优化极限学习机的序列最小优化方法

丁晓剑，赵银亮
（西安交通大学电子与信息工程学院，710049，西安）

摘要：针对传统二次规划求解方法训练优化极限学习机(OMELM)存在速度慢和效率低的问题，提出了单变量迭代序列最小优化(SSMO)算法.该算法通过在框式约束中优化拉格朗日乘子来实现目标函数的最小化：首先在初始化拉格朗日乘子中选择使目标函数值下降最大的拉格朗日乘子，将该拉格朗日乘子作为目标函数的唯一变量；然后求解目标函数的最小值并更新该变量的值；重复这个过程直到所有的拉格朗日乘子都满足二次规划问题的 Karush-Kuhn-Tucker 条件为止.实验结果表明：SSMO 算法只需调节很少的参数值便可得到足够好的泛化性能；采用 SSMO 算法的 OMELM 方法在泛化性能上要好于采用序列最小优化算法的支持向量机方法；在随机数据集测试中，SSMO 算法具有较好的鲁棒性.

关键词：极限学习机；支持向量机；序列最小优化
中图分类号：TP181 **文献标志码**：A **文章编号**：0253-987X(2011)06-0007-06

A Sequential Minimal Optimization Method for Optimization Extreme Learning Machine

DING Xiaojian, ZHAO Yinliang
(School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: A sequential minimal optimization (SSMO) algorithm iterated with single variable is proposed to solve the slow speed and low efficiency problems in training optimization method based extreme learning machine (OMELM) by traditional quadratic programming solver. The algorithm searches the minimum value of the objective function by optimizing Lagrange multipliers within box constraints. The Lagrange multiplier that can generate maximum reduction in the objective function is selected from the initial Lagrange multipliers as for the unique variable of the objective function. Then the objective function is minimized with respect to the variable, and a new value of the Lagrange multiplier is obtained. The process is repeated until all Lagrange multipliers satisfy the Karush-Kuhn-Tucker condition of the quadratic programming problem. Experimental results show that SSMO algorithm can obtain good enough generalization performance with adjusting few parameter values. The generalization performance of OMELM method using SSMO algorithm is better than that of the support vector machine (SVM) method using sequential minimal optimization (SMO) algorithm. SSMO algorithm is robust in random datasets trails.

Keywords: extreme learning machine; support vector machine; sequential minimal optimization

近年来，基于单隐层神经网络的极限学习机(extreme learning machine，ELM)模型[1]得到了广泛的关注[2-4].该模型对网络的输入权值和隐层节点偏置进行随机赋值，只需通过一步计算即可解析地

收稿日期：2010-09-09. 作者简介：丁晓剑(1982—)，男，博士生；赵银亮(联系人)，男，教授，博士生导师. 基金项目：国家高技术研究发展计划资助项目(2008AA01Z136).
网络出版时间：2011-03-25 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20110325.1511.006.html

求出网络的输出权值. ELM 可极大地提高网络的学习速度,并以较强的泛化性能实现机器学习任务. 由于神经网络在训练样本较小时容易发生过拟合问题,所以 ELM 在小样本数据上的学习能力不太理想.

Bartlett 理论[5]指出,前馈型神经网络训练误差较小时,网络输出权值的范数值越小,网络具备的泛化性能就越好. 基于最小化网络输出权值的思想,Huang 等人[6]提出了优化方法的极限学习机(optimization method based extreme learning machine, OMELM). OMELM 既克服了传统的 ELM 方法在小样本数据集测试中泛化性能较低的缺点,又保留了 ELM 方法随机参数赋值的优势.

训练 OMELM 的主要计算代价是求解其二次规划问题,OMELM 的二次规划问题是具有框式约束的新型优化问题. 传统的方法由于无法有效利用框式约束的特点,求解时会出现速度较慢和迭代效率较低的问题,本文深入研究了框式约束和目标函数的关系,提出了基于单变量迭代的序列最小优化(SSMO)求解算法,有效地提高了 OMELM 的训练速度和计算效率. SSMO 算法能以较低的计算代价完成目标函数值的快速下降,并且只需调节很少的参数值便可得到足够好的泛化性能.

