

用于RBF-SVM参数搜索的伪梯度动态步长算法

张宏达, 王晓丹, 徐海龙

(三原25信箱 陕西 三原 713800)

【摘要】针对基于径向基核函数(RBF)的支持向量机(SVM)超参数选择问题,提出了一种基于差分方程的新算法——伪梯度动态步长算法。该算法根据径向基核函数的特点提出由训练集的空间特性确定的核参数搜索范围,并采用对数刻度表示搜索空间;利用参数空间中SVM在两个临近点的分类精度的变化估计参数的搜索方向,并且随着搜索方向的变化动态调整搜索步长,从而实现较快的搜索。通过与Grid和PSO方法的对比实验,表明该算法具有良好的性能。

关键词 动态步长; 梯度; 超参数; 支持向量机

中图分类号 TP181

文献标识码 A

doi:10.3969/j.issn.1001-0548.2010.04.010

Pseudo Gradient and Dynamic Step Optimization Algorithm for RBF-SVM Parameter Search

ZHANG Hong-da, WANG Xiao-dan, and XU Hai-long

(Mailbox No.25, Sanyuan Sanyuan Shanxi 713800)

Abstract To the issue of hyper-parameter selection for radial basis function (RBF) based support vector machines (SVM), a new algorithm named as pseudo gradient and dynamic step optimization is proposed. Based on the characteristics of RBF, the kernel parameter is pre-estimated according to the distribution of the train set and the logarithmic scale is employed for the parameter space. The search direction is estimated with the changing of classification accuracy and by tuning the search step accordingly. At last, comparative experiments with Grid approach and PSO algorithm indicate the validity of the proposed algorithm.

Key words dynamic step; gradient; hyper-parameter; support vector machines

选择合理的参数是支持向量机(support vector machine, SVM)发挥最佳泛化性能的前提。目前SVM参数优化方法可归结为两类:(1) 根据影响分类器性能的各种因素,构造优化目标函数,基于梯度方法通过迭代方法得出最优超参数^[1-3]; (2) 在超参数空间中用各种搜索方法,按照测试误差从搜索空间中进行参数优选,如网格搜索法及其改进算法^[4]以及各种启发式搜索算法^[5-6]。第一类方法的优点在于不用训练SVM本身,且速度较快等,但是该类方法容易陷入局部最优解,特别是当迭代初始值选择不当,对最终结果影响更大^[7]。第二类方法中,网格法面临的困难是网格密度难以设置;采用各种智能算法能够得到较优的参数组合,但进行SVM参数优化面临运算量过大的问题。本文针对基于径向基核函数(radial basis function, RBF)的SVM,提出了基于差分方程的伪梯度动态步长算法。与以往方法相比,该算法的特点是:(1) 算法根据估计的梯度方向确定

参数的搜索方向,可理解性更强;(2) 动态调整搜索步长,搜索效率更高。通过对UCI数据集的实验验证了该算法的有效性。

1 SVM

设训练样本集为 $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, $\mathbf{x}_i \in R^d$ 为训练样本, $y_i \in \{1, -1\}$ 为输入样本 \mathbf{x}_i 的类别,标准支持向量机(C-SVM)的优化形式为:

$$\begin{cases} \min J(\mathbf{w}, b, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i \\ \text{s.t. } y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, 2, \dots, l \end{cases} \quad (1)$$

式中 $\mathbf{w} \in R^d$ 为分类超平面的法向量; $\phi(\mathbf{x}_i)$ 是由核函数 $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ 隐式定义的映射, RBF核函数形如:

收稿日期: 2009-01-04; 修回日期: 2009-09-30

基金项目: 国家自然科学基金(60975026); 陕西省自然科学基金(2007F19)

作者简介: 张宏达(1981-), 男, 博士生, 主要从事智能信息处理、机器学习方面的研究。

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \quad (2)$$

通过Lagrange方法, 可得与式(1)等价的对偶形式:

$$\begin{cases} \max_{\alpha} L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \sum_{i=1}^l y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C, \quad i=1,2,L,l \end{cases} \quad (3)$$

用式(4)判别未知样本 \mathbf{x} :

$$\varphi(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right) \quad (4)$$

