

自然进化策略的特征选择算法研究*

张鑫^{1,2}, 李占山^{1,2}



¹(吉林大学 计算机科学与技术学院, 吉林 长春 130012)

²(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: 李占山, E-mail: zslizsli@163.com

摘要: 特征选择是一种 NP-难问题,旨在剔除数据集中不相关及冗余的特征来减少模型训练的时间,提高模型的精确度.因此,特征选择在机器学习、数据挖掘和模式识别等领域中是一种重要的数据预处理手段.提出一种新的基于自然进化策略的特征选择算法——MCC-NES.首先,算法采用了基于对角协方差矩阵建模并通过梯度信息自适应调整参数的自然进化策略;其次,为了使算法有效地处理特征选择问题,在初始化阶段引入了一种特征编码方式;之后,结合分类准确率和维度缩减给出了算法的适应度函数;此外,面对高维数据引入了合作协同进化的思想,将原问题分解为相对较小的子问题并分别对每个子问题独立求解,然后,通过所有子问题相互联系来优化原问题的解决方案;进一步引入分布式种群进化的概念,实现多个种群竞争进化来增加算法的探索能力,并设计了种群重启策略以防止种群陷入局部最优解.最后将提出的算法与几种传统的特征选择算法在一些 UCI 公共数据集上进行对比实验,实验结果显示:所提出的算法可以有效地完成特征选择问题,并且与经典特征选择算法相比有一定的竞争力,尤其是在处理高维数据时有着出色的表现.

关键词: 进化策略;特征选择;合作协同进化;竞争进化;高维

中图法分类号: TP18

中文引用格式: 张鑫,李占山.自然进化策略的特征选择算法研究.软件学报,2020,31(12):3733-3752. <http://www.jos.org.cn/1000-9825/5874.htm>

英文引用格式: Zhang X, Li ZS. Research on feature selection algorithm based on natural evolution strategy. Ruan Jian Xue Bao/ Journal of Software, 2020, 31(12): 3733-3752 (in Chinese). <http://www.jos.org.cn/1000-9825/5874.htm>

Research on Feature Selection Algorithm Based on Natural Evolution Strategy

ZHANG Xin^{1,2}, LI Zhan-Shan^{1,2}

¹(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

²(Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012, China)

Abstract: Feature selection is an NP-hard problem that aims to improve the accuracy of the model by eliminating irrelevant or redundant features to reduce model training time. Therefore, feature selection is an important data preprocessing technique in the fields of machine learning, data mining, and pattern recognition. This study proposes a new feature selection algorithm MCC-NES based on natural evolutionary strategy. Firstly, the algorithm adopts natural evolutionary strategy based on diagonal covariance matrix modeling, which adaptively adjusts parameters through gradient information. Secondly, in order to enable the algorithm to effectively deal with feature selection problems, a feature coding mechanism is introduced in the initialization phase, and combined with classification accuracy and

* 基金项目: 国家自然科学基金(61672261); 吉林省自然科学基金(20180101043JC); 吉林省发展和改革委员会产业技术研究与开发项目(2019C053-9)

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

收稿时间: 2018-12-07; 修改时间: 2019-02-25, 2019-04-07, 2019-06-17; 采用时间: 2019-07-08

dimensional reduction, given the new fitness function. In addition, the idea of sub-population cooperative co-evolution is introduced to solve high-dimensional data. The original problem is decomposed into relatively small sub-problems to reduce the combined effect of the original problem scale and each sub-question is solved independently, and then all sub-problems are correlated to optimize the solution to the original problem. Further, applying multiple competing evolutionary populations to enhance the exploration ability of the algorithm and designing a population restart strategy to prevent the population from falling into the local optimal solution. Finally, the proposed algorithm is compared with several traditional feature selection algorithms on some UCI public datasets. The experimental results show that the proposed algorithm can effectively complete the feature selection problem and has excellent performance compared with the classical feature selection algorithm, especially when dealing with high-dimensional data.

Key words: evolution strategy; feature selection; cooperative co-evolution; competitive evolution; high-dimensional

特征选择在机器学习、数据挖掘和模式识别等领域中有着重要的应用.由于数据收集技术的不断完善,一些专业领域中数据量不断积累,导致在机器学习和数据挖掘中要处理的数据特征总数变大.在这些特征中,选取能够描述目标概念的特征子集就尤为重要.特征选择就是从原始特征集中选取有效的特征子集来减少数据特征维数的过程^[1].因此,特征选择自然就成为了机器学习和数据挖掘领域中一个重要的数据预处理手段^[2].

特征选择算法有很多种,总的来讲可以分为3类:过滤式方法、包裹式方法以及嵌入式方法^[3,4].过滤式方法不依赖于其他学习算法,首先将数据特征按照一定的标准进行排序,然后选择那些表现比较好的特征,由于这种方式具有很高的计算效率^[1],因此普遍被用来处理一些高维数据;包裹式方法通过引入一个学习算法来评估所选的特征子集^[5],因为直接对目标进行优化,所以包裹式方法具有更好的性能;嵌入式方法试图将特征选择和分类器学习整合在同一个过程中,从而综合了包裹式方法和过滤式方法的优点.

特征选择在分类任务是 NP-hard 的,由于搜索空间会随着特征个数的增加呈指数级增长.对于包含 n 个特征的实例,其搜索空间就会有 2^n 个可能的结果^[4,6].关于特征选择的方法已经进行了许多工作,根据产生特征子集所采用搜索策略的不同,特征选择方法可以分为穷举法、启发法和随机法.

穷举法试图枚举搜索空间中的解来寻找一个能够正确区分所有类别的最小特征集合.当数据特征个数较多时,评价所有特征子集在计算上是不可行的,因此在包含大量特征的数据集中,穷举法很少用于特征选择.

启发式特征选择算法主要包括贪婪爬山算法^[7]、分支限界法、定向搜索以及最佳优先算法.SFS(sequential forward selection)和 SBS(sequential backward selection)是两种经典的爬山算法,但这两种方法容易陷入局部最优解.为克服这一缺点,SFFS 算法、SBFS 算法被提出^[8].另外,基于支持向量机并应用启发式搜索策略的特征选择方法是近年发展起来的通用方法,它本身是一种有监督学习算法.Moustakidis 等人提出了一种新的基于模糊互补准则的支持向量机特征选择方法 SVM-FuzCoC^[9].

随机法的主要优点是具有可接受的时间复杂性.特别是一些自然启发式算法,由于其出色的全局搜索能力在特征选择问题中引起了大量的关注^[10].具体的,Zhu 等人将遗传算法和局部搜索方法结合起来,提出了 FS-NEIR 算法^[11].同年,Xue 等人提出了基于粒子群优化的特征选择算法 PSO^[12].Tabakhi 等人提出了基于蚁群优化的无监督特征选择算法 UFSACO^[13].近年来,关于演化算法在特征选择问题中的应用也有很多,例如 Ghaemi 等人提出的基于森林优化算法的特征选择方法 FSFOA^[14]、Zhang 等人基于萤火虫算法提出的 Rc-BBFA^[15]、Marwa 等人提出的混合过滤和包裹方法的多目标演化特征选择算法 FW-NSGA-II^[16]、Hancer 等人基于差分进化提出的特征选择方法 DE-FLS^[17]、Chen 等人基于细菌觅食算法提出 ISEDBFO^[18]及 ACBFO^[18]、Qian 等人基于进化帕累托优化提出的 POSS 算法^[19].之后,Qian 等人又提出了基于帕累托优化的无监督特征选择算法 POCSS^[20].POCSS 采用随机迭代过程来最小化重建误差和所选择的特征数量.此外,基于多种群的协同演化算法也被用于处理特征选择问题,例如 J.Derrac 等人提出的基于协同进化算法的特征选择方法 IFS-CoCo^[21]等.

相比于传统的特征选择方法,基于演化计算方法的明显优势在于其通常不需要领域内的相关知识和对搜索空间做任何的假设,并且由于它基于种群机制,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.因此,将演化计算引入特征选择取得了不错的效果.尽管如此,目前演化计算方法仍存在一些不足之处.

- 首先,演化计算过程中随机性较高,算法每次运行产生的结果不能够保证,并且算法在迭代中不能够准

确地控制种群的搜索范围和强度,这使得算法的鲁棒性不足;

- 其次,演化计算方法是通过对个体在搜索空间进行探索来寻求最优解,当搜索空间比较大时,算法容易陷入局部最优解,这导致演化计算方法对于高维数据的处理能力不足。

因此,特征选择算法上仍存在较多可改进和完善之处,特别是在处理高维数据的特征选择问题上。

作为进化算法的一种,进化策略(ES)^[22,23]在一些强化学习任务中的成功案例证明了进化策略能够有效适用于评价学习^[24]。评价学习不同于监督学习和非监督学习,算法通常需要从一些离散的、有噪音的、延迟的标量信息中学习。作为强化学习中一种重要的学习算法,策略迭代通常会使用神经网络作为决策机制^[25],并使用误差反向传播来优化策略函数。文献[24]中替代了策略迭代中误差反向传播优化参数的方式,直接使用进化策略对决策网络进行优化,最后达到了预期的目标效果。

包裹式的特征选择方法首先选取特征子集,然后使用学习算法对其进行评估,因此,特征选择也可以看作是一种评价学习。受此启发,本文使用 ES 作为全局搜索算法进行特征选择,之后的工作证明了 ES 能够完成特征选择任务,同时,与其他的特征选择方法相比也有一定的竞争力,并且能够有效地以分布式的方式进行多种群竞争进化^[26]。

本文算法的具体实现过程为:在自然进化策略的基础上,通过采用一种新的编码方式使其能够适应于特征选择问题;然后,结合模型分类准确率和特征维度缩减设计了一种新的适应度函数;同时,基于新的适应度函数提出一种适应度塑造方法用于构造目标优化函数;之后,为了有效处理高维数据,在算法模型中引入亚种群合作协同进化的思想;更进一步,以多种群分布式竞争进化的方式加速算法的收敛,并设计种群重启策略来防止种群陷入局部最优解;最后,使用 GPU-Tensorflow 作为算法运行框架实现多种群分布式进化。我们将提出的算法记为 MCC-NES(multi-group cooperative coevolution feature selection algorithm based on natural evolution strategy)。同时,我们将本文提出的算法与近几年提出的其他比较高效的特征选择算法在 16 个 UCI^[27]的数据集中进行了对比实验,验证了我们算法的有效性和优越性,尤其是在处理一些高维数据时有出色的表现。具体体现在:

