

基于自学习二元差分进化的多目标特征选择

胡振稳 杨改贞

(黄冈师范学院计算机学院 湖北 武汉 438000)

摘要 为提升特征选择算法的搜索能力,加快收敛速度,提出一种基于自学习二元差分进化的多目标特征选择方法。引入三种算子,基于概率差的二元变异算子来产生最优解,从而快速地引导个体定位潜在的最优区域。另外,引入的净化搜索算子可以提高处于最优区域的精英个体的自学习能力,而具有拥挤距离的非支配排序算子可以降低差分进化中选择算子的计算复杂度。在多个数据集的实验结果表明,提出的方法能够实现高效精确的多目标特征选择。

关键词 自学习 二元差分 多目标 特征选择

中图分类号 TP311.13

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2024.05.041

MULTIPLE OBJECTIVE FEATURE SELECTION BASED ON SELF-LEARNING BINARY DIFFERENTIAL EVOLUTION

Hu Zhenwen Yang Gaizhen

(School of Computer Science, Huanggang Normal University, Wuhan 438000, Hubei, China)

Abstract In order to improve the group search ability and accelerate the convergence speed, a multiple objective feature selection method based on self-learning binary differential evolution is proposed. Three operators were introduced, and the binary mutation operator based on probability difference was used to generate the optimal solution, so as to quickly guide individuals to locate the potential optimal region. The clean search operator was introduced to improve the self-learning ability of the elites in the optimal region, while the non-dominated sorting operator with crowding distance could reduce the computational complexity of the selection operator in differential evolution. The experimental results on multiple data sets show that the proposed method is efficient and accurate on multiple objective feature selection.

Keywords Self learning Binary difference Multiple objective Feature selection

0 引言

分类是机器学习中的一个重要课题,特征选择算法通过去除不相关或冗余的特征,有效地降低了数据的维数,缩短了学习时间,提高了分类性能^[1]。特征选择的效果极大地影响了分类的精度与效率,因此针对特征选择算法的研究引起了广大学者的注意^[2-3]。

在过去的十年中,提出了许多特征选择算法。其中,元启发式方法由于其强大的探索能力,在处理特征

选择问题上显示出了许多优势,包括遗传算法、蚁群优化和粒子群优化等^[4-6]。而上述元启发式方法往往很难同时实现高精度与高效的特征选择。原则上,特征选择是一个 NP 难的组合问题,因为搜索空间的大小随着特征数的增加而显著增加。由于差分进化(Differential Evolution, DE)能通过全局搜索策略找到最优解,因此在处理特征选择问题中得到了广泛的应用。Cheraghchi 等^[7]提出了一种轮盘赌轮结构,使浮点优化器生成的解适合离散特征选择。Faris 等^[8]研究了一种在不消除任何特征的情况下缩小搜索空间的

新型轮结构方法。王萌等^[9]提出了一种基于数据元素的特征选择算法来解决资源匮乏语言中的消解问题。然而,上述研究只考虑了一个目标,即分类精度。将特征选择问题描述为多目标优化问题有利于获得一组最优特征子集,可以满足决策者的各种需求。

为此提出了一系列多目标特征选择方法,李金霞等^[10]提出了两种改进的多目标特征选择算法,将粒子群优化算法中的拥挤距离和 Pareto 优势关系结合起来。Hancer 等^[11]提出了一种结合非支配排序和遗传算子(MOABCFS)的人工蜂群多目标特征选择方法,并引入了二进制 B-MOABCFS 和连续 C-MOABCFS。黄学雨等^[12]研究了一种基于 DE(B-DEMOFS)的二值多目标特征选择算法,用于生物医学文本的实体提取。然而,这些方法有以下缺点。首先,大多采用 DE/rand/1/bin 策略产生候选个体,其中突变的基向量是从群体中随机选择的。由于基向量的随机性,该策略总体收敛速度较慢。其次,群体中的精英个体只负责指导其他个体的搜寻,搜索能力较差。

针对上述方法存在的问题,提出了一种基于自学习二元差分进化的多目标特征选择方法(MOFS-BDE)。在算法中引入了三种算子,其中二元变异算子快速地引导个体定位潜在的最优区域,净化搜索算子提高精英个体的自学习能力,非支配排序算子降低计算复杂度。实验结果表明了提出方法的有效性。

1 问题描述

假设 S 是一个包含 K 个样本和 D 个特征的数据集, $Fset$ 是所有特征的集合,则特征选择问题可以描述为:从所有特征中选择 d 个特征($d \leq D$),用来优化分类错误率和分类精度等目标函数。由于选定特征的数量决定了分类算法的计算成本,因此它也是一个关键的目标函数。本文考虑两个目标:最小化分类错误率(Err)和选择特征的个数。使用二进制字符串对 FS 问题的解集进行编码:

$$X = (x_1, x_2, \dots, x_D), x_j \in \{0, 1\} \quad (1)$$

式中: $x_j = 1$ 表示第 j 个特征被归类到子集 X 中;否则,不归类到子集 X 中。因此,多目标 FS 问题可表述为:

$$\min Err(X), \min |X| \quad (2)$$

$$\text{s. t. } X = (x_1, x_2, \dots, x_D), x_j \in \{0, 1\}, \\ j = 1, 2, \dots, D, 1 \leq |X| \leq D$$

式中: $|X|$ 是集合 X 的特征数。一般而言,两个目标 $Err(X)$ 和 $|X|$ 是相互冲突的,多目标特征选择问题的

目的是同时最小化 $Err(X)$ 和 $|X|$ 。

2 差异进化

在 DE 中,每个个体表示优化问题的一个可能的解决方案。DE 首先随机生成 N 个个体组成初始群体,在每一进化世代中对个体实施以下操作(算子)。

(1) 突变。在突变过程中,DE 利用两个随机向量之间的差异生成一个参考向量。对于第 i 个目标载体 $X_i(t)$,生成新的突变载体 $V_i(t)$:

$$V_i(t) = X_{r_1}(t) + F \cdot (X_{r_2}(t) - X_{r_3}(t)) \quad (3)$$

式中: t 是迭代步骤。三个向量 $V_{r_1}(t)$ 、 $V_{r_2}(t)$ 和 $V_{r_3}(t)$ 是从总体中随机选择的,满足 $r_1 \neq r_2 \neq r_3 \neq i$; $F \in (0, 2)$ 是放大 $V_{r_2}(t)$ 和 $V_{r_3}(t)$ 之间差异的参数。

(2) 交叉。DE 运用离散重组法,即将来自目标向量 $V_i(t)$ 的元素与来自新突变向量 $V_i(t)$ 的元素组合以产生试验向量 $U_i(t)$:

$$U_i(t) = (u_{i,1}(t), u_{i,2}(t), \dots, u_{i,D}(t)) \\ u_{i,j}(t) = \begin{cases} v_{i,j}(t) & U(0,1) < CR \text{ 或 } j = h \\ x_{i,j}(t) & \text{其他} \end{cases} \quad (4)$$

式中: D 是可变维数; $U(0,1)$ 在 0 和 1 之间随机取值, $CR \in [0, 1]$ 是交叉概率,这里应用随机索引 h 来确保 $U_i(t)$ 总是会从 $V_i(t)$ 接收至少一个元素。

(3) 选择。DE 的选择过程非常简单,如果试验向量 $U_i(t)$ 适应度比目标向量 $X_i(t)$ 更好,则将目标向量作为下一代的试验向量 $U_i(t)$,即 $X_i(t+1) = U_i(t)$; 否则, $X_i(t+1) = X_i(t)$ 。

3 多目标特征选择算法

3.1 二元变异算子

考虑群体 $X_i(t)$ 中的第 i 个目标向量,新的突变算子描述如下:

$$\begin{cases} V_i(t) = (v_{i,1}(t), v_{i,2}(t), \dots, v_{i,D}(t)) \\ v_{i,j}(t) = \begin{cases} x_{\text{best},j}(t) & c_{i,j} < \text{随机数} \\ 1 - x_{\text{best},j}(t) & \text{其他} \end{cases} \\ C_i = \begin{cases} \sigma & \text{如果 } X_{\text{best}}(t) < X_i(t) \\ \min(1, F \cdot (X_{r_1}(t) \oplus X_{r_2}(t)) + \sigma) & \text{其他} \end{cases} \end{cases} \quad (5)$$

式中: $X_{\text{best}}(t)$ 是指随机选择的三个向量中的最佳向量; $X_{r_1}(t)$ 和 $X_{r_2}(t)$ 是这三个向量中非最佳的两个向量。 $X_{\text{best}}(t) < X_i(t)$ 表示 $X_{\text{best}}(t)$ 支配 $X_i(t)$ 。符号“ \oplus ”是异或运算符,并且 $C_i = (C_{i,1}, C_{i,2}, \dots, C_{i,D})$ 是通过将 $X_{r_1}(t)$ 和 $X_{r_2}(t)$ 执行异或而产生的概率向量。参

数 σ 是一个小的湍流系数。 $F \in (0, 1)$ 是一个比例参数,用于控制个体在 $X_{\text{best}}(t)$ 中的学习速率。 $\min(1, F \cdot (X_{r1}(t) \oplus X_{r2}(t)) + \sigma)$ 的作用是确保 $F \cdot (X_{r1}(t) \oplus X_{r2}(t)) + \sigma$ 的值小于 1。与式(3)中的常规变异相比,本文的变异算子具有以下特点。

