文章编号: 1001-0920 (2016) 04-0755-04 **DOI:** 10.13195/j.kzyjc.2015.0041

密度加权孪生支持向量回归机

程昊翔, 王 坚

(同济大学 a. 电子与信息工程学院，b. CIMS 研究中心，上海 201804)

摘 要: 为了使数据集的内在分布更好地影响训练模型, 提出一种密度加权孪生支持向量回归机算法. 该算法通过 k 近邻算法计算获得每个数据点基于数据密度分布的密度加权值, 并将密度加权值引入到标准孪生支持向量回归机算法中. 算法能够很好地反映训练数据集的内在分布, 使数据点准确影响训练模型. 通过 6 个 UCI 数据集上的实验结果分析验证了所提出算法的有效性.

关键词: 密度加权；k 近邻法；内在分布

中图分类号: TP273 **文献标志码:** A

Density-weighted twin support vector regression

CHENG Hao-xiang, *WANG Jian*

(a. College of Electronics and Information Engineering，b. CIMS Research Center，Tongji University，Shanghai 201804, China．Correspondent：CHENG Hao-xiang，E-mail：373668304@qq.com)

Abstract: To better impact the training model with the inherent distribution of the training dataset, a twin support vector regression called density-weighted twin support vector regression is proposed. Firstly, the density-weighted value is computed based on the k-nearest neighbor algorithm. Then, the values of density-weighted are introduced to the standard twin support vector regression. It is found that the proposed algorithm can well reflect the inherent distribution of the training dataset and lead to a more accurate impact on the training model. Experimental results on six UCI datasets show the effectiveness of the proposed algorithm.

Keywords: density-weighted；k-nearest neighbor；inherent distribution

0 引 言

近几年, 支持向量机[1](SVM) 已成功应用于分类和回归问题. SVM 是以 VC 维理论和结构风险最小化为理论基础[2]提出的, 相比于人工神经网络等其他机器学习算法, SVM 有如下优势: 1) SVM 解决一个带约束的二次规划问题, 能够得到唯一的最优解; 2) SVM 基于结构风险最小化理论原则, 有更好的泛化性. 然而, SVM 对于大数据量的模型训练问题, 存在时间复杂度较高的问题. SVM 模型训练的时间复杂度为 $O(m^3)$, m 为训练样本的规模.

最近, Jayadeva 等[3]提出了一种用于解决分类问题的新算法 —— 孪生支持向量机 (TWSVM). TWSVM 算法的动机来源于 GEPSVM[4], 与 SVM 训练获得两个平行的超平面不同, TWSVM 通过解决两个规模较小的二次规划问题获得两个不平行的超平面. 通过计算分析, 在数据集规模相同的情况下, TWSVM 的训练时间是标准 SVM 的 1/4. Peng[5]将 TWSVM 从解决分类问题扩展到回归问题, 提出了孪生支持向量回归机 (TSVR). TSVR 经过近几年的研究有了不同的扩展[6-10], 但是这些研究所考虑的每一个训练数据对于生成超平面的贡献和作用是相同的, 没有反映出数据在真实空间中的内在分布情况.

针对上述问题, 本文提出一种基于密度加权的孪生支持向量回归机 (DW-TSVR), 在目标函数中对不同数据所对应的经验误差进行密度加权, 密度加权值在真实空间中采用 kNN 算法进行计算, 并通过计算规则将密度加权值的范围缩小在 [0,1]. 对标准 TSVR 的目标函数进行正则化, 将其扩展到结构风险最小化误差. 在 6 个 UCI 数据集上的实验比较结果表明, 所提出的 DW-TSVR 相比于 TSVR、SVR、Weighted

收稿日期: 2015-01-10；**修回日期:** 2015-04-11.

基金项目: 国家自然科学基金面上项目(71273188)；国家自然科学基金重大项目(91024031).

作者简介: 程昊翔(1986−), 男, 博士, 从事系统工程、机器学习的研究；王坚(1961−), 男, 教授, 博士生导师, 从事系统工程、智能优化算法等研究.

TSVR[6]具有更高的精确度.

1 TSVR

对TSVR进行简单介绍. 假设一个大小为m的数据集S, 表示为

$$S=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}\in(\chi\times y)^m. \tag{1}$$

其中: $\chi\in R^n$, $y\in R$, $x_i\in R^n$, $i=1,2,\cdots,m$, n为输入空间的特征数. 目标是通过构建两个优化问题获得两个函数来预测无输出的实例. 两个预测函数表示为

$$f_1(x)=w_1^{\mathrm{T}}x+b_1,\ f_2(x)=w_2^{\mathrm{T}}x+b_2.$$

最后的预测函数为

$$f(x)=\frac{1}{2}(f_1(x)+f_2(x)).$$

TSVR的几何解释如图1所示.

