文章编号：1000-8608(2008)03-0446-05

一种基于卫向量的简化支持向量机模型

王　宇*，毛玉欣

（大连理工大学 管理学院，辽宁 大连 116024）

摘要：针对支持向量机(SVM)在处理大规模训练集时，训练速度和分类速度变慢的缺点，提出了一种基于卫向量的简化SVM模型.用对偶变换及求解线性规划方法提取卫向量，缩小训练集规模；在此基础上对训练得到的支持向量集，用线性相关性去除冗余支持向量，从而达到简化目的.对UCI标准数据集的实验表明：在保证不损失分类精度的前提下，该模型一定程度上改进了传统SVM，缩短了学习时间，取得了良好的效果.

关键词：卫向量；支持向量机；训练集；支持向量集

中图分类号：TP18　　**文献标志码**：A

0　引　言

支持向量机(support vector machine，SVM)是在统计学习理论基础上发展起来的一种新型机器学习方法，在处理小样本、非线性及高维模式识别问题时，具有独特的优势[1].SVM依据结构风险最小化原则，能尽量提高学习机的泛化能力，有效地解决过学习问题，具有较好的推广性和分类精确性.由于出色的学习性能，SVM已广泛应用于模式识别、函数回归以及密度估计[2]等问题中.

与传统的学习机器相比，SVM有某些优势，但在学习速度、计算量、参数选择等方面仍显不足，特别是在速度问题上极大地限制了它的应用，成为SVM求解大规模实际问题的瓶颈[3].为了提高学习速度，研究人员做了大量工作，提出了许多快速算法，如SMO算法[4]、SVM^{light}[5]、卫向量方法[6]等.但上述算法都是针对训练阶段的改进，而不涉及分类过程，因此对分类速度的提高没有影响.类似地，Osuna等提出了3种方案以提高分类速度[7]，其中第一种方案被Burges等实现[8]，该方法虽然缩短了分类时间，但对训练过程并没有改进.

SVM在处理实际问题时，不仅训练集规模较大，而且训练得到的支持向量集也较大，为了使其训练过程和分类过程都得到简化，提高学习速度，本文提出一种基于卫向量的简化支持向量机模型.

1　支持向量机

SVM是从线性可分情况下的最优超平面发展而来的[1].对于两类问题，设样本集为$(\boldsymbol{x}_i, y_i)$，$i=1,\cdots, n$，$\boldsymbol{x}_i \in \mathbf{R}^d$，$y_i \in \{+1, -1\}$，训练算法就是要找到最优分类面$\boldsymbol{w}\cdot\boldsymbol{x}+b=0$使得它不仅能够将两类样本正确分开，并且应使两类样本到最优分类面的最小距离最大，即最大化分类间隔.SVM学习问题表示为

$$\begin{aligned} \min \quad & \frac{1}{2}(\boldsymbol{w}\cdot\boldsymbol{w}) + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i[(\boldsymbol{w}\cdot\boldsymbol{x}_i)+b] \geqslant 1-\xi_i; \\ & \xi_i \geqslant 0, i=1,\cdots,n \end{aligned} \tag{1}$$

其中ξ_i为松弛项.$\xi_i=0$时，表示线性可分情况；$\xi_i>0$时，表示线性不可分情况下允许一定的错分.惩罚参数C用于控制错分程度.通过引入Lagrange系数，把上述学习问题转化为其对偶问题：

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j (\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i y_i = 0;\ 0 \leqslant a_i \leqslant C, \\ & i=1,\cdots,n \end{aligned} \tag{2}$$

该问题是典型的二次规划问题，已有算法求解.结果中少部分不为零的a_i对应的样本为支持向量.

收稿日期：2006-06-20；　**修回日期**：2008-03-12.
基金项目：国家自然科学基金资助项目(重点项目70431001).
作者简介：王　宇*(1959-)，男，博士，教授.

对于非线性分类情况，可以通过非线性映射函数 φ 将数据从原空间映射到高维线性特征空间 H 中. 然后在 H 中，寻找最优分类面. 特征空间 H 的维数极高，SVM 算法巧妙地运用满足 Mercer 条件[9] 的核函数代替内积运算，即 $K(\boldsymbol{x},y)=\varphi(\boldsymbol{x})\cdot\varphi(y)$，从而不必明确知道函数 $\varphi(\boldsymbol{x})$ 的表达式，并且有效地避免了“维数灾”问题[9]. 常用的核函数有多项式核函数、RBF 核函数、Sigmoid 核函数[10] 等. 求得支持向量后得到相应的 SVM 最优分类函数为

$$f(\boldsymbol{x})=\operatorname{sgn}\Big[\sum_{SV}a_iy_iK(\boldsymbol{x}_i,\boldsymbol{x})+b\Big] \tag{3}$$

