

# 基于向量投影的支撑向量预选取

李 青 焦李成 周伟达

(西安电子科技大学智能信息处理研究所 西安 710071)

**摘要** 支撑向量机是近年来新兴的模式识别方法,在解决小样本、非线性及高维模式识别问题中表现出了突出的优点。但在支撑向量机中,支撑向量的选取相当困难,这也成为限制其应用的瓶颈问题。该文对支撑向量机的机理经过认真分析,研究其支撑向量的分布特性,在不影响分类性能的前提下,提出了基于向量投影的支撑向量预选取法,从训练样本中预先选择具有一定特征的边界向量来代替训练样本进行训练,这样就减少了训练样本,大大加快了支撑向量机的训练速度。

**关键词** 支撑向量机;向量投影;预选取

**中图法分类号** TP301

## Pre-extracting Support Vector for Support Vector Machine Based on Vector Projection

LI Qing JIAO Li-Cheng ZHOU Wei-Da

(Institute of Intelligent Information Processing, Xidian University, Xi'an 710071)

**Abstract** Support Vector Machine (SVM), a novel method of the pattern recognition, presents excellent performance in solving the problems with small sample, nonlinear and local minima. However, training a support vector machine (SVM) is equivalent to solving a linearly constrained quadratic programming (QP) problem in a number of variables equal to the number of data points. This optimization problem is known to be challenging when the number of data points exceeds few thousands. Also, it is well known that the ratio of support vectors (SVs) is far low in many practical circumstances. So the method of pre-extracting SVs to train classifier becomes a novel task in SVM field. In this paper, on a deep investigation into the principle of SVM and its characteristic, we a new method for pre-extracting SVs based on vector projection is introduced, which reduces the training samples greatly and speeds up the SVM learning, while the ability of SVM remains unchanged.

**Keywords** support vector machine; vector projection; pre-exacting

## 1 引言

统计学习理论最早提出于 20 世纪 60 年代,它是针对小样本情况研究统计学习规律的理论<sup>[1]</sup>;在

20 世纪 90 年代中期,Vapnik 及其工作组基于此理论提出了一种新的学习算法——支撑向量机。支撑向量机是一种小样本学习方法,具有很强的推广能力,它可以看作是基于结构风险最小化的多项式神经网络或者径向基函数分类器。从具体的运行过程

收稿日期:2004-02-19;修改稿收到日期:2004-11-08.本课题得到国家自然科学基金(60372050,60133010)和国家“八六三”高技术研究发展计划项目基金(2002AA135080)资助. 李青,男,1979 年生,博士研究生,主要研究领域包括模式识别、智能信号处理及统计学习理论. E-mail: kingdomyangfan@hotmail.com. 焦李成,男,1959 年生,教授,博士生导师,主要研究领域包括非线性理论、神经网络、数据挖掘、进化算法与子波理论等. 周伟达,男,1974 年生,博士,主要研究领域包括智能信号处理、机器学习、统计理论学习和数据挖掘等.

来看,支撑向量机的训练就是求解一个线性凸约束二次规划(QP). 然而问题也由此而生: 当样本数量非常多时, 训练就要花费大量的时间, 这也成为制约支撑向量机应用于实际的主要问题<sup>[5]</sup>. 基于此问题,许多学者都提出了一些快速算法来加速支撑向量机的训练. 但是, 换一种思想, 在训练支撑向量机前, 能不能根据已有样本(或训练样本)的某些特性来预选取一些样本, 在对支撑向量机性能无大影响的前提下, 将这些预选取出来的样本代替整个训练样本集, 再结合已有的快速算法进行训练, 从而进一步加快训练速度呢?

本文作者通过认真分析支撑向量机的分类原理及其支撑向量的几何特性, 提出了一种基于向量投影的训练样本预选取方法, 并在实际的实验中得到了验证.

## 2 支撑向量机

### 2.1 线性支撑向量机

首先, 我们从两类的线性可分的问题出发: 假设一组训练样本  $D=\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ , 其中,  $\mathbf{x}_i \in R^n$ ,  $y_i \in \{-1, 1\}$ ,  $R^n$  表示输入模式的特征空间. 训练的目的就是寻找判决函数  $f(\mathbf{x}, \alpha)$  中的参数  $\alpha$ , 使这两类数据可以用  $f(\mathbf{x}, \alpha)$  分开, 同时, 泛化误差达到最小(或者有上界), 这也是结构风险最小化原理的思想<sup>[1~3]</sup>. Vapnik 指出: 具有最大间隔的分类超平面就可以满足上面的条件, 这里的间隔定义为两类模式中与超平面最近的模式到超平面的距离之和. 假定该超平面是  $\mathbf{w} \cdot \mathbf{x} + b = 0$ , 要找到这个超平面, 我们需要求解下面的二次规划问题<sup>[1]</sup>:

$$\begin{cases} \min \Phi(\mathbf{w}) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) \\ \text{s. t. } y_i[(\mathbf{x}_i \cdot \mathbf{w}) - b] \geq 1, \quad i=1, 2, \dots, l \end{cases} \quad (1)$$

