

二元裂解算子交替方向乘子法的核极限学习机

苏一丹 续嘉 覃华*

(广西大学计算机与电子信息学院 南宁 530004)

摘要: 凸优化形式的核极限学习机(KELM)具有较高的分类准确率, 但用迭代法训练凸优化核极限学习机要较传统核极限学习机的解线性方程法花费更长时间。针对此问题, 该文提出一种二元裂解算子交替方向乘子法(BSADMM-KELM)来提高凸优化核极限学习机的训练速度。首先引入二元裂解算子, 将求核极限学习机最优解的过程分裂为两个中间算子的优化过程, 再通过中间算子的迭代计算而得到原问题的最优解。在22个UCI数据集上所提算法的训练时间较有效集法平均快29倍, 较内点法平均快4倍, 分类精度亦优于传统的核极限学习机; 在大规模数据集上该文算法的训练时间优于传统核极限学习机。

关键词: 核极限学习机; 2次规划模型; 二元裂解算子; 交替方向乘子法

中图分类号: TP301.6

文献标识码: A

文章编号: 1009-5896(2021)09-2586-08

DOI: 10.11999/JEIT200884

Kernel Extreme Learning Machine Based on Alternating Direction Multiplier Method of Binary Splitting Operator

SU Yidan XU Jia QIN Hua

(College of Computer and Electronic Information, Guangxi University, Nanning 530004, China)

Abstract: The Kernel Extreme Learning Machine (KELM) with convex optimization form has higher classification accuracy, but it takes longer time to train kelm with iterative method than solving linear equation method of traditional kelm. To solve this problem, an Alternating Direction Multiplier Method(ADMM) of Binary Splitting (BSADMM-KELM) is proposed to improve the training speed of convex optimization kernel extreme learning machine. Firstly, the process of finding the optimal solution of the kernel extreme learning machine is split into two intermediate operators by introducing a binary splitting operator, and then the optimal solution of the original problem is obtained through the iterative calculation of the intermediate operators. On 22 UCI datasets, the training time of the proposed algorithm is 29 times faster than that of the effective set method and 4 times faster than that of the interior point method. The classification accuracy of the proposed algorithm is also better than that of the traditional kernel extreme learning machine. On large-scale datasets, the training time of the proposed algorithm is better than that of the traditional kernel extreme learning machine.

Key words: Kernel Extreme Learning Machine(KELM); Quadratic programming; Binary splitting operator; Alternating Direction Multiplier Method(ADMM)

1 引言

极限学习机^[1](Extreme Learning Machine, ELM)是一种单隐层前馈神经网络, 具有实现简单、计算速度快等优点, 目前已广泛应用在图像识别^[2]、电力监测^[3]、医学^[4,5]等领域。ELM在设计时采用解线性方程组方法实现训练, 但线性方程组的系数矩阵存在病态问题(ill-posed problem), 会导致学习机的分类精度下降。近年出现了许多改进的极限学习机, 大体可分为两大类, 一类是在传统ELM/核极

限学习机(Kernel Exterme Learning Machine, KELM)的基础上继续优化, 仍然保持解线性方程组获取分类参数的特性, 如Wu等人^[6]提出将参数优化和结构学习结合到学习过程中, 提高了泛化能力的同时缓解了过拟合问题; Zhang等人^[7]提出将局部搜索策略嵌入全局优化框架中得到最优的网络参数。第2大类改进极限学习机的整体思路是把ELM/KELM的训练转换成一个凸优化问题(线性规划、2次规划等), 通过迭代法求解凸优化问题获取ELM/KELM的最优分类参数, 如Peng等人^[8]提出了正交ELM模型, 采用正交约束的权重矩阵生成正交基, 具有比原始ELM优越的局部保存能力; Zuo等人^[9]提出稀

疏ELM, 将ELM的训练转化成2次规划问题进行求解。

交替方向乘子法(Alternating Direction Multiplier Method, ADMM)^[10]是求解具有可分结构凸优化问题的重要方法, 该算法有良好的收敛性, 已广泛应用于机器学习和神经网络优化等领域, 如Liu等人^[11]提出一种加速方差缩减随机ADMM用于大规模机器学习; Dhar等人^[12]提出一种ADMM算法用于可伸缩机器学习。传统ADMM通常采用1元裂解算子, 文献^[13]提出一种2元裂解算子的ADMM算法。受该文献启发, 为进一步提高KELM的学习速度, 本文提出2元裂解算子交替方向乘子法的核极限学习机(Binary Splitting ADMM KELM, BSADMM-KELM), 基本思想是: 将KELM的训练过程看作一个2次规划问题(Quadratic Programming, QP), 引进2个裂解算子对QP问题进行分解, 目的是提高迭代求解QP的速度, 从而提高KELM的训练速度和分类精度, 最后在测试数据集上验证所提算法的可行性和有效性。

2 核极限学习机的2次规划模型

Huang等人在文献^[14]中讨论KELM与支持向量机的关系时指出从标准优化方法的角度来看, ELM分类和支持向量机(Support Vector Machine, SVM)分类是等价的, 但ELM由于其特殊的可分性特征, 优化约束较少, 最优化形式的极限学习机(Optimization ELM, OELM)目的是最小化输出权值和训练误差。给定 N 个训练样本 $(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N)$, 其中输入样本 $x_i \in R^d$; 输入样本的类别标签 $t_i \in R^1$, $\mathbf{T} = [t_1, t_2, \dots, t_N] \in R^N$, OELM模型为

$$\begin{cases} \min \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N x_i \\ \text{s.t. } t_i \beta \cdot \mathbf{h}(\mathbf{x}_i) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = 1, 2, \dots, N \end{cases} \quad (1)$$