2　简化支持向量机算法

支持向量机由训练和分类两阶段构成，其学习速度也包括训练速度和分类速度. 从式(2)的目标函数可以看出，训练阶段问题的复杂度与训练样本数的平方相关. 运用 SVM 求解实际问题时，训练样本规模往往较大，因此在二次型寻优过程中要进行大量的矩阵运算，这不仅使算法占用大量内存空间，同时使训练速度大大降低. 从式(3)可看出，SVM 分类速度慢的主要原因是分类过程中所有支持向量都参与运算. 支持向量数较多时，导致计算量大，分类速度变慢. 为此如何在保证不损失分类精度的前提下，减小 SVM 学习过程的复杂性，就成为本文研究的目的.

2.1　训练集简化

按照支持向量的定义[11]，训练集上的点成为支持向量必须具备如下两个条件：

①能找到一个经过该点，并将其余样本点正确分开的超平面；

② 该点到最优分类面的距离为 $1/\|\boldsymbol{w}\|$.

值得注意的是训练集上有一部分样本点仅满足条件 ①. 如果能将这部分点从训练集上提取出来形成子集，然后在该子集中求解支持向量，就能达到简化训练集的目的.

定义 1　对于训练集上的样本点 $\boldsymbol{P}_i$，若能找到一超平面 l，经过 $\boldsymbol{P}_i$ 并将其余点根据其属性正确分开，则称 $\boldsymbol{P}_i$ 为卫向量.

结合定义和上述条件可知，支持向量一定是卫向量，但卫向量不一定是支持向量. 判断 $\boldsymbol{P}_i$ 是否是卫向量，就要看是否能找到满足定义的超平面 l.

如何从训练集上提取卫向量？首先引入计算几何的对偶变换方法[12]，对于点 $P(a,b)$，以 a、$-b$ 分别作为直线的斜率和截距，建立点与直线之间的一种对偶变换 D，即直线 $L:y=ax-b$ 对应于点 P.

对偶变换 D 具有以下性质：

① D 是其自身的逆，即 $D(D(P))=P$，其中 P 是一个点或一条直线.

② D 是平面上所有直线与所有点之间的一一对应关系.

上述性质说明：已知某点，就能找到惟一的变换直线，反之亦然. 卫向量的提取可以通过对偶变换后的一组线性规划(linear programming，LP)求得. 以平面上的点 $P_1(x_1,y_1)$、$P_2(x_2,y_2)$、$P_3(x_3,y_3)$ 为例：如图 1(a)所示，其中 P_1、P_2 是同类点用“+”表示，P_3 用“−”表示. 假设点 $P(a,b)$ 的对偶直线 $l_1:y=ax-b$ 经过 P_1，判断 P_1 是否是卫向量可用一组线性规划表示：

$$\begin{aligned}\max\quad & a\\ \text{s.t.}\quad & ax_1-b=y_1 \quad (l_1)\\ & ax_2-b\leqslant y_2 \quad (l_2)\\ & ax_3-b\geqslant y_3 \quad (l_3)\end{aligned} \tag{4}$$

若能找到满足式(4)约束条件的 a、b，就表示存在点 $P(a,b)$，使其对偶直线 l_1 通过 P_1，并且 P_2 位于 l_1 的上方，P_3 位于 l_1 的下方，那么 P_1 就是卫向量. 卫向量提取过程可用图 1(b) 进行几何解释，其中 l_1、l_2、l_3 分别表示经过点 P_1、P_2、P_3 的对偶直线. 图中粗线部分就是式(4)对应的可行解，P_1 就是卫向量，若图中 l_2 和 l_3 所夹的 l_1 部分为空，就表示式(4)无可行解，P_1 就不是卫向量. 在卫向量提取过程中，只需关心 LP 问题有无可行解，不必求出具体的最优值.

