文章编号: 1001-0920(2017)09-1556-09 DOI: 10.13195/j.kzyjc.2016.0825

二维分割贯序正则化超限学习机

郭 威[1,2], 徐 涛[1,3†], 于建江[2], 汤克明[2]

(1. 南京航空航天大学 计算机科学与技术学院，南京 210016；2. 盐城师范学院 信息工程学院，江苏 盐城 224002；3. 中国民航大学 计算机科学与技术学院，天津 300300)

摘 要: 针对大规模在线学习问题,提出一种二维分割贯序正则化超限学习机(BP-SRELM). BP-SRELM以在线贯序超限学习机为基础,结合分治策略的思想,从实例和特征两个维度对高维隐层输出矩阵进行分割,以降低问题求解的规模和计算复杂性,从而极大地提高对大规模学习问题的执行效率. 同时,BP-SRELM 通过融合使用Tikhonov正则化技术进一步增强其在实际应用中的稳定性和泛化能力. 实验结果表明,所提出的BP-SRELM不仅具有更高的稳定性和预测精度,而且在学习速度上优势明显,适用于大规模数据流的在线学习与实时建模.

关键词: 在线贯序超限学习机；Tikhonov正则化；分割；在线学习；大数据流

中图分类号: TP183 **文献标志码:** A

Bidimensionally partitioned sequential regularized extreme learning machine

GUO Wei[1,2], *XU Tao*[1,3†], *YU Jian-jiang*[2], *TANG Ke-ming*[2]

(1. School of Computer Science and Technology，Nanjing University of Aeronautics and Astronautics，Nanjing 210016，China；2. School of Information Engineering，Yancheng Teachers University，Yancheng 224002，China；3. School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

Abstract: To solve the large-scale online learning problem, this paper proposes a bidimensionally partitioned sequential regularized extreme learning machine(BP-SRELM). Based on the online sequential extreme learning machine, combining the divide-and-conquer strategy, the BP-SRELM partitions a high-dimensional hidden layer output matrix into several small matrices from the aspects of instance dimension and feature dimension, so as to reduce the scale and the complexity of the problem, and consequently, the execution efficiency of the algorithm for large-scale learning problem is significantly improved. Meanwhile, the Tikhonov regularization technology is incorporated in the BP-SRELM to further enhance the stability and the generalization capability of the algorithm in real applications. Experimental results show that the proposed BP-SRELM can provide better performances in the sense of stability and prediction accuracy with greatly improved leaning speed than its counterparts, and it can be applied to the online learning and real-time modeling of large-scale data streams.

Keywords: OSELM；Tikhonov regularization；partitioning；online learning；big data stream

0 引 言

在线学习是当前机器学习领域的一个重要研究方向,其基本思想是随着数据样本的不断到达,学习机能够对当前输入样本进行“局部”学习并保持模型的同步更新. 与离线学习方法相比,在线学习方法能以增量的方式实现对数据样本的持续快速学习,非常适合对流式数据环境下的数据流进行实时处理,同时也对计算资源受限条件下离线方法无法有效处理的大数据学习问题提供了替代的解决方案.

在线贯序超限学习机(OSELM)是近年来提出的一种新颖而实用的在线学习算法[1]. OSELM以超限学习机(ELM)[2]的插值理论和逼近定理为基础,将单隐层前馈神经网络(SLFNs)的训练问题转化为线性方程组的求解问题,并采用递归最小二乘(RLS)方法递推计算输出权值以实现模型的在线更新. 与其他流行在线学习算法相比,OSELM具有学习速度快、泛化能力强、模型结构简单等诸多优点,并在时间序列预测等问题中获得了成功应用[3-5]. 然而,对于大规

收稿日期: 2016-06-26；**修回日期:** 2016-12-15.

基金项目: 国家自然科学基金项目(61603326, 61379064, 61273106)；国家科技支撑计划课题(2014BAJ04B02).

作者简介: 郭威(1983−), 男, 讲师, 博士生, 从事数据挖掘、机器学习的研究；徐涛(1962−), 男, 教授, 博士生导师, 从事数据挖掘、智能信息处理等研究.

†通讯作者. E-mail: xutao@nuaa.edu.cn

模复杂学习问题,OSELM仍存在模型更新复杂、实时建模效率较低的不足. 当待处理的数据集规模较大或维度较高时,OSELM通常需要较多的隐层节点对数据样本进行特征映射以更好地逼近目标函数[6],这同时也导致算法迭代计算的复杂性显著增加,学习效率快速下降. 针对该问题,文献[7]提出了一种分割OSELM(P-OSELM)算法,该算法将一个大的数据矩阵分割成若干个小的子矩阵,然后对每个子矩阵分别应用RLS方法进行求解,最后将求得的各子向量解合并起来构成最终的完整解. 文献[7]从计算复杂度分析的角度给出了最优分割参数的计算公式,并通过高阶系统建模实验表明采用分割策略的P-OSELM算法比原始OSELM具有更高的建模效率. 但由于P-OSELM在迭代学习过程中涉及较多的子矩阵求逆操作,使得该算法在实际应用中面临潜在的由于病态矩阵求逆而可能导致的不稳定性问题,且算法所采用的粗放近似计算方法进一步加剧了这种不稳定性. 此外,P-OSELM仅实现了对单个数据的逐一学习,而对块数据的在线学习并不支持. 文献[8]在分析OSELM中矩阵计算相互依赖关系的基础上,提出一种基于MapReduce的并行OSELM算法,提高了对大规模学习问题的执行效率和可扩展能力,但该算法要求所有训练样本均可一次性获得,对于常见的数据流学习问题并不适用. 深度学习[9]是当前大数据分析的热点技术,借鉴深度神经网络的思想,人们将ELM从单隐层拓展到多隐层,提出了多层ELM算法以满足大数据处理的需求[10-11],但这些算法均基于离线批处理学习方式. 最近,文献[12]以多层ELM为基础提出一种可贯序学习训练样本的多层OSELM算法,但该文主要关注算法学习精度的提高,而对算法的学习效率并未作具体的分析与验证.