1 ELM 优化模型

1.1 ELM 特征空间

给定一个含有 N 个样本的训练样本集 $\{\boldsymbol{x}_i, t_i\}_{i=1}^N$,其中输入为 d 维向量 $\boldsymbol{x}_i \in \mathbf{R}$,输出为 $t_i \in \{-1,+1\}$,则 ELM 网络的输出为

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{L} \beta_j G(\boldsymbol{a}_j, b_j, \boldsymbol{x}) = \boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i) \quad (1)$$

式中:$\boldsymbol{a}_i$ 为连接第 j 个隐节点的输入权值;b_j 是第 j 个隐节点的偏差;β_j 为第 j 个隐节点到 ELM 网络输出节点的权值;$G(\boldsymbol{a}_j, b_j, \boldsymbol{x}_i) = G(\boldsymbol{a}_j \cdot \boldsymbol{x}_i + b_j)$ 为第 j 个隐节点的加乘型输出函数. $\boldsymbol{h}(\boldsymbol{x}_i) = [G(\boldsymbol{a}_1, b_1, \boldsymbol{x}_i), \cdots, G(\boldsymbol{a}_L, b_L, \boldsymbol{x}_i)]^{\mathrm{T}}$ 为隐藏层关于 $\boldsymbol{x}_i$ 的输出向量,$\boldsymbol{h}(\boldsymbol{x}_i)$ 的作用是将样本 $\boldsymbol{x}_i$ 由 d 维输入空间映射到 L 维的特征空间(ELM 特征空间) H. 对于两类分类问题,ELM 的决策函数可表示为

$$f(\boldsymbol{x}_i) = \mathrm{sgn}\left(\sum_{j=1}^{L} \beta_j G(\boldsymbol{a}_j, b_j, \boldsymbol{x}_i)\right) = \mathrm{sgn}(\boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i)) \quad (2)$$

1.2 OMELM

与传统的神经网络算法不同,ELM 网络的输出权值能以最小化模的方式计算. 根据 Bartlett 的理论[5],网络输出权值的模越小,该网络所具有的泛化性能就越好. 基于优化理论,最小化 ELM 的训练误差和网络输出权值的模

$$\left.\begin{aligned} &\min: \sum_{i=1}^{N} \| \boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i) - t_i \| \\ &\min: \| \boldsymbol{\beta} \| \end{aligned}\right\} \quad (3)$$

根据 ELM 理论[1],任意的训练样本集在 ELM 的特征空间中都是线性可分的,即可由 ELM 准确地拟合. 但是,ELM 在准确拟合所有训练样本时可能降低模型的推广性,需要引入误差容忍参数 ξ_i 来避免过拟合问题

$$\left.\begin{aligned} &\boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i) \geqslant 1 - \xi_i, \quad t_i = 1 \\ &\boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i) \leqslant -1 + \xi_i, \quad t_i = -1 \end{aligned}\right\} \quad (4)$$

式(4)可合并为

$$t_i(\boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i)) \geqslant 1 - \xi_i,\ i = 1,2,\cdots,N \quad (5)$$

利用代价参数 C 来平衡最小化输出权值的模 $\| \boldsymbol{\beta} \|$ 和最小化训练误差 ξ_i 的权重,则式(3)可表示为

$$\begin{aligned} &\min: \frac{1}{2} \| \boldsymbol{\beta} \|^2 + C \sum_{i=1}^{N} \xi_i \\ &\text{s.t. } t_i \boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i) \geqslant 1 - \xi_i \\ &\xi_i \geqslant 0,\ i = 1,2,\cdots,N \end{aligned} \quad (6)$$

