文章编号: 1001-0920 (2010) 08-1133-06

基于带特征染色体遗传算法的支持向量机特征选择和参数优化

赵明渊[1,2], 唐 勇[1], 傅 翀[1], 周明天[1]

(1. 电子科技大学 计算机科学与工程学院，成都 610054；2. 中国农业银行 四川省分行，成都 610015)

摘 要: 鉴于支持向量机特征选择和参数优化对其分类准确率有重大的影响, 将支持向量机渐近性能融入遗传算法并生成特征染色体, 从而将遗传算法的搜索导向超参数空间中的最佳化误差直线. 在此基础上, 提出一种新的基于带特征染色体遗传算法的方法, 同时进行支持向量机特征选择和参数优化. 在与网格搜索、不带特征染色体遗传算法和其他方法的比较中, 所提出的方法具有较高的准确率、更小的特征子集和更少的处理时间.

关键词: 特征染色体；遗传算法；特征选择；参数优化；支持向量机

中图分类号: TP183 **文献标识码:** A

Feature selection and parameter optimization for SVM based on genetic algorithm with feature chromosomes

ZHAO Ming-yuan[1, 2], *TANG Yong*[1], *FU Chong*[1], *ZHOU Ming-tian*[1]

(1. School of Computer Science and Technology，University of Electronic Science and Technology of China, Chengdu 610054，China；2. Sichuan Branch，Agricultural Bank of China，Chengdu 610054，China. Correspondent: ZHAO Ming-yuan，E-mail：zmgyn@mail.sc.cninfo.net)

Abstract: The classification accuracy of support vector machines(SVM) depends on feature selection and parameter optimization of SVM strongly. The asymptotic behaviors of support vector machines are fused with genetic algorithm and the feature chromosomes are generated, which directs the search of genetic algorithm to the straight line of optimal generalization error in the superparameter space. On this basis, a new approach based on genetic algorithm with feature chromosomes is proposed to simultaneously optimize the feature subset and the parameters for SVM. Compared with the grid search, the genetic algorithm without feature chromosomes and other approaches, the proposed approach has higher classification accuracy, smaller feature subset and fewer processing time.

Key words: Feature chromosomes； Genetic algorithm； Feature selection； Parameters optimization； Support vector machines

1 引 言

支持向量机是由 Vapnik 首先提出的一种新兴的数据分类技术[1], 已广泛应用于模式分类、文本分类、生物信息学、金融学等多种领域. 在改进支持向量机的学习能力和泛化性能的研究工作中, 一个重要的问题是如何同时优化支持向量机的特征子集和参数以提高其分类准确率.

在相关的研究工作中, 文献 [2] 提出的网格搜索是一个可供选择的简单易用的搜索方法, 但其搜索能力差且不能进行特征选择. [3] 提出了基于遗传算法的特征选择方法, [4] 提出了支持向量机特征选择方法, 但这些文献均没有处理支持向量机参数优化. [5] 提出了基于遗传算法的同时进行支持向量机特征选择和参数优化的方法.

文献 [2-5] 所提出的方法均未能融入支持向量机的渐近性能. 文献 [6] 提出的双一维搜索以支持向量机的渐近性能为基础, 但未能与遗传算法的搜索功能相结合, 也不能进行特征选择. 基于此, 本文提出一种新的基于带特征染色体遗传算法的方法, 使支持向量机的渐近性能融入遗传算法并生成特征染色体, 从而将遗传算法的搜索导向超参数空间中的最佳泛化误差直线, 并同时进行支持向量机特征选择和参数优化.

收稿日期: 2009-07-22；**修回日期:** 2009-12-23.

基金项目: 国家自然科学基金项目(60671033)；教育部博士点基金项目(20060614015).

作者简介: 赵明渊(1940−), 男, 重庆人, 高级工程师, 博士生, 从事支持向量机、遗传算法等研究；周明天(1939−), 男, 广西容县人, 教授, 博士生导师, 从事网络计算、分布式计算等研究.

