软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(11):2535−2547 [doi: 10.3724/SP.J.1001.2013.04472]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

动态粒度支持向量回归机*

郭虎升[1], 王文剑[1,2]

[1](山西大学 计算机与信息技术学院,山西 太原 030006)

[2](计算智能与中文信息处理教育部重点实验室(山西大学),山西 太原 030006)

通讯作者: 王文剑, E-mail: wjwang@sxu.edu.cn, http://scit.sxu.edu.cn/SchoolTeacherD.aspx?id=326

摘 要: 粒度支持向量机(granular support vector machine,简称 GSVM)可以有效提高支持向量机(support vector machine,简称 SVM)的学习效率,但由于经典 GSVM 通常将粒用个别样本替代,且粒划和学习在不同空间进行,因而不可避免地改变了原始数据分布,从而可能导致泛化能力降低.针对这一问题,通过引入动态层次粒划的方法,设计了动态粒度支持向量回归(dynamical granular support vector regression,简称 DGSVR)模型.该方法首先将训练样本映射到高维空间,使得在低维样本空间无法直接得到的分布信息显示出来,并在该特征空间中进行初始粒划.然后,通过衡量样本粒与当前回归超平面的距离,找到含有较多回归信息的粒,并通过计算其半径和密度进行深层次的动态粒划.如此循环迭代,直到没有信息粒需要进行深层粒划时为止.最后,通过动态粒划过程得到的不同层次的粒进行回归训练,在有效压缩训练集的同时,尽可能地使含有重要信息的样本在最终训练集中保留下来.在基准函数数据集及 UCI 上的回归数据集上的实验结果表明,DGSVR 方法能够以较快的速度完成动态粒划的过程并收敛,在保持较高训练效率的同时可有效提高传统粒度支持向量回归机(granular support vector regression machine,简称 GSVR)的泛化性能.

关键词: 支持向量回归;动态粒度支持向量回归;动态粒划;信息粒;半径;密度

中图法分类号: TP181 **文献标识码**: A

中文引用格式: 郭虎升,王文剑.动态粒度支持向量回归机.软件学报,2013,24(11):2535−2547. http://www.jos.org.cn/1000-9825/4472.htm

英文引用格式: Guo HS, Wang WJ. Dynamical granular support vector regression machine. Ruan Jian Xue Bao/Journal of Software, 2013,24(11):2535−2547 (in Chinese). http://www.jos.org.cn/1000-9825/4472.htm

Dynamical Granular Support Vector Regression Machine

GUO Hu-Sheng[1], WANG Wen-Jian[1,2]

[1](School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China)

[2](Key Laboratory of Computational Intelligence and Chinese Information Processing (Shanxi University), Ministry of Education, Taiyuan 030006, China)

Corresponding author: WANG Wen-Jian, E-mail: wjwang@sxu.edu.cn, http://scit.sxu.edu.cn/SchoolTeacherD.aspx?id=326

Abstract: Although granular support vector machine (GSVM) can improve the learning speed, the generalization performance may be decreased because the original data distribution will be changed inevitably by two reasons: (1) A granule is usually replaced by individual datum; (2) Granulation and learning are carried out in different spaces. To address this problem, this study presents a granular support vector regression (SVR) model based on dynamical granulation, namely DGSVR, by using the dynamical hierarchical granulation method. With DGSVR, the original data are mapped into the high-dimensional space by mercer kernel to reveal the distribution features implicit in original sample space, and the data are divided into some granules initially. Then, some granules are obtained with important regression

* 基金项目: 国家自然科学基金(60975035, 61273291); 山西省回国留学人员科研基金(2012008); 山西省研究生教育创新项目(20133001)

收稿时间: 2013-04-08; 修改时间: 2013-07-16, 2013-08-02; 定稿时间: 2013-08-27

information by measuring the distances of granules and regression hyperplane. By computing the radius and density of granules, the deep dynamical granulation process executes until there are no informational granules need to be granulated. Finally, those granules in different granulation levels are extracted and trained by SVR. The experimental results on benchmark function datasets and UCI regression datasets demonstrate that the DGSVR model can quickly finish the dynamical granulation process and is convergent. It concludes this model can improve the generalization performance and achieve high learning efficiency at the same time.

Key words: support vector regression; dynamical granular support vector regression (DGSVR); dynamical granulation; informational granule; radius; density

随着科学技术的进步以及人们的管理水平和知识水平的提高,现实世界中需要处理的数据量越来越庞大.互联网数据中心(Internet data center,简称 IDC)发布的研究报告指出,2011 年,全球数据存储量达到 1.8ZB[1].因此,如何处理海量数据的挖掘问题,近年来已经成为机器学习领域的一个研究热点.支持向量机[2]由于其出色的泛化性能,近年来被广泛地应用于分类和回归问题,并在手写体识别、人脸图像识别、时间序列预测等方面得到了成功的应用.但是,SVM 在处理大规模数据时训练效率低下,一直是制约其发展和应用的一个瓶颈.目前,针对 SVM 的训练效率问题,研究人员已经提出了一些改进方法[3−10].这些方法尽管在一定程度上提高了学习器的训练效率,但却降低了泛化性能.

为了提高传统 SVM 的学习效率,Tang 等人[11]首先结合粒度计算理论提出了粒度支持向量机(granular support vector machine,简称 GSVM)这一术语.GSVM 方法首先在原始样本空间中建立一系列信息粒,然后在划分后的粒中提取部分重要的信息样本进行学习,最后将不同粒上学习所得到的不同重要信息进行融合,得到最终的分类器.GSVM 的基本学习过程如图 1 所示.GSVM 不仅对于非线性问题具有较好的泛化能力,而且可以增强非线性可分问题的线性可分性,甚至可以将非线性可分问题转换为简单的线性可分问题.与传统 SVM 方法相比,GSVM 的训练效率大幅度提高,同时也保持了较好的泛化性能.

Fig.1 Basic process of GSVM model

图 1 GSVM 模型的基本过程

可以看出,粒度支持向量机只是一种思想,任何采用分组、分类、聚类甚至矩阵分解手段,将大规模的支持向量机学习问题转换成小规模的子学习问题的加速支持向量机方法,都可以认为是一种具体的粒度支持向量机的实现方法.实际上,在 Tang 之前,有很多学者已经提出了一些加速的 SVM 模型,它们可被看作 GSVM 的雏形,如经典的分块算法[2]、分解算法[12]、序贯最小优化算法(SMO)[13]等.许多学者还设计了不同的改进 GSVM 模型,如:Wang 等人[14,15]提出了基于聚类的 GSVM 方法.该方法通过定义样本间的相似度度量,并通过相似度来选择适当的聚类算法将数据集划分为多个粒,同时,选择含有较多分类或回归信息的粒进行训练(如使用含有较多支持向量的粒进行训练),从而得到较高精度的学习器.Cheng 等人[16]提出了基于样本和近似最优超平面的距离的方法.该方法同时考虑了样本到近似最优超平面的距离以及近似最优超平面与实际最优超平面的差距.

尽管传统的 GSVM 模型提高了 SVM 的学习效率,使 SVM 能够方便地进行大规模数据的处理,但由于其粒划操作是在统一的粒度层次下执行,可能造成支持向量信息丢失,从而降低了学习器的泛化能力.这主要是因为传统 GSVM 模型在训练之前进行一次性粒划,训练时选取的是处于同一层次的信息样本(如粒中心)来代替整个粒参与训练.然而在实际问题中,数据集在不同区域的分布及重要性往往不同,这就可能导致采用相同的粒划标准而降低了 GSVM 模型的泛化性能.针对分类问题,Yu 等人[17]提出了基于层次粒度的支持向量分类机.该方法在提高支持向量机分类效率的同时保持了较好的泛化能力.尽管支持向量机回归和分类问题一样,需要计算和存储规模庞大的核矩阵,其时间复杂度和空间复杂度同样较高,但目前关于粒度支持向量机处理回归问题的研究[15]相对较少,对大规模的回归问题如何在加速回归过程的同时保持较高的泛化能力还缺乏有效方法.

