文章编号:1005-0523(2007)01-0129-04

多输出支持向量回归算法

胡　蓉

(广东金融学院 应用数学系,广东 广州 510640)

摘要:提出了一种新的多输出支持向量回归算法(M-SVR),给出了定义在超球上的损失函数,并将训练SVM的问题转化为迭代解线性方程组的问题.在求解过程中采用边计算边使矩阵降阶的办法,使得在求解的同时找到了支持向量.实验结果表明:M-SVR算法与SVR算法相比,支持向量明显减少,并且具有更好的整体预测精度和抗噪性能.

关　键　词:支持向量机;多输出;回归;SVR

中图分类号:O17　　　　**文献标识码**:A

在现实生活及生产应用中,经常会碰到要对多个变量或指标进行预测的情况,例如经济学中对时间序列指标的预测、工程应用中多维数据的拟合等,系统的输出变量 y 是一个向量,即 $y\in R^k(k>1)$.这就是所谓的多输出回归问题.目前多元统计和神经网络是解决多输出回归问题的常用方法,但这些方法对数据的依赖性较高,存在一定的局限性.

支持向量机是近年发展起来的一种性能优越的统计学习方法.它是依据Vapnik统计学习理论,以结构风险最小化为原则的学习系统,可以避免局部极小和过学习的问题.支持向量机最初用于解决模式识别问题,近些年来对其回归算法的研究也越来越多,但目前SVM回归算法都是针对单输出问题,对多输出问题讨论得很少.虽然某些情况下可以构造多个SVM来解决多输出问题,但在将多维问题化为多个一维问题的条件还没有得到很好地证明的情况下,这样做是不合理的.另外,在还没有提高SVM训练速度的情况下,这种方法在运算上也是不现实的.这些都使得对多输出回归的算法研究更具应用前景和实际意义.

1　支持向量回归算法

设训练本集 $D=\{(x_i,y_i)|i=1,2,\cdots,l\}$, $x_i\in R^M$, $y_i\in R$,构造回归函数

$$f(x)=w^T\cdot\Phi(x)+b \tag{1}$$

非线性映射 $\Phi(x)$,将数据 x 映射到高维特征空间,w 和 b 分别为权向量和偏置.在单输出回归算法中遍采用的损失函数是 ε 不敏感函数,其定义为:

$$L(|y-f(x,a)|)=\begin{cases}O, & |y-f(x,a)|\le\varepsilon\\ |y-f(x,a)|-\varepsilon, & |y-f(x,a)|>\varepsilon\end{cases} \tag{2}$$

如果把回归估计的定义为对上式的 ε 不敏感损失函数($\varepsilon\geqslant0$)进行风险最小化问题,并引入权弛变量 ζ_i 和 ζ_i^*,那么用支持向量机原理解决回归问题,可化为求解下面的二次规划问题最小化泛涵

$$R(\omega,\zeta^*\zeta)=\frac{1}{2}w^T\cdot w+C\sum_{i=1}^{l}(\zeta_i^*+\zeta_i) \tag{3}$$

式(3)的约束条件为:

收稿日期:2006-07-09

作者简介:胡蓉(1980-),女,湖南衡阳人,助教.

$$\begin{cases}(w^T\cdot\Phi(x_i)+b)-y_i\leqslant\varepsilon+\zeta_i\\ y_i-(w^T\cdot\Phi(x_i)+b)\leqslant\varepsilon+\zeta_i^*\\ \zeta_i,\zeta_i^*\geqslant 0\end{cases}\tag{4}$$

引入拉格朗日函数：

$$L=\frac{1}{2}w^T\cdot w+C\sum_{i=1}^{l}(\zeta_i^*+\zeta_i)-\sum_{i=1}^{l}a_i(y_i-w^T\cdot\Phi(x_i)-b+\varepsilon+\zeta_i)$$
$$-\sum_{i=1}^{l}a_i^*(w^T\cdot\Phi(x_i)+b-y_i+\varepsilon+\zeta_i^*)-\sum_{i=1}^{l}(\gamma_i^*\zeta_i^*,\gamma_i\zeta_i)\tag{5}$$

