智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

面向大规模特征选择的自监督数据驱动粒子群优化算法

黎建宇, 詹志辉

引用本文:
黎建宇,詹志辉. 面向大规模特征选择的自监督数据驱动粒子群优化算法[J]. 智能系统学报, 2023, 18(1): 194–206.
LI Jianyu,ZHAN Zhihui. A self–supervised data–driven particle swarm optimization approach for large–scale feature selection[J]. *CAAI Transactions on Intelligent Systems*, 2023, 18(1): 194–206.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202206008

DOI: 10.11992/tis.202206008
网络出版地址: https://kns.cnki.net/kcms/detail/23.1538.TP.20221013.0949.002.html

面向大规模特征选择的自监督数据驱动粒子群优化算法

黎建宇，詹志辉
（华南理工大学 计算机科学与工程学院，广东 广州 510006）

摘　要：大规模特征选择问题的求解通常面临两大挑战：一是真实标签不足，难以引导算法进行特征选择；二是搜索空间规模大，难以搜索到满意的高质量解。为此，提出了新型的面向大规模特征选择的自监督数据驱动粒子群优化算法。第一，提出了自监督数据驱动特征选择的新型算法框架，可不依赖于真实标签进行特征选择。第二，提出了基于离散区域编码的搜索策略，帮助算法在大规模搜索空间中找到更优解。第三，基于上述的框架和方法，提出了自监督数据驱动粒子群优化算法，实现对问题的求解。在大规模特征数据集上的实验结果显示，提出的算法与主流有监督算法表现相当，并比前沿无监督算法具有更高的特征选择效率。

关键词：特征选择；大规模优化；粒子群优化算法；进化计算；群体智能；数据驱动；自监督学习；离散区域编码
中图分类号：TP391　　**文献标志码**：A　　**文章编号**：1673−4785(2023)01−0194−13

中文引用格式：黎建宇，詹志辉．面向大规模特征选择的自监督数据驱动粒子群优化算法 [J]．智能系统学报，2023，18(1)：194–206.
英文引用格式：LI Jianyu, ZHAN Zhihui. A self-supervised data-driven particle swarm optimization approach for large-scale feature selection[J]. CAAI transactions on intelligent systems, 2023, 18(1): 194–206.

A self-supervised data-driven particle swarm optimization approach for large-scale feature selection

LI Jianyu, ZHAN Zhihui
(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract: Large-scale feature selection problems usually face two challenges: 1) Real labels are insufficient for guiding the algorithm to select features, and 2) a large-scale search space encumbers the search for a satisfactory high-quality solution. To this end, in this paper, a novel self-supervised data-driven particle swarm optimization algorithm is proposed for large-scale feature selection, including three contributions. First, a novel algorithmic framework named self-supervised data-driven feature selection is proposed, which can perform the feature selection without real labels. Second, a discrete region encoding-based search strategy is proposed, which helps the algorithm to find better solutions in a large-scale search space. Third, based on the above framework and method, a self-supervised data-driven particle swarm optimization algorithm is proposed to solve the large-scale feature selection problem. Experimental results on datasets with large-scale features show that the proposed algorithm performs comparably to the mainstream supervised algorithms and has higher feature selection efficiency than state-of-the-art unsupervised algorithms.

Keywords: feature selection; large-scale optimization; particle swarm optimization; evolutionary computation; swarm intelligence; data-driven; self-supervised learning; discrete region encoding

在现实世界中，人工智能问题的求解和模型的应用往往涉及大量的数据特征[1-4]。然而，并非所有的特征都有考虑的必要。在实际问题中，许

收稿日期：2022−06−06.　**网络出版日期**：2022−10−13.
基金项目：国家重点研发计划项目（2019YFB2102102）；国家自然科学基金面上项目资助（62176094）.
通信作者：詹志辉. E-mail: zhanapollo@163.com.

