中国图象图形学报
Journal of Image and Graphics

基于几何思想的快速支持向量机算法

孔　锐　张　冰
（暨南大学珠海学院计算机科学系，珠海　519070）

摘　要　为了快速地进行分类，根据几何思想来训练支持向量机，提出了一种快速而简单的支持向量机训练算法——几何快速算法。由于支持向量机的最优分类面只由支持向量决定，因此只要找出两类样本中所有支持向量，那么最优分类面就可以完全确定。该新的算法根据两类样本的几何分布，先从两类样本的最近点开始；然后通过不断地寻找违反 KKT 条件的样本点来找出支持向量；最后确定最优分类面。为了验证新算法的有效性，分别利用两个公共数据库，对新算法与 SMO 算法及 DIRECTSVM 算法进行了实验对比，实验结果显示，新算法的分类精度虽与其他两个方法相当，但新算法的运算速度明显比其他两个算法快。
关键词　几何算法　支持向量　支持向量机　分类
中图法分类号：TP181　　**文献标识码**：A　　**文章编号**：1006-8961(2007)06-1064-05

A Fast Algorithm of SVM Based on Geometry

KONG Rui, ZHANG Bing
(*Department of Computer Science of Zhuhai College of Jinan University, Zhuhai* 519070)

Abstract　In the paper, based on geometry theory, a new fast iterative algorithm for support vector machine(SVM) classifier design is presented. It is known that the optimal hyper-plane of SVM is completely constructed using its support vectors. Once all support vectors of two classes are identified, the optimal hyper-plane can be determined. Based on geometric distribution of the trained sample points, the new algorithm establishes an initial candidate support vectors set by locating the two closest points of the two opposite class. The new algorithm starts from two closest points of the opposite classes to seek the support vectors accumulatively. The new algorithm continually seeks the points which are the violators of KKT condition as support vectors. At last, the new algorithm acquires all support vectors and establishes an optimal hyper-plane. To validate the new algorithm, some experiments which compare the new algorithm with the SMO algorithm and DIRECTSVM algorithm are performed. The experimental results have shown the generalization ability of the new algorithm is the same as that of SMO algorithm and DIRECTSVM algorithm. The speed of the new algorithm is superior to the other two algorithms.
Keywords　geometric algorithm, support vector, support vector machine(SVM), classification

1　引　言

近年来，支持向量机（support vector machine, SVM）[1,2]因为其具有强大的非线性处理能力和良好的推广能力，尤其是在处理非线性分类和回归问题时，所表现出的优越性能，使得其在机器学习领域获得了高度的重视，相应地，也出现了很多 SVM 的经典算法。这些经典算法都要解决一个二次规划问题，由于解决二次规划问题的计算量非常大，因此算法的应用较为困难，从 Vapnik 提出的 Chunking 算法到一系列分解算法及其改进算法，都是为了提高算法的速度，如 Osuna 等提出的分解算法[3]等，这些算法，主要利用 SVM 解的稀疏性，但需要大量的存储空间以及进行大量的矩阵运算，这些方法虽然确实提高了速度，但还是需要解决二次优化问题。

收稿日期：2005-11-23；**改回日期**：2006-12-05
第一作者简介：孔锐（1964～　），男。高级工程师，2004 年 6 月在中国科学技术大学获博士学位。主要研究方向为机器学习、基于核函数的学习理论、图像处理、模式识别。E-mail：tkongrui@jnu.edu.cn

Platt 提出的序贯最小优化(sequential minimal optimization, SMO)算法[4]利用快速迭代的方法很好地解决了 SVM 问题,而且既提高了算法的速度,又减少了在运算过程中,对机器存储量的要求。最近,机器学习的研究者们,又对支持向量机所体现出的几何原理进行了研究,并从另一个角度,即几何角度,提出了一些非常好的简单而快速的算法,如 Keerthi 等所提出的最近点算法(nearest point algorithm, NPA)及 Roobaert 提出的 DIRECTSVM 算法[6]等,实验证明,这些算法性能非常好,不仅从根本上改变了人们对 SVM 的认识,并且使得 SVM 这个复杂的学习机器很容易被人们理解和应用。本文就是受到上面两个算法思想的启发,从几何的角度提出了一种快速而简单的 SVM 算法,本文称之为“几何快速算法”。该算法根据两类样本的几何分布情况,利用迭代的方法先求出支持向量,进而求出最优分类面。在保证获得高性能 SVM 解的同时,使算法的训练速度也得到了明显的提高。