其中, β 为输出权值, ξ_i 为训练误差, C 是惩罚系数, $\mathbf{h}(\mathbf{x}_i)$ 是隐藏层相对于输入 x 的输出。根据拉格朗日理论, 最小化输出权值的范数等价于求解式(1)的对偶优化问题

$$\begin{cases} \min L_D = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N t_i t_j v_i v_j x_i x_j - \sum_{i=1}^N v_i \\ \text{s.t. } 0 \leq v_i \leq C, i = 1, 2, \dots, N \end{cases} \quad (2)$$

其中, v_i 是拉格朗日乘子且非负。式(2)模型与SVM具有一致性, 所以该模型存在支持向量 \mathbf{x}_s , 由库恩-塔克(Karush-Kuhn-Tucker, KKT)条件和拉格朗日定理可知, 当 \mathbf{x}_i 为支持向量, 对应的拉格朗日

乘子 v_i 不为0, 若 \mathbf{x}_i 不是支持向量, 则对应的 v_i 为0, 因此OELM的输出为

$$\begin{aligned} f(x) &= \text{sign}(\beta \cdot \mathbf{h}(\mathbf{x}_i)) \\ &= \text{sign} \left(\sum_{s=1}^{N_s} v_s t_s \mathbf{h}(\mathbf{x}_s) \cdot \mathbf{h}(\mathbf{x}_i) \right) \\ &= \text{sign} \left(\sum_{s=1}^{N_s} v_s t_s \cdot \mathbf{K}_{\text{ELM}}(\mathbf{x}_s, \mathbf{x}_i) \right) \end{aligned} \quad (3)$$

其中, N_s 为支持向量的个数, $\mathbf{K}_{\text{ELM}}(\mathbf{x}_s, \mathbf{x}_i)$ 是满足Mercer's定理的核函数

$$\Omega_N = \mathbf{H}\mathbf{H}^T, \Omega_{N_i, j} = \mathbf{h}(\mathbf{x}_i) \mathbf{h}(\mathbf{x}_j) = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \quad (4)$$

对式(2)模型进行简化, 将其变成标准的2次规划问题

$$\begin{cases} \min f(v) = \frac{1}{2} \mathbf{v}^T \cdot \mathbf{P} \cdot \mathbf{v} + \mathbf{q}^T \cdot \mathbf{v} \\ \text{s.t. } \mathbf{A}\mathbf{v} = \mathbf{z}, \mathbf{z} \in \eta \end{cases} \quad (5)$$

其中, $\mathbf{A} \in R^{M \times N}$, $\mathbf{v} = [v_1, \dots, v_N]^T \in R^N$ 是拉格朗日乘子, $\eta = [0, C] = \{z \in R^M | 0 \leq z_i \leq C\}$, $\mathbf{P} = \mathbf{T}^T \cdot \mathbf{T} \cdot \mathbf{K}_{\text{ELM}} \in R^{N \times N}$ 是一个半正定矩阵, $\mathbf{q} = [-1, -1, \dots, -1]^T \in R^N$ 。

3 2元裂解算子的核极限学习机训练算法

3.1 2元裂解算子的交替乘子法

为快速求解式(5), 引入了两个分裂算子 $\tilde{\mathbf{v}} \in R^N$ 和 $\tilde{\mathbf{z}} \in R^M$, 将问题式(5)改写为

$$\begin{cases} \min \frac{1}{2} \tilde{\mathbf{v}}^T \mathbf{P} \tilde{\mathbf{v}} + \mathbf{q}^T \tilde{\mathbf{v}} + L_{\mathbf{A}\mathbf{v}=\mathbf{z}}(\tilde{\mathbf{v}}, \tilde{\mathbf{z}}) + L_C(\mathbf{z}) \\ \text{s.t. } (\tilde{\mathbf{v}}, \tilde{\mathbf{z}}) = (\mathbf{v}, \mathbf{z}) \end{cases} \quad (6)$$

可将 $(\tilde{\mathbf{v}}, \tilde{\mathbf{z}}), (\mathbf{v}, \mathbf{z})$ 分别看做传统ADMM算法中的两个变量。 $L_{\mathbf{A}\mathbf{v}=\mathbf{z}}(\cdot)$ 和 $L_C(\cdot)$ 为指示函数, 定义为

$$\begin{cases} L_{\mathbf{A}\mathbf{v}=\mathbf{z}}(\mathbf{v}, \mathbf{z}) = \begin{cases} 0, & \mathbf{A}\mathbf{v} = \mathbf{z} \\ +\infty, & \text{其他} \end{cases} \\ L_C(\mathbf{z}) = \begin{cases} 0, & \mathbf{z} \in \eta \\ +\infty, & \text{其他} \end{cases} \end{cases} \quad (7)$$

$L_{\mathbf{A}\mathbf{v}=\mathbf{z}}(\cdot)$ 和 $L_C(\cdot)$ 表示若满足约束条件则式(6)可解; 若约束不被满足, 则问题(6)无法求解。

式(6)的原始残差 r_{prim}^k 和对偶残差 r_{dual}^k 定义为

$$r_{\text{prim}} = \mathbf{A}\mathbf{v} - \mathbf{z} \quad (8)$$

$$r_{\text{dual}} = \mathbf{P}\mathbf{v} + \mathbf{q} + \mathbf{A}^T \mathbf{y} \quad (9)$$

