软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(5):1511−1524 [doi: 10.13328/j.cnki.jos.005654]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

森林优化特征选择算法的增强与扩展*

刘兆赓[1,4], 李占山[1,2,4], 王 丽[3], 王 涛[3], 于海鸿[2,4]

[1](吉林大学 软件学院,吉林 长春 130012)

[2](吉林大学 计算机科学与技术学院,吉林 长春 130012)

[3](长春工业大学 计算机科学与工程学院,吉林 长春 130012)

[4](符号计算与知识工程教育部重点实验室(吉林大学),吉林 长春 130012)

通讯作者: 于海鸿, E-mail: yuhh@jlu.edu.cn

摘 要: 特征选择作为一种重要的数据预处理方法,不但能解决维数灾难问题,还能提高算法的泛化能力.各种各样的方法已被应用于解决特征选择问题,其中,基于演化计算的特征选择算法近年来获得了更多的关注并取得了一些成功.近期研究结果表明,森林优化特征选择算法具有更好的分类性能及维度缩减能力.然而,初始化阶段的随机性、全局播种阶段的人为参数设定,影响了该算法的准确率和维度缩减能力;同时,算法本身存在着高维数据处理能力不足的本质缺陷.从信息增益率的角度给出了一种初始化策略,在全局播种阶段,借用模拟退火控温函数的思想自动生成参数,并结合维度缩减率给出了适应度函数;同时,针对形成的优质森林采取贪心算法,形成一种特征选择算法 EFSFOA(enhanced feature selection using forest optimization algorithm).此外,在面对高维数据的处理时,采用集成特征选择的方案形成了一个适用于 EFSFOA 的集成特征选择框架,使其能够有效处理高维数据特征选择问题.通过设计对比实验,验证了 EFSFOA 与 FSFOA 相比在分类准确率和维度缩减率上均有明显的提高,高维数据处理能力更是提高到了 100 000 维.将 EFSFOA 与近年来提出的比较高效的基于演化计算的特征选择方法进行对比,EFSFOA 仍具有很强的竞争力.

关键词: enhanced feature selection using forest optimization algorithm(EFSFOA);高维;特征选择;演化计算

中图法分类号: TP18

中文引用格式: 刘兆赓,李占山,王丽,王涛,于海鸿.森林优化特征选择算法的增强与扩展.软件学报,2020,31(5):1511−1524. http://www.jos.org.cn/1000-9825/5654.htm

英文引用格式: Liu ZG, Li ZS, Wang L, Wang T, Yu HH. Enhancement and extension of feature selection using forest optimization algorithm. Ruan Jian Xue Bao/Journal of Software, 2020,31(5):1511−1524 (in Chinese). http://www.jos.org.cn/1000-9825/5654.htm

Enhancement and Extension of Feature Selection Using Forest Optimization Algorithm

LIU Zhao-Geng[1,4], LI Zhan-Shan[1,2,4], WANG Li[3], WANG Tao[3], YU Hai-Hong[2,4]

[1](College of Software, Jilin University, Changchun 130012, China)

[2](College of Computer Science and Technology, Jilin University, Changchun 130012, China)

[3](College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China)

[4](Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012, China)

* 基金项目: 国家自然科学基金(61672261); 吉林省自然科学基金(20180101043JC); 吉林省发改委产业技术研究与开发专项资金(2019C053-9)

Foundation item: National Natural Science Foundation of China (61672261); Natural Science Foundation of Jilin Province (20180101043JC); Industrial Technology Research and Development Special Project of Jilin Province Development and Reform Commission (2019C053-9)

收稿时间: 2018-07-12; 修改时间: 2018-08-05; 采用时间: 2018-09-06

Abstract: As an important data preprocessing method, feature selection can not only solve the dimensionality disaster problem, but also improve the generalization ability of algorithms. A variety of methods have been applied to solve feature selection problems, where evolutionary computation techniques have recently gained much attention and shown some success. Recent study has shown that feature selection using forest optimization algorithm has better classification performance and dimensional reduction ability. However, the randomness of initialization phase and the artificial parameter setting of global seeding phase affect the accuracy and the dimension reduction ability of the algorithm. At the same time, the algorithm itself has the essential defect of insufficient high-dimensional data processing capability. In this study, an initialization strategy is given from the perspective of information gain rate, parameter is automatically generated by using simulated annealing temperature control function during global seeding, a fitness function is given by combining dimension reduction rate, using greedy algorithm to select the best tree from the high-quality forest obtained, and a feature selection algorithm EFSFOA (enhanced feature selection using forest optimization algorithm) is proposed. In addition, in the face of high-dimensional data processing, ensemble feature selection scheme is used to form an ensemble feature selection framework suitable for EFSFOA, so that it can effectively deal with the problem of high-dimensional data feature selection. Through designing some contrast experiments, it is verified that EFSFOA has significantly improved classification accuracy and dimensionality reduction rate compared with FSFOA, and the high-dimensional data processing capability has been increased to 100 000 dimensions. Comparing EFSFOA with other efficient evolutionary computation for feature selection approaches which have been proposed in recent years, EFSFOA still has strong competitiveness.

Key words: enhanced feature selection using forest optimization algorithm (EFSFOA); high-dimensional; feature selection; evolutionary computation

特征选择是从原始特征中选择出一些最有效特征以降低数据维度的过程[1],是模式识别中重要的数据预处理方法,是数据发掘中频繁使用的数据处理技术,也是提高机器学习算法性能的一个重要手段.

针对不同的选择策略,可以将特征选择方法大致分为包裹式、过滤式和嵌入式 3 种.

- 包裹式方法:依赖于预定义的学习算法来评估所选特征的质量.根据给定的一个具体学习算法,典型的包裹式方法将执行两个步骤:(1) 搜索特征子集;(2) 评估选择的特征.重复步骤(1)和步骤(2),直到满足停止条件或获得了所需的学习表现后结束.由于包裹式方法在评估选择的特征时能够针对不同问题选择不同的评价方法,因此目前这类方法的研究更多关注于评价方法的使用和改进上面,如基于 K 近邻的包裹式方法[2]、基于支持向量机的包裹式方法[3]、基于决策树的包裹式方法[4]等.
- 过滤式方法:不依赖于任何学习算法,仅依靠对数据的某些特征进行评估,以评估特征的重要性.典型的过滤式方法由两个步骤组成:第 1 步,根据某些特征评估标准,将特征重要性进行特征评分来排序;第 2 步,过滤重要性低的特征,保留重要性评价高的特征.一些代表性的标准包括特征相关性标准[5]、互信息标准[6,7]、特征判别能力标准[8]、保存数据流形结构能力标准[9,10]以及重建原始数据能力标准[11,12]等.过滤式方法通常比包裹式方法更高效,但是由于缺乏具体的学习算法指导特征选择阶段,所选特征可能并不是最适合目标学习算法的.
- 嵌入式方法:提供了包裹式方法和过滤式方法之间的折衷解决方案.通过将特征选择与学习模型相结合的方法,从而继承了包裹式方法和过滤式方法的优点:(1) 与学习算法的交互;(2) 比包裹式方法更高效.一种典型的嵌入式方法就是 LARS[13].