针对大规模在线学习问题,本文提出一种新的二维分割贯序正则化超限学习机(BP-SRELM). BP-SRELM以OSELM学习模型为基础,从实例和特征两个维度对高维隐层输出矩阵进行分割,将一个高阶模型分解为若干个低阶模型并分别求解,大大降低了模型求解的计算复杂性,从而显著提高对大规模学习问题的执行效率;同时,BP-SRELM通过引入Tikhonov正则化技术使得算法具有良好的稳定性和可靠性,进一步提高算法的泛化能力. 通过5个典型时间序列预测实例验证了BP-SRELM的有效性和高效性.

1　OSELM算法

OSELM是ELM的在线学习版本. 对于N个任意的相异样本$(x_i, y_i) \in R^d \times R^m$,具有$n$个隐层节点的SLFNs的数学模型为

$$\sum_{j=1}^{n}\beta_j g_j(x_i)=\sum_{j=1}^{n}\beta_j G(a_j,b_j,x_i),\quad i=1,2,\cdots,N. \tag{1}$$

其中: a_j和b_j是第j个隐层节点的学习参数,β_j是连接第j个隐层节点与输出层的输出权值,$g_j(x_i)=G(a_j,b_j,x_i)$表示第j个隐层节点关于输入x_i的输出.

该SLFNs能以零误差逼近这N个样本意味着存在(a_j,b_j)及β_j使得

$$\sum_{j=1}^{n}\beta_j G(a_j,b_j,x_i)=y_i,\ i=1,2,\cdots,N. \tag{2}$$

上面N个等式可写成如下矩阵形式:

$$H\beta=Y. \tag{3}$$

其中

$$H=\begin{bmatrix}h_1\\ \vdots\\ h_N\end{bmatrix}=\begin{bmatrix}G(a_1,b_1,x_1) & \cdots & G(a_n,b_n,x_1)\\ \vdots & \ddots & \vdots\\ G(a_1,b_1,x_N) & \cdots & G(a_n,b_n,x_N)\end{bmatrix}_{N\times n}, \tag{4}$$

$$\beta=\begin{bmatrix}\beta_1^{\mathrm{T}}\\ \vdots\\ \beta_n^{\mathrm{T}}\end{bmatrix}_{n\times m},\quad Y=\begin{bmatrix}y_1^{\mathrm{T}}\\ \vdots\\ y_N^{\mathrm{T}}\end{bmatrix}_{N\times m}, \tag{5}$$

H称为网络的隐层输出矩阵. H的第i行h_i为所有隐层节点关于输入x_i的特征映射,H的第j列为第j个隐层节点关于输入$x_1,x_2,\cdots,x_N$的输出向量.

根据文献[2]中的插值理论,给定一个SLFN和N个任意不同样本,只要隐层激活函数无限可微,隐层节点参数(a_j,b_j)可随机赋值并保持不变,此时隐层输出矩阵H为一常数矩阵,则SLFNs的训练问题就能转化为求解线性系统$H\beta=Y$的最小二乘解$\hat{\beta}$,有

$$\|H\hat{\beta}-Y\|=\min_{\beta}\|H\beta-Y\|. \tag{6}$$

ELM采用最小模最小二乘解作为输出权值,即

$$\hat{\beta}=H^{\dagger}Y, \tag{7}$$

其中$H^{\dagger}$为H的Moore-Penrose广义逆. 如果$H^{\mathrm{T}}H$非奇异,则式(7)可进一步写为

$$\hat{\beta}=H^{\dagger}Y=(H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}Y. \tag{8}$$

为了适应实时在线学习的需要,Liang等[1]将贯序学习的思想用于ELM并提出一种可增量学习训练样本的在线学习算法OSELM,其学习过程描述如下.

在初始化阶段,给定初始训练子集$\Omega_0=\{(x_i,$

$y_i)|i=1,2,\cdots,N_0\}$,根据式(8),初始输出权值为

$$\beta(0)=P(0)H^{\mathrm{T}}(0)Y(0). \tag{9}$$

其中: $P(0)=(H^{\mathrm{T}}(0)H(0))^{-1}$, $H(0)=[h_1^{\mathrm{T}},h_2^{\mathrm{T}},\cdots,h_{N_0}^{\mathrm{T}}]^{\mathrm{T}}$, $Y(0)=[y_1,y_2,\cdots,y_{N_0}]^{\mathrm{T}}$.

在贯序学习阶段,每当获取到新的训练数据块

$$\Omega_k=\Big\{(x_i,y_i)|i=\Big(\sum_{j=0}^{k-1}N_j\Big)+1,\cdots,\sum_{j=0}^{k}N_j\Big\},$$

$k=1,2,\cdots$时,按照下式递推计算输出权值:

$$\begin{aligned}P(k)&=P(k-1)-P(k-1)H^{\mathrm{T}}(k)(I+\\&\quad H(k)P(k-1)H^{\mathrm{T}}(k))^{-1}H(k)P(k-1),\\\beta(k)&=\beta(k-1)+P(k)H^{\mathrm{T}}(k)(Y(k)-\\&\quad H(k)\beta(k-1)).\end{aligned} \tag{10}$$

其中

$$H(k)=\Big[h^{\mathrm{T}}_{\left(\sum\limits_{j=0}^{k-1}N_j\right)+1}\ h^{\mathrm{T}}_{\left(\sum\limits_{j=0}^{k-1}N_j\right)+2}\cdots h^{\mathrm{T}}_{\sum\limits_{j=0}^{k}N_j}\Big]^{\mathrm{T}},$$

$$Y(k)=\Big[y_{\left(\sum\limits_{j=0}^{k-1}N_j\right)+1}\ y_{\left(\sum\limits_{j=0}^{k-1}N_j\right)+2}\cdots y_{\sum\limits_{j=0}^{k}N_j}\Big]^{\mathrm{T}}.$$

2 二维分割贯序正则化超限学习机

2.1 BP-SRELM的数学描述

由OSELM算法的推导过程可知,OSELM迭代更新的关键在于重复使用Woodbury公式对维度大小为$n\times n$的自相关矩阵$H^{\mathrm{T}}H$进行求逆计算,而一旦$H^{\mathrm{T}}H$为奇异或病态时,OSELM的泛化性能将严重下降甚至算法完全失效. 此外,在大规模复杂学习问题中,OSELM一般选择较大的隐层节点个数n以保证模型的逼近精度,这也将导致模型更新的计算开销显著增加,建模效率快速下降.