根据上述, 为式(6)构造的增广拉格朗日函数为

$$\begin{aligned} L_{\rho, \sigma}(\tilde{\mathbf{v}}, \tilde{\mathbf{z}}, \mathbf{v}, \mathbf{z}, \mathbf{w}, \mathbf{y}) &= \frac{1}{2} \tilde{\mathbf{v}}^T \mathbf{P} \tilde{\mathbf{v}} + \mathbf{q}^T \tilde{\mathbf{v}} + L_{\mathbf{A}\mathbf{v}=\mathbf{z}}(\tilde{\mathbf{v}}, \tilde{\mathbf{z}}) \\ &+ L_C(\mathbf{z}) + \mathbf{w}^T (\tilde{\mathbf{v}} - \mathbf{v}) + \mathbf{y}^T (\tilde{\mathbf{z}} - \mathbf{z}) \\ &+ \frac{\sigma}{2} \|\tilde{\mathbf{v}} - \mathbf{v}\|^2 + \frac{\rho}{2} \|\tilde{\mathbf{z}} - \mathbf{z}\|^2 \end{aligned} \quad (10)$$

其中, $\sigma > 0, \rho > 0$ 是步长参数, \mathbf{w} 和 \mathbf{y} 是约束条件对应的拉格朗日乘子。以下给出解式(6)的2元裂解算子ADMM算法

$$\begin{aligned} (\tilde{\mathbf{v}}^{k+1}, \tilde{\mathbf{z}}^{k+1}) = \arg \min_{(\tilde{\mathbf{v}}, \tilde{\mathbf{z}})} & \frac{1}{2} \tilde{\mathbf{v}}^T \mathbf{P} \tilde{\mathbf{v}} + \mathbf{q}^T \tilde{\mathbf{v}} \\ & + \left(\frac{\sigma}{2}\right) \|\tilde{\mathbf{v}} - \mathbf{v}^k + \sigma^{-1} \mathbf{w}^k\|_2^2 \\ & + \left(\frac{\rho}{2}\right) \|\tilde{\mathbf{z}} - \mathbf{z}^k + \rho^{-1} \mathbf{y}^k\|_2^2 \end{aligned} \quad (11)$$

$$\mathbf{v}^{k+1} = \arg \min L_{\rho, \sigma}(\tilde{\mathbf{v}}^{k+1}, \mathbf{v}^k, \sigma^{-1} \mathbf{w}^k) \quad (12)$$

$$\mathbf{z}^{k+1} = \arg \min L_{\rho, \sigma}(\tilde{\mathbf{z}}^{k+1}, \mathbf{z}^k, \rho^{-1} \mathbf{y}^k) \quad (13)$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \sigma(a\tilde{\mathbf{v}}^{k+1} + (1-a)\mathbf{v}^k - \mathbf{v}^{k+1}) \quad (14)$$

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \rho(a\tilde{\mathbf{z}}^{k+1} + (1-a)\mathbf{z}^k - \mathbf{z}^{k+1}) \quad (15)$$

其中, 式(11)通过求解一个最小化问题得到 $\tilde{\mathbf{v}}^{k+1}, \tilde{\mathbf{z}}^{k+1}$, 然后通过式(12)和式(13)更新 $\mathbf{v}^{k+1}, \mathbf{z}^{k+1}$ 而得到 $\mathbf{v}^{k+1}, \mathbf{z}^{k+1}$ 。式(14)和式(15)更新约束条件对应的拉格朗日乘子值, 供下一次迭代求解式(11)~式(13)用。 $a \in (0, 2)$ 是松弛参数。当初始变量 $\mathbf{w}^0 = \mathbf{0}$ 时, 根据式(12)和式(14), 对所有的 $k \geq 0$ 都有 \mathbf{w}^{k+1} 恒为 $\mathbf{0}$, 因此可以将式(14)从迭代过程中去除, 式(11)对应的KKT条件为

$$\left. \begin{aligned} \mathbf{P}\tilde{\mathbf{v}}^{k+1} + \mathbf{q} + \sigma(\tilde{\mathbf{v}}^{k+1} - \mathbf{v}^k) + \mathbf{A}^T \mathbf{u}^{k+1} &= \mathbf{0} \\ \rho(\tilde{\mathbf{z}}^{k+1} - \mathbf{z}^k) + \mathbf{y}^k - \mathbf{u}^{k+1} &= \mathbf{0} \\ \mathbf{A}\tilde{\mathbf{v}}^{k+1} - \tilde{\mathbf{z}}^{k+1} &= \mathbf{0} \end{aligned} \right\} \quad (16)$$

其中, $\tilde{\mathbf{v}}^{k+1}, \tilde{\mathbf{z}}^{k+1}$ 是原问题的最优解, \mathbf{u}^{k+1} 是对偶问题的最优解, 并且 \mathbf{u} 是对应于 $\mathbf{A}\tilde{\mathbf{v}} = \tilde{\mathbf{z}}$ 条件的对偶变量, 利用消元法将 $\tilde{\mathbf{z}}^{k+1}$ 去掉得到

$$\begin{bmatrix} \mathbf{P} + \sigma \mathbf{I} & \mathbf{A}^T \\ \mathbf{A} & -\rho^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}}^{k+1} \\ \mathbf{u}^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma \mathbf{v}^k - \mathbf{q} \\ \mathbf{z}^k - \rho^{-1} \mathbf{y}^k \end{bmatrix} \quad (17)$$