2 相关工作

2.1 支持向量机及其参数

给定训练集的实例 - 标号对 $(\boldsymbol{x}_i,\ y_i), i=1,2,\cdots,l,\ \boldsymbol{x}_i\in R^{(n)},\ y_i\in\{1,-1\}$. 支持向量机优化问题的原问题转换为其对偶问题[6], 即

$$\min_{\alpha}\ \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{i=1}^{N}\alpha_i;$$

$$\text{s.t.}\ \sum_{i=1}^{l}\alpha_i y_i=0,\ 0\leqslant\alpha_i\leqslant C,\ i=1,2,\cdots,l. \quad (1)$$

其中: $\{\alpha_i\}_{i=1}^{l}$是 Lagrange 乘子, $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$是核函数. 通过映射函数 $z_i=\varphi(\boldsymbol{x}_i)$, 训练向量$\boldsymbol{x}_i$被映射到一个高维空间.

核函数 $K(\boldsymbol{x},\boldsymbol{x}_i)$有多种形式, 本文仅讨论其中应用广泛的高斯核函数. 描述为

$$K(\boldsymbol{x},\boldsymbol{x}_i)=\exp(-\gamma\|\boldsymbol{x}-\boldsymbol{x}_i\|^2) \quad (2)$$

或

$$K(\boldsymbol{x},\boldsymbol{x}_i)=\exp\Big(-\frac{1}{\sigma^2}\|\boldsymbol{x}-\boldsymbol{x}_i\|^2\Big). \quad (3)$$

高斯核支持向量机的参数指惩罚参数 C 和核参数 γ, 即参数对 (C,γ) 均由用户指定.

2.2 支持向量机的渐近性能

在 Keerthi 等[6]提出的高斯核支持向量机的渐近性能中, 采用 $\log C$ 和 $\log\sigma^2$ 作为超参数空间的参数. 在 $(\log C,\log\sigma^2)$ 空间的渐近区域中, 存在一个泛化误差等高线, 分离超参数空间成两个区域: 一个过拟合和欠拟合区域, 一个良好区域. 该良好区域最可能具有最佳泛化误差的超参数集, 如图 1 所示.

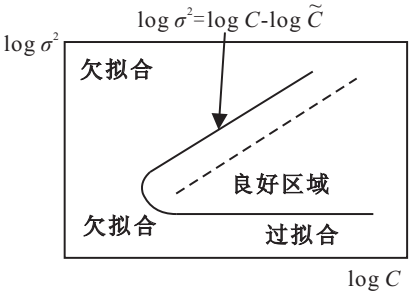

图 1 超参数空间

对于每一个固定的 $\tilde{C}$, 等式

$$\log\sigma^2=\log C-\log\tilde{C} \quad (4)$$

定义了一条具有单位斜率的直线. 当 $\sigma^2\to\infty$ 且沿着这条直线时, SVM 分类器收敛到具有惩罚参数 $\tilde{C}$ 的线性 SVM 分类器. 图 1 中的点线对应 $\tilde{C}$ 值的选择, 给出了线性 SVM 最佳泛化误差.

3 带特征染色体的遗传算法

遗传算法是一个基于模拟达尔文自然选择和生物系统遗传的自适应优化搜索方法, 它采用适应度函数和概率变换规则来指导搜索方向. 遗传操作包括交叉操作、变异操作和选择操作等[7,8]. 将遗传算法应用于支持向量机特征选择和参数优化, 所得到的结果是优化的参数和特征子集.

本文提出的带特征染色体遗传算法以遗传算法为基础, 它与不带特征染色体遗传算法的根本区别是: 当应用于支持向量机特征选择和参数优化时, 融入了支持向量机的渐近性能, 通过生成特征染色体操作生成特征染色体. 新生成的特征染色体和子代染色体经过适应度评价, 选择其中适应值高的染色体构成下一代种群, 继续进行下一代的遗传操作和特征染色体操作. 带特征染色体遗传算法流程如图 2 所示.