特征选择困难的主要原因是搜索空间随特征数呈指数增长[5],因此,如何采用高效的搜索策略进行搜索,往往是特征选择问题能否有效求解的关键.演化计算技术由于具有良好的全局搜索能力,近年来在特征选择领域获得了越来越多的关注[14].与传统的搜索方法相比,演化计算方法的明显优势在于其通常不需要领域知识和对搜索空间做任何的假设(例如是线性或非线性可分的),并且由于它基于种群机制的特点,能够在一次运行中产生多种结果,使其更适合用来进行多目标特征选择以确保在特征数量和分类性能中取得平衡.一些较新的基于演化计算的特征选择算法包括 Zhu 等人将遗传算法和局部搜索方法结合起来提出的 FS-NEIR[15]、Xue 等人基于粒子群算法提出的 PSO(4-2)[16]、Tabakhi 等人基于蚁群算法提出的无监督特征选择算法 UFSACO[17]、Zhang 等人基于萤火虫算法提出的 Rc-BBFA[18]等.Ghaemi 等人提出了森林优化特征选择算法(feature selection using forest optimization algorithm,简称 FSFOA)[19],FSFOA 相对于其他算法而言,仅需较小的计算代价就能达到较高

的分类准确率,并保证良好的泛化性能[20].但是 FSFOA 仍有一些需要改进之处表现在了如下几个方面.

- 第一,初始化森林时 FSFOA 单纯采用随机的方式从 *m* 维的特征中选取特征,而这种方式具有一定的盲目性[20].对于 FSFOA 来说,在局部播种阶段寻找最优的特征子集围绕初始化展开,全局播种更依赖于局部播种的结果,因此初始化方案的好坏,不仅直接影响后续的搜索能否选到分类准确率更高的解,也间接影响维度缩减率的高低.
- 第二,FSFOA 在全局播种阶段的参数 *GSC*(global seeding change)设置上类似该算法局部播种阶段的参数 *LSC*(local seeding change)的设置,均由实验人员根据经验设定,这样的固定参数设定限制了全局播种辅助局部播种寻找全局最优解的能力,并没有发挥出全局播种策略的最大优势.
- 第三,在最优树选择方面,FSFOA 单纯从已知的优质森林中选择其中一棵树作为最优的特征子集树,极大地浪费了经过若干轮选出的优质森林中的其他次最优树.
- 第四,FSFOA 存在着高维数据处理能力不足的本质缺陷.

因此,森林优化特征选择算法仍存在较多可以完善之处.参考最近的相关研究[21−23]可以看出,结合信息增益方法对最优特征子集的选取具有良好的指导作用.模拟退火算法思想最早是由 Metropolis 等人于 1953 年提出的,1983 年,Kirkpatric 等人成功地将退火思想引入组合优化领域[24].模拟退火算法中模拟了冶金行业的操作过程,通过设计递减的温度控制函数,使粒子由高温无序状态最终趋于低温时的基态.基于以上思想,本文采用一种新的初始化策略和 *GSC* 参数选择策略,同时,在算法选优阶段引入贪心算法来改进 FSFOA.最后,针对 FSFOA 高维数据处理能力不足的问题,我们采用集成特征选择处理的思路对高维数据集进行处理,很好地弥补了算法存在的缺陷.为了便于描述,我们将改进后的 FSFOA 记为 EFSFOA(enhanced feature selection using forest optimization algorithm).最后,我们选择了 15 个 UCI[25]中的数据集,将 EFSFOA 与 FSFOA 和近几年提出的其他比较高效的基于演化计算的特征选择算法进行了对比实验,通过实验得出,EFSFOA 在分类性能和泛化能力上均具有不错的表现.

1 FSFOA 概述

FSFOA 是 Ghaemi 等人基于森林优化算法(forest optimization algorithm,简称 FOA)[26]提出的一种新型的特征选择算法,并将此算法在离散型数据集上进行特征选择实验,实验取得了良好的效果.

FSFOA 将每一个解集(即特征子集)表示为一棵树的形式,树的长度与特征的长度相等,每个特征在树中用“0”或“1”的形式记录:“0”表示不选择该特征参与后续的学习任务,“1”表示选择该特征用于后续的学习.

FSFOA 包含森林初始化、局部播种、形成候选森林、全局播种、更新优质树、挑选最优树等 6 个阶段,其中,局部播种、形成候选森林、全局播种和更新优质树为主要迭代对象.通过不断的迭代,产生新树更新森林,最终挑选森林中的最优树作为最优特征子集.

- 初始化森林:产生 *N* 棵特征随机的树并形成森林.每棵树通过 0/1 字符串记录所选择的对应的特征子集;同时,通过一个 *Age* 属性记录树的“年龄”,每棵新生成的树 *Age* 均设置为 0.这里,初始化过程单纯采用随机的方式来完成,具有很大的盲目性.对于 FSFOA 来说,在局部播种阶段寻找最优的特征子集围绕初始化展开,全局播种阶段更依赖于局部播种的结果,因此,针对每棵树的初始选取是十分关键的.初始化方案的好坏,不仅直接影响算法能否通过后续的搜索选到分类准确率更高的解,也间接影响算法维度缩减率的高低.在后文实验结果对比分析中,我们会结合具体的对比实验结果做进一步说明.
- 局部播种:依据 *LSC* 参数对 *Age* 为 0 的树进行局部播种生成新树,即对 *Age* 为 0 的树随机选择“某一个”特征,并将该特征对应的变量值从 1 改为 0,反之亦然.具体过程如图 1 所示,将新生成的树 *Age* 设置为 0 并将其添加到森林中,将森林中剩余树的 *Age* 加 1.
- 形成候选森林:为限制局部播种后森林的规模,FSFOA 中引入两个新的参数 *life time* 和 *area limit* 来控制森林中树的数量.这一过程中,先将森林中 *Age* 大于年龄上限参数 *life time* 的树放入候选森林,然后如果森林中剩余树的数量仍然超出区域上限参数 *area limit*,则根据森林中树的适应度值从小到大依次

将多余的树放入候选森林,直到森林中树的数量等于 *area limit* 为止.

- 全局播种:依据设定的 *transfer rate* 参数从候选森林中随机选择若干棵树,每棵树按 *GSC* 参数随机选取一些特征进行全局播种生成新树,同时,这些特征对应的变量值从 1 改为 0;反之亦然.具体过程如图 2 所示,全局播种后,此时新生成的树 *Age* 与生成它的树 *Age* 相同,将它们加入候选森林,然后在候选森林中选取对应特征子集的分类准确率最高的树,将其加入到森林中;同时,*Age* 置为 0.全局播种这样通过人为给定参数的播种方式过于盲目,不同于局部播种的 *LSC* 参数,更多影响的是关于问题求解时间的大小,并不影响本质策略——无论 *LSC* 的数值如何选取,改变的特征都是 1 个,人为给定的方式往往是通过该参数控制算法搜索时间.而 *GSC* 的作用更多的是通过该参数辅助局部播种寻求最优解,寄希望于通过改变数个特征(个数为 *GSC* 的取值)来进行一次扩展的全局搜索,进而搜索到一个局部最优的解.因此,该参数的好坏对于全局播种阶段来说是至关重要的.

