

文章编号: 1001-0920(2005)10-1129-04

一种快速支持向量机增量学习算法

孔锐, 张冰

(暨南大学 珠海学院 计算机科学系, 广东 珠海 519070)

摘要: 经典的支持向量机(SVM)算法在求解最优分类面时需求解一个凸二次规划问题, 当训练样本数量很多时, 算法的速度较慢, 而且一旦有新的样本加入, 所有的训练样本必须重新训练, 非常浪费时间. 为此, 提出一种新的SVM快速增量学习算法. 该算法首先选择那些可能成为支持向量的边界向量, 以减少参与训练的样本数目; 然后进行增量学习. 学习算法是一个迭代过程, 无需求解优化问题. 实验证明, 该算法不仅能保证学习机器的精度和良好的推广能力, 而且算法的学习速度比经典的SVM算法快, 可以进行增量学习.

关键词: 支持向量; 边界向量; 增量学习; 支持向量机

中图分类号: TP181

文献标识码: A

A Fast Incremental Learning Algorithm for Support Vector Machine

KONG Rui, ZHANG Bing

(Department of Computer Science of Zhuhai College of Jinan University, Zhuhai 519070, China Correspondent: KONG Rui, E-mail: tkongrui@jnu.edu.cn)

Abstract: A kind of algorithm for support vector machine (SVM) is proposed, which can train SVM fast and incrementally. The new algorithm selects border vectors which may be support vectors, so as to reduce training samples and advance training speed. Then an incremental algorithm is presented to train SVM by using the selected border vectors. Experiment results show that the algorithm not only acquires the same precision with that of the classical algorithms, but also is faster than that of the classical algorithms.

Key words: Support vectors; Border vectors; Incremental learning; Support vector machines

1 引言

支持向量机^[1](SVM)是近几年出现的一种具有良好性能的学习机器, 由于SVM具有强大的非线性处理能力和良好的推广能力而得到了广泛应用. 经典的SVM训练算法基本都需求解一个凸二次规划问题^[2,3], 所有训练样本同时参与训练, 求出最优分类面. 如果有新的样本加入, 则需要对所有训练样本重新进行训练, 非常浪费时间. 如果能将原有的知识保留起来与新加入的样本一起进行训练, 则既能继承先前所学习的知识, 又能减少由于新样本的加入而重新学习的时间, 这正符合人类的学习习惯. 实际中, 样本的采集都是不断积累的, 经常有新的样本加入是很常见的, 而且支持向量机的最优

分类面只与支持向量有关. 因此, 对SVM算法中增量学习的研究具有重要的理论意义和实用价值. 文献[4]的方法中所有样本都参与训练, 因而训练速度慢, 如果样本集非常大, 则训练无法进行. 文献[5]提出一种增量学习方法, 解决了在大样本情况下的SVM学习问题, 但其初始训练还是在全部的初始训练集上进行, 使其训练速度有所降低. 在增量学习过程中采用了原来的支持向量和所有新的错分样本作为新的SVM训练集合, 没有把虽然分类正确, 但其与分类超平面较近, 下一次可能成为错分样本或支持向量的样本包括在内, 因此其循环次数会明显增加, 这从另一方面降低了支持向量机的训练速度. 文献[6]提出了 α -ISVM支持增量学习, 训练集的获得主要从支持

收稿日期: 2004-10-29; 修回日期: 2005-02-28

作者简介: 孔锐(1964—), 男, 安徽肥西人, 副教授, 博士, 从事机器学习、统计学习理论等研究; 张冰(1965—), 女, 江苏苏州人, 副教授, 从事机器学习、数据挖掘等研究.