如果这两类样本是不可分的, 则引入了松弛变量

$$\xi_i \geq 0, \quad i=1, 2, \dots, l \quad (2)$$

从而有下面的规划:

$$\begin{cases} \min \Phi(\mathbf{w}, \xi) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) + C \cdot \left( \sum_{i=1}^l \xi_i \right) \\ \text{s. t. } y_i[(\mathbf{x}_i \cdot \mathbf{w}) - b] \geq 1 - \xi_i, \\ \xi_i \geq 0, \quad i=1, 2, \dots, l \end{cases} \quad (3)$$

此时, 支撑向量机可以看作是寻找一个分类超平面, 该超平面通过设定一个正常数因子  $C$ , 在最大间隔和最小错分误差两者之间取一个折衷<sup>[2]</sup>; 该正常数因子  $C$  必须预先取定. 通过拉格朗日乘子法, 把式

(3) 变成其对偶形式, 得到

$$\begin{cases} \max W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{s. t. } 0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, l \\ \sum_{i=1}^l y_i \alpha_i = 0 \end{cases} \quad (4)$$

此时, 我们所求得的线性分类器是

$$f(x) = \text{sign}\left(\sum_{i=1}^l \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i) + b\right) \quad (5)$$

其中, 系数  $\alpha_i$  是 QP 问题的解, 它定义在超立方体  $[0, C]^l$  上. 在求得的解中, 每一个系数  $\alpha_i$  对应着一个训练样本, 同时, 存在许多系数  $\alpha_i$  严格等于 0; 只有那些具有非零系数的样本才会影响结果. 因而, 分类超平面只与这些样本有关, 并将对应系数  $\alpha_i$  不为零的样本称为支撑向量<sup>[3]</sup>; 从而看出, 支撑向量机的训练只与支撑向量有关, 而与非支撑向量无关<sup>[4]</sup>. 直观地看, 支撑向量存在于两类样本的边界上.

### 2.2 非线性支撑向量机

#### 2.2.1 非线性分类算法

对于给定的训练样本  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l) \in R^n \times \{-1, 1\}$ , 非线性函数  $\psi(\cdot)$  将样本从输入空间  $R^n$  映射到高维特征空间  $\psi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \dots, \varphi_n(\mathbf{x}))$  中, 从而, 二次规划式(3)变为如下形式

$$\begin{cases} \min J(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + c \sum_{i=1}^l \xi_i \\ \text{s. t. } y_i[(\varphi(\mathbf{x}_i) \cdot \mathbf{w}) - b] \geq 1 - \xi_i, \\ \xi_i \geq 0, \quad i=1, 2, \dots, l \end{cases} \quad (6)$$

利用拉格朗日乘子法, 我们得到式(6)的对偶规划,

$$\begin{cases} \max W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j (\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)) \\ \text{s. t. } 0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, l \\ \sum_{i=1}^l y_i \alpha_i = 0 \end{cases} \quad (7)$$

令核函数  $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ , 我们就可将上述优化问题写为

$$\begin{cases} \max W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s. t. } 0 \leq \alpha_i \leq C, \quad i=1, 2, \dots, l \\ \sum_{i=1}^l y_i \alpha_i = 0 \end{cases} \quad (8)$$

最终,我们所选择的超平面就变为

$$\sum_{\text{support vector}} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b = 0 \quad (9)$$

其相应的非线性分类器为

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign} \left( \sum_{\text{support vector}} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (10)$$

### 2.2.2 核函数<sup>[5]</sup>

前面叙述的核函数是要满足一定的条件的,这就是所谓的 Mercer 条件<sup>[3]</sup>. 只要核函数满足此条件,我们就可以选择该核函数来建立相应的支撑向量机. 通常说用到的核函数有以下几种类型:

- (1) 多项式核  $K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x}, \mathbf{x}_i) + 1]^d$ ;
- (2) 径向基核  $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\|\mathbf{x} - \mathbf{x}_i\|/2p)$ ;
- (3) Sigmoid 核  $K(\mathbf{x}, \mathbf{x}_i) = S(v(\mathbf{x}, \mathbf{x}_i) + c)$ .

## 3 支撑向量预选取——向量投影法

### 3.1 分类支撑向量机基本几何知识

由 2.1 节,在支撑向量机的求解过程中,一个系数  $\alpha_i$  对应着一个训练样本,同时,存在许多系数  $\alpha_i$  严格等于 0;只有那些具有非零系数  $\alpha_i$  的样本才会影晌结果,这些对应系数  $\alpha_i$  不为零的样本称为支撑向量<sup>[3]</sup>;从而得出,支撑向量机的训练只与支撑向量有关,而与非支撑向量无关<sup>[4]</sup>.

图 1 是一个二维空间中支撑向量机的图例,图中打圈的点代表支撑向量;中间的实线代表分界线. 我们可以清晰的看出支撑向量的分布情况.