(1) 可以生成具有良好多样性的突变载体。当 $X_{\text{best}}(t)$ 低于 $X_i(t)$ 时,使用概率差,即 $\min(1, F \cdot (X_{r1}(t) \oplus X_{r2}(t)) + \sigma)$,来产生突变向量 $V_i(t)$ 。由于 $X_{r1}(t)$ 和 $X_{r2}(t)$ 都是随机选择的,将它们差异附加到 $X_{\text{best}}(t)$ 中有利于保持突变载体的多样性。

(2) 可以提高种群的收敛性。当 $X_{\text{best}}(t)$ 支配 $X_i(t)$,即 $X_{\text{best}}(t) < X_i(t)$ 时,可以直接将 $X_{\text{best}}(t)$ 设为第 i 个目标载体的突变载体。由于基向量 $X_{\text{best}}(t)$ 优于目标向量 $X_i(t)$,将它们结合在一起可以使新生成的解从 $X_{\text{best}}(t)$ 继承良好的性能。

(3) 采用湍流系数 σ 来保证基向量上的突变概率始终大于 0。这将防止新的解集陷入局部最优。事实上,较大的 σ 值可以提高新解集的多样性,但可能破坏从 $X_{\text{best}}(t)$ 继承的有用信息。另一方面,如果 σ 值太小,则很难使个体摆脱局部最优解。实验结果表明,设置较小的取值范围 $[0.001, 0.01]$ 是合适的。

3.2 净化搜索算子

首先引入相对重要性的概念。假设 u_1 和 u_2 是从非支配解中随机选择的两个特征位, $X = (x_1, x_2, \dots, x_D)$, 满足 $x_{u_1} = 1$ 和 $x_{u_2} = 0$ 。 $X_{X(u_1)=0, X(u_2)=1}$ 是通过分别将 X 的 u_1 和 u_2 设为 0 和 1 而生成的新解。此外, $X_{X(u_1)=0}$ 是通过将 X 的 u_1 设为 0 而生成的另一个解。如果分类精度的退化程度满足以下条件,则相对于 X , u_1 特征比 u_2 特征更重要,表示为 $u_1 >_{\text{imp}} u_2$:

$$\begin{aligned} & | \text{Cerror}(X_{X(u_1)=0}) - \text{Cerror}(X) | > \\ & | \text{Cerror}(X_{X(u_1)=0}) - \text{Cerror}(X_{X(u_1)=0, X(u_2)=1}) | \end{aligned} \quad (6)$$

$$u_1 = 2, u_2 = 5$$

式中: $\text{Cerror}()$ 表示分类错误率; $| \text{Cerror}(X_{X(u_1)=0}) - \text{Cerror}(X) |$ 描述当第 1 个特征添加到子集 $X_{X(u_1)=0}$ 时分类性能的变化程度, $| \text{Cerror}(X_{X(u_1)=0}) - \text{Cerror}(X_{X(u_1)=0, X(u_2)=1}) |$ 描述当第 2 个特征添加到子集 $X_{X(u_1)=0}$ 时分类性能的变化程度。

算法 1 净化搜索

输入: 群体 P_t , 非支配解的集合 S_t 。

输出: 新的群体 P_t 。

1. 从 ST 中随机选择一种解决方案, 设为对照方案, $X_{\text{ref}} = |x_{\text{ref},1}, x_{\text{ref},2}, \dots, x_{\text{ref},D}|$;
2. 从 X_{ref} 中随机选取两个特征位 u_1 和 u_2 , 满足 $X_{\text{ref},u_1} = 1$ 和

$X_{\text{ref},u_2} = 0$

3. 判断两个特征位 u_1 和 u_2 的相对重要性;
4. for $X_h \in S_t$ do
5. 通过以下四种情况生成一个新的个体 X'_h ;
6. 初始化 $X'_h = X_h$;
7. if $x_{h,u_1} = x_{h,u_2} = 1$ then $x'_{h,u_2} = 0$
8. else if $x_{h,u_1} = x_{h,u_2} = 0$ then $x'_{h,u_1} = 1$
9. else if $x_{h,u_1} = 1, x_{h,u_2} = 0$ then $x'_{h,u_1} = 0$
10. else $x'_{h,u_1} = 1, x'_{h,u_2} = 0$
11. end if
12. end if
13. end if
14. if X'_h 对 X_h 产生的影响很大 then 群体 P_t 会用 X'_h 代替 X_h ;
15. else if X_h 对 X'_h 产生的影响很大 then 保持 X_h 不变;
16. else 将 X_h 和 X'_h 都归为 P_t 中;
17. end if
18. end if
19. end for
20. if P_t 的尺寸比 N 大 then 移除 $|P_t| - N$ 个具有高秩的个体, 用第 5.3 节中的方法减少与 P_t 的拥挤距离;
21. 输出群体 P_t 。

3.3 非支配排序算子

快速非支配排序(FNS)是一种较有效的选择方法,已被应用于 DEMOFS 中的最优个体排序。然而,该方法计算量大,时间复杂度为 $O(M \times N^2)$,其中 N 是总体大小, M 是目标数。本节介绍一种改进的选择算法,其将高效非支配排序(Efficient Non-dominated Sorting, ENS)与拥挤距离相结合^[6]。

将第 t 代的目标载体集合设为 P_t , 交叉变异产生的试验载体集合设为 Q_t , 新群体设为 $P_{t+1} = \emptyset$ 。选择过程如下:首先,对个体进行检查。对于 P_t 中的单个 $X_i(t)$, 如果它受控于一个试验向量,则 $U_i(t) \in Q_t$, $U_i(t)$ 将保存到 P_{t+1} 中;如果它控制 $U_i(t)$, 则 $X_i(t)$ 将保存到 P_{t+1} 中;否则, $U_i(t)$ 和 $X_i(t)$ 都将保存到 P_{t+1} 中。如果 P_{t+1} 的尺寸大于总体大小 N , 则总体 P_{t+1} 中的这些解将根据第一个目标函数值按升序排序。如果第一个目标函数的两个解具有相同的值,则使用第二个目标函数对它们排序。其次,通过给每个个体分配一个秩并将每个个体分类到一个秩集中来应用 ENS。最后,根据个体顺序选择个体,形成新的群体。如果两个解具有相同的秩,则选择拥挤距离较大的解。

3.4 MOFS-BDE

算法 2 详细介绍了本文提出的 MOFS-BDE, 该算法在确定个体的基向量时,如果三个随机向量包含多个最优解,则选择拥挤距离最大的一个作为基向

量 X_{best} 。

算法 2 MOFS-BDE

参数:最大迭代次数 T_{max} ,群体大小 N ,实现净化搜索算子的频率 T_{loc} ,规模 F ,交叉概率 CR 。

输入:分类数据集。

输出:Pareto 最优解对应的特征子集。

1. 初始化一系列个体, $P_0 = [X_1, X_2, \dots, X_N]^T$;
2. 令 $t = 0$
3. for $t = 0 : 1 : T_{max}$ do
4. 令集合 $P_{t+1} = \emptyset, i = 1, 2, \dots, N$
5. for $i = 1 : 1 : N$ do
6. 从群体 P_t 中随机选择三个向量, 表示为 $X_{r_1}(t), X_{r_2}(t)$ 和 $X_{r_3}(t), r_1 \neq r_2 \neq r_3 \neq i$;
7. 从这三个向量中选择最佳向量, 作为基本向量 $X_{best}(t)$;
8. 根据式(5)为第 i 个个体生成新的突变载体 $V_i(t)$;
9. 根据式(4)为第 i 个个体生成试验向量 $U_i(t)$;
10. 评价试验向量 $U_i(t)$ 的适合度;
11. if $X_i(t)$ 对 $U_i(t)$ 产生影响 then 将 $X_i(t)$ 保存到 P_{t+1}
12. else if $U_i(t)$ 对 $X_i(t)$ 产生影响 then $U_i(t)$ 保存到 P_{t+1}
13. else 将两者都保存到 P_{t+1} 中。
14. end if
15. end if
16. end for
17. if P_{t+1} 的大小大于 N then 从 P_{t+1} 中移除 $|P_{t+1}| - N$ 个等级更高、拥挤距离更短的个体;
18. end if
18. if $t/T_{loc} = \lfloor t/T_{loc} \rfloor$, 运行局部搜索;
19. else if $t < T_{max}$ then 令 $t++$, 返回步骤 3;
20. else 停止算法, 输出 Pareto 最优解集。
21. end if
22. end if

MOFS-BDE 与 DEMOF 算法的复杂度的比较结果见图 1, 分析仅考虑在每个迭代中评估的个体的数量。可以发现, MOFS-BDE 的总体复杂度为 $O(M \times N \times \log N)$ 。如上所述, 基于多目标 DE 的特征选择算法 (DEMOFS)^[13]、基于 NSGA II 的特征选择算法 (NSGAFS)^[14]、基于粒子群优化的特征选择算法 (MOPSOFS)^[15]、基于二进制多目标 DE 的特征选择算法 (B-DEMOFS)^[16] 和基于人工蜂群优化的 Pareto 前沿特征选择算法 (B-MOABCFS)^[17] 是五种具有代表性的特征选择算法。由于 B-MOABCFS、DEMOFS 和 NSGAFS 都使用非支配排序策略更新种群, 因此它们具有相同的计算复杂度。B-DEMOFS 和 M 净化搜索算子 OFS 由于采用了拥挤距离来精简个体, 因此计算复杂度为 $O(M \times N^2)$ 。因此, 与上述五种进化 FS 算法相比, 所提出的 MOFS-BDE 算法的计算复杂度是较低的。