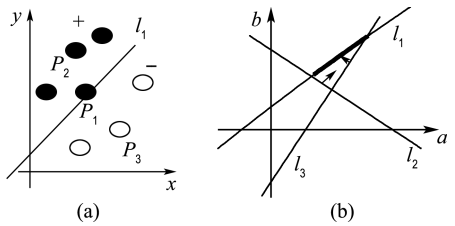

图 1　卫向量说明图

Fig. 1　Demonstration on guard vectors

以上是针对二维情况提出的. 一般地，对于 n 维空间中 m 个样本点的训练集 (x_{ij},y_i)，$i=1,2,\cdots,m$，$j=1,2\cdots,n$，$\boldsymbol{x}_i\in\mathbf{R}^n$，$y_i\in\{+1,-1\}$ 而言，点 (a_i,b)，$i=1,2,\cdots,n$ 的对偶超平面通过 $\boldsymbol{x}_i$

表示为

$$x_{i1}a_1 + x_{i2}a_2 + \cdots + x_{in}a_n - b = y_i \quad (5)$$

类似于二维空间中求解卫向量的方法，对于 $\boldsymbol{x}_1$ 建立相应的 LP 问题：

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} a_i \\ & x_{11}a_1 + x_{12}a_2 + \cdots + x_{1n}a_n - b = y_1 \\ \text{s.t.} \quad & \begin{pmatrix} x_{21} & x_{22} & \cdots & -1 \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & -1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ b \end{pmatrix} \geqslant\leqslant \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} \end{aligned} \quad (6)$$

式(6)用于判断 $\boldsymbol{x}_1$ 是否是卫向量，其中 $\geqslant\leqslant$ 根据其类别属性与第 1 个约束的类别属性相同与否选取. 对于任意点 $\boldsymbol{x}_i$ 判断其是否是卫向量，只需将式(6)中第 i 个约束条件改为等式约束，其余的约束条件根据其类别属性改成相应的不等符号即可.

上述问题是针对线性情况提出的，对于非线性问题而言，可以通过非线性映射将原空间中的点映射到高维线性空间中，然后运用同样的方法求解.

2.2 支持向量简化

用卫向量集代替原有训练集，大大缩小了训练集的规模，减少了训练时间. 实际情况表明基于卫向量集的 SVM 训练得到的支持向量数仍然很多，其中一部分支持向量是冗余的，即将这部分冗余的支持向量从支持向量集中去除，不会影响分类精确性，反而会加快分类速度.

由式(3)知，a_i 是 Lagrange 乘子，它所对应的样本点 x_i 就是支持向量，$\mathrm{K}(\cdot,\cdot)$ 是核函数. 这些支持向量之间是否存在某些联系，首先考虑到它们之间的线性相关性，即部分支持向量可用其余支持向量代替，而不会改变训练得到的整个支持向量集的性质. 现在假设支持向量集 $(\boldsymbol{x}_i, y_i)$，$i = 1,2,\cdots,k$ 中第 j 个支持向量是线性相关的，即

$$\mathrm{K}(\boldsymbol{x},\boldsymbol{x}_j) = \sum_{i=1,i\neq j}^{k} c_i \mathrm{K}(\boldsymbol{x},\boldsymbol{x}_i) \quad (7)$$

从而式(3)可重新表示为

$$f(\boldsymbol{x}) = \mathrm{sgn}\Big[\sum_{i=1,i\neq j}^{k} a_i y_i \mathrm{K}(\boldsymbol{x},\boldsymbol{x}_i) + a_j y_j \sum_{i=1,i\neq j}^{k} c_i \mathrm{K}(\boldsymbol{x},\boldsymbol{x}_i) + b\Big] \quad (8)$$

定义 $a_j y_j c_i = a_i y_i b_i$，那么式(8)就可以改写成

$$f(\boldsymbol{x}) = \mathrm{sgn}\Big[\sum_{i=1,i\neq j}^{k} a_i(1+b_i) y_i \mathrm{K}(\boldsymbol{x},\boldsymbol{x}_i) + b\Big] \quad (9)$$

设 $\beta_i = a_i(1+b_i)$ 并将其代入式(9)，得到最终的 SVM 判别函数

$$f(\boldsymbol{x}) = \mathrm{sgn}\Big[\sum_{i=1}^{m} \beta_i y_i \mathrm{K}(\boldsymbol{x},\boldsymbol{x}_i) + b\Big] \quad (10)$$

其中 m 表示调整后的支持向量集规模. 式(10)的 SVM 判别函数虽然在形式上与式(3)相同，但是它们表达的意义却存在区别. β_i 并不是通过求解二次规划得到的 Lagrange 乘子，而是将求解式(2)得到的线性相关的 a_j 用其余 a_i 表示，即去除了 a_j 对应线性相关的支持向量. 某些情况下，m 要比 k 小得多，故变换后的分类函数的运算规模明显减小，分类器速度能够大幅度提高. 特别在训练集规模庞大或者支持向量数目较多的情况下，其速度优势十分明显.