根据上述式子, 表1给出了解2次规划问题的2元裂解算子ADMM算法流程, 表1算法的部分细节说明如下:

步骤1.1采用LDL矩阵分解法解线性方程组, 其中 \mathbf{L} 是一个下三角矩阵, \mathbf{D} 是一个主对角具有非零元素的对角矩阵。然后再利用三角矩阵进行向前或向后替换求解目标值。步骤1.2求分裂算子 $\tilde{\mathbf{z}}$ 是根据KKT条件式(18)得出的计算公式; 步骤1.3和1.4等价于式(14)和式(15), 将分裂算子 $\tilde{\mathbf{v}}, \tilde{\mathbf{z}}$ 作为中间变量来求得最优的 \mathbf{v}, \mathbf{z} 值, 其中 Π_C 代表在 C 上的欧几里得投影; 步骤1.5是求拉格朗日乘子 \mathbf{y} 的迭代值, 与传统ADMM算法类似。步骤2的终止条件为残差的范数 $r_{\text{prim}}^k, r_{\text{dual}}^k$ 小于公差 $\varepsilon_{\text{prim}} > 0, \varepsilon_{\text{dual}} > 0$, 即

$$\|r_{\text{prim}}^k\|_{\infty} \leq \varepsilon_{\text{prim}}, \|r_{\text{dual}}^k\|_{\infty} \leq \varepsilon_{\text{dual}} \quad (18)$$

公差为

$$\begin{aligned} \varepsilon_{\text{prim}} &= \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} + \max \{ \|\mathbf{A}\mathbf{v}^k\|_{\infty}, \|\mathbf{z}^k\|_{\infty} \}, \\ \varepsilon_{\text{dual}} &= \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} + \max \{ \|\mathbf{P}\mathbf{v}^k\|_{\infty}, \|\mathbf{A}^T \mathbf{y}^k\|_{\infty}, \|\mathbf{q}\|_{\infty} \} \end{aligned} \quad (19)$$

其中, $\varepsilon_{\text{abs}} > 0, \varepsilon_{\text{rel}} > 0$ 分别为绝对公差和相对公差。

表1中的参数说明如下: 根据上文可知 $\mathbf{v}^0, \mathbf{z}^0$ 为待求解变量的初始值, \mathbf{y}^0 为对应的拉格朗日乘子, $\rho > 0, \sigma > 0$ 为步长参数, a 是松弛参数, 半正定矩阵 \mathbf{P} , 单位矩阵 \mathbf{A} , 向量 \mathbf{q} 在式(5)下方进行了说明, 单位矩阵 $\mathbf{I} \in R^{N \times N}$, $\tilde{\mathbf{v}}, \tilde{\mathbf{z}}$ 是上文中提到的分裂算子。由式(16)可得 \mathbf{u}^{k+1} 是对偶问题的最优解, Π_C 代表在 C 上的欧几里得投影, $\tilde{\mathbf{v}}^{k+1}, \tilde{\mathbf{z}}^{k+1}$ 是原问题的最优解, \mathbf{y}^{k+1} 是其对应的拉格朗日乘子。

3.2 BSADMM-KELM

综合上述, 2元裂解算子ADMM核极限学习机(BSADMM-KELM)算法流程如表2所示, 其中步骤2中的预处理是指将数据集进行标准化处理。

3.3 BSADMM-KELM算法时间复杂度分析

表2算法的步骤1包含了初始化操作时间复杂度为 $O(1)$ 。步骤2进行数据预处理, 其计算量与数据集的大小 N 成正比时间复杂度为 $O(N)$ 。步骤3计算 \mathbf{P} 值, 其中包括核矩阵的计算, 计算时间复杂度为

表1 解QP问题的2元裂解算子ADMM算法流程

输入: $\mathbf{v}^0, \mathbf{z}^0, \mathbf{y}^0$, 参数 $\rho > 0, \sigma > 0, a \in (0, 2)$, 半正定矩阵 \mathbf{P} , 矩阵 \mathbf{A} , 单位矩阵 \mathbf{I} , 向量 \mathbf{q} 。
输出: 最优解 \mathbf{v} 。
步骤1 通过迭代寻找最优解:
1.1 解矩阵形式 $\begin{bmatrix} \mathbf{P} + \sigma \mathbf{I} & \mathbf{A}^T \\ \mathbf{A} & -\rho^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}}^{k+1} \\ \mathbf{u}^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma \mathbf{v}^k - \mathbf{q} \\ \mathbf{z}^k - \rho^{-1} \mathbf{y}^k \end{bmatrix}$ 得到 $\tilde{\mathbf{v}}^{k+1}, \mathbf{u}^{k+1}$ 的值;
1.2 $\tilde{\mathbf{z}} = \mathbf{z}^k + \rho^{-1}(\mathbf{u}^{k+1} - \mathbf{y}^k)$
1.3 $\mathbf{v}^{k+1} = a\tilde{\mathbf{v}}^{k+1} + (1-a)\mathbf{v}^k$
1.4 $\mathbf{z}^{k+1} = \Pi_C(a\tilde{\mathbf{z}}^{k+1} + (1-a)\mathbf{z}^k + \rho^{-1}\mathbf{y}^k)$
1.5 $\mathbf{y}^{k+1} = \mathbf{y}^k + \rho(a\tilde{\mathbf{z}}^{k+1} + (1-a)\mathbf{z}^k - \mathbf{z}^{k+1})$
步骤2 在约束条件内如果满足终止条件则输出 \mathbf{v} , 否则转步骤1进行下一次迭代。