首先,本文的方法可以高度并行化,通过引入特定的通信机制,可以使得多个工作线程在运行时并行加速;其次,本文提出的方法表现出一种很好的探索能力,因为在已知均值和方差的情况下,正态分布是一种信息熵最大、最不稳定的分布,算法在迭代中,通过正态分布生成搜索点能够保证算法对于搜索空间进行有效的探索;最后,本文的方法有很好的鲁棒性,算法能够根据搜索情况自适应改变分布均值和方差来调整搜索的强度和范围。

1 相关工作

1.1 特征选择问题

特征选择问题可以描述为:用 S 表示 K 个样本 D 个特征的数据集, F 为 D 个特征的集合。特征选择任务就是从集合 F 中选取 d 个特征,其中, $d \leq D$,使得目标函数 $f(\cdot)$ 取最大值,这里的目标函数一般表示为模型分类准确率。也就是说,特征选择的目的是通过选取最少特征数的特征子集 $X(X \in F)$ 来使目标函数最大化。

我们采用如下二进制编码方式来表示选取的特征子集:

$$X_i = (x_1^i, \dots, x_D^i), x_j^i \in \{0, 1\} \quad (1)$$

其中, $x_j^i = 1$ 表示第 i 个特征子集 X_i 中第 j 个特征被选取;相应的, $x_j^i = 0$ 表示这个特征不会被选取。那么,特征选择问题就可以表示成如下的优化问题^[15]:

$$\begin{cases} \max f(X) \\ \text{s. t. } X_i = (x_1^i, \dots, x_D^i), x_j^i \in \{0, 1\}, j = 1, 2, \dots, D \\ 1 \leq |X| \leq D \end{cases} \quad (2)$$

其中, $|X|$ 表示在特征子集 X 中所选取的特征数量。

1.2 进化策略

进化策略是一种黑盒优化算法,首次由 Ingo 等人提出^[28,29],并被设计用来处理高维连续性优化问题。如文献

[30]所述,ES 是自然启发优化算法的一种,那么 ES 中同样包含基因突变、重组以及选取候选解的过程,并以迭代的方式进化越来越好的解.进化策略中包含一个用于评估种群个体基因序列的适应度函数,每次选取候选解的时候,最好的个体将会被保留,其他的个体直接舍去.然后,在保留下来的个体中,通过扰动基因序列来生成下一代的个体,整个过程通过迭代就会找到较优的结果.

其中,公式(3)为给定优化问题的一般形式,那么进化策略可以应用于优化问题的所有领域,包括无约束和有约束以及混合约束的搜索空间 Y 中:

$$y^* = \operatorname{argopt}_{y \in Y} f(y) \quad (3)$$

进化策略又有两种候选解选取策略: (μ, λ) -ES 和 $(\mu + \lambda)$ -ES.这两种策略的不同之处在于:前者每次迭代产生 λ 个新解,选择较好的 μ 个成为下一次迭代的父代($\mu \leq \lambda$),其他的直接舍去,其中,每个个体只存活一代,随即被新的个体代替;而后者每次迭代产生 λ 个新解,通过和父代中的个体进行比较,选择较好的 μ 个成为下一次迭代的父代,其他的直接舍去.本文采用第一种候选解选取策略,因为前者能够更有效地对搜索空间进行探索而不容易陷入局部最优.ES 中每个个体 x 的一般形式为

$$\text{ES individual } x := (y, \theta, F(y)) \quad (4)$$

其中, $y \in Y$ 表示需要优化的目标序列; θ 表示策略参数,并且 θ 中相关参数在 ES 算法中会自适应的进行调整; $F(\cdot)$ 为目标函数.使用这种表述形式,特征选择问题(2)也可以看作是进化策略的一种优化问题进行求解.

1.3 进化策略的框架

近年来,进化策略的框架已被广泛开发,包括使用完全协方差矩阵作为分布模型表示相关突变,这使得算法框架在变异生成下一代个体时能够利用属性之间的协方差关系来捕获它们的相关性,因此,突变可以自适应局部搜索情况,并且可以显著增加向最优解的收敛性能.协方差矩阵自适应进化策略(CMA-ES)^[31]已在多项研究中证明是成功的^[32-34].

我们的工作是基于自然进化策略(NES)^[35,36]展开的,NES 也是进化策略在框架上的一种扩展.NES 通过使用样本的估计梯度来迭代地更新搜索分布.算法过程如下:搜索分布采样用于生成一批搜索点(即每代种群),然后根据适应度函数评估每一个搜索点(种群中的每个个体),之后计算搜索点的搜索梯度并使用梯度上升的方法,通过最大化种群适应度均值来自适应地调整模型中相关的参数.

NES 的核心就是使用搜索梯度^[37,38]来更新搜索分布中的参数.我们将搜索梯度定义为样本预期适应度的梯度,其中,搜索分布采用多元正态分布.如果用 $\pi(x|\theta)$ 表示参数为 θ 的分布策略,用 $f(\cdot)$ 表示的样本适应度函数,我们可以将搜索分布下的预期适应度表示如下:

$$J(\theta) = E_{\theta}[f(x)] = \int f(x)\pi(x|\theta)dx \quad (5)$$

然后对公式(5)求导,并采用对数似然表示:

$$\nabla_{\theta} J(\theta) = E_{\theta}[f(x)] \nabla_{\theta} \log \pi(x|\theta) \quad (6)$$

使用公式(6),我们可以从种群个体 $(x_1, x_2, \dots, x_{\lambda})$ 得到搜索梯度的估计值:

$$\nabla_{\theta} J(\theta) \approx \frac{1}{\lambda} \sum_{i=1}^{\lambda} f(x_i) \nabla_{\theta} \log \pi(x_i|\theta) \quad (7)$$

公式(7)中, λ 表示种群的大小,样本预期梯度在搜索分布空间中提供搜索方向,采用梯度上升来迭代更新搜索分布:

$$\theta \leftarrow \theta + \eta \nabla_{\theta} J(\theta) \quad (8)$$

公式(8)中, η 表示学习率,算法 1 表示在搜索分布上使用搜索梯度.

算法 1. Search-gradient-algorithm.

Initialize parent population $p_{\mu}^0 = \{x_1, \dots, x_{\mu}\}$

Initialize distribution parameters θ^0 , fitness function $f(\cdot)$

Repeat

For generation $g=1,2,\dots,\tau$ **do**

Sample λ independent points from distribution $p(x|\theta^g) \rightarrow x_1, \dots, x_\lambda$

Evaluate the sample (x_1, \dots, x_λ) on $f(\cdot)$

Calculate log-derivatives $\nabla_\theta \log \pi(x|\theta)$

End for

$$\nabla_\theta J(\theta) \leftarrow \frac{1}{\lambda} \sum_{g=1}^{\lambda} f(x_g) \nabla_\theta \log \pi(x_g | \theta)$$

$$F \leftarrow \frac{1}{\lambda} \sum_{g=1}^{\lambda} \nabla_\theta \log \pi(x_g | \theta) \nabla_\theta \log \pi(x_g | \theta)^T$$

$$\theta \leftarrow \theta + \eta F^{-1} \nabla_\theta J(\theta)$$

Until termination criterion met

NES 将普通随机梯度替换为自然梯度,自然梯度首先被 Amari 引入机器学习领域,并且已被证明自然梯度比普通梯度更有优势^[39,40].自然梯度能够加速梯度在平原地区的收敛速率,并减缓梯度在高原地区的收敛速率.

2 MCC-NES

ES 能够有效地进行评价学习,并且具有良好的全局搜索能力和并行化能力.进一步地,NES 可以通过使用样本梯度信息来更新参数.由于 NES 需要处理梯度,我们使用 GPU-Tensorflow 作为 NES 运行框架,同时引入一些优化策略,提出了一种新的特征选择算法(MCC-NES).之后,通过实验验证 MCC-NES 算法在特征选择问题中的性能.实验结果表明:MCC-NES 算法能够有效地完成特征选择问题,而且在处理高维数据时有着出色的表现.

2.1 MCC-NES算法建模及特征编码

结合前文分析,本文的工作采用了基于协方差矩阵建模的自然进化策略.分布模型中的参数包括分布均值向量 m 和分布协方差矩阵 C ,其中, $|m| = \text{rank}(C)$ 表示数据集中特征的个数.当特征个数较小时,对完全协方差矩阵建模是非常有益的.但是面对高维数据时,使用完全协方差矩阵复杂度就会很高.因为协方差矩阵的大小和数据特征维数相关,对于 n 维数据来说,完全协方差矩阵中元素的个数就会有 n^2 个.当 n 取值增大时,其计算复杂度就会呈指数级增长.因此,为了有效处理高维数据,我们选择保留矩阵主要信息,只关心每个特征的离散性而不去考虑特征之间的相关性,使用对角协方差矩阵代替完全协方差矩阵进行建模,从而降低算法时间复杂度.

在算法的初始化阶段,通过在超参数 α 上重复添加高斯扰动(扰动强度 $\sigma=0.01$)来初始化分布均值向量 m ,对于分布协方差矩阵 C ,选择使用单位矩阵进行初始化,使得算法模型中每个特征对应于分布均值向量 m 中的一个元素和分布协方差矩阵 C 中的一个对角元素.在算法迭代时,将向量 m 中的每个元素与矩阵 C 中对应的对角元素作为正态分布的参数做高斯采样,生成关于每个特征的分布取值,然后,将得到的数值结果通过使用一种新的编码方式映射到布尔型的特征选择上,具体的特征编码方式如下:

$$x_i^g = \begin{cases} 1, & \frac{1}{1 + e^{-y_i^g * 2}} > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

ES 算法一般用于解决连续型优化问题,为了处理特征选择问题,文中使用公式(9)将特征高斯采样的结果映射到 $\{0,1\}$ 序列中,其中, y_i^g 表示在第 g 次迭代中第 i 个特征的高斯采样结果.使用公式(9)进行特征映射,如图 1 所示.若映射后的结果大于 0.5,就选取该特征(用二进制 1 表示选取该特征);否则的话,就不进行选择(用二进制 0 来表示不选取该特征).