向量, 误分数据, 有选择地淘汰一些样本来进行的, 但该算法也存在一些不足, 如训练时需人为选择很多参数, 而确定这些参数需要相当的工作量。

本文在文献[7]的基础上, 提出一种 SVM 快速增量学习算法。该算法首先通过选择那些可能成为支持向量的边界向量, 以减少参与训练的样本数量, 提高训练速度; 然后利用边界向量进行 SVM 的增量学习。实验显示, 该算法在保证训练结果准确性的同时, 确实能提高 SVM 训练速度和进行增量学习。

2 提取边界向量

利用 SVM 进行两类分类的关键是求出最优分类面。因为最优分类面只由支持向量确定, 所以只需求出支持向量便可求出最优分类面。并不是所有的训练样本都可能成为支持向量, 只有那些在两类边界上的向量才有可能成为支持向量。因此, 为了提高算法的速度, 可首先对训练样本进行选择, 选择那些可能成为支持向量的边界样本, 并用这些边界向量训练 SVM, 这样便可在保证算法精度的同时, 减少参与训练的样本数, 从而提高算法的速度。边界向量不一定是支持向量, 但支持向量一定是边界向量。在两类分类问题中, 两类的类内样本集都是凸集^[3], 所以可首先计算出各类的类中心; 然后寻找边界点, 也就是两类之间相邻的边界向量。

若训练样本为 $(x_1^+, x_2^+, \dots, x_{N_+}^+)$, $(x_1^-, x_2^-, \dots, x_{N_-}^-)$, 且 $n = N_+ + N_-$, n 为训练样本总数, N_+ 和 N_- 分别为正样本和负样本的总数。

1) 首先计算各类的类中心向量。为了对原空间中线性不可分的样本向量进行分类, SVM 通过核函数将原空间的向量经过一个非线性映射

$$\phi_{R^d} \rightarrow H, x \rightarrow \phi(x),$$

将原空间的数据映射到一个高维的特征空间 H (维数可以无穷大) 中, 使得两类样本变得线性可分 (或近似线性可分), 这时两类训练样本变为

$$\phi(x_1^+), \phi(x_2^+), \dots, \phi(x_{N_+}^+),$$

$$\phi(x_1^-), \phi(x_2^-), \dots, \phi(x_{N_-}^-),$$

所以两类的类中心

$$\begin{aligned} c_+^\phi &= \frac{1}{N_+} \sum_{i=1}^{N_+} \phi(x_i^+), \\ c_-^\phi &= \frac{1}{N_-} \sum_{i=1}^{N_-} \phi(x_i^-). \end{aligned} \quad (1)$$

因为非线性映射 $\phi_{R^d} \rightarrow H$ 不能明确知道, 所以采取一种变通的方法求取类中心, 即求取与类中心最近的样本点 $\phi(x_c^+)$ 和 $\phi(x_c^-)$ 代替两类的类中心。具体方法如下:

在每个类别中, 分别以每个样本为类中心, 计

算类内其他各样本点与类中心的距离, 并算出距离之和。距离之和最小的样本就是该类的类中心 c_+^ϕ 和 c_-^ϕ 。

2) 计算两类类中心之间的距离

$$D^\phi = |c_+^\phi - c_-^\phi| \quad (2)$$

3) 计算两类类中心之间连线的中点向量

$$\begin{aligned} m^\phi &= \frac{1}{2}(c_+^\phi + c_-^\phi) = \\ &= \frac{1}{2} \left(\frac{1}{N_+} \sum_{i=1}^{N_+} \phi(x_i^+) + \frac{1}{N_-} \sum_{i=1}^{N_-} \phi(x_i^-) \right). \end{aligned} \quad (3)$$

4) 计算两类样本与 m^ϕ 的距离, 将所有距离小于 $D^\phi/2$ 的样本作为边界向量

$$\begin{aligned} D_{m^\phi}^\phi &= |\phi(x_j) - m^\phi| = \\ &= \left| \phi(x_j) - \frac{1}{2} \left(\frac{1}{N_+} \sum_{i=1}^{N_+} \phi(x_i^+) + \frac{1}{N_-} \sum_{i=1}^{N_-} \phi(x_i^-) \right) \right| = \\ &= \frac{1}{2} \left| \left(\phi(x_j) - \frac{1}{N_+} \sum_{i=1}^{N_+} \phi(x_i^+) \right) + \right. \\ &\quad \left. \left(\phi(x_j) - \frac{1}{N_-} \sum_{i=1}^{N_-} \phi(x_i^-) \right) \right| = \\ &= \frac{1}{2} (|\phi(x_j) - c_+^\phi| + |\phi(x_j) - c_-^\phi|), \end{aligned} \quad (4)$$