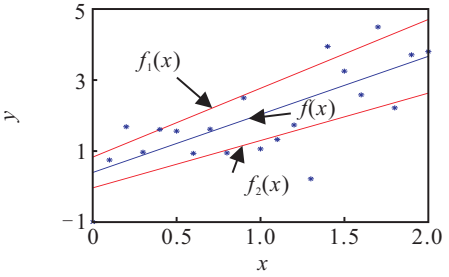

图1 TSVR几何解释

为了实现本文目标, TSVR构建如下带约束的优化函数:

$$\begin{aligned}\min\ &\frac{1}{2}(Y-e\epsilon_1-(Aw_1+eb_1))^{\mathrm{T}}\times\\&(Y-e\epsilon_1-(Aw_1+eb_1))+c_1e^{\mathrm{T}}\xi,\\\text{s.t. }&Y-(Aw_1+eb_1)\geqslant e\epsilon_1-\xi,\ \xi\geqslant 0;\end{aligned}\tag{2}$$

$$\begin{aligned}\min\ &\frac{1}{2}(Y-e\epsilon_2-(Aw_2+eb_2))^{\mathrm{T}}\times\\&(Y-e\epsilon_2-(Aw_2+eb_2))+c_2e^{\mathrm{T}}\eta,\\\text{s.t. }&(Aw_2+eb_2)-Y\geqslant e\epsilon_2-\eta,\ \eta\geqslant 0.\end{aligned}\tag{3}$$

其中

$$c_1,c_2>0,\ A=\{x_1^{\mathrm{T}},x_2^{\mathrm{T}},\cdots,x_m^{\mathrm{T}}\}\in R^{m\times n},$$

$$Y=\{y_1,y_2,\cdots,y_m\}\in R^m,\ e\in R^m,\ \epsilon_1,\epsilon_2\geqslant 0,$$

$\eta,\xi\in R^m$为松弛变量.

通过拉格朗日变换, 引入α和γ, 将式(2)和(3)转化为如下对偶形式:

$$\begin{aligned}\max\ &-\frac{1}{2}\alpha^{\mathrm{T}}G(G^{\mathrm{T}}G)^{-1}G^{\mathrm{T}}\alpha+\\&f^{\mathrm{T}}G(G^{\mathrm{T}}G)^{-1}G^{\mathrm{T}}\alpha-f^{\mathrm{T}}\alpha,\\\text{s.t. }&0\leqslant\alpha\leqslant c_1e;\end{aligned}\tag{4}$$

$$\begin{aligned}\max\ &-\frac{1}{2}\gamma^{\mathrm{T}}G(G^{\mathrm{T}}G)^{-1}G^{\mathrm{T}}\gamma-\\&h^{\mathrm{T}}G(G^{\mathrm{T}}G)^{-1}G^{\mathrm{T}}\gamma+h^{\mathrm{T}}\gamma,\\\text{s.t. }&0\leqslant\gamma\leqslant c_2e.\end{aligned}\tag{5}$$

其中

$$G=[A\ \ e],\ f=Y-e\epsilon_1,\ h=Y+e\epsilon_2,$$

$$u_1=[w_1^{\mathrm{T}}\ \ b_1],\ u_2=[w_2^{\mathrm{T}}\ \ b_2].$$

在式(5)对偶转化过程中可以得到u_1、u_2与α、γ之间的关系为

$$u_1=(G^{\mathrm{T}}G+\sigma I)^{-1}G^{\mathrm{T}}(f-\alpha),$$

$$u_2=(G^{\mathrm{T}}G+\sigma I)^{-1}G^{\mathrm{T}}(h+\gamma).$$

通过求解式(4)和(5)两个二次规划问题, 可以求得α和γ, 进一步求解得到u_1和u_2, 从而获得回归预测模型$f(x)$为

$$f(x)=\frac{1}{2}(f_1(x)+f_2(x))=$$

$$\frac{1}{2}(w_1+w_2)^{\mathrm{T}}x+\frac{1}{2}(b_1+b_2).$$

2 DW-TSVR

引入密度加权值$\rho_i\,(i=1,2,\cdots,m)$, 定义为

$$\rho_i=1-\frac{d(x_i,x_i^k)}{\max\limits_{j\in\text{train set}}d(x_j,x_j^k)}. \tag{6}$$

其中: x_i^k为通过kNN算法[11]在输入原空间计算获得的与数据点x_i第k个邻近的点, 本文取$k=10$; $d(x_i,x_i^k)$为x_i与x_i^k之间在原空间的距离.