通过这种编码方式,我们希望在迭代过程中对于那些已经确定的特征尽量使其保持稳定(在图 1 中表现为两端较为平缓的部分),而对于不确定的特征提供更多的探索(在图 1 中表现为中间部分的曲线,有较高的斜率,对于梯度信息更为敏感).

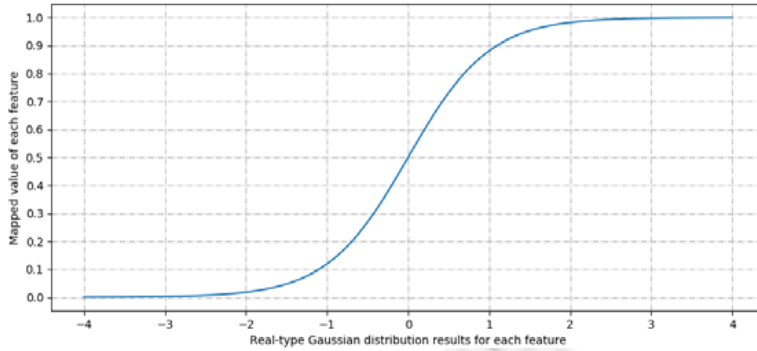


Fig.1 Feature coding function

图 1 特征编码函数

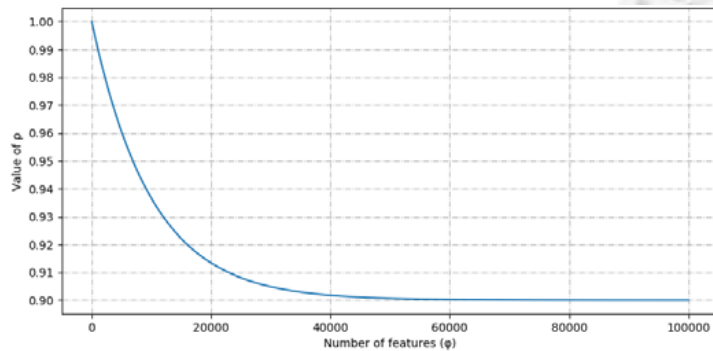
2.2 新的适应度函数及适应度塑造方法

在计算样本适应度时,传统的适应度函数具有潜在的局限性.一般而言,大多数算法中对于样本的评估都是基于测试集的分类准确率的,也就是说,每一次更新都会选取表现最好的一些特征序列,因此就会忽略维度缩减的重要性.但是有时候在处理一些高维数据的时候,能够选择更少的特征也是具有实际应用价值的,所以本文就传统适应度函数的弊端提出了一个新的适应度函数,将特征缩减也加入特征选取的考虑范围.

新的适应度函数如公式(10)所示:

$$\begin{cases} f(x_i) = \rho \cdot CA + (1 - \rho) \cdot DR \\ \rho = \frac{9 + 0.99 \left(\frac{\varphi}{100} \right)}{10} \end{cases} \quad (10)$$

φ 表示数据特征维数;关于参数 ρ 的取值与数据特征个数相关,其函数图像由图 2 给出;式中 CA 和 DR 为分类准确率和维度缩减率(CA 和 DR 的定义在第 3.1 中进行详细说明).

Fig.2 Curve of the parameter ρ about φ 图 2 参数 ρ 关于特征维数 φ 的曲线

如图 2 所示:当数据特征维数较小时, ρ 的取值接近于 1,适应度函数的考察主要以分类准确率为主;而随着特征维数增加, ρ 取值减小,直至接近 0.9,维度缩减的重要性也随之提高.因此在新的适应度函数下,算法既考虑了模型分类准确性,也考虑了特征维度缩减,但是分类准确率依然是最主要的.

进一步地,为了防止一些异常的适应度值导致不准确的梯度信息而增加算法陷入局部最优解的可能,我们使用了一种适应度塑造的方法,如公式(11)所示,将样本的真实适应度 $f(x_i)$ 依据排名映射到一个均匀分布的离散空间中作为样本的有效适应度 $u(x_i)$ 来求解梯度,这使得梯度信息不依赖于具体的适应度值.同时,我们提出了一

种自适应调整分布空间的方式,就是在每次迭代中,根据样本中最优的适应度值与最差的适应度值之间的差距来调整分布空间的取值范围,这使得当种群中个体表现差异比较大的时候,能够增加参数调整的幅度;而当种群中个体表现相似时,减小参数调整的步长:

$$\begin{cases} u(x_i) = u(x_1) + (i-1)\delta \\ \delta = \frac{f(x_\lambda) - f(x_1)}{\lambda - 1}, f(x_{1,\lambda}) \leq \dots \leq f(x_{\lambda,\lambda}) \\ u(x_1) = -\frac{f(x_\lambda) - f(x_1)}{2} \end{cases} \quad (11)$$

2.3 种群内的亚种群合作协同进化

我们使用 ES 算法解决特征选择问题最直接的想法就是基于数据集中所有的特征进行建模,但是这种建模方式在处理一些高维数据时就会显现出弊端(因为搜索空间随数据维度呈指数级增长,由于计算资源的限制无法完成矩阵建模).因此,我们在算法中引入了亚种群合作协同进化的思想^[41].合作协同进化通过将原始特征集划分成相对较小的子特征集,然后在每个子特征集中独立应用进化策略进行局部搜索,子特征集完成搜索后,将通过合作协同进化来组合优化原特征选择问题的解决方案.

具体的,首先对数据集 D 按照特征随机划分成若干组 $A = \{A_1, \dots, A_n\}$, 各组包含的特征相互独立且唯一,然后初始化 n 个对角协方差矩阵 $C = \{C_1, \dots, C_n\}$ 、 n 个分布均值向量 $M = \{m_1, \dots, m_n\}$ 以及 n 个亚种群 $S = \{S_1, \dots, S_n\}$, 并使每个亚种群 S_i 对应一个子特征子集 A_i 、一个均值向量 m_i 以及一个对角协方差矩阵 $C_i (|A_i| = |S_i| = |m_i| = \text{rank}(C_i))$. 初始化完成之后,让 n 个亚种群同步进行迭代更新.由于每个亚种群只维护一个子特征集的信息,而原特征选择问题需要在原始特征集中进行全局搜索,因此需要在每次迭代中,在 n 个亚种群间进行信息交互.具体的,每次迭代时,每个亚种群随机产生 m 个搜索点,对于 n 个亚种群来讲,就会有 m^n 个子特征序列的组合,如果完整评估所有特征序列的组合,当 n 和 m 增大时,很容易陷入组合爆炸.不失一般性,我们保留一定的历史信息,在组合的过程中,我们保留种群历史表现最好的全局特征序列,每个亚种群的搜索就可以看成是对当前全局最优特征序列的一次局部探索,每个亚种群产生的搜索点只改变当前亚种群在全局最优特征序列中所对应的这一部分特征,同时对局部改动的全局最优特征序列进行评估,作为对当前亚种群采样的评估.如此,通过保留历史最优全局特征序列,就会极大地降低算法搜索复杂度 $O(m^n)$.进一步地,为了防止搜索过程中陷入局部最优解的情况,每次迭代时选择保留表现排名前三的全局特征序列和一个随机产生的全局特征序列.这里,随机特征序列的产生是由当前全局最优特征序列决定的,根据当前全局最优特征序列中特征的个数,随机地在原始特征集中选取相同数量的特征构成随机特征序列,然后,每次迭代时,每个亚种群同时在这 4 个特征序列上进行局部搜索.以上过程是在一个种群中,通过亚种群协同进化的方式来实现种群优化的完整过程.

协同进化是一种进化算法,算法一般通过种群间相互激励来提升种群各自的性能.如在 IFS-CoCo^[21]算法中,通过特征选择与实例选取协同进化来提升分类器准确率.我们所采用的方式与一般的协同进化方法不同:它类似于分治法的思想,将原始集合划分为多个分组,通过分组与原始集合间协同进化来指导搜索.具体,算法在迭代过程中,每个分组采用 ES 算法进行搜索,在分组进化过程中,我们会保留历史前三全局最优序列,同时使用一个全局随机序列,每个分组在进化过程中都会和这 4 个全局序列进行交互,并更新全局最优序列.通过这种方式,一方面为了防止算法陷入局部最优解,同时解决分组间组合爆炸的问题.

2.4 多种群间竞争进化及种群重启策略

第 2、3 节详细描述了一个种群使用合作协同进化解特征选择问题的完整过程,进一步的,我们通过分布式操作将这个过程扩展到多个种群实现多种群竞争进化.同时,为了防止种群陷入局部最优解,提出了一种种群重启策略:

因为在算法初始阶段种群采用随机初始化,所以一个种群在迭代中的收敛速率和性能都很依赖于种群在搜索空间中的初始化位置.引入一种种群重启机制,当一个种群的整体性能表现不好或者当种群性能上升停滞

时,就会采取相应的重启策略,包括重新初始化该种群的分协方差矩阵 C ,并将其分布均值 m 向最优种群的分布均值转移.算法采用这种重启机制来使种群跳出局部最优解,不断将表现差的种群向表现好的种群方向转移,并使其在当前最优解附近进行新的探索.之后的实验中也验证了这种方式的有效性.

种群性能评价函数定义为

$$F(p_i^g) = (1 - \beta)\nabla f(p_i^g) + \beta F(p_i^{g-1}) \quad (12)$$

其中, $\nabla f(p_i^g)$ 表示种群 p_i 在第 g 次迭代时的适应度变化量(使用当前样本适应度均值和上一代样本适应度均值之差表示), β 为超参数表示历史信息的利用率.这种设定是为了综合考虑种群的历史累积表现.同时,种群重启概率设计如下:

$$\begin{cases} \text{True, } 0.01 + 0.99 \cdot \frac{e^{10 \cdot \frac{\tilde{g}-1}{\omega}} - 1}{e^{10} - 1} > \text{rand} \\ \text{false, otherwise} \end{cases} \quad (13)$$

其中, \tilde{g} 表示当前迭代次数与上一次策略重启时迭代次数之差; ω 是一个超参数,当种群 p_i 使用公式(13)返回为 True 并且其性能 $F(p_i^g)$ 在所有种群中最小时,种群 p_i 就会进行重启,并将 \tilde{g} 重置为 0.