	MOFS-BDE	DEMOFS
初始化	$O(N)$	$O(N)$
初始化	$O(N)$	$O(N)$
交叉	$O(N)$	$O(N)$
净化搜索	$O(M \times N \times \log N)$	0
个体评估	$O(N)$	$O(N)$
选择	$O(M \times N \times \log N)$	$O(M \times N^2)$
整体复杂度	$O(M \times N \times \log N)$	$O(M \times N^2)$

图 1 算法时间复杂度的比较结果

4 实验与结果分析

4.1 算法和性能指标

为了验证本文提出的 MOFS-BDE 算法的性能, 和四种经典特征选择算法: 基于改进差分的多目标特征选择算法 (DEMOFS)^[7]、基于 NSGA-II 的特征选择算法 (NSGAFS)^[8]、基于粒子群优化的特征选择算法 (MOPSOFS)^[15]、基于 l_2 范数 chebycheff 分解和最大适应度改进的多目标进化算法 (MOEA/D-2TMFI)^[9]、基于分解的多目标进化算法 (MOEA/D)^[5] 和基于人工蜂群优化的 Pareto 前沿特征选择算法 (B-MOABCFS)^[17] 进行比较。由于 NSGA-II 是应用最广泛的多目标进化算法之一, NSGAFS 作为基准方法。为了将 MOEA/D-2TMFI 应用于处理二进制特征选择问题, 使用以下策略将真实个体转换为二进制版本: 如果新个体的元素大于或等于 0.5, 则将其值设为 1; 否则设为 0。

本文利用 K-NN 的 Leave-One-Out 交叉验证方法计算特征子集的分类错误率, 在 LOOCV 中, 从原始数据集中选取一个样本作为测试数据, 其余样本作为训练数据。K-NN 分类器可以预测该测试数据的类标签。如果预测结果不正确, 则增加一个错误预测的样本数, 数据集中的每个样本只使用一次, 特征子集的错误率是错误预测的样本占有所有样本的比例。

在 NSGAFS 中, 采用了位翻转变异和单点交叉, 变异率为 $1/D$, 交叉概率为 0.9。根据文献 [15], 在 DEMOFS 中, 本文将交叉率和比例因子分别设为 0.3 和 0.5。惯性权重 w 设置为 $[0.1, 0.5]$ 内的随机值, 加速度常数 c_1 和 c_2 为 $[1.5, 2.0]$ 内的两个随机值, 突变率为 $1/D$ 。在 B-MOABCFS 中, 群体大小的数目被设置为 50, 资源来源的数目等于群体大小的一半, 极限试验参数设为 5。对于 MOEA/D-2TMFI, 使用 DE 和多项式变异生成新个体, 其中变异率、交叉率和邻域的选

择概率分别设置为 $1/D$ 、0.3 和 0.9。比例因子随机在 $[0,1]$ 之内随机取值。在 MOFS-BDE 中,将选择比例因子 F 设为 $0.5 \times \text{rand}$,交叉概率设为 0.3, T_{loc} 设为 5。对于所有这六种算法,总体大小和内存大小都设置为 50,对于特征少于 100 的数据集,最大评估数设为 5 000,对于其他数据集,最大评估数设置为 15 000。此外,表 2 给出了上述算法的参数配置。

由于 HV 度量可以同时评价解的分布和收敛性,为了评估多目标特征选择算法的性能,在实验中使用了超体积(HV)度量^[17]。此外,利用解集覆盖率(SC)^[16]比较了两种算法得到的 Pareto 最优解的收敛性。利用特征子集的个数(FN)来比较算法得到的解的多样性。因此,FN 值越高,意味着决策者可以选择更多样化的特征子集。

4.2 算法分析

如上所述,本文采用了两种新的算子,即改进的二进制变异算子和净化搜索算子算法来提高 MOFS-BDE 的性能。表 3 显示了表 1 中 8 个典型数据集的 MOFS-BDE、不含净化搜索算子的 MOFS-BDE 和 DEMOF 的平均 HV 值。

首先分析了改进的二元突变对提高 MOFS-BDE 性能的有效性。DEMOFS 和 MOFS-BDE 的主要区别在于突变和选择过程。如上所述,DEMOFS 和 MOFS-BDE 分别采用基于 FNS 和 ENS 的选择。虽然 ENS 算法的计算复杂度比 FNS 算法低,但这两种算法对每个输入得到的结果是相同的。因此,通过比较不带净化搜索算子的 MOFS-BDE 和 DEMOFS 来验证改进的二元变异算子是合理的。从表 3 可以看出,对于 8 个数据集的 6 个数据集,没有净化搜索算子的 MOFSBDE 比 DEMOFS 获得更大的 HV 值,这表明提出的二进制变异对改进算法更有效。

然后比较 MOFS-BDE 和不含净化搜索算子的 MOFS-BDE。表 3 的结果表明,对于具有小特征的数据集(即 Vowel、Vehicle 和 Ionosphere),这两种算法获得的 HV 值相似。然而,对于具有更多特性的 MOFS-BDE,在 HV 度量指标方面,有净化搜索算子的 MOFS-BDE 比没有净化搜索算子的 MOFS-BDE 的性能更好。也就是说,净化搜索算子在 MOFS-BDE 中起着重要作用。此外,还设计了一种基于最佳基向量的最优学习策略,以提高种群的收敛性。式(5)可以改写为:

$$\begin{cases} \mathbf{V}_i(t) = (v_{i,1}(t), v_{i,2}(t), \dots, v_{i,D}(t)) \\ v_{i,j}(t) = \begin{cases} x_{r_3,j}(t) & c_{i,j} < \text{随机数} \\ 1 - x_{r_3,j}(t) & \text{其他} \end{cases} \\ C_i = \min(1, F \cdot (\mathbf{X}_{r_1}(t) \oplus \mathbf{X}_{r_2}(t)) + \sigma) \end{cases} \quad (7)$$

式中: $\mathbf{X}_{r_1}(t)$ 、 $\mathbf{X}_{r_2}(t)$ 和 $\mathbf{X}_{r_3}(t)$ 是随机从群体中选择的,满足 $r_1 \neq r_2 \neq r_3 \neq i$ 。此处用式(6)中的 MOFS-BDE 表示 MOFS-BDE/random。以 Vowel、Ionosphere、Hill-valley 和 LSVT 数据集为例,图 2 描绘了 MOFS-BDE 和 MOFS-BDE/random 的 HV 曲线。研究发现,对于 Vowel 和 Ionosphere,最优学习策略明显加快了 MOFS-BDE 早期种群的收敛速度。对于特征较多的数据集,Hill-valley 和 LSVT,在这种优化学习策略的帮助下,MOFS-BDE 比 MOFS-BDE/random 具有更好的收敛性。

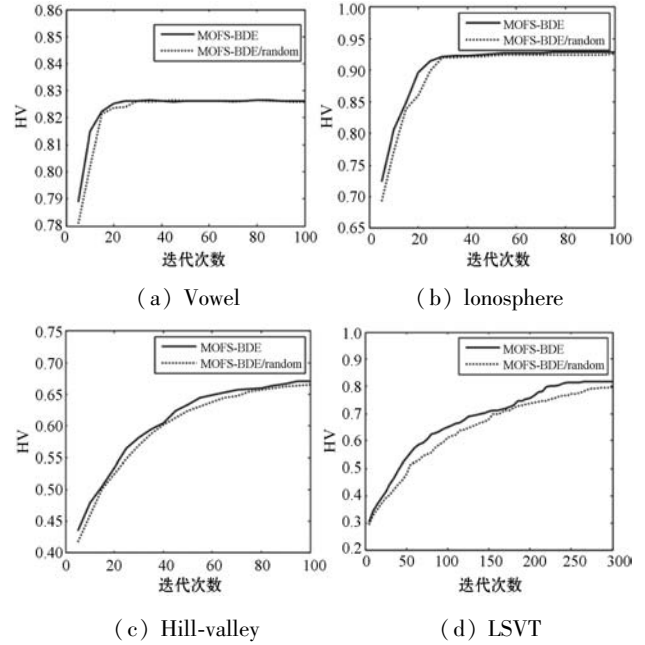


图 2 HV 曲线

表 1 测试数据集的基本信息

数据集	样本个数	类别数	特征数
Vowel	990	11	10
Wine	178	3	14
Zoo	101	7	17
Vehicle	846	4	18
WDBC	569	2	32
Ionosphere	351	2	34
Satellite	856	7	37
SPECTF	160	2	44
Parkinson	240	2	46
Sonar	208	2	60
Libras Movement	360	15	90
Hill-valley	606	2	100
Urban land cover(ULC)	507	9	148
Musk	476	2	167
SCADI	70	7	206
LSVT	126	2	309

续表 1

数据集	样本个数	类别数	特征数
CNAE-9	540	9	857
Yale_64	165	15	1 024
SRBCT	83	4	2 308
DLBCL	45	2	5 469