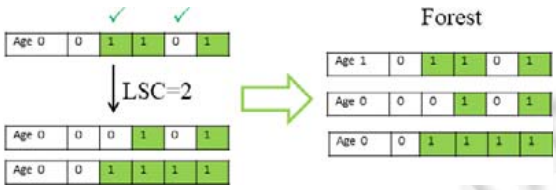

Fig.1 Local seeding with *LSC*=2

图 1 局部播种(*LSC*=2)

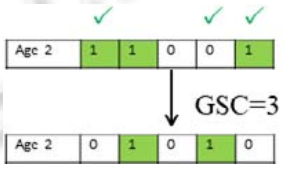

Fig.2 Global seeding with *GSC*=3

图 2 全局播种(*GSC*=3)

- 更新阶段:比较经过全局播种选出的树的准确率,把准确率最大的树作为优质树,并将其 *Age* 置 0,重新放入森林中.
- 挑选最优树:在迭代结束后的森林中,依据每棵树的准确率,选择准确率最大的一棵作为最优树,即全局最优解.这种择优方式在一定程度上造成了森林中其他树的浪费,因为相对于所选的最优树来说,它们虽然不是最优的,不过却能在森林中存活到最后,足以表明它们所选择的特征相对来说也是十分良好的.因此,并不能单纯地认为现有森林中的最优树相比于取森林中全部树的特征交集得来的树更优秀.

2 EFSFOA

前文概述了 FSFOA,并分析了其不足之处.本节针对以上不足提出 3 点改进方案,同时考虑文献[19]中指出的 FSFOA 对高维数据集的处理需要完善,且其适应度函数需要综合维度缩减率进行优化等方面的内容.我们给出了最新的完善方案:更新、更高效的特征选择选择算法 EFSFOA.通过对比实验表明,EFSFOA 不仅在低维数据集的处理上比 FSFOA 有了更好的表现力,尤其在高维数据集的处理上,很好地解决了 FSFOA 的问题.

2.1 初始森林的生成形式

针对 FSFOA 随机生成的初始化森林可能出现不利于后续搜索的局面,我们结合信息论中的理论,提出一种结合了信息增益率的启发式初始化策略.信息增益率 *IGR*(information gain ratio)的计算公式由公式(1)给出.

$$\left.\begin{aligned} &P(X=x_i)=p_i, i=1,2,...,m\\ &H(c)=-\sum_{i=1}^{n}p(c_i)\cdot\log_2 p(c_i)\\ &H(c\mid X)=-\sum_{i=1}^{m}p(c,X=x_i)\log_2 p(c\mid X=x_i)\\ &IG(X)=H(c)-H(c\mid X)\\ &IGR(X)=IG(X)/H(X) \end{aligned}\right\} \tag{1}$$

通过采用这种策略,保证了初始化森林的优良性.如果把每个特征比喻成一个基因,把森林的局部播种、全局播种看成树的繁衍行为,那么通过启发后的初始森林将确保具备足够的优秀基因(即最利于进行分类的特

征),为后代的生长带来更大的优势:即后续搜索将依大概率趋于全局最优.但如果全部的树都具有该基因,可能导致该基因的过于强大而错失了基因多样性(算法过多的依赖某一特征展开搜索).而众所周知,多样性对于演化算法寻求最优解往往起到关键性的作用.因此,我们的初始化策略在原有初始化策略的基础上,对随机生成的 N 棵树中的 $N/2$ 棵树启发的赋予依据信息增益率选出的最优特征(如果原来该特征标记已经为 1,则保持不变;否则置为 1).

2.2 采用模拟退火过程的自适应全局播种策略

如前文所述,FSFOA 在全局播种阶段,通过人为给定 *GSC* 参数的方式并不能充分利用全局播种辅助寻优.

- 首先,因为全局播种策略本质上是一种全局搜索策略,这种随机全局搜索的策略不仅搜索效率高,而且搜索范围广,往往很容易找到相对优秀的搜索空间.但由于 *GSC* 参数的固定,使得这样的全局搜索固定了搜索空间的大小,在算法找到了相对优秀的搜索空间之后,只能依赖于局部播种策略的单点改变来使该搜索空间逐渐收敛,使得收敛到区间最优解的难度过大.
- 其次,由于在算法的迭代过程中,生成的候选森林中的每棵树存在的形态并不完全一致,有些树生成时就相对劣质,因而会导致其在较低年龄时即被淘汰放入候选森林;而有些树由于自身特征集合相对优秀,使得它们可以在森林中存活相对长的时间,直到达到年龄上限后才被放入候选森林.所以对于算法经过多轮迭代后逐步形成的候选森林而言,在候选森林中存在的年龄较大的树比年龄较小的树更为优秀,即它所对应的特征集合及搜索空间更趋近于最优解(局部最优或全局最优).

基于上述两点考虑,我们受到模拟退火算法中曾使用到的固体退火原理的启发,提出了自适应全局播种策略.在该策略中,针对候选森林中的树,通过其年龄生成 *GSC* 参数进行全局播种,其表示形式如公式(2)所示.

$$GSC=T(Age) \tag{2}$$

类比于退火过程,构造满足单调递减性质的函数 T,且需保证 $2\leqslant T(Age)\leqslant 0.5\times features$.满足 $2\leqslant T(Age)$是为了确保全局播种不退化为局部播种,保留其特色;满足 $T(Age)\leqslant 0.5\times features$ 是为了控制特征改变的范围,因为形式上改变 50%的特征已经达到了上界.满足这类性质的函数 T 可以找到若干个,我们从 *Age* 变化导致步长范围变化的考虑出发,经过反复实验,最终选取了如公式(3)所示的函数 T.

$$\left.\begin{aligned} T_0 &= \min(2+2\times life\ time,\ features\times 0.5)\\ T(Age) &= T_0/(Age+1)\end{aligned}\right\} \tag{3}$$

通过这种自适应全局播种策略,不仅使得解空间容易收敛到最优解,而且充分利用了全局播种策略辅助寻优的特性.

2.3 贪心取优策略

在最优树选择上,FSFOA 仅从局部的角度考虑最优树存在于不断迭代产生的优质森林中,这样的寻优策略极大地造成了优质森林中资源的浪费,忽略了其他若干次最优树(这里的次最优树指最后形成的优质森林中除最优树以外的所有树)所蕴含的特征.为了避免这样的资源浪费,我们从全局出发,合理地认为最优树并非存在于最后所生成的优质森林中,而是应该由优质森林中的每一棵树共同作用所产生.具体取优策略如下.

EFSFOA 贪心取优策略:第 1 步,在优质森林中选出最优树作为 1 号待选最优树(选取方式与 FSFOA 选取最优树一致)并统计其选取的特征个数 N;第 2 步,统计所有特征在优质森林中的每一棵树中出现的总频数并按从大到小的顺序排序;第 3 步,选取总频数排在前 N 个的特征生成 2 号待选最优树,选取总频数排在前 N–1 个的特征生成 3 号待选最优树;第 4 步,比较 1 号待选最优树和 2 号待选最优树(1 号和 2 号的维度缩减能力一致),选择准确率较高的一棵和 3 号待选最优树进行比较,再选择准确率较高的一棵作为最后的最优树,如果准确率一致,选择 3 号待选最优树作为最后的最优树(因为 3 号的维度缩减能力高于 1 号和 2 号).

2.4 高维数据集的处理

鉴于 FSFOA 需要进一步完善以达到高维数据处理的目的,我们考虑特征选择森林适用于集成的特性,因此尝试通过加入集成特征选择的特点来进一步完善我们的算法.综合考虑我们算法的特点和文献[27]中介绍的集

成特征选择框架,我们设计了如图 3 所示的适用于 EFSFOA 进行高维数据处理的集成特征选择框架.