公式(13)的概率曲线如图 3 所示.

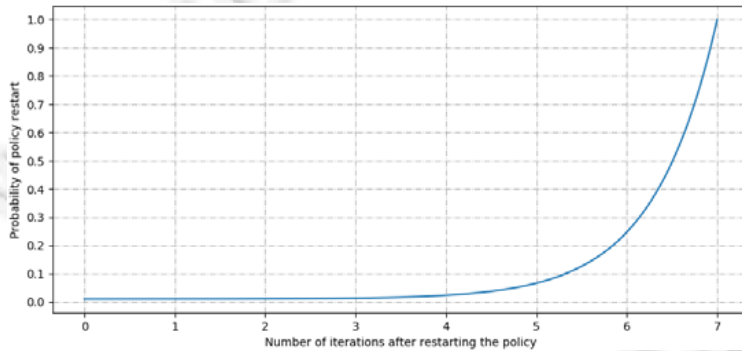


Fig.3 Restart probability curve

图 3 种群重启概率曲线

由于使用多种群竞争进化提供了更多的探索性,所以极大地提升了算法的探索能力,之后的实验中也验证了这一观点.

2.5 MCC-NES算法伪代码

基于以上分析,MCC-NES 的伪代码在算法 2 中给出.其中:第 1 行~第 3 行是关于算法参数、适应度函数以及适应度塑造方法的初始化过程;第 5 行完成每个种群中的 3 个最优全局特征序列和 1 个随机全局特征序列的初始化工作;第 6 行是将原始特征集合根据设定的大小进行划分;第 7 行~第 9 行初始化每个种群中每个亚种群的分布参数;第 12 行开始算法的迭代过程;第 14 行~第 37 行表示每个种群的进化过程,具体的,第 15 行~第 36 行是每个亚种群的进化操作,其中包括分布采样(第 16 行、第 17 行)、生成种群随机特征序列(第 18 行)、评估每个样本适应度(第 19 行~第 31 行)、实现亚种群的更新操作,使用样本的搜索梯度来更新分布参数(第 32 行~第 35 行).第 38 行~41 行检测是否有种群满足重启条件,如果存在满足条件的种群,就对该种群进行重启操作.

算法 2. MCC-NES.

- 1: Input: The number of population σ , the size of subpopulation m , the number of individuals in subpopulation λ , the number of iterations τ , the Initialization parameters α , restart threshold ω , historical utilization β ;
- 2: Initialize: fitnessfunction $f(\cdot)$ as equation 10, fitnessfunction $u(\cdot)$ as equation 11, set \tilde{g} to 0

- 3: Initialize: the number of population genes φ
- 4: **For** population $p=1,2,\dots,\sigma$ **do**
- 5: Initialize population feature sequence with zero $\xi_p^1, \xi_p^2, \xi_p^3, \xi_p^4$
- 6: Randomly divide the population gene into m Subpopulation
- 7: **For** subpopulation $s=1,2,\dots,m$ **do**
- 8: Initialize distribution parameters $\theta_{p,s}^{(0)}$
- 9: **End for**
- 10: **End for**
- 11: **Repeat**
- 12: **For** generation $g=1,2,\dots,\tau$ **do**
- 13: $\tilde{g} \leftarrow \tilde{g} + 1$
- 14: **For** population $p=1,2,\dots,\sigma$ **do**
- 15: **For each** subpopulation $s=1,2,\dots,m$ **do**
- 16: Sample λ individual from distribution $p(x|\theta_{p,s}^{(g)}) \rightarrow x_1, \dots, x_\lambda$
- 17: Feature selection for generated individuals according to Eq.9
- 18: Generating random sequences ξ_p^4 according ξ_p^3
- 19: **For each** individual $\mu=1,2,\dots,\lambda$ **do**
- 20: Local exploration on four feature sequences $\xi_p^1, \xi_p^2, \xi_p^3, \xi_p^4$ to generates four new feature sequences
 $\tilde{\xi}_p^1, \tilde{\xi}_p^2, \tilde{\xi}_p^3, \tilde{\xi}_p^4$
- 21: Evaluate four new feature sequences $\tilde{\xi}_p^1, \tilde{\xi}_p^2, \tilde{\xi}_p^3, \tilde{\xi}_p^4$ using fitness function $f(\cdot)$
- 22: Choose best fitness value on four feature sequences as individual fitness
- 23: **For each** new feature sequences $\tilde{\xi}_p^i (1 \leq i \leq 4)$ **do**
- 24: **If** $f(\tilde{\xi}_p^i) > f(\xi_p^3)$ **do**
- 25: $\xi_p^1 \leftarrow \xi_p^2, \xi_p^2 \leftarrow \xi_p^3, \xi_p^3 \leftarrow \tilde{\xi}_p^i$
- 26: **Else If** $f(\tilde{\xi}_p^i) > f(\xi_p^2)$ **do**
- 27: $\xi_p^1 \leftarrow \xi_p^2, \xi_p^2 \leftarrow \tilde{\xi}_p^i$
- 28: **Else if** $f(\tilde{\xi}_p^i) > f(\xi_p^1)$ **do**
- 29: $\xi_p^1 \leftarrow \tilde{\xi}_p^i$
- 30: **End for**
- 31: **End for**
- 32: Fitness shaping for each individual fitness value using function $u(\cdot)$
- 33: Calculate log-derivatives $\nabla_\theta \log \pi(x|\theta)$
- 34: $\nabla_\theta J(\theta) \leftarrow \frac{1}{\lambda} \sum_{g=1}^{\lambda} u(x_g) \nabla_\theta \log \pi(x_g | \theta)$
- 35: Update distribution parameters using Adam
- 36: **End for**
- 37: **End for**
- 38: Evaluate all populations performance $F(p_i) (i=1,2,\dots,\sigma)$ by Equ.12
- 39: **IF** Eq.13 returns true and $F(p_i)$ has the worst overall performance **do**
- 40: Reset the exploration ability of p_i , Transfer the distribution of p_i to the optimal population and reset

\tilde{g} to 0

41: **End If**

42: **End for**

43: **Until** termination criterion met

3 实验结果与分析

在实验中,我们使用公开的机器学习工具包 scikit-learn.编程语言为 python.实验电脑的配置是 Intel Core i7 CPU(3.60GHz),8G 内存;使用显卡为 NVIDIA GeForce GTX 1050.

3.1 数据集

我们在 16 个 UCI 公开数据集上对 MCC-NES 算法的性能予以测试,涵盖了 6 个小维数据集(Glass,Heart,Cleveland,Wine,Vehicle,Segmentatin)、2 个中维数据集(Ionosphere,Dermatology)和 8 个高维数据集(Spambase,Sonar,Musk2,LSVT,SRBCT,Arcene,RNA-Seq,Dorothea).根据文献[42],如果数据集中的特征数量是[0,19],[20,49],[50,∞],那么对应数据集的规模分别是小维数据集、中维数据集、高维数据集.关于数据集的具体信息由表 1 给出.

Table 1 Specific information of the selected datasets

表 1 数据集详细信息

Dataset	Feature	Instance	Class
Glass	9	214	7
Heart	13	270	2
Cleveland	13	303	5
Wine	13	178	3
Vehicle	18	846	4
Segmentatin	19	2 310	7
Ionosphere	34	351	2
Dermatology	34	366	6
Spambase	57	4 061	2
Sonar	60	208	2
Musk2	166	6 598	2
LSVT	310	126	2
SRBCT	2 308	63	4
Arcene	10 000	200	2
RNA-Seq	20 531	801	5
Dorothea	100 000	800	2

根据对比算法,我们对每个数据集的划分采用 70%作为训练集 30%作为测试集、10-折交叉验证、2-折交叉验证这 3 种方式.测试集的分类准确率 CA(classification accuracy)和维度缩减能力 DR(dimensionality reduction)作为 MCC-NES 算法及对比算法的评价准则.分类准确率的具体定义如公式(14),其中,N_{CC}(number of correct classification)是正确分类的实例数,N_{AS}(number of all samples)是数据集实例总数;维度缩减率的定义如公式(15),其中,N_{SF}(number of selected features)是被选择的特征数,N_{AF}(number of all features)是数据集的特征总数:

$$CA = N_{CC} / N_{AS} \quad (14)$$

$$DR = 1 - (N_{SF} / N_{AF}) \quad (15)$$

3.2 对比算法和参数设置

将提出的 MCC-NES 算法与其他的经典算法进行比较,这些算法的具体信息由表 2 给出.

Table 2 Information of the methods for comparisons**表 2** 对比算法的详细信息

算法名称	数据集划分	描述/发表年份
FSFOA	10-fold,2-fold,70%~30%	基于森林优化算法 ^[14] /2016
Rc-BBFA	70%~30%	基于收益成本的萤火虫算法 ^[15] /2017
UFSACO	70%~30%	基于ACO的无监督算法 ^[13] /2014
FS-NEIR	10-fold	近邻有效信息比 ^[11] /2013
PSO(4-2)	70%~30%	基于粒子群优化算法 ^[12] /2013
IFS-CoCo	10-flod	基于多种群协同进化 ^[21] /2010
NSM	10-fold	近邻软间隔 ^[43] /2010
SVM-FuzCoc	70%~30%	基于支持向量机 ^[9] /2010
SFS,SBS,SFFS	70%~30%	贪婪爬山算法 ^[8] /2010
HGAFS	2-fold	混合遗传算法 ^[44] /2007
ACOFSS+mRMR	10-flod	蚁群优化多目标特征选择 ^[45] /2015
FW-NSGA-II	10-flod	多目标演化特征选择算法 ^[16] /2018
DE-FLS	70%~30%	差分进化的特征选择方法 ^[17] /2018
ISEDDBFO,ACBFO	10-flod	细菌觅食算法 ^[18] /2017