密度加权值衡量了训练数据点的相对密度, 通过计算每个数据点与其对应的第k个近邻点之间的距离与训练数据中此计算距离的最大值进行比较获得. 密度加权值的范围为[0,1]. 通过密度加权值的引入, 当数据点拥有较大密度加权值时, 其对于训练获得的模型有较大的影响, 能准确地反映训练数据集的内在分布.

通过上述计算, 训练数据集S中每个数据点x_i都有对应的密度加权值ρ_i, 将ρ_i组成$m\times m$大小的对角矩阵$W=\mathrm{diag}(\rho_1,\rho_2,\cdots,\rho_m)$, 其对角线值为对应的密度加权值.

DW-TSVR将标准TWSVR的经验风险最小化加入正则项$(w^2+b^2)/2$, 扩展到结构风险最小化, 然后将密度加权矩阵W引入到新的目标函数中, 形成新的二次规划问题, 表示为

$$\begin{aligned}\min\ &\frac{1}{2}(Y-(Aw_1+eb_1))^{\mathrm{T}}W(Y-(Aw_1+eb_1))+\\&\frac{1}{2}c_3(w_1^{\mathrm{T}}w_1+b_1^2)+c_1e^{\mathrm{T}}\xi,\\\text{s.t. }&Y-(Aw_1+eb_1)\geqslant-e\epsilon_1-\xi,\ \xi\geqslant 0;\end{aligned}\tag{7}$$

$$\begin{aligned}\min\ &\frac{1}{2}(Y-(Aw_2+eb_2))^{\mathrm{T}}W(Y-(Aw_2+eb_2))+\\&\frac{1}{2}c_4(w_2^{\mathrm{T}}w_2+b_2^2)+c_2e^{\mathrm{T}}\eta,\\\text{s.t. }&(Aw_2+eb_2)-Y\geqslant-e\epsilon_2-\eta,\ \eta\geqslant 0.\end{aligned}\tag{8}$$

其中: $c_1,c_2,c_3,c_4>0$; $\eta,\xi\in R^m$为松弛变量.

引入α和γ, 通过拉格朗日变换, 得到式(7)和(8)的对偶问题形式为

$$\begin{aligned}\max\ & -\frac{1}{2}\alpha^{\mathrm{T}}G(G^{\mathrm{T}}WG+c_3I)^{-1}G^{\mathrm{T}}\alpha+\\ & Y^{\mathrm{T}}G(G^{\mathrm{T}}WG+c_3I)^{-1}G^{\mathrm{T}}\alpha-\\ & (e^{\mathrm{T}}\epsilon_1+Y^{\mathrm{T}}W)\alpha,\\ \text{s.t. }& 0\leqslant\alpha\leqslant c_1e;\end{aligned}\tag{9}$$

$$\begin{aligned}\max\ & -\frac{1}{2}\gamma^{\mathrm{T}}G(G^{\mathrm{T}}WG+c_4I)^{-1}G^{\mathrm{T}}\gamma-\\ & Y^{\mathrm{T}}G(G^{\mathrm{T}}WG+c_4I)^{-1}G^{\mathrm{T}}\gamma+\\ & (Y^{\mathrm{T}}W-e^{\mathrm{T}}\epsilon_2)\gamma,\\ \text{s.t. }& 0\leqslant\gamma\leqslant c_2e.\end{aligned}\tag{10}$$

在上述问题(9)和(10)的转化过程中得到

$$\begin{aligned}&h_1=[w_1^{\mathrm{T}}\ \ b_1]^{\mathrm{T}}=\\ &(G^{\mathrm{T}}WG+c_3I)^{-1}G^{\mathrm{T}}(WY-\alpha),\\ &h_2=[w_2^{\mathrm{T}}\ \ b_2]^{\mathrm{T}}=\\ &(G^{\mathrm{T}}WG+c_4I)^{-1}G^{\mathrm{T}}(WY+\gamma).\end{aligned}$$