多特征的相关性很低, 甚至与问题不相关。因此, 使用这些特征容易误导算法模型, 降低模型的泛化能力、鲁棒性等性能[5-8]。特征选择的目的正是解决这个问题。特征选择旨在从原来的大规模特征集中选出合适的一部分相关特征。通过去除不相关和多余的特征, 特征选择可以降低数据特征的维度, 简化学习模型, 加快学习过程, 提高模型性能[9-12]。

然而, 由于数据特征的众多, 特征选择的搜索空间规模巨大。因此, 特征选择是一个难以求解的优化问题。例如, 对于有 D 个特征的数据集, 可能的特征选择方案总数为 2^D 个[1]。而且, 随着大数据技术的发展进步和现实问题复杂性的增加, 数据的特征数量 D 越来越大, 特征选择问题的搜索空间规模呈指数级急剧上升。当问题规模 D 大于 1 000 时, 这样的特征问题被称为大规模特征选择问题。在解决大规模特征选择问题时, 由于问题潜在的解有 2^D 个, 通过穷举式搜索找到给定的数据集的最佳特征子集十分不切实际。因此, 许多学者提出了各种搜索技术求解大规模特征选择问题, 如贪婪搜索、启发式搜索和随机搜索[13-18]。近年来, 进化计算 (evolutionary computation, EC) 方法因其全局搜索能力和搜索效率而广为人知, 被广泛应用在特征选择问题之中[19-23]。其中, 作为进化计算中群体智能的代表算法, 粒子群优化算法 (particle swarm optimization, PSO) 在大规模特征选择问题中表现出了巨大的优势和发展潜力, 受到了越来越多专家学者的关注[24-27]。因此, 本文主要考虑利用 PSO 求解大规模特征选择问题。

目前, 已有很多专家学者提出新型的 PSO 算法及其变种来求解大规模特征选择问题[28-29]。这些研究主要可以分为两类。第一类是设计适合求解特征选择问题的 PSO 算法。由于初始提出的 PSO 是用于求解连续优化问题的算法, 而特征选择问题是典型的离散优化问题, 因此需要将 PSO 转变为适用于求解离散优化问题的算法。这些研究包括粒子的编码方式(解码方式), 离散的粒子速度和位置更新策略[1]。其中, 二进制 PSO (binary PSO, BPSO) 受到了广泛的关注和研究[25]。第二类研究关注如何让算法更高效地求解特征选择问题。例如, Gu 等[26] 将擅于求解大规模优化问题的竞争粒子群优化算法 (competitive swarm optimizer, CSO) 用于求解特征选择优化问题, 而 Tran 等[27] 提出可变长度的 PSO 以避开局部最优和增强搜索效率。Xue 等[28] 提出基于自适应参数和策略的 PSO, 用于对大规模特征选择问题进行自适应求解。Luo 等[29] 提出混合粗糙超立方体方法与 BPSO的算法来对大规模特征选择问题进行高效求解。

尽管已有不少相关的研究, 使用 PSO 进行大规模特征选择时仍然存在以下两个问题。第一, 现有的 PSO 算法都是在有监督的条件下执行, 需要依赖数据的真实标签。然而, 在大多数实际场景中, 由于真实标签的获取十分困难而且成本高昂, 数据只有很少的真实标签, 甚至没有标签。因此, 现有的算法无法很好地求解这类缺乏真实标签的大规模特征选择问题。第二, 在处理大规模特征时, 由于搜索空间巨大, 粒子很容易陷入局部最优, 导致搜索性能不佳, 即 PSO 面临着“维度诅咒”的挑战, 因此非常需要高效的搜索机制[1]。

为此本文提出了自监督数据驱动粒子群优化算法 (self-supervised data-driven PSO, SDPSO) 来高效地求解大规模特征选择问题。本文的贡献和创新之处主要有以下三点。

1) 本文提出了自监督数据驱动特征选择 (self-supervised data-driven feature selection, SDFS) 的新型算法框架, 可通过自监督任务生成数据标签, 从而使用生成的标签来驱动算法进行特征选择, 避免了对真实标签的依赖。另外, 为了实现数据标签生成, 本文提出了基于多模态聚类的任务生成方法 (multi-modal clustering-based task generation, MCTG) 来生成自监督任务, 从而对数据赋予合适的标签。

2) 近期研究工作发现, 基于区域编码的 PSO 能够高效求解大规模连续优化问题[30]。受此启发, 本文提出了基于离散区域编码的搜索策略 (discrete region encoding-based search, DRES) 来辅助 PSO 在大规模离散搜索空间中搜索到更优解, 提升 PSO 求解大规模特征选择问题的性能。

3) 基于上述提出的框架和方法, 本文提出了完整的 SDPSO 算法, 从而实现对大规模特征选择问题的高效求解。

为了验证算法性能, 本文采用了领域内通用的 6 个大规模数据集 (人脸数据集 Yale[31] 和 ORL[32], 物品识别数据集 COIL20[33], 以及 3 个基因数据集 Leukemia[34]、DLBCL[34] 和 Braintumor[34]) 进行实验分析, 并与主流和前沿算法进行对比。