表 2 比较算法的参数

算法	相关参数值
NSGAFS	突变率 = $1/D$, 交叉概率 $\gamma = 0.9$
DEMOFS	交叉率 $CR = 0.3$, 比例因子 $F = 0.5$
M 净化搜索算子 OFS	惯性权重 $w = 0.1 + 0.4 \times \text{rand}$, 加速度常数, $c_1 = c_2 = 1.5 + 0.5 \times \text{rand}$, 突变率 $1/D$, 存档大小 = 50
B-MOABCFS	群体大小 $N = 50$, 食物来源数量 = $N/2$, 限制实验参数 = 5
MOEA/D-2TMFI	变异率 = $1/D$, 交叉率 $CR = 0.3$, 邻域选择概率 = 0.9, 尺度因子 $F = \text{rand}$
MOFS-BDE	比例因子 $F = 0.5 \times \text{rand}$, 基础交叉概率 $CR = 0.3$, 参数 $T_{loc} = 5$, 湍流系数 $\sigma = 0.01$

表 3 MOFS-BDE、无净化搜索算子的 MOFS-BDE 和 DEMOFS 的平均 HV 值

算法	Vowel	Vehicle	Ionosphere	Sonar	Hill-valley	ULC	LSVT	CNAE-9
MOFS-BDE	0.826 2	0.696 4	0.930 0	0.917 5	0.668 1	0.887 3	0.816 3	0.838 1
DEMOFS	0.826 0	0.689 4	0.929 6	0.903 8	0.651 7	0.878 8	0.751 0	0.811 3
无净化搜索算子的 MOFS-BDE	0.826 4	0.693 6	0.928 6	0.910 5	0.659 0	0.883 3	0.797 5	0.820 9

4.3 参数敏感性分析

在本文算法中,两个参数,即运行频率 T_{loc} , 和湍流系数 σ 都起着重要的作用。 T_{loc} 的取值范围为 $\{1, 3, 5, 7, 9\}$, σ 的取值范围为 $\{0.001, 0.005, 0.01, 0.02, 0.05, 0.1\}$ 。表 4 统计了具有不同 T_{loc} 值的 MOFS-BDE 的 HV 值,显然,对于两个简单的数据集, Vowel 和 Ionosphere, 当 T_{loc} 值从 1 到 9 变化时,该算法得到了较好的 HV 值。对于另外两个数据集 Hill-valley 和 LSVT, 只有当 T_{loc} 值从 1 变为 5 时,才得到满意的 HV 值。由于频繁地实施净化搜索算法会增加算法的计算量,因此将 T_{loc} 设为 5。

表 5 统计了不同 σ 值的 MOFS-BDE 的 HV 值。对于 Vowel 和 Ionosphere 这两个数据集,当 σ 值在 0.001 到 0.05 之间变化时,该算法得到了满意的 HV 值。然而,对于 Hill-valley 和 LSVT 这两个特征更丰富的数据集,当 σ 值大于 0.01 时,其性能会显著下降。因此, σ 的最佳取值范围为 $[0.001, 0.01]$ 。

表 4 本文算法在不同 T_{loc} (平均值/标准值)下得到的 HV 值

数据集	$T_{loc} = 1$	$T_{loc} = 3$	$T_{loc} = 5$	$T_{loc} = 7$	$T_{loc} = 9$
Vowel	0.826 4/ 0.000 4	0.826 2/ 0.000 5	0.826 2/ 0.000 4	0.826 0/ 0.000 3	0.826 1/ 0.000 5
Ionosphere	0.929 3/ 0.002 0	0.930 3/ 0.001 6	0.930 0/ 0.003 0	0.929 8/ 0.003 4	0.929 7/ 0.002 1
Hill-valley	0.666 2/ 0.002 4	0.669 2/ 0.003 5	0.668 1/ 0.002 9	0.661 2/ 0.003 9	0.660 1/ 0.004 0
BHLSVT	0.813 4/ 0.027 2	0.819 7/ 0.022 8	0.816 3/ 0.027 5	0.810 4/ 0.048 7	0.808 4/ 0.048 7

表 5 用不同的 σ (平均值/标准值)计算得到的 HV 值

数据集	$\sigma = 0.001$	$\sigma = 0.005$	$\sigma = 0.01$	$\sigma = 0.02$	$\sigma = 0.05$	$\sigma = 0.1$
Vowel	0.826 0/ 0.000 3	0.826 2/ 0.000 4	0.826 2/ 0.000 4	0.826 0/ 0.000 5	0.825 3/ 0.000 6	0.822 8/ 0.000 7
Ionosphere	0.929 4/ 0.002 8	0.929 6/ 0.002 8	0.930 0/ 0.003 0	0.930 1/ 0.003 2	0.929 0/ 0.003 5	0.927 5/ 0.004 3
Hill-valley	0.668 0/ 0.005 7	0.668 9/ 0.002 7	0.668 1/ 0.002 9	0.661 4/ 0.003 1	0.660 7/ 0.003 9	0.653 1/ 0.006 8
LSVT	0.815 5/ 0.024 1	0.816 7/ 0.022 8	0.816 3/ 0.027 5	0.810 1/ 0.029 1	0.779 7/ 0.032 1	0.704 9/ 0.037 6

4.4 对最佳 Pareto 解集的分析

为了减少特征选择结果的随机性,本文对每个数据集分别将这些算法运行 30 次。从 30 个结果中,选择最佳结果。图 3 - 图 4 记录了基于八个数据集,六种算法的运行结果。如图 3 所示,对于具有小特征、Vowel 和 Vehicle 的数据集, MOFS-BDE、DEMOFS、M 净化搜索算子 OFS 和 B-MOABCFS 算法的结果相同,明显优于 NSGAFS。MOEA/D-2TMFI 的结果也略差于 MOFS-BDE、DEMOFS、M 净化搜索算子 OFS 和 B-MOABCFS。特别是对于 Vehicle 数据集,当选择两个以上的特征时,这五种算法的错误率都小于 NSGAFS 算法。因此,这两个数据集实际上对 MOFS-BDE、DEMOFS、M 净化搜索算子 OF、B-MOABCFS 和 MOEA/D-2TMFI 来说均能够实现较好的特征选择。

对于具有 34 个特征的 Ionosphere 数据集, MOFS-BDE 和 MOEA/D-2TMFI 算法的结果相同,明显优于 NSGAFS 和 M 净化搜索算子 OFS。对于 MOFS-BDE、DEMOFS、B-MOABCFS 和 MOEA/D-2TMFI, 当所选特征种类从 1 增加到 3 时,它们的分类错误率都从 20% 下降到 8%。然而,当所选特征种类大于 3 时, B-MOABCFS 和 DEMOF 仍然有两种解集,这两种解集主要是由 MOFS-BDE 和 MOEA/D-2TMFI 算法构成。在九个特征情况下, MOFS-BDE 和 MOEA/D-2TMFI 的错误率最小,为 2.56%。

对于包含 60 种特征的数据集, Sonar、MOFS-BDE

在错误率和所选特征的种类方面均优于 DEMOFS、M 净化搜索算子 OFS 和 NSGAFS。尽管 B-MOABCFS 和 MOEA/D-2TMFI 的解集比 MOFS-BDE 的更好,但当选定 6 种特征时,它们剩下的大多数解集都由 MOFS-BDE 算法构成。当选择 13、16 和 18 种特征时,DEMOFS、M 净化搜索算子 OFS 和 NSGAFS 的错误率最小,分别为 6.25%、7.21% 和 6.73%。在六种比较算法中,当选取 10 种特征时,MOFS-BDE 的错误率最小,为 4.81%。B-MOABCFS 的错误率也是最小的,4.81%,但它需要 24 种特征。此外,当特征数少于 10 类时,NSGAFS 无法获得任何解集。

如图 4 所示,对于具有 100 类特征的数据集, Hill valley, MOFS-BDE 在错误率和所选特征数方面都比 DEMOFS、M 净化搜索算子 OFS、NSGAFS、B-MoABC 和 MOEA/D-2TMFI 的性能更好。当选择 19 种特征时, DEMOFS 的错误率最小,为 33.3%,比 MOFS-BDE 要高,此时 MOFS-BDE 只有 12 个特征。此外, B-MOABCFS 的错误率最小时对应 36 个特征,为 33.2%; MOEA/D-2TMFI 的错误率最小时对应 9 个特征,为 34.3%; DEMOFS 的错误率最小时有 19 个特征,为 33.3%。