用于对比的特征选择算法有:Ghaemi 等人利用分类器准确率作为评估标准提出的 FSFOA^[14]、Hu 等人基于邻居软间隔评估特征子集而提出的 NSM^[43]、Moustakidis 等人提出的 SVM-FuzCoc^[9]、Huang 等人提出的 FS 混合基因算法(HGAFS)^[44]、Zhu 等人利用邻居有效信息率评价准则提出的 FS-NEIR^[11]、Tabakhi 等人基于特征之间的相似性评估准则提出的 UFSACO^[13]、Xue 等人提出的将分类器准确率和特征子集规划作为评价准则的 PSO(4-2)^[12]、Zhang 等人基于成本收益萤火虫算法提出的 Rc-BBFA^[15]、J. Derrac 等人基于协同进化算法提出的 IFS-CoCo^[21]、贪婪爬山算法 SFFS^[8]、Khan A 等人基于 mRMR 增强的蚁群多目标特征选择算法 ACOFSS+mRMR^[45]、Marwa 等人提出的混合过滤和包裹方法的多目标演化特征选择算法 FW-NSGA-II^[16]、Hancer 等人基于差分进化提出的特征选择方法 DE-FLS^[17]、Chen 等人基于细菌觅食算法提出 ISEDDBFO^[18]及 ACBFO^[18]等。

实验中,MCC-NES 算法相关的参数设定由表 3 给出^[46]。其中,在处理不同维度数据集时,设定了不同的种群个数、子种群的大小以及子种群中个体的数量等。其原因是:随着数据维数的增加,出于计算资源的限制,相应减少种群个数以及子种群中个体的数量;而增加子种群的个数是为了让每个种群能够尽可能多地探索局部搜索空间,以此避免种群过早陷入局部最优解。同时,随着特征维数的增加,我们不断减小种群初始化参数,使得种群在高维数据上能够发掘出更关键的特征信息。最后,通过给高维数据设定更高的重启阈值和迭代次数,以此让种群在高维搜索空间中进行更充分的探索。

Table 3 Set of MCC-NES algorithm parameters**表 3** MCC-NES 参数设置

Dataset	Feature	Subpopulation size m	Individuals size λ	Restart threshold σ	Initialization parameters α	Historical utilization β	Number of population σ	Number of iteration τ
Glass	9	3	36	7	0.0	0.6	6	30
Heart	13	4	36	7	0.0	0.6	6	30
Cleveland	13	4	36	7	0.0	0.6	6	30
Wine	13	4	36	7	0.0	0.6	6	30
Vehicle	18	4	36	7	0.0	0.6	5	30
Segmentatin	19	4	36	7	0.0	0.6	5	30
Ionosphere	34	6	36	8	-0.1	0.6	5	30
Dermatology	34	6	36	8	-0.1	0.6	5	30
Spambase	57	10	24	8	-0.1	0.6	4	35
Sonar	60	10	24	8	-0.2	0.6	4	35
Musk2	166	12	24	8	-0.3	0.6	4	35
LSVT	310	15	24	9	-0.4	0.6	4	40
SRBCT	2 308	25	12	9	-0.5	0.6	3	40
Arcene	10 000	50	12	10	-0.6	0.6	3	40
RNA-Seq	20 531	60	12	10	-0.6	0.6	2	40
Dorothea	100 000	160	12	10	-0.8	0.6	1	40

3.3 算法迭代过程

关于算法的迭代过程,我们选取了两个比较有代表性的数据集 Sonar 和 Ionosphere,图 4 表示 MCC-NES 算法在 Sonar 以及 Ionosphere 数据集上采用 5-NN 分类器时的迭代搜索过程.拿 Sonar 数据集为例,算法在开始阶段,分类器有最低的分类性能以及最高的维度缩减,这是由于算法开始迭代时最优全局特征序列为空,每个亚种群不会与全局特征序列进行联系,所以在亚种群性能评估时,最优特征序列上只有每个亚种群中选取的特征,而其他亚种群位置上的特征均为 0,因此维度压缩率就会很高;另一方面,也导致一开始算法的分类性能比较低.但是随着迭代,之后的迭代过程中会保存历史最优的特征序列,每个亚种群进行评估时都会与历史最优特征序列进行联系,所以就会包含其他亚种群的历史特征信息,维度压缩率迅速降低,而性能则迅速上升.在到达最优解附近时,算法就会加大局部搜索力度,如算法在 7th~10th 迭代时接近最优解,在 11st 迭代达到最优解,之后算法继续迭代,就会对那些已选择的但是冗余的特征进行删除,比如在 12nd 以及 15th 时维度缩减率得到进一步提升.

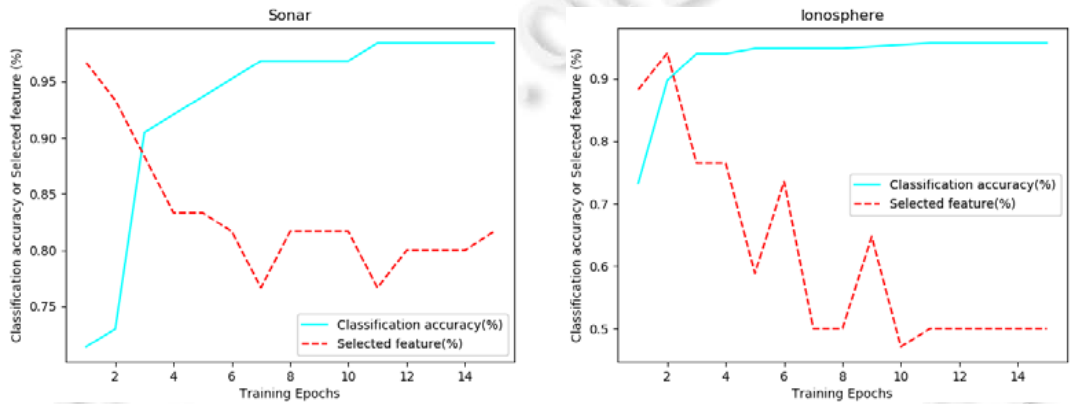


Fig.4 Classification accuracy and the number of selected features w.r.t. the number of iterations

图 4 分类准确率和所选特征个数在算法迭代中表现

3.4 多种群协同进化

图 5 表示 MCC-NES 算法在 Sonar 以及 Ionosphere 数据集上采用 Rbf-SVM 分类器搜索过程中,3 个分布式种群的最佳性能表现.

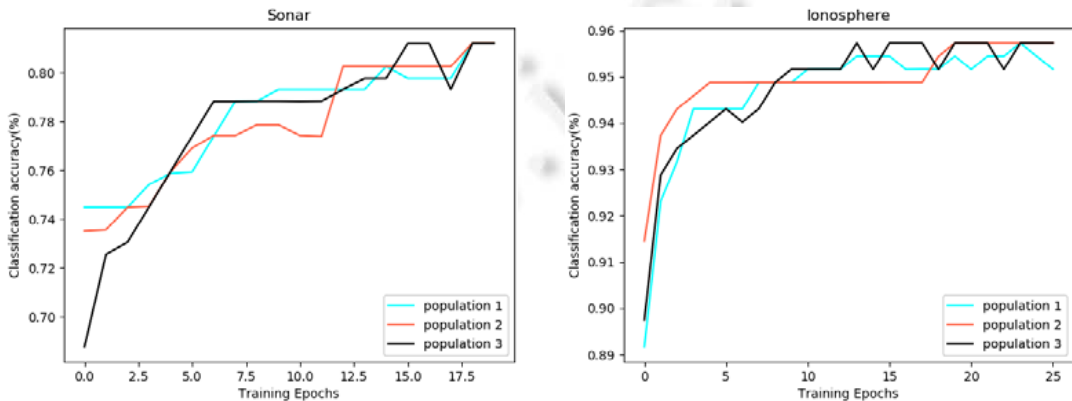


Fig.5 Performance of optimal solutions for each population in the evolution of population competition

图 5 分布式种群在迭代过程中最佳的性能表现

由于算法中设计了重启机制,当某个种群迭代达到阈值而且种群性能停滞时,就会让该种群的搜索点向表

现最优的种群转移.例如,Sonar 数据集中的红色种群在 11st 迭代的时候达到阈值,而且此时该种群历史性能表现最差,说明种群搜索极有可能陷入局部最优解.选择重启后,种群在 12nd 次迭代中性能大幅提高.当种群达到全局最优解的时候,种群性能就会趋于稳定.算法通过引入竞争重启机制来实现种群间的信息交互,防止种群陷入局部最优,并提高种群全局寻找最优解的能力,加速算法收敛速率.

图 6 表示 MCC-NES 算法在 Sonar 以及 Ionosphere 数据集上采用 1-NN 分类器时,3 个分布式种群的平均性能表现.在开始阶段种群平均性能波动性较大,这是因为算法在开始时具有较强的探索性,在搜索空间中的搜索范围比较大,所以搜索点的差异比较明显.随着迭代次数增加,搜索范围逐渐向最优解附近靠近,因此种群间个体表现的差异就会减小,种群就会减小搜索范围.所以,在图 6 中就会表现为种群的平均搜索性能趋于稳定.这也说明了算法可以根据种群的搜索性能来自适应调整搜索范围.

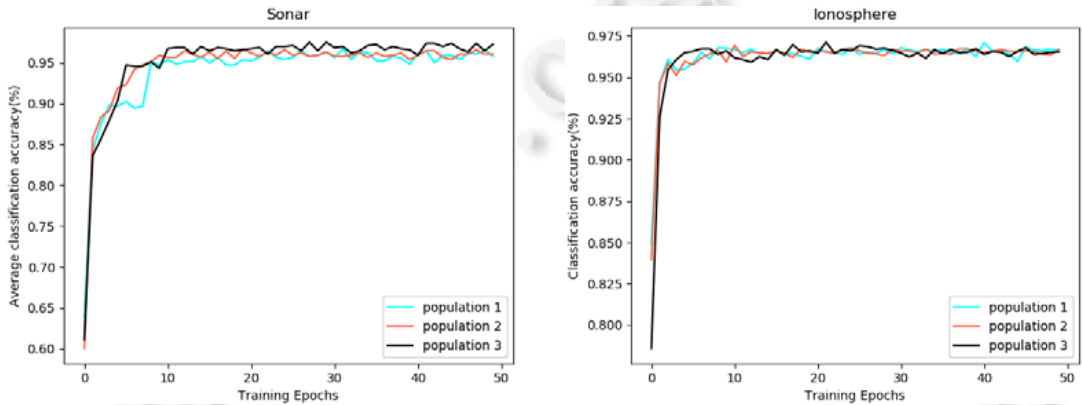


Fig.6 Average performance of each population in the evolution of population competition

图 6 分布式种群在迭代过程中的平均性能表现

3.5 种群内合作协同进化

图 7 中表示 MCC-NES 算法在 Ionosphere 数据集上采用 CART 分类器时,一个种群中的各个亚群之间的进化搜索过程.由于种群使用随机划分,所以算法开始的时候每个亚群的准确率表现差异较大;之后通过迭代,算法通过保留历史最优序列,使得所有的亚群准确率大致相似.图 7 中,在 2nd 迭代之后,所有的亚群准确率在 92% 左右.通过亚种群之间的合作协同进化,在 5th 时到达最优解解附近.所有亚群准确率开始稳定波动,随着分布参数的逐渐收敛,波动也逐渐趋于稳定.