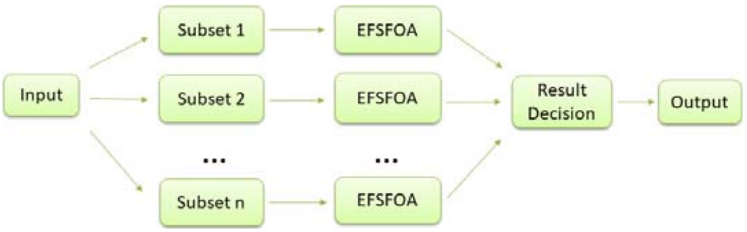

Fig.3 Ensemble feature selection framework of EFSFOA

图 3 EFSFOA 的集成特征选择框架

实际操作中,我们首先设置一个参数 *sliding*,并根据 *sliding* 的大小将原始的高维数据集按照特征数量进行顺序切分,然后将切分后的数据集同时使用 EFSFOA 进行并行化处理,最后将所有的计算结果进行一次决策处理,产生最终的计算结果.在决策处理的过程中,我们首先从局部的角度出发,考虑到每一部分数据集经过特征选择后的结果亦是原始问题的一个局部最优解,因此首先对所有求得的解进行一次比较,选取其中最优的解作为局部的最优解;然后,我们从全局角度进行考虑,考虑将 n 个特征选择的结果进行合并,用来产生一个全局的最优解.由于 n 个解中仍含有大量的特征,并不适用于盲目的拼接后穷举搜索的策略.不过比较令人欣慰的是,这 n 个解中所包含的特征是对原始特征经过 *sliding* 提取之后的进一步提取,它们更加具有代表性.因此,我们对这 n 个解中的每个特征采用类似亚线性抽样算法的方式进行特征的筛选,来达到进一步搜索的效果.每个特征的抽样概率如公式(4)所示.

$$P(IsSelected)=CA\times f\times 100\% \tag{4}$$

其中,CA 为每个 EFSFOA 结束后计算的分类准确率(具体由公式(6)给出,关于公式(6),会在第 3.1 节中进行详细说明),f 为每个 EFSFOA 结束后计算的最优特征在该森林中出现的频率.

2.5 新的适应度函数

由于 FSFOA 主要面向低维数据的处理,因此其仅使用分类准确率作为特征子集选择时的适应度函数是可以接受的.而 EFSFOA 不仅致力于在处理低维数据时令结果更加优秀,而且旨在完善 FSFOA 在高维数据特征选择上的缺陷.鉴于 Ghaemi 等人已经指出了原有的适应度函数如果用在高维数据的特征选择处理上时会存在维度缩减率考虑不充分的情况,为此,我们提出了一个新的适应度函数,来确保 EFSFOA 在处理高维数据时能够考虑维度缩减率.新的适应度函数如公式(5)所示.

$$Fitness(tree)=\alpha\times CA+\beta\times DR \tag{5}$$

其中,CA 和 DR 为分类准确率和维度缩减率(DR 具体由公式(7)给出,关于公式(7),会在第 3.1 节中进行详细说明),α和β为调节参数.引入两个调节参数的目的是考虑到了算法的实际应用情况,比如一些应用场景中需要把准确率放在首位,而另一些则需要把维度缩减率放在优先考虑的位置,因此增加α和β两个调节参数来提高算法的灵活性(在我们的实验中,通常将α和β的关系设为$\alpha\gg\beta>0$ 的关系,用来保证在添加维度缩减率作为考量更新机制因素的同时,分类准确率仍然处于首要位置,如果$\beta\gg\alpha>0$,将导致算法过多考虑维度缩减率,有时会导致准确率过低,算法失去稳定性.因此,即使需要优先考虑维度缩减率,也建议在$\alpha>\beta>0$ 的条件下进行调节).

从公式(5)可以看出,新的适应度函数更为综合地考量了分类准确率和维度缩减率,因此也能够进一步提高在低维数据处理上的性能,所以我们在 EFSFOA 对低维数据的处理上也应用了该函数.后面的实验结果验证了我们的观点.

2.6 算法伪代码

为了便于对比,将 FSFOA 伪代码[19]在算法 1 中给出,EFSFOA 伪代码在算法 2 和算法 3 中给出.算法 2 是我们在 FSFOA 的基础上进行增强后得到的应对低维数据处理的方法,其中粗体标明了改进的地方:第 3*步对应我

们在第 2.1 节中提出的新的初始化策略,其目的是根据给定的数据集返回信息增益率最高的特征,然后将森林中一半的树中该特征所对应的标志更新为“1”,该过程中应用的 *InfoGainRatio*(*dataSet*)函数的作用是依据公式(1)计算出具有最优信息增益率的特征;迭代过程中的第 11 步和第 16 步中分别使用了第 2.2 节中介绍的 *GSC* 更新策略以及第 2.5 节中提出的新的适应度函数进行最优树的选取;第 4*步、第 5*步对应我们在第 2.3 节中提出的新的取优策略,其中,*GroupSelection*(*forest*)函数通过给定迭代后生成的优质森林,返回我们在第 2.3 节中描述的 1 号~3 号待选最优树.算法 3 是我们为了让 EFSFOA 能够处理高维数据而在算法 2 的基础之上所做的进一步扩展,其主要思路如第 2.4 节所述.为了便于理解,我们在此对重要步骤做进一步说明:第 1*步将高维数据集按照 *sliding* 参数的大小切分成 *N* 个算法 2 中所述算法能够处理的低维数据集;第 2*步使用算法 2 中所述算法并行计算第 1*步中切分后的 *N* 个数据集;第 5*步是将第 2*步的计算结果合并成一个包含全部重要特征的集合;循环阶段使用的 *Sample*(*important_features*)函数是对第 5*步中得到的集合模拟多次抽样,以便找出一个更为优秀的全局最优解,其中,每个特征依据公式(4)所计算的特征抽样概率进行抽样;第 7*步从局部最优解和全局最优解中选择一个较为优秀的解作为算法的最终特征选择结果.

算法 1. *FSFOA*(*life time*, *LSC*, *GSC*, *transfer rate*, *area limit*).

输入:*life time*, *LSC*, *GSC*, *transfer rate*, *area limit*.

输出:适应度最高的特征子集.

1*: 初始化森林

2*: 将森林中的每棵树的 *Age* 置为 0

While

1: 对森林中 *Age* 为 0 的树进行局部播种

2: **for** *i*=1: *LSC* **do**

3: 在选定的树中随机选择一个位置

4: 将选定位置为 0 的值变为 1,反之亦然

5: **end for**

6: 将森林中所有树的 *Age* 加 1

7: 根据 *life time* 和 *area limit* 进行种群规模限制

8: 对候选森林进行全局播种

9: 从候选森林中选择 *transfer rate* 比例的树

10: **for** 每棵选择的树 **do**

11: 在选定的树中随机选取 *GSC* 个位置

12: 将对应位置为 0 的值变为 1,对应位置为 1 的值变为 0

13: **end for**

14: 更新最优树

15: 根据树的适应度值进行排序

16: 将适应度值最高的树的 *Age* 重置为 0

End while

3*: 返回适应度值最好的树并表示最终的特征子集

算法 2. *EFSFOA_LOW*(*life time*, *LSC*, *transfer rate*, *area limit*).

输入:*life time*, *LSC*, *transfer rate*, *area limit*.

输出:适应度最高的特征子集.