根据 KKT 条件，有如下等式成立：

$$w=\sum_{i=1}^{l}(a_i^*-a_i)\Phi(x_i)\tag{6}$$

$$\sum_{i=1}^{l}(a_i^*-a_i)=0\quad a_i,a_i^*\in[0,c];\quad i=1,2,\cdots,l\tag{7}$$

其中 a_i,a_i^* 为接格朗日乘子．将式(6)、(7)代入式(5)中，可将上面的二次规划问题转化为求解下面泛函的最大值：

$$W(a,a^*)=\frac{1}{2}\sum_{i=1}^{l}\sum_{i=1}^{l}(a_i^*-a_i)(a_j^*-a_j)K(x_i\cdot x_j)$$
$$+\varepsilon\sum_{i=1}^{l}(a_i^*+a_i)-\sum_{i=1}^{l}y_i(a_i^*-a_i)\tag{8}$$

其约束条件为：

$$\sum_{i=1}^{l}(a_i^*-a_i)=0\quad a_i,a_i^*\in[0,c];\quad i=1,2,\cdots,l\tag{9}$$

将在约束条件(9)下，最大化(8)求得的参数 a_i 和 a_i^* 代入式(6)得到回归函数

$$f(x)=\sum_{i=1}^{l}(a_i^*-a_i)K(x_i\cdot x)+b\tag{10}$$

式中 $a_i^*-a_i$ 不等于零对应的样本就是支持向量．

2 多输出支持向量回归算法

多输出支持向量加在归算法是针对系统函数的输出变量 y 是一个向量(即 $y\in R^k$, $k>1$)而提出一种新的SVM 算法．它主要是对单输出回归算法中的损失函数进行了改进，用定义在超球 上的损失函数代替定义在超立方体上的损失函数，将式(2)改为：

$$L(x)=\begin{cases}o & \|x\|<\varepsilon;\\ (\|x\|<\varepsilon)^2 & \|x\|\geqslant\varepsilon;\end{cases}\tag{11}$$

式(11)定义的同失函数的优点在于，它能将各分量的拟合误差都考虑进来，使目标函数与各分量的误差有关，从而达到整体优化的目的．另一方面这样定义的损失函数可弱化噪声对结果的影响，提高算法的抗噪性能．

对于 M 维输入，N 维输出的函数拟合问题，假定给定的数据样本为 $D=\{(x_i,y_i)\mid i=1,2,\cdots,L\}$, $x\in R^M$, $y\in R^M$，第 j 个输出的函数模型为 G_j:$f_j(x_i,w_j,b_j)=w_j\cdot\varphi(x_i)+b_j$, $b_j\in R$．可以将函数表过为 $F(x)=\varphi(x_i)^TW+B$，其中 $\varphi(\cdot)$是高维空间的非线性映射，$W=[\mathrm{w}^1,w^2,\cdots,w^N]$, $\mathrm{B}=[\mathrm{b}^1,b^2,\cdots,b^N]$．因此要解决多维回归问题就是要对每一个输出求出回归量 w^j 和 $b^j(j=1,2,\cdots,N)$．其目标函数如式(12)：

$$\mathrm{Min}\quad L(W,B)=\frac{1}{2}\sum_{i=1}^{l}\|W^j\|2+C\sum_{i=1}^{l}L(u_i)\tag{12}$$

其中，$L(u)$是在超球 上定义的损失函数，$u_1=\|e_i\|=\sqrt{e_ie_i^T}$, $e_i=y_i-\varphi(x_i)TW-B$．当 $\varepsilon=0$ 时，该问题就是对每一个输出分量做最小二乘回归；当 $\varepsilon\neq 0$ 时，在求解每一个输出函数的回归量 wj 和 bj 时会兼顾到其他输出分量的拟合效果，这样得到的解钭会是一个整体拟合最优的解．