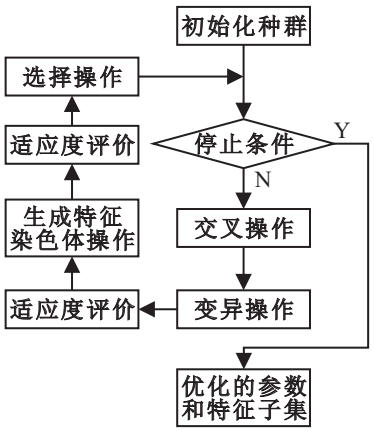

图 2 带特征染色体遗传算法流程

3.1 染色体编码

编码是带特征染色体遗传算法的重要步骤, 它将支持向量机参数对(C,γ) 的变量值和特征子集选择 f 转换为二进制编码. 带特征染色体遗传算法的染色体编码由参数对编码和特征子集选择编码两部分组成, 如图 3 所示.

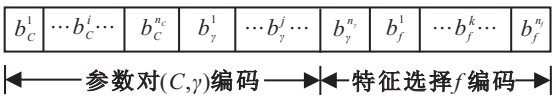

图 3 染色体编码

在参数对 (C,γ) 的编码中, $b_C^1\sim b_C^{n_C}$ 为参数 C 的二进制编码, $b_\gamma^1\sim b_\gamma^{n_\gamma}$ 为参数 γ 的二进制编码, n_C 为参数 C 的二进制位数, n_γ 为参数 γ 的二进制位数, n_C 和 n_γ 依据计算精度选择.

在特征子集选择 f 的编码中, 1 代表该特征被选择, 0 代表该特征未被选择; $b_f^1\sim b_f^{n_f}$ 为特征子集选择 f 的二进制编码, n_f 为其二进制位数, n_f 等于该数据集的特征数.

3.2 初始种群的选择

随机产生 N 对母体组成的初始种群 $X(0)$, 置 $t:=0$. 如果种群规模过大, 则算法复杂度较高, 计算量较大; 如果种群规模过小, 则降低算法优化性能, 易

陷入局部优化解. 依据训练样本的规模, 种群规模大小选择 40~200 为宜.

3.3 交叉操作

交叉操作通过独立地对当前种群 $X(t)$中的 N 对母体执行交叉, 生成中间种群 $C(t)$, 交叉算子 T_c : $X(t) \to C(t)$作用于个体空间的子空间.

3.4 变异操作

变异操作通过独立地对中间种群 $C(t)$ 中的中间个体执行变异, 生成子代种群 $M(t)$. 变异算子 T_m : $C(t) \to M(t)$ 不断改变子空间, 具有个体空间全空间的搜索能力.

3.5 适应度函数

支持向量机分类准确率、被选择的特征数和特征代价用于构造适应度函数. 高的适应度取决于高的分类准确率、小的特征数和低的特征代价[5]. 此外, 考虑到避免分母趋于零, 适应度函数为

$$\text{fit} = W_A A + W_F \Big(P + F_i \sum_{i=1}^{n_f} C_i\Big)^{-1}. \tag{5}$$

其中: A 为分类准确率, W_A 为分类准确率权值, 由用户设置, 一般可设置为 0.75~1; P 为避免分母趋近于零而设置的常数, 一般可取为 1~10; C_i 为特征代价, UCI 的有关数据集有不同的特征代价, 如果没有特征代价信息, 则可设置为 1~8 之间的同样值; F_i 为特征值, 如果第 i 个特征被选择, 则 F_i 为 1, 如果第 i 个特征未被选择, 则 F_i 为 0; W_F 为特征权值且 $W_F = 1 - W_A$.

3.6 生成特征染色体操作

在 $(\log C, \log \sigma^2)$ 超参数空间的良好区域中, 选择适当的 $\tilde{C}$ 值, 可得最佳泛化误差直线 (图 1 中的点线), 即最高准确率的优化直线, 从而找到优化参数, 难点是如何选择适当的 $\tilde{C}$ 值.