为了提高OSELM在大规模学习问题中的执行效率和泛化能力,本文引入分治策略和Tikhonov正则化技术,提出一种新的二维分割贯序正则化超限学习机算法BP-SRELM. BP-SRELM以OSELM为基础,结合分治策略的思想,从实例和特征两个维度对大小为$N\times n$的隐层输出矩阵H(式(4))进行分割,将高维矩阵分解为多个低维子矩阵并分别进行求解,以降低问题求解的规模和计算复杂性;同时,BP-SRELM通过融合使用正则化方法,不仅有效避免了算法潜在的由于病态矩阵求逆而导致的不稳定性问题,而且有助于算法泛化性能的提高.

首先,在水平方向假定所有N个实例样本按行分割成k个数据块并以贯序的方式输入学习器,每个实例数据块的大小分别为$N_t(t=1,2,\cdots,k)$,且$N=\sum\limits_{t=1}^{k}N_t$,则隐层输出矩阵$H$可表示为行分块矩阵的形式,即

$$H=[H^{\mathrm{T}}(1)\ \ H^{\mathrm{T}}(2)\ \ \cdots\ \ H^{\mathrm{T}}(k)]^{\mathrm{T}},$$

其中$H(t)(t=1,2,\cdots,k)$为ELM关于第t个数据块的特征空间. 类似地,将目标输出矩阵Y按行分割表示为$Y=[Y^{\mathrm{T}}(1)\ \ Y^{\mathrm{T}}(2)\ \ \cdots\ \ Y^{\mathrm{T}}(k)]^{\mathrm{T}}$,则BP-SRELM算法等价于最小化求解如下带正则化项的最小二乘代价函数:

$$J(\beta(k))=\sum_{t=1}^{k}|Y(t)-H(t)\beta(k)|^2+\delta\|\beta(k)\|^2, \tag{11}$$

其中δ为正则化参数. 将代价函数$J(\beta(k))$对$\beta(k)$求微分并令$\dfrac{\partial J(\beta(k))}{\partial\beta(k)}=0$,可得法方程

$$R(k)\beta(k)=\Phi(k). \tag{12}$$

其中

$$R(k)=\sum_{t=1}^{k}H^{\mathrm{T}}(t)H(t)+\delta I, \tag{13}$$

$$\Phi(k)=\sum_{t=1}^{k}H^{\mathrm{T}}(t)Y(t). \tag{14}$$

然后,在垂直方向将每个大小为$N_t\times n$的特征空间$H(t)$按列分割为m个特征子空间,即

$$H(t)=[H_1(t)|H_2(t)|\cdots|H_m(t)],$$

其中子空间$H_i(t)(i=1,2,\cdots,m)$的大小为$N_t\times n_i$,且$n=\sum\limits_{i=1}^{m}n_i$,则式(13)中自相关矩阵$R(k)$可重写为

$$\begin{aligned}R(k)=&\\&\sum_{t=1}^{k}\left\{\begin{bmatrix}H_1^{\mathrm{T}}(t)\\H_2^{\mathrm{T}}(t)\\\vdots\\H_m^{\mathrm{T}}(t)\end{bmatrix}[H_1(t)\ \ H_2(t)\ \ \cdots\ \ H_m(t)]\right\}+\delta I=\\&\begin{bmatrix}R_{11}(k)&R_{12}(k)&\cdots&R_{1m}(k)\\R_{21}(k)&R_{22}(k)&\cdots&R_{2m}(k)\\\vdots&\vdots&\ddots&\vdots\\R_{m1}(k)&R_{m2}(k)&\cdots&R_{mm}(k)\end{bmatrix},\end{aligned} \tag{15}$$

其中$i,j=1,2,\cdots,m$. 当$i=j$时,有

$$\begin{aligned}R_{ii}(k)&=\sum_{t=1}^{k}H_i^{\mathrm{T}}(t)H_i(t)+\delta I^*=\\&\sum_{t=1}^{k-1}H_i^{\mathrm{T}}(t)H_i(t)+\delta I^*+H_i^{\mathrm{T}}(k)H_i(k)=\\&R_{ii}(k-1)+H_i^{\mathrm{T}}(k)H_i(k),\end{aligned} \tag{16}$$

其中I^*是大小为$n_i\times n_i$的单位矩阵,且对角矩阵δI^*

的引入保证了子自相关矩阵$R_{ii}(k)$的良态性以及后续对其进行求逆计算的可靠性. 当$i \neq j$时,有

$$R_{ij}(k)=\sum_{t=1}^{k}H_i^{\mathrm{T}}(t)H_j(t)=\sum_{t=1}^{k-1}H_i^{\mathrm{T}}(t)H_j(t)+H_i^{\mathrm{T}}(k)H_j(k)=R_{ij}(k-1)+H_i^{\mathrm{T}}(k)H_j(k). \tag{17}$$

综合式(16)和(17),对于所有$i,j=1,2,\ldots,m$,有

$$R_{ij}(k)=R_{ij}(k-1)+H_i^{\mathrm{T}}(k)H_j(k). \tag{18}$$

类似地,式(14)中互相关矩阵$\Phi(k)$可分割表示为

$$\Phi(k)=\sum_{t=1}^{k}\left\{\begin{bmatrix}H_1^{\mathrm{T}}(t)\\H_2^{\mathrm{T}}(t)\\\vdots\\H_m^{\mathrm{T}}(t)\end{bmatrix}Y(t)\right\}=\begin{bmatrix}\Phi_1(k)\\\Phi_2(k)\\\vdots\\\Phi_m(k)\end{bmatrix}. \tag{19}$$

其中

$$\Phi_i(k)=\sum_{t=1}^{k}H_i^{\mathrm{T}}(t)Y(t)=\Phi_i(k-1)+H_i^{\mathrm{T}}(k)Y(k),\quad i=1,2,\cdots,m. \tag{20}$$

将式(15)和(19)代入(12),输出权值$\beta(k)$可表示为各输出权值子向量$\beta_i(k)$的串联形式,即

$$\beta(k)=\begin{bmatrix}\beta_1(k)\\\beta_2(k)\\\vdots\\\beta_m(k)\end{bmatrix}=\begin{bmatrix}R_{11}(k)&R_{12}(k)&\cdots&R_{1m}(k)\\R_{21}(k)&R_{22}(k)&\cdots&R_{2m}(k)\\\vdots&\vdots&\ddots&\vdots\\R_{m1}(k)&R_{m2}(k)&\cdots&R_{mm}(k)\end{bmatrix}^{-1}\begin{bmatrix}\Phi_1(k)\\\Phi_2(k)\\\vdots\\\Phi_m(k)\end{bmatrix}. \tag{21}$$