2.3 算法描述

前文给出了简化训练集和支持向量集的具体方法，旨在缩短 SVM 算法的训练和分类时间，提高学习效率. 下面给出该简化 SVM 算法的模型，如图 2 所示.

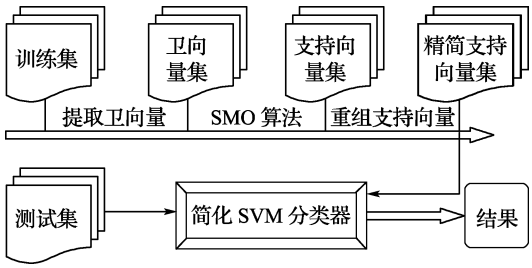

图 2 简化 SVM 模型

Fig. 2 Model for simplification SVM

具体步骤如下：

步骤 1 运用对偶变换方法，将训练集中每个样本点代入某点 $P(a,b)$ 的对偶超平面中，并设 $i = 1$.

步骤 2 为 $\boldsymbol{x}_i$ 点建立等式约束，其他与 $\boldsymbol{x}_i$ 具有相同类别属性的点建立“$\leqslant$”约束，剩下的样本点建立“$\geqslant$”约束.

步骤 3 求解 LP 问题，如果存在可行解，则 $\boldsymbol{x}_i$ 为卫向量；若没有可行解，改变所有不等式约束的符号，重新求解 LP 问题；若再没有可行解，则表示 $\boldsymbol{x}_i$ 不是卫向量.

步骤 4 令 $i = i+1$，返回步骤 2，直到所有的

样本点检测完，形成卫向量集.

步骤5　用SMO或其他标准SVM算法训练卫向量集，形成支持向量集.

步骤6　对支持向量集进行缩减，去除线性相关的支持向量.

步骤7　用精简的支持向量集构造SVM分类器，进行分类.

3　实验结果与分析

(1) 实验过程

本文采用了UCI标准数据集中的Fourclass、Pima Indians Diabetes和German三个子集作为实验数据[13]，其中Fourclass是2维数据，Pima Indians Diabetes是8维数据，German是24维数据. 为了严格检验该模型的性能，实验之前，对下载的部分数据集进行规范化，并把每个数据集分别按照2∶1划分成完全不相交的测试集(TeS)和训练集(TrS). 实验过程中，采用本文编制的卫向量提取程序和Matlab工具包作为工具.

对每个数据集进行实验时，首先应用SMO算法，得到训练和分类结果；然后应用本文提出的简化SVM模型实验，得到训练和分类结果；最后将两种算法的结果进行比较. 算法分别采用了多项式核函数(Poly-Kernel) $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i\cdot\boldsymbol{x}_j+1)^d$ 和RBF核函数(RBF-Kernel) $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left\{-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{\sigma^2}\right\}$.

由于实验过程中采用了3种不同性质的数据集，d、σ、C没有取为固定值. 因为取某一固定值，可能对部分数据集能得到理想效果，但是对其余数据集会造成分类精确性不高等缺点. 因此本文根据不同的数据集，将 d、σ、C取不同的值，实验过程中它们的取值范围分别为 $d\in[1,5]$、$\sigma\in[0.5,3.0]$、$C\in[100,500]$.

(2) 实验结果及分析

为了说明简化SVM算法模型的效果，本文把训练集和支持向量集的规模、SVM的学习时间(训练时间＋分类时间)作为考察指标，其中简化SVM算法的训练时间包括提取卫向量的时间和运用SMO算法的时间，见算法的具体步骤. 表1给出了实验结果，表2将标准SVM的实验结果和简化SVM的实验结果进行了比较.

表1　基于卫向量的简化SVM实验结果

Tab.1　Experimental results of simplification SVM based on guard vectors

数据集	N_{TrS}/N_{TeS}	核函数	标准SVM			简化SVM				
			N(SV)	时间/s	准确率/%	N(卫向量)	N(SV)	N(精简SV)	时间/s	准确率/%
Fourclass	400/200	Poly-Kernel	42	4.265	86.44	168	42	34	4.144	79.50
		RBF-Kernel	40	4.164	96.28	168	38	35	4.029	93.50
Pima Indians Diabetes	512/256	Poly-Kernel	154	27.527	80.07	291	164	102	12.281	80.86
		RBF-Kernel	149	20.271	92.38	291	92	46	8.095	92.97
German	900/300	Poly-Kernel	271	67.472	86.00	487	271	103	17.511	85.67
		RBF-Kernel	212	62.905	88.64	487	236	75	13.264	90.33