由式(9)和(10)两个带约束的二次规划问题可以求解α和γ, 从而得到h_1和h_2, 最终得到如下预测回归模型:

$$\begin{aligned}&f(x)=\frac{1}{2}(f_1(x)+f_2(x))=\\ &\frac{1}{2}(w_1+w_2)^{\mathrm{T}}x+\frac{1}{2}(b_1+b_2).\end{aligned}$$

3　逐次超松弛技术

采用逐次超松弛技术[12], 对式(4)、(5)、(9)、(10)这4个二次规划问题进行求解. 首先, 将上述4个问题转化为如下统一形式:

$$\min\ \frac{1}{2}\alpha^{\mathrm{T}}Q\alpha-d^{\mathrm{T}}\alpha,\tag{11}$$

$$\text{s.t. }0\leqslant\alpha\leqslant c.\tag{12}$$

其中Q为正定矩阵. 以二次规划问题(9)为例, 有

$$\begin{aligned}&Q=G(G^{\mathrm{T}}WG+c_3I)^{-1}G^{\mathrm{T}},\\ &d=Y^{\mathrm{T}}G(G^{\mathrm{T}}WG+c_3I)^{-1}G^{\mathrm{T}}-(e^{\mathrm{T}}\epsilon_1+Y^{\mathrm{T}}W),\\ &c=c_1e.\end{aligned}$$

逐次超松弛技术描述如下. 已知α^i, 按照下式计算α^{i+1}:

$$\begin{aligned}&\alpha^{i+1}=\\ &(\alpha^i-tE^{-1}(Q\alpha^i-d+L(\alpha^{i+1}-\alpha^i))).\end{aligned}\tag{13}$$

其中: $t\in(0,2)$, $L\in R^{m\times m}$为Q的下三角部分, 对角矩阵$E\in R^{m\times m}$为Q的对角线. 逐次超松弛技术按式(13)迭代计算, 直到$\|\alpha^{i+1}-\alpha^i\|$小于某个规定的容错值, 本文容错值取$10^{-3}$.

逐次超松弛技术对解决大数据问题的孪生支持向量回归机非常有效, 而且已经证明该算法对于二次规划问题求解线性收敛[13]. 文中标准TWSVR和DW-TWSVR算法均采用逐次超松弛技术进行求解.

4　实验分析

为了验证DW-TSVR算法的回归性能, 选择UCI标准数据集上的6个数据集, 使用高斯核将所提出的基于密度加权的孪生支持向量回归机与标准SVR, 标准TWSVR算法和Weighted TSVR进行对比测试, 表1为6个数据集的规模和属性信息.

表 1　测试数据集

数据集名称	数据量	特征维度
Boston housing	506	13
Auto price	159	15
Auto-Mpg	398	7
Machine CPU	209	17
Diabetes	43	2
Pyrim	74	28

采用网格搜索算法5折交叉验证算法进行参数的选择. 5折交叉验证即随机将数据集分成5个子集, 4个子集用来训练模型, 剩下的子集用来测试训练得到的模型, 最后将5次的评价结果求平均值获得最终的训练模型. 高斯核函数为

$$K(x_i,x_j)=\mathrm{e}^{-\frac{\|x_i-x_j\|^2}{r^2}}.$$

设定$c_1=c_2,c_3=c_4,\epsilon_1=\epsilon_2$. 超参数$c_1,c_2,c_3,c_4$从$\{2^i|i=-2,-1,\cdots,8\}$中选取; 高斯核参数$r$从$\{2^i|i=-4,-3,\cdots,8\}$中选取; ϵ_1,ϵ_2从$\{0.1,0.2,\cdots,0.9,1\}$中选取. 性能评价标准采用均方根误差(RMSE)标准和规一化均方误差(NMSE)两个评价标准. RMSE定义为

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{m}(y_i-\hat{y}_i)^2}{m}};$$

NMSE定义为

$$\mathrm{NMSE}=10\log_{10}\frac{\sum_{i=1}^{m}|y_i-\hat{y}_i|^2}{\sum_{i=1}^{m}|y_i|^2}.$$

其中: m为训练数据集数据的数量, y_i为实际的第i个输出值, $\hat{y}_i$为预测的第i个输出值. 表2为SVR、TSVR、Weighted TSVR和DW-TSVR四个算法通过5折交叉验证得到的各个数据集的RMSE、NMSE值.