其中

$$\beta_i(k)=R_{ii}^{-1}(k)\Big(\Phi_i(k)-\sum_{j=1,j\neq i}^{m}R_{ij}(k)\beta_j(k)\Big),\quad i=1,2,\cdots,m. \tag{22}$$

2.2 BP-SRELM的递归求解与实现

下面将推导出各输出权值子向量$\beta_i(k)(i=1,2,\cdots,m)$的递推计算表达式以满足在线学习的要求. 将Sherman-Morrison-Woodbury公式[13]应用于式(16),并令$P_{ii}(k)=R_{ii}^{-1}(k)$,可得

$$P_{ii}(k)=P_{ii}(k-1)-G_{ii}(k)H_i(k)P_{ii}(k-1), \tag{23}$$

其中

$$G_{ii}(k)=P_{ii}(k-1)H_i^{\mathrm{T}}(k)(I+H_i(k)P_{ii}(k-1)H_i^{\mathrm{T}}(k))^{-1}. \tag{24}$$

将式(24)变形为

$$G_{ii}(k)+G_{ii}(k)H_i(k)P_{ii}(k-1)H_i^{\mathrm{T}}(k)=P_{ii}(k-1)H_i^{\mathrm{T}}(k), \tag{25}$$

并在式(25)左右两端同时减去左端第2项,可得

$$G_{ii}(k)=(P_{ii}(k-1)-G_{ii}(k)H_i(k)P_{ii}(k-1))H_i^{\mathrm{T}}(k)=P_{ii}(k)H_i^{\mathrm{T}}(k). \tag{26}$$

综合式(18)、(20)、(23)及(26),则式(22)中右端第1项可表示为

$$\begin{aligned}&R_{ii}^{-1}(k)\Phi_i(k)=\\&P_{ii}(k)\Phi_i(k-1)+P_{ii}(k)H_i^{\mathrm{T}}(k)Y(k)=\\&P_{ii}(k-1)\Phi_i(k-1)-G_{ii}(k)H_i(k)\times\\&P_{ii}(k-1)\Phi_i(k-1)+G_{ii}(k)Y(k).\end{aligned} \tag{27}$$

式(22)中右端第2项可表示为

$$\begin{aligned}&R_{ii}^{-1}(k)\sum_{j=1,j\neq i}^{m}R_{ij}(k)\beta_j(k)=\\&\sum_{j=1,j\neq i}^{m}P_{ii}(k)R_{ij}(k)\beta_j(k)=\\&\sum_{j=1,j\neq i}^{m}P_{ii}(k)(R_{ij}(k-1)+H_i^{\mathrm{T}}(k)H_j(k))\beta_j(k)=\\&\sum_{j=1,j\neq i}^{m}P_{ii}(k)R_{ij}(k-1)\beta_j(k)+\\&\sum_{j=1,j\neq i}^{m}G_{ii}(k)H_j(k)\beta_j(k)=\\&\sum_{j=1,j\neq i}^{m}P_{ii}(k-1)R_{ij}(k-1)\beta_j(k)-\\&\sum_{j=1,j\neq i}^{m}G_{ii}(k)H_i(k)\times\\&P_{ii}(k-1)R_{ij}(k-1)\beta_j(k)+\\&\sum_{j=1,j\neq i}^{m}G_{ii}(k)H_j(k)\beta_j(k).\end{aligned} \tag{28}$$

将式(27)和(28)代入(22),合并同类项,可得

$$\beta_i(k)=P_{ii}(k)\Big[\Phi_i(k)-\sum_{j=1,j\neq i}^{m}R_{ij}(k)\beta_j(k)\Big]=$$

$$P_{ii}(k-1)\Big[\Phi_i(k-1)-\sum_{j=1,j\neq i}^{m}R_{ij}(k-1)\beta_j(k)\Big]-$$
$$G_{ii}(k)H_i(k)P_{ii}(k-1)\Big[\Phi_i(k-1)-$$
$$\sum_{j=1,j\neq i}^{m}R_{ij}(k-1)\beta_j(k)\Big]+$$
$$G_{ii}(k)\Big[Y(k)-\sum_{j=1,j\neq i}^{m}H_j(k)\beta_j(k)\Big]. \tag{29}$$

假定在较短的时间间隔内$\Delta\beta_j=\beta_j(k)-\beta_j(k-1)$很小,则式(29)可进一步近似表示为

$$\beta_i(k)=$$
$$\beta_i(k-1)-G_{ii}(k)H_i(k)\beta_i(k-1)+$$
$$G_{ii}(k)\Big[Y(k)-\sum_{j=1,j\neq i}^{m}H_j(k)\beta_j^*(k)\Big]=$$
$$\beta_i(k-1)+G_{ii}(k)\times$$
$$\Big[Y(k)-\sum_{j=1,j\neq i}^{m}H_j(k)\beta_j^*(k)-H_i(k)\beta_i(k-1)\Big]. \tag{30}$$

其中

$$\beta_j^*(k)=\begin{cases}\beta_j(k-1),\ j<i;\\ \beta_j(k-1),\ j>i;\end{cases}$$
$$i,j=1,2,\cdots,m,\ j\neq i. \tag{31}$$

式(23)、(24)及(30)构成了BP-SRELM算法中输出权值子向量$\beta_i(k)$的递推计算公式.

至此,可以给出BP-SRELM算法的完整描述.

BP-SRELM算法 假定数据样本(x_i,y_i)以数据流的形式逐一或逐块到达,激活函数为$G(a,b,x)$,隐层节点个数为n,实例数据块大小为N_k,特征分割数为m,正则化参数为δ.

Step 1 初始化阶段. 对于给定的初始训练样本子集$\Omega_0=\{(x_i,y_i)|i=1,2,\cdots,N_0\},N_0>n$:

1) 随机生成隐层节点参数$(a_j,b_j),\ j=1,2,\cdots,n$.