显然, 当训练样本确定, $\varphi(\mathbf{x})$ 由超参数确定, 对于RBF-SVM, 可记 $\varphi(\cdot) = \text{svm}(\sigma, C)$ 。

2 参数的搜索空间和空间刻度

目前, 对于SVM参数搜索范围通常采用按经验指定的方法。实际上, 对于任何搜索算法, 设置合理的搜索范围, 是保证算法快速得到最优参数组合的重要条件。若范围太大, 会增加不必要的寻优操作; 如果范围太小, 则可能不包括最优参数组合。

RBF本质上是基于欧式距离的核函数, 所以 σ 的选择应当参考样本间的距离, 考虑所有训练样本, 令 $d_i^{\min} = \min_{j,j \neq i} d_{ij}$ 和 $d_i^{\max} = \max_{j,j \neq i} d_{ij}$ 分别表示每个样本的最近样本距和最远样本距, 且 $d_{\min} = \frac{1}{l} \sum_{i=1}^l d_i^{\min}$ 和 $d_{\max} = \frac{1}{l} \sum_{i=1}^l d_i^{\max}$ 分别表示训练样本的平均最近距离和平均最远距离, $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ 为欧式距离, 核参数的合理区间应满足 $k_1 d_{\min} = \sigma_{\text{lb}} < \sigma < \sigma_{\text{ub}} = k_2 d_{\max}$ 。对于支持向量 \mathbf{x}_i 与距其最近的样本 \mathbf{x}_j 和距其最远的样本 \mathbf{x}_k , 当取 $k_1 = 0.293$, 有 $K(\mathbf{x}_i, \mathbf{x}_j)$ 约为0.003(概率论中常将出现概率小于0.003的事件称作小概率事件, 此处借用该概念, 认为小于0.003的数字是一个很小的数字。), 说明此时支持向量难以覆盖^[9]距其最近的样本, 此时容易造成过适应; 当取 $k_2 = 13$ 时, $K(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}_k) \approx 0.003$, 说明支持向量对距离其最近和最远样本的影响几乎没有差别, 此时容易造成欠适应。因此:

$$0.293 d_{\min} = \sigma_{\text{lb}} < \sigma < \sigma_{\text{ub}} = 13 d_{\max} \quad (5)$$

应当是一个较为合理的参数搜索空间。

SVM性能受超参数的影响有如下特点: 在最优参数附近, 对于同样大小的参数变化, 若最优参数

值较大, 目标函数受到的影响较小, 而若最优参数为一个较小的值, 目标函数将受到很大影响。为综合考虑算法的开拓能力和开掘能力^[4], 算法采用对数刻度。为便于区分, 记线性刻度下的参数为 σ^0 、 C^0 , 对数刻度下的参数为 σ 、 C 。根据经验设置参数 C : $C^0 \in [C_{\text{lb}}^0, C_{\text{ub}}^0] = [1, 5000]$, 根据式(5)得出 $\sigma^0 \in [\sigma_{\text{lb}}^0, \sigma_{\text{ub}}^0]$, 采用对数刻度, 参数空间变为:

$$\begin{aligned} \sigma &\in [\sigma_{\text{lb}}, \sigma_{\text{ub}}] = [\lg(\sigma_{\text{lb}}^0), \lg(\sigma_{\text{ub}}^0)] \\ C &\in [C_{\text{lb}}, C_{\text{ub}}] = [\lg(C_{\text{lb}}^0), \lg(C_{\text{ub}}^0)] \end{aligned} \quad (6)$$

本文将这种依据数据估计的参数搜索空间和采用对数刻度的方式简称为参数预估和对数刻度(pre-estimate parameter space and logarithmic-scale, PAL)。

3 伪梯度动态步长参数搜索算法

3.1 算法思想

由图1可知, 在RBF-SVM参数空间中, 使得SVM具有较好泛化性能的参数集中在“好区”中虚线附近, 分布在实曲线外围区域的参数组合会造成SVM欠适应或者过适应($C_{\text{lim}}^{\text{[10]}}$ 是采用线性SVM时得出的最优参数 C)。理想的参数选择应当仅限于在“好区”中进行, 但是, 所谓“好区”是随着不同的分类问题或者分类数据变化的, 因此对于特定问题, 无法提前确定其“好区”范围。各种搜索算法用于在提前确定的参数空间中搜索较优参数组合。