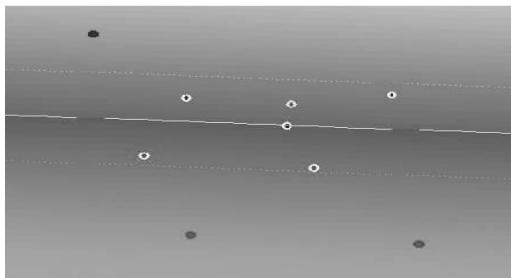


图 1 二维空间分类支撑向量机

从几何上直观地看,支撑向量就是在线性可分的空间中两类样本的交遇区内,那些靠的最近的处于两类样本边界上的样本.

我们在预选取支撑向量时,首先从线性支撑向量机出发,讲述其原理,然后将结论推向非线性支撑向量机.

### 3.2 线性支撑向量预选取

- (i) 首先,我们给出本文用到的一些定义.

**定义 1(样本中心  $\mathbf{m}_i$ )**. 定义第  $i$  类样本( $i \in \{1, 2\}$ )的平均特征为该类样本的样本中心  $\mathbf{m}_i$ ,即

$$\mathbf{m}_i = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad (11)$$

**定义 2(特征方向).** 我们将  $\mathbf{m}_1$  到  $\mathbf{m}_2$  的方向  $\overrightarrow{\mathbf{m}_1 \mathbf{m}_2}$  定义为第一类模式的特征方向,  $\mathbf{m}_2$  到  $\mathbf{m}_1$  的方向  $\overrightarrow{\mathbf{m}_2 \mathbf{m}_1}$  定义为第二类模式的特征方向.

**定义 3(特征距离  $d$ )**. 假设样本点  $\mathbf{x}_i$  到本类特征方向上的投影点为  $\mathbf{x}_i^o$ ,则  $\mathbf{x}_i^o$  到本类样本中心  $\mathbf{m}_i$  的距离定义为特征距离.

$$d(\mathbf{x}_i^o, \mathbf{m}_i) = \|\mathbf{x}_i^o - \mathbf{m}_i\|_2 = \sqrt{\sum_{j=1}^n (\xi_j - \epsilon_j)^2} \quad (12)$$

$$\mathbf{x}_i^o = (\xi_1, \xi_2, \dots, \xi_n), \quad \mathbf{m}_i = (\epsilon_1, \epsilon_2, \dots, \epsilon_n)$$

#### (ii) 边界向量预选取算法

如图 2,假设  $\mathbf{x}_i^0$  表示第一类样本的某个样本点,  $\mathbf{y}_j^0$  表示第二类样本的某个样本点,我们分别将向量  $\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^0}$  和  $\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^0}$  向特征方向上作正投影得到  $\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^o}$  和  $\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^o}$ .

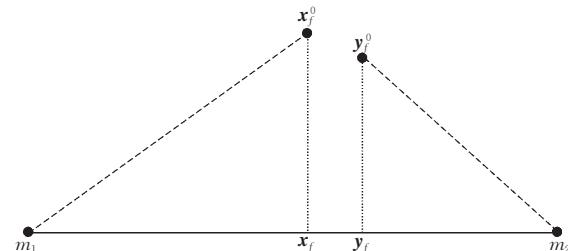


图 2 模式在特征方向上的正投影

对此,我们有如下结论.

**结论 1(边界向量).** 已知样本中心距离  $d =$

$\|\mathbf{m}_1 - \mathbf{m}_2\|$ , 分别计算特征距离  $\|\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^o}\|_2$  和  $\|\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^o}\|_2$ , 令

$$r_1 = \max_{\mathbf{x}_i^o \in \{1\} \text{类}} (\|\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^o}\|_2) \quad (13)$$

$$r_2 = \max_{\mathbf{y}_j^o \in \{2\} \text{类}} (\|\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^o}\|_2) \quad (14)$$

引入非负修正因子  $\delta \geq 0$ ,

(1) 当  $r_1 + r_2 < d$  时,若样本的特征距离  $\|\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^o}\|_2$  和  $\|\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^o}\|_2$  满足

$$r_1 - \delta \leq \|\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^o}\|_2 \leq r_1 \quad (15)$$

$$r_2 - \delta \leq \|\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^o}\|_2 \leq r_2 \quad (16)$$

时,则定义该模式为边界向量.

(2) 当  $r_1 + r_2 \geq d$  时,若特征距离  $\|\overrightarrow{\mathbf{m}_1 \mathbf{x}_i^o}\|_2$  和  $\|\overrightarrow{\mathbf{m}_2 \mathbf{y}_j^o}\|_2$  满足

$$d - r_2 - \delta \leq \|\overrightarrow{m_i x_i}\|_2 \leq r_1 + \delta \quad (17)$$

$$d - r_1 - \delta \leq \|\overrightarrow{m_j y_j}\|_2 \leq r_2 + \delta \quad (18)$$

时, 则定义该模式为边界向量.

我们的目的是希望用边界向量集来代替训练样本集进行训练, 在对支撑向量机的性能影响不大的前提下, 降低训练复杂度, 加快支撑向量集的训练时间. 那么按上述方式定义的边界向量能否满足我们的这种设计要求, 下面, 我们就通过图 3 与图 4 来加以说明.