本文提出, 应用遗传算法的搜索功能, 在每一代的染色体中, 选择适应值最高的 r 个染色体, 进行 $\tilde{C}$ 值的适当选择. 由超参数空间到遗传算法环境, 将式 (4) 变形得

$$\tilde{C} = C/\sigma^2. \tag{6}$$

为表述方便, 将式 (6) 中的 $\tilde{C}$ 替换为 K, σ^2 替换为 γ, 得到

$$K = C\gamma. \tag{7}$$

式 (7) 是支持向量机渐近性能应用到遗传算法的重要公式.

在选取的 r 个染色体中的每一个染色体中, 分别取出参数对 (C, γ)编码并转换成相应的变量值, 由式 (7) 分别计算其 K 值. 将参数 C 变量值的搜索范围离散化为 d 个值, 通过 $\gamma = K/C$ 计算出参数 γ 的 d 个值, 生成 $r \times d$ 个新的参数对. 再将它们分别转换为二进制编码, 并与原选取的染色体中相应的特征选择编码相连接, 共生成如下 $r \times d$ 个新的染色体: $(C_{11}, \gamma_{11}, f_1), \cdots, (C_{1d}, \gamma_{1d}, f_1), (C_{21}, \gamma_{21}, f_2), \cdots, (C_{2d}, \gamma_{2d}, f_2), \cdots, (C_{r1}, \gamma_{r1}, f_r), \cdots, (C_{rd}, \gamma_{rd}, f_r)$.

新生成的染色体选择了适当的 $\tilde{C}$ 值 (即 K 值), 蕴涵了支持向量机渐近性能的特征, 将新生成的染色体称为特征染色体. 在当代和进化后各代的特征染色体中, 含有包含优化参数 C 和参数 γ 的特征染色体. 生成特征染色体的操作步骤如下:

Step 1: 选取生成特征染色体的父本. 初始化时设定生成特征染色体个数为 fc, 参数 C 变量值搜索范围的离散值个数为 d, 则应选取适应值最高的染色体个数 $r = fc/d$. 将当代经交叉和变异后的染色体按适应值大小从高到低排列, 选取其中适应值最高的 r 个染色体作为生成特征染色体的父本. 取出该父本染色体的参数对 (C_i, γ_i) 进行编码并将其转换成相应的变量值.

Step 2: 计算每一个父本的 K 值为

$$K_i = C_i \gamma_i,\ i = 1, 2, \cdots, r. \tag{8}$$

Step 3: 将参数 C 变量值的搜索范围离散化为 d 个值, 则参数 C_i 的 d 个值为 $C_{i1}, C_{i2}, \cdots, C_{id}, i = 1, 2, \cdots, r$.

Step 4: 由 K 和 C, 利用式 (9) 计算对应参数 γ 为

$$\gamma_{ij} = K_i/C_{ij},\ i = 1, 2, \cdots, r,\ j = 1, 2, \cdots, d. \tag{9}$$

Step 5: 生成特征染色体. 将生成的 $r \times d$ 个参数对的变量值分别转换为二进制编码, 并与原选取的染色体中相应的特征选择编码相连接, 共生成 $r \times d$ 个特征染色体.

在生成特征染色体操作中, 通过计算子代种群 $M(t)$ 中高适应值染色体的 $\tilde{C}$ 值, 将支持向量机的渐近性能融入遗传算法并生成特征染色体种群 $F(t)$. 生成的特征染色体算子 $T_f : M(t) \to F(t)$加强了遗传算法搜索力度并提高了支持向量机的分类准确率.

3.7 选择操作

选择操作采用父子混合遗传算法中的父子混合选择[8], 在父代种群 $X(t)$, 子代种群 $M(t)$和特征染色体种群 $F(t)$中, 选择 N 对母体作为新一代母体种群 $X(t+1)$, 选择算子 $T_s : M(t) \bigcup F(t) \bigcup X(t) \to X(t+1)$.

4 带特征染色体遗传算法的收敛性分析

定理 1　本文提出的带特征染色体遗传算法依概率 1 收敛到最优解集.