本文提出一种改进的基于动态粒度的支持向量回归机(dynamical granular support vector regression,简称 DGSVR)方法.该方法首先将原始数据映射到高维核空间,从而将隐藏在原始样本空间的分布特征显现出来,然

后根据样本在核空间的分布分为多个粒,并将最远离近似回归超平面的粒提取出来,根据其密度和半径进行深层次粒划.如此循环往复,从而根据粒的重要性得到在不同粒划层次上的样本粒.最后,使用不同层次、不同细化程度的粒训练得到回归平面.与传统 GSVR 方法相比,本文提出的 DGSVR 模型能够在保持较高回归效率的同时提高传统 GSVR 模型的泛化能力.

1 粒度支持向量回归机

支持向量机是基于最优化理论、在高维特征空间实现线性可分的学习方法.SVM 不仅可以应用于分类问题,而且通过引入ε-不敏感损失函数,近年来广泛地应用于回归问题[18].SVR 首先将输入样本x通过非线性映射ϕ映射到高维特征空间,从而在这个特征空间求解一个线性回归问题.与分类问题类似,回归问题也需要构造核矩阵并求解一个凸二次规划问题,其时间复杂度和空间复杂度分别为 $O(l^3)$和 $O(l^2)$,其中,l 为回归样本集规模.对于大规模数据集,经典 SVR 常常无法直接训练.

为了解决 SVR 训练效率低下的问题,结合粒度计算理论与统计学习理论,提出了粒度支持向量回归机(GSVR)模型[15,19,20].典型的基于聚类的粒度支持向量回归机(clustering based GSVR,简称 CGSVR)模型[15]采用定义的启发式规则来约减SVR的训练集(见算法1),即首先将整个样本集通过聚类的方式分为多个组,对于每组样本只保留中心样本进行训练得到回归超平面.通过这种方式,SVR 的训练集规模减小,训练时间有效减少.

算法 1. 基于聚类的粒度支持向量回归机.

Step 1. 给定初始样本集 $T=(X,D)=\{(x_i,d_i),i=1,\ldots,l\}$以及粒划参数 k.

Step 2. 将样本集聚为 k 个类(粒),即 $X\rightarrow\{G_1,\ldots,G_k\}$.

Step 3. 对每个粒 G_j,计算其粒心μ_j,并且选择与粒心相似度最高的样本点 x'_j 作为训练集中的样本,将 x'_j 的观察值作为训练集中相应的观察值,从而构造出新的训练集 T^*.两个样本相似度的计算如下:

$$S(x,y)=g(1/\|x-y\|_2)=g\left(1\Big/\sqrt{\sum_{i=1}^{M}(x_i-y_i)^2}\right) \tag{1}$$

为了简化问题,可直接取 $g(t)=t$.

Step 4. 采用新构造的训练集来训练 SVR 模型.

Step 5. 测试 SVR 回归超平面,算法结束.

假设函数 f 为标准 SVR 模型在整个数据集 T 上得到的回归超平面,f^*为 SVR 模型在采用聚类方法压缩后的新训练集 T^*上得到的回归超平面.作为目前支持向量机回归问题加速研究中最典型的粒度支持向量机回归模型,基于聚类的粒度支持向量机回归方法(CGSVR)采用粒中心或其近似样本代替整个粒参与训练,尽管简单、有效,但可能会使 T^*的分布与 T 完全不同,从而导致泛化性能的降低.图 2 为 CGSVR 回归可能导致的错误示意图.图中空心圆圈为训练样本,f 为标准 SVR 在整个训练集上得到的回归超平面.若采用 CGSVR 模型,将训练样本通过粒划得到 3 个粒 G_1,G_2 和 G_3,粒心用黑色的正方形表示,则 f^*为 CGSVR 在粒心上训练得到的超平面.由图 2 可知,CGSVR 得到的回归超平面与标准 SVR 得到的最优回归超平面相比,存在较大的、直接影响了学习器的泛化性能.

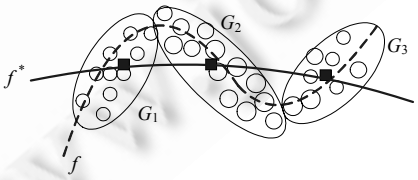


Fig.2 Wrong regression result of CGSVR

图 2 CGSVR 得到错误的回归结果

为了进一步提高 CGSVR 模型的泛化能力,本文从 3 个方面进行了改进:

(1) 将粒的划分、代替及变换衡量均在核空间进行,消除粒映射前后分布的不一致性.

(2) 对于不同密度和重要性的粒,在不同的粒度层次下进行划分代替.即,当粒含有较多重要的回归信息时,需要进行较细的粒划分(甚至将一个样本作为一个粒),从而得到更多含有重要回归信息的样本;而当粒含有较少回归信息时,则在较粗的粒度层次下进行划分,大大压缩训练集的规模,加速 SVR 算法的训练过程.

(3) 在不同的粒度层次上提取合并回归信息.通过这些方法,可以减少由于粒划分而导致的分布不一致的问题,从而提高学习器的泛化能力.

2 基于动态粒度的支持向量回归机

目前,关于 GSVR 的研究大多数采用的是静态粒度(单层粒划)的方法,基于动态粒度(多层粒度)的 GSVR 方法还缺乏研究.本文所提出的 DGSVR 方法是对典型的基于聚类的静态粒度 SVR 方法(CGSVR 方法)[16]的改进.对于回归问题,在回归间隔边界上的点最有可能成为最终的支持向量,因此最为重要;而位于分类间隔内部的点成为支持向量的可能性较小,因此也相对较为次要.本文提出的 DHSVR 方法首先将原始数据映射到高维核空间,从而将隐藏在原始样本空间的分布特征显现出来;然后根据样本在核空间的分布分为多个粒,并将最远离近似回归超平面的粒提取出来,根据其密度和半径进行动态的深层次粒划,尽可能地在最终的训练集中保留多数的重要样本,删除大量对于回归决策不重要的样本,从而在不同粒划层次上使用不同细化程度的粒训练得到高效的回归平面.

2.1 基于核的初始粒划分

对原始训练集 $T=(X,D)=\{(x_i,d_i),i=1,\ldots,l\}$,$x_i\in R^n$,$d_i\in R$,经过非线性映射设 ϕ,样本在高维空间 R^N 中表示为 $T'=\{(\phi(x_i),d_i),i=1,\ldots,l\}$,将样本划分为 k 个粒 $G_1,\ldots,G_k$,其中,$G_i=\{(\phi(x_{ij}),d_{ij}),i=1,\ldots,k;j=1,\ldots,n_i(n_i$ 为第 i 个粒中样本个数)$\}$.可将每个粒看作一个超球,其中心和半径定义如下:

