第 28 卷第 6 期
2011 年 6 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 28 No. 6
Jun. 2011

文章编号: 1000−8152(2011)06−0793−06

支持向量机的进化多核设计

李仁兵[1,2], 李艾华[1], 白向峰[1], 蔡艳平[1], 王德生[3]

(1. 第二炮兵工程学院 502教研室, 陕西 西安 710025; 2. 中国空气动力研究与发展中心, 四川 绵阳 621000;

3. 第二炮兵青州士官学校 204教研室, 山东 青州 262500)

摘要: 为提高支持向量机分类精度, 提出一种基于遗传程序设计的进化多核算法. 算法中每个个体表示一个多核函数, 并采用树形结构进行编码, 增强了多核函数的非线性; 初始种群由生长法产生, 经过遗传操作后得到适合具体问题的进化多核函数. 遗传程序设计的全局搜索性能使得算法设计不需要先验知识. 与单核函数及其他多核函数的对比实验结果表明, 进化多核有效提高了支持向量机分类性能.

关键词: 进化多核; 遗传程序设计; 支持向量机; 核函数

中图分类号: TP181　　**文献标识码**: A

Evolutionary multiple kernels design for support vector machines

LI Ren-bing[1,2], LI Ai-hua[1], BAI Xiang-feng[1], CAI Yan-ping[1], WANG De-sheng[3]

(1. No. 502 Faculty, the Second Artillery Engineering College, Xi'an Shaanxi 710025, China;

2. China Aerodynamics Research and Development Center, Mianyang Sichuan 621000, China;

3. No. 204 Faculty, the Second Artillery Petty Officer School, Qingzhou Shandong 262500, China)

Abstract: To boost the classification accuracy of support-vector-machines(SVM), we propose an algorithm with evolutionary multiple kernels(EMK), based on the genetic programming(GP). In this algorithm, each individual represents a multiple kernel function, and is encoded by the tree-structure for enhancing the non-linearity of the multiple kernel function. Grow method is applied to initialize the GP population, from which the EMK adapting to practical problems is obtained by genetic operations. No priori knowledge is required due to the global search of GP. Comparisons of experimental results of EMK with the single kernel function and other multiple kernel functions show that EMK effectively improves the classification performance of SVM.

Key words: evolutionary multiple kernels; genetic programming; support vector machines; kernel function

1 引言(Introduction)

支持向量机是一种建立在核函数基础上的机器学习算法[1]. 随着支持向量机应用领域的扩展, 基于单一核函数的支持向量机分类算法很难满足复杂分类问题的需要, 尤其对于多源异构数据分类问题, 单核算法更是显得力不从心[2]. 为提高支持向量机泛化能力, 多核函数开始引起广泛关注[3]. 与单核函数相比, 多核函数具有3个优点[4,5]: 一是多核函数对数据特征的描述能力更强, 不同核函数对应不同的特征空间和非线性映射, 多核函数面对的是一组函数集而非单个函数, 因此能够更全面地描述数据特征; 二是多核函数具有更强的推广能力. 多核函数是单核函数的线性组合或复杂组合, 其自由参数可以通过经验获取, 也可以通过学习机器自动调整获得, 单核函数可以看成多核函数的退化形式. 因此, 多核函数比单核函数具有更强的鲁棒性; 三是增强了决策函数的可解释性.

目前, 关于多核函数的研究成果主要有: Lanckriet等[5]基于核矩阵对称半正定的事实, 提出利用半定规划(semidefinite programming, SDP)技术进行核矩阵处理, 并将混合核的组合参数优化问题转化为求解一个二次限制二次规划(quadratically-constrained quadratic program, QCQP)凸最优化问题; Bach等[6]延续了Lanckriet等的工作, 提出了QCQP的一种新的对偶表达式, 并利用传统SMO技术进行了求解; Sonnenburg等[7,8]在Lanckriet等基础上将二分类多核学习问题描述为半无限线性规划(semi-infinite linear program, SILP), 并用现有的线性规划求解器和标准SVM算法进行了求解. 此外还有线性组合核[9,10]、多加性回归核[11]、SimpleMKL方法[12]等等.