RBF-SVM的参数寻优问题可以看作是一个以超参数为自变量, 以对应的SVM泛化精度为因变量的函数 $f(\sigma, C)$ 。根据 (σ, C) , 由式(2)~式(4)得 $\varphi(\cdot) = \text{svm}(\sigma, C)$, 考虑分离测试情形, 给定测试集:

$$D_{\text{st}} = \{\mathbf{x}_i, y_i\}_{i=1}^n, \quad \text{记 } I(\cdot) = \begin{cases} 1 & \text{“.” 为真} \\ 0 & \text{否则} \end{cases}$$

有:

$$f(\sigma, C) = \frac{1}{n} \sum_{i=1}^n I(\varphi(\mathbf{x}_i) \neq y_i) \quad (7)$$

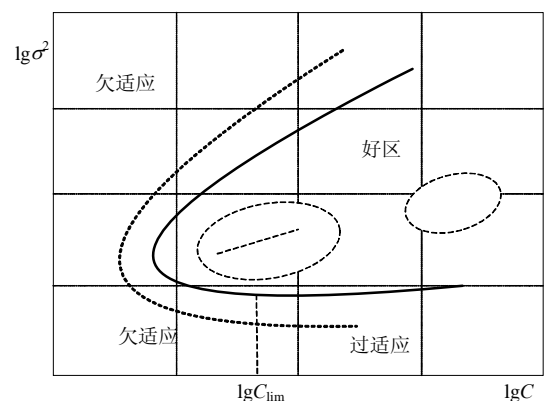


图1 RBF-SVM超参数空间示意图

根据文献[10], 函数 $f(\sigma, C)$ 与超参数的关系如图1所示。若 $f(\sigma, C)$ 在该区间中是连续的(实际应用中, 因验证或测试样本数总是有限的, 最终得到的“曲面” $f(\sigma, C)$ 实际上是阶梯形的。), 采用梯度方法寻优是可行的。由于SVM分类精度是由超参数与训练集和(或)测试集同时决定的, 无法直接将SVM泛化精度表示为超参数的显函数, 因此难以直接通过求导的方式确定函数的梯度。

根据导数的定义, 任意点 (σ, C) 处的梯度可表示为 $\text{grad}f(\sigma, C) = \frac{\partial f}{\partial \sigma} \mathbf{i} + \frac{\partial f}{\partial C} \mathbf{j}$ 。近似计算每一点的梯

度时, 利用差分代替微分有 $\frac{\partial f}{\partial \sigma} \approx \frac{f(\sigma + \Delta\sigma, C) - f(\sigma, C)}{\Delta\sigma}$, $\frac{\partial f}{\partial C} \approx \frac{f(\sigma, C + \Delta C) - f(\sigma, C)}{\Delta C}$, 采用较小的 $(\Delta\sigma, \Delta C)$ 近似

计算偏导数, 理论上是可行的。如果计算出的梯度是有效的, 那么根据梯度方向, 可以很快收敛到(局部)最优。但实际上存在两个问题: (1) $f(\sigma, C)$ 曲面是阶梯形的, 因此在点 (σ, C) 处的增量 $(\Delta\sigma, \Delta C)$ 较小时, 得到的梯度常为0, 对确定算法的搜索方向没有意义。(2) 搜索每前进一步, 若搜索步长为 σ_{stp} 、 C_{stp} , 需要计算3个点 $(\sigma + \Delta\sigma, C)$ 、 $(\sigma, C + \Delta C)$ 、 $(\sigma + \sigma_{\text{stp}}, C + C_{\text{stp}})$ 处的函数值, 对于样本规模为1、样本维数为 d 、交叉验证折数为 k 的SVM超参数搜索问题, 进行一次梯度估计SVM训练的运算规模为 $3dl^2(k-1)^2 k^{1-\lambda}$, 采用分离验证时, SVM训练的运算规模为 $3dl^2$, 通常有 $\lambda \geq 2$ 。因此采用该方式计算代价太高。为此, 本文采用如下方法: 每次仅选择一个参数分量进行更新, 并且根据参数当前取值和上一取值时函数的差分估计该点处的梯度方向, 并按照估计的梯度方向进行参数更新, 每一次梯度方向估计和参数更新只需要计算一次 $f(\sigma, C)$ 。本文称该梯度方向估计方法为伪梯度方向法。