表 2 BSADMM-KELM的训练算法

输入: 训练数据集 X , 测试数据集 Y , 约束条件 C , 核参数 δ , 矩阵 A , 单位矩阵 I , 向量 q 。
输出: 训练集和测试集分类准确率。
步骤1 载入训练样本和测试样本;
步骤2 对数据集 X, Y 进行预处理;
步骤3 根据模型式(5)计算半正定矩阵 P ;
步骤4 传入 P, C, A, I, q 调用表1算法, 返回 N 个 v_i 的最优值组成的 $v = [v_1, v_2, \dots, v_N]$;
步骤5 根据式(3)上面的性质求出所有的支持向量 x_s ;
步骤6 对于训练集 X 和测试集 Y 所有样本, 根据式(3)计算其对应的分类类别;
步骤7 将每个样本得到的分类类别其真实值对比, 所有样本对比完后输出分类准确率。

$O(N^2)$ 。步骤4中调用表1算法计算2次规划问题时间复杂度为 $O(t) \times O(N^3)$, 其中 t 是算法的迭代次数。步骤6的计算量与支持向量的个数有关, 计算时间复杂度为 $O(N_s)$ 。综上所述整个算法的时间复杂度为 $O(tN^3 + N^2 + N + N_s + 1)$, 因此所提算法的计算时间复杂度为多项式时间。

4 仿真实验与分析

实验所用的22个UCI数据集(<http://archive.ics.uci.edu/ml/index.php>)如表3所示。

表4是全部22个数据集上所提算法与传统核极限学习机算法、另外两种迭代式核极限学习机算法训练时间和分类精度的比较, 所有结果均来自于自测。所对比的算法KELM是传统解线性方程组的训练算法, 基于子空间搜索的有效集法解一般界定约束2次规划(General Definite and Bound Constrained Indefinite Quadratic Programming in Matlab based on subspace searches and an active set strategy, MINQ^[15])是采用有效集法解2次规划模型KELM; MOSEK数学软件优化包(MOSEK Optimization Tools, MOSEK)是一个知名的最优化问题商业求解器, 其所提供的内点法(Interior Point Method, IPM)是解大规模

2次规划问题的代表性迭代算法, 具有较快的计算速度。本文将BSADMM-KELM中求解算法用这两种算法替代, 分别得到它们构建的两种极限学习机MINQ-KELM和MOSEK-KELM, 通过对KELM, MINQ-KELM和MOSEK-KELM进行实验得到训练时间和分类精度。

表4给出了22个数据集上KELM, MINQ-KELM, MOSEK-KELM, BSADMM-KELM的训练时间和精度。训练时间方面: BSADMM-KELM在15个数据集上最快, MOSEK-KELM在1个数据集上最快, KELM在6个数据集上最快。在数据集规模大于1000的10个数据集上, 本文算法在其中的7个上优于传统KELM, 也就是在中型、大型数据集本文算法的训练时间优于传统KELM; 而MOSEK-KELM仅有1个数据集比KELM的训练时间快; 而MINQ-KELM的表现较差。本文算法与MINQ-KELM相比, 在全部22个数据集上的训练时间平均快29倍, 与MOSEK-KELM相比平均快4倍。在分类精度方面: 在全部22个数据集上, 本文算法均优于传统的KELM, 分类精度平均提高了10%; 本文算法较MINQ-KELM的分类精度平均高出5.3%, 较MOSEK-KELM平均高出3.6%。综述表4分析结果可知, 本文引入2元

表 3 实验所用UCI数据集列表

数据集	样本数量	属性数目	数据集	样本数量	属性数目
Banknote	1372	4	Debrecen	1151	20
Fertility	100	10	Creditcard	30000	24
Hill	606	101	Occupancy	20560	7
Monks1	432	7	HTRU2	17898	9
Monks2	432	7	Heart	270	13
SPECTF	267	44	Australian	690	14
Pdspeech	1040	26	Liver	345	7
Epileptic	11500	179	Ionosphere	351	34
Mushroom	8124	22	Pima	768	9
Spambase	4601	57	Cancer	683	10
Adult	48842	14	Sonar	258	60

表4 4种KELM算法训练时间和计算精度比较(自测)

数据集	KELM		MINQ-KELM		MOSEK-KELM		BSADMM-KELM	
	精度(%)	求解时间(s)	精度(%)	求解时间(s)	精度(%)	求解时间(s)	精度(%)	求解时间(s)
Banknote	100.00	0.0673	100.00	1.5989	100.00	0.3014	100.00	0.0521
Fertility	86.00	0.0029	92.15	0.0999	91.25	0.0090	91.25	0.0018
Heart	82.96	0.0032	86.11	0.1318	88.43	0.0074	90.74	0.0007
Hill	58.09	0.0128	62.47	5.4970	60.41	0.1390	62.06	0.1950
Monks1	65.74	0.0070	97.36	0.1214	96.29	0.0361	98.31	0.0048
Monks2	68.98	0.0074	98.88	0.0765	99.63	0.0163	99.87	0.0021
SPECTF	85.71	0.0034	92.06	0.3839	93.93	0.0512	95.93	0.0107
Pdspeech	92.89	0.0878	89.40	2.7579	88.08	0.3429	94.55	0.0696
Epileptic	94.24	6.7190	95.90	86.0792	95.90	8.9292	96.78	8.4618
Mushroom	95.55	4.8600	95.31	23.4706	99.31	8.4933	98.40	5.5572
Spambase	88.04	0.6364	90.46	4.7311	92.02	1.6849	92.77	0.5321
Adult	81.74	89.2852	84.08	851.8944	84.19	134.0564	85.10	69.5138
Debreceen	62.61	0.0328	69.06	0.9339	69.92	0.0475	71.30	0.0324
Creditcard	77.00	80.2243	72.33	752.0268	80.46	97.3124	79.08	53.0256
Occupancy	98.05	25.8158	98.94	78.6522	98.95	30.4110	99.02	22.7719
HTRU2	96.37	14.2591	96.93	78.5169	97.88	10.4880	97.92	11.2290
Australian	88.22	0.0144	83.88	0.3019	84.63	0.3108	90.72	0.0198
Liver	63.77	0.0061	60.87	0.1657	66.30	0.0115	76.95	0.0059
Ionosphere	87.14	0.0042	84.64	0.0486	88.57	0.0095	96.43	0.0036
Pima	74.68	0.0145	75.41	0.4374	75.90	0.0177	79.12	0.0111
Cancer	95.79	0.0143	96.34	0.1310	94.87	0.0196	96.58	0.0170
Sonar	83.33	0.0026	75.90	0.0293	77.71	0.0066	95.58	0.0018