(1)  $r_1 + r_2 < d$  时, 如图 3 所示,  $m_1$  表示第一类样本的中心, 则由定义可知第一类样本的边界向量预选取区域是夹在直线  $D_1$  与  $\text{maxpro\_1}$  之间的部分; 同理, 第二类样本的边界向量预选取区域是夹在直线  $D_2$  与  $\text{maxpro\_2}$  之间的部分.

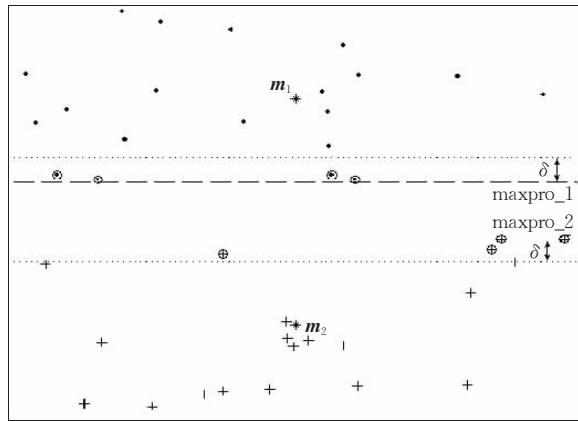


图 3  $r_1 + r_2 < d$  时的边界向量选取

(2) 当  $r_1 + r_2 \geq d$  时, 如图 4 所示,  $m_1$  表示第一类样本的中心, 则由定义知第一类样本的边界向量预选取区域是夹在直线  $D_{1-up}$  与  $D_{1-down}$  之间的部分; 同理, 第二类样本的边界向量预选取区域是夹在直线  $D_{2-up}$  与  $D_{2-down}$  之间的部分.

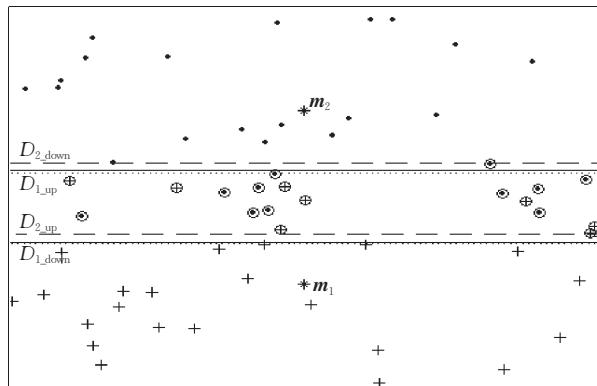


图 4  $r_1 + r_2 \geq d$  时的边界向量选取

结合上面的图和分析, 我们可以看出训练样本的分布与  $\delta$  的选取决定了边界向量的选取. 如果  $\delta$  选取适合, 边界向量集合就可以包含绝大多数的支撑向量, 甚至全部支撑向量. 最好的情况就是边界向量集等于支撑向量集. 这样, 我们在训练时就可以用边界向量集来代替支撑向量集, 从而加速训练速度.

### (iii) 修正因子 $\delta$

这里, 有两个问题: (1) 在实际的样本获取过程中是否存在噪声? (2) 在边界向量预选取中, 边界向量的分布区域是否覆盖了支撑向量的区域? 基于上述两个问题, 我们给出了修正因子  $\delta \geq 0$ , 由经验,  $\delta$  可以由下式确定:

$$\delta = \mu \cdot \text{center\_dis} + \frac{1}{(N_1 + N_2)/D} \quad (19)$$

$$\mu \in [0, 0.2], D \in [0, 10].$$

其中参数  $\mu$  描述了边界向量分布区域覆盖支撑向量分布区域的能力:  $\mu$  越大, 边界向量的分布区域覆盖支撑向量分布区域的程度越高,  $\mu$  越小, 边界向量的分布区域覆盖支撑向量分布区域的程度越低; 参数  $D$  是噪声平衡因子:  $D$  越大, 平衡噪声的能力越强,  $D$  越小, 平衡噪声的能力越差.

### 3.3 非线性支撑向量预选取

对于给定的训练样本  $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l) \in R^n \times \{-1, 1\}$ , 非线性函数  $\psi(\cdot)$  将样本从输入空间  $R^n$  映射到高维特征空间  $\psi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_n(x))$  中; 由于此时, 我们并不知道  $\psi(\cdot)$  的具体形式, 那么在高维空间中, 我们怎样计算 Euclidean 距离与向量投影呢?

**引理 1**<sup>[4]</sup>. 已知两个向量  $x = (x_1, x_2, \dots, x_n)$  和  $y = (y_1, y_2, \dots, y_n)$ , 经过非线性映射  $\psi(\cdot)$  作用映射到特征空间  $H$  中, 则这两个向量在特征空间中的 Euclidean 距离为

$$d^H(x, y) = \sqrt{K(x, x) - 2K(x, y) + K(y, y)} \quad (20)$$

其中,  $K(\cdot, \cdot)$  是 2.2.2 节中提到的核函数.