这些方法都是人工选择现有核函数进行组合,

收稿日期: 2010−08−29; 收修改稿日期: 2010−11−11.

并采取相应算法对组合参数进行优化, 虽然在一定程度上提高了算法性能, 但仍存在以下不足: 一是组合核中单核的选择过分依赖先验知识. 由于不同核函数对应不同映射函数和特征空间, 针对具体问题选择合适的单核显得非常重要, 这就需要掌握足够的先验知识; 二是上述组合核非线性表达能力不强. 对于复杂分类问题, 往往需要更为复杂的非线性组合核, 而上述组合核只是对单核进行简单组合, 无法有效提高算法在复杂分类问题上的性能. 为此, 本文提出一种基于遗传程序设计(genetic programming, GP)的进化多核(evolutionary multiple kernels, EMK)设计方法, 用于构造更为有效的核函数.

2 支持向量机(Support vector machine)

支持向量机是从线性可分情况下的最优超平面发展而来的[13]. 对于两类问题, 设样本集为$(\boldsymbol{x}_i, y_i)$, 其中$\boldsymbol{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$, 学习的目的就是寻找最优分类面$(\boldsymbol{w} \cdot \boldsymbol{x}) + b = 0$, 使得它不仅能将两类样本正确分开, 并且分类间隔最大. 支持向量机学习问题可表示为

$$\begin{cases} \min \dfrac{1}{2}||\boldsymbol{w}||^2 + C\sum\limits_{i=1}^{m}\varepsilon_i, \\ \text{s.t. } y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b) \geqslant 1-\varepsilon_i, \\ \quad \varepsilon_i \geqslant 0,\ i = 1, \cdots, m, \end{cases} \tag{1}$$

其中: ε_i为松弛项, $\varepsilon_i = 0$表示线性可分情况, $\varepsilon_i > 0$表示线性不可分情况下允许一定的错分; 惩罚因子C用于控制错分程度. 利用Lagrange乘数法可以把上式变成其对偶形式

$$\begin{cases} \max \sum\limits_{i=1}^{m}\alpha_i - \dfrac{1}{2}\sum\limits_{i,j=1}^{m}\alpha_i y_i \alpha_j y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j), \\ \text{s.t. } \sum\limits_{i=1}^{m}\alpha_i y_i = 0,\ \alpha_i \in [0, C], \\ \quad i = 1, \cdots, m. \end{cases} \tag{2}$$

这是典型的二次规划问题, 已有算法求解. 结果中少部分不为零的α_i对应的样本即为支持向量.

对于非线性分类问题, 可以通过非线性映射函数$\phi : \mathbb{R}^n \to H$将数据从原空间映射到高维线性特征空间$H$中, 然后在$H$中寻找最优分类面. 虽然特征空间$H$的维数极高, 但支持向量机算法巧妙地运用满足Mercer条件[14]的核函数代替内积运算, 即$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j))$, 从而不必明确知道函数$\phi(\boldsymbol{x})$的表达式, 有效避免了“维数灾难”问题. 常用的核函数有多项式(polynomial)核函数、径向基(radial basis function, RBF)核函数、多层感知器(sigmoid)核函数[13]等, 如表1所示. 求得支持向量后得到相应的支持向量机最优分类函数为

$$f(\boldsymbol{x}) = \operatorname{sgn}(\sum_{sv}\alpha_i y_i K(\boldsymbol{x}_i \cdot \boldsymbol{x}) + b). \tag{3}$$