1　背景知识与相关工作

1.1　大规模特征选择问题

有 D 维特征的特征选择问题可表示为

$$\max_{O} f(O)$$
$$\text{s.t.}\quad O_d \in \{0,1\},\quad 1 \leqslant d \leqslant D \tag{1}$$

式中：O_d 为 1 表示第 d 维特征被选择，O_d 为 0 表示第 d 维特征不被选择；f 是衡量特征组合的性能的函数，例如，f 可以是基于特征组合进行分类时的准确率。由式 (1) 可见，特征选择问题是一个 0/1 组合优化问题，其候选解集合的规模为 2^D。因此，随着问题规模增大，即特征数量 D 增加（例如 D>1 000），问题的候选解数量呈指数型上升，给优化算法带来巨大挑战，这样的特征选择问题通常被称为大规模特征选择问题。

1.2　二进制粒子群优化算法

Kennedy 等[24] 提出的 PSO 算法是求解复杂优化问题的高效群体智能算法。然而，最初提出的 PSO 算法只适用于求解连续搜索空间的优化问题，不能直接用于求解特征选择等离散二进制搜索空间的优化问题。因此，研究者们提出了离散的 PSO 算法来对这类离散优化问题进行求解。其中，比较著名的是 BPSO 算法[25]。与原始的 PSO 一样，BPSO 中每个粒子个体都带有 1 个位置向量 $\boldsymbol{X}$ 和 1 个速度向量 $\boldsymbol{V}$。其中 $\boldsymbol{X}$ 和 $\boldsymbol{V}$ 的维度与问题维度一致。然而，与原始的 PSO 不同的地方是，BPSO 中每一个 $\boldsymbol{X}$ 对应一个问题的候选解，因此，针对特征选择问题，$\boldsymbol{X}$ 中的每一维取值只能是 0 或 1，即位置向量 $\boldsymbol{X}$ 的每一维都是离散的值。然而，$\boldsymbol{V}$ 中的每一维仍然是连续值，以便粒子个体在优化过程中进行位置更新。在优化过程中，BPSO 使用 Sigmoid 函数将 $\boldsymbol{V}$ 的每一维取值映射到 [0, 1]，从而决定粒子位置 $\boldsymbol{X}$ 对应维度取值为 1 的概率。

速度 $\boldsymbol{V}$ 的更新公式可以表示为

$$\boldsymbol{V}_{i,d} = w \cdot \boldsymbol{V}_{i,d} + c_1 \cdot r_{1_{i,d}} \cdot (\boldsymbol{X}_{i,d}^{\text{pbest}} - \boldsymbol{X}_{i,d}) + c_2 \cdot r_{2_{i,d}} \cdot (\boldsymbol{X}_{d}^{\text{gbest}} - \boldsymbol{X}_{i,d}) \tag{2}$$

式中：i 是个体的索引；d 代表维度；w 是惯性权重参数；c_1 和 c_2 是加速因子；$r_{1_{i,d}}$ 和 $r_{2_{i,d}}$ 是不同的属于 [0, 1] 区间的随机数；$\boldsymbol{X}_{i,d}^{\text{pbest}}$ 是个体 i 的历史最优位置；$\boldsymbol{X}_{d}^{\text{gbest}}$ 是所有个体的最优位置。

在更新速度 $\boldsymbol{V}$ 后，位置 $\boldsymbol{X}$ 根据 $\boldsymbol{V}$ 的值和 Sigmoid 函数（以 ϕ 表示）取值为

$$\boldsymbol{X}_{i,d} = \begin{cases} 1, & r_{i,d} < \phi(\boldsymbol{V}_{i,d}) \\ 0, & \text{其他} \end{cases} \tag{3}$$

式中：i 是个体的索引；d 是维度索引；$r_{i,d}$ 是 [0, 1] 区间的随机数；而 ϕ 的表达式为

$$\varphi(z) = \frac{1}{1+\mathrm{e}^{-z}} \tag{4}$$

在位置 $\boldsymbol{X}$ 更新后，算法评估新个体的适应值，然后更新对应的 $\boldsymbol{X}^{\text{pbest}}$ 和 $\boldsymbol{X}^{\text{gbest}}$。式 (2)~(4) 不断重复，直到算法满足停止条件。