裂解算子ADMM能有效提高KELM的训练速度和分类精度。

图1展示了6个比较有代表性数据集上 C 和 δ 的取值对极限学习机分类准确率的影响。参数 C 和 δ 之间是一种组合优化关系,若考虑对两者进行优化, KELM模型和算法需作较大改动,这已超出本文的关注点,本文用最简单的网格搜索法(grid search)来获取 C 和 δ 的近优组合,其中 C 分别取(0.005, 0.050, 0.010, 0.100, 1.000, 10.000, 100.000), δ 分别取(0.05, 0.01, 0.10, 1.00, 2.00, 5.00, 10.00, 50.00)。由图1可看出, BSADMM-KELM对核参数 δ 较敏感,当 δ 从0开始增大时, δ 越大分类准确率越高;但当 δ 取值过大时,准确率不但没有提高,反而下降。固定 δ 值时,训练准确率随参数 C 取值变化的趋势较缓;而固定 C 值时,训练准确率随参数 δ 变化的趋势较为剧烈(断面出现明显的峰值曲线),这说明 δ 是影响分类精度的关键参数。在22个数据集上,本文所获的 C 和 δ 近优组合值如表5所示。

表6给出了所提算法与近年5种相关算法的分类精度比较。本文选取相关文献中较常出现的7个数

数据集并在此基础上与文献结果比较。5种相比较的算法分别是基于新的半正定内核函数的核极限学习机(Extreme Learning Machines With Expectation Kernels, EKELM)^[16]、双极限学习机(Twin Extreme Learning Machines, TELM)^[17]、 ν 正则化极限学习机(Optimization Extreme Learning Machine With ν Regularization, ν -OELM)^[18]、基于连续优化算法的稀疏极限学习机(A Sparse Extreme Learning Machine Framework By Continuous Optimization Algorithms, Sparse-ELM)^[19]、基于半监督学习的拉普拉斯双极限学习机(Laplacian Twin Extreme Learning Machine, LapTELM)^[20]。相比较算法的分类精度取自对应的文献,表中“-”表示该算法文献未给出此数据集的分类精度。由表6可知,本文所提算法在Australian, Heart, Ionosphere, Pima, Sonar数据集分类精度均优于相比较的5种算法,分类精度相比于其他5种算法精度最高的分别提高了5%, 5.59%, 0.72%, 1.41%, 4.09%。而在Liver数据集上,所提算法较TELM精度低0.01%, Cancer数据集上较EKELM精度低了0.78%, 差距较小。