根据目标函数及约束条件,可以得到以下的拉格朗日函数:

$$L(W,B)\frac{1}{2}\sum_{i=1}^{l}\|W^{j}\|2+C\sum_{i=1}^{l}L(u_i)-\sum_{i=1}^{l}a_i(u_i^2-\|y_i-\varphi(x_i)^TW-B\|^2) \tag{13}$$

将 w_j 表示为特征空间的一个线性组合,设 $w_j=\sum\beta_i^j\varphi(x_i)=\Phi^T\beta^j$,则求解式(13)可得知如下的矩阵方程

$$\begin{bmatrix} K+D_a & -1 \\ a^TK & 1^Ta \end{bmatrix}\begin{bmatrix}\beta^j \\ b^j\end{bmatrix}\begin{bmatrix}y^j \\ a^Ty^j\end{bmatrix} j=1,\cdots n \tag{14}$$

其中 $(K)_{ij}=K(x_i,x_j)$,只要将 β_j 的值求出来即可. 求解可采用迭代过程来完成. 具体步骤如下:

步骤1:初始化. 设 $k=0$, $W^k=0$, $B^k=0$ 计算 u_i^k 和 a_i 的值.

步骤2:求解(4),设得到的解为 W_s 和 B_s, $P_k=\begin{bmatrix}W^s-W^k \\ B^s-B^k\end{bmatrix}$. 需要注意的是这里的 P_k 不是一个向量而是一个矩阵. 它的每一列表示一个输出分量的回归系数的下降方向.

步骤3:获得下一步的解 $\begin{bmatrix}W^{k+1} \\ B^{k+1}\end{bmatrix}$ η^kP^k,其中 η^k 表示步长. 在计算 η^k 时,采用一维搜索的算法. 首先设 $\eta^k=1$ 看看得的 W^{k+1} 和 B^{k+1} 能否满足 $L(W^{k+1},B^{k+1})<L(W^k,B^)$,如果不能,可将 η^k 乘以一个小于1的正数,再计算 W^{k+1} 和 B^{k+1},一直到 W^{k+1} 和 B^{k+1} 满足 $L_p(W^{k+1},B^{k+1})<L_p(W^k,B^k)$,这时 W^{k+1} 和 B^{k+1} 才是本步骤所求的 W^{k+1} 和 B^{k+1}.

步骤4:根据上步所得的 W^{k+1} 和 B^{k+1} 计算 u_i^{k+1} 和 a_i,并令 $k=k+1$,返回步骤2,一直到整个过程收敛,找到目标函数的最小值,算法停止.

3　数值仿真实验

为了进一步说明上述M—SVR算法的可行性及有效性,在四个数据集上对其进行了仿真实验. 对于M—SVR和多个单输出SVR(文件称为S—SVR)的性能将从训练平均误差、训练正在角率、测试平均误差、测试正确率、训练时间、训练平均整体误差、测试平均整体误差这几个方面来比较. 其中,训练(测试)正确率定义为:当训练(测试)样本中相对误差小于0.01的样本个数与训练(测试)样本个数的比值. 整体误差的求解公式为:整体误差$=\sqrt{e_1^2+e_2^2+\cdots e_N^2}$($N$ 为输出的维数).

数据集1:函数形式为:

$$\begin{cases} x_{1,k+2}=0.1\sin(\pi x_{2,k+1})+(0.8-0.5\exp(-x_{1,k+1}^2))\cdot x_{1,k+1}-(0.3+0.9\exp(-x_{1,k+1}^2))\cdot x_{1,k} \\ x_{2,k+2}=0.6x_{2,k+1}+0.2x_{2,k+1}\cdot x_{2,k}+1.2\tan(x_{1,k}) \end{cases},$$

$k=0,1,2,\cdots$,其中初始值 $X_0=\begin{bmatrix}0.1 \\ 0.1\end{bmatrix}$.