证明 设$\{\boldsymbol{X}(t)\}$是杰出者遗传算法产生的种群序列, $\boldsymbol{\Gamma}$为等位基因, $\boldsymbol{L}$为给定编码长度, 集合$\boldsymbol{H}_L=\{\boldsymbol{A}=a_1a_2\cdots a_L|a_i\in\boldsymbol{\Gamma},\ i=1,2,\cdots,\boldsymbol{L}\}$为个体空间, 乘积$\boldsymbol{H}_L^N=\boldsymbol{H}_L\times\boldsymbol{H}_L\times\cdots\times\boldsymbol{H}_L$为$N$阶种群空间, B^*是问题$\max\limits_{A\in H_L}F(A)$的最优解集. 记

$$B_0=\{\boldsymbol{Y}=(Y_1,Y_2,\cdots,Y_N)\in\boldsymbol{H}_L^N|Y_i\in\boldsymbol{H}_L,\ 1\leqslant i\leqslant N-1,Y_N\in B^*\}. \tag{10}$$

根据文献[8]的定理5.10, $\{\boldsymbol{X}(t)\}$依概率1收敛到种群集B_0, $\lim\limits_{t\to\infty}P\{X(t)\in B_0\}=1$, 即杰出者遗传算法依概率1收敛到最优解集. 同理可证父子混合遗传算法依概率1收敛到最优解集. 带特征染色体遗传算法是父子混合遗传算法, 故本文算法依概率1收敛到最优解集. □

5 实验结果

基于带特征染色体遗传算法的支持向量机特征选择和参数优化的系统体系如图4所示, 采用的平台是Intel Pentium Ⅳ2.2 GHz CPU, 512 MB RAM, Windows XP操作系统, 开发环境为Matlab 7.3, 支持向量机软件为LIBSVM[9].

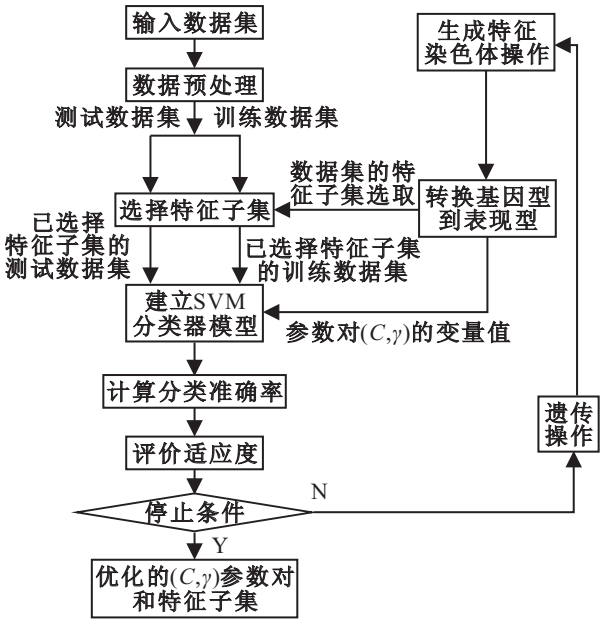

图4 支持向量机特征选择和参数优化的系统体系

在实验中, 参数C的搜索范围为(0.01,35 000), 参数γ的搜索范围为(0.000 1,32)[2]. 为了测试所提出方法的性能, 实验采用现实世界中的基准数据集(UCI机器学习库)[10]中的数据集: Australian, Breast cancer, German, Heart Disease, Ionosphere, Iris, Liver disorders, Pima, Sonar, Vehicle, Vowel和Wine. 其分类数、样本数和特征数如表1所示.

在UCI数据集的实验中, 采用k次折迭交叉确认. k值设置为10, 数据集被随机划分成10个互不相交的子集, 每个子集大小大致相等, 其中1个子集为独立的测试数据集, 其余9个子集为训练模型[11]. 程序运行10次, 每1个子集被轮流1次作为测试数据集, 对每一次运行的测试准确率求和, 总数被10除, 即可得到该数据集在此次实验中的分类准确率. 交叉确认的优点是所有测试集都是独立的, 且测试结果的可靠性能够改进. 在UCI数据集实验中, 10次折迭交叉确认用来估计本文方法在该数据集上的分类准确率.