定义 1(核空间粒超球的中心和半径). 将粒划分后形成的任一 N 维样本粒 X_i 称为一个粒超球(为简便起见,本文仍将粒超球记作 X_i),其中心(粒心)μ_i 和半径 r_i 分别为

$$\mu_i=\frac{1}{n_i}\sum_{p=1}^{n_i}\phi(x_p)=\sqrt{\frac{1}{n_i^2}\left(\sum_{p=1}^{n_i}\phi(x_p)\right)^2}=\frac{1}{n_i}\sqrt{\sum_{\substack{p=1\\q=1}}^{n_i}K(x_p,x_q)} \tag{2}$$

$$r_i=\max_{x_s\in G_i}(\phi(x_s)-\mu_i)=\max_{x_s\in G_i}\sqrt{(\phi(x_s))^2-2\phi(x_s)\cdot\mu_i+\mu_i^2}=\max_{x_s\in G_i}\sqrt{K(x_s,x_s)-\frac{2}{n_i}\sum_{p=1}^{n_i}K(x_s,x_p)+\frac{1}{n_i^2}\sum_{\substack{p=1\\q=1}}^{n_i}K(x_p,x_q)} \tag{3}$$

根据定义 1,N 维空间中任一样本 $\phi(x_j)$ 到第 i 个粒超球 G_i 的距离为

$$d(\phi(x_j),G_i)=\sqrt{K(x_j,x_j)-\frac{2}{n_i}\sum_{p=1}^{n_i}K(x_j\cdot x_p)+\frac{1}{n_i^2}\sum_{\substack{p=1\\q=1}}^{n_i}K(x_p,x_q)} \tag{4}$$

本文采用粒超球及其相关度量来进行粒划分,粒划分算法如下:

算法 2. 基于核的粒划分.

Step 1. 给定初始样本集 T 以及粒划参数 k.

Step 2. 任意选择 k 个样本点作为粒心.

Step 3. 按照公式(4)中样本点距离一个粒的距离公式来对所有样本点采用核空间近邻法进行粒划分.

Step 4. 依据公式(2)调整粒心,观察粒心是否有变化(或者变化是否小于某个范围 θ),如果有变化(或者变化大于 θ),则返回 Step 3;否则,转 Step 5.

Step 5. 算法结束,得到划分粒集 $X\to\{G_1,\ldots,G_k\}$.

2.2 动态粒划

假设粒 G_i 的中心和半径分别为 μ_i 和 r_i.在 CGSVR 算法中,其中心 μ_i(或 μ_i 附近样本)被选为训练样本.SVR 在所有粒中心上训练得到的近似回归超平面为 f_1.通过衡量样本粒到初始近似超平面 f_1 的距离及粒的密度进行选

择性的深层次粒划,以改进近似最优回归超平面.为了方便阐述 DGSVR 模型,首先给出粒到超平面距离的定义.

定义 2(粒到超平面的距离). N 维空间的粒 G_i 到超平面的 $f:y=w\cdot\phi(x)+b$ 的距离为

$$d(G_i,f)=\frac{(w,-1)\cdot\mu_i^T+b}{\sqrt{w^2+1}}-r_i=\frac{\frac{1}{n_i}\sum_{k=1}^{n_i}\sum_{j=1}^{|SVs|}\alpha_j y_j K(x_j,x_k)+b}{\sqrt{\sum_{j=1}^{|SVs|}\sum_{k=1}^{|SVs|}\alpha_j\cdot\alpha_k\cdot y_j\cdot y_k\cdot K(x_j,x_k)}}-r_i \tag{5}$$

其中,SVs 是支持向量集合.下面给出信息粒和粒密度的概念.

在粒度支持向量回归机模型中,假设存在近似回归超平面 $f:y=w\cdot\phi(x)+b$ 及划分得到的粒 G_i 和 G_j,回归的间隔为 2η,根据定义 2 求得它们到近似回归超平面 f 的距离分别为 $d(G_i,f)$与 $d(G_j,f)$.在传统支持向量回归机中,支持向量往往落在回归间隔的边界上,因此,若粒 G_i 落在回归间隔的边界上($d(G_i,f)>\eta-2r_i$),那么其包含支持向量的可能性较大,因此其对于回归就更重要;反之,若一个粒 G_j 落在回归间隔内($d(G_i,f)<\eta-2r_i$),那么其包含支持向量的可能性较小,其对于回归问题就不太重要.根据这个原理,本文中引入了信息粒的概念.

定义 3(信息粒). 假设存在粒 G_i,其粒心和半径分别为 μ_i 和 r_i 以及近似的回归超平面 $f:y=w\cdot\phi(x)+b$,若粒 G_i 到回归平面 f 的距离 $d(G_i,f)$大于或等于 $\eta-2r_i$(其中,η是 SVR 回归间隔),则粒 G_i 为信息粒.

由于传统 SVR 模型中支持向量在超平面间隔边界上,且其与超平面的距离等于 η,根据定义 3 可知,粒与超平面 f 的回归间隔区域有重叠或者在间隔之外时为信息粒.此外,当粒内样本分布疏密不均时,我们需要考虑粒密度对于结果的影响.若一个粒密度较大,则根据独立同分布原则,分布在该粒区域内及附近的测试集数量规模也较为庞大,因此,其错分的代价也较大,因此需要在更细的层次上进行学习,以获得足够多的训练样本信息,从而增加模型的测试精度.若一个粒密度较小,分布在该粒区域内及附近的测试集数量规模也较小,其错分的代价较小,因此可以在较粗的粒下进行训练,以减小训练集规模,压缩支持向量回归机的训练时间.

定义 4(粒密度). 假设存在粒 $G_i=\{x_{ij}\}(j=1,\ldots,n_i)$,其粒心和半径分别为 μ_i 和 r_i,所含样本数为 n_i,则粒 G_i 的密度 ρ_i 定义为

$$\rho_i=\frac{n_i}{\sum_{j=1}^{n_i}d(\mu_i,x_{ij})}=\frac{n_i}{\sum_{j=1}^{n_i}\sqrt{\frac{1}{n_i}\sum_{p=1}^{n_i}\sum_{q=1}^{n_i}K(x_p,x_q)-\frac{2}{n_i}\sum_{p=1}^{n_i}K(x_p,x_{ij})+K(x_{ij},x_{ij})}} \tag{6}$$

DGSVR 模型首先将初始样本集 T_0 划分得到第 1 粒度层次上的粒集$\{G_{1_i}\}(i=1,\ldots,k_1)$,其中,粒 G_{1_i} 所对应的粒心和半径分别为 μ_{1_i} 和 r_{1_i},假设在初始训练集$\{\mu_{1_i},d_{1_i}\}$上得到的回归超平面为 f_1,以上涉及字母下标中的 1 表示当前的粒为第 1 层次上的粒.然后,根据定义 3 提取信息粒集 $\{G'_{1\_j}\}(\{G'_{1\_j}\}\subseteq\{G_{1\_i}\};i=1,...,k_1,j=1,...,k'_1)$, 其中,信息粒 G'_{1_j} 所对应的粒心和半径分别为 μ'_{1_j} 和 r'_{1_j}.然后计算每个信息粒 G'_{1_j} 的粒密度 ρ'_{1_j}.假设 G'_{Lev_j} 表示第 Lev 层上的信息粒.显然,其密度 ρ'_{Lev_j} 较大时,该粒包含更多的回归信息,因此需要对该粒进行深层次的粒划且粒划参数较大,在下一层次中得到更多的回归信息.同理,若第 Lev 层的信息粒 G'_{Lev_j} 的半径 r'_{Lev_j} 较大,那么直接采用中心代替可能导致泛化性能下降较多,泛化性能可能不好,需要对该粒进行深层次的粒划以得到更多回归信息.据此,引入动态粒划参数的概念,假设存在第 Lev 层的信息粒 G'_{Lev_j} 并对信息粒 G'_{Lev_j} 按照公式(7)计算得到的粒划参数进行第 Lev 层的动态粒划:

$$k'_{Lev\_j}=\left\lceil\frac{r'_{Lev\_j}\times\rho'_{Lev\_j}}{d\_para}\right\rceil \tag{7}$$

其中,d_para 为动态粒划参数,用来控制不同密度数据集的动态粒划进程,其通过网格搜索的方式进行设置.当训练集规模大于 100 时,动态粒划参数分别取[1.5,2,2.5]进行搜索;当训练集规模小于 100 时,d_para 分别取[1,1.25,1.5]进行搜索.这种不断迭代的动态粒划过程一直进行,直到所有信息粒的粒划参数均为 1 时停止.在动态粒划过程中,每一层次的 SVR 模型均采用本层所有粒(包含信息粒与非信息粒)的粒心进行训练来得到本层的回归超平面.因此,在动态粒划的过程中,每层的粒个数(SVR 在该层的实际训练集规模)符合如下规律:

$$k_{Lev+1} = k_{Lev} - k'_{Lev} + \sum_{j=1}^{k'_{Lev}} k'_{Lev\_j} \tag{8}$$

假设动态粒划过程结束后得到最终粒个数为 k_{last},然后,SVR 在最终得到的不同层次粒心构成的训练集上进行训练,得到最终回归超平面 f_{last}.