$$\text{边界向量} = \{\phi(x_j) \mid D_{m^\phi}^\phi \leq D^\phi/2\}. \quad (5)$$

即所有与两个类中心距离小于域值的样本都是边界向量。对应于图 1 中圆内的向量都是边界向量。

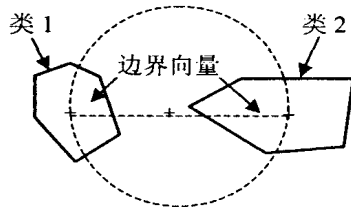


图 1 提取边界向量方法示意图

由图 1 可见, 所提取的边界向量约比原训练样本减少了 1/2, 因而算法的速度可以提高。改变边界向量的判断标准 (域值), 就可以改变所提取边界向量的数量。这种边界向量的选择方法适用于任何分布的训练样本集。

从以上提取边界向量的算法中可见, 在高维特征空间中求取类中心以及判断边界向量时, 所有的运算都是在高维的特征空间中计算两个样本之间的距离。如果采用 Mercer 核^[1,2]时, 在核空间中两个样本之间的距离计算公式^[8]为

$$\begin{aligned} \|\phi(x_i) - \phi(x_j)\|^2 &= \\ &= k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j). \end{aligned} \quad (6)$$

因此, 每计算一次距离就需要计算 3 次核函数, 如果训练样本很多, 势必运算量很大。这里采用一种新的

核函数——条件正定核函数^[8,9](简称 CPD 核)来计算特征空间中的距离

下面是一个条件正定核函数:

$$k_{\text{CPD}}(x,y)=-\|x-y\|^q+1,0< q\leq 2 \tag{7}$$

当应用式(7)的 CPD 核函数时,在核空间中两个样本之间的距离运算可简化为

$$\|\phi(x_i)-\phi(x_j)\|^2=-k_{\text{CPD}}(x_i,x_j)=\|x_i-x_j\|^q,i,j=1,\cdots,N. \tag{8}$$

比较式(6)和(8)可见,应用 CPD 核函数,在核空间中两个样本间的距离的运算非常简单.这样在特征空间中的距离计算就只需计算一次核函数,从而大大减小了运算量,提高了算法的速度.为了进一步减少高维空间中样本之间距离的计算时间,将如下核矩阵^[3](对称矩阵)进行存储:

$$K=\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{n2} \\ \vdots & \vdots & & \vdots \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix}. \tag{9}$$

这样在高维空间中任意两个样本之间的距离就是核矩阵中的一个元素(核函数的值),核矩阵中的任何一个元素都是两个不同样本之间的距离.因此,在高维空间中计算两个样本之间的距离不需要重复计算,只需查表即可,从而进一步减少了运算量,提高了速度.

3 SVM 快速增量学习算法

快速增量算法的思路如下:首先选择训练样本,本文选择那些靠近边界的样本,因为只有这些样本才可能成为支持向量(选择的目的是为了进一步减少训练样本数,从而提高算法的速度);然后利用下面的增量学习算法^[7]进行训练,找出支持向量,最终求出最优分类面.

SVM 增量学习算法是一种迭代算法,每次选择一个训练样本进行训练,具体算法如下:

因为支持向量机的分类函数可表示为核函数与支持向量的线性组合(S 表示支持向量集),即

$$f(x)=\sum_{i\in S}\alpha_iy_ik(x_i,x)+b \tag{10}$$

原问题的对偶问题可表示为

$$\min_{0\leq\alpha_i\leq C}Q=1/2\sum_{i,j}\alpha_iK_{ij}\alpha_j-\sum_i\alpha_i+b\sum_iy_i\alpha_i \tag{11}$$

这里: α_i 为对应的系数, b 为偏移量, $K_{ij}=y_iy_jk(x_i,x_j)$ 为对称正定的核矩阵, C 为惩罚系数.