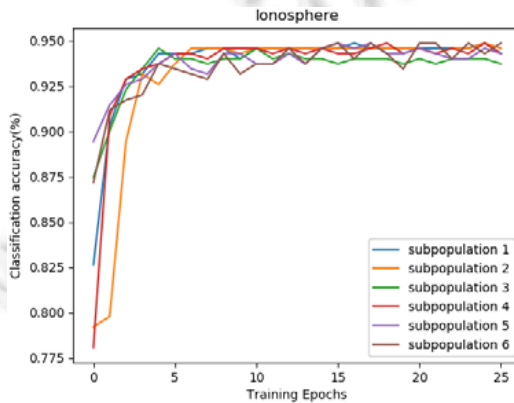


Fig.7 Performance of each subpopulation within the population

图 7 种群内的每个亚种群的性能

3.6 实验结果对比分析

本文提出的特征选择算法 MCC-NES 是在 NES 的基础上使用了一些优化策略,为了验证这些策略的有效性,我们对比了原始 NES, NES+F(在 NES 的基础上使用新的适应度函数), NES+CC(在 NES 基础上使用合作协同进化策略), MNES(基于 NES 的分布式多种群进化策略), MNES+RE(在 MNES 基础上使用重启策略)在 5 个数据集(按照特征维数分别代表低维数据、中维数据、高维数据以及更高维度数据)上的表现,实验中使用 1-NN 分类器,实验结果由表 4 给出.表 4 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).如表 4 中数据所示:在 NES 的基础上,通过采用相关的优化策略,可以有效地提高算法的性能.分析其原因分别为:在引入新的适应度函数后,因为综合考虑了维度缩减的影响,因此大幅提升了算法的 DR 值;在采用子种群合作协同进化策略后,通过将原始特征集合划分成多个子块进行协同进化,使得种群能更充分地探索局部搜索空间并发掘出关键信息,因此大幅提升了算法的分类准确率和维度压缩率;通过使用多种群分布式进化策略,实现多个种群并行搜索,更大程度地在搜索空间进行探索,因此提升了算法的性能;而通过引入种群重启策略后,让种群能够跳出局部最优解,并不断地将表现差的种群向表现好的种群进行转移,因此对分类准确率也有一定的改进.

Table 4 Classification accuracy and dimensional reduction based on various optimization strategies of NES

表 4 基于 NES 各种优化策略的分类准确率和维度缩减率

	CA (%)	DR (%)	Classifier
Cleveland			
MCC-NES	67.11(70%~30%)	74.61	1-NN
NES	58.89(70%~30%)	61.54	1-NN
NES+F	57.78(70%~30%)	69.23	1-NN
NES+CC	63.33(70%~30%)	73.92	1-NN
MNES+RE	62.22(70%~30%)	53.85	1-NN
MNES	61.11(70%~30%)	69.23	1-NN
Ionosphere			
MCC-NES	98.58(70%~30%)	75.88	1-NN
NES	94.34(70%~30%)	64.71	1-NN
NES+F	93.40(70%~30%)	72.47	1-NN
NES+CC	95.28(70%~30%)	73.41	1-NN
MNES+RE	96.23(70%~30%)	73.53	1-NN
MNES	95.28(70%~30%)	73.53	1-NN
Sonar			
MCC-NES	97.93(70%~30%)	78.0	1-NN
NES	92.06(70%~30%)	61.67	1-NN
NES+F	92.06(70%~30%)	66.67	1-NN
NES+CC	95.24(70%~30%)	76.67	1-NN
MNES+RE	94.71(70%~30%)	65.0	1-NN
MNES	93.65(70%~30%)	63.33	1-NN
Musk2			
MCC-NES	97.01(70%~30%)	80.28	1-NN
NES	85.24(70%~30%)	61.67	1-NN
NES+F	86.13(70%~30%)	72.67	1-NN
NES+CC	94.73(70%~30%)	76.67	1-NN
MNES+RE	94.71(70%~30%)	77.10	1-NN
MNES	93.65(70%~30%)	72.33	1-NN
Arcene			
MCC-NES	97.50(70%~30%)	98.44	1-NN
NES	91.67(70%~30%)	72.83	1-NN
NES+F	91.67(70%~30%)	75.97	1-NN
NES+CC	95.83(70%~30%)	98.27	1-NN
MNES+RE	93.37(70%~30%)	79.67	1-NN
MNES	93.33(70%~30%)	69.35	1-NN

同时,我们将 MCC-NES 算法与其他 17 个特征选择算法进行比较.对比算法详情已由表 2 给出.表 5~表 7 依据数据集维度大小的划分总结了全部实验结果,并用粗体标出了不同分类器对应的表现最佳的分类准确率和维度缩减率.按照文献[14]中给出的一般做法,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[14~18,21,45]中公开发表的实验结果.实验中使用的分类器主要包括 KNN(1-NN,3-NN,5-NN),CART 和 Rbf-SVM.

对比分类准确率,通过表 5~表 7 不难看出:当分类器和数据集划分方法相同时,对比其他所有算法,MCC-NES 在 Glass,Heart,Cleveland,Dermatology,Spambase,Musk2,LSVT,SRBCT,Arcene,RNA-Seq,Dorothea 等 11 个数据集上,其分类准确率都是最高的;而在 Wine 数据集的 Rbf-SVM 分类器上,分类准确率仅低于 HGAFS 算法 0.6%;在 Vehicle 数据集的 Rbf-SVM 分类器上,低于 HGAFS 算法 3%;在 Segmentatin 数据集的 3-NN 分类器上,低于 FAFOA 算法 2.6%;在 Ionosphere 数据集的 Rbf-SVM 分类器上,低于 ISEDBFO 算法 0.9%;在 Sonar 数据集的 Rbf-SVM 分类器上,低于 ACBFO 算法 12%。其中:在 Sonar 数据集中,当分类器为 1NN 时,MCC-NES 分类准确率比 SBS 高出 35%。

Table 5 Classification accuracy and dimension reduction of MCC-NES and compared methods (1)

表 5 MCC-NES 及其对比算法的分类准确率和维度缩减率(1)

Glass	CA (%)	DR (%)	Classifier
MCC-NES	85.23(70%~30%)	67.11	1-NN
FSFOA	71.88(70%~30%)	40.0	1-NN
SFS	72.24(70%~30%)	26.66	1-NN
SFFS	71.77(70%~30%)	37.77	1-NN
IFS-CoCo	55.44(10-fold)	41.11	1-NN
MCC-NES	70.61(2-fold)	55.56	RBF-SVM
FSFOA	68.22(2-fold)	60.0	RBF-SVM
HGAFS	65.51(2-fold)	44.44	RBF-SVM
MCC-NES	80.39(10-fold)	44.44	CART
FSFOA	75.7(10-fold)	50.0	CART
FS-NEIR	68.53(10-fold)	22.22	CART
Heart	CA (%)	DR (%)	Classifier
MCC-NES	85.56(10-fold)	61.53	3-NN
FSFOA	85.18(10-fold)	35.71	3-NN
NSM	84.0(10-fold)	69.23	3-NN
MCC-NES	85.18(2-fold)	53.84	RBF-SVM
FSFOA	84.07(2-fold)	50.0	RBF-SVM
HGAFS	82.59(2-fold)	76.92	RBF-SVM
MCC-NES	85.18(10-fold)	76.92	CART
FSFOA	85.15(10-fold)	48.07	CART
FS-NEIR	79.86(10-fold)	46.15	CART
Cleveland	CA (%)	DR (%)	Classifier
MCC-NES	67.11(70%~30%)	74.61	1-NN
FSFOA	55.55(70%~30%)	71.42	1-NN
SVM-FuzCoc	61.01(70%~30%)	46.1	1-NN
SFS	51.79(70%~30%)	47.7	1-NN
SBS	54.80(70%~30%)	38.5	1-NN
SFFS	49.50(70%~30%)	53.8	1-NN
Wine	CA (%)	DR (%)	Classifier
MCC-NES	99.53(70%~30%)	69.23	5-NN
FSFOA	99.20(70%~30%)	30.76	5-NN
PSO(4-2)	95.26(10-fold)	51.6	5-NN
FW-NSGA-II	98.98(10-fold)	53.84	5-NN
MCC-NES	99.44(70%~30%)	66.15	1-NN
FSFOA	98.07(70%~30%)	50.0	1-NN
SVM-FuzCoc	97.12(70%~30%)	53.84	1-NN
SFS	97.69(70%~30%)	35.38	1-NN
SBS	94.77(70%~30%)	46.15	1-NN
SFFS	96.56(70%~30%)	36.92	1-NN
MCC-NES	97.75(2-fold)	53.85	RBF-SVM
FSFOA	96.06(2-fold)	37.17	RBF-SVM
HGAFS	98.31(2-fold)	53.85	RBF-SVM
MCC-NES	99.26(70%~30%)	61.53	CART
FSFOA	96.0(70%~30%)	57.14	CART
UFSACO	95.08(70%~30%)	61.53	CART
MCC-NES	97.70(10-fold)	61.53	CART
FSFOA	96.06(10-fold)	21.42	CART
FS-NEIR	95.04(10-fold)	61.53	CART
FW-NSGA-II	98.62(10-fold)	57.69	CART

Table 6 Classification accuracy and dimension reduction of MCC-NES and compared methods (2)**表 6** MCC-NES 及其对比算法的分类准确率和维度缩减率(2)