2) 按照式(4)计算初始的隐层输出矩阵$H(0)$.

3) 计算初始的输出权值$\beta(0)=P(0)H^{\mathrm{T}}(0)Y(0)$. 其中:$P(0)=(H^{\mathrm{T}}(0)H(0)+\delta I)^{-1},Y(0)=[y_1,y_2,\cdots,y_{N_0}]^{\mathrm{T}}$.

4) 设定初始的$\beta_i(0)$及$P_{ii}(0),i=1,2,\cdots,m$,$\beta_i(0)$初始化为$\beta(0)$中$(i-1)\times c+1$到$i\times c$之间的矢量,$P_{ii}(0)$初始化为$(H^{\mathrm{T}}(0)H(0)+\delta I)$中$(i-1)\times c+1$到$i\times c$之间对角分块方阵的逆矩阵,这里$c=n/m$为每个特征子空间的大小.

5) 置$k=1$.

Step 2 贯序学习阶段. 对于每个大小为N_k的新数据块$\Omega_k=\Big\{(x_i,y_i)|i=\Big(\sum_{j=0}^{k-1}N_j\Big)+1,\cdots,\sum_{j=0}^{k}N_j\Big\}$,有:

1) 按照式(4)和(5)分别计算Ω_k对应的局部隐层输出矩阵$H(k)$及目标输出$Y(k)$.

2) 将$H(k)$按列分割为

$$H(k)=[H_1(k)|H_2(k)|\cdots|H_m(k)].$$

3) 按照式(23)、(24)及(30)递归计算输出权值子向量$\beta_i(k),\ i=1,2,\cdots,m$.

4) 按照式(21)将各个$\beta_i(k)$串联起来得到完整的$\beta(k)$.

5) 置$k=k+1$,返回Step 2.

3 仿真实验

为了验证BP-SRELM算法的有效性和高效性,本文将其应用于5个规模较大的时间序列数据集的在线建模与预测,并将实验结果与P-OSELM、OSELM、ELM进行比较和分析.

3.1 数据集与实验设计

本文选择Mackey-Glass、Logistic、Henon、Santa Fe、ESTSP'08五个时间序列作为实验数据集. 其中Mackey-Glass、Logistic和Henon是3个标准混沌时间序列,分别描述如下:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=\frac{0.2x(t-17)}{1+x^{10}(t-17)}-0.1x(t); \tag{32}$$
$$x(t+1)=4x(t)(1-x(t)); \tag{33}$$
$$x(t+1)=1-1.4x^2(t)+y(t),$$
$$y(t+1)=0.3x(t). \tag{34}$$

根据式(32)~(34)分别生成22 017, 50 004, 220 004个时间序列数据,为使预测问题更加真实,在原始数据上添加了水平为0.01的噪声数据. Santa Fe和ESTSP'08也是时间序列预测研究中的两个常用标杆数据集(http://research.ics.aalto.fi/eiml/datasets.shtml). 5个时间序列数据集的基本信息及构建参数(嵌入维度d,时间延迟τ)如表1所示.

对于所有数据集,使用BP-SRELM、P-OSELM、OSELM和ELM四种学习算法对训练样本进行学习建模,并将得到的学习模型在测试集上进行预测. 由于OSELM可看作P-OSELM在特征分割数$m=1$情形下的特例,本文中OSELM的实验过程和实验结果均通过P-OSELM($m=1$)统一实现和给出.

表 1　数据集描述

数据集	样本总数	(d,τ)	训练样本	测试样本
Mackey-Glass	22 017	(17, 1)	20 000	2 000
Logistic	50 004	(4, 1)	45 000	5 000
Henon	220 004	(4, 1)	200 000	20 000
Santa Fe	10 093	(9, 1)	8 084	2 000
ESTSP'08	31 614	(1, 1)	28 613	3 000

在上述算法中, BP-SRELM 和 P-OSELM(包括 OSELM) 采用在线的方式贯序学习训练样本, 而 ELM 采用批处理方式一次性学习所有训练样本. 所有算法均使用相同的 Sigmoid 激活函数, 其中输入权值向量和偏移为 $[-1,1]$ 之间的随机数. 如文献 [14-15] 所述, 只要隐层节点个数 n 足够大, 则基于 ELM 的学习算法一般都能取得较为理想的泛化性能. 参照文献 [14-15], 将所有学习算法的 n 统一设置为 1 000.

对于 P-OSELM 和 BP-SRELM, 设定初始训练样本数 $N_0 = 2\,000$, 设定特征分割数 m 的典型取值范围为 $\{1,2,5,10,20,50\}$; 因 P-OSELM 仅支持单个数据样本的逐一学习, 故每个实例数据块大小 $N_k = 1$, 而 BP-SRELM 可支持不同大小数据块的在线学习, 这里设定 N_k 的典型取值范围为 $\{1,10,20,50,100,200\}$. 此外, 对于 BP-SRELM, 本文验证了其在不同正则化参数条件下的泛化性能 ($\delta = \{10^{-10}, 10^{-9}, \cdots, 10^4, 10^5\}$), 并给出最优条件下的实验结果. 对于每个测试实例, 给出的预测精度和学习时间均为 30 次独立实验的平均值, 算法性能评价标准采用均方根误差 (RMSE), 即

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{35}$$

其中: y_i、$\hat{y}_i$ 分别为真实值和预测值, N 为测试样本数目. 实验执行的硬件平台为 3.4 GHz CPU, 4 GB RAM 的个人 PC 机, 软件环境为 Matlab R2010b.