**引理 2.** 已知三个向量  $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n)$  及  $z = (z_1, z_2, \dots, z_n)$ , 经过非线性映射  $\psi(\cdot)$  作用映射到特征空间  $H$  中, 那么在特征空间中, 向量  $\overrightarrow{\psi(x)\psi(z)}$  在向量  $\overrightarrow{\psi(x)\psi(y)}$  上的投影  $\overrightarrow{\psi(x)\psi(z^o)}$  的 Euclidean 距离为

$$\|\overrightarrow{\psi(x)\psi(z^o)}\|_2 =$$

$$\frac{K(z, y) - K(z, x) - K(x, y) + K(x, x)}{\sqrt{K(x, x) - 2K(x, y) + K(y, y)}} \quad (21)$$

引理的证明见附录。

经过  $\psi(\cdot)$  的映射作用后,问题变为在高维特征空间中的线性分类问题,因此,借助引理 1 和引理 2,我们可以将结论 1 引伸到特征空间,得出特征空间中的边界向量预选取算法。

**结论 2(特征空间中的边界向量).** 设样本在特征空间中的中心距离  $d = \|\psi(\mathbf{m}_1) - \psi(\mathbf{m}_2)\|$ , 分别计算样本在特征空间中的特征距离  $\|\overrightarrow{\psi(\mathbf{m}_1)\psi(\mathbf{x}_i^o)}\|_2$  和  $\|\overrightarrow{\psi(\mathbf{m}_2)\psi(\mathbf{y}_j^o)}\|_2$ , 令

$$r_1 = \max_{\mathbf{x}_i^o \in \{1\text{类}\}} (\|\overrightarrow{\psi(\mathbf{m}_1)\psi(\mathbf{x}_i^o)}\|_2) \quad (22)$$

$$r_2 = \max_{\mathbf{y}_j^o \in \{2\text{类}\}} (\|\overrightarrow{\psi(\mathbf{m}_2)\psi(\mathbf{y}_j^o)}\|_2) \quad (23)$$

引入非负修正因子  $\delta \geq 0$ ,

(1) 当  $r_1 + r_2 < d$  时, 若特征距离  $\|\overrightarrow{\psi(\mathbf{m}_1)\psi(\mathbf{x}_i^o)}\|_2$  和  $\|\overrightarrow{\psi(\mathbf{m}_2)\psi(\mathbf{y}_j^o)}\|_2$  满足

$$r_1 - \delta \leq \|\overrightarrow{\psi(\mathbf{m}_1)\psi(\mathbf{x}_i^o)}\|_2 \leq r_1 \quad (24)$$

$$r_2 - \delta \leq \|\overrightarrow{\psi(\mathbf{m}_2)\psi(\mathbf{y}_j^o)}\|_2 \leq r_2 \quad (25)$$

定义该模式为边界向量。

(2) 当  $r_1 + r_2 \geq d$  时, 若特征距离  $\|\overrightarrow{\psi(\mathbf{m}_1)\psi(\mathbf{x}_i^o)}\|_2$  和  $\|\overrightarrow{\psi(\mathbf{m}_2)\psi(\mathbf{y}_j^o)}\|_2$  满足

$$d - r_2 - \delta \leq \|\overrightarrow{\psi(\mathbf{m}_1)\psi(\mathbf{x}_i^o)}\|_2 \leq r_1 + \delta \quad (26)$$

$$d - r_1 - \delta \leq \|\overrightarrow{\psi(\mathbf{m}_2)\psi(\mathbf{y}_j^o)}\|_2 \leq r_2 + \delta \quad (27)$$

则定义该模式为边界向量。

这样,我们就得出了在特征空间中的边界向量预选取方法。在实际的训练中,用较少的边界向量来代替整个训练样本集进行训练,加速支撑向量机的训练速度。

## 4 仿真实验

### 例 1. 线性可分的实例.

我们随机产生了两类均匀分布的样本,第一类样本为  $U([0,2] \times [0,0.95])$ , 第二类样本是  $U([0,2] \times [1.05,2])$ 。两类样本共 1600 个,其中选取 600 个作为训练样本,1000 个作为检验样本; 分别用标准 SVM 算法和本文提出的边界向量预选取算法进行实验,选取的参数如下:  $D=0, \mu=0.1$ ; 我们是在 matlab 环境下,PIV 2.6GHz,512MB 内存的微机上独立进行 150 次实验取平均的结果。图 5 是在实验中随机选取的一幅图,其中“+”和“◇”分别表示两类样本,“\*”表示预选取出来的边界向量,打圈的点是用标准 SVM 算法得出的支撑向量点。从图中,我们可以看出: 预选取的边界向量集完全包含了支撑向量,又剔除了绝大多数非支撑向量,极大地减少了训练样本,从而加快了训练时间。

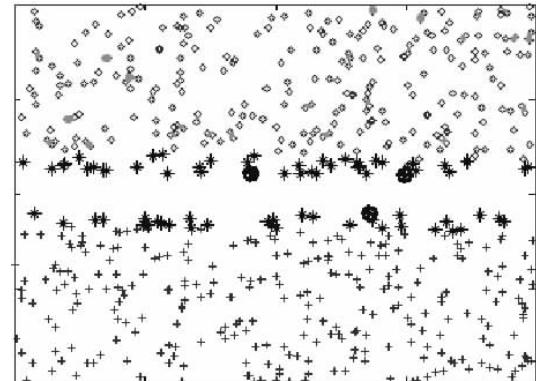


图 5 线性可分时边界向量预选取

表 1 给出了训练结果,其中,预选取算法的训练时间包括预选取边界向量和标准 SVM 规划的总和时间。

表 1 线性可分时分类性能的比较

训练算法	训练样本(个)	边界向量(个)	支撑向量(个)	训练时间(s)	检验样本(个)	识别率(%)
标准 SVM	600	无	3	287	1000	100
预选取+标准 SVM	600	59	3	0.3495	1000	100