1*: 初始化森林

2*: 将森林中的每棵树的 *Age* 置为 0

3*: 依据 *InfoGainRatio*(*dataSet*)更新森林中一半的树

While 满足执行条件 **do**

1: 对森林中 *Age* 为 0 的树进行局部播种

2: **for** *i*=1: *LSC* **do**

3: 在选定的树中随机选择一个位置

4: 将选定位置为 0 的值变为 1,反之亦然

5: **end for**

6: 将森林中所有树的 *Age* 加 1

7: 根据 *life time* 和 *area limit* 进行种群规模限制

8: 对候选森林进行全局播种

9: 从候选森林中选择 *transfer rate* 比例的树

10: **for** 每棵选择的树 **do**

11: ***GSC*←*T*(*Age*)**

12: 在选定的树中随机选取 *GSC* 个位置

13: 将对应位置为 0 的值变为 1,对应位置为 1 的值变为 0

14: **end for**

15: 更新最优树

16: **根据 *Fitness*(*tree*)选出适应度值最高的树**

17: 将适应度值最高的树的 *Age* 重置为 0

End while

4*: **根据 *GroupSelection*(*forest*)生成 1 号、2 号、3 号待选最优树**

5*: **从 1 号、2 号、3 号待选最优树中找出最好的树作为最优树**

6*: 返回最优树并表示最终的特征子集

算法 3. *EFSFOA_HIGH*(*life time*, *LSC*, *transfer rate*, *area limit*, *sliding*).

输入:*life time*, *LSC*, *transfer rate*, *area limit*, *sliding*.

输出:适应度最高的特征子集.

1*: 根据 *sliding* 大小将高维数据集划分成 *N* 个低维数据集

2*: 并行执行 *EFSFOA_LOW*(*life time*, *LSC*, *transfer rate*, *area limit*)处理划分后的 *N* 个低维数据集

3*: *results*←并行执行后的 *N* 个结果

4*: *micro_final_result*←从 *results* 中选择最好的结果

5*: *important_ features*←将 *results* 中的 *N* 个结果组合

6*: 初始化变量 *macro_final_result* 为 None

While 满足执行条件 **do**

1: *macro_temp_final_result*←*Sample*(*important_features*)

2: **if** *macro_temp_final_result* 优于 *macro_final_result*

then *macro_final_result*←*macro_temp_final_result*

End while

7*: **if** *micro_final_result* 优于 *macro_final_result*

then *final_result*←*micro_final_result*

else *final_result*←*macro_final_result*

8*: 返回 *final_result* 作为高维数据集的最终特征选择结果

3 实验结果与分析

在本文实验中,主要采用 python3.6 以及工具包 scikit-learn 实现代码.所有实验均在一台配置为 Intel i7-7700K、16GB 内存、500GB 硬盘的计算机上完成.

3.1 数据集

我们在 15 个 UCI[25]数据集上对我们提出的方法进行了对比实验(见表 1).

Table 1 Experimental dataset

表 1 实验数据集

Dataset	#Feature	#Instance	#Class
Glass	9	214	7
Heart	13	270	2
Cleveland	13	303	5
Wine	13	178	3
Vehicle	18	846	4
Segmentation	19	2 310	7
Ionosphere	34	351	2
Dermatology	34	366	6
Sonar	60	208	2
LSVT	310	126	2
CNAE-9	856	1 080	9
SRBCT	2 308	63	4
Arcene	10 000	200	2
RNA-Seq	20 531	801	5
Dorothea	100 000	800	2

分类准确率的具体定义如公式(6)所示.

$$CA=NCC/NAS \tag{6}$$

其中,*NCC* 代表正确的分类数,*NAS* 代表数据集的实例总数.维度缩减率的定义如公式(7)所示.

$$DR=1-(NSF/NAF) \tag{7}$$

其中,*NSF* 代表选择的特征数,*NAF* 代表数据集的特征总数.

3.2 参数设置

实验过程中,EFSFOA 除了适应度函数中新增的参数α和β、高维数据处理中新增的参数 *sliding size* 以及无法列出的改进的自适应参数 *GSC* 外,参数设置与 FSFOA 完全相同以便确保对比实验公平性.FSFOA 和 EFSFOA 的参数设置分别由表 2 和表 3 给出.

Table 2 Specific information of the parameters of FSFOA

表 2 FSFOA 参数的具体信息

Dataset	Life time	Area limit	Transfer rate	*LSC*	*GSC*
Glass	15	50	0.05	2	4
Heart	15	50	0.05	3	6
Cleveland	15	50	0.05	3	6
Wine	15	50	0.05	3	6
Vehicle	15	50	0.05	4	9
Segmentation	15	50	0.05	4	9
Ionosphere	15	50	0.05	7	15
Dermatology	15	50	0.05	7	15
Sonar	15	50	0.05	12	30
LSVT	15	50	0.05	68	115
CNAE-9	15	50	0.05	26	100
SRBCT	15	50	0.05	460	700
Arcene	15	50	0.05	22	500
RNA-Seq	15	50	0.05	32	1 027
Dorothea	15	50	0.05	71	5 000

Table 3 Specific information of the parameters of EFSFOA

表 3 EFSFOA 参数的具体信息

Dataset	Life time	Area limit	Transfer rate	*LSC*	α	β	Sliding size
Glass	15	50	0.05	2	100	0.01	–
Heart	15	50	0.05	3	100	0.01	–
Cleveland	15	50	0.05	3	100	0.01	–
Wine	15	50	0.05	3	100	0.01	–
Vehicle	15	50	0.05	4	100	0.01	–
Segmentation	15	50	0.05	4	100	0.01	–
Ionosphere	15	50	0.05	7	100	0.01	–
Dermatology	15	50	0.05	7	100	0.01	–
Sonar	15	50	0.05	12	100	0.01	–
LSVT	15	50	0.05	68	100	0.01	–
CNAE-9	15	50	0.05	26	100	0.01	–
SRBCT	15	50	0.05	460	100	0.01	–
Arcene	15	50	0.05	22	100	0.01	1 000
RNA-Seq	15	50	0.05	32	100	0.01	2 054
Dorothea	15	50	0.05	71	100	0.01	10 000

3.3 实验结果对比分析

我们在比较EFSFOA与FSFOA的同时,也加入了近年来提出的特征选择算法进行对比.其他对比算法的具体信息由表4给出,实验结果由表5给出.为了确保对比实验结果的准确性,部分数据结果均采用了文献[19]中公开发表的实验结果.表 5 中的粗体部分对应了该组实验中最佳分类准确率(*CA*)和维度缩减率(*DR*);10-fold、2-fold、70%~30%、50%~50%分别表示 10 折,2 折,70%作训练集、30%作测试集以及 50%作训练集、50%作测试集这 4 种验证方式;1-NN、3-NN、5-NN、SVM、DT 分别表示 1 近邻、3 近邻、5 近邻、支持向量机、决策树这 5 种分类器.