对偶问题的 KKT 条件如下:

$$g_i=\frac{\partial Q}{\partial \alpha_i}=\sum_jK_{ij}\alpha_j+y_ib-1, \tag{12}$$

$$\frac{\partial Q}{\partial b}=\sum_iy_i\alpha_i=0 \tag{13}$$

KKT 条件将训练集 D 划分成 3 个部分: 支持向量集 S ($0<\alpha_i<C, g_i=0$); 误差向量集 E ($\alpha_i=C, g_i<0$); 正确分类集 R ($\alpha_i=0, g_i>0$). 如果加一个新样本 c 到训练集中,则 g_i 的变化如下:

$$\Delta g_i=K_{ic}\Delta\alpha+\sum_jK_{ij}\Delta\alpha_j+y_i\Delta b, \tag{14}$$

$$0=y_c\Delta\alpha+\sum_{j\in S}y_j\Delta\alpha_j, \tag{15}$$

定义矩阵

$$J=\begin{bmatrix} 0 & y_1 & \cdots & y_S \\ y_1 & K_{11} & \cdots & K_{1S} \\ \vdots & \vdots & & \vdots \\ y_S & K_{S1} & \cdots & K_{SS} \end{bmatrix}, \tag{16}$$

则

$$J(\Delta b,\Delta\alpha,\cdots,\Delta\alpha)^T=-(y_c,K_{1c},\cdots,K_{Sc})^T\Delta\Delta\alpha. \tag{17}$$

由此可得

$$\Delta b=\beta\Delta\alpha, \tag{18}$$

$$\Delta\alpha_i=\beta_i\Delta\alpha. \tag{19}$$

定义 $A=J^{-1}$, 则有

$$(\beta,\beta_1,\cdots,\beta_S)^T=-A(y_c,K_{1c},\cdots,K_{Sc})^T. \tag{20}$$

如果新加入的训练样本 c 是支持向量,那么 $g_c=0$,由此

$$g_c=K_{cc}\Delta\alpha+\sum_jK_{cj}(\alpha_j+\Delta\alpha_j)+y_c(b+\Delta b)-1=0, \tag{21}$$

则

$$\Delta\alpha=\frac{\sum_jK_{cj}\alpha_j+y_cb-1}{K_{cc}+\sum_jK_{cj}\beta_j+y_c\beta} \tag{22}$$

定义

$$y_c=K_{cc}+\sum_jK_{cj}\beta_j+y_c\beta, \tag{23}$$

则当加入的训练样本是支持向量时, A 的更新公式为

$$A=\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}+\frac{1}{y_c}(\beta,\beta_1,\cdots,\beta_S,1)^T(\beta,\beta_1,\cdots,\beta_S,1). \tag{24}$$

更新后

$$\alpha=(\alpha+\Delta\alpha,\alpha)^T, \tag{25}$$

$$b=b+\Delta b. \tag{26}$$

于是可利用这种迭代更新技术,不断地加入新的训练样本进行 SVM 训练.当然,如果加入的是支持向量,则按上述公式更新 α, b ; 如果加入的不是支持向量,则不用更新 α, b .

4 SVM 快速增量学习算法实验

为了验证所提方法的有效性,应用该方法进行了一系列的实验.实验结果显示,所提出的 SVM 快速增量学习算法在保证学习机器良好推广能力的同时,不仅学习速度快,而且可以进行增量学习.所有实验都在 pentium III450, 192M 内存,MA TLAB6.1 环境中进行.