若算法初期搜索步长过小, 容易造成搜索范围小, 收敛速度过慢, 若搜索后期搜索步长过大, 容易造成震荡, 难以收敛到最优解。为保证搜索算法的效率, 减少无效计算, 采用较大的初始步长, 并随着梯度方向的变化减小搜索步长, 兼顾算法充分搜索参数空间的能力以及快速收敛到最优解的能力。

采用该搜索方法得出的解受搜索起始点影响较大, 一般为局部最优解, 为了得到更好的参数组合, 采用多个位于不同起点的个体同时进行, 选其中最

优解作为最终SVM参数。

采用伪梯度方向法和动态变化的搜索步长, 算法记为伪梯度动态步长算法(seudo gradient and dynamic step optimization, SGADO)。

3.2 算法实现

为便于理解, 首先给出如下标记及说明:

(1) $D_{\text{tm}} = \{\mathbf{x}_i, y_i\}_{i=1}^l$ 、 $D_{\text{st}} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ 分别为训练集和测试集。

(2) np 表示群体规模, n_{iter} 为算法迭代次数。

(3) r_{sig} 表示算法选择 σ 进行更新的概率, 由于相比参数 C , 分类器性能对参数 σ 的变化更为敏感, 因此可选择 $0.5 \leq r_{\text{sig}} \leq 1$ 。

(4) $0 < r_{\text{thrs}}, r_{\text{thrc}} < 1$ 为参数 σ 和 C 的步长缩小系数, 该系数若较小, 将影响算法的寻优速度, 该系数过大易造成震荡, 难以收敛到最优解, 同时由于选择 σ 和 C 进行更新的概率不同, 为确保搜索步长在两个方向上的相对一致性, 因此 r_{thrs} 、 r_{thrc} 应满足:

$$r_{\text{thrc}} = r_{\text{thrs}}^{(r_{\text{sig}}/(1-r_{\text{sig}}))} \quad (8)$$

(5) σ_{stp} 、 C_{stp} 为初始搜索步长, 由下式确定:

$$\sigma_{\text{stp}} = r_{\text{sig}}(\sigma_{\text{ub}} - \sigma_{\text{lb}}), C_{\text{stp}} = r_{\text{c}}(C_{\text{ub}} - C_{\text{lb}}) \quad (9)$$

r_{sig} 、 r_{c} 称作初始步长因子。

(6) $\sigma_{\text{dr}}, C_{\text{dr}} \in \{-1, 1\}$ 表示参数搜索方向。

用于SVM参数优化的伪梯度动态步长算法如下:

输入: $D_{\text{tm}}, D_{\text{st}}, n_{\text{iter}}, np, r_{\text{sig}}, r_{\text{thrs}}, r_{\text{thrc}}, r_{\text{c}}$ 。

(1) 按照式(5)和式(6)计算 $[\sigma_{\text{lb}}, \sigma_{\text{ub}}], [C_{\text{lb}}, C_{\text{ub}}]$, 由式(9)得 $\{\sigma_{\text{stp}}(j), C_{\text{stp}}(j)\}_{j=1}^{np}$, 随机生成初始化参数 $\{\sigma_0(j), C_0(j)\}_{j=1}^{np}$ 及搜索方向 $\{\sigma_{\text{dr}}(j), C_{\text{dr}}(j)\}_{j=1}^{np}$, 并由式(7)计算 $f(\sigma_0(j), C_0(j))$, 且令 $\text{tbest}(j) = \text{lbest}(j) = f(\sigma_0(j), C_0(j))$, 令 $i=1, j=1$ 。