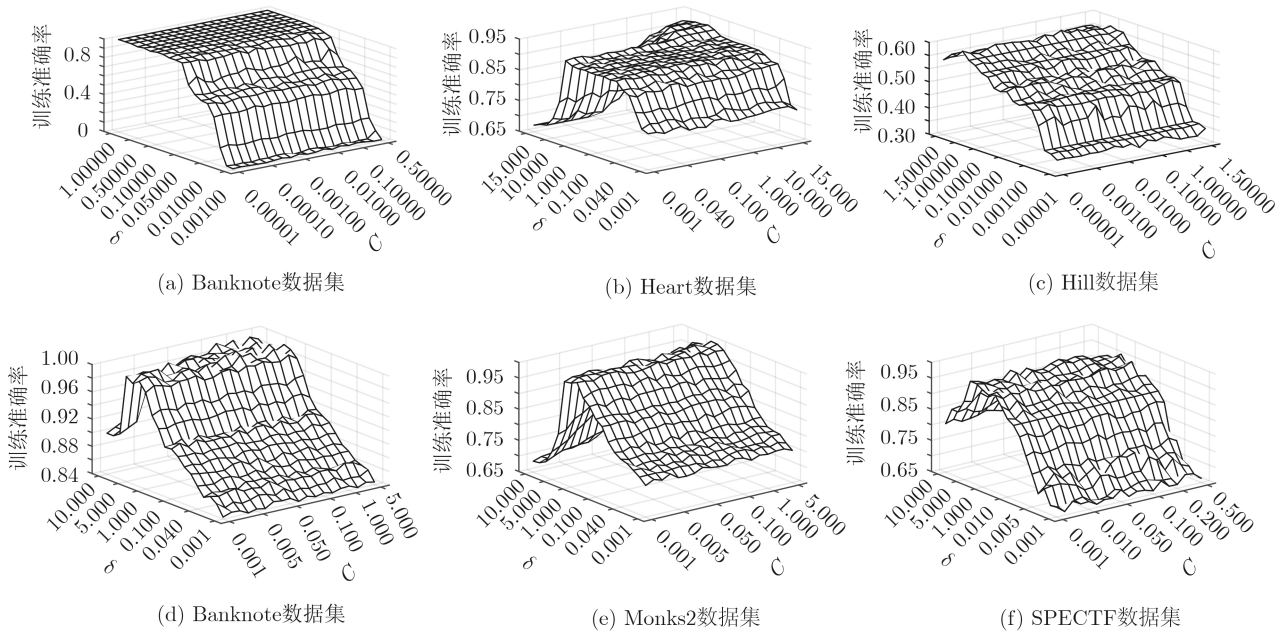


图1 C 和 δ 对分类准确率的影响

表5 C 和 δ 的格子搜索法近优组合值

数据集	C	δ	数据集	C	δ
Banknote	0.100	0.01	Debrecen	1.000	2.00
Fertility	0.005	0.01	Creditcard	1.000	10.00
Hill	0.010	0.10	Occupancy	0.500	5.00
Monks1	1.000	5.00	HTRU2	0.010	10.00
Monks2	0.500	2.00	Heart	10.000	5.00
SPECTF	0.010	2.00	Australian	10.000	5.00
Pdspeech	0.050	5.00	Liver	10.000	0.05
Epileptic	1.000	50.00	Ionosphere	0.010	1.00
Mushroom	0.010	0.10	Pima	0.100	0.01
Spambase	0.100	10.00	Cancer	0.500	2.00
Adult	10.000	5.00	Sonar	100.000	10.00

表6 6种极限学习机算法分类精度比较(%)

数据集	BSADMM-KELM	EKELM ^[16]	TELM ^[17]	v -OELM ^[18]	Sparse ELM ^[19]	LapTELM ^[20]
Australian	90.72	85.22	75.14	68.20	72.49	75.51
Liver	76.95	71.51	76.96	72.41	-	72.10
Heart	88.74	-	83.15	76.00	70.74	76.09
Ionosphere	96.43	95.44	95.71	90.13	91.80	90.94
Pima	79.12	-	77.71	77.23	69.34	68.94
Cancer	96.58	97.36	-	96.58	94.25	-
Sonar	95.58	89.40	91.49	78.80	68.50	85.68

在7个数据集上，本文算法的分类精度总体优于相比较的算法，这说明本文引入BSADMM改进KELM的分类精度是可行的、有效的。

图2给出了本文算法的时间复杂度特征曲线图。将表6的7个数据集通过数据复制方法分别以2,

4, 8, 16倍的规模扩大数据集，最终形成样本更大的新数据集，在新数据集上应用本文算法算出训练时间，绘制出所提算法的训练时间随数据集规模变化的趋势。从图2可知，当数据规模呈指数级增长时，本文算法的训练时间为多项式曲线变化，实

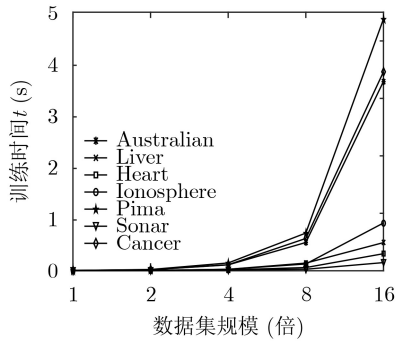


图2 所提算法时间复杂度特征曲线

验结果与本文算法计算时间复杂度分析的结论相吻合。

5 结束语

本文将传统KELM寻找最优输出层权重 β 值的过程变成了2次规划问题,提出用2元裂解算子交替乘法改进核极限学习机的训练时间和分类精度。首先使用算子分裂交替乘法解2次规划问题得出支持向量,再结合支持向量机的输出函数得到BSADMM-KELM的输出。由实验分析本文得到两个结论:一是2元裂解算子的交替乘法解2次规划KELM速度较快,甚至优于MOSEK最优化商业软件包,采用迭代法来研究KELM的训练算法仍有探讨空间,迭代法的训练速度不一定慢于传统的解线性方程组法;二是采用2次规划模型的KELM其分类精度均优于传统KELM模型,从凸优化模型角度研究提高KELM的分类精度是可行的。但BSADMM-KELM在大型数据集上的运算速度还是略有不足,而且 $C-\delta$ 的最优组合本文并没有找到。因此本文下一步的工作重点放在两个方面,一方面应该在保证准确率的情况下尽可能减少算法的迭代次数,提高算法运算速度,另一方面是寻找 $C-\delta$ 的最优组合。