从实验结果看, 在选取的6个数据集中, 本文所提出的DW-TSVR算法较SVR、TSVR、Weighted TSVR算法在回归精确度上有较明显的提高, 在RMSE和NMSE两个指标上都显示出了较好的性能, 验证了所提出的DW-TSVR算法的有效性. 本文所提出的密度加权孪生支持向量回归机算法通过引入密

表 2 算法测试结果比较

数据集	算法	RMSE	NMSE	Time/s
Boston housing	SVR	8.307	0.148	486.5
	TSVR	7.968	0.122	0.230
	Weighted TSVR	7.605	0.184	0.172
	DW-TSVR	7.462	0.118	0.170
Auto price	SVR	7 824.6	0.389	8.677
	TSVR	7 233.0	0.320	0.425
	Weighted TSVR	3 845.0	0.303	0.422
	DW-TSVR	3 432.6	0.288	0.410
Auto-Mpg	SVR	9.106	0.127	388.6
	TSVR	4.212	0.116	1.876
	Weighted TSVR	3.656	0.114	1.890
	DW-TSVR	3.356	0.112	1.825
Machine CPU	SVR	156.7	0.236	20.80
	TSVR	148.8	0.225	0.539
	Weighted TSVR	78.16	0.204	0.467
	DW-TSVR	69.45	0.210	0.425
Diabetes	SVR	0.572	0.216	0.150
	TSVR	0.552	0.205	0.072
	Weighted TSVR	0.533	0.198	0.063
	DW-TSVR	0.510	0.186	0.061
Pyrim	SVR	0.078	0.112	1.245
	TSVR	0.068	0.092	0.568
	Weighted TSVR	0.066	0.088	0.492
	DW-TSVR	0.066	0.089	0.488

度加权值, 使模型能够更好地反映训练数据集的内在分布, 从而获得了更加精确的训练模型. 相比于标准SVR算法, 在训练速度上有明显的提高, 同时也验证了采用逐次超松弛技术解决TSVR算法求解问题的可行性.

5 结 论

本文提出了一种新的孪生支持向量回归机, 称为密度加权孪生支持向量回归机(DW-TSVR). 引入密度加权值的概念, 将标准的孪生支持向量回归机优化目标从经验风险最小化扩展到结构风险最小化, 并将密度加权值引入到新的目标函数中, 所提出算法的对偶形式采用逐次超松弛算法进行求解. DW-TSVR算法与SVR、TSVR、Weighted TSVR在6个UCI数据集上的比较结果表明, 所提出的算法在预测精确度上有一定的提高, 验证了所提出算法的有效性.

参考文献(References)

[1] Cortes C, Vapnik V. Support vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[2] Vapnik V. The nature of statistical learning theory[M]. New York: Springer-Verlag, 1995: 93-110.

[3] Jayadeva, Khemchandani R, Chandra S. Twin support vector machines for pattern classification[J]. IEEE Trans on Pattern Analysis and Machine Intellegence, 2007, 29(5): 905-910.

[4] Mangasarian O L, Wild E W. Multisurface proximal support vector classification via generalized eigenvalues[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2006, 28(1): 69-74.

[5] Peng X. TSVR: An efficient twin support vector machine for regression[J]. Neural Networks, 2010, 23(3): 365-372.

[6] Xu Yitian, Wang Laisheng. A weighted twin support vector regression[J]. Knowledge-Based Systems, 2012, 33: 92-101.

[7] Shao Yuanhai, Zhang Chunhua, Wang Xiaobo, et al. Improvements on twin support vector machines[J]. IEEE Trans on Neural Networks, 2011, 20(6): 962-968.

[8] Peng Xinjun, Xu Dong. A twin projection support vector machine for data regression[J]. Neurocomputing, 2014, 138: 131-141.

[9] Ding Shifei, Wu Fulin, Shi Zhongzhi. Wavelet twin support vector machine[J]. Neural Computer & Application, 2014, 25(6): 1241-1247.

[10] Shao Yuanhai, Zhang Chunhua. An ϵ-twin support vector machine for regression[J]. Neural Computer & Application, 2013, 23(1): 175-185.

[11] Li I, Chen J, Wu J. A fast prototype reduction method based on template reduction and visualization-induced self-organizing map for nearest neighbor algorithm[J]. Application Intelligence, 2013, 39(3): 564-582.

[12] Olvi L Mangasarian, David R Musicant. Successive overrelaxation for support vector machines[J]. IEEE Trans on Neural Networks, 1999, 10(5): 1032-1037.

[13] Luo Z Q, Tseng P. Error bounds and convergence analysis of feasible descent methods: A general approach[J]. Annals of Operations Research, 1993, 67(2): 157-178.

(责任编辑：郑晓蕾)