此函数的设计是为了产生一个类似时间序列的数据集,训练样本取400个,测试样本取100个. 参数取值为:$C=1\ 000$, $\sigma=1.0$, $\varepsilon=0.000\ 1$. M—SVR算法的训练平均整体误差为0.001 308,测试平均整体误差为0.000 942. S—SVR算法的训练整体标准差为0.001 743,测试整体标准差为0.001 248. 其结果如表1所示.

表1　数据集1的M—SVR算法和S—SVR算法结果比较

输出	训练平均误差		训练正确率		测试平均误差		测试正确率	
	M—SVR	S—SVR	M—SVR	S—SVR	M—SVR	S—SVR	M—SVR	S—SVR
y_1	0.000 96	0.001 25	0.98	0.97	0.000 70	0.000 91	1.00	0.99
y_2	0.000 74	0.001 05	0.99	0.98	0.000 54	0.000 75	0.99	0.98

数据集2由以下函数生成:

$$\begin{cases} x_{1,k+1}=x_{1,k}+\sin(x_{2,k}+\sin(t))/x_{2,k}+\sin(t) \\ x_{2,k+1}=x_{1,k}^2+\sin(x_{1,k}\cdot x_{2,k}), k=0,1,2,\cdots \text{其中初始值 } X_0=\begin{bmatrix}0 \\ 0\end{bmatrix}, t\in[-10,10]. \end{cases}$$

训练样本取 800 个,测试样本取 200 个.参数取值为:$C=1\ 000$, $\sigma=1.0$, $\varepsilon=0.0001$. M—SVR 的训练整体标准差为 0.224 588,测试整体标准差为 0.220 273. S—SVR 的训练平均整体误差为 0.248 118,测试平均整体误差为 0.240 033.所得的结果如表 2 所示.

表 2 数据集 4 的 M—SVR 算法和 S—SVR 算法结果比较

输出	训练平均误差		训练正确率		测试平均误差		测试正确率	
	M—SVR	S—SVR	M—SVR	S—SVR	M—SVR	S—SVR	M—SVR	S—SVR
y_1	0.039 65	0.039 70	1.00	1.00	0.047 16	0.047 08	1.00	1.00
y_2	0.216 54	0.240 68	0.94	0.91	0.210 09	0.230 41	0.94	0.93

4 结束语

本文在单输出 SVR 算法的基础上提出了多输出 SVM 回归算法 M—SVR,并进行了仿真实验.实验结果表明:M—SVR 与单输出 SVR 算法相比,提高了精度,增强了抗噪能力.对 M—SVR 算法的研究还仅仅只是开始阶段,对其参数优化、在线学习以及实际应用方面还有待进一步的研究.

参考文献:

[1]Vapnik V. The Nature of Statistical Learning Theory[M]. New York: Springer, 1999.

[2]Drucker H, Burges C J, Kaufman L, et al. Support vector regression machines[A]. Adv Neural Infor Proc Syst[C]. Cambride: MIT Press, 1997.155—161.

[3]Golub G. H. Van L. C. F., Matrix Computations[M]. Baltimore MD: Johs Hopkins University Press, 1989.

[4]丁军, 杨丽丽. 科学与工程数值算法(Visual C++版)[M]. 北京:清华大学出版社, 2002.

[5]Suykens J. A. K., Vandewalle J.. Least Squares Support Vector Machine Classifiers[J]. Neural Processing Letters, 1999, 9(3): 293—300.

Support Vector Machine for Multi-output Regression

HU Rong

(Department of Applied Mathematics, Guangdong University of Finance, Guangzhou 510640, China)

Abstract: A new approach is proposed for SVM multi-output regression in which a hyper-spherical insensitive function is defined and iterative procedure is used to obtain the desired solution. During the operation process support vectors can be found directly by lowering the rank of matrix. The results of the experimentations illustrate that M-SVR can get higher whole precision and enhanced capability of anti-noise while the time expense is little.

Key words: support vector machine; multi—output; regression; SVR