Table 4 Brief description of the methods for comparisons

表 4 对比算法的简要描述

算法名称	描述/发表年份
FSFOA	基于森林优化的特征选择算法[[19]]/2016
Rc-BBFA	基于收益成本的萤火虫算法的特征选择算法[[18]]/2017
UFSACO	基于蚁群优化的无监督特征选择算法[[17]]/2014
PSO(4-2)	基于粒子群优化算法的特征选择算法[[16]]/2013
FS-NEIR	基于近邻有效信息比的特征选择算法[[15]]/2013
SVM-FuzCoc	基于支持向量机的特征选择算法[[28]]/2010
NSM	基于近邻软间隔的特征选择算法[[29]]/2010
SFS、SBS、SFFS	前向选择、后向选择、序列浮动前向选择的特征选择算法[[28]]/2010
HGAFS	基于混合遗传算法的特征选择算法[[30]]/2007

Table 5 Experimental results

表 5 实验结果

Glass	*CA*	*DR*	Validation	Classifier	**Cleveland**	*CA*	*DR*	Validation	Classifier
EFSFOA	**80.95%**	**44.44%**	70%~30%	1-NN	EFSFOA	**62.07%**	**84.62%**	70%~30%	1-NN
FSFOA	71.88%	40%	70%~30%	1-NN	FSFOA	55.55%	71.42%	70%~30%	1-NN
SFS	72.24%	26.66%	70%~30%	1-NN	SVM-FuzCoc	61.01%	46.1%	70%~30%	1-NN
SFFS	71.77%	37.77%	70%~30%	1-NN	SFS	51.79%	47.7%	70%~30%	1-NN
EFSFOA	**71.03%**	44.44%	2-fold	SVM	SBS	54.8%	38.5%	70%~30%	1-NN
FSFOA	68.22%	**60%**	2-fold	SVM	SFFS	49.5%	53.8%	70%~30%	1-NN
HGAFS	65.51%	44.44%	2-fold	SVM	**LSVT**	*CA*	*DR*	Validation	Classifier
EFSFOA	**84.39%**	**56.67%**	10-fold	DT	EFSFOA	**94.74%**	97.1%	70%~30%	1-NN
FSFOA	75.7%	50%	10-fold	DT	FSFOA	89.47%	**98.71%**	70%~30%	1-NN
FS-NEIR	68.53%	22.22%	10-fold	DT	Rc-BBFA	94.60%	56.45%	70%~30%	1-NN

Table 5 Experimental results (Continued)
表 5 实验结果(续)

Heart	*CA*	*DR*	Validation	Classifier	**Sonar**	*CA*	*DR*	Validation	Classifier
EFSFOA	**92.59%**	**70%**	10-fold	3-NN	EFSFOA	**90.82%**	76%	70%~30%	5-NN
FSFOA	85.18%	35.71%	10-fold	3-NN	FSFOA	86.98%	44.26%	70%~30%	5-NN
NSM	84%	69.23%	10-fold	3-NN	PSO(4-2)	78.16%	**81.26%**	70%~30%	5-NN
EFSFOA	**85.93%**	63.08%	2-fold	SVM	EFSFOA	**98.41%**	**83.33%**	70%~30%	1-NN
FSFOA	84.07%	50%	2-fold	SVM	FSFOA	85.43%	57.37%	70%~30%	1-NN
HGAFS	82.59%	**76.92%**	2-fold	SVM	Rc-BBFA	95.57%	53.33%	70%~30%	1-NN
EFSFOA	**93%**	**66.15%**	10-fold	DT	SVM-FuzCoc	73.17%	68.33%	70%~30%	1-NN
FSFOA	85.15%	48.07%	10-fold	DT	SFS	66.43%	61.33%	50%~50%	1-NN
FS-NEIR	79.86%	46.15%	10-fold	DT	SBS	62.2%	45.33%	50%~50%	1-NN
Arcene	*CA*	*DR*	Validation	Classifier	SFFS	64.55%	61.33%	50%~50%	1-NN
EFSFOA	**95%**	**92.13%**	70%~30%	1-NN	EFSFOA	82.02%	**84.33%**	2-fold	SVM
FSFOA	88.33%	61.9%	70%~30%	1-NN	FSFOA	65.86%	54.09%	2-fold	SVM
Rc-BBFA	92.5%	48.66%	70%~30%	1-NN	HGAFS	**87.02%**	75%	2-fold	SVM
EFSFOA	**81.15%**	63.24%	70%~30%	DT	EFSFOA	**97.72%**	73.17%	10-fold	DT
FSFOA	73.69%	77.67%	70%~30%	DT	FSFOA	82.69%	52.45%	10-fold	DT
UFSACO	67.4%	**99.8%**	70%~30%	DT	FS-NEIR	75.97%	**91.66%**	10-fold	DT
Wine	*CA*	*DR*	Validation	Classifier	**Ionosphere**	*CA*	*DR*	Validation	Classifier
EFSFOA	**99.26%**	**68.38%**	10-fold	3-NN	EFSFOA	**99.43%**	87.06%	10-fold	3-NN
FSFOA	98.87%	42.58%	10-fold	3-NN	FSFOA	92.3%	61.76%	10-fold	3-NN
NSM	98%	53.84%	10-fold	3-NN	NSM	92%	**88.23%**	10-fold	3-NN
EFSFOA	99.08%	**68.38%**	70%~30%	1-NN	EFSFOA	**98.1%**	75.29%	70%~30%	1-NN
FSFOA	98.07%	50%	70%~30%	1-NN	FSFOA	89.52%	54.28%	70%~30%	1-NN
Rc-BBFA	**99.66%**	38.46%	70%~30%	1-NN	Rc-BBFA	96.18%	58.82%	70%~30%	1-NN
SVM-FuzCoc	97.12%	53.84%	70%~30%	1-NN	SVM-FuzCoc	89.46%	**88.23%**	70%~30%	1-NN
SFS	97.69%	35.38%	70%~30%	1-NN	SFS	87.75%	65.88%	50%~50%	1-NN
SBS	94.77%	46.15%	70%~30%	1-NN	SBS	84.61%	77.64%	50%~50%	1-NN
SFFS	96.56%	36.92%	70%~30%	1-NN	SFFS	88.32%	75.29%	50%~50%	1-NN
EFSFOA	**98.31%**	**57.69%**	2-fold	SVM	EFSFOA	**95.67%**	65.59%	2-fold	SVM
FSFOA	96.06%	37.17%	2-fold	SVM	FSFOA	94.58%	57.14%	2-fold	SVM
HGAFS	**98.31%**	53.85%	2-fold	SVM	HGAFS	92.76%	**82.35%**	2-fold	SVM
EFSFOA	**98.76%**	**68.38%**	70%~30%	DT	EFSFOA	**96.78%**	**64.71%**	70%~30%	DT
FSFOA	96%	57.14%	70%~30%	DT	FSFOA	95.12%	47.05%	70%~30%	DT
UFSACO	95.08%	61.53%	70%~30%	DT	UFSACO	88.61%	11.17%	70%~30%	DT
EFSFOA	**99.25%**	**81.54%**	10-fold	DT	EFSFOA	**98.4%**	77.65%	10-fold	DT
FSFOA	96.06%	21.42%	10-fold	DT	FSFOA	93.16%	68.57%	10-fold	DT
FS-NEIR	95.04%	61.53%	10-fold	DT	FS-NEIR	92.59%	**82.35%**	10-fold	DT
Vehicle	*CA*	*DR*	Validation	Classifier	**Dermatology**	*CA*	*DR*	Validation	Classifier
EFSFOA	75.79%	50%	70%~30%	5-NN	EFSFOA	**99.63%**	52.94%	70%~30%	1-NN
FSFOA	73.98%	50%	70%~30%	5-NN	FSFOA	97.27%	45.71%	70%~30%	1-NN
PSO(4-2)	**85.3%**	**68.4%**	70%~30%	5-NN	SVM-FuzCoc	94.11%	**64.7%**	70%~30%	1-NN
EFSFOA	**78.35%**	55.56%	70%~30%	1-NN	SFS	94.02%	44.7%	70%~30%	1-NN
FSFOA	73.81%	**61.11%**	70%~30%	1-NN	SBS	91.78%	58.23%	70%~30%	1-NN
Rc-BBFA	75.79%	**61.11%**	70%~30%	1-NN	SFFS	93.7%	62.35%	70%~30%	1-NN
EFSFOA	71.35%	**71.11%**	2-fold	SVM	EFSFOA	**99.91%**	**73.24%**	10-fold	DT
FSFOA	62.41%	47.22%	2-fold	SVM	FSFOA	96.99%	21.42%	10-fold	DT
HGAFS	**76.36%**	38.99%	2-fold	SVM	FS-NEIR	93.95%	70.58%	10-fold	DT
EFSFOA	**82.8%**	43.89%	10-fold	DT	EFSFOA	**98.15%**	**63.24%**	70%~30%	DT
FSFOA	73.04%	31.57%	10-fold	DT	FSFOA	90.09%	44.11%	70%~30%	DT
FS-NEIR	70.98%	**50%**	10-fold	DT	UFSACO	95.28%	26.47%	70%~30%	DT
RNA-Seq	*CA*	*DR*	Validation	Classifier	**Dorothea**	*CA*	*DR*	Validation	Classifier
EFSFOA	**97.6%**	**65.69%**	70%~30%	1-NN	EFSFOA	**96.25%**	**98.35%**	70%~30%	1-NN
FSFOA	–	–	70%~30%	1-NN	FSFOA	–	–	70%~30%	1-NN
Rc-BBFA	94.53%	58.66%	70%~30%	1-NN	Rc-BBFA	–	–	70%~30%	1-NN
EFSFOA	**98.75%**	**68.11%**	70%~30%	SVM	EFSFOA	**93.75%**	**98.11%**	70%~30%	SVM
FSFOA	–	–	70%~30%	SVM	FSFOA	–	–	70%~30%	SVM
EFSFOA	**98.13%**	**42.97%**	70%~30%	DT	EFSFOA	**97.21%**	**99.8%**	70%~30%	DT
FSFOA	–	–	70%~30%	DT	FSFOA	–	–	70%~30%	DT
CNAE-9	*CA*	*DR*	Validation	Classifier	**Segmentation**	*CA*	*DR*	Validation	Classifier
EFSFOA	**95.06%**	35.86%	70%~30%	1-NN	EFSFOA	96.98%	**78.95%**	70%~30%	1-NN
FSFOA	91.05%	24.88%	70%~30%	1-NN	FSFOA	96.51%	36.84%	70%~30%	1-NN
Rc-BBFA	**95.06%**	**50.35%**	70%~30%	1-NN	Rc-BBFA	**98.27%**	36.84%	70%~30%	1-NN
SRBCT	*CA*	*DR*	Validation	Classifier	EFSFOA	**96.36%**	**71.05%**	10-fold	3-NN
EFSFOA	**99.87%**	96.58%	70%~30%	1-NN	FSFOA	96.2%	30%	10-fold	3-NN
FSFOA	94.73%	49.06%	70%~30%	1-NN	NSM	95%	63.15%	10-fold	3-NN
SVM-FuzCoc	98.88%	**98.57%**	70%~30%	1-NN	–	–	–	–	–