Vehicle	CA (%)	DR (%)	Classifier
MCC-NES	86.20(70%~30%)	72.22	5-NN
FSFOA	73.98(70%~30%)	50.0	5-NN
PSO(4-2)	85.3(70%~30%)	68.4	5-NN
MCC-NES	86.89(70%~30%)	69.44	1-NN
FSFOA	73.81(70%~30%)	61.11	1-NN
Rc-BBFA	75.79(70%~30%)	61.11	1-NN
IFS-CoCo	70.85(10-fold)	42.75	1-NN
MCC-NES	73.40(2-fold)	88.88	RBF-SVM
FSFOA	62.41(2-fold)	47.22	RBF-SVM
HGAFS	76.36(2-fold)	38.99	RBF-SVM
MCC-NES	81.06(10-fold)	66.67	CART
FSFOA	73.04(10-fold)	31.57	CART
FS-NEIR	70.98(10-fold)	50.0	CART
Segmentatin	CA (%)	DR (%)	Classifier
MCC-NES	93.65(10-fold)	68.42	3-NN
FSFOA	96.20(10-fold)	30.0	3-NN
NSM	95.0(10-fold)	63.15	3-NN
Ionosphere	CA (%)	DR (%)	Classifier
MCC-NES	93.20(10-fold)	91.17	5-NN
FSFOA	89.43(10-fold)	54.28	5-NN
PSO(4-2)	87.27(70%~30%)	90.41	5-NN
FW-NSGA-II	86.80(10-fold)	42.64	5-NN
MCC-NES	98.58(70%~30%)	75.88	1-NN
FSFOA	89.52(70%~30%)	54.28	1-NN
SVM-FuzCoc	89.46(70%~30%)	88.23	1-NN
SFS	87.75(50%~50%)	65.88	1-NN
SBS	84.61(50%~50%)	77.64	1-NN
SFFS	88.32(50%~50%)	75.29	1-NN
MCC-NES	95.73(2-fold)	58.82	RBF-SVM
FSFOA	94.58(2-fold)	57.14	RBF-SVM
HGAFS	92.76(2-fold)	82.35	RBF-SVM
ACOFSS+mRMR	93.20(10-fold)	41.17	RBF-SVM
ISEDFO	96.60(10-fold)	52.64	RBF-SVM
ACBFO	96.20(10-fold)	50.58	RBF-SVM
MCC-NES	99.06(70%~30%)	85.29	CART
FSFOA	95.12(70%~30%)	47.05	CART
UFSACO	88.61(70%~30%)	11.17	CART
MCC-NES	94.36(10-fold)	79.41	CART
FSFOA	93.16(10-fold)	68.57	CART
FS-NEIR	92.59(10-fold)	82.35	CART
ACOFSS+mRMR	92.40(10-fold)	67.6	CART
Dermatology	CA (%)	DR (%)	Classifier
MCC-NES	99.36(70%~30%)	71.47	1-NN
FSFOA	97.27(70%~30%)	45.71	1-NN
SVM-FuzCoc	94.11(70%~30%)	64.7	1-NN
SFS	94.02(70%~30%)	44.7	1-NN
SBS	91.78(70%~30%)	58.23	1-NN
SFFS	93.70(70%~30%)	62.35	1-NN
IFS-CoCo	94.93(10-fold)	55.88	1-NN
MCC-NES	97.54(10-fold)	70.58	CART
FSFOA	96.99(10-fold)	21.42	CART
FS-NEIR	93.95(10-fold)	70.58	CART
ACOFSS+mRMR	96.10(10-fold)	50.0	CART
MCC-NES	98.18(70%~30%)	70.58	CART
FSFOA	90.09(70%~30%)	44.11	CART
UFSACO	95.28(70%~30%)	26.47	CART

Table 7 Classification accuracy and dimension reduction of MCC-NES and compared methods (3)**表 7** MCC-NES 及其对比算法的分类准确率和维度缩减率(3)

Spambase	CA (%)	DR (%)	Classifier
MCC-NES	92.47(10-fold)	57.89	RBF-SVM
ACOFSS+mRMR	91.40(10-fold)	59.32	RBF-SVM
MCC-NES	93.84(10-fold)	49.12	CART
ACOFSS+mRMR	93.60(10-fold)	40.35	CART
Sonar	CA (%)	DR (%)	Classifier
MCC-NES	96.83(70%~30%)	81.67	5-NN
FSFOA	86.98(70%~30%)	44.26	5-NN
PSO(4-2)	78.16(70%~30%)	81.26	5-NN
DE-FLS	76.77(70%~30%)	65.0	5-NN
MCC-NES	97.93(70%~30%)	78.0	1-NN
FSFOA	85.43(70%~30%)	57.37	1-NN
SVM-FuzCoc	73.17(70%~30%)	68.33	1-NN
SFS	66.43(50%-50%)	61.33	1-NN
SBS	62.20(50%-50%)	45.33	1-NN
SFFS	64.55(50%-50%)	61.33	1-NN
IFS-CoCo	85.70(10-fold)	57.17	1-NN
MCC-NES	81.20(2-fold)	90.0	RBF-SVM
FSFOA	65.86(2-fold)	54.09	RBF-SVM
HGAFS	87.02(2-fold)	75.0	RBF-SVM
ISEDBFO	92.80(10-fold)	56.66	RBF-SVM
ACBFO	93.50(10-fold)	56.5	RBF-SVM
MCC-NES	88.04(10-fold)	71.67	CART
FSFOA	82.69(10-fold)	52.45	CART
FS-NEIR	75.97(10-fold)	91.66	CART
Musk2	CA (%)	DR (%)	Classifier
MCC-NES	97.01(70%~30%)	80.28	1-NN
MCC-NES	96.92(70%~30%)	82.65	5-NN
DE-FLS	86.86(70%~30%)	78.67	5-NN
LSVT	CA (%)	DR (%)	Classifier
MCC-NES	97.11(70%~30%)	97.0	1-NN
FSFOA	89.47(70%~30%)	98.71	1-NN
Rc-BBFA	94.60(70%~30%)	56.45	1-NN
SRBCT	CA (%)	DR (%)	Classifier
MCC-NES	99.87(70%~30%)	98.62	1-NN
FSFOA	94.73(70%~30%)	49.06	1-NN
SVM-FuzCoc	98.88(70%~30%)	98.57	1-NN
Arcene	CA (%)	DR (%)	Classifier
MCC-NES	97.50(70%~30%)	98.44	1-NN
FSFOA	88.33(70%~30%)	61.9	1-NN
Rc-BBFA	92.50(70%~30%)	48.66	1-NN
MCC-NES	98.33(70%~30%)	98.27	CART
FSFOA	73.69(70%~30%)	77.67	CART
UFSACO	67.40(70%~30%)	99.8	CART
RNA-Seq	CA (%)	DR (%)	Classifier
MCC-NES	99.87(70%~30%)	99.46	1-NN
Rc-BBFA	94.53(70%~30%)	58.66	1-NN
Dorothea	CA (%)	DR (%)	Classifier
MCC-NES	95.42(70%~30%)	99.58	1-NN
MCC-NES	96.53(70%~30%)	99.63	CART
FSFOA	95.83(70%~30%)	26.03	CART

在处理更高维的数据时,很多特征选择方法就显得无能为力了,比如在面对 LSVT,SRBCT,Arcene,RNA-Seq,Dorothea 数据集时.我们在设定的最大迭代时间内(24 hours)内测试了本文提出的方法,并与最新提出的几个特征选择方法 FSFOA,UFSACO,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)进行比较.实验结果表明:在处理高维问题时,我们算法的准确率几乎都是最优的,并且特征数量即使达到 100 000 时仍能给出令人满意的解.其中,在 Arcene 的 CART 分类器上更是比次优的准确率高出 25%,比 UFSACO 算法高出 30%左右.而且与 Rc-BBFA,FSFOA 相比,我们的算法在解决高维问题时的分类准确率和维度缩减表现更为出色.

比较表 5~表 7 中各算法的 DR 值,MCC-NES 在 Cleveland,Wine,Vehicle,Segmentatin,Dermatology,Musk2,SRBCT,RNA-Seq 和 Dorothea 数据集上有最高的维度压缩率.对于其他的数据集,虽然不能在所有分类器上达到最高,但是 MCC-NES 的 DR 值仍具有很强的竞争力,特别是在处理高维数据时,在 Ionosphere 数据集 70%-30%划分条件下,分类器为 CART 时,MCC-NES 要比效果最好的 FSFOA 算法高出较接近 40%;在 Arcene 数据集 70%-30%划分条件下,分类器为 1-NN 时,MCC-NES 比次优 DR 的算法高出 36%;在 RNA-Seq 的 1-NN 分类器下,MCC-NES 的 DR 值高出 Rc-BBFA 算法 40%;在 Dorothea 的 CART 分类器下,MCC-NES 的 DR 值高出 FSFOA 算法 70%.

表 8 中列出了 MCC-NES 算法在 14 个数据集上使用 1-NN 分类器时,关于分类准确率及维度缩减率的标准差.同时,表中添加参数 D^* 用来表示算法最终选择的特征个数.分析表中数据可以看出,算法在所有数据集上均有稳定的分类准确率(这也侧面体现出算法的适应度函数主要以分类准确率为评价标准).除了在 Vehicle 数据集上的 2.18 和 LSVT 数据集上的 2.3 之外,算法在大多数数据集上分类准确率的标准差均在 1 左右,特别是在高维数据集上标准差更小,这反映出算法处理高维数据时出色的性能.另一方面,分析维度缩减的标准差,由于在低维数据集上特征总数较少,所选特征子集轻微变动就会导致比较明显的 DR 值差异,因此在低维数据集上表现为 D^* 值标准差较小;同时,DR 值的标准差较高.而在高维数据集上,由于特征个数较多,所选择的特征子集间差异就会增大,但是这些差异相对于庞大的原始特征集合而言影响不大,这也导致在高维数据集上特征子集的 DR 值比较相似,因此算法在高维数据集上表现为 D^* 的标准差变大,而 DR 值标准差变小.但是总体而言,算法在特征维数缩减上也有稳定的表现.最后,通过结合分类准确率、维度缩减率以及所选子集特征个数给出算法在处理每个数据集时的显著性测试结果.

