0539 & -0.6075 \\ -1.0016 & -1.9030 \end{bmatrix},$$

$$B_{c2} = \begin{bmatrix} -1.5396 & -0.7656 \\ -1.2339 & -2.1270 \end{bmatrix}.$$

5 结 语

利用 Lyapunov 稳定性理论和 LMI 技术,给出了 T-S 模糊系统的一个有效的输出反馈耗散控制器的设计方法. 该方法充分考虑了模糊子系统之间的相互作用. 仿真例子说明了所提出的方法是可行的.

参考文献 (References)

- [1] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control[J]. IEEE Trans on SMC, 1985, 15(1): 116-132.
- [2] Wang Y, Zhang Q L, Sun Z Q, et al. Analysis and design of discrete fuzzy system with fuzzy Lyapunov approach[J]. Acta Automatica Sinica, 2004, 30(2): 255-260.
- [3] Liu X D, Zhang Q L. Approaches to quadratic stability conditions and H control designs for T-S fuzzy systems [J]. IEEE Trans on Fuzzy Systems, 2003, 11(6): 830-839.
- [4] Chen B S, Tseng C S, Uang H J. Mixed H_2/H fuzzy output feedback control design for nonlinear dynamic systems: An LMI approach[J]. IEEE Trans on Fuzzy Systems, 2000, 8(2): 249-265.
- [5] Sing Kiong Nguang, Peng Shi. H fuzzy output feedback control design for nonlinear systems: An LMI approach[J]. IEEE Trans on Fuzzy Systems, 2003, 11(3): 331-340.
- [6] Haddad W M, Bernstein D S, Wang Y W. Dissipative H_2/H controller synthesis [J]. IEEE Trans on Automatic Control, 1994, 39(4): 827-831.
- [7] Trentelman H L, Willems J C. Synthesis of dissipative systems using quadratic differential forms: Part II[J]. IEEE Trans on Automatic Control, 2002, 47(1): 70-86.
- [8] Willems J C, Trentelman H L. Synthesis of dissipative systems using quadratic differential forms: Part I[J]. IEEE Trans on Automatic Control, 2002, 47(1): 53-69.
- [9] Khalil H K. Nonlinear systems[M]. NJ: Prentice-Hall, 1996.
- [10] Slotine J E, Li W. Applied nonlinear control[M]. NJ: Prentice-Hall, 1991.
- [11] Huey-Jian Uang. On the dissipativity of nonlinear systems: Fuzzy control approach[J]. Fuzzy Sets and Systems, 2005, 156(2): 185-207.
- [5] Shin H J, Cho S Z. Fast pattern selection for support vector classifiers[C]. Proc of 7th Pacific-Asia Conf on Knowledge Discovery and Data Mining. Berlin: Springer, 2003: 376-387.
- [6] Koggalage R, Halgamuge S. Reducing the number of training samples for fast support vector machine classification[J]. Neural Information Processing-letters and Reviews, 2004, 2(3): 57-65.
- [7] Lin K M, Lin C J. A study on reduced support vector machines[J]. IEEE Trans on Neural Networks, 2003, 14(6): 1449-1459.
- [8] Vapnik V N. The nature of statistical learning theory [M]. New York: Springer-Verlag, 2000.
- [9] 张莉. 支撑向量机与核方法研究[D]. 西安: 西安电子科技大学, 2002. (Zhang Li. Research on support vector machines and kernel methods[D]. Xi'an: Xidian University, 2002.)
- [10] Han J, Kamber M. Data mining: Concepts and techniques [M]. San Francisco: Morgan Kaufmann, 2000.
- [11] Eden W M Ma, Tommy Chow W S. A new shifting grid clustering algorithm [J]. Pattern Recognition, 2004, 37(3): 503-514.
- [12] Gunn S R. Support vector machines for classification and regression [R]. Southampton: School of Electronics and Computer Science, University of Southampton, 1998.
- [13] Ho T K, Kleinberg E K. Building projectable classifiers of arbitrary complexity[C]. Proc of 13th Int Conf on Pattern Recognition. Vienna, 1996: 880-885.
- [14] Hsu C W, Chang C C, Lin C J. A practical guide to support vector classification [R]. Taipei: National Taiwan University, 2003.

(上接第 759 页)