对比分类准确率,通过表 5 不难看出,EFSFOA 在 Ionosphere、Cleveland、Dermatology、Heart、Glass、LSVT、CNAE-9、SRBCT、Arcene 这 9 个数据集上均达到了最高.EFSFOA 在 Wine、Sonar、Segmentation、Vehicle 这 4 个数据集中,在 Wine 的 1-NN 分类器上低于 Rc-BBFA;在 Sonar 的 SVM 分类器上低于 HGAFS;在 Segmentation 的 1-NN 分类器上低于 Rc-BBFA;在 Vehicle 的 5-NN 和 SVM 分类器上分别低于 PSO(4-2)和 HGAFS.以上数据集中,EFSFOA 虽然不能保证分类准确率均为最高,但是在决策树分类器上(除 Segmentation 外)均取得了最高的准确率.之所以会产生这种结果,是因为决策树算法是基于信息论理论提出的,而 EFSFOA 在初始化过程中则充分考虑了依据信息增益率进行有针对性的选择,这也从侧面论证了初始化阶段对于算法的后续工作是十分重要的.

对比 FSFOA,EFSFOA 无论是在分类准确率还是在维度缩减率上的提高都是十分明显的.在 Ionosphere 的 3-NN 分类器上,*CA* 值高出 7%,*DR* 值高出 26%;在 Cleveland 的 1-NN 分类器上,*CA* 值高出 6%,*DR* 值高出 13%;在 Sonar 的 5-NN 分类器上,*CA* 值高出 4%,*DR* 值高出 32%;在 Vehicle 的 SVM 分类器上,*CA* 值高出 9%,*DR* 值高出 24%;在 Dermatology 的 DT 分类器上,10-fold 测试时,*CA* 值高出 3%,*DR* 值高出 52%,70%~30%测试时,*CA* 值高出 8%,*DR* 值高出 19%.

对比最近提出的分类准确率较高的算法——Rc-BBFA(该算法最适应于 1-NN 分类器,所以只选取了 1-NN 分类器的实验结果进行对比)可以看出,EFSFOA 在大部分数据集的分类准确率上与 Rc-BBFA 都在伯仲之间,Rc-BBFA 在 Wine,Segmentation 两个数据集上的分类准确率略高于 EFSFOA;EFSFOA 在 Vehicle、LSVT 两个数据集上的分类准确率稍高于 Rc-BBFA.但是值得一提的是,EFSFOA 在 Ionosphere 和 Sonar 上的分类准确率和维度缩减率都优于 Rc-BBFA,其中,准确率平均高出 2%以上,维度缩减率平均高出 20%以上.可见,在这两个数据集上,EFSFOA 的表现还是稍占优势的.

从实验结果可以看出,在高维数据的处理上,EFSFOA 的处理能力显然是优于 FSFOA 甚至是 Rc-BBFA 的.当 Rc-BBFA 处理的特征数量超过 20 000 时,就无法在短时间内给出计算结果(我们设置的时间上限为 24h);而 FSFOA 这个数量仅为 10 000;但 EFSFOA 通过采用集成特征选择的方式对高维数据进行处理,使得特征数量即使达到 100 000 时仍能给出令人满意的解.

EFSFOA 相对于 FSFOA 的几点改进都是围绕提高分类准确率进行优化的,因此带来 *CA* 值上的提高是符合预先设想的,令人比较惊喜的是 EFSFOA 带来的在 *DR* 值上的极大改进(可以看出,FSFOA 的维度缩减率效果并不是十分优秀,甚至在很多时候普遍弱于其他分类器).会造成这种结果的原因除了 EFSFOA 在最后寻优阶段通过进一步贪心获取尽可能高的维度缩减以外,主要原因在于初始化时将具有最高信息增益率的特征赋给了森林中的大部分树(事实上,经过最初的几轮迭代所淘汰的树往往没有选择该特征)和我们在新的适应度函数中充分考虑了维度缩减因素.EFSFOA 在时间上的总体开销要多于 FSFOA,但是由于改进的几点中,只有 *GSC* 参数的自动生成的改进是在算法框架的主要迭代阶段进行的,其余的改进分别针对的是算法的预处理和后处理阶段,因此所增加的额外时间开销主要都应来自于迭代阶段的参数生成过程,而该过程的时间复杂度是 $O(1)$,所以时间上的多余开销并不明显.这也是本文遵从于一般演化计算方法,并未采用时间来衡量算法性能的一个主要原因.