为了更好地阐述 DGSVR 模型,首先给出关于其泛化性能的分析.假设初始回归样本集为 $T=(X,D)=\{(x_i,d_i), i=1,\ldots,l\}$,且 $x_i\in R^n, d_i\in R$.假设一种粒度支持向量回归机模型通过将训练样本划分为 k 个粒,同时构造了压缩的训练样本集 $T'\subseteq T$,且压缩的训练样本集 T'的规模为 $l'(l'<<l)$.假设在压缩后的训练集上得到的回归超平面为 f',那么 f'的经验风险如下:

$$R_{emp}[f'] = \frac{1}{l'}\sum_{i=1}^{l'} c(x_i, d_i, f'(x_i), \varepsilon) \tag{9}$$

这里,$c(x_i,d_i,f'(x_i),\varepsilon)$为 ε回归损失函数.这里,首先引入一般 GSVR 模型误差的概念:

定义 5(模型误差). 在粒度支持向量回归机中,假设原始回归数据集 T 经过粒划、重构、代替、压缩后所得新训练集为 T',且在原始训练集 T 和压缩后的训练集 T'上训练得到的回归超平面分别为 f 和 f',则模型误差为

$$E_M = \lim_{l,l'\to\infty} |R[f'] - R[f]| \tag{10}$$

显然,模型误差可以用来衡量学习器的回归逼近能力.对于原始的数据集 T,采用 DGSVR 模型得到的训练集为 T',而采用 CGSVR 模型得到的训练集为 T''.由于 DGSVR 模型在不同层次上进行数据的动态粒划和压缩,因此其保留了更多的原始支持向量信息,而 CGSVR 模型采用单一层次的静态压缩,压缩后的数据集丢失了较多的支持向量信息,因此,采用 DGSVR 模型得到的压缩后的训练集更逼近于原始数据集中重要样本的分布,也能够使与原始训练集独立同分布的测试集得到更高的泛化能力,使其逼近标准 SVR 模型的性能[2].

DGSVR 模型采用动态层次的方式进行粒划,根据样本与超平面的分布动态地调整粒划的进程,提取了更多重要的回归信息.这里需要注意的是,一个粒当前为非信息粒,但随着超平面的改进,在后续动态粒划过程中可能随着超平面的调整变为信息粒.动态粒划的过程如图 3 所示.

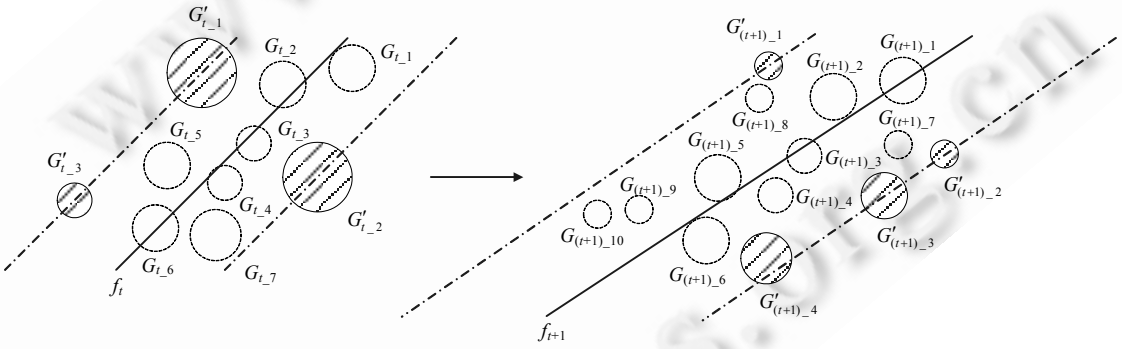


Fig.3 Dynamical granulation procedure of DGSVR model

图 3 DGSVR 模型的动态粒划过程

图 3 中,f_t 为在第 t 层粒上训练得到的回归超平面.由定义 2 知,$G_{t\_1}, G_{t\_2}, G_{t\_3}, G_{t\_4}, G_{t\_5}, G_{t\_6}$ 和 G_{t_7} 为该层的非信息粒,G'_{t_1}, G'_{t_2} 和 G'_{t_3} 是该层的信息粒.采用算法 2 对信息粒进行粒划分,其中,G'_{t_1} 划分为 $G'_{(t+1)\_1}$ 和 $G_{(t+1)\_8}$,G'_{t_2} 划分为 $G'_{(t+1)\_2}, G'_{(t+1)\_3}$ 和 $G_{(t+1)\_7}$,G'_{t_3} 划分为 $G_{(t+1)\_9}$ 和 $G_{(t+1)\_10}$.在新的粒上得到更新的回归超平面 f_{t+1},且新的信息粒为 $G'_{(t+1)\_1}, G'_{(t+1)\_2}, G'_{(t+1)\_3}$ 和 $G'_{(t+1)\_4}$.因此,每层粒划产生的信息粒包括两类,即由上层信息粒划分得到的子粒(如 $G'_{(t+1)\_1}, G'_{(t+1)\_2}, G'_{(t+1)\_3}$)或者由于超平面更新而导致其他非信息粒转化为信息粒(如 $G'_{(t+1)\_4}$).

2.3 DGSVR算法

DGSVR 模型将原空间训练样本映射到高维特征空间,使原来隐藏的分部信息在特征空间显现出来,从而对训练样本进行初步的粒划分;然后提取出那些远离近似回归超平面的粒,并通过计算它们的密度和半径,衡量其

所含有的回归信息的重要性,对于重要的粒,通过迭代的动态分层粒划方法,可以保持大多数的重要回归信息,减小由于粒划分代替而带来的分布不一致的误差;最后,通过处于不同层次的粒来抽取训练样本,进行SVR训练并得到回归超平面.在 DGSVR 算法中,含有较少回归信息的粒往往划分层次较浅,从而有效地减小了训练集规模,提高了训练效率;而含有较多回归信息的粒往往划分层次较深,有效地保留了含有大量回归信息的样本,保证了模型的泛化能力.

本文主要探讨 DGSVR 模型是否能够通过动态层次粒划的方法同时获得较高的训练效率和泛化性能,因此,对于如何选择合适的核及其参数不进行讨论,相关问题可参考文献[21−23].本文提出的 DGSVR 可以结合相关方法进行核选择与确定.本文全部采用高斯核进行实验.DGSVR 算法主要步骤如下:

算法 3. 基于动态粒划的 SVR 算法.

初始化:假设给定的训练集为 $T_0=(X_0,D)=\{(x_i,d_i)\}(i=1,\ldots,l_0)$且 $x_i\in R^n,d_i\in R$.初始粒划参数为 k_0,核函数采用高斯核,初始化粒划层次参数 $Lev=0$ 和动态粒划参数 d_para.