4.1 多项式核函数阶次对结果的影响

为了检验算法的有效性,采用UCI数据库中的 Sonar 数据集进行测试,该数据集共有样本点 208 个,每个数据点是 60 维,其中训练集 104 个样本,测试集 104 个样本,选用多项式内积核函数

表 1 Sonar 数据实验结果

多项式阶次 d	1	2	3	4	5
误差样本数	29	15	12	16	10
支持向量	58	63	60	68	72
边界向量			82		
训练所用时间			约 25 s		
测试所用时间			约 1 s		

4.2 提取边界样本对训练速度的影响

采用了UCI数据库中部分数据集进行实验,以验证通过提取边界向量使得算法的速度有所提高. SVM 训练算法是增量学习算法,核函数选用径向基内积核函数

表 2 UCI数据库实验结果

数 据	基于所有训练样本的训练时间(s)	边界向量数	基于边界向量的训练时间/s	支持向量数	测试时间/s
IRIS	1.00	27	1.00	16	0.10
WINE	2.00	74	1.50	29	0.50
GLASS	50.00	183	45.50	129	1.00
LETTER	9 185.40	13 805	7 393.00	10 129	732.15
SHUTTLE	1 014.80	1 253	217.50	330	29.95

实验结果显示,通过提取边界向量,有效地减少了参与训练的样本数,算法的速度有了明显提高,尤其当支持向量占总训练样本数的比例较小时,本文的快速算法速度更快

4.3 快速增量学习算法与 SMO 算法的比较

在经典的优化算法中,SMO 算法^[10]的性能较好.采用MNIST数据库中的手写数字数据集进行实验,该数据集共有 60 000 个训练数据和 10 000 个测试数据,每个数据是 28×28 维向量.核函数选用 5 阶多项式核函数,且 $C = 10$

实验结果显示,本文的快速算法与 SMO 算法相比,在测试误差上差不多,支持向量数目也相差不

表 3 MNIST 数据库实验结果

数 字	SMO 方法		快速增量算法		
	测试集误差/%	支持向量	边界向量	支持向量	测试集误差/%
0	1.7	1 206	1 688	1 201	1.7
1	1.5	757	1 050	743	1.4
2	3.4	2 183	2 615	2 180	3.3
3	3.2	2 506	3 007	2 500	3.3
4	3.0	1 784	2 320	1 775	3.0
5	2.9	2 255	3 050	2 240	2.7
6	3.0	1 347	1 885	1 305	2.9
7	4.3	1 712	2 220	1 700	4.2
8	4.7	3 053	3 760	3 035	4.5
9	5.6	2 720	3 803	2 702	5.6

表 4 实验结果(MNIST 数据库一个分类器训练时间,采用 5 阶多项式核函数)

惩罚系数 C	SMO 方法训练时间/s	快速增量算法训练时间/s
10	125.480	57.721
100	147.355	73.678

多,但本文的快速算法速度比 SMO 快得多.

5 结 语

支持向量机不仅可以进行样本的分类,而且可以进行函数的回归(实值函数的逼近),求解最优分类面就是求出一个最优的判决函数,使得分类或回归时,实际风险最小.本文提出了一种快速的支持向量机增量学习算法,该学习算法在保证学习机器性能的同时,可以进行快速增量学习.算法首先通过选择那些可能成为支持向量的边界向量来减少参与训练的样本总数,从而达到提高训练速度的目的.针对不同的训练数据分布,提出了一种选择边界向量的方案,这种方案适合于任意分布的两类样本,包括两类分布不均匀以及两类样本数量相差较大的情况.为了获得最优分类面,利用选择的边界向量进行增量学习,学习过程是一个迭代过程,不要求解二次规划问题.实验证明,快速算法在保证分类精度和学习机器良好推广能力的前提下,速度比经典的二次优化算法快得多,尤其当支持向量较少时,更显出该算法的优越性.

参考文献(References)

[1] Vapnik V. N. *The Nature of Statistical Learning Theory* [M]. New York: Springer Verlag, 1995.
[2] Muller K. R., Mika S., Ratsch G., et al. An Introduction to Kernel-based Learning Algorithms [J]. *IEEE Transactions on Neural Networks*, 2001, 12(2): 181-201.