2　基于二次规划的支持向量机算法

支持向量机是在统计学习理论[2]基础上发展起来的一种新的模式分类方法。统计学习理论是 Vapnik 等人所创立的,它是一种针对小样本情况研究统计学习规律的理论。其核心思想是:通过控制学习机器的容量来实现对其推广能力的控制。在统计学习理论中是将学习问题看作是利用有限数量的观察来寻找待求的依赖关系的问题,即从给定的函数集中选择出能够最好地逼近训练器响应函数的问题。SVM 就是遵循结构风险最小化(structural risk minimization, SRM)原则的学习机器。SVM 首先是从两类分类问题提出的,其目标是求得一个线性分类超平面,以便不仅使得两类分开,而且使得两类的分开间隔最大。

设两类样本集为 $\boldsymbol{x}_i, y_i, i=1,\cdots,n; \boldsymbol{x}\in\mathbf{R}^d$, $y\in\{+1,-1\}$是类别标号,那么在一个 d 维空间中,求解最优分类面就可以通过求解以下的二次约束优化问题[1]来实现:

$$\min\left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right)\right) \tag{1}$$

$$\text{s.t}\quad y_i[(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b]-1+\xi_i\geqslant 0, i=1,2,\cdots,n \tag{2}$$

其中,$\boldsymbol{w}$ 是决定最优分类面方向的权向量;C 是对误差的惩罚系数,C 越大,惩罚越重;b 是偏移量,其用于决定最优分类面的位置。

通过求解上两式,就可以获得以下最优分类函数:

$$f(\boldsymbol{x}) = \operatorname{sgn}\left\{\sum_{i=1}^{N_S}\alpha_i y_i \boldsymbol{k}(\boldsymbol{x}_i,\boldsymbol{x}) + b\right\} \tag{3}$$

这里,松弛项 $\xi_i\geqslant 0$ 是误差,N_S(S 代表 support vector)是支持向量的个数,$0\leqslant\alpha\leqslant C$,$\boldsymbol{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)$是核函数。

当核函数是线性函数时,则所求解就是原空间中的 SVM 最优分类面;当核函数是非线性函数时,则所求解就是非线性 SVM 的解,即对于线性不可分数据,可首先通过一个非线性映射,将原空间的数据映射到一个高维的特征空间;然后在这个高维的特征空间中,求取最优的线性分类面。高维特征空间中的所有运算都是通过原空间的内积核函数来进行的,这正是 SVM 的精髓所在。由于特征空间是由内积核函数定义的,因此选定了一个内积核函数,也就定义了一个特征空间。由上面的 SVM 求解算法可知,由于 SVM 的解(最优分类面)完全是由支持向量(对应于 $\alpha\neq 0$ 的向量)决定的,因此,为了求出最优分类面,只需要求出支持向量即可[1](如图 1 所示)。

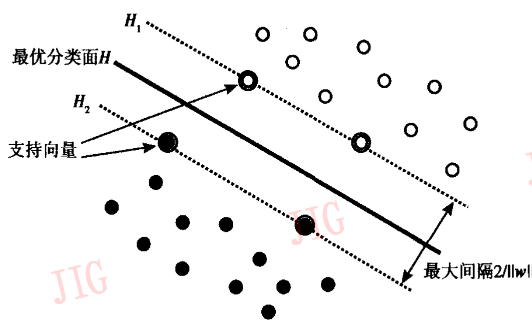

图 1　SVM 所求出的最优分类面(H 是最优分类面 H_1 和 H_2 是支持平面,加圈的样本是支持向量)

Fig. 1　Optimal classification hyperplanes of SVM(H is optimal classification hyperplane. H_1 and H_2 are support planes. The support vectors are circled)

3　KKT 条件

SVM 的优化方法是基于二阶优化条件的,即 KKT(Karush-Kuhn-Tucker)条件[1,7],以下就是 SVM 的 KKT 条件:

(1) $\alpha_i=0\Rightarrow y_i f(\boldsymbol{x}_i)\geqslant 1$ 且 $\xi_i=0$

(2) $0\leqslant\alpha_i\leqslant C\Rightarrow y_i f(\boldsymbol{x}_i)=1$ 且 $\xi_i=0$

(3) $\alpha_i=C\Rightarrow y_i f(\boldsymbol{x}_i)\leqslant 1$ 且 $\xi_i\geqslant 0$