Step 1. 初始粒划并进行 SVR 训练.

对初始训练集 T_0 中的样本集 X_0 采用用算法 2 进行粒划,并得到粒划结果 $X_0\to\{G_{1\_1},\ldots,G_{1\_k}\}$,其中,$G_{1_i}=\{\phi(x_i)\}(i=1,\ldots,l_{1_i})(l_{1_i}$为第 1 层第 i 个粒中的样本规模,并更新粒划层次参数 $Lev=1$.

Step 2. 动态粒划.

Step 2.1. 在第 Lev 粒划层次上抽取训练样本集(本文采用抽取粒心的方法构造训练样本集)$\{\mu_{Lev\_1},\ldots,\mu_{Lev\_k_{Lev}}\}$ 上进行 SVR 训练,并得到近似回归超平面 f_{Lev};

Step 2.2. 按照公式(5)计算每个粒到近似回归超平面的距离,并结合定义 3 挑选第 Lev 粒划层次上的信息粒 $\{G'_{Lev\_j}\}(j=1,\ldots,k'_{Lev})$;

Step 2.3. 按照公式(6)计算第 Lev 粒划层次上的信息粒的密度 ρ'_{Lev_j},按照公式(7)计算 Lev 粒划层次上的动态粒划参数 k'_{Lev_j},k'_{Lev_j} 代表第 Lev 粒划层次上的第 j 个信息粒的动态粒划个数;

Step 2.4. 对 Step 2.3 计算得到的粒划个数大于 2 的信息粒,采用算法 2 进行粒划,更新粒划层次参数 $Lev=Lev+1$;

Step 2.5. 若 Step 2.4 执行过程中有新粒产生,则计算其粒心和半径,然后转到 Step 2.1 继续执行;否则,转到 Step3.

Step 3. 最终 SVR 训练.

对最后得到的各个不同层次的粒求取粒心,并采用粒心进行 SVR 训练得到最终的回归超平面 l_{last}.

Step 4. 算法结束.

由于标准的 SVR 模型需要计算和存储大规模的核矩阵,因此其算法的时间复杂度和空间复杂度都很高,分别是 $O(n^3)$和 $O(n^2)$,其中,n 为参与训练样本集的规模.因此,模型的训练过程非常缓慢.特别地,当训练集的数据量较大时,传统 SVR 方法往往无法在有效时间内得到学习模型.因此,与传统 SVR 模型相比,GSVR 模型的粒划分过程尽管有一定的时间开销,但由于 GSVR 模型大幅度地压缩了实际训练集的规模,提高了模型的学习效率,因此,初始粒划的时间开销可以忽略不计.但是由于传统 CGSVR 模型采用单层的粒划分方法,不能有效地抓住训练集中的关键样本,在样本压缩时容易导致含有重要回归信息的样本被错误地删除,从而严重影响了模型的泛化能力.本文提出的 DGSVR 方法尽管采用了迭代的动态粒划分方法,但由于多数位于回归间隔内的粒由于不包含重要的回归信息,因此在进行一次粒划分后就停止迭代,只有少数位于回归间隔边缘上的粒因为含有重要的回归信息(支持向量信息)而需要进行多次的迭代,因此其动态粒划的过程时间不会太长.与传统 CGSVR 模型相比,DGSVR 模型在保持较高学习效率的同时能够保留重要的回归信息,有效提高模型的泛化能力.

3 数值实验与分析

3.1 实验数据集

本文在两类数据集上进行了实验:(1) 基准函数数据集,见表 1(每个数据集包含 1 000 个训练样本和 500 个

测试样本);(2) UCI 标准回归数据集[24],见表 2.SVR 的高斯核参数取 1.0,惩罚因子取 200.实验在一台 CPU 为 2.66Ghz,内存为 1G 的计算机上运行,实验平台为 Matlab2008.

Table 1 Benchmark function datasets

表 1 基准函数数据集

数据集	基准函数	分布区间
2D Mexican Hat	$y=\sin c\|x\|=\sin\|x\|/\|x\|$	$x\sim U[-10,10]$
3D Mexican Hat	$y=\sin c\sqrt{x_1^2+x_2^2}=\sin\sqrt{x_1^2+x_2^2}\Big/\sqrt{x_1^2+x_2^2}$	$x_1,x_2\sim U[-4\pi,4\pi]$
Friedman #1	$y=10\sin(\pi x_1x_2)+20(x_3-0.5)^2+10x_4+5x_5$	$x_1,x_2,x_3,x_4,x_5\sim U[0,1]$
Gabor	$y=(\pi/2)\exp[-2(x_1^2+x_2^2)]\cos[2\pi(x_1+x_2)]$	$x_1,x_2\sim U[0,1]$
Multi	$y=0.79+1.27x_1x_2+1.56x_1x_4+3.42x_2x_5+2.06x_3x_4x_5$	$x_1,x_2,x_3,x_4,x_5\sim U[0,1]$
Plane	$y=0.6x_1+0.3x_2$	$x_1,x_2\sim U[0,1]$
Polynomial	$y=1+2x+3x^2+4x^3+5x^4$	$x\sim U[0,1]$
SinC	$y=\sin(x)/x$	$x\sim U[0,2\pi]$

Table 2 UCI regression datasets

表 2 UCI 回归数据集

数据集	#训练集	#测试集	#特征
Concrete	500	530	8
Forest_fires	317	200	12
Slump	53	50	7
Winequality_red	1 000	599	11
Winequality_white	2 000	2 898	11