### 例 2. 非线性可分的实例.

产生两类交错的同心圆样本  $\begin{cases} x = \rho \cdot \cos\theta \\ y = \rho \cdot \sin\theta \end{cases}, \theta \in U[0,2\pi]$ , 其中第一类样本的半径是均匀分布  $U[0,6]$ , 第二类样本的半径是均匀分布  $U[5,10]$ , 两类样本共 1600 个, 其中选取 600 个作为训练样本, 1000 个作为检验样本; 分别采用径向基核函数

$K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2p^2)$ ,  $p=8$ , 及二次多项式核函数  $K(\mathbf{x}, \mathbf{y}) = [(x, y) + 1]^2$ , 对样本进行了两次实验, 实验参数为  $D=10, \mu=0.1$ ; 在 matlab 环境下, PIV 2.6GHz, 512MB 内存的微机上独立进行 150 次实验取平均。图 6 是在实验中随机选取的一幅图, 其中“+”和“◇”分别表示两类样本,“\*”表示预选取出来的边界向量, 打圈的点是用标准 SVM 算

法得出的支撑向量点。从图中，我们可以看出：预选取的边界向量集基本上包含了所有的支撑向量。

表 2 给出了训练结果，其中，预选取算法的训练时间包括预选取边界向量和标准 SVM 规划的总和时间。

表 2 非线性可分时分类性能的比较

	训练算法	训练样本(个)	边界向量(个)	支撑向量(个)	训练时间(s)	检验样本(个)	识别率(%)
RBF 核	标准 SVM	600	无	127	338.4654	1000	90.98
	预选取+SVM	600	238	130	48.3600	1000	90.67
多项式核	标准 SVM	600	无	108	274.3669	1000	90.98
	预选取+SVM	600	171	108	25.7959	1000	90.31

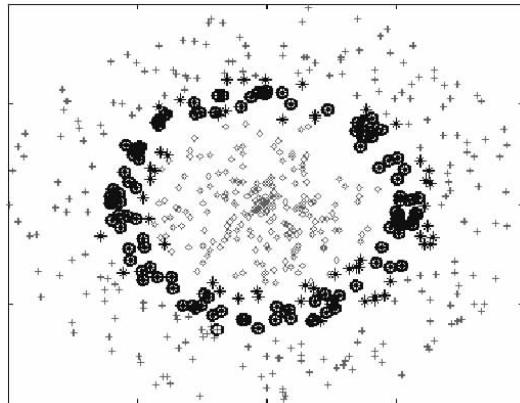


图 6 非线性可分时边界向量预选取

为了衡量边界向量选取的效率，我们定义如下几个量作为衡量。

**定义 4(标准支撑向量).** 用标准 SVM 算法对训练样本进行训练，所得出的支撑向量称作标准支撑向量。

**定义 5(支撑向量比率).** 预选取出来的边界向量所含标准支撑向量的比率。该性能指标体现了边界向量算法的识别率，指标越高，识别率越高。

**定义 6(有效边界向量比率).** 边界向量中有有效支撑向量的比率。该性能指标影响算法的速度，指标越高，速度越快，但是相应的，识别率有所降低。在实际中，应根据问题的具体要求，在支撑向量比率和有效边界向量比率之间取一个折衷。

表 3 预选取边界向量性能指标

预选取边界向量数	支撑向量比率(%)	有效边界向量比率(%)
RBF 核	96.84	54.7
多项式核	95.78	63.50

### 例 3. 大规模数据的训练.

为了测试文中的方法对大规模数据的性能，我们也进行了大规模数据的测试实验。

(1) 实验数据采用例 2 中的二维数据，选取训练样本和检验样本分别为 2000 和 8000；分别采用径向基函数  $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / 2p^2)$  ( $p=8$ ) 及二次多项式核函数  $K(\mathbf{x}, \mathbf{y}) = [(\mathbf{x}, \mathbf{y}) + 1]^2$  对样本进行了两次实验，实验参数为  $D=2, \mu=0.02$ 。在 matlab 环境下，PIV 2.6GHz, 512MB 内存的微机上独立进行 10 次实验取平均，结果如表 4 所示。

表 4 二维大规模数据训练结果

	训练算法	训练样本(个)	训练时间(s)	检验样本(个)	识别率(%)
RBF 核	标准 SVM	2000	24841	8000	91.01
	预选取+SVM	2000	583	8000	90.89
多项式核	标准 SVM	2000	114310	8000	90.92
	预选取+SVM	2000	309	8000	90.86