式(6)称为 OMELM 的原始优化问题. 根据拉格朗日理论,求解式(6)等价于求解下面的对偶优化问题

$$\begin{aligned} &\min: L(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} t_i t_j \alpha_i \alpha_j \boldsymbol{h}(\boldsymbol{x}_i) \cdot \boldsymbol{h}(\boldsymbol{x}_j) - \sum_{i=1}^{N} \alpha_i \\ &\text{s.t. } 0 \leqslant \alpha_i \leqslant C,\ i = 1,2,\cdots,N \end{aligned} \quad (7)$$

式中:$\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_N]^{\mathrm{T}}$ 为引入的非负拉格朗日乘子,一般使用核函数来代替特征向量的内积形式. 为此,定义 ELM 核函数

$$K_{\mathrm{ELM}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{h}(\boldsymbol{x}_i) \cdot \boldsymbol{h}(\boldsymbol{x}_j) \quad (8)$$

OMELM 的对偶优化问题(式(7))与支持向量机(SVM)的对偶优化问题在本质上是有区别的:① SVM 对于不同的学习任务需要调节不同的参数,以选择合适的学习模型,OMELM 使用的是随机 ELM 核,与学习任务无关,ELM 核中的参数都是随机生成的,无需人工调节;②OMELM 的对偶优化问题与 SVM 的对偶优化问题相比,无需满足约束 $\sum_{i=1}^{N} \alpha_i t_i = 0$.

2 SSMO 算法

Platt[7]提出的 SMO 算法是广泛使用的求解

SVM 的算法，由于 SVM 对偶优化问题需要满足约束条件 $\sum_{i=1}^{N}\alpha_i t_i = 0$，所以 SMO 算法在每次迭代中需要优化 2 个变量 α_1 和 α_2，优化问题中的其他变量保持不变，使得

$$\alpha_1\ t_1 + \alpha_2\ t_2 + \sum_{i=3}^{N}\alpha_i t_i = \alpha_{1,\mathrm{new}}\ t_1 + \alpha_{2,\mathrm{new}}\ t_2 + \sum_{i=3}^{N}\alpha_i t_i = 0 \tag{9}$$

由于 OMELM 优化问题无需满足该约束条件，每次迭代只需优化一个变量 α_1，其他变量 $\alpha_2,\cdots,\alpha_N$ 保持不变. 设 $\alpha_{1,\mathrm{new}}=\alpha_1+g$，则式(7)的目标函数可写为

$$L(g) = \frac{1}{2}K_{11}t_1^2(\alpha_1+g)^2 + (\alpha_1+g)t_1\sum_{i=2}^{N}t_i\alpha_i K_{1j} - (\alpha_1+g) + S \tag{10}$$

式中

$$S = \frac{1}{2}\sum_{i=2}^{N}\sum_{j=2}^{N}\alpha_i\alpha_j y_i y_j K_{ij} - \sum_{i=2}^{N}\alpha_i \tag{11}$$

$$K_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{12}$$

S 为与变量 α_1 无关项. 由于 $t_1\in\{-1,+1\}$，有 $t_1^2=1$. 令 $f(\boldsymbol{x}_i)=\sum_{j=1}^{N}\alpha_j t_j K_{ij}$ 为 OMELM 分类器的输出函数，则有

$$L(g) = \frac{1}{2}K_{11}(\alpha_1+g)^2 + (\alpha_1+g)t_1(f_1-\alpha_1 t_1 K_{11}) - (\alpha_1+g) + S \tag{13}$$

式中：$f_1=f(\boldsymbol{x}_1)$. 求解式(13)的最小值即为求最优解 $g^*=\arg\min L(g)$. 当且仅当 $\partial L/\partial g=0$ 时，式(13)得到最优解 g^*，即有

$$\frac{\partial L}{\partial g} = (a_1+g^*)K_{11} + t_1(f_1-\alpha_1 t_1 K_{11}) - 1 = 0 \tag{14}$$