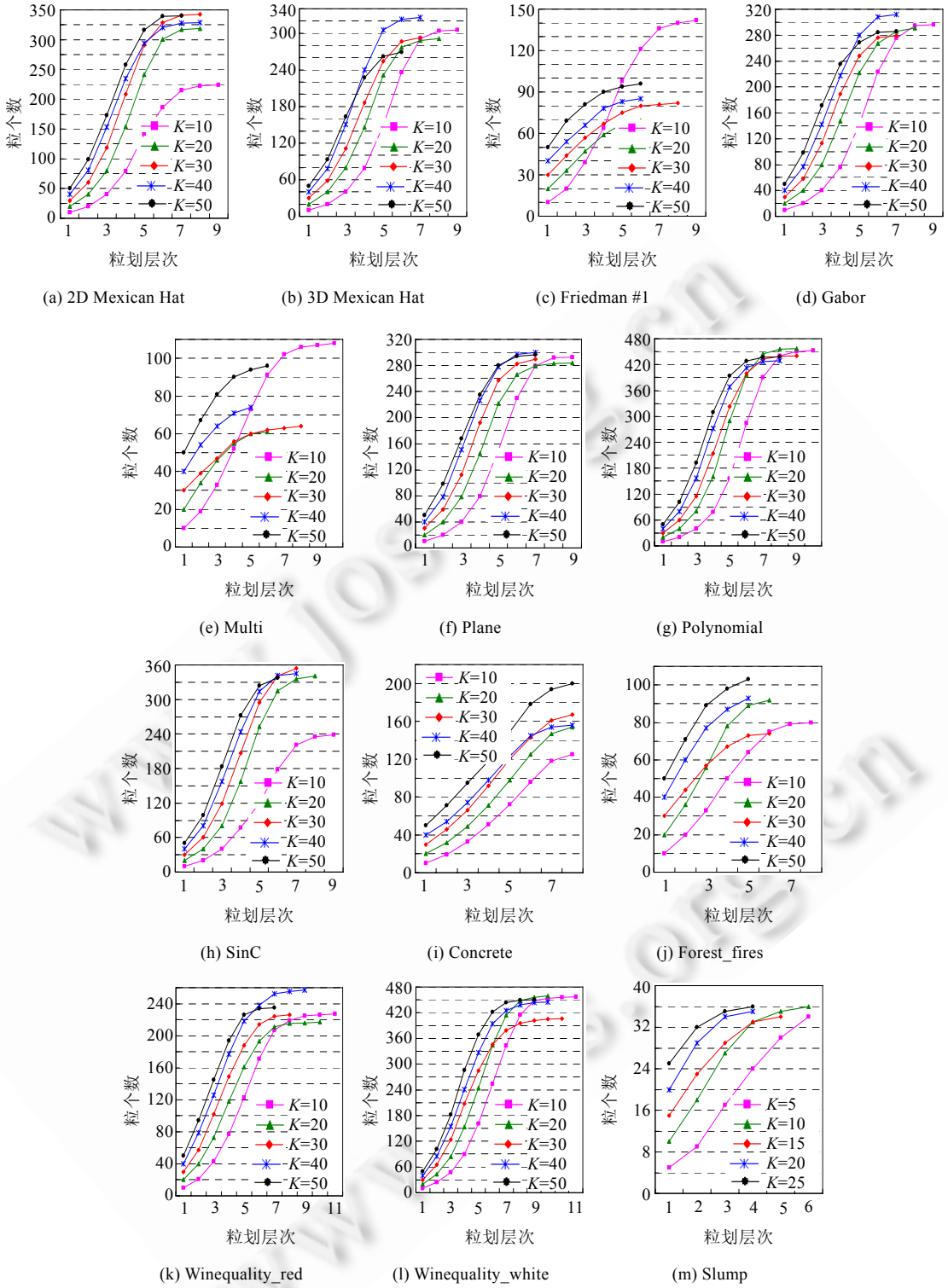
3.2 粒划结果分析

对于任何 GSVR 模型,粒划是其关键步骤,它决定了最终实际训练集的规模及提取的回归信息的多少,从而直接影响到学习器的训练效率和泛化性能.若粒划结果中得到粒数目过多,那么实际参与的训练集规模依然很大,达不到加速算法的目的;但如果得到粒数目太小,则又可能导致丢失重要回归信息,影响模型的泛化性能.因此,首先对 DGSVR 模型的粒划结果与传统 CGSVR 模型进行实验比较.

实验中检测了每个粒划层次下的粒个数变化情况.由于动态粒划过程受到初始粒划参数 k_0 的影响,这里除 UCI 回归数据集中的 Slump 外,初始粒划参数分别取 10,20,30,40 和 50 进行了测试,由于 Slump 数据集较小,其初始粒划参数设置为 5,10,15,20,25.回归参数 ε 均取 0.1.图 4 为两类数据集在不同的粒化层次下粒数目的变化统计(这里仅列出 *d_para* 参数在 Slump 数据集上取 1.0 和在其他数据集上取 1.5 的结果).图中横轴代表动态粒划层次,纵轴代表粒数目.

从图 4 中可以看出,对于所有数据集,随着动态粒划层次的增加,粒数目也在增加,且粒个数增加趋势在初始动态粒划分时增加较快,而在最后动态粒划过程中增加较慢,其参数 *d_para* 在其他设置下也得到了类似的结果.这是由于在初始动态粒划时,许多信息粒半径较大,动态粒划过程的粒划分个数较多,因此增加较快;但随着动态粒划过程的深入,信息粒半径减小,粒划分个数减少,因此增长过程变慢;其次,除了 UCI 回归数据集中的 Winequality_red 数据集和 Winequality_white 数据集(*d_para*=1.5)以外,其他情况下动态粒划层次数都在 10 以内.此外,实验中我们还发现:随着 *d_para* 的增长,动态粒个数和动态粒划层次数均减少;特别地,当 *d_para*=2.5 时,数据集 3D Mexican Hat,Friedman #1,Gabor,Multi 只进行了 1 次粒划分.

因此,从动态粒划结果中可以发现:首先,在动态粒划过程的后期,在有效提取重要回归信息的同时,训练样本规模并没有大幅度增加,从而保证了模型的训练效率;其次,DGSVR 模型能够在较少的动态粒划迭代后收敛.在实际应用中,可以根据问题精度和效率的需求情况来选择合适的动态粒度参数,以使模型在训练效率和泛化性能间有更好的折中.



(a) 2D Mexican Hat (b) 3D Mexican Hat (c) Friedman #1 (d) Gabor

(e) Multi (f) Plane (g) Polynomial

(h) SinC (i) Concrete (j) Forest_fires

(k) Winequality_red (l) Winequality_white (m) Slump

Fig.4 Dynamical granulation results

图 4 动态粒划结果

3.3 训练及测试结果比较

将 DGSVR 与标准的 SVR 模型、基于静态单次粒划的 CGSVR 模型在训练时间和测试精度两个方面进行了对比实验.参数设置与第 3.2 节一致.为了简化实验结果,这里仅给出了在 UCI 回归数据集上的测试精度和训练时间,分别见表 3 和表 4.

Table 3 Testing accuracy of UCI regression datasets (%)

表 3 UCI 回归数据集测试精度 (%)

数据集	模型	K_0=10			K_0=20			K_0=30			K_0=40			K_0=50		
		d_para =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5
Concrete	DG-SVR	**94.406**	**94.403**	**94.403**	94.388	94.388	94.388	94.388	94.388	94.388	94.388	94.384	94.384	94.388	94.384	94.384
	CG-SVR	**94.374**			94.374			94.374			94.374			94.374		
	SVR	**94.406**														
Forest_ fires	DG-SVR	**92.8766**	**92.8766**	**92.846**	92.830	92.830	92.830	92.830	92.830	92.830	92.830	92.830	92.830	92.830	92.830	92.830
	CG-SVR	**92.830**			92.830			92.830			92.830			92.830		
	SVR	**92.877**														
Wine_ red	DG-SVR	**99.215**	99.119	99.096	99.198	99.116	99.076	99.197	99.101	99.082	99.200	99.121	99.100	99.193	**99.122**	**99.109**
	CG-SVR	99.064			99.070			99.080			99.099			**99.107**		
	SVR	**99.382**														
Wine_ white	DG-SVR	**99.810**	**99.805**	**99.804**	99.806	99.800	99.799	99.805	99.800	99.799	99.805	99.800	99.799	99.805	99.800	99.800
	CG-SVR	99.798			99.798			99.799			99.799			**99.799**		
	SVR	**99.821**														

数据集	模型	K_0=5			K_0=10			K_0=15			K_0=20			K_0=25		
		d_para =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5
Slump	DG-SVR	**62.692**	**62.516**	**62.428**	62.428	62.428	62.428	62.428	62.428	62.428	62.428	62.428	62.428	62.428	62.428	62.428
	CG-SVR	**62.428**			62.428			62.428			62.428			62.428		
	SVR	**62.692**														

由于 CGSVR 模型只有 1 个参数 K_0,因此 CGSVR 在每个 K_0 值下只有 1 个测试结果;而 SVR 模型不受参数 K_0 和 *d_para* 的影响,因此 SVR 在每个数据集上只有 1 个测试值.在表 3 中,DGSVR 方法所对应的每个数据集中 3 个带下划线的粗体值分别为不同动态粒度参数下的测试精度最优值(当出现多个测试精度最大值相等时,取训练时间最小时所对应的值作为测试精度最优值).CGSVR 方法所对应的粗体值为其在不同粒度划分参数下得到的精度最优值.为了方便比较,表中标准 SVR 得到的测试精度值也进行加粗.从表 3 可看出,DGSVR 方法在所有数据集上比传统 CGSVR 方法的回归精度都有了明显的提高,而且在许多数据集上,DGSVR 方法的测试精度已经接近于标准 SVR 所得到的测试精度.