(2) 三维空间中线性不可分大规模数据测试。产生两类样本  $\begin{cases} x = \rho \cdot \sin \varphi \cdot \cos \theta \\ y = \rho \cdot \sin \varphi \cdot \sin \theta, \theta \in U[0, 2\pi], \varphi \in U[0, \pi] \end{cases}$ , 其中第一类样本的参数  $\rho$  是均匀分布  $U[0, 6]$ , 第二类样本的参数  $\rho$  是均匀分布  $U[5, 10]$ ,

两类样本共 10000 个，其中选取 2000 个作为训练样本，8000 个作为检验样本；采用径向基核函数  $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / 2p^2)$ ,  $p=8$ , 对样本进行实验，实验参数为  $D=2, \mu=0.02$ ；在 matlab 环境下，PIV 2.6GHz, 512MB 内存的微机上独立进行 10 次实验取平均，结果如表 5 所示。

表 5 三维大规模数据训练结果

	训练算法	训练样本(个)	训练时间(s)	检验样本(个)	识别率(%)
RBF 核	标准 SVM	2000	35216	8000	91.11
	预选取+SVM	2000	900	8000	91.07

#### 例 4. UCI 数据测试

选取 UCI<sup>①</sup> 机器学习库中的 Waveform 数据, Waveform 数据由 21 个含噪特征属性和一个类别属性构成, 它是一个三类问题, 我们选取原样本的 0 类和 2 类作为测试样本的负类和正类共 3353 个样本进行测试. 其中选取 1000 个样本作为训练样本, 其余 2353 个样本作为检验样本. 实验参数: 采用径

表 6 边界向量预选取算法对 Waveform 数据的测试

训练算法	参数	训练样本(个)	边界向量(个)	支撑向量比率(%)	训练时间(s)	检验样本(个)	识别率(%)
标准 SVM	无	1000	(# sv: 324)	100	582.56	2353	91.80
预选取+SVM	$\mu=0.0714, D=5$	1000	628	91.98	204.64	2353	90.76
预选取+SVM	$\mu=0.1347, D=8$	1000	651	94.75	230.83	2353	91.76
标准 SVM	无	1500	(# sv: 448)	100	1632.90	1853	92.31
预选取+SVM	$\mu=0.01, D=2$	1500	932	93.53	593.53	1853	89.80
预选取+SVM	$\mu=0.2, D=10$	1500	1028	99.33	656.94	1853	92.01

## 5 总 结

本文对支撑向量机经过认真分析, 研究其支撑向量的特性, 提出了向量投影的支撑向量预选取, 预先选取位于分类边界附近的样本作为训练样本, 这样可以大大减少训练样本的个数, 以很小的代价, 提高了支撑向量机的训练速度.

本文选取人工数据及 UCI 机器学习数据库的实际数据对支撑向量预选取算法进行检测, 通过理论分析并对照实验结果, 我们得出如下结论:

(1) 应用于数据的有效性. 本文提出的方法是利用向量投影预选取边界向量来代替原训练样本进行训练, 通过分析可知: 当支撑向量仅占原训练样本的一部分时, 预选取边界向量可以减少训练样本的规模, 此时, 方法的应用才是有效的; 也就是说, 训练样本必须是冗余的, 并且冗余性越大(即支撑向量所占比例越小), 方法效果越为突出.

(2) 修正因子问题. 在对实际数据的测试中, 我们发现公式(19)所定义的修正因子的选取具有这样的经验——当样本输入特征含噪时, 应当选取较大的  $\mu$ , 当类别属性含噪时, 应当选取较大的  $D$  值.

## 附 录. 引理 2 的证明.

证明. 假设三个向量  $x, y, z$  经非线性函数  $\psi(\cdot)$  映射

后, 向量  $\overrightarrow{\psi(x)\psi(z)}$  与向量  $\overrightarrow{\psi(x)\psi(y)}$  的夹角为  $\theta$ , 则

$$\cos(\theta) = \frac{(\overrightarrow{\psi(x)\psi(z)}, \overrightarrow{\psi(x)\psi(y)})}{\|\overrightarrow{\psi(x)\psi(z)}\|_2 \cdot \|\overrightarrow{\psi(x)\psi(y)}\|_2}.$$

向基核函数  $K(x, y) = \exp(-\|x-y\|^2/2p^2)$ ,  $p=20, C=1$ ; 分别用标准 SVM 算法和本文提出的边界向量预选取算法并采用不同的参数  $D, \mu$  进行实验, 以检测边界向量预选取算法在应用于实际数据中的效果. 我们是在 matlab 环境下, PIV 2.6GHz, 512MB 内存的微机上独立进行 30 次实验取平均的结果.