求式(14)可得 $g^*=(1-t_1f_1)/K_{11}$. 令 $E_i=f_i-t_i$ 为输出函数预测的误差值，则

$$g^* = (1-t_1(E_1+t_1))/K_{11} = -t_1E_1/K_{11} \tag{15}$$

优化后的变量 $\alpha_{1,\mathrm{new}}=\alpha_1-t_1E_1/K_{11}$，由于约束条件 $0\leqslant\alpha_i\leqslant C$，需要调整 $\alpha_{1,\mathrm{new}}$ 使之满足约束，调整后的变量为

$$\alpha_{1,\mathrm{new1}} = \begin{cases} C, & \alpha_{1,\mathrm{new}} \geqslant C \\ \alpha_{1,\mathrm{new}}, & 0 < \alpha_{1,\mathrm{new}} < C \\ 0, & \alpha_{1,\mathrm{new}} \leqslant 0 \end{cases} \tag{16}$$

当给定变量 α_1 优化为 $\alpha_{1,\mathrm{new1}}$ 后，目标优化问题的全部变量更新为 $\boldsymbol{\alpha}_{\mathrm{new}}=[\alpha_{1,\mathrm{new1}},\alpha_2,\cdots,\alpha_N]$，OMELM 的输出函数和输出函数预测的误差值分别更新为

$$f_{i,\mathrm{new}} = \alpha_{1,\mathrm{new1}}t_1K_{1i} + \sum_{j=2}^{N}\alpha_j t_j K_{ij};$$
$$E_{i,\mathrm{new}} = f_{i,\mathrm{new}} - t_i \tag{17}$$

2.1　选取需优化变量策略

由于目标是选取最小化的目标函数 $L(\boldsymbol{\alpha})$，所以选择优化变量的最佳策略是选择一个变量 α_i，使得函数 $L(\boldsymbol{\alpha})$的值下降最快，即 $L(\boldsymbol{\alpha})-L(\boldsymbol{\alpha}_{\mathrm{new}})$最大化

$$L(\boldsymbol{\alpha}) - L(\boldsymbol{\alpha}_{\mathrm{new}}) = \frac{1}{2}K_{ii}(\alpha_i^2 - (\alpha_{i,\mathrm{new}})^2) + (\alpha_i-\alpha_{i,\mathrm{new}})t_i(f_i-\alpha_i t_i K_{ii}) - (\alpha_i-\alpha_{i,\mathrm{new}}) \tag{18}$$

将 $\alpha_{i,\mathrm{new}}=\alpha_i+g^*$ 代入式(18)可得

$$L(\boldsymbol{\alpha}) - L(\boldsymbol{\alpha}_{\mathrm{new}}) = \frac{1}{2}K_{ii}(\alpha_i^2-\alpha_i^2-(g^*)^2 - 2\alpha_i g^*) - g^*t_i(f_i-\alpha_i t_i K_{ii}) + g^* = g^*\left(-\frac{1}{2}K_{ii}g^* - K_{ii}\alpha_i - t_i(f_i-\alpha_i t_i K_{ii}) + 1\right)\cdot g^*\left(-\frac{1}{2}K_{ii}g^* - t_i f_i + 1\right) \tag{19}$$

将 $g^*=(1-t_if_i)/K_{ii}$ 代入式(19)可得

$$L(\boldsymbol{\alpha}) - L(\boldsymbol{\alpha}_{\mathrm{new}}) = \frac{1-t_if_i}{K_{ii}}\left(-\frac{1}{2}K_{ii}\frac{1-t_if_i}{K_{ii}} - t_if_i + 1\right) = \frac{1-t_if_i}{K_{ii}}\frac{1-t_if_i}{2} = \frac{(t_iE_i)^2}{2K_{ii}} = \frac{E_i^2}{2K_{ii}} \tag{20}$$

所以每次迭代的优化变量 α_i 的指标 i 应该满足

$$i = \arg\max E_i^2/2K_{ii} \tag{21}$$

2.2　Karush-Kuhn-Tucker 条件

SSMO 算法是一种迭代求解方法，每次迭代完成时需要判断迭代得到的向量是否满足目标问题的 Karush-Kuhn-Tucker(KKT)条件[8]：如果满足，则算法得到最优解；否则，继续进行迭代操作.