从表 4 可以看出:与 CGSVR 相比,DGSVR 方法的学习效率由于样本规模的增加有所减小;但与标准 SVR 模型相比,DGSVR 模型的效率是可以接受的.由于 SVR 模型的训练时间是由实际参与训练的样本规模决定的,而标准 SVR 求解过程要求解一个规模为 l^2 的核矩阵(其中,l 为训练样本规模),传统 SVR 方法在整个训练集上直接进行训练,因此,标准 SVR 的训练时间特别长,学习效率非常低.而传统 CGSVR 方法采用一次性静态粒划的方法,压缩了大量的数据集,学习效率得到了大幅度提高,但由于删除了大量对于 SVR 非常重要的边界信息,因此其泛化性能有明显的下降.尽管 DGSVR 在动态粒划的过程中为了提取重要的回归信息而比 CGSVR 保留了略多的训练样本(这一点从图 3、图 4 中最后得到的粒个数与初始粒个数差异中也可以反映出来,因为 DGSVR 的训练样本规模实际上就是最终粒个数,而 CGSVR 的训练样本规模为初始粒个数),模型训练时间比 CGSVR 略长.

Table 4 Training time of UCI regression datasets (s)

表 4 UCI 回归数据集训练时间 (s)

数据集	模型	K_0=10			K_0=20			K_0=30			K_0=40			K_0=50		
		d_para =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5
Concrete	DG-SVR	**1.2188**	**0.0156**	**0.0156**	2.8750	0.0469	0.0313	2.7656	0.1250	0.0781	1.9375	0.2344	0.1406	3.7031	0.4063	0.2969
	CG-SVR	**0.0156**			0.0469			0.0781			0.1250			0.2188		
	SVR	**127.156**														
Forest_ fires	DG-SVR	**0.4438**	**0.0313**	**0.0156**	0.6875	0.0938	0.0313	0.3406	0.1719	0.0781	0.6719	0.2656	0.1563	0.8250	0.4531	0.2969
	CG-SVR	**0.0156**			0.0313			0.0781			0.1563			0.2969		
	SVR	**52.813**														
Wine_ red	DG-SVR	**10.0313**	0.4688	0.0313	8.5313	0.8438	0.1250	12.4531	1.2500	0.1719	14.9219	1.0938	0.2969	11.0781	**1.1063**	**0.5611**
	CG-SVR	0.0156			0.0625			0.1563			0.2969			**0.5000**		
	SVR	**1362.688**														
Wine_ white	DG-SVR	**123.66**	**1.5000**	**0.1406**	109.14	2.3594	0.2656	91.350	3.0469	0.4844	125.06	4.4625	0.8125	125.77	5.0781	0.8281
	CG-SVR	**0.0156**			0.0625			0.2031			0.3906			0.7031		
	SVR	**12261.844**														
数据集	模型	K_0=5			K_0=10			K_0=15			K_0=20			K_0=25		
		d_para =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5	*d_para* =1.5	*d_para* =2	*d_para* =2.5
Slump	DG-SVR	**0.0438**	**0.0156**	**0.0156**	0.0424	0.0156	0.0156	0.0537	0.0313	0.0156	0.0538	0.0469	0.0313	0.0738	0.0625	0.0469
	CG-SVR	**0.0156**			0.0156			0.0156			0.0313			0.0469		
	SVR	**0.2500**														

在基准函数数据集上也得到了类似的结果.即与 CGSVR 模型相比,DGSVR 模型在提高标准 SVR 学习效率的基础上,进一步减小了 CGSVR 的模型误差,提高了其泛化性能.

为了更好地反映 DGSVR 在测试精度和训练时间两个方面所取得的整体折中情况,本文给出训练效率及测试精度相对量.具体定义如下:

定义 6(测试精度相对量). 假设 DGSVR,CGSVR 与标准 SVR 的测试精度分别表示为 A(DGSVR), A(CGSVR)和 A(SVR),则测试精度相对量定义为

$$Re\_acc = \frac{A(\text{DGSVR}) - A(\text{CGSVR})}{A(\text{SVR}) - A(\text{CGSVR})} \tag{11}$$

若 A(SVR)=A(CGSVR)时,说明 CGSVR 已达到标准 SVR 的最优测试精度,此时体现不出 DGSVR 在泛化性能方面的优势,因此当 A(SVR)−A(CGSVR)=0 时,默认 *Re_acc* 为 0.

定义 7(训练效率相对量). 假设 DGSVR,CGSVR 与标准 SVR 的训练时间分别表示为 T(DGSVR),T(CGSVR)和 T(SVR),则训练效率相对量定义为

$$Re\_eff = \frac{T(\text{DGSVR}) - T(\text{CGSVR})}{T(\text{SVR}) - T(\text{CGSVR})} \tag{12}$$

由于 CGSVR 可以明显压缩训练样本规模,所以 T(SVR)与 T(CGSVR)一般不会相等,即 T(SVR)−T(CGSVR)≠0.

显然,*Re_acc* 越大,DGSVR 的测试精度相对于 CGSVR 方法就提高得越多,而且越逼近于标准 SVR 的测试精度,模型的泛化性能越好.特别地,若 *Re_acc*=1,则说明 DGSVR 方法采用少数训练数据集得到了精确的最优回归超平面.而 *Re_eff* 越小,DGSVR 的训练时间的增加量就越小,其相对于 SVR 的训练时间可以忽略不计.特别地,若 *Re_eff* 值为负,则表明 DGSVR 的训练效率比 CGSVR 还要高.本文统计了表 3 和表 4 中最优值所对应的测试精度相对量和训练效率相对量(见表 5).

Table 5 Relative value of testing accuracy and training efficiency

表 5 测试精度与训练效率相对量

数据集	测试精度相对量			训练效率相对量		
	d_para=1.5	*d_para*=2	*d_para*=2.5	*d_para*=1.5	*d_para*=2	*d_para*=2.5
2D Mexican Hat	0	0	0	9.314373×10^{-3}	1.598008×10^{-3}	8.4417×10^{-5}
3D Mexican Hat	0.75	0	0	2.234639×10^{-2}	-9.633×10^{-6}	0
Friedman #1	0.725 191	0	0	2.59121×10^{-3}	-1.1867×10^{-5}	0
Gabor	0.761 905	0.095 238	0.095 238	2.547655×10^{-2}	7.2181×10^{-5}	-1.0368×10^{-5}
Multi	0.666 667	0	0	7.01401×10^{-4}	0	0
Plane	1	1	1	1.684447×10^{-2}	3.3892×10^{-5}	0
Polynomial	1	0	0	6.134505×10^{-2}	6.93653×10^{-4}	4.3353×10^{-5}
SinC	0	0	0	1.055289×10^{-2}	8.6455×10^{-5}	8.629×10^{-6}
Concrete	1	0.906 25	0.906 25	9.463554×10^{-3}	0	0
Forest_fires	0.991 489	0.991 489	0.340 426	8.110248×10^{-3}	2.97363×10^{-4}	0
Wine_red	0.392 727	0.054 545	0.007 273	6.997052×10^{-3}	4.45093×10^{-4}	4.4854×10^{-5}
Wine_white	0.5	0.272 727	0.227 273	1.008368×10^{-2}	1.21059×10^{-4}	1.0194×10^{-5}
数据集	测试精度相对量			训练效率相对量		
	d_para=1	*d_para*=1.25	*d_para*=1.5	*d_para*=1	*d_para*=1.25	*d_para*=1.5
Slump	1	0.333 333	0	1.203072×10^{-1}	0	0