4 总 结

本文通过对 FSFOA 的分析,针对其不足之处和待研究任务分别提出了 3 点优化方案,并推广其工作到高维数据处理上去,形成了一种新算法 EFSFOA.EFSFOA 结合信息论理论给出了一个全新的初始化策略.通过该策略,更高效地指导了算法的后续展开,避免了算法初始化不利所造成的被动局面;在全局播种阶段,通过使用控制函数 *T*,自动地生成了 *GSC* 参数,充分利用了每棵树的年龄特性,使算法的自适应能力得到增强;算法寻优阶段,采用合理的贪心算法,在保证算法不低于原始结果的同时,进一步利用了经过若干轮迭代生成的优质森林中的每一棵树;面向高维数据处理时,针对其高维数据处理能力不足的问题,给出了集成化特征选择的算法框架,同时,综合维度缩减率进一步优化了原有的适应度函数.最后,我们通过涵盖低维、中维、高维的 15 个数据集和

近些年提出的 11 个特征选择算法与 EFSFOA 一起设置了对比实验,经过 1-NN、3-NN、5-NN、SVM、DT 这 5 个分类器的验证发现,EFSFOA 在分类准确率和维度缩减率上均有明显提高,EFSFOA 的特征选择处理能力甚至达到了 100 000 维.在今后的工作中,我们将尝试将 EFSFOA 与一些其他非基于演化计算的特征选择算法进行对比,进一步检验我们的算法的性能,同时还将尝试将我们的算法运用到一些如生物医疗等方面的真实数据集中来进一步优化我们的算法,并对比基于演化计算的特征选择算法在实际的生物医疗数据集中与传统的处理生物医疗数据的方法——基于信息理论的特征选择算法的性能差异.

References:

[1] Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. on Knowledge and Data Engineering, 2005,17(4):491−502.

[2] Oh IS, Lee JS, Moon BR. Hybrid genetic algorithms for feature selection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2004,26(11):1424−1437.

[3] Maldonado S, Weber R. A wrapper method for feature selection using support vector machines. Information Sciences, 2009, 179(13):2208−2217.

[4] Shah SC, Kusiak A. Data mining and genetic algorithm based gene/SNP selection. Artificial Intelligence in Medicine, 2004,31(3): 183−196.

[5] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of Machine Learning Research, 2003,3(6): 1157−1182.

[6] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2005,27(8):1226−1238.

[7] Yu L, Liu H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proc. of the 20th Int'l Conf. on Machine Learning (ICML 2003). AAAI, 2003. 856−863.

[8] Robnik-Šikonja M, Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning, 2003,53(1-2): 23−69.

[9] Gu Q, Han J. Towards feature selection in network. In: Proc. of the 20th ACM Int'l Conf. on Information and Knowledge Management. ACM, 2011. 1175−1184.

[10] Zhao Z, Liu H. Spectral feature selection for supervised and unsupervised learning. In: Proc. of the 24th Int'l Conf. on Machine Learning. ACM, 2007. 1151−1157.

[11] Masaeli M, Yan Y, Cui Y, *et al*. Convex principal feature selection. In: Proc. of the 2010 SIAM Int'l Conf. on Data Mining. SIAM, 2010. 619−628.

[12] Farahat AK, Ghodsi A, Kamel MS. An efficient greedy method for unsupervised feature selection. In: Proc. of the 2011 IEEE 11th Int'l Conf. on Data Mining (ICDM). IEEE, 2011. 161−170.

[13] Efron B, Hastie T, Johnstone I, *et al*. Least angle regression. The Annals of statistics, 2004,32(2):407−499.

[14] Xue B, Zhang M, Browne WN, *et al*. A survey on evolutionary computation approaches to feature selection. IEEE Trans. on Evolutionary Computation, 2016,20(4):606−626.

[15] Zhu W, Si G, Zhang Y, *et al*. Neighborhood effective information ratio for hybrid feature subset evaluation and selection. Neurocomputing, 2013,99:25−37.

[16] Xue B, Zhang M, Browne WN. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing, 2014,18:261−276.

[17] Tabakhi S, Moradi P, Akhlaghian F. An unsupervised feature selection algorithm based on ant colony optimization. Engineering Applications of Artificial Intelligence, 2014,32:112−123.

[18] Zhang Y, Song X, Gong D. A return-cost-based binary firefly algorithm for feature selection. Information Sciences, 2017,418-419: 561−574.

[19] Ghaemi M, Feizi-Derakhshi MR. Feature selection using forest optimization algorithm. Pattern Recognition, 2016,60:121−129.

[20] Chu B, Li ZS, Zhang ML, *et al*. Research on improvements of feature selection using forest optimization algorithm. Journal of Software, 2018,29(9):2547−2558 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5395.htm [doi: 10.13328/j.cnki.jos.005395]

[21] Jadhav S, He H, Jenkins K. Information gain directed genetic algorithm wrapper feature selection for credit rating. Applied Soft Computing, 2018,69:541−553.

[22] Pereira RB, Plastino A, Zadrozny B, *et al*. Information gain feature selection for multi-label classification. Journal of Information and Data Management, 2015,6(1):48−58.

[23] Yiğit F, Baykan ÖK. A new feature selection method for text categorization based on information gain and particle swarm optimization. In: Proc. of the 2014 IEEE 3rd Int'l Conf. on Cloud Computing and Intelligence Systems (CCIS). IEEE, 2014. 523−529.

[24] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science, New Series, 1983,220(4598):671−680.

[25] Dua D, Graff C. UCI machine learning repository. Irvine: School of Information and Computer Science, University of California, 2017. http://archive.ics.uci.edu/ml

[26] Ghaemi M, Feizi-Derakhshi MR. Forest optimization algorithm. Expert Systems with Applications, 2014,41(15):6676−6687.

[27] Cai J, Luo J, Wang S, *et al*. Feature selection in machine learning: A new perspective. Neurocomputing, 2018,300:70−79.

[28] Moustakidis SP, Theocharis JB. SVM-FuzCoC: A novel SVM-based feature selection method using a fuzzy complementary criterion. Pattern Recognition, 2010,43(11):3712−3729.

[29] Hu Q, Che X, Zhang L, *et al*. Feature evaluation and selection based on neighborhood soft margin. Neurocomputing, 2010, 73(10-12):2114−2124.

[30] Huang J, Cai Y, Xu X. A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recognition Letters, 2007,28(13):1825−1844.

附中文参考文献:

[20] 初蓓,李占山,张梦林,等.基于森林优化特征选择算法的改进研究.软件学报,2018,29(9):2547−2558. http://www.jos.org.cn/1000-9825/5395.htm [doi: 10.13328/j.cnki.jos.005395]

刘兆赓(1993－),山东沂水人,男,硕士生,主要研究领域为机器学习.

李占山(1966－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为约束优化与约束求解,机器学习,基于模型的诊断,智能规划与调度.

王丽(1994－),女,硕士生,主要研究领域为机器学习.

王涛(1969－),女,副教授,主要研究领域为约束优化与约束求解,机器学习.

于海鸿(1975－),男,博士,讲师,主要研究领域为约束优化与约束求解,大数据与数据挖掘,智能规划与调度.