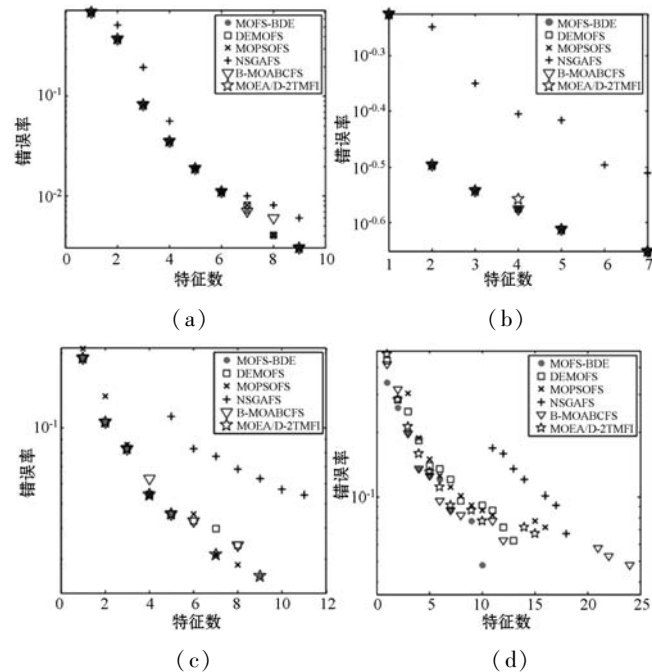


图 3 算法的运行结果

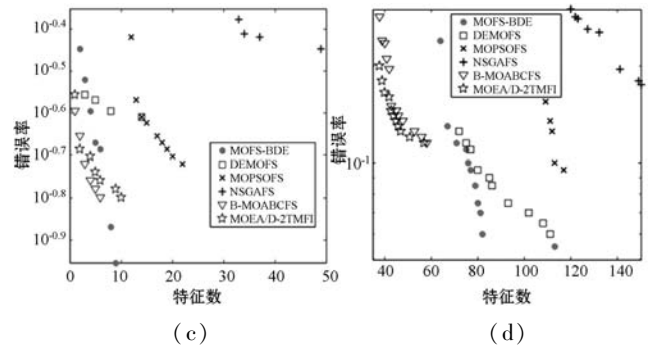
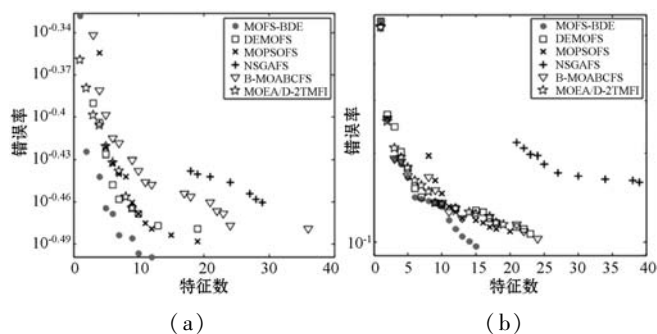


图 4 算法的运行结果

对于具有 148 类特征的数据集, ULC、MOFS-BDE 在错误率和所选特征数方面均优于 DEMOFS、M 净化搜索算子 OFS 和 NSGAFS。尽管 B-MOABCFS 和 MOEA/D-2TMFI 可以找到比 MOFS-BDE 性能更好的解集,但当选定特征个数为一个或两个时,它们剩下的大多数解集仍然由 MOFS-BDE 算法构造。特别是, M 净化搜索算子 OFS 和 NSGAFS 不能将数据分为七类特征。对于选定的 15 个特征, MOFS-BDE 将错误率降低到 9.66%,比 DEMOFS 得到的 23 类特征的最小错误率低 0.98 百分点,比 M 净化搜索算子 OFS 得到的 20 个特征的最小错误率低 1.16 百分点,比 NSGAFS 得到的 39 个特征的最小错误率低 6.32 百分点,比 B-MOABCFS 得到的 24 个特征的最小错误率低 0.60 百分点,比 B-MOEA/D2TMFI 得到的 19 个特征的最小错误率低 1.78 百分点。

对于具有 309 类特征的数据集, LSVT、MOFS-BDE 在错误率和所选特征数方面都优于 M 净化搜索算子 OFS 和 NSGAFS。尽管 DEMOFS 找到了一个比 MOFS-BDE 性能更好的解集,但当所选特征的个数为 2 时,其剩余的解集都由 MOFS-BDE 算法构造。当所选特征的种类小于 7 时, B-MOABCFS 和 MOEA/D-2TMFI 能识别出比 MOFS-BDE 更好的特征子集。然而,当选择更多的特征时,它们的错误率均高于 15.87%。当特征个数为 9 时, MOFS-BDE 的错误率最低,为 11.11%,比 MOABCFS 和 MOEA/D-2TMFI 的错误率低 4.76 百分点。

对于具有 857 个特征的数据集 CNAE-9, MOFS-BDE 得到一个性能较好的 Pareto 解,可以影响 DEMOFS、M 净化搜索算子 OFS 和 NSGAFS 获得的所有解。当所选特征种类小于 60 个时, B-MOABCFS 和 MOEA/D-2TMFI 都能定位非支配解,当总共选择 58 个特征时,错误率最低,为 11.50%。虽然 MOFS-BDE 在选择 64 个特征之前确定了第一个非主导解决方案,但选择 113 个特征时可以帮助其将错误率降低到 5.50%,这比 B-MOABCFS 和 MOEA/D-2TMFI 获得 58 个特征

的最小错误率低 6.0 百分点,比 M 净化搜索算子 OFS 获得 117 个特征的最小错误率低 4.0 百分点,比 NSGAFS 获得 150 个特征时的最小错误率低 12 百分点。当获得 111 个特征时,DEMOFS 的错误率最低,为 6.0%,而与 MOFS-BDE 获得 82 个特征的错误率相同。

一般来说,MOFS-BDE、DEMOFS、M 净化搜索算子 OFS、B-MOABCFS 和 MOEA/D-2TMFI 在处理少于 40 个特征的数据集时性能相当。然而,对于 Sonar、Hill valley 和 ULC 数据集,MOFS-BDE 在错误率和所选特

征的数量方面都比其他五种算法的性能更好。对于特征较多的数据集 LSVT 和 CNAE-9,MOFS-BDE 具有很好的降低分类错误率的能力,MOFS-BDE 得到的最小错误率明显低于 B-MOABCFS 和 MOEA/D-2TMFI 得到的最小错误率。

4.5 平均性能分析

本节比较了 MOFS-BDE、DEMOFS、M 净化搜索算子 OFS、NSGAFS、B-MOABCFS 和 MOEA/D-2TMFI 算法分别独立运行 30 次的平均性能。表 6 和表 7 分别给出了这六种算法相对于 SC 和 FN 的平均性能。

表 6 MOFS-BDE 和其他五种算法(平均值/标准值)之间的集合覆盖值

数据集	算法	SC(MOFS-BDE, *)	SC(*, MOFS-BDE)	数据集	算法	SC(MOFS-BDE, *)	SC(*, MOFS-BDE)
Vowel	DEMOFS	0.044 4/0.030 4	0/0	Libras Movement	DEMOFS	0.543 2/0.209 5	0.348 4/0.166 4
	M 净化搜索算子 OFS	0/0	0/0		M 净化搜索算子 OFS	0.784 0/0.234 6	0.155 8/0.181 2
	NSGAFS	0.885 1/0.111 7	0/0		NSGAFS	1/0	0/1
	B-MOABCFS	0.022 2/0.024 8	0/0		B-MOABCFS	0.365 4/0.217 4	0.276 5/0.201 3
	MOEA/D-2TMFI	0.057 1/0.078 2	0/0		MOEA/D-2TMFI	0.492 1/0.219 9	0.377 6/0.200 7
Wine	DEMOFS	0/0	0.052 0/0.109 6	Hill-valley	DEMOFS	0.949 2/0.070 5	0.067 7/0.039 5
	M 净化搜索算子 OFS	0.028 6/0.090 4	0/0		M 净化搜索算子 OFS	0.849 2/0.093 0	0.155 6/0.111 7
	NSGAFS	0.912 4/0.154 5	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.091 7/0.168 2	0/0		B-MOABCFS	0.942 9/0.053 1	0.064 7/0.060 9
	MOEA/D-2TMFI	0.066 7/0.140 5	0/0		MOEA/D-2TMFI	0.924 4/0.073 5	0.047 2/0.064 8
Zoo	DEMOFS	0/0	0/0	ULC	DEMOFS	0.548 2/0.136 5	0.282 4/0.073 6
	M 净化搜索算子 OFS	0.056 7/0.091 7	0/0		M 净化搜索算子 OFS	0.981 8/0.040 6	0/0
	NSGAFS	1/0	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.056 7/0.091 7	0/0		B-MOABCFS	0.661 3/0.141 5	0.202 7/0.097 6
	MOEA/D-2TMFI	0.020 0/0.063 2	0/0		MOEA/D-2TMFI	0.498 3/0.119 9	0.248 4/0.074 5
Vehicle	DEMOFS	0.033 3/0.037 2	0/0	Musk	DEMOFS	0.723 5/0.226 5	0.191 8/0.175 8
	M 净化搜索算子 OFS	0.066 7/0.074 5	0/0		M 净化搜索算子 OFS	1/0	0/0
	NSGAFS	0.933 3/0.074 5	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.053 3/0.071 2	0/0		B-MOABCFS	0.525 1/0.286 1	0.262 1/0.168 9
	MOEA/D-2TMFI	0/0	0/0		MOEA/D-2TMFI	0.804 1/0.177 7	0.142 6/0.150 3
WDBC	DEMOFS	0.193 2/0.145 8	0.140 5/0.137 8	SCADI	DEMOFS	0.529 9/0.272 8	0.244 9/0.215 0
	M 净化搜索算子 OFS	0.447 7/0.252 3	0.184 7/0.147 5		M 净化搜索算子 OFS	0.795 0/0.237 0	0.059 6/0.123 4
	NSGAFS	1/0	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.500 6/0.183 2	0.132 9/0.145 2		B-MOABCFS	0.505 1/0.204 6	0.229 4/0.204 0
	MOEA/D-2TMFI	0.461 9/0.138 0	0.053 6/0.069 4		MOEA/D-2TMFI	0.725 3/0.215 1	0.113 6/0.158 3
Ionosphere	DEMOFS	0.355 7/0.083 4	0.200 7/0.084 4	LSVT	DEMOFS	0.720 0/0.198 1	0.171 4/0.183 3
	M 净化搜索算子 OFS	0.392 4/0.102 1	0.153 6/0.071 2		M 净化搜索算子 OFS	1/0	0/0
	NSGAFS	1/0	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.563 1/0.137 0	0.153 6/0.072 5		B-MOABCFS	0.633 3/0.280 1	0.233 3/0.152 8
	MOEA/D-2TMFI	0.470 7/0.157 6	0.207 1/0.086 5		MOEA/D-2TMFI	0.516 7/0.258 0	0.309 3/0.224 9
Satellite	DEMOFS	0.325 3/0.205 2	0.251 1/0.154 6	CNAE-9	DEMOFS	0.634 9/0.140 9	0.180 0/0.095 7
	M 净化搜索算子 OFS	0.641 6/0.138 6	0.099 9/0.109 3		M 净化搜索算子 OFS	1/0	0/0
	NSGAFS	1/0	0/1		NSGAFS	1/0	0/0
	B-MOABCFS	0.518 2/0.140 4	0.151 5/0.116 4		B-MOABCFS	0/0	0.244 8/0.102 4
	MOEA/D-2TMFI	0.470 5/0.186 2	0.164 8/0.147 9		MOEA/D-2TMFI	0/0	0.313 9/0.172 3