(2) 进行参数位置、搜索方向和搜索步长更新

if $\text{rand}() < r_{\text{sig}}$

$\Delta\sigma = \sigma_{\text{dr}}(j)\sigma_{\text{stp}}(j)(1 + \text{rand}())/2; \Delta C = 0;$

else

$\Delta C = C_{\text{dr}}(j)C_{\text{stp}}(j)(1 + \text{rand}())/2; \Delta\sigma = 0;$

$\sigma_i(j) = \sigma_{i-1}(j) + \Delta\sigma; C_i = C_{i-1}(j) + \Delta C;$

if $(f(\sigma_i(j), C_i(j)) - f(\sigma_{i-1}(j), C_{i-1}(j))) > 0$

{

if $(\Delta\sigma > 0)$

$\sigma_{\text{dr}}(j) = -\sigma_{\text{dr}}(j); \sigma_{\text{stp}}(j) = \sigma_{\text{stp}}(j)r_{\text{thrs}};$

else $C_{\text{dr}}(j) = -C_{\text{dr}}(j); C_{\text{stp}}(j) = C_{\text{stp}}(j)r_{\text{thrc}};$

}

(3) if $i \leq n_{\text{iter}}, i = i + 1$, 否则输出。

(4) if $j < np$, $j = j + 1$, 回到(2), 否则回到(3).
输出: $(\sigma, C) = \arg \min_{i,j} f(\sigma_i(j), C_i(j))$.

考虑采用分离测试方法计算 $f(\sigma, C)$, 表1用伪代码描述了算法详细流程。采用上述方法, 算法会快速收敛至局部最优解, 为避免无效搜索, 设定无进展代数阈值 θ_{npj} , 若搜索进行 θ_{npj} 代临时局部最优无进展, 则记录局部最优, 并按照:

$$\begin{cases} \sigma_i(j) = \sigma_{\text{lb}} + \text{rand}()(\sigma_{\text{ub}} - \sigma_{\text{lb}}) \\ C_i(j) = C_{\text{lb}} + \text{rand}()(C_{\text{ub}} - C_{\text{lb}}) \end{cases} \quad (10)$$

重新起始个体, 重新计数无进展代数, 并且按式(9)将搜索步长 $\sigma_{\text{stp}}(j)$ 、 $C_{\text{stp}}(j)$ 变为初始值。本文局部最优指个体搜索到的最优解, 临时局部最优值是指个体重新起始之后搜索到的最优解。

对于表1中的数据集2, 采用SGADO算法, 取 $np = 6$, $n_{\text{iter}} = 20$, $\theta_{\text{npj}} = 8$, $r_{\text{thrs}} = 0.62$, $r_{\text{thrc}} = 0.49$, $r_{\text{Ssig}} = 0.6$, $r_{\text{sig}} = r_c = 0.15$, 算法的搜索路径如图2所示。

表1 实验数据设置

数据集	类别处理		数据集规模	
	positive	negative	Train set	Test set
heart disease	label=0	other	150	153
hepatitis	label=1	other	100	54
balance	label=1	other	200	327

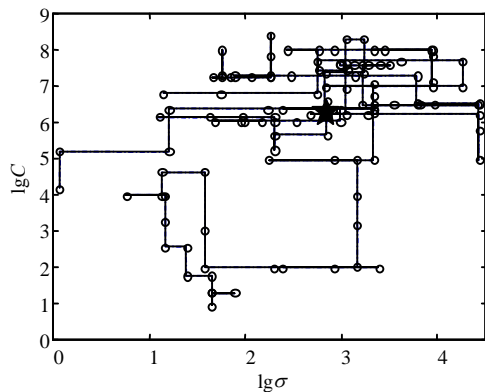


图2 SGADO算法的搜索路径

图2中, 黑“五角星”表示算法搜索到的最优点, 该点附近及其右上角圆点的密度较大, 6条搜索路径有4条最终收敛到该区域, 说明该搜索算法向该“好区”搜索的能力。

4 实验与分析

为验证算法的有效性, 采用经典的Grid算法和目前热门的PSO算法作为对比算法。实验中采用的3

个UCI数据设置如表2所示。实验机器配置为512 MB内存, 1.80 GHz CPU, 算法用Matlab 7.0编程实现, 其中SVM采用Steve Gunn的工具箱。