表 1 常用核函数表达式

Table 1 Expressions of major kernel functions

核函数名	核函数表达式
多项式核函数	$K_{\mathrm{Pol}}(\boldsymbol{x}, \boldsymbol{y}) = ((\boldsymbol{x}\cdot\boldsymbol{y}) + c)^d,\ c \geqslant 0$
径向基核函数	$K_{\mathrm{RBF}}(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\sigma\|\boldsymbol{x}-\boldsymbol{y}\|^2),$ $\sigma > 0$
多层感知器核函数	$K_{\mathrm{Sig}}(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\kappa(\boldsymbol{x}\cdot\boldsymbol{y}) + \upsilon),$ $\kappa > 0,\ \upsilon < 0$

3 基于GP的EMK设计(EMK design based on GP)

3.1 算法基本模型(Basic model of the algorithm)

GP是在遗传算法基础上发展起来的一种新的进化计算方法[15], 其基本思想是: 用树的分层结构表示解空间, 每个树结构对应于问题解空间中的一个计算机程序; 通过选择、交叉和变异等遗传操作动态地改变这些树结构, 并且一代一代地演化下去, 直到找到适合于问题求解的计算机程序.

在本文中每个树结构对应于一个多核函数, 基于GP的支持向量机EMK基本模型如图1所示.

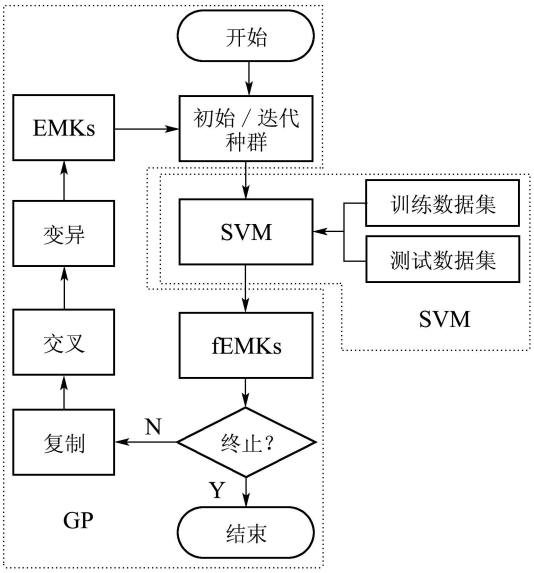

图 1 支持向量机进化多核模型

Fig. 1 Model of evolutionary multiple kernels for SVM

该模型由GP和SVM两部分组成, GP部分首先按照产生规则生成由指定数量个体组成的初始种群, 每个个体是一棵对组合核进行编码的算法树. 然后由SVM根据训练数据集和测试数据集进行训练和测试, 产生携带适应度(即每个个体对应的测试精度)的新种群fEMKs. 判断终止条件是否满足, 若满足, 则结束; 否则对fEMKs进行遗传操作, 产生下一代种群EMKs. 如此循环直到设定的终止条件满足为止.

3.2 个体的算法树表示(Expression tree of the individual)

为说明EMK的算法树表示, 首先介绍核函数构造方法.

核函数的构造主要有两种途径: 第1种方法是根据具体问题, 直接构造满足核函数定义的新函数. 这种方法比较难, 一般很少使用; 第2种方法是选择一些比较常用的单核函数进行合理组合, 从而构成新的核函数, 并有如下定理[14].

定理 1　设K_1和K_2是$X \times X$上的核, $X \in \mathbb{R}^n$. 设常数$\alpha \geqslant 0$, 则下面的函数均是核:

1) $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + K_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$;

2) $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \alpha K_1(\boldsymbol{x}_i, \boldsymbol{x}_j)$;

3) $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K_1(\boldsymbol{x}_i, \boldsymbol{x}_j) K_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$;

4) $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(K_1(\boldsymbol{x}_i, \boldsymbol{x}_j))$.

定理1的证明可参考文献[14], 此处略去. 由定理1可知, 核函数对$\{+, \times, \exp\}$3种运算封闭.