表2　标准SVM与简化SVM实验结果对照

Tab.2　Comparison of experiment results for standard SVM and simplification SVM

数据集	训练集缩小比率/%	支持向量集缩小比率/%		算法运行时间缩短比率/%	
		Poly-Kernel	RBF-Kernel	Poly-Kernel	RBF-Kernel
Fourclass	58.00	19.05	7.89	2.84	3.24
Pima Indians Diabetes	43.16	37.80	50.00	55.38	60.07
German	45.89	61.99	68.22	74.05	78.91

由表1可以看出在解决样本数较小、维数较低的数据集时，简化SVM算法与标准SVM算法在学习速度上区别不大，几乎不存在优势. 但是随着训练样本数目的增多或者维数的增高，简化SVM的优势就逐渐体现出来.

表2显示，在对样本数为1 200、维数为24的German数据集实验时，训练集规模缩小了45.89%，采用不同的核函数，支持向量集分别缩小了61.99%和68.22%，从而使简化SVM算法的学习时间比传统SVM分别缩短了74.05%和78.91%. 由此看出，简化SVM模型在处理规模较大的训练集时，在保证不损失分类精度的前提

下,能取得比较理想的效果.此外,由实验看出在参数选择合适的情况下,采用RBF核函数的分类准确率要比采用多项式核函数的分类准确率高.

4 结 语

支持向量机作为一种新型学习方法,具有严格的理论基础,较好地解决了传统方法不能解决的过学习、局部极小点等问题.但是SVM也存在一些不足之处.本文就是针对处理大规模问题时,SVM学习速度慢的缺点进行了改进.运用该模型处理分类问题,一定范围内不仅没有影响到SVM的分类精确性,而且大大缩短了SVM的学习时间.然而求解支持向量时,可用核函数代替内积运算,避免维数问题;与该方法不同,提取卫向量实际是求解线性规划,维数问题不能避免.这可能给该模型处理极高维问题带来了一定的困难,因此这也是今后需要进一步研究的问题.

参考文献:

[1] CRISTIANINI N, SHAWE-TAYLOR J. 支持向量机导论[M]. 李国正译. 北京: 电子工业出版社, 2004

[2] MÜLLER 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

[3] 王晓丹, 王积勤. 支持向量机训练和实现算法综述[J]. 计算机工程与应用, 2004, **13**: 75-78

[4] PLATT J C. Sequential minimal optimization: a fast algorithm for training support vector machines[R]// **Technical Report MSR-TR-98-14**. Seattle: Microsoft Research, 1998

[5] JOACHIMS T. Making large-scale SVM learning practical [C]// **Advances in Kernel Method: Support Vector Learning**. Cambridge: MIT Press, 1998

[6] YANG M H, AHUJA N. A geometric approach to train support vector machines [C]// **Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition**. Hilton Head Island: IEEE, 2000

[7] OSUNA E E, GIROSI F. Reducing the run-time complexity of support vector machines [M] // **Advances in Kernel Method: Support Vector Learning**. Cambridge: MIT Press, 1999

[8] BURGES C. Simplified support vector decision rules [C]//**Proceedings of the 13th International Conference on Machine Learning**. CA: Morgan Kaufmann, 1996

[9] STITSON M O, WESTON J A E, GAMMERMAN A, *et al*. Theory of support vector machines[R]// **Technical Report CSD-TR-96-17**. London: Royal Holloway University of London, 1996

[10] OSUNA E, FREUND R, GIROSI F. Support vector machines: training and applications [R]// **AIM-1602**. Cambridge: MIT AI Lab, 1997

[11] 邓乃扬,田英杰. 数据挖掘中的新方法:支持向量机[M]. 北京:科学出版社,2004

[12] 周培德. 计算几何[M]. 北京:清华大学出版社, 2000

[13] CHANG Chih-chung, LIN Chih-jen. LIBSVM data: classification, regression, and multi-label [EB/OL]. [2005-11-17]. http://www.csie.ntu.edu.tw/~ cjlin/libsvmtools/ datasets/

A model for simplification SVM based on guard vectors

WANG Yu*, MAO Yu-xin

(School of Management, Dalian University of Technology, Dalian 116024, China)

Abstract: A simplification SVM model based on guard vectors is proposed for overcoming the slow speed of training and classification for large scale training set. In order to simplify SVM, the methods of dual transform and linear programming are used to distill guard vectors; based on that, the linearly dependent support vectors are eliminated from SV set. The experiments on the UCI database are done with this algorithm. Results show that in the condition of undeclined correct rate, the running time of this model is reduced and better performance than the standard SVM is achieved.

Key words: guard vector; support vector machine; training set; support vector set