4.1 参数预估与对数刻度的合理性实验

为验证提出的核参数范围和对数刻度的合理性, 分别对Grid和PSO算法在3种不同设置下进行对比实验。为方便说明, 引入如下标记:

Grid1采用经验设置 $\sigma \in [0.1, 100]$, $C \in [1, 5000]$ 及线性刻度进行搜索, 采用 20×20 网格;

Grid2采用参数预估和线性刻度, 20×20 网格;

PALGrid采用PAL, 20×20 网格;

PSO1采用线性刻度和经验搜索区间的PSO;

PSO2采用线性刻度和参数预估的PSO;

PALPSO采用参数预估和对数刻度的PSO。

PSO算法的种群规模和迭代次数均为20; 为确保3种PSO算法的稳定性和收敛速度, 采用文献[8]的系数设置: $w=0.729$, $c_1=c_2=1.49445$ 。

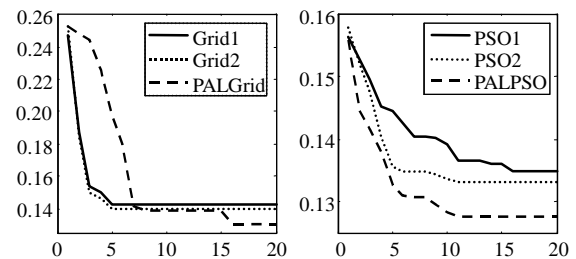


图3 不同设置下Grid和PSO算法最优值收敛曲线

对于Grid算法, 实际上并不存在进化曲线, 为便于比较, 将算法每计算过一行网格节点记为一代。由于确定了算法设置之后, 所有网格节点都确定了, 因此3种Grid算法仅对3个UCI数据集各进行一次实验。采用PSO1、PSO2、PALPSO 3种PSO算法, 对3个UCI数据集分别进行10次实验, 即对每种算法进行30次实验。实验结果如图3所示。

图3中横坐标表示进化代数, 纵坐标表示SVM最低分类误差。比较各种方法搜索到的最优值, PALGrid和Grid2比Grid1算法最终分类误差分别降低8.65%、2.14%; PALGrid算法比Grid算法降低6.65%; PALPSO和PSO2分别比PSO1降低5.21%、1.14%; PALPSO比PSO2降低4.11%。比较数据说明, 采用参数预估和对数刻度的方法能有效提高性能, 即给出的参数预估与对数刻度法是合理的。

4.2 算法SGADO有效性实验

本文实验已经验证了参数预估与对数刻度的合理性, 后面实验中各种算法均在参数预估和对数刻

度设置下进行。

下面通过与PALGrid和PALPSO算法的性能进行比较,验证SGADO的有效性。为确保对比实验的公平性,SGADO算法中设置 $np = 20$, $n_{iter} = 20$, 其余参数设置为: $r_{sig} = 0.6$, $n_{iter} = 20$, $\theta_{npg} = 8$, $r_{thrs} = 0.62$, $r_{thrc} = 0.49$, $r_{sig} = r_C = 0.15$ 。对每个数据集进行10次实验并取平均值,3种算法在不同数据集上的性能如图4所示,横坐标为进化代数,纵坐标为算法得到的最优参数对应的分类误差值。

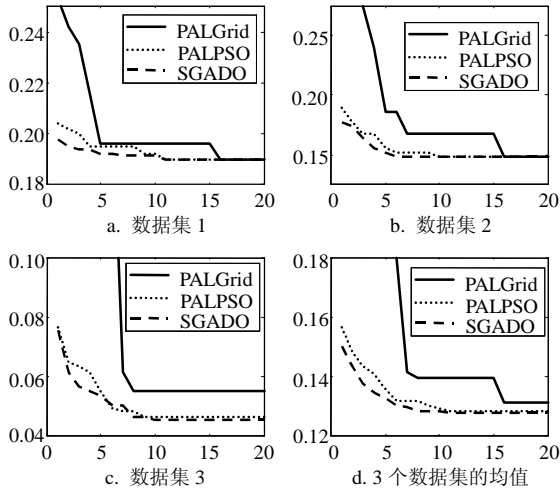


图4 SGADO与Grid和PSO方法的性能对比

从图4的平均值看,SGADO和PALPSO算法均优于PALGrid,说明了启发式算法的优势;从各个数据集看,3种算法在数据集1和2上都收敛到最优值;从最终分类精度看,SGADO在数据集3上优于PALPSO。