表1 UCI库中的数据集

序号	数据集	分类数	样本数	特征数
1	Australian (Statlog project)	2	690	14
2	Breast cancer (Wisconsin)	2	699	10
3	German (Statlog project)	2	1000	24
4	Heart Disease (Statlog project)	2	270	13
5	Ionosphere	2	351	34
6	Iris	3	150	4
7	Liver disorders	2	345	6
8	Pima (Indian diabetes)	2	768	8
9	Sonar	2	208	60
10	Vehicle (Statlog project)	4	846	18
11	Vowel	11	990	13
12	Wine	3	178	13

比较的结果取自本文方法和基于GA[5]的方法. 表2为其分类准确率的比较结果. 在实验中, 10次折迭交叉确认用来估计各方法在数据集上的分类准确率, 得到的分类准确率用平均数±标准差的形式表示. 本文方法在9个数据集中产生较高的分类准确率, 在1个数据集中产生的分类准确率与基于GA的方法相同. 因此, 本文方法产生更高的分类准确率.

表2 本文方法和基于GA的方法的分类准确率比较 %

数据集	本文方法	基于GA的方法
Australian	91.59±2.14	88.10±2.25
Breast cancer	99.00±1.66	96.19±1.24
German	86.10±1.97	85.60±1.96
Heart Disease	95.56±2.34	94.80±3.32
Ionosphere	99.43±1.21	98.56±2.03
Iris	100.00±0.00	100.00±0.00
Pima	83.84±5.14	81.50±7.13
Sonar	99.00±2.11	98.00±3.50
Vehicle	88.24±1.47	84.06±3.54
Vowel	99.60±0.71	99.30±0.82

表3表示本文算法和网格搜索[2]在12个UCI数据集上的实验结果. 在实验中, 10次折迭交叉确认用来估计各方法在数据集上的分类准确率. 可以看出, 本文方法产生小的特征子集, 而网格搜索用到所有的特征. 为了比较本文方法和网格搜索的分类准确率, 对所有数据集采用非参数的Wilcoxon符号秩和检验, 除Wine的P值大于规定的显著性水平0.005外, 其他P值小于0.005. 总体而言, 与网格搜索相比较, 本文方法有更高的准确率和更小的特征数.

为了鉴别本文算法和不带特征染色体遗传算法的差异, 在12个UCI数据集上采用10次折迭交叉确认估计各方法在数据集上的分类准确率, 并用非参数的Wilcoxon符号秩和检验进行检验. 如表4所示, 除Ionosphere, Pima, Vowel和Wine的P值大于规定的显著性水平0.005外, 其他P值均小于0.005. 因此, 与不带特征染色体遗传算法相比较, 本文方法有更高的准确率和较小的特征数.

为了进行本文算法、不带特征染色体遗传算法和网格搜索性能对比实验, 数据集采用UCI机器学习库中的4个数据集: Breast cancer, Heart Disease, Ionosphere和Liver. 实验中, 10次折迭交叉确认用来估计各方法在数据集上的分类准确率, 得出的分类准确率、选择的特征数和处理时间都用10次实验的平均数表示, 如表5所示. 可以看出, 在Breast cancer数据集上, 本文方法获得的分类准确率为99.00%, 相应的选择特征数为2.5, 平均处理时间为13.28 s. 然而在此数据集上, 网格搜索仅获得95.43%的分类准确率, 不能进行特征选择, 相应的平均处理时间为15.90 s. 同样在此数据集上, 不带特征染色体遗传算法只获得96.04%的分类准确率, 相应的选择特征数为2.9, 平均处理时间为14.49 s. 本文方法的分类准确率比网格搜索和不带特征染色体遗传算法分别高出3.57%和2.96%, 且本文方法的平均处理时间比网格搜索和不带特征染色体遗传算法分别少2.62 s和1.21 s. 在其余3个数据集中, 除去在Heart Disease数