3.2　实验结果比较与分析

表 2 给出了 P-OSELM(含 OSELM)、BP-SRELM 以及 ELM 在各数据集上的预测 RMSE. 由表 2 容易看

表 2　各种算法的预测 RMSE 比较

数据集	特征分割数 m	P-OSELM	BP-SRELM						ELM
		$N_k = 1$	$N_k = 1$	$N_k = 10$	$N_k = 20$	$N_k = 50$	$N_k = 100$	$N_k = 200$	
Mackey-Glass	1	2.72e-03	2.70e-03	2.70e-03	2.70e-03	2.70e-03	2.70e-03	2.70e-03	2.69e-03
	2	2.71e-03	2.70e-03	2.70e-03	2.70e-03	2.71e-03	2.70e-03	2.70e-03	
	5	6.94e-03	2.73e-03	2.73e-03	2.73e-03	2.73e-03	2.73e-03	2.73e-03	
	10	2.80e-03	2.76e-03	2.76e-03	2.76e-03	2.76e-03	2.76e-03	2.76e-03	
	20	2.85e-03	2.80e-03	2.79e-03	2.79e-03	2.79e-03	2.79e-03	2.79e-03	
	50	2.87e-03	2.82e-03	2.82e-03	2.82e-03	2.82e-03	2.82e-03	2.82e-03	
Logistic	1	9.24e-03	9.12e-03	9.12e-03	9.12e-03	9.12e-03	9.12e-03	9.12e-03	8.46e-03
	2	9.15e-03	9.11e-03	9.11e-03	9.11e-03	9.11e-03	9.11e-03	9.11e-03	
	5	1.15e-02	9.12e-03	9.12e-03	9.12e-03	9.12e-03	9.12e-03	9.12e-03	
	10	5.52e+09	9.16e-03	9.16e-03	9.16e-03	9.17e-03	9.16e-03	9.16e-03	
	20	2.01e+04	9.35e-03	9.35e-03	9.36e-03	9.35e-03	9.36e-03	9.35e-03	
	50	9.32e-03	9.67e-03	9.67e-03	9.67e-03	9.66e-03	9.67e-03	9.66e-03	
Henon	1	1.33e-02	1.31e-02	1.31e-02	1.31e-02	1.31e-02	1.31e-02	1.31e-02	1.27e-02
	2	1.32e-02	1.31e-02	1.31e-02	1.31e-02	1.31e-02	1.31e-02	1.31e-02	
	5	1.95e+01	1.32e-02	1.32e-02	1.32e-02	1.32e-02	1.32e-02	1.32e-02	
	10	2.03e+06	1.32e-02	1.32e-02	1.32e-02	1.32e-02	1.32e-02	1.32e-02	
	20	1.34e-02	1.34e-02	1.34e-02	1.34e-02	1.34e-02	1.34e-02	1.34e-02	
	50	1.35e-02	1.35e-02	1.35e-02	1.35e-02	1.35e-02	1.35e-02	1.35e-02	
Santa Fe	1	7.10	6.56	6.87	6.78	6.71	6.89	6.83	35.32
	2	131.06	6.85	6.95	6.92	6.94	6.89	6.89	
	5	1.70e+03	7.00	7.09	6.92	6.88	7.00	7.04	
	10	1.90e+04	7.17	7.10	7.13	6.99	7.14	6.92	
	20	1.12e+09	7.44	7.31	7.32	7.44	7.31	7.25	
	50	2.57e+04	7.45	7.49	7.55	7.56	7.67	7.58	
ESTSP'08	1	6.29	6.24	6.24	6.25	6.24	6.24	6.25	6.29
	2	6.48	6.26	6.26	6.26	6.27	6.26	6.26	
	5	6.98	6.29	6.29	6.29	6.29	6.28	6.28	
	10	11.05	6.31	6.30	6.30	6.29	6.30	6.30	
	20	18.69	6.29	6.28	6.30	6.28	6.28	6.29	
	50	2.03e+03	6.44	6.44	6.45	6.44	6.43	6.43	

出,P-OSELM算法极不稳定,其得到的预测RMSE波动很大,例如在Mackey-Glass数据集上,当特征分割数$m = 5$时,其对应的预测RMSE明显偏大. 类似的实验结果在其他4个数据集上表现得更为明显. 原因在于P-OSELM在迭代学习过程中涉及比OSELM更多的矩阵求逆操作,一旦被求逆矩阵为奇异或病态时,算法的泛化性能将严重下降,并可能得到一个极大的毫无意义的预测结果. 相对而言,BP-SRELM通过融合使用正则化技术,有效避免了潜在的病态矩阵求逆问题,从而保证了算法的持续稳定性,而且其预测精度比P-OSELM也有了一定的提高.

综合比较BP-SRELM在不同分割条件下的实验结果可知: BP-SRELM在不同N_k下的预测RMSE基本相同,表明实例分割数据块的大小对BP-SRELM的泛化性能几乎没有影响;而对于不同的特征分割数m,尽管BP-SRELM的预测误差随m的增加而有所增加,但当m较小时,例如$m = 1, 2, 5, 10$时,误差的增幅非常小,这表明适度增加m并不会对BP-SRELM的预测性能造成明显影响. 另外,当$m = 1$时,P-OSELM和BP-SRELM分别退化为一般的OSELM和带正则化的OSELM,其对应的预测误差与批处理学习算法ELM得到的预测误差基本相当,这与文献[1]中的实验结论是一致的. 但值得注意的是,ELM在Santa Fe数据集上的预测结果明显不可靠,原因同样在于矩阵求逆计算的不稳定性,尽管ELM算法仅包含一次矩阵求逆操作.

表3给出了各种算法的学习时间(单位s). 对于P-OSELM,当$m \geqslant 2$时,其对应的学习时间较$m = 1$时明显减少,这表明特征分割策略对于提高OSELM在大规模学习问题中的学习效率是显著有效的. 对于BP-SRELM,其在$N_k = 1$时的学习时间与相同条件下P-OSELM的学习时间基本相同,且随着N_k的增加,其学习时间又进一步明显缩短,这表明适当增加N_k有助于进一步提升BP-SRELM的执行效率. 在