构造式(7)的拉格朗日问题

$$\overline{L} := \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}t_it_jK(\boldsymbol{x}_i,\boldsymbol{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\lambda_i\alpha_i - \sum_{i=1}^{N}\mu_i(C-\alpha_i) \tag{22}$$

式中：λ_i 和 μ_i 为非负拉格朗日乘子. 式(7)的最优解必须满足下列条件：

(1) $\dfrac{\partial\overline{L}}{\partial\alpha_i}=0\Rightarrow t_iE_i-\lambda_i+\mu_i=0$；

(2) $0\leqslant\alpha_i\leqslant C,\forall i$；

(3) $\lambda_i\geqslant 0,\forall i$；

(4) $\mu_i\geqslant 0,\forall i$；

(5)$\lambda_i\alpha_i=0,\forall i$;

(6)$\mu_i(C-\alpha_i)=0,\forall i$.

其中,条件(1)为稳定点条件,条件(2)为式(7)的可行性条件,条件(3)、(4)为式(22)的可行性条件,条件(5)、(6)为式(7)和式(22)之间的互补松弛性条件.由以上这 6 个条件,可对 α_i 作如下划分:

(1)$\alpha_i=0$,由条件(6)可知 $\mu_i=0$,再由条件(3)可得 $t_iE_i\geqslant 0$;

(2)$\alpha_i=C$,由条件(5)可知 $\lambda_i=0$,再由条件(4)可得 $t_iE_i\leqslant 0$;

(3)$0<\alpha_i<C$,由条件(5)和条件(6)可知 $\lambda_i=0$,$\mu_i=0$,可得 $t_iE_i=0$.

因此,当且仅当以下条件成立时,式(7)才会得到最优解

$$\left.\begin{aligned} &t_iE_i>0,\ \alpha_i=0\\ &t_iE_i=0,\ 0<\alpha_i<C\\ &t_iE_i<0,\ \alpha_i=C \end{aligned}\right\} \tag{23}$$

综上分析,SSMO 算法可描述如下.

初始化 设 $k=0$,设初始可行迭代向量 $\boldsymbol{\alpha}_k$($\boldsymbol{\alpha}_k\in[0,C]^N$)、$\boldsymbol{f}_k$、$\boldsymbol{E}_k$ 和容忍参数 τ.

步骤 1 如果 $\boldsymbol{\alpha}_k$ 满足式(23)或者$|\boldsymbol{f}_k-\boldsymbol{f}_{k-1}|<\tau$,算法停止,否则根据式(21)选择更新变量 $\alpha_{i,k}$.

步骤 2 根据式(15)更新 $\boldsymbol{\alpha}_{i,\mathrm{new}}$和 $\boldsymbol{\alpha}^{\mathrm{new}}$,根据式(17)更新 $\boldsymbol{f}_{\mathrm{new}}$和 $\boldsymbol{E}_{\mathrm{new}}$.

步骤 3 令 $\boldsymbol{\alpha}_{k+1}=\boldsymbol{\alpha}_{\mathrm{new}}$,$\boldsymbol{f}_{k+1}=\boldsymbol{f}_{\mathrm{new}}$,$\boldsymbol{E}_{k+1}=\boldsymbol{E}_{\mathrm{new}}$,$k=k+1$,转向步骤 1.

在 SSMO 算法中,算法的停止条件采用 2 个准则:所有变量满足 KKT 条件或者函数值下降的值满足设定的容忍值.SSMO 算法每次迭代都选择使函数值下降最大的变量进行优化,如果某次迭代函数值下降非常小,则算法结束迭代.