基于GP的EMK设计中, 个体是一棵对进化多核数学表达式进行编码的算法树, 由根节点、中间结点和叶节点构成. 个体的生成就是在给定的函数集(function set, FS)和终端集(terminal set, TS)中随机选择元素, 逐层生成算法树. 本文中, 函数集取$\mathrm{FS} = \{+, \times, \exp\}$, 终端集取

$$\mathrm{TS} = \{K_{\mathrm{Pol}}^{c,d}, K_{\mathrm{RBF}}^{\sigma}, K_{\mathrm{Sig}}^{\kappa,\upsilon}, \alpha_i, i = 1, 2, \cdots, n\},$$

其中: $K_{\mathrm{Pol}}^{c,d}$, $K_{\mathrm{RBF}}^{\sigma}$和$K_{\mathrm{Sig}}^{\kappa,\upsilon}$是带有参数的单核函数, α_i是正常数. 在生成算法树过程中, 一般将根节点的选择限制在函数集FS中, 以便生成层次化的复杂结构. 如果从函数集FS中选出的函数f有$\arg(f)$个自变量, 则该节点有$\arg(f)$个分支. 对于每个分支, 需要从终端集TS和函数集FS的并集$C = \mathrm{FS} \cup \mathrm{TS}$中随机选出一个元素作为该分支的尾节点. 如果选出的元素是一个函数, 则重复执行上述过程; 如果选出的是终端集中的元素, 或算法树深度达到了设定的最大深度值, 则该分支上的树就终止生长.

树形结构增强了个体的非线性表达能力和对层次化问题的描述能力. 图2描述了一个个体为$(K_1 + \alpha_1 K_2)\exp(K_3)$的算法树模型, 定理1保证了生成的个体是一个核函数.

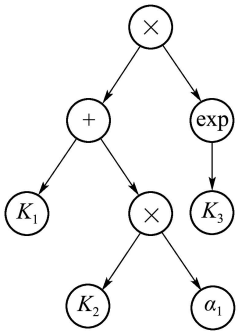

图 2 个体的算法树结构

Fig. 2 Tree-structure of individual

3.3 初始种群产生与适应度计算(Population initialization and fitness computation)

初始种群采用生长法(grow method, GM)[16]产生, 产生过程中必须遵循以下原则:

1) 根节点必须从函数集FS中选择元素;

2) 某一节点元素来自函数集FS, 则其子节点既可以从函数集FS中选取, 也可以从终端集TS中选取;

3) 叶节点必须从终端集TS中选择元素;

4) 每棵算法树至少包含一个单核函数元素;

5) 当算法树达到设定的最大深度, 初始化过程结束;

6) 设定的算法树最大深度必须足够大, 以保证算法树的性能.

个体适应度是驱使GP算法进化的源动力. 由于支持向量机分类精度很好地描述了核函数性能, 本文中采用分类精度作为相应个体的适应度.

3.4 遗传操作(Genetic operation)

遗传程序设计中的遗传操作与遗传算法中的遗传操作相同, 主要包括复制、交叉和变异, 文中不再赘述.

4 算法描述与复杂度分析(Algorithm description and complexity analysis)

4.1 算法描述(Algorithm description)

本算法中, 以是否达到最大进化代数或同一代中个体适应度的最大差值是否小于设定值作为终止准则, 并以进化过程中保存的最佳个体作为运行结果. 具体算法描述如下:

1) 设定核参数取值范围和正常数α_i取值, 建立终端集TS;

2) 设定控制参数, 包括种群规模M、最大进化代数G、算法树最大深度D、复制概率P_{r}、交叉概率P_{c}、变异概率P_{m}和迭代停止条件等;

3) 按生长法产生初始个体集合;

4) 设置$l = 0$;

5) 利用SVM和训练数据集、测试数据集计算初始个体的分类精度, 并作为个体的适应度f_i^l, 其中$i = 1, 2, \cdots, M$, 记携带适应度的初始个体集合为初始种群EMKs$^{(l)}$;

6) 若$\max\limits_{i,j} |f_i^{(l)} - f_j^{(l)}| < \varepsilon$或达到最大进化代数$G$, 则算法停止; 否则对种群EMKs$^{(l)}$进行遗传操作, 产生新的个体集合, 同时置$l = l + 1$, 并返回5).