续表 6

数据集	算法	SC(MOFS-BDE, *)	SC(*, MOFS-BDE)	数据集	算法	SC(MOFS-BDE, *)	SC(*, MOFS-BDE)
SPECTF	DEMOFS	0.115 5/0.111 5	0.073 4/0.089 6	Yale_64	DEMOFS	1/0	0/0
	M 净化搜索算子 OFS	0.257 0/0.162 6	0.058 8/0.080 9		M 净化搜索算子 OFS	1/0	0/0
	NSGAFS	1/0	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.186 8/0.135 9	0.056 3/0.076 2		B-MOABCFS	0/0	0.853 8/0.139 7
Parkinson	MOEA/D-2TMFI	0.336 6/0.172 2	0.042 6/0.064 5	SRBCT	MOEA/D-2TMFI	1/0	0/0
	DEMOFS	0.468 6/0.220 5	0.254 0/0.197 2		DEMOFS	0.937 5/0.139 8	0/0
	M 净化搜索算子 OFS	0.657 8/0.224 7	0.139 4/0.098 5		M 净化搜索算子 OFS	1/0	0/0
	NSGAFS	1/0	0/1		NSGAFS	1/0	0/0
Sonar	B-MOABCFS	0.508 3/0.215 7	0.188 3/0.137 6	DLBCL	B-MOABCFS	1/0	0/0
	MOEA/D-2TMFI	0.460 2/0.193 4	0.208 4/0.174 1		MOEA/D-2TMFI	1/0	0/0
	DEMOFS	0.627 0/0.085 2	0.279 5/0.112 5		DEMOFS	0.949 1/0.105 8	0/0
	M 净化搜索算子 OFS	0.830 2/0.072 6	0.030 8/0.034 4		M 净化搜索算子 OFS	1/0	0/0
	NSGAFS	1/0	0/0		NSGAFS	1/0	0/0
	B-MOABCFS	0.617 1/0.215 2	0.243 6/0.103 2		B-MOABCFS	1/0	0/0
	MOEA/D-2TMFI	0.487 9/0.219 4	0.276 6/0.111 5		MOEA/D-2TMFI	1/0	0/0

表 7 基于 20 个数据集的六种算法 FN 值

数据集	MOFS-BDE	DEMOFS	M 净化搜索算子 OFS	NSGAFS	B-MOABCFS	MOEA/D-2TMFI
Vowel	9.00/0	9.00/0	8.60/0.55	8.20/0.13	9.00/0	8.00/0
Wine	5.85/0.36	5.45/0.51	5.80/0.63	5.80/0.78	4.50/1.43	5.20/0.42
Zoo	5.80/0.41	5.75/0.44	5.60/0.51	5.10/1.56	5.50/0.52	5.40/0.51
Vehicle	5.80/0.45	5.40/0.55	5.60/0.55	3.80/0.16	5.20/0.45	5.10/0.32
WDBC	7.60/0.67	7.30/0.48	7.80/1.14	5.00/1.70	7.30/1.34	6.70/0.48
Ionosphere	7.56/0.53	7.56/0.73	6.40/0.89	5.20/1.31	7.45/1.05	6.65/0.87
Satellite	13.00/1.49	12.70/1.41	12.10/1.52	8.20/2.30	11.22/0.97	10.22/1.48
SPECTF	10.05/0.68	9.00/0.72	10.30/0.48	8.20/1.75	10.00/0.56	10.30/0.67
Parkinson	9.58/1.08	9.40/1.17	9.90/1.37	5.80/1.98	8.60/1.17	8.70/1.15
Sonar	11.20/2.28	10.78/1.99	11.00/2.73	5.80/1.30	10.70/1.75	10.00/1.83
Libras Movement	12.10/0.99	10.75/1.48	10.90/1.79	5.67/2.29	10.15/1.29	9.70/1.05
Hill-valley	8.80/0.45	9.60/1.67	8.00/2.64	6.80/2.17	9.75/2.63	8.30/1.820
ULC	14.20/2.15	12.20/2.39	11.00/1.20	9.20/2.17	12.05/2.37	10.95/1.633
Musk	15.0/2.50	12.80/1.55	9.16/1.17	7.88/3.26	10.83/2.48	13.67/1.86
SCADI	6.50/0.97	6.40/1.07	4.10/0.73	3.20/1.81	5.60/1.17	5.90/0.87
LSVT	6.15/1.22	4.60/0.89	5.15/3.17	2.80/0.84	4.15/1.14	4.25/1.80
CNAE-9	9.80/2.17	7.65/1.95	5.20/0.84	7.40/2.70	9.70/3.11	8.15/1.84
Yale_64	7.33/1.52	5.37/1.59	3.14/1.46	5.33/1.52	15.50/2.64	4.25/1.50
SRBCT	4.12/1.07	3.20/0.91	2.57/0.78	3.70/1.88	1.58/0.72	1.86/1.05
DLBCL	2.07/0.57	2.67/0.57	1.16/0.41	2.69/0.53	4.16/0.75	1.42/0.55

对于少于 20 个特征 (Vowel、Wine、Zoo 和 Vehicle) 的数据集, MOFSBDE、DEMOFS、M 净化搜索算子 OFS、B-MOABCFS 和 MOEA/D-2TMFI 得到的最优解集具有

相似的收敛特性。以数据集 Wine 为例, 在最坏情况下, MOFS-BDE 的解集中只有 5.20% 受 DEMOFS 所控制, 而 B-MOABCFS 的解集中有 9.17% 被 MOFS-BDE 所控制。NSGAFS 得到的最优解集收敛性最差。对于四个数据集, 超过 88.5% 的 NSGAFS 解集由 MOFS-BDE 解集控制。此外, MOFS-BDE 在解集多样性方面比 M 净化搜索算子 OFS、NSGAFS 和 MOEA/D-2TMFI 具有更好的性能, 其产生的最优解集数量比 M 净化搜索算子 OFS、NSGAFS 和 MOEA/D-2TMFI 多。

对于所有具有 30 到 500 个特征的 12 个数据集, MOFS-BDE 在收敛性方面优于 DEMOFS、M 净化搜索算子 OFS、NSGAFS、B-MOABCFS 和 MOEA/D-2TMFI。五种算法中 MOFS-BDE 占主要地位的比例明显高于后者。以数据集 LSVT 为例, MOFS-BDE 占 DEMOFS 解集的 72.0%, M 净化搜索算子 OFS 和 NSGAFS 解的全部, B-MOABCFS 解的 63.33%, MOEA/D-2TMFI 解的 51.67%。在最坏的情况下, 只有 30.93% 的 MOFS-BDE 解集由 MOEA/D-2TMFI 控制。此外, 对于 12 个数据集集中的 8 个数据集, 即 Ionosphere、Satellite、Sonar、Libras Movement、ULC、Musk、SCADI 和 LSVT, MOFS-BDE 在解集多样性方面也表现出比 DEMOFS、M 净化搜索算子 OFS、NSGAFS、B-MOABCFS 和 MOEA/D-2TMFI 更好的性能。

对于具有 857 个特征的数据集, 即 CNAE-9, MOFS-BDE 无法找到一个解集支配 BMOABCFS 和 MOEA/D-2TMFI 获得的解集。它们的 SC 值表明, MOFS-BDE 解集中, B-MOABCFS 和 MOEA/D-2TMFI 分别占 24.48% 和 31.39%。B-MOABCFS 和 MOEA/D-2TMFI 的 SC 性

能之所以优异,主要是因为它们具有快速降维的能力。如图 6 所示,B-MOABCFS 和 MOEA/D-2TMFI 的每个解集中的特征数明显少于 MOFS-BDE。然而,过快地减少特征可能会牺牲 B-MOABCFS 和 MOEA/D-2TMFI 的分类精度性能。MOFS-BDE 得到的最小错误率比 B-MOABCFS 和 MOEA/D-2TMFI 低 6 个百分点。此外,MOFS-BDE 的 FN 值较大,为 9.8,而 B-MOABCFS 和 MOEA/D-2TMFI 的 FN 值分别为 9.7 和 8.15。