从上面的KKT条件可知,SVM的最终解将训练集样本分成以下3个部分:①符合条件(1)的正确分类的样本集;②符合条件(2)的支持向量集;③符合条件(3)的误差样本集。

一般的SVM二次优化算法都是以KKT条件作为判断样本是否正确分类的标准,如果样本违反了KKT条件,那么此样本就是误差样本,必须进行惩罚,可以采用一次或二次惩罚,第2节算法所采用的就是一次惩罚。

4 几何快速算法

几何快速算法是基于SVM的几何原理,借鉴Keerthi提出的NPA算法[5]和Roobaert提出的DIRECTSVM算法思想[6]所提出的一种快速而简单的算法。几何快速算法不仅克服了求解二次规划算法时计算量大的缺点,而且算法的速度非常快,具体算法如下:

对于两类分类问题,SVM算法的目的就是求出这两类样本的最优分类面,但几何快速算法与经典的二次规划算法不同,它无需求解二次规划问题,而是直接根据训练样本的几何分布,累积地求出支持向量集,即可以确定最优分类面。具体方法是:从样本分布的几何位置上,首先找出两类中相距最近的两个样本作为初始的两个支持向量,然后通过判断最大违反KKT条件的样本来找出离此样本最近的异类样本,并将这两个样本作为支持向量,以此类推,最后即可得到没有违反KKT条件的样本,这样最优分类面就找出了。

4.1 原空间中几何快速算法

(1) 从当前训练集中找出分别属于两类的两个距离最近的样本点,并将该两点作为支持向量;

(2) 将过两个最近点的中点 C_0,并垂直于两点连线的一个超平面作为初始最优超平面;

(3) 找出以此超平面为最优分类面时,最大违反KKT条件的点;

(4) 将此点作为一个新的支持向量,同时在另一类样本点中,找出与此点相距最近的点,并将此点也作为支持向量,同时对原超平面以 C_0 为支点进行旋转,再通过调整超平面的位置,使得新的超平面经过后两个支持向量连线的中点 C_1,并调整最优超平面的参数;

(5) 重复步骤(3)和步骤(4),直到没有违反KKT条件为止(或允许一个最小误差);

(6) 当有新的训练样本加入时,只在新的样本中找出最大违反KKT条件者,并重复步骤(4)和步骤(5);

(7) 对原支持向量集中所有样本支持向量的条件进行重新判断,将不符合支持向量条件的样本从原支持向量集中剔除;