表 3　本文方法和网格搜索的实验结果

数据集	特征数	带特征染色体遗传算法		网格搜索	Wilcoxon
		分类准确率/%	选择的特征数	分类准确率/%	检验的 P 值
Australian	14	91.59±2.14	5.2±2.15	84.74±4.52	<0.005
Breast cancer	10	99.00±1.66	2.5±0.88	95.43±2.50	<0.005
German	24	86.10±1.97	10.3±1.76	78.90±1.73	<0.005
Heart Disease	13	95.56±2.34	6.2±1.12	88.15±5.18	<0.005
Ionosphere	34	99.43±1.21	13.9±3.45	94.29±3.56	<0.005
Iris	4	100.00±0.00	1.2±0.28	94.09±4.77	<0.005
Liver	6	83.14±7.19	2.8±0.99	72.89±5.60	<0.005
Pima	8	83.84±5.14	3.7±1.26	76.58±5.14	<0.005
Sonar	60	99.00±2.11	26.4±3.20	90.50±8.32	<0.005
Vehicle	18	88.24±1.47	9.2±1.71	83.94±2.74	<0.005
Vowel	13	99.60±0.71	5.5±1.58	95.36±2.24	<0.005
Wine	13	100.00±0.00	4.2±0.50	97.16±2.88	0.0054

表 4　本文方法和不带特征染色体遗传算法的实验结果

数据集	特征数	带特征染色体遗传算法		不带特征染色体遗传算法		Wilcoxon
		分类准确率/%	选择的特征数	分类准确率/%	选择的特征数	检验的 P 值
Australian	14	91.59±2.14	5.2±2.15	86.81±3.64	6.7±3.16	<0.005
Breast cancer	10	99.00±1.66	2.5±0.88	96.04±2.18	2.9±0.99	<0.005
German	24	86.10±1.97	10.3±1.76	80.80±2.10	11.8±3.33	<0.005
Heart Disease	13	95.56±2.34	6.2±1.12	91.11±2.58	7.0±1.05	<0.005
Ionosphere	34	99.43±1.21	13.9±3.45	98.57±2.02	15.4±3.32	0.3222
Iris	4	100.00±0.00	1.2±0.28	96.00±3.44	1.8±0.38	<0.005
Liver	6	83.14±7.19	2.8±0.99	81.43±7.29	3.2±1.14	<0.005
Pima	8	83.84±5.14	3.7±1.26	81.97±5.34	5.1±1.63	0.4427
Sonar	60	99.00±2.11	26.4±3.20	95.00±2.36	28.7±4.00	<0.005
Vehicle	18	88.24±1.47	9.2±1.71	84.74±2.32	10.3±2.72	<0.005
Vowel	13	99.60±0.71	5.5±1.58	98.79±1.70	6.9±1.60	0.3980
Wine	13	100.00±0.00	4.2±0.50	99.44±1.76	4.6±0.72	0.3681

表 5　本文方法、不带特征染色体遗传算法和网格搜索性能对比

数据集	分类准确率/%			选择的特征数		平均处理时间/s		
	带特征染色体遗传算法	不带特征染色体遗传算法	网格搜索	带特征染色体遗传算法	不带特征染色体遗传算法	带特征染色体遗传算法	不带特征染色体遗传算法	网格搜索
Breast cancer	99.00	96.04	95.43	2.5	2.9	13.28	14.49	15.90
Heart Disease	95.56	91.11	88.15	6.2	7.0	14.53	11.41	12.08
Ionosphere	99.43	98.57	94.29	13.9	15.4	11.84	12.15	16.32
Liver	83.14	81.43	72.89	2.8	3.2	6.37	10.45	11.90

据集上的平均处理时间外, 本文方法获得了最高的分类准确率、最少的选择的特征数和最少的平均处理时间. 因此, 与不带特征染色体遗传算法和网格搜索相比, 本文方法具有更好的性能.

6 结 论

本文提出了一种新的基于带特征染色体遗传算法的方法, 该方法具有以下特点:

1) 在构成特征染色体时选择了超参数空间中适当的$\tilde{C}$值, 将遗传算法的搜索导向超参数空间中的最佳泛化误差直线, 其本质是将应用问题的性能特征融入算法, 因此加强了遗传算法的搜索力度, 并提高了支持向量机的分类准确率.