在具有 1 024 个特征的数据集 Yale_64 中,MOFS-BDE 在收敛性和多样性方面的性能仅次于 DEMOFS、M 净化搜索算子 OFS、NSGAFS 和 MOEA/D-2TMFI。由于 B-MOABCFS 在 scout-bee 阶段具有显著的变异性能,因此 B-MOABCFS 能够获得最佳的收敛性和解集的多样性。

对于 2 308 个以上特征的数据集,SRBCT、B-MOABCFS 的收敛性最好,优于其他五种算法。NSGAFS、M 净化搜索算子 OFS、B-MOABCFS 和 MOEA/D-2TMFI 得到的所有解集均以 MOFS-BDE 为主,DEMOFS 得到的 93.75% 的解集以 MOFS-BDE 为主。此外,MOFS-BDE 在解集多样性方面表现最佳,如表 7 所示。

对于具有 5 000 个以上特征的数据集,DLBCL,虽然 B-MOABCFS 比 MOFS-BDE 得到了更多的解,但它们的 SC 值表明这些解的收敛性比 DEMOFS 要好。

此外,还利用 t 检验分析了 MOFS-BDE 的鲁棒性。

这是一个具有两个样本的位置检验,用于检验两个群体平均数相等的假设。本文将显著性水平设为 0.05,这里使用 HV 指标,表 8 给出了关于 HV 的六种算法的结果。“Y+”表示 MOFS-BDE 在 0.05 水平上显著优于双尾检验选择的算法,“Y-”表示 MOFS-BDE 在 0.05 水平上显著优于双尾检验选择的算法,“N”表示这两种算法之间的差异不显著在 0.05 水平。可以发现,对于 Vowel、Wine、Zoo 和 Vehicle 这四个少于 20 个特征的数据集,MOFS-BDE 可以得到与 DEMOFS、M 净化搜索算子 OFS、B-MOABCFS 和 MOEA/D-2TMFI 相似的结果。然而,对于其余 16 个数据集中的 8 个数据集,即 Parkinson、Sonar、Hill-valley、ULC、Musk、LSVT、SRBCT 和 DLBCL,MOFS-BDE 的 HV 指标性能在 0.05 水平上明显优于 DEMOFS、M 净化搜索算子 OFS、NSGAFS、B-MOABCFS 和 MOEA/D-2TMFI。对于 16 个数据集中的 4 个数据集,即 WDBC、Satellite、Libras Movement 和 SCADI,MOFS-BDE 得到的 HV 值与 DEMOFS 的相当,其 HV 值在 0.05 水平上显著优于 M 净化搜索算子 OFS、NSGAFS、B-MOABCFS 和 MOEA/D-2TMFI。对于 16 个数据集中的两个数据集,即 Ionosphere 和 SPECTF,MOFS-BDE 也获得了与其他算法相比具有竞争力的 HV 值。在 20 个数据集中,只有一个数据集,即 Yale_64,MOFS-BDE 的 HV 值比 B-MOABCFS 的 HV 值差得多(0.05)。

表 8 六种算法基于 20 个数据集的 HV 值

数据集	MOFS-BDE 均值/标准值	DEMOFS		M 净化搜索算子 OFS		NSGAFS		B-MOABCFS		MOEA/D-2TMFI	
		均值/标准值	t-检验	均值/标准值	t-检验	均值/标准值	t-检验	均值/标准值	t-检验	均值/标准值	t-检验
Vowel	0.826 2/0.000 4	0.826 0/0.000 7	N	0.826 7/0.000 6	N	0.809 9/0.037 2	N	0.825 9/0.000 4	N	0.825 2/0.000 5	N
Wine	0.883 5/0.002 2	0.884 9/0.001 9	N	0.884 5/0.002 1	N	0.806 0/0.048 4	Y+	0.884 1/0.007 6	N	0.883 0/0.002 0	N
Zoo	0.887 0/0.002 1	0.886 8/0.002 2	N	0.885 6/0.002 6	N	0.782 2/0.042 0	Y+	0.885 2/0.002 9	N	0.885 0/0.002 6	N
Vehicle	0.696 4/0.005 8	0.689 4/0.006 2	N	0.692 6/0.006 3	N	0.596 1/0.036 7	Y+	0.684 6/0.006 7	N	0.687 1/0.004 2	N
WDBC	0.943 3/0.000 8	0.942 5/0.000 6	N	0.941 5/0.001 6	Y+	0.753 7/0.061 0	Y+	0.939 7/0.003 0	Y+	0.938 2/0.002 7	Y+
Ionosphere	0.930 0/0.003 0	0.929 6/0.003 4	N	0.927 0/0.003 7	N	0.837 4/0.098 1	Y+	0.921 8/0.003 4	N	0.923 8/0.005 1	N
Satellite	0.870 4/0.002 2	0.869 6/0.002 7	N	0.864 5/0.003 5	Y+	0.688 5/0.038 1	Y+	0.865 3/0.004 1	Y+	0.861 8/0.005	Y+
SPECTF	0.688 7/0.000 9	0.664 2/0.019 4	Y+	0.687 5/0.002 6	N	0.585 8/0.030 6	Y+	0.687 8/0.002 1	N	0.686 8/0.002 3	Y+
Parkinson	0.892 8/0.002 2	0.886 5/0.008 3	Y+	0.882 9/0.006 4	Y+	0.714 9/0.035 4	Y+	0.884 6/0.004 7	Y+	0.883 6/0.007 7	Y+
Sonar	0.917 5/0.004 3	0.903 8/0.005 5	Y+	0.885 1/0.016 1	Y+	0.669 3/0.021 8	Y+	0.904 9/0.004 5	Y+	0.903 1/0.101 0	Y+
Libras Movement	0.870 9/0.003 3	0.865 5/0.003 2	N	0.854 7/0.012 3	Y+	0.612 3/0.026 7	Y+	0.861 2/0.005 0	Y+	0.859 6/0.005 8	Y+
Hill-valley	0.668 1/0.002 9	0.651 7/0.003 5	Y+	0.650 1/0.006 2	Y+	0.440 7/0.015 9	Y+	0.638 4/0.018 4	Y+	0.639 2/0.004 8	Y+
ULC	0.887 3/0.004 0	0.878 8/0.005 8	Y+	0.830 6/0.010 1	Y+	0.577 1/0.025 4	Y+	0.867 9/0.008 8	Y+	0.870 1/0.003 4	Y+
Musk	0.953 1/0.003 3	0.940 7/0.004 5	Y+	0.879 9/0.020 1	Y+	0.601 9/0.015 5	Y+	0.937 3/0.008 1	Y+	0.940 1/0.003 8	Y+
SCADI	0.911 7/0.009 4	0.909 7/0.008 8	N	0.881 1/0.012 6	Y+	0.555 9/0.013 3	Y+	0.888 1/0.017 4	Y+	0.896 1/0.012 8	Y+

续表 8

数据集	MOFS-BDE 均值/标准值	DEMOFS		M 净化搜索算子 OFS		NSGAFS		B-MOABCFS		MOEA/D-2TMFI	
		均值/标准值	t-检验	均值/标准值	t-检验	均值/标准值	t-检验	均值/标准值	t-检验	均值/标准值	t-检验
LSVT	0.866 3/0.027 5	0.751 0/0.041 4	Y+	0.742 1/0.004 2	Y+	0.396 7/0.006 4	Y+	0.815 9/0.080 9	Y+	0.798 1/0.093 7	Y+
CNAE-9	0.838 1/0.007 6	0.811 3/0.010 6	Y+	0.746 1/0.008 1	Y+	0.463 9/0.018 6	Y+	0.836 1/0.011 9	N	0.840 5/0.015 9	N
Yale_64	0.748 9/0.003 5	0.671 9/0.009 3	Y+	0.623 8/0.005 5	Y+	0.422 1/0.011 1	Y+	0.753 5/0.003 8	Y-	0.713 3/0.012 7	Y+
SRBCT	0.825 0/0.009 5	0.748 0/0.010 9	Y+	0.714 8/0.014 9	Y+	0.479 9/0.010 1	Y+	0.771 5/0.013 0	Y+	0.726 9/0.008 9	Y+
DLBCL	0.697 0/0.015 4	0.655 6/0.008 2	Y+	0.625 0/0.006 3	Y+	0.466 7/0.013 7	Y+	0.594 4/0.022 5	Y+	0.508 0/0.007 9	Y+

利用非参数统计检验(Friedman)检验验证了不同算法对所有测试问题的分类精度的相似性。根据表 8 的结果,可以得出 Friedman 检验的 p 值为 $1.609\ 9E-13$ 。 p 值越小表明所有算法在 0.05 显著性水平上的 HV 性能越相似。考虑到表 8 中的 HV 值和两个统计测试值,可以得出结论:MOFS-BDE 在特征选择问题上的性能较好。

4.6 运行时间

本研究还比较了 MOFS-BDE、DEMOFS、M 净化搜索算子 OFS、NSGAFS、B-MOABCFS 和 MOEA/D-2TMFI 的运行时间,表 9 统计了它们的平均运行时间。