3 实验分析与性能比较

为了验证 SSMO 算法的有效性,本文利用 SVM 和 OMELM 来进行泛化性能的比较.为了公平地进行比较,SVM 的求解方法使用 Platt 提出的 SMO 算法[7].所有算法都是基于 Pentium 4,2.53 GHZ,MATLAB2007 环境下实现.实验用到的数据来自 UCI 数据库[9]和 Statlog 数据库[10],数据的具体描述如表 1 所示.

3.1 参数设置

首先选择 OMELM 和 SVM 所需的核函数,OMELM 优化问题的核函数选择 Sigmoid 型函数 $G(\boldsymbol{a},\boldsymbol{b},\boldsymbol{x})=1/(1+\exp(-(\boldsymbol{a}\cdot\boldsymbol{x}+\boldsymbol{b}))$,SVM 优化问题的核函数选择高斯核函数 $K_{\mathrm{SVM}}(\boldsymbol{u},\boldsymbol{v})=\exp(-\|\boldsymbol{u}-\boldsymbol{v}\|^2/2\gamma^2)$.由于 SMO 的泛化性能受代价参数 C 和核参数 γ 影响较大,因此需要选择合适的参数值,以使算法的泛化性能最好.参照文献[11]中的参数选择方法,共有 225 组参数对,并选择性能最好的组合(C_{SMO},γ_{SMO}).15 个 C_{SMO} 值分别取 0.001、0.01、0.05、0.1、0.2、0.5、1、2、5、10、20、50、100、1 000 和 10 000,15 个 γ_{SMO} 值分别取 0.001、0.01、0.1、0.2、0.4、0.8、1、2、5、10、20、50、100、1 000和 10 000.SSMO 算法需要设定 2 个参数:隐节点数目 L 和代价参数 C_{SMO}.由文献[6]所述,当 L 为 10^3 数量级时,OMELM 的泛化性能趋向稳定,在此设 $L=2\,000$.同时,由于 SSMO 算法对 C_{SMO} 不太敏感,在此选择 8 个 C_{SMO} 值来测试 SSMO 算法的最佳泛化能力 0.001、0.01、0.1、1、10、100、1 000、10 000.SSMO 算法的容忍参数 τ 设为 10^{-5}.

表 1 标准数据集描述

数据集	特征数	训练样本数	测试样本数
Breast-cancer	10	300	383
Liver-disorders	6	200	145
Ionosphere	34	300	251
Pimadata	8	400	368
Pwlinear	10	100	100
Sonar	60	100	158
Monk's Problem 1	6	124	432
Splice	60	1 000	2 175
Australian	14	300	390
A7a	123	16 100	16 461
A8a	123	22 696	9 865
A9a	123	32 561	16 281

3.2 SSMO 算法与 SMO 算法性能比较

对每个数据集进行 20 次测试,每次测试随机产生指定的训练数据集和测试数据集.在此利用 20 次测试的平均训练时间、平均测试精度、相应的标准方差、非边界支持向量数(记作 A)、边界支持向量数(记作 B),作为算法的评价指标.表 2 和表 3 包含了上述指标的测试结果.

从表 2 和表 3 可以看出,在 12 个数据集的测试中,SSMO 算法在 10 个数据集上的平均测试精度较高,而 SMO 算法在其他 2 个数据集上的平均测试精度较高.相比 SMO 算法需要比较 225 对值才能

达到最优值,SSMO 算法只需在 8 个 C_{SSMO} 值范围内就可以找到最优值. 这个性质很重要,如果加上调节参数的时间,SSMO 算法的训练时间平均只需 SMO 算法的 1/30,这对于某些实时应用是相当有用的.

3.3 SSMO 算法与 SMO 算法稳定性比较

本节主要对两种算法在 20 次随机测试时的训练时间的稳定性进行比较. 在此采用 Pwlinear 和 Ionosphere数据集,两种算法的最优参数根据表 3 设置. 图 1 和图 2 分别是 Pwlinear 数据集和 Ionosphere 数据集的比较结果. 从图 1 和图 2 可以看出,SSMO 算法的训练时间在 2 个数据集上均比 SMO 算法稳定得多.