4.2 复杂度分析(Complexity analysis)

进化多核算法计算时间来源于两部分: 遗传操作和SVM训练. 遗传操作部分以遗传算法为核心, 其时间复杂度为$\mathrm{O}(Tm^2)$, 其中: T为进化代数, m为

种群规模. 在种群规模为M、最大进化代数为G的情况下, 遗传操作部分最大时间复杂度为$\mathrm{O}(GM^2)$. SVM训练部分用于计算每个个体对应的SVM分类精度, 并作为适应度, 驱动遗传操作进行迭代. 因此, 每一个个体的适应度计算都是一个标准的SVM训练过程, 其实质是求解二次规划问题, 时间复杂度为$\mathrm{O}(n^3)$, 其中n是训练样本数. 在种群规模为M, 最大进化代数为G的情况下, SVM训练部分的最大时间复杂度为$\mathrm{O}(MGn^3)$. 因此, 进化多核算法的最大时间复杂度为$\mathrm{O}(GM^2)+\mathrm{O}(MGn^3)$. 由于一般情况下$M \ll n$, 所以进化多核算法的实际复杂度为$\mathrm{O}(MGn^3)$.

5 数据实验(Numerical experiments)

为验证进化多核算法的有效性和优越性, 本部分将所提算法与单核函数以及部分有代表性的多核函数进行对比实验. 实验中单核函数选择鲁棒性较强的高斯径向基核函数, 多核算法选择SILP[7,8]和SimpleMKL[12]. 实验数据选用UCI数据库中的6组二分类数据集Ionosphere, Breast-cancer, A1a, Mushrooms, Pima, Liver-disorders[17], 数据特征描述见表2. 所有实验均在酷睿双核3.0 G, 2 G内存的微机上进行, 并采用MATLAB R2007a编程实现.

表 2 UCI数据集特征描述

Table 2 Feature description of UCI datasets

数据集	#样本	#属性	#类别
Ionosphere	351	34	2
Breast-cancer	683	10	2
Ala	1605	123	2
Mushrooms	8124	112	2
Pima	768	8	2
Liver-disorders	345	6	2

为了更好地分析实验结果, 将实验分为两组: 第1组实验中正则化参数C固定. 固定正则化参数C虽然没有考虑模型选择问题, 但可以更好地对比分析不同算法的性能; 第2组则不固定正则化参数C, 通过寻优准则寻找最佳的C, 并分析不同算法在寻优过程中的性能差异.

5.1 固定正则化参数C(Fixed regularization parameter C)

固定正则化参数$C=100$, 并按表3对核参数进行设定.

由核参数的设定可知, 实验中用到的单核函数$K_{\mathrm{Pol}}^{c,d}$有10个, $K_{\mathrm{RBF}}^{\sigma}$有50个, $K_{\mathrm{Sig}}^{\kappa,\upsilon}$有150个, 总共210个单核函数. 此外, EMK算法中还有TS集非负常数α_i和控制参数需要设定. 由文献[5, 9]可知, 终端集TS中非负常数α_i代表了单核函数的相对权值, 且其建议取值范围为[0, 1]. 本文在[0, 1]区间内随机选取10个值, 与单核函数组成终端集TS. 控制参数用来控制算法的运行过程, EMK算法中主要控制参数有种群规模M、最大进化代数G、算法树最大深度D、复制概率P_{r}、交叉概率P_{c}和变异概率P_{m}, 具体设置如表4所示.