参考文献

- [1] HUANG Guangbin, ZHU Qinyu, SIEW C K, *et al.* Extreme learning machine: Theory and applications[J]. *Neurocomputing*, 2006, 70(1/3): 489–501. doi: [10.1016/j.neucom.2005.12.126](https://doi.org/10.1016/j.neucom.2005.12.126).
- [2] QING Yuanyuan, ZENG Yijie, LI Yue, *et al.* Deep and wide feature based extreme learning machine for image classification[J]. *Neurocomputing*, 2020, 412: 426–436. doi: [10.1016/j.neucom.2020.06.110](https://doi.org/10.1016/j.neucom.2020.06.110).
- [3] CHEN Xudong, YANG Haiyue, WUN Jhangshang, *et al.* Power load forecasting in energy system based on improved extreme learning machine[J]. *Energy Exploration & Exploitation*, 2020, 38(4): 1194–1211. doi: [10.1177/0144598720903797](https://doi.org/10.1177/0144598720903797).
- [4] KHAN M A, ABBAS S, KHAN K M, *et al.* Intelligent forecasting model of COVID-19 novel coronavirus outbreak empowered with deep extreme learning machine[J]. *Computers, Materials & Continua*, 2020, 64(3): 1329–1342. doi: [10.32604/cmc.2020.011155](https://doi.org/10.32604/cmc.2020.011155).
- [5] MUTHUNAYAGAM M and GANESAN K. Cardiovascular disorder severity detection using myocardial anatomic features based optimized extreme learning machine approach[J]. *IRBM*, To be published. doi: [10.1016/j.irbm.2020.06.004](https://doi.org/10.1016/j.irbm.2020.06.004).
- [6] WU Yu, ZHANG Yongshan, LIU Xiaobo, *et al.* A multiobjective optimization-based sparse extreme learning machine algorithm[J]. *Neurocomputing*, 2018, 317: 88–100. doi: [10.1016/j.neucom.2018.07.060](https://doi.org/10.1016/j.neucom.2018.07.060).
- [7] ZHANG Yongshan, WU Jia, CAI Zhihua, *et al.* Memetic extreme learning machine[J]. *Pattern Recognition*, 2016, 58: 135–148. doi: [10.1016/j.patcog.2016.04.003](https://doi.org/10.1016/j.patcog.2016.04.003).
- [8] PENG Yong, KONG Wanzeng, and YANG Bing. Orthogonal extreme learning machine for image classification[J]. *Neurocomputing*, 2017, 266: 458–464. doi: [10.1016/j.neucom.2017.05.058](https://doi.org/10.1016/j.neucom.2017.05.058).
- [9] ZUO Bai, HUANG Guangbin, WANG Danwei, *et al.* Sparse extreme learning machine for classification[J]. *IEEE Transactions on Cybernetics*, 2014, 44(10): 1858–1870. doi: [10.1109/TCYB.2014.2298235](https://doi.org/10.1109/TCYB.2014.2298235).
- [10] GHADIMI E, TEIXEIRA A, SHAMES I, *et al.* Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems[J]. *IEEE Transactions on Automatic Control*, 2015, 60(3): 644–658. doi: [10.1109/TAC.2014.2354892](https://doi.org/10.1109/TAC.2014.2354892).
- [11] LIU Yuanyuan, SHANG Fanhua, and CHENG J. Accelerated variance reduced stochastic ADMM[C]. The Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017: 2287–2293.
- [12] DHAR S, YI Congrui, RAMAKRISHNAN N, *et al.* ADMM based scalable machine learning on Spark[C]. 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, USA, 2015.
- [13] STELLATO B, BANJAC G, GOULART P, *et al.* OSQP: An operator splitting solver for quadratic programs[J]. *Mathematical Programming Computation*, 2020, 12(4): 637–672. doi: [10.1007/s12532-020-00179-2](https://doi.org/10.1007/s12532-020-00179-2).
- [14] HUANG Guangbin, DING Xiaojian, and ZHOU Hongming. Optimization method based extreme learning machine for classification[J]. *Neurocomputing*, 2010, 74(1/3): 155–163. doi: [10.1016/j.neucom.2010.02.019](https://doi.org/10.1016/j.neucom.2010.02.019).
- [15] HUYER W and NEUMAIER A. MINQ: General definite and bound constrained indefinite quadratic programming[EB/OL]. <https://www.mat.univie.ac.at/~neum/software/minq/>, 2017.

- [16] ZHANG Wenyu, ZHANG Zhenjiang, WANG Lifu, *et al.* Extreme learning machines with expectation kernels[J]. *Pattern Recognition*, 2019, 96: 106960. doi: [10.1016/j.patcog.2019.07.005](https://doi.org/10.1016/j.patcog.2019.07.005).
- [17] WAN Yihe, SONG Shiji, HUANG Gao, *et al.* Twin extreme learning machines for pattern classification[J]. *Neurocomputing*, 2017, 260: 235–244. doi: [10.1016/j.neucom.2017.04.036](https://doi.org/10.1016/j.neucom.2017.04.036).
- [18] DING Xiaojian, LAN Yuan, ZHANG Zhifeng, *et al.* Optimization extreme learning machine with ν regularization[J]. *Neurocomputing*, 2017, 261: 11–19. doi: [10.1016/j.neucom.2016.05.114](https://doi.org/10.1016/j.neucom.2016.05.114).
- [19] YANG Liming and ZHANG Siyun. A sparse extreme learning machine framework by continuous optimization algorithms and its application in pattern recognition[J]. *Engineering Applications of Artificial Intelligence*, 2016, 53: 176–189. doi: [10.1016/j.engappai.2016.04.003](https://doi.org/10.1016/j.engappai.2016.04.003).
- [20] LI Shuang, SONG Shiji, and WAN Yihe. Laplacian twin extreme learning machine for semi-supervised classification[J]. *Neurocomputing*, 2018, 321: 17–27. doi: [10.1016/j.neucom.2018.08.028](https://doi.org/10.1016/j.neucom.2018.08.028).
- 苏一丹：男，1962年生，教授，研究方向为自然计算、数据挖掘。
续 嘉：男，1998年生，硕士生，研究方向为机器学习、数据挖掘。
覃 华：男，1972年生，教授，研究方向为凸优化机器学习。
- 责任编辑：余 蓉