表 2　SSMO 算法与 SMO 算法的泛化性能比较

数据集	平均训练时间/s		平均测试精度/%		标准方差/%	
	SMO 算法	SSMO 算法	SMO 算法	SSMO 算法	SMO 算法	SSMO 算法
Breast-cancer	0.416 3	0.123 2	93.66	96.49	1.00	0.60
Liver-disorders	0.254 2	6.075 7	67.86	70.03	4.29	3.06
Ionosphere	0.072 8	0.047 9	90.60	89.44	1.48	1.49
Pimadata	6.435 0	0.654 4	76.88	77.45	1.64	1.83
Pwlinear	0.027 0	0.037 2	84.05	84.65	1.96	2.61
Sonar	0.306 9	0.414 5	83.75	81.76	3.05	0.66
Monk's Problem 1	0.213 5	9.223 6	94.30	94.69	2.01	1.65
Splice	12.781	2.850 8	82.27	83.63	0.78	0.82
Australian	0.678 0	0.249 1	67.82	67.92	1.32	2.18
A7a	2 857	3 052	83.65	83.72	1.89	2.05
A8a	4 135	4 658	84.03	84.21	1.56	1.75
A9a	10 568	11 258	83.41	83.45	1.43	1.34

表 3　SSMO 算法与 SMO 算法的最优参数和支持向量数比较

数据集	C_{SMO},γ_{SMO}	C_{SSMO}	A/个		B/个	
			SMO 算法	SSMO 算法	SMO 算法	SSMO 算法
Breast-cancer	(5,50)	10^{-3}	4	7	219	74
Liver-disorders	(10,2)	1	9	9	155	141
Ionosphere	(5,5)	10^{-2}	40	25	2	17
Pimadata	(10^3,50)	10^{-2}	9	9	222	259
Pwlinear	(10^4,10^3)	10^{-3}	5	4	56	53
Sonar	(20,1)	10^{-1}	73	22	0	44
Monk's Problem 1	(10,1)	10^4	67	43	1	2
Splice	(2,20)	10^{-3}	59	38	621	492
Australian	(2,2)	10^{-2}	25	62	173	157
A7a	(5,5)	10^{-2}	535	318	5 829	5 990
A8a	(5,5)	10^{-2}	781	532	8 233	8 307
A9a	(5,5)	10^{-2}	976	775	11 588	11 701

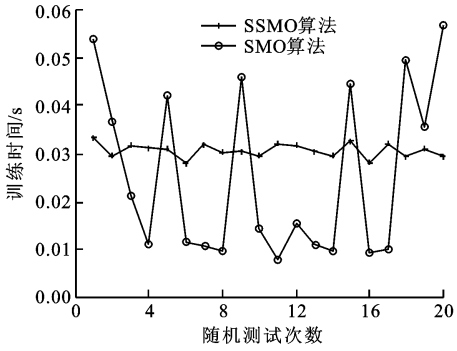

图1 Pwlinear数据集的比较结果

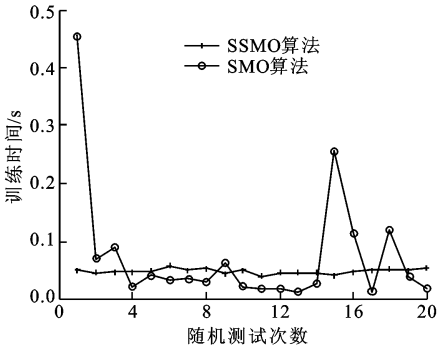

图2 Ionosphere数据集的比较结果

SMO算法的主要迭代过程是优化2个拉格朗日乘子α_1和α_2,在每次迭代完毕要重新计算偏置b.由文献[7]可知,当每次迭代优化完的2个乘子都在边界上时,偏置b是无法精确计算的.事实上,在一个可选的范围内,b的取值都是有效的,而且不违反KKT条件.由于在有些迭代过程中只能近似地估计b的值,这将导致SMO算法在重复运行中有较小的误差.如果估计出的b值接近最优值,那么该次运行所需的迭代次数就比较少,同时算法的训练时间就比较短.如果估计出的b值偏离最优值较大,那么该次运行就需要更多的迭代过程来修正这个错误,这样就需要相对较多的计算时间.