由于SVM参数搜索问题的复杂性,算法的早期收敛能力是一项重要指标,表2给出了3种算法的早期寻优能力的对比。

表2 3种搜索算法的早期寻优能力对比

进化代数	PALPSO-vs-PALGrid	SGADO-vs-PALGrid	SGADO-vs-PALPSO
5	31.56	32.93	2.01
10	7.41	8.02	0.65
15	8.19	8.33	0.16
20	2.32	2.48	0.16

以表3第一行第一列数据为例, PALPSO-vs-PALGrid表示PALPSO和PALGrid算法运行到第5代时, PSO得出的最优分类误差比PALGrid低31.56%。

对比实验显示,SGADO不仅最终分类精度最高,而且早期寻优能力最优,意味着在同样的精度要求下,SGADO搜索参数速度更快。

5 结 论

根据RBF核的特点,讨论了核参数的取值范围并采用对数刻度表示搜索空间。将RBF-SVM分类误差看作超参数的函数,由函数差分估计搜索方向并且动态变化搜索步长,提出伪梯度动态步长(SGADO)算法。对比实验结果显示,提出的算法从最终寻优结果和寻优速度均优于Grid和PSO算法,是一种有效的参数搜索算法。算法的其他优势在于,不同于基于种群的智能算法,如遗传算法和PSO算法。该算法的搜索方向不依赖于其他个体,可以采用较小的群体或者采用单个个体搜索的方法,因此更加有利于采用并行方式进行。根据各种机器学习问题的共性,即算法的连续可调参数和泛化性能之间隐藏的函数关系,以及该类函数体现的连续性,SGADO算法可以推广应用于多种机器学习的参数优化问题。

参 考 文 献

- [1] DONG Y L, XIA Z H, XIA Z Q. A two-level approach to choose the cost parameter in support vector machines[J]. Expert Systems with Applications, 2008, 34: 1366-1370.
- [2] AYAT N E, CHERIET M, SUEN C Y. Automatic model selection for the optimization of SVM kernels[J]. Pattern Recognition, 2005,38: 1733-1745.
- [3] 刘向东, 骆 斌, 陈兆乾. 支持向量机最优模型选择的研究[J]. 计算机研究与发展, 2005, 42(4): 576-581. LIU Xiang-doug, LUO Bing, CHEN Zhao-qun. Optimal model selection for support vector machines[J]. Journal of Computer Research and Development, 2005, 42(4): 576-581.
- [4] 朱家元, 杨 云, 张恒喜, 等. 支持向量机的多层动态自适应参数优化[J]. 控制与决策, 2004, 19(2): 223-225. ZHU Jia-yuan, YANG Yun, ZHANG Hen-xi, et al. Multi-layer adaptive parameters optimization approach for support vector machine[J]. Control and Decision, 2004, 19(2): 223-225.
- [5] 郑春红, 焦李成, 丁爱玲. 基于启发式遗传算法的SVM模型自动选择[J]. 控制理论与应用, 2006, 23(2): 187-192. ZHENG Chun-hong, JIAO Li-cheng, DING Ai-ling. Automatic model selection for support vector machines using heuristic genetic algorithm[J]. Control Theory and Applications, 2006, 23(2): 187-192.
- [6] 任江涛, 赵少东, 许盛灿, 等. 基于二进制PSO算法的特征选择及SVM参数同步优化[J]. 计算机科学, 2007, 34(6): 179-182. REN Jiang-tao, ZHAO Shao-dong, XU Sen-chan, et al. Simultaneous feature selection and SVM parameters optimization algorithm based on binary PSO[J]. Computer Science, 2007, 34(6): 179-182.