表 3 核参数设定

Table 3 Settings of kernel parameters

核函数	核参数
$K_{\mathrm{Pol}}^{c,d}$	$c=1, d=\{1,2,\cdots,10\}$
$K_{\mathrm{RBF}}^{\sigma}$	$\sigma=\eta\cdot 10^{\tau}, \eta=\{1,2,\cdots,10\}$ $\tau=\{-5,-4,\cdots,-1\}$
$K_{\mathrm{Sig}}^{\kappa,\upsilon}$	$\kappa=\sigma, \upsilon=\{0.1,1,10\}$

表 4 控制参数设定

Table 4 Settings of control parameters

参数名称	参数设定值
种群规模	$M=100$
最大进化代数	$G=50$
算法树最大深度	$D=8$
复制概率	$P_{\mathrm{r}}=0.1$
交叉概率	$P_{\mathrm{c}}=0.9$
变异概率	$P_{\mathrm{m}}=0.01$

本文提出的EMK算法采用MATLAB编程实现, 文献[8]提供了SILP算法的源代码下载地址, 文献[12]提供了SimpleMKL算法的源代码下载地址, 两者均采用MATLAB编程实现.

为增加可比性, 所有算法中采用相同的停机准则, 即对偶间隙小于0.01或者迭代次数超过1000时, 算法停止.

针对每一个数据集, 算法运行50次, 每次从数据集中随机选取70%作为训练样本, 30%作为测试样本, 记录平均实验结果. 实验之前, 已对所有样本进行归一化处理.

表5记录了上述算法的实验结果. 其中: M表示训练样本数, N表示单核函数个数.

表 5 UCI数据集实验结果

Table 5 Experimental results on UCI datasets

Ionosphere $M=246,\ N=210$				
算法	RBF	SILP	SimpleMKL	EMK
测试精度/%	89.8±2.3	92.2±1.8	92.4±1.7	94.1±1.5
训练时间/s	47±19	408±91	114±35	79±28
单核函数个数	1±0	18±3.2	21±3.6	25±4.2
迭代的次数	21±6	382±44	104±33	62±20

(转下页)

(接上页)

Breast-cancer $M=478, N=210$				
算法	RBF	SILP	SimpleMKL	EMK
测试精度/%	90.1±1.1	92.6±1.4	92.5±1.5	95.1±0.8
训练时间/s	35±11	192±35	86±22	53±19
单核函数个数	1±0	17±1.6	18±1.2	26±1.2
迭代的次数	18±7	104±22	33±16	21±17
Ala $M=1124, N=210$				
算法	RBF	SILP	SimpleMKL	EMK
测试精度/%	82.1±3.9	84.3±1.1	84.3±1.2	87.5±0.9
训练时间/s	70±12	547±49	213±33	111±22
单核函数个数	1±0	21±2.1	25±2.8	29±2.7
迭代的次数	20±2	179±29	68±18	38±15
Mushrooms $M=5687, N=210$				
算法	RBF	SILP	SimpleMKL	EMK
测试精度/%	89.3±1.1	92.6±1.0	92.5±1.4	95.1±1.6
训练时间/s	533±88	2011±106	1861±118	844±96
单核函数个数	1±0	32±4.1	39±3.6	45±3.4
迭代的次数	14±11	221±26	99±29	58±18
Pima $M=538, N=210$				
算法	RBF	SILP	SimpleMKL	EMK
测试精度/%	74.6±1.8	77.2±2.2	77.2±2.4	81.1±2.9
训练时间/s	38±9	205±32	99±19	66±14
单核函数个数	1±0	14±1.6	15±2.1	21±3.3
迭代的次数	18±5	113±19	38±11	25±8
Liver-disorders $M=242, N=210$				
算法	RBF	SILP	SimpleMKL	EMK
测试精度/%	60.1±1.9	65.9±2.6	65.9±2.3	66.8±2.5
训练时间/s	14.8±8.2	56.6±10.5	26.7±14.3	18.4±14.1
单核函数个数	1±0	16.6±1.3	17.8±1.2	24.9±1.5
迭代的次数	19±6	106±22	44±19	31±15