## 参 考 文 献

- Vapnik V. N.. The Nature of Statistical Learning Theory. New York: Spring-Verlag, 1995  
(张学工译. 统计学习理论的本质. 北京: 清华大学出版社, 2000)
- Burges C. J. C.. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 1998, 2(2): 1~47
- Vapnik V. N.. An overview of statistical learning theory. IEEE Transactions on Neural Network, 1999, 10(5): 988~999
- Jiao Li-Cheng, Zhang Li, Zhou Wei-Da. Pre-extracting support vectors for support vector machine. Acta Electronica Sinica, 2001, 29(3): 383~386(in Chinese)  
(焦李成, 张莉, 周伟达. 支撑向量预选取的中心距离比值法. 电子学报, 2001, 29(3): 383~386)
- Osuna Edgar, Freund Robert, Girosi Federico. An improved training algorithm for support vector machines. In: Proceedings of IEEE NNNSP'97, Amelia Island, FL, 1997, 24~26
- Bian Zhao-Qi, Zhang Xue-Gong. Pattern Recognition. Beijing: Tsinghua University Press, 2000(in Chinese)  
(边肇祺, 张学工. 模式识别. 北京: 清华大学出版社, 2000)
- Smola A.. Regression estimation with support vector learning machines[M. S. dissertation]. Technology University of Munich, 1996

故  $\overrightarrow{\psi(x)\psi(z)}$  在  $\overrightarrow{\psi(x)\psi(y)}$  上的投影  $\overrightarrow{\psi(x)\psi(z^\circ)}$  的 Euclidean 距离为

$$\|\overrightarrow{\psi(x)\psi(z^\circ)}\|_2 = \|\overrightarrow{\psi(x)\psi(z)}\|_2 \cdot \cos(\theta)$$

① URL: <http://www.ics.uci.edu/mlearn>.

$$\begin{aligned}
 &= \overline{\|\psi(\mathbf{x})\psi(\mathbf{z})\|_2} \cdot \frac{\overline{(\psi(\mathbf{x})\psi(\mathbf{z}), \psi(\mathbf{x})\psi(\mathbf{y}))}}{\overline{\|\psi(\mathbf{x})\psi(\mathbf{z})\|_2} \cdot \overline{\|\psi(\mathbf{x})\psi(\mathbf{y})\|_2}} \\
 &= \frac{\overline{(\psi(\mathbf{x})\psi(\mathbf{z}), \psi(\mathbf{x})\psi(\mathbf{y}))}}{\overline{\|\psi(\mathbf{x})\psi(\mathbf{y})\|_2}} \quad (*) 
 \end{aligned}$$

而

$$\begin{aligned}
 \overline{(\psi(\mathbf{x})\psi(\mathbf{z}), \psi(\mathbf{x})\psi(\mathbf{y}))} &= ((\psi(\mathbf{z}) - \psi(\mathbf{x})), (\psi(\mathbf{y}) - \psi(\mathbf{x}))) \\
 &= (\psi(\mathbf{z}), \psi(\mathbf{y})) - (\psi(\mathbf{z}), \psi(\mathbf{x})) - 
 \end{aligned}$$



**LI Qing**, born in 1979, Ph. D. candidate. His current research interests include machine learning, pattern recognition and statistic learning theory.

## Background

This work is supported by the National Natural Science Foundation of China with the title “The Theories and Applications of Evolutionary Computation” (No. 60133010), “The Recognition and Classification of Self-adaptive SVM” (No. 60372050) and the National High Technology Research and Development Program of China (863 Program) project with the title “The Intelligent Techniques for SAR Image Processing” (No. 2002AA135080). The authors have made researches on many fields of the support vector machine,

$$(\psi(\mathbf{x}), \psi(\mathbf{y})) + (\psi(\mathbf{x}), \psi(\mathbf{x})).$$

利用核函数  $K(\mathbf{x}, \mathbf{y}) = (\psi(\mathbf{x}), \psi(\mathbf{y}))$ , 则上式变为

$$\begin{aligned}
 \overline{(\psi(\mathbf{x})\psi(\mathbf{z}), \psi(\mathbf{x})\psi(\mathbf{y}))} &= K(\mathbf{z}, \mathbf{y}) - K(\mathbf{z}, \mathbf{x}) - \\
 &\quad K(\mathbf{x}, \mathbf{y}) + K(\mathbf{x}, \mathbf{x}),
 \end{aligned}$$

将此代入式(\*)并结合引理 1, 可得

$$\overline{\|\psi(\mathbf{x})\psi(\mathbf{z})\|_2} = \frac{K(\mathbf{z}, \mathbf{y}) - K(\mathbf{z}, \mathbf{x}) - K(\mathbf{x}, \mathbf{y}) + K(\mathbf{x}, \mathbf{x})}{\sqrt{K(\mathbf{x}, \mathbf{x}) - 2K(\mathbf{x}, \mathbf{y}) + K(\mathbf{y}, \mathbf{y})}}.$$

命题得证. 证毕.

**JIAO Li-Cheng**, born in 1979, Ph. D., professor and Ph. D. supervisor. His current research interests include nonlinear theory, neural network, data mining, evolutionary computation and wavelet theory.

**ZHOU Wei-Da**, born in 1974, Ph. D.. His current research interests include intelligent information processing, machine learning, statistic learning theory and data mining.

such as Linear programming support vector machine, wavelet support vector machine, support vector regression based on unconstrained convex quadratic programming and so on, and have applied these methods to the field of SAR image processing and many other fields. This paper belongs to the part of novel method of machine learning and focuses on proposing a new method for pre-exacting support vectors to speed training SVM, so as to develop the practical applications of the SVM.