(8) 停止迭代。

图2、图3显示了算法的求解过程。

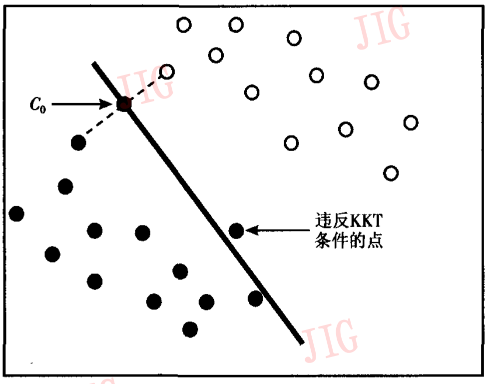

图2 快速算法的初始化

Fig. 2 Initialization of geometrical fast algorithm

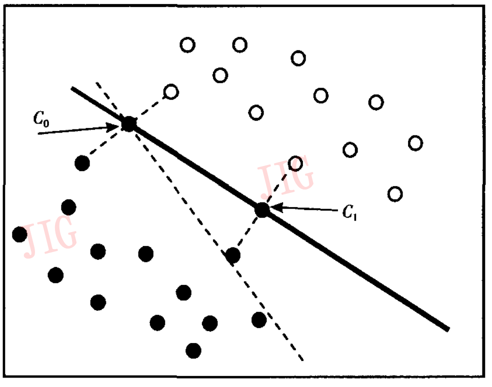

图3 快速算法的迭代更新

Fig. 3 Iterative update of geometrical fast algorithm

图中黑白圆点为两类样本,粗线就是分类面,从图中可以看出,通过不断地找出违反KKT条件的样本,并不断地调整分类面的方向,最后就可以获得最优分类面。

4.2 高维特征空间中几何快速算法

高维特征空间中的几何快速算法与原空间中的几何快速算法类似,即首先,原空间的向量经过一个

非线性映射 $\varphi:\mathbf{R}^d\rightarrow \boldsymbol{H},\boldsymbol{x}\rightarrow\varphi(\boldsymbol{x})$ 将原空间的数据映射到一个高维的特征空间 $\boldsymbol{H}$ 中,这时两类训练样本就变为:$\varphi(\boldsymbol{x}_1^+),\varphi(\boldsymbol{x}_2^+),\cdots,\varphi(\boldsymbol{x}_{N_+}^+),\varphi(\boldsymbol{x}_1^-),\varphi(\boldsymbol{x}_2^-),\cdots,\varphi(\boldsymbol{x}_{N_-}^-)$,$n=N_++N_-$ 是样本总数;然后在特征空间中用 4.1 节中所述的算法进行训练,只是在高维特征空间中进行距离计算比较复杂,但是由于引入了核函数,距离的计算都是通过核函数来进行的,因此算法的实现还是比较简单。如果采用 Mercer 核时,则在核空间中,两个样本之间的距离计算公式[8]为

$$\|\varphi(\boldsymbol{x}_i)-\varphi(\boldsymbol{x}_j)\|^2=k(\boldsymbol{x}_i,\boldsymbol{x}_i)+k(\boldsymbol{x}_j,\boldsymbol{x}_j)-2k(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{4}$$

由于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,则势必运算量很大。若采用一种新的核函数——条件正定核函数(conditionally positive definite kernel, CPD 核)[8,9]来计算特征空间中的距离,则高维特征空间(也称核空间)中两个样本之间的距离计算公式为

$$\|\varphi(\boldsymbol{x}_i)-\varphi(\boldsymbol{x}_j)\|^2=-k_{\mathrm{CPD}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^q \tag{5}$$

这样在特征空间中的距离计算,由于只需要计算一次核函数,大大减小了运算量,从而提高了算法的速度。为了进一步减少高维空间中样本之间距离的计算时间,可以将如下的核矩阵(对称矩阵)进行存储:

$$\boldsymbol{K}=\begin{pmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & k(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{x}_n)\\ k(\boldsymbol{x}_2,\boldsymbol{x}_1) & k(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_2,\boldsymbol{x}_n)\\ \vdots & \vdots & \cdots & \vdots\\ k(\boldsymbol{x}_n,\boldsymbol{x}_1) & k(\boldsymbol{x}_n,\boldsymbol{x}_2) & \cdots & k(\boldsymbol{x}_n,\boldsymbol{x}_n)\end{pmatrix} \tag{6}$$

这样在高维空间中,由于任意两个样本之间距离就是核矩阵中的一个元素(核函数的值),因此核矩阵中的任何一个元素都是两个不同样本之间的距离。这样在高维空间中计算两个样本之间的距离就不需要重复计算,只需要查表即可,从而减少了运算量,也就提高了运算速度。

5　实　验

为了验证所提出算法的有效性,利用两个公共数据库 UCI 和 ORL 人脸数据库进行了实验,并将几何快速算法与 DIRECTSVM 算法、SMO 算法进行了比较。实验环境是 Pentium 3.0GHz,MATLAB 语言,实验结果如下:

5.1　ORL 数据库实验

实验中是利用 ORL 人脸库进行人脸识别实验,ORL 人脸库中共有 40 个人(每人 10 幅图像)的 400 幅图像(表情变化、有/无眼镜、光照和姿势也有轻度变化)。实验时,分别选择每个人的 5 幅图像作为训练集,其他 5 幅图像作为测试集,训练集和测试集不重合。在构造每个人的训练集时,尽量选择具有代表性的人脸图像作为训练集样本。识别时,首先对 ORL 人脸库中的人脸图像进行归一化处理,使图像的灰度值归一化到[0,1]区间,人脸尺寸归一化为 32 × 32,并采用主分量分析(principal component analysis, PCA)方法进行人脸图像的特征提取[10];然后利用训练集特征进行 SVM 训练;最后利用测试集特征进行识别率测试。实验中,核函数选用了以下径向基核函数:$k(\boldsymbol{x},\boldsymbol{y})=\exp(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/0.7^2)$,实验共进行了 3 次,每次都进行所有 10 个人的人脸识别,其平均实验结果如表 1 所示。

表 1　ORL 数据库实验结果

Tab.1　The experiment results of ORL database

识别算法	特征维数	平均训练时间(s)	平均测试时间(s)	识别率(%)
SMO 算法	50	6.6	2.5	88.5
DIRECTSVM 算法	50	3.8	2.3	88.7
几何快速算法	50	2.5	2.4	88.3