2) 特征染色体是一种新生成的染色体, 它为进化的中、后期在群体中引进适应值高的新染色体和避免早熟收敛提供了一种新的方法和思路.

参考文献(References)

[1] Vapnik V N. The nature of statistical learning theory[M]. New York: Springer, 1995.

[2] Hsu C W, Chang C C, Lin C J. A practical guide to support vector classification[DB/OL]. http://www.csie.ntu.edu.tw/˜cjlin/papers/guide/guide.pdf. 2006-07-22.

[3] Yang J, Honavar V. Feature subset selection using a genetic algorithm[J]. IEEE Intelligent Systems, 1998, 13(2): 44-49.

[4] Mao K Z. Feature subset selection for support vector machines through discriminative function pruning analysis[J]. IEEE Trans on Systems, Man and Cybernetics, 2004, 34(1): 60-67.

[5] Huang C L, Wang C J. A GA-based feature selection and parameters optimization for support vector machines[J]. Expert Systems with Applications, 2006, 31(2): 231-240.

[6] Keerthi S S, Lin C J. Asymptotic behaviors of support vector machines with Gaussian Kernel[J]. Neural Computation, 2003, 15(7): 1667-1689.

[7] Wang D W, Wang J W, Wang H F, et al. Intelligent optimization methods[M]. Beijing: Higher Education Press, 2007.

[8] XU Z B. Computational intelligence simulated evolutionary computation[M]. Beijing: Higher Education Press, 2004.

[9] Chang C C, Lin C J. LIBSVM: A library for support vector machines[DB/OL]. http : // www. csie. ntu. edu. tw/ cjlin/libsvm. 2006-04-06.

[10] Murphy P M, Aha D W. UCI repository of machine learning database[DB/OL]. http : // www. ics. uci. edu/ mlearn /MLRepository.html. 2006-05-12.

[11] Han J, Kamber M. Data mining: Concepts and techniques[M]. San Francisoc: Morgan Kaufmann, 2003.

(上接第1132页)

[2] Deb K. An efficient constraint handling method for genetic algorithm[J]. Computation Methods in Applied Mechanics and Engineering, 2000, 186(2-4): 311-338.

[3] Runarsson TP, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Trans on Evolutionary Computation, 2000, 4(3): 284-294.

[4] Farmani R, Wright JA. Self adaptive fitness formulation for constrained optimization[J]. IEEE Trans on Evolutionary Computation, 2003, 7(5): 445-455.

[5] Coello Coello C A, Montes E M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection[J]. Adavanced Engineering Informatics, 2002, 16(3): 193-203.

[6] Cai Z X, Wang Y. A multiobjective optimization based evolutionary algorithm for constrained optimization[J]. IEEE Trans on Evolutionary Computation, 2006, 10(6): 658-675.

[7] 肖赤心, 蔡自兴, 王勇. 一种基于佳点集原理的约束优化进化算法[J]. 控制与决策, 2009, 24(2): 249-253.
(Xiao C X, Cai Z X, Wang Y. Constrained optimization evolutionary algorithm based on good lattice points principle[J]. Control and Decision, 2009, 24(2): 249-253.)

[8] 王勇, 蔡自兴, 曾威. 求解约束优化问题的一种新的进化算法[J]. 中南大学学报, 2006, 37(1): 119-123.
(Wang Y, Cai Z X, Zeng W. A new evolutionary algorithm for solving constrained optimization problems[J]. J of Central South University, 2006, 37(1): 249-253.)

[9] Gen M, Cheng R W. Genetic algorithm and engineering design[M]. New York: John Wiley and Sons, 1997.

[10] Michalewicz Z. Genetic algorithm + data structure = evolutionary programs[M]. New York: Springer-Verlag, 1994.

[11] Montes E M, Coello C A. A simple multimembered evolution strategy to solve constrained optimization problems[J]. IEEE Trans on Evolutionary Computation, 2005, 9(1): 1-17.

[12] Aguirre A H, Rionda S B, Coello Coello C A. Handling constraints using multiobjective optimization concepts[J]. Int J for Numerical Methods in Engineering, 2004, 59(15): 1989-2017.