(下转第 1136 页)

- via Fast Sampling for Sampled-data Control Systems [A] *Proc of the 36th Conf on Decision and Control* [C] San Diego, 1997: 2157-2162
- [7] Yamamoto Y, Anderson B D O. Approximating Sampled-data Systems with Applications to Digital Redesign [A] *Proc of the 41st IEEE Conf on Decision and Control* [C] Las Vegas, 2002: 3724-3709
- [8] Araki M, Ito Y, Hagiwara T. Frequency Response of Sampled-data Systems [J] *Automatica*, 1996, 32 (4): 483-497.
- [9] Braslavsky J H, Middleton R H, Freudenberg J S. L_2 -Induced Norms and Frequency Gains of Sampled-data Sensitivity Operators [J] *IEEE Trans Automatic Control*, 1998, 43(2): 252-258
- [10] Anderson B D O. Controller Design: Moving from Theory to Practice [J] *IEEE Control Systems*, 1993, 13(4): 16-25
- [11] Franklin G F, Powell J D, Workman M. *Digital Control of Dynamic Systems* [M] 3rd Edition. Beijing: Tsinghua University Press, 2001
- [12] Wang G X, Liu Y W, He Z, et al. The Lifting Technique for Sampled-data Systems: Useful or Useless? [J] *Acta Automatica Sinica*, 2005, 31 (3): 491-494

(上接第 1128 页)

- [8] 吴祈宗 运筹学 [M] 北京: 机械工业出版社, 2002: 95-108
(Wu Q Z. *Operation Research* [M] Beijing: China Machine Press, 2002: 95-108)
- [9] Shamsur R, David K S. Use of Location-allocation Models in Health Service Development Planning in Developing Nations [J] *European J of Operational Research*, 2000, 123(3): 437-452
- [10] Bongartz I A. Projection Method for Norm Location-allocation Problems [J] *Mathematical Programming*, 1994, 66(3): 283-312
- [11] 赵晓煜, 汪定伟 供应链中二级分销网络的优化设计模型 [J] *管理科学学报*, 2001, 4(4): 22-26
(Zhao X Y, Wang D W. Optimization Model for Bi-level Distribution Network Design Supply Chain Management [J] *J of Management Science*, 2001, 4 (4): 22-26)
- [12] Alsuwaiyel M H. *Algorithms Design Techniques and Analysis* [M] Singapore: World scientific publishing Co Pte Ltd, 2003: 279-298

(上接第 1132 页)

- [3] Burges C J C. A tutorial on Support Vector Machines for Pattern Recognition [J] *Knowledge Discovery and Data Mining*, 1998, 2(2): 121-167.
- [4] 曾文华, 马健 一种新的支持向量机增量学习算法 [J] *厦门大学学报(自然科学版)*, 2002, 41(6): 687-691.
(Zeng W H, Ma J. A Novel Approach to Incremental SVM Learning Algorithm [J] *J of Xiamen University (Natural Science)*, 2002, 41(6): 687-691)
- [5] 李凯, 黄厚宽 支持向量机增量学习算法研究 [J] *北方交通大学学报*, 2003, 27(5): 34-37.
(Li K, Huang H K. Research on Incremental Learning Algorithm of Support Vector Machine [J] *J of Northern Jiaotong University*, 2003, 27(5): 34-37.)
- [6] 萧嵘, 王继成, 孙正兴, 等 一种 SVM 增量学习算法 α -ISVM [J] *软件学报*, 2001, 12(12): 1818-1824
(Xiao R, Wang J C, Sun Z X, et al. An Incremental SVM Learning Algorithm α -ISVM [J] *J of Software*, 2001, 12(12): 1818-1824)
- [7] Gert C, Tomaso P. Incremental and Decremental Support Vector Machine Learning [A] *Advances in Neural Information Processing Systems (NIPS * 2000)* [C] Cambridge MA: MIT Press, 2001, 13
- [8] Schölkopf B. *The Kernel Trick for Distances* [R] MSR-TR-2000-51, Microsoft Research, 2000
- [9] Smola A J. *Learning with Kernels* [D] Berlin: Technische Universität, 1998
- [10] Platt J. Fast Training of Support Vector Machines Using Sequential Minimal Optimization [A] *Advances in Kernel Methods - Support Vector Learning* [C] Cambridge, MA: MIT Press, 1999: 185-208