从分类精度来看, RBF单核算法的分类精度最低, SILP和SimpleMKL基本相同, EMK算法的分类精度最高. 这一点充分说明了多核函数能够更好地描述数据特征, 尤其是EMK函数, 由于其具有较强的非线性表达能力而大大提高了算法的分类精度. 这一点也可以从基于不同核函数算法中用到的单核函数个数进行分析, 由表5可以看出, SimpleMKL选择的单核函数略多于SILP, 而EMK选择的单核函数最少比SimpleMKL多出15%以上(Mushrooms), 最多的则达到了44%(Breast-cancer). 由前文分析可知, 不同核函数对应不同映射函数和特征空间, 因此, 单核函数选择越多, 组合的非线性越强, 算法对实际问题的描述能力也就越强. 这也是EMK算法分类精度高于其他多核算法的原因.

从训练时间来看, RBF单核算法的训练时间最少, 3种多核算法中以EMK训练速度最快, SILP最慢. 这一点可以从迭代次数来分析, 由表5可以看出, 单核函数RBF迭代次数最少, 而多核算法中SILP迭代次数最多, SimpleMKL次之, EMK迭代次数最少. 由于上述算法的迭代过程即是支持向量机训练过程, 因此, 迭代次数的多少直接反映了训练速度的大小.

5.2 寻优正则化参数C(Searching optimal regularization parameter C)

实际应用中, 正则化参数C的最优值并不知道, 只能通过多次求解支持向量机, 在一定范围内搜索其最优值, 而交叉验证方法是其常用的判断准则.

本节实验用到的数据与5.1节相同, 目的是通过实验对比分析不同核函数在正则化参数C的寻优过程中所耗费的时间差异. 实验中采用5–折交叉验证方法作为判断准则, 在区间[0.01,1000]内以10的幂次级均匀抽样获得C值. 表6记录了不同算法在10次C寻优过程中耗费的平均时间.

由表6结果可以看出, 多核算法中, EMK的寻优效率最高, SimpleMKL次之, SILP最差. 同时, 多核算法的寻优时间均高于单核函数RBF.

表 6 寻优正则化参数耗时对比

Table 6 Comparison of consuming time in searching optimal regularization coefficient

数据集	RBF	SILP	SimpleMKL	EMK
Ionosphere	242±106	7980±1912	1189±762	866±641
Breast-cancer	181±97	3120±867	967±536	802±443
Ala	396±145	8641±1421	2017±1211	1113±988
Mushrooms	2663±912	14771±5662	9982±2110	5321±1449
Pima	210±81	5906±1163	986±201	667±186
Liver-disorders	62±8	824±119	132±31	89±22

5.3 结果讨论(Result and discussion)

显然, 进化多核算法EMK比现有多核算法具有更高的分类精度和更快的训练速度, 这一结论可以从实验结果得出. 但还应注意到, 与单核算

法RBF相比, 多核算法在提高分类精度的同时, 却降低了训练效率.

限于篇幅, 文中没有给出进化多核算法EMK在大规模数据集上的性能讨论, 以及随着单核函数集合数量的变化, EMK性能变化情况. 有兴趣的读者可与作者进一步交流.

6 结论(Conclusion)

本文借助GP思想, 构造了进化多核函数. 算法中采用树形结构对多核函数的数学表达式进行编码, 丰富了多核函数的层次化结构, 提高了其非线性表达能力. 此外, 由于GP算法是在搜索空间中进行全局搜索, 因而可以在没有具体问题先验知识的前提下得到最优解. 对比实验结果表明: 进化多核算法EMK比现有多核算法具有更高的泛化能力和收敛速度.

下一步将研究如何根据具体问题确定最佳的算法树最大深度, 并将EMK方法应用到回归问题和多分类问题中.

参考文献(References):

[1] VAPNIK V. *The Nature of Statistical Learning Theory*[M]. New York: Springer-Verlag, 1995.