表3 各种算法的学习时间比较

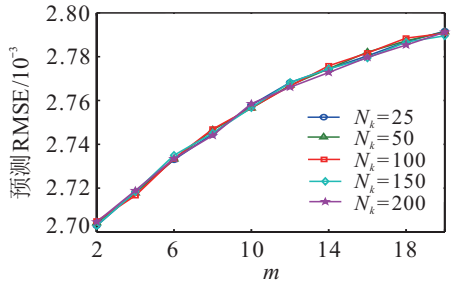
数据集	特征分割数m	P-OSELM	BP-SRELM						ELM
		$N_k = 1$	$N_k = 1$	$N_k = 10$	$N_k = 20$	$N_k = 50$	$N_k = 100$	$N_k = 200$	
Mackey-Glass	1	4 767.64	5 298.08	551.51	257.73	116.52	77.04	47.75	29.77
	2	1 457.69	1 581.52	175.55	88.19	44.29	32.34	24.99	
	5	281.66	282.74	38.32	24.09	15.69	14.88	16.25	
	10	137.81	138.92	24.05	16.41	12.19	13.50	17.95	
	20	140.56	141.41	26.43	18.86	13.58	16.42	25.05	
	50	77.35	77.92	54.24	36.60	24.54	29.85	48.98	
Logistic	1	12 342.30	12 359.75	1 306.80	659.01	289.86	169.66	113.45	53.33
	2	3 835.95	3 849.84	410.81	220.08	105.08	68.18	55.47	
	5	630.74	657.24	84.20	52.02	31.94	28.90	34.17	
	10	309.76	315.21	50.43	33.31	23.79	25.92	39.26	
	20	330.57	330.27	58.05	38.59	27.58	33.08	57.08	
	50	179.43	172.25	27.97	19.96	55.10	65.30	117.74	
Henon	1	54 095.26	57 328.13	5 687.48	2 942.72	1 247.78	743.34	488.07	259.60
	2	16 366.31	16 211.84	1 728.25	947.80	449.15	299.25	234.58	
	5	2 976.35	2 978.79	371.63	227.13	130.72	127.32	141.47	
	10	1 536.68	1 529.31	233.47	148.74	99.21	120.88	163.90	
	20	1 857.23	1 836.36	301.07	190.78	123.29	171.63	249.10	
	50	1 195.31	1 171.44	166.14	100.89	283.12	384.82	541.01	
Santa Fe	1	1 800.84	1 761.16	193.19	102.05	48.15	30.88	22.84	14.17
	2	566.10	552.16	62.92	35.84	19.47	14.32	12.61	
	5	97.45	97.54	16.07	11.42	8.58	8.03	8.87	
	10	49.72	49.66	11.15	8.68	7.35	7.63	9.44	
	20	50.28	50.40	11.93	9.45	7.78	8.54	11.84	
	50	29.84	29.90	21.56	15.49	11.65	13.12	20.27	
ESTSP'08	1	7 285.98	7 742.64	837.41	423.11	187.64	110.25	73.64	33.60
	2	2 350.77	2 392.61	255.43	136.84	66.53	43.54	35.83	
	5	408.20	408.14	53.56	33.58	21.06	19.30	22.28	
	10	199.14	196.75	32.02	21.93	16.16	17.26	25.30	
	20	204.88	203.16	36.48	25.24	18.48	21.75	36.25	
	50	111.05	111.89	19.72	14.02	35.76	41.48	74.09	

本文实验中,对于5个规模大小不同的数据集,BP-SRELM一般在$m = 10, N_k = 50$时学习效率较高,且其对应的学习时间不仅远小于相同m条件下P-OSELM的学习时间,而且也明显低于ELM执行一次批量学习所需要的时间.

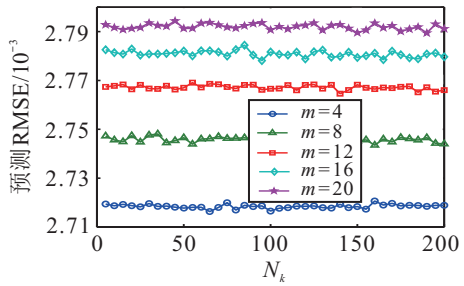
由表2和表3的实验结果可知,合并使用二维分割策略与正则化技术的BP-SRELM算法能在基本不损失预测精度和保持持续稳定性的同时,极大地提高对大规模学习问题的执行效率.

3.3　关键参数对BP-SRELM性能的影响

为了更详尽和直观地说明特征分割数m及实例数据块大小N_k这两个关键参数对BP-SRELM的预测精度和学习效率的影响,本文以Mackey-Glass数据集为例,给出BP-SRELM在典型参数条件下预测误差和学习时间曲线,分别如图1、图2所示. 这里,m的完整取值范围为$\{2,4,\cdots,18,20\}$,N_k的完整取值范围为$\{5,10,\cdots,195,200\}$.

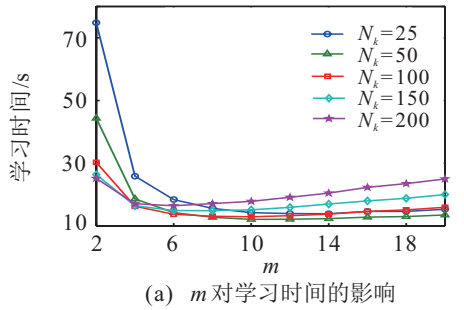


(a)　m对预测精度的影响

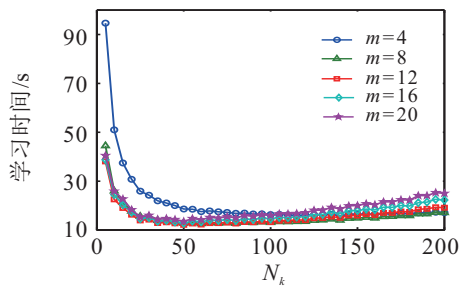


(b)　N_k对预测精度的影响

图1　BP-SRELM在不同(m,N_k)下的预测误差



(a)　m对学习时间的影响



(b)　N_k对学习时间的影响

图2　BP-SRELM在不同(m,N_k)下的学习时间

首先由图1(a)可以看出,随着m的增加,BP-SRELM的预测误差有一定的上升,但综合5个数据集上的实验结果可以发现,当m较小时,如$m \leqslant 12$时,其预测误差的增加幅度很小,基本与无特征分割$(m = 1)$条件下的预测性能相当;由图1(b)可知,在不同的m下,BP-SRELM的预测误差随N_k的变化基本保持不变,这再次证明N_k对BP-SRELM的预测性能几乎没有影响.

图2展示了BP-SRELM的学习效率与m及N_k之间的关系:由图2(a)可以看出,随着m的增加,BP-SRELM的学习时间快速下降,但随后又开始缓慢上升,这是由于当m取值过大时,尽管每个子问题求解的计算复杂性进一步降低,但子问题的数量也同步增加,且数据分割过程本身也会带来额外的计算开销,从而导致总的计算时间反而增加,故m的取值应该有一个适中的范围;类似地,图2(b)也表明,适当增加N_k对于降低BP-SRELM的学习时间是显著有效的,但当N_k取值过大时,BP-SRELM的学习时间又随之缓慢上升,原因是较大的N_k虽然能减少贯序学习的次数,但同时也增加了单次贯序学习过程的计算复杂性,从而导致算法总的计算复杂性增加,因此N_k也需要适中选择.