在SSMO算法中,由于不需要计算偏置b,很显然每次运行的迭代次数是相同的,根据式(17)和式(21)可以看出,每次迭代的计算代价是相同的,由此可见SSMO算法的训练时间是很稳定的.

4 结 论

本文在Huang[6]提出的OMELM优化问题的基础上,分析并给出了序列最小优化算法——SSMO算法.与SMO算法每次迭代需求解2个拉格朗日乘子不同,SSMO算法在每次迭代中只需求解一个拉格朗日乘子,并根据使ELM目标函数值下降最快的变量来选择所需优化的拉格朗日乘子.当所有拉格朗日乘子满足相应的KKT条件时,OMELM的目标优化问题得到最优解.

由于SSMO算法每次优化一个拉格朗日乘子,所以它在随机测试时的训练时间比SMO算法要稳定.SSMO算法在数据集最优参数C较小时(一般小于0.1)训练时间较短.但是,当在数据集最优参数C较大时,SSMO算法需要较多的迭代次数才能收敛到优化问题的最优解.将来的工作主要研究SSMO算法在数据集最优代价参数C较大的情况下如何快速迭代到最优解.

参考文献:

[1] HUANG Guangbin, ZHU Qinyu, SIEW C K. Extreme learning machine: theory and applications [J]. Neurocomputing, 2006, 70(1/2/3): 489-501.

[2] HUANG Guangbin, ZHU Qinyu, SIEW C K. Real-time learning capability of neural networks [J]. IEEE Transactions on Neural Network, 2006, 17 (2): 863-878.

[3] 程松, 闫建伟, 赵登福, 等. 短期负荷预测的集成改进极端学习机方法[J]. 西安交通大学学报, 2009, 43 (2): 106-110.
CHENG Song, YAN Jianwei, ZHAO Dengfu, et al. Short-term load forecasting method based on ensemble improved extreme learning machine [J]. Journal of Xi'an Jiaotong University, 2009, 43 (2): 106-110.

[4] 邓万宇, 郑庆华, 陈琳, 等. 神经网络极速学习方法研究[J]. 计算机学报, 2010, 33 (2): 279-287.
DENG Wanyu, ZHENG Qinghua, CHEN Lin, et al. Research on extreme learning of neural networks [J]. Chinese Journal of Computers, 2010, 33 (2): 279-287.

[5] BARTLETT P L. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network [J]. IEEE Transactions on Information Theory, 1998, 44 (2): 525-536.

[6] HUANG Guangbin, DING Xiaojian, ZHOU Hongming. Optimization method based extreme learning machine for classification [J]. Neurocomputing, 2010, 74(1/2/3): 155-163.

[7] PLATT J. Fast training of support vector machines using sequential minimal optimization [M]// Advances in Kernel Methods: Support Vector Learning. Cambridge, MA, USA: MIT Press, 1999: 185-208.

[8] FLETCHER R. Practical methods of optimization:

constrained optimization [M]. New York, USA: John Wiley and Sons, 1981: 2.

[9] BLAKE C L, MERZ C J. UCI repository of machine learning databases [EB/OL]. (1998-04-02)[2010-02-12]. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[10] MICHIE D, SPIEGELHALTER D J, TAYLOR C C. Machine learning, neural and statistical classification [M]. Englewood Cliffs, NJ, USA: Prentice Hall, 1994.

[11] GHAUTY P, PAUL S, PAL N R. NEUROSVM: an architecture to reduce the effect of the choice of kernel on the performance of SVM [J]. Journal of Machine Learning Research, 2009, 10(3): 591-622.

（编辑　刘杨　武红江）