表 9 六种算法基于 20 个数据集的运行时间 单位:min

数据集	MOFS-BDE	DEMOFS	M 净化搜索算子 OFS	NSGAFS	B-MOABCFS	MOEA/D-2TMFI
Vowel	7.91/0.27	8.34/0.35	7.32/0.41	8.29/0.57	7.65/0.39	6.83/0.26
Wine	0.35/0.00	0.19/0.00	0.25/0.00	0.28/0.01	0.36/0.009	0.32/0.00
Zoo	0.17/0.00	0.12/0.00	0.08/0.00	0.13/0.00	0.19/0.010	0.23/0.00
Vehicle	0.16/0.01	0.15/0.01	0.07/0.00	0.10/0.00	0.09/0.014	0.18/0.00
WDBC	3.14/0.12	3.29/0.08	2.99/0.14	4.27/0.42	5.23/0.49	2.33/0.07
Ionosphere	1.13/0.04	1.29/0.05	1.02/0.04	1.51/0.07	1.12/0.10	1.08/0.05
Satellite	9.72/0.33	9.31/0.22	8.18/0.63	11.70/0.36	14.42/0.89	7.04/0.25
SPECTF	0.38/0.00	0.31/0.00	0.25/0.00	0.33/0.00	0.51/0.01	0.38/0.00
Parkinson	0.84/0.01	0.77/0.04	0.67/0.04	1.08/0.12	1.16/0.06	0.71/0.02
Sonar	0.69/0.05	0.77/0.06	0.65/0.07	1.03/0.04	0.67/0.09	0.67/0.03
Libras Movement	3.00/0.10	2.87/0.15	2.47/0.27	4.62/0.16	3.87/0.45	1.92/0.08
Hill-valley	8.82/0.83	9.48/0.78	6.69/1.10	12.06/0.48	9.25/1.45	4.86/0.39
ULC	17.18/2.11	17.44/1.95	18.17/2.42	31.15/3.25	16.80/2.87	14.60/1.02
Musk	18.01/1.71	17.38/0.90	20.07/2.95	37.11/4.29	25.72/2.88	14.14/0.47
SCADI	0.79/0.02	0.63/0.02	0.42/0.05	1.41/0.07	0.65/0.02	0.76/0.03
LSVT	2.10/0.11	2.63/0.12	1.95/0.17	4.00/0.13	1.82/0.09	1.73/0.06
CNAE-9	20.30/0.46	20.82/0.71	18.93/1.08	33.59/1.17	19.21/1.89	15.34/0.98
Yale_64	16.54/0.36	23.67/0.63	20.34/0.97	36.41/0.96	34.17/1.51	13.99/1.33
SRBCT	16.58/0.32	17.37/0.27	13.91/0.50	22.10/0.39	30.18/0.53	20.61/0.31
DLBCL	29.45/2.69	21.30/2.65	14.58/0.69	17.80/0.29	27.97/0.45	17.45/0.85

很明显,对于大多数数据集,MOFS-BDE 比 M 净化搜索算子 OFS 或 MOEA/D-2TMFI 的运行时间更长。主要有两个原因:(1) 为了提高分类精度,MOFS-BDE 比 M 净化搜索算子 OFS 和 MOEA/D-2TMFI 选择了更多的特征。因为选择的特征数量越多,用于训练分类器的数据集的规模就越大,评价算法性能的运行时间就越长。也就是说,M 净化搜索算子 OFS 和 MOEA/D-2TMFI 以牺牲分类精度为代价,获得了较短的运行时间。(2) MOFSBDE 采用了计算复杂度较高的非支配排序策略,即 FNS,在一定程度上可以延长其运行时间。考虑到 ENS 的时间复杂度低于 FNS,MOFS-BDE 比简单数据集的 DEMOFS 消耗的时间更短。虽然 B-MOABCFS 也使用非支配排序策略,但由于选择的特征较少,因此对于某些数据集,它比 MOFS-BDE 和 DEMOFS 的时间要短。

5 结 语

为了提升群体搜索能力,加快收敛速度,提出了一种基于自学习二元差分进化的多目标特征选择方法。通过分析多个数据集的实验结果可以得出如下结论:

(1) 在最优学习策略的基础上,提出的基于概率差分的二进制变异算法能有效地提高算法的收敛性。

(2) 净化搜索算子在 MOFS-BDE 中起着重要作用,能够有效提升方法的搜索能力,从而降低错误率。提出算法虽然运行时间较长,但是能够显著提升分类精度。

(3) 在多个数据集上均体现出了提出方法相对于其他方法的优势,且整体性能明显优于其他算法,表明了该方法具有较广泛的特征选择应用前景。

参 考 文 献

[1] 张芳芳,李楠. 基于独立成分分析的 fMRI 数据分类[J].

- 计算机应用与软件,2019,36(11):107-111.
- [2] 雍菊亚,周忠眉.基于互信息的多级特征选择算法[J].计算机应用,2020,40(12):3478-3484.
- [3] 裴作飞,李兆玉,王云锋,等.基于自适应遗传算法的混合特征选择方法[J].计算机应用与软件,2020,37(8):256-259,306.
- [4] 李占山,刘兆庚,俞寅,等.量子化信息素蚁群优化特征选择算法[J].东北大学学报(自然科学版),2020,41(1):17-22.
- [5] 唐晓娜,张和生.一种混合粒子群优化遗传算法的高分影像特征优化方法[J].遥感信息,2019,34(6):113-118.
- [6] Chen K, Zhou F Y, Yuan X F. Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection [J]. *Expert Systems with Applications*,2019,128(7):140-156.
- [7] Cheraghchi F, Abualhaol I, Falcon R, et al. Modeling the speed-based vessel schedule recovery problem using evolutionary multi-objective optimization [J]. *Information Sciences*,2018,448(6):53-74.
- [8] Faris H, Mafarja M, Heidari A, et al. An efficient binary Salp swarm algorithm with crossover scheme for feature selection problems [J]. *Knowledge-Based Systems*, 2018, 154(6):43-67.
- [9] 王萌,丁志军.一种新的设备指纹特征选择及模型构建方法[J].计算机科学,2020,47(7):257-262.
- [10] 李金霞,赵志刚,李强,等.改进的局部和相似性保持特征选择算法[J].计算机科学,2020,47(S1):480-484.
- [11] Hancer E, Xue B, Zhang M J. Differential evolution for filter feature selection based on information theory and feature ranking[J]. *Knowledge-Based Systems*,2018,140(6):103-119.
- [12] 黄学雨,徐浩特,陶剑文.具有特征选择的多源自适应分类框架[J].计算机应用,2020,40(9):2499-2506.
- [13] 钟昌康.基于K近邻和粒子群优化的特征选择算法[J].现代计算机,2020(9):21-24,40.
- [14] Ma X L, Zhang Q F, Tian G D, et al. On Thebycheff decomposition approaches for multi-objective evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*,2018,22(2):226-244.
- [15] Mafarja M, Aljarah I, Heidari A, et al. Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems[J]. *Knowledge-Based Systems*, 2018,145(6):25-45.
- [16] Pan A Q, Wang L, Guo W A, et al. A diversity enhanced multi-objective particle swarm optimization[J]. *Information Sciences*,2018,436(6):441-465.
- [17] Taradeh M, Mafarja M, Heidari A, et al. An evolutionary gravitational search-based feature selection [J]. *Information Sciences*,2019,497(5):219-239.
-
- (上接第253页)
- [16] Yin W P, Schütze H, Xiang B, et al. ABCNN: Attention-based convolutional neural network for modeling sentence pairs[J]. *Transactions of the Association for Computational Linguistics*,2016,4:259-272.
- [17] 王海涛,宋文,王辉.一种基于LSTM和CNN混合模型的文本分类方法[J].小型微型计算机系统,2020,41(6):1163-1168.
- [18] 吴汉瑜,严江,黄少滨,等.用于文本分类的CNNBiLSTM-Attention混合模型[J].计算机科学,2020,47(S2):24-27,34.
- [19] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [EB]. arXiv:1609.02907, 2016.
- [20] Jedrzejowicz J, Zakrzewska M. Text classification using LDA W2V hybrid algorithm[M]//*Intelligent Decision Technologies*. Singapore: Springer,2020:227-237.
- [21] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation [J]. *Journal of machine Learning research*,2003,3:993-1022.
- [22] Maier D, Waldherr A, Miltner P, et al. Applying LDA topic modeling in communication research: Toward a valid and reliable methodology[J]. *Communication Methods and Measures*,2018,12(2-3):93-118.
- [23] Yao L, Mao C S, Luo Y. Graph convolutional networks for text classification[C]//33rd AAAI Conference on Artificial Intelligence,2019:7370-7377.
- [24] Li M S, Chen S, Chen X, et al. Actional-structural graph convolutional networks for skeleton-based action recognition [C]//*IEEE Conference on Computer Vision and Pattern Recognition*,2019:3595-3603.
- [25] Liu P F, Qiu X P, Huang X J. Recurrent neural network for text classification with multi-task learning[EB]. arXiv:1605.05101,2016.
- [26] Le Q, Mikolov T. Distributed representations of sentences and documents[C]//31st International Conference on Machine Learning,2014:1188-1196.
- [27] Joulin A, Grave E, Bojanowski P, et al. Fasttext.zip: Compressing text classification models[EB]. arXiv:1612.03651, 2016.
- [28] Tang J, Qu M, Mei Q Z. PTE: Predictive text embedding through large-scale heterogeneous text networks [C]//21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,2015:1165-1174.