综上,关于BP-SRELM中m和N_k这两个参数的选择有如下结论:一般来说,在合适的范围内,适当增加m能显著提高BP-SRELM的学习效率,但同时也会损失少量的预测精度,故m应折衷选择以保持预测精度和学习效率之间的平衡;由于N_k对BP-SRELM的预测性能几乎没有影响,其取值主要考虑使算法的学习效率最佳. 事实上,在本文5组实验中,当$m \in [2,12]$时,BP-SRELM的预测精度下降甚微,基本与$m = 1$时的预测精度相当;而当$m \in [6,16]$,$N_k \in [35,100]$时,BP-SRELM的学习时间较短,故实验结果显示当$m \in [6,12], N_k \in [35,100]$时,BP-SRELM整体的预测性能和学习效率均较高.

此外,本文还分析并验证了正则化参数δ对BP-SRELM预测性能的影响. 根据前面的实验结论,由于N_k对BP-SRELM的泛化性能影响不大,这里仅以$N_k = 100$为例给出具有不同特征分割数m的BP-SRELM在典型正则化参数条件下的预测RMSE,如

图3所示. 由图3可以看出,BP-SRELM的预测误差随m的增加而有所上升,但对于不同的m,其预测误差随δ的变化趋势基本相同,且在本文Mackey-Glass实例中,当正则化参数$\delta = 10^{-6}$时算法的整体预测性能较好.

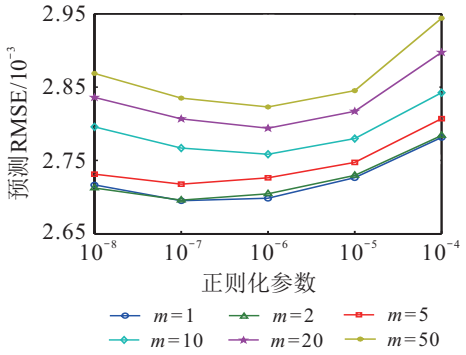


图 3 BP-SRELM在不同正则化参数下的预测误差

综合其他4个数据集上的实验结果可以发现,对于不同的数据集,δ的最优取值不尽相同. 当δ取值过小时,由于其正则化效果不够明显而可能得到不佳甚至无效的预测结果;而当δ取值过大时,又相当于引入了额外的噪声,使得算法的预测误差也随之增加. 因此,选择一个合适的正则化参数是很重要的. 实际应用中,正则化参数的选取通常可采用黄金分割法或L-曲线法[16].

4 结 论

为提高OSELM在大规模复杂学习问题中的学习效率和稳定性,本文提出了一种新的二维分割贯序正则化超限学习机BP-SRELM. BP-SRELM以OSELM学习模型为基础,通过适当的分割求解策略和近似计算方法得到关于输出权值的一个递归逼近解,从而能在基本不损失算法学习精度的同时使得算法的学习效率得到极大提升;此外,BP-SRELM内置的正则化技术在保证算法稳定性的同时还进一步提高了其泛化性能. 5个典型时间序列数据集上的仿真实验表明了BP-SRELM对大规模学习问题的有效性和高效性.

BP-SRELM能支持不同大小数据块的快速在线学习,非常适用于高速数据流环境下的实时建模问题;同时,BP-SRELM也可拓展到离线学习环境,用于解决传统批量学习算法由于计算资源不足而无法有效处理的大数据学习问题.

参考文献(References)

[1] Liang N Y, Huang G B, Saratchandran P, et al. A fast and accurate online sequential learning algorithm for feedforward networks[J]. IEEE Trans on Neural Networks, 2006, 17(6): 1411-1423.

[2] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.

[3] 毛文涛, 田杨阳, 王金婉, 等. 面向贯序不均衡分类的粒度极限学习机[J]. 控制与决策, 2016, 31(12): 2147-2154.
(Mao W T, Tian Y Y, Wang J W, et al. Granular extreme learning machine for sequential imbalanced data[J]. Control and Decision, 2016, 31(12): 2147-2154.)

[4] Wang X, Han M. Online sequential extreme learning machine with kernels for nonstationary time series prediction[J]. Neurocomputing, 2014, 145: 90-97.

[5] Lima A R, Cannon A J, Hsieh W W. Forecasting daily streamflow using online sequential extreme learning machines[J]. J of Hydrology, 2016, 537: 431-443.

[6] Zhou H, Huang G B, Lin Z, et al. Stacked extreme learning machines[J]. IEEE Trans on Cybernetics, 2015, 45(9): 2013-2025.

[7] Lim J S. Partitioned online sequential extreme learning machine for large ordered system modeling[J]. Neurocomputing, 2013, 102(2): 59-64.

[8] Wang B, Huang S, Qiu J, et al. Parallel online sequential extreme learning machine based on MapReduce[J]. Neurocomputing, 2015, 149: 224-232.

[9] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[10] Kasun L L C, Zhou H, Huang G B, et al. Representational learning with extreme learning machine for big data[J]. IEEE Intelligent Systems, 2013, 28(6): 31-34.

[11] Tang J, Deng C, Huang G B. Extreme learning machine for multilayer perceptron[J]. IEEE Trans on Neural Networks and Learning Systems, 2016, 27(4): 809-821.

[12] Mirza B, Kok S, Dong F. Multi-layer online sequential extreme learning machine for image classification[C]. Proc of ELM-2015. Berlin: Springer, 2016: 39-49.

[13] Golub G H, Van Loan C F. Matrix computations[M]. Baltimore: JHU Press, 2013: 65.

[14] Huang G B, Zhou H, Ding X, et al. Extreme learning machine for regression and multiclass classification[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2012, 42(2): 513-529.

[15] Shao Z, Meng J E. An online sequential learning algorithm for regularized extreme learning machine[J]. Neurocomputing, 2016, 173: 778-788.

[16] Huynh H T, Won Y. Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks[J]. Pattern Recognition Letters, 2011, 32(14): 1930-1935.

（责任编辑：李君玲）