[2] CHAPELLE O, VAPNIK V, BOUSQUET O, et al. Choosing multiple parameters for support vector machines[J]. *Machine Learning*, 2002, 46(1/3): 131 – 159.

[3] ONG C S, SMOLA A J, WILLIAMSON R C. Hyperkernels[M] //*Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003: 478 – 485.

[4] HU M Q, CHEN Y Q, KWOK J T Y. Building sparse multiple-kernel SVM classifiers[J]. *IEEE Transactions on Neural Network*, 2009, 20(5): 827 – 839.

[5] LANCKRIET G, CRISTIANINI N, BARTLETT P, et al. Learning the kernel matrix with semidefinite programming[J]. *Journal of Machine Learning Research*, 2004, 5(1): 27 – 72.

[6] BACH F R, LANCKRIET G R G, JORDAN M I. Multiple kernel learning, conic duality and the smo algorithm[C] //*Proceedings of the 21st International Conference on Machine Learning*. Banff: ACM Press, 2004: 6 – 13.

[7] SONNENBURG S, RÄTSCH G, SCHÄFER C. A general and efficient multiple kernel learning algorithm[J]. *Advances in Neural Information Processing Systems*, 2006, 18(1): 1273 – 1280.

[8] SONNENBURG S, RÄTSCH G, SCHÄFER C, et al. Large scale multiple kernel learning[J]. *Journal of Machine Learning Research*, 2006, 7(7): 1531 – 1565.

[9] DIOSAN L, OLTEAN M, ROGOZAN A, et al. *Improving SVM Performance Using a Linear Combination of Kernels*[M]. Heidelberg: Springer, 2007.

[10] CRAMMER K, KESHET J, SINGER Y. Kernel design using boosting[M] //*Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003.

[11] BENNETT K P, MOMMA M, EMBRECHTS M J. MARK: a boosting algorithm for heterogeneous kernel models[C] //*Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Edmonton: ACM Press, 2002: 24 – 31.

[12] RAKOTOMAMONJY A, BACH F, CANU S, et al. Simple MKL[J]. *Journal of Machine Learning Research*, 2008, 9(11): 2491 – 2521.

[13] CRISTIANINI N, SHAWE-TAYLOR J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*[M]. Cambridge: Cambridge University Press, 2000.

[14] 邓乃扬, 田英杰. 数据挖掘中的新方法: 支持向量机[M]. 北京: 科学出版社, 2006.
(DENG Naiyang, TIAN Yingjie. *New Approach for Data Mining: Support Vector Machine*[M]. Beijing: Science Press, 2006.)

[15] KOZA J R. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*[M]. Cambridge: MIT Press, 1992.

[16] 王四春. GP技术及应用研究[D]. 长沙: 中南大学, 2006.
(WANG Sichun. *Research on GP technologies and applications*[D]. Changsha: Central South University, 2006.)

[17] ASUNCION A, NEWMAN D J. *UCI Machine Learning Repository*[DB/OL]. http://www.ics.uci.edu/∼mlearn/MLRepository.html, Irvine, CA: University of California, 2007.

作者简介:

李仁兵 (1982—), 男, 博士研究生, 主要研究方向为模式识别、机器学习算法及导弹故障诊断, E-mail: pioneerbull@sina.com;

李艾华 (1966—), 男, 教授, 博士生导师, 主要研究方向为智能控制、信号处理及故障诊断, E-mail: plamissile@sina.com;

白向峰 (1981—), 男, 博士研究生, 主要研究方向为目标检测与跟踪, E-mail: bxf2024@163.com;

蔡艳平 (1982—), 男, 博士研究生, 主要研究方向为故障诊断与信号处理, E-mail: 287468105@qq.com;

王德生 (1984—), 男, 讲师, 主要研究方向为导弹发动机故障检测与诊断, E-mail: 287290889@qq.com.