注:训练时间是每个人脸的 SVM 的平均训练时间。测试时间是单个人脸的平均测试时间。

5.2　Sonar database 数据实验

实验中,还利用 UCI 数据库中的 Sonar database 数据进行了实验。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,核函数选用与 ORL 人脸数据库识别一样的径向基核函数,共进行了 5 次实验,实验结果如表 2 所示。

表 2　Sonar database 数据实验结果

Tab.2　The experiment results of Sonar database

识别算法	特征维数	平均训练时间(s)	平均测试时间(s)	识别率(%)
SMO 算法	60	126.6	2.6	90.5
DIRECTSVM 算法	60	72.3	1.2	89.2
几何快速算法	60	45.5	1.6	88.8

注:同表 1。

6 结论与讨论

本文提出了一种基于几何思想的 SVM 快速算法,该几何快速算法克服了经典 SVM 算法中必须求解二次规划问题的缺点,它可直接根据两类数据的几何分布,利用迭代的方法依次找出支持向量,进而求出最优超平面,且算法简单直观。实验结果显示,几何快速算法的识别率与 SMO 算法、DIRECTSVM 算法的识别率基本相当,之所以几何快速算法的识别率略低于其他两种方法,其主要是因为几何快速算法每次迭代都产生两个新的支持向量,而 SMO 算法和 DIRECTSVM 算法每次迭代都产生一个新的支持向量,从而使几何快速算法的精度可能受到影响。但是,由于几何快速算法的训练时间明显比 SMO 算法、DIRECTSVM 算法的训练时间少,因此算法的训练速度较快。因为几何快速算法一次可以找出两个支持向量(最大违反 KKT 条件者和它的一个最近邻异类样本点),所以,算法的收敛比较快。实验发现,如果两类样本中的支持向量占样本总数的比例较小时,则几何快速算法和 DIRECTSVM 算法的速度优势更加明显。因为支持向量机的最优分类面完全是由支持向量决定的,所以只要能正确找出支持向量就可以确定最优分类面。对于同一个分类问题,当基于几何思想的算法的迭代终止误差选择不同时,其所获得的最优分类面就不同;同样,基于二次规划的算法中,当分别采用硬间隔、二次软间隔和一次软间隔方法时,其所产生的最优分类面也不同,但几何快速算法和 DIRECTSVM 算法只能用于分类问题,不能用于函数回归(逼近)。

参考文献(References)

1 Burges C J C. A tutorial on support vector machines for pattern recognition [J]. Knowledge Discovery and Data Mining, 1998, **2**(2): 121 ~ 167.

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

3 Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines [A]. In: Principe J, Gile L, Morgan N, *et al*, editors, Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing Ⅶ [C], New York, 1997: 276 ~ 285.

4 Platt J. Fast training of support vector machines using sequential minimal optimization [A]. In: Schölkopf B, Burges C J C, Smola A J, Eds. Advances in Kernel Methods—Support Vector Learning [C]. Cambridge, MA, USA: MIT Press, 1999: 185 ~ 208.

5 Keerthi S S, Shevade S K, Bhattacharyya C, *et al*. A fast iterative nearest point algorithm for support vector machineclassifier design [J]. IEEE Transactions on Neural Networks, 2000, **11**(1): 124 ~ 136.

6 Roobaert D. DIRECTSVM: A fast and simple support vector machine perceptron [A]. In: Proceedings IEEE International Workshop on Neural Networks for Signal Processing (NNSP'2000) [C], Sydney, Australia, December 2000.

7 Muller K, Mika S, Ratsch G, *et al*. An introduction to kernel-based learning algorithms [J]. IEEE Transactions on Neural Networks, 2001, **12**(2): 181 ~ 201.

8 Schölkopf B. The Kernel Trick for Distances [R]. Technical Report MSR-TR-2000-51, Microsoft Research, 1 Guildhall Street Camdridge, UK, 19 May 2000.

9 Smola A J. Learning with kernels [D]. Ph. D. dissertation, Berlin, German: Technische Universität, 1998.

10 Kong Rui, Shi Ze-sheng, Guo Li, *et al*. Improving performance of kernel principal component analysis using combination kernel functions [J]. Journal of Image and Graphics, 2004, **9**(1): 40 ~ 45. [孔锐,施泽生,郭立等. 利用组合核函数提高核主分量分析的性能[J]. 中国图象图形学报, 2004, **9**(1): 40 ~ 45.]