从表 5 可以看出:首先,当 *d_para*=1.5 时,在前 12 个数据集中有 9 个数据集的 *Re_acc* 值均大于等于 0.5;当 *d_para*=2 和 *d_para*=2.5 时,也都存在 6 个数据集的 *Re_acc* 值非 0.此外,在数据集 Slump 上,当 *d_para* 取 1 和 1.25 时,*Re_acc* 值也均在 0.3 以上.尽管在基准函数数据集上也存在不少 *Re_acc* 为 0 的值,但这仅仅是由于多数基准函数数据集相对比较简单.因此,采用 3 种方法得到的测试精度几乎相等,当数据集复杂时(如数据集 Gabor 和所有 UCI 的回归数据集),DGSVR 在测试精度方面的优势非常明显.这充分说明,与传统 CGSVR 算法相比,DGSVR 方法提高了模型的测试精度,使得学习器的泛化性能增强.其次,当 *d_para* 取 1.5 时,*Re_eff* 值大多分布在 10^{-4}~10^{-2} 量级之间;当 *d_para* 取 2 时,*Re_eff* 值大多分布在 10^{-6}~10^{-4} 之间,且在数据集 3D Mexican Hat,Friedman #1,Multi 及 Concrete 上取到了 0 甚至负值,这说明在这些数据集上,DGSVR 方法的训练效率持平甚至超过了 CGSVR 方法;而当 *d_para* 取 2.5 时,取 0 或负值的数据集达到了 7 个.尽管部分基准函数数据集本身的测试精度非常接近或者很容易达到 100%,因此无法显示出 DGSVR 方法的优势,但从所有数据集的训练结果足以看出,本文提出的 DGSVR 方法在保持较高训练效率的同时提高了传统 GSVR 方法的回归测试精度,增强了学习器的泛化能力.

4 总结与展望

针对传统 GSVR 模型无法有效兼顾训练效率和泛化性能的问题,本文提出了一种基于动态粒度的支持向量回归机模型.通过多层次的动态粒划方法,在充分压缩非重要样本的同时,在更细层次上提取了含有回归信息的重要样本,能够在保证方法训练效率的同时有效提高算法的泛化性能.在未来的工作中,将继续研究探讨基于双方向粒划(分解和合并)的动态粒化 SVR 算法,从而根据超平面的调整,及时合并那些误判为含有重要回归信息的粒,进一步提高方法执行的效率,从而更好地解决大规模数据的回归问题.

References:

[1] Li XY. The Research Report of IDC. 2011 (in Chinese with English abstract). http://storage.chinabyte.com/163/12110163.shtml

[2] Vapnik V. Statistical Learning Theory. New York: Wiley, 1998. 493−520.

[3] Zeng ZQ, Gao J. Simplified support vector machine based on reduced vector set method. Ruan Jian Xue Bao/Journal of Software, 2007,18(11):2719−2727 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/2719.htm [doi: 10.1360/jos182719]

[4] Tsang IW, Kwok JT, Cheung PM. Core vector machines: Fast SVM training on very large datasets. Journal of Machine Learning Research, 2005,6:363−392. [doi: 10.1360/jos182719]

[5] Joachims T. Making large-scale SVM learning practical. In: Scholkopf B, Burges C, Smola A, eds. Advances in Kernel Methods-Support Vector Learning. Cambridge: MIT Press, 1998. 169−184.

[6] Li DC, Fang YH. An algorithm to cluster data for efficient classification of support vector machines. Expert Systems with Applications, 2008,34(3):2013−2018. [doi: 10.1016/j.eswa.2007.02.016]

[7] Hao PY, Chiang JH, Tu YK. Hierarchically SVM classification based on support vector clustering method and its application to document categorization. Expert System with Applications, 2007,33(3):627−635. [doi: 10.1016/j.eswa.2006.06.009]

[8] Nath JS, Shevade SK. An efficient clustering scheme using support vector methods. Pattern Recognition, 2006,39(8):1473−1480. [doi: 10.1016/j.patcog.2006.03.012]

[9] Reddy IS, Shevade S, Murty MN. A fast quasi-newton method for semi-supervised SVM. Pattern Recognition, 2011,44(10-11): 2305−2313. [doi: 10.1016/j.patcog.2010.09.002]

[10] Zhong W, He JY, Harrison R, Tai PC, Pan Y. Clustering support vector machines for protein local structure prediction. Expert Systems with Applications, 2007,33(2):518−526. [doi: 10.1016/j.eswa.2005.12.011]

[11] Tang YC, Jin B, Zhang YQ. Granular support vector machines with association rules mining for protein homology prediction. Artificial Intelligence in Medicine, 2005,35(1):121−134. [doi: 10.1016/j.artmed.2005.02.003]

[12] Osuna E, Freund R, Girosi F. Training support vector machines: An application to face detection. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 1997. 130−136. [doi: 10.1109/cvpr.1997.609310]

[13] Platt JC. Fast training of support vector machines using sequential minimal optimization. In: Scholkopf B, *et al*., eds. Advances in Kernel Methods-Support Vector Learning. Cambridge: MIT Press, 1998. 41−64.

[14] Wang WJ, Guo HS, Jia YF, Bi JY. Granular support vector machine based on mixed measure. Neurocomputing, 2013,101:116−128. [doi: 10.1016/j.neucom.2012.08.006]

[15] Wang WJ, Xu ZB. A heuristic training for support vector regression. Neurocomputing, 2004,61:259−275. [doi: 10.1016/j.neucom. 2003.11.012]

[16] Cheng SX, Shih FY. An improved incremental training algorithm for support vector machines using active query. Pattern Recognition, 2007,40(3):964−971. [doi: 10.1016/j.patcog.2006.06.016]

[17] Yu H, Yang J, Han JW, Li XL. Making SVMs scalable to large datasets using hierarchical cluster indexing. Data Mining and Knowledge Discovery, 2005,11(3):295−321.

[18] Vapnik V, Golowich SE, Smola A. Support vector method for function approximation, regression estimation, and signal processing. In: Mozer M, Jordan M, Petsche T, eds. Proc. of the Neural Information Processing Systems, Vol.9. Cambridge: MIT Press, 1997.

[19] Tang YC. Granular support vector machines based on granular computing, soft computing and statistical learning [Ph.D. Thesis]. Georgia Stage University, 2006.

[20] Collobert R, Bengio S. SVMTorch: Support vector machines for large-scale regression problems. Journal of Machine Learning Research, 2001,1:143−146. [doi: 10.1162/15324430152733142]

[21] Wang WJ, Xu ZB, Lu VZ, Zhang XY. Determination of the spread parameter in the Gaussian kernel for classification and regression. Neurocomputing, 2003,55(3-4):643−663. [doi: 10.1016/S0925-2312(02)00632-X]

[22] Ali S, Smith-Miles KA. A meta-learning approach to automatic kernel selection for support vector machines. Neurcocomputing, 2006,70(3):173−186. [doi: 10.1016/j.neucom.2006.03.004]

[23] Wu KP, Wang SD. Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. Pattern Recognition, 2009,42(5):710−717. [doi: 10.1016/j.patcog.2008.08.030]

[24] UCI machine learning repository. 2010. http://archive.ics.uci.edu/ml/datasets.html

附中文参考文献:

[1] 李旭阳.IDC 研究报告.2011. http://storage.chinabyte.com/163/12110163.shtml

[3] 曾志强,高济.基于向量集约简的精简支持向量机.软件学报,2007,18(11):2719−2727. http://www.jos.org.cn/1000-9825/18/2719.htm [doi: 10.1360/jos182719]

郭虎升(1986－),男,山西太谷人,博士生,主要研究领域为机器学习,数据挖掘.
E-mail: chaofei142@163.com

王文剑(1968－),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为机器学习,计算智能,数据挖掘.
E-mail: wjwang@sxu.edu.cn