Table 8 Standard deviation of classification accuracy and dimension reduction of MCC-NES on each data set

表 8 MCC-NES 在各个数据集上分类准确率和维度缩减率的标准差

Dataset	CA (%)		D^*		DR (%)		P-value
	Average	Stdev	Average	Stdev	Average	Stdev	
Glass	85.23	1.29	5.9	0.73	67.11	8.19	0.57
Heart	88.77	1.08	3.9	0.88	70.0	6.74	0.72
Cleveland	67.11	1.40	3.3	1.05	74.61	8.14	0.82
Wine	99.44	0.89	4.4	0.69	66.15	5.38	0.74
Vehicle	86.89	2.18	5.5	0.52	69.44	2.92	0.90
Ionosphere	98.58	0.49	8.2	1.75	75.88	5.15	0.98
Dermatology	99.36	0.43	9.7	0.67	71.47	1.98	0.97
Sonar	97.93	1.30	13.2	1.81	78.0	3.02	0.99
Musk2	97.01	0.34	32.74	2.95	80.28	1.78	0.98
LSVT	97.11	2.30	9.3	2.21	97.0	0.71	0.38
SRBCT	99.87	0.32	31.7	3.88	98.62	0.17	0.53
Arcene	97.5	0.87	155.6	10.64	98.44	0.11	0.49
RNA-Seq	99.87	0.20	109.9	9.85	99.46	0.05	0.49
Dorothea	95.42	0.24	423.3	23.71	99.58	0.02	0.45

4 总 结

本文提出了一种基于自然进化策略的特征选择算法 MCC-NES,算法在使用自然进化策略的基础上提出了几点优化方案.在 MCC-NES 的初始化阶段,我们使用基于对角协方差矩阵建模的自然进化策略,并通过初始化函数将特征高斯采样结果编码为 $\{0,1\}$ 特征序列;在种群进化阶段,为了使算法能够扩展到高维数据,在种群内部采用了合作协同进化的思想,并通过多种群分布式竞争进化的方式来提供更多的探索;针对算法陷入局部最优解的问题,采用种群重启机制,不断地将表现差的种群转移到表现好的种群周围,并为其提供新的探索能力;在算法更新阶段,克服传统更新机制的局限性,提出新的适应度函数,将维度缩减也纳入到特征选取的考虑范围,从而提高了特征子集的选取精度;进一步,基于新的适应度函数提出了一种适应度塑造方法,通过降低梯度对于适应度值的依赖来提高算法的更新精度;最后,使用 GPU-Tensorflow 作为算法的运行框架进行并行计算.之后,将 MCC-NES 算法在 16 个数据集和 17 个特征选择算法上进行测试、比较,实验结果表明:MCC-NES 普遍提高了数据集的分类准确率和维度缩减能力,特别是在高维数据中有出色的表现.今后的工作中,我们尝试将我

们的算法应用到更多的优化领域,特别是一些可以并行的优化过程中来检验我们算法的性能.同时,尝试将我们的算法应用到实际的工业领域来考察算法的适用性.

References:

- [1] Guyon I, Elisseeff A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003,1157–1182.
- [2] Liu H, Motoda H, Setiono R, *et al.* Feature selection: An ever evolving frontier in data mining. *Journal of Machine Learning Research*, 2010,4–13.
- [3] Dash M, Liu H. Feature selection for classification. *Intelligent Data Analysis*, 1997,1(3):131–156.
- [4] Chandrashekar G, Sahin F. A survey on feature selection methods. *Computers & Electrical Engineering*, 2014,40(1):16–28.
- [5] Kohavi R, John GH. Wrappers for feature subset selection. *Artificial Intelligence*, 1997,97(1-2):273–324.
- [6] Tan KC, Teoh EJ, Yu Q, *et al.* A hybrid evolutionary algorithm for attribute selection in data mining. *Expert Systems with Applications*, 2009,36(4):8616–8630.
- [7] Bart S, Gomes CP. Hill-climbing Search. *Encyclopedia of Cognitive Science*, 2006.
- [8] Pudil P, Novovicova J, Kittler J, *et al.* Floating search methods in feature selection. *Pattern Recognition Letters*, 1994,15(11):1119–1125.
- [9] 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.
- [10] 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.
- [11] Zhu W, Si G, Zhang Y, *et al.* Neighborhood effective information ratio for hybrid feature subset evaluation and selection. *Neurocomputing*, 2013,99:25–37.
- [12] Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Trans. on Cybernetics*, 2013,43(6):1656–1671.
- [13] Tabakhi S, Moradi P, Akhlaghian F, *et al.* An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence*, 2014,112–123.
- [14] Ghaemi M, Feizi-Derakhshi MR. Feature selection using forest optimization algorithm. *Pattern Recognition*, 2016,60:121–129.
- [15] Zhang Y, Song XF, Gong DW. A return-cost-based binary firefly algorithm for feature selection. *Information Sciences*, 2017,418.
- [16] Hammami M, Bechikh S, Hung C, *et al.* A multi-objective hybrid filter-wrapper evolutionary approach for feature selection. *Memetic Computing*, 2019,11(2):193–208.
- [17] Hancer E. Differential evolution for feature selection: A fuzzy wrapper-filter approach. *Soft Computing*, 2019,23(13):5233–5248.
- [18] Chen YP, Li Y, Wang G, *et al.* A novel bacterial foraging optimization algorithm for feature selection. *Expert Systems with Applications*, 2017,83:1–17.
- [19] Qian C, Yu Y, Zhou Z, *et al.* Subset selection by Pareto optimization. In: *Proc. of the Neural Information Processing Systems*. 2015. 1774–1782.
- [20] Feng C, Qian C, Tang K, *et al.* Unsupervised feature selection by Pareto optimization. In: *Proc. of the National Conf on Artificial Intelligence*. 2019.
- [21] Derrac J, García S, Herrera F. Ifs-Coco: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognition*, 2010,43(6):2082–2105.
- [22] Hoffmeister F, Back T. Genetic algorithms and evolution strategies—Similarities and differences. In: *Proc. of the Parallel Problem Solving from Nature*. 1990. 455–469.
- [23] Beyer H, Schwefel H. Evolution strategies—A comprehensive introduction. *Natural Computing*, 2002,1(1):3–52.
- [24] Salimans T, Ho J, Chen X, *et al.* Evolution strategies as a scalable alternative to reinforcement learning. *arXiv: Machine Learning*, 2017.
- [25] Mnih V, Kavucuoğlu K, Silver D, *et al.* Human-Level control through deep reinforcement learning. *Nature*, 2015,518(7540):529–533.
- [26] Gu S, Cheng R, Jin Y, *et al.* Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, 2018,22(3):811–822.
- [27] Dua D, Karra Taniskidou E. UCI machine learning repository. Irvine: University of California, School of Information and Computer Science, 2017. <http://archive.ics.uci.edu/ml>
- [28] Rechenberg I. Evolutionsstrategie optimierung technischer systeme nach prinzipien der biologischen evolution. 1973.

- [29] Kellermayer DI. Numerische optimierung von computer-modellen mittels der evolutionsstrategie Hans-Paul Schwefel Birkhäuser, Basel and Stuttgart. 1977 370 pages Hardback SF/48 ISBN 3-7643-0876-1. Cybernetics and Systems, 1977.
- [30] Beyer HG. Towards a theory of evolution strategies: Self-adaptation. *Evolutionary Computation*, 1995,3(3):311–347.
- [31] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 2001,9(2): 159–195.
- [32] Friedrichs F, Igel C. Evolutionary tuning of multiple SVM parameters. *Neurocomputing*, 2005,107–117.
- [33] Muller SD, Marchetto J, Airaghi S, *et al.* Optimization based on bacterial chemotaxis. *IEEE Trans. on Evolutionary Computation*, 2002,6(1):16–29.
- [34] Shepherd J, McDowell DL, Jacob KI, *et al.* Modeling morphology evolution and mechanical behavior during thermo-mechanical processing of semi-crystalline polymers. *Journal of the Mechanics and Physics of Solids*, 2006,54(3):467–489.
- [35] Schaul T, Glasmachers T, Schmidhuber J, *et al.* High dimensions and heavy tails for natural evolution strategies. In: *Proc. of the Genetic and Evolutionary Computation Conf.* 2011. 845–852.
- [36] Wierstra D, Schaul T, Glasmachers T, *et al.* Natural evolution strategies. *Journal of Machine Learning Research*, 2011.
- [37] Berny A. Selection and reinforcement learning for combinatorial optimization. In: *Proc. of the Parallel Problem Solving from Nature.* 2000. 601–610.
- [38] Berny A. Statistical machine learning and combinatorial optimization. In: *Proc. of the Theoretical Aspects of Evolutionary Computing.* Berlin, Heidelberg: Springer-Verlag, 2001. 287–306.
- [39] Amari S. Natural gradient works efficiently in learning. *Neural Computation*, 1998,10(2):251–276.
- [40] Amari S, Douglas SC. Why natural gradient. In: *Proc. of the Int'l Conf. on Acoustics Speech and Signal Processing.* 1998. 1213–1216.
- [41] Omidvar MN, Li X, Yang Z, *et al.* Cooperative co-evolution for large scale optimization through more frequent random grouping. In: *Proc. of the Congress on Evolutionary Computation.* 2010. 1–8.
- [42] Tahir MA, Bouridane A, Kurugollu F. Simultaneous feature selection and feature weighting using hybrid tabu search/K-nearest neighbor classifier. *Pattern Recognition Letters*, 2007,28(4):438–446.
- [43] Hu Q, Che X, Zhang L, *et al.* Feature evaluation and selection based on neighborhood soft margin. *Neurocomputing*, 2010, 73(10-12):2114–2124.
- [44] 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.
- [45] Khan A, Baig AR. Multi-Objective feature subset selection using mRMR based enhanced ant colony optimization algorithm (mRMR-EACO). *Journal of Experimental and Theoretical Artificial Intelligence*, 2016,28(6):1061–1073.
- [46] Zhang X. Research of feature selection algorithm based on natural evolution strategy [MS. Thesis]. Changchun: Jilin University, 2020.

附中文参考文献:

- [46] 张鑫.基于自然进化策略的特征选择算法研究[硕士学位论文].长春:吉林大学,2020.



张鑫(1994—),男,硕士,主要研究领域为进化计算,强化学习.



李占山(1966—),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为机器学习,约束推理.