1.3　基于二进制粒子群优化算法的特征选择

由于特征选择问题是一个 0/1 组合优化问题，使用基于二进制的粒子群优化算法（例如 BPSO 算法）进行求解非常合适、直观。为了实现有效的特征选择，通常需要两个不重叠的数据集：用于进行特征选择的数据集（后文简称为特征选择数据集）和用于测试所选特征的效果的数据集（后文简称为测试集）。因此，BPSO 等算法只使用特征选择数据集进行特征选择，然后在测试集上测试选出特征的性能。基于 BPSO 求解特征选择问题的经典算法步骤如算法 1 所示。

算法 1　求解特征选择的经典 BPSO 算法

输入　种群规模 N，特征数目（问题维度）D，样本数据 S，对应 S 的真实标签数据 L，最大的适应值评估次数 $Z_{\max}$；

输出　搜索到的最优特征选择方案 $\boldsymbol{X}^{\text{gbest}}$。

1) 初始化 N 个粒子的速度 $\boldsymbol{V}$ 和位置 $\boldsymbol{X}$；

2) 设真实标签数据 L 为计算适应值时的标签数据 T；

3) 计算每个粒子的适应值；//算法 2

4) Z=N；//N 个个体消耗了 N 次适应值评估次数

5) 记录个体的历史最优 $\boldsymbol{X}^{\text{pbest}}$ 和种群全局最优 $\boldsymbol{X}^{\text{gbest}}$；

6) WHILE（Z<$Z_{\max}$）DO

7) 更新粒子速度 $\boldsymbol{V}$ 和位置 $\boldsymbol{X}$；

8) 计算每个粒子的适应值；//算法 2

9) Z= Z +N；

10) 更新个体的历史最优 $\boldsymbol{X}^{\text{pbest}}$ 和种群全局最优 $\boldsymbol{X}^{\text{gbest}}$；

11) END WHILE

12) 输出 $\boldsymbol{X}^{\text{gbest}}$。

在特征选择的优化过程中，需要计算每个粒子个体的适应值。此处的适应值可设置为特征组合的性能指标。本文以分类准确率作为粒子个体的适应值。为了计算适应值，特征选择数据集 S 及 S 中各个数据对应的分类标签 T 将被划分成训练集和验证集，分别记为 S_{train} 和 S_{valid} 以及标签集合 T_{train} 和 T_{valid}。在现有的研究中，分类标签数据 T 都是直接使用数据 S 对应的真实分类标签数据 L（即将 L 当作 T）。然而，如果将真实标签 L 当作 T，需要确保特征选择数据集 S 中的样本都有对应的真实标签，否则，没有对应真实标签的样本将不可使用。因此，L 中真实标签的数量决定了特征选择数据集中可用样本的数量，这会导致在真实标签数据不足的情况下，特征选择数据集的可用样本过少，算法无法很好地进行特征选

择。这也是本文提出可不依赖真实标签 L 的自监督特征选择框架的研究动机。即在真实标签 L 不足、甚至没有的情况下, 本文的自监督特征选择框架仍然可以通过采用基于多模态聚类的任务生成方法生成标签 T, 以辅助特征选择。

在划分特征数据集后, 对于需要计算适应值的个体 $\boldsymbol{X}$, 将其转换为对应的特征组合($\boldsymbol{X}$ 第 d 维取值为 1 则表示第 d 维特征被选中并使用, 否则不使用), 并基于划分后的训练集 S_{train} 和对应的标签集 T_{train} 建立和训练分类器 G。分类器可以选择使用任意的分类模型, 例如经典的 K 最邻近分类方法。

为了避免上述划分的偶然性导致适应值计算偏差, 通常使用 H–折交叉验证法对 S 和 T 进行 H 次的划分并计算分类准确率在 H 次划分情况下的平均结果, 其中 H 为一个正整数参数, 常设为 H=5 或 H=10。H–折交叉验证法计算适应值的过程如下: 1)数据集 S 和对应的标签集 T 会被均分成 H 份, 每一份也称为一折。其中, 根据粒子个体的 $\boldsymbol{X}$, S 中的数据只保留被 $\boldsymbol{X}$ 选中的特征列。2)对于第 h 次划分, 第 h 份数据集及对应的第 h 份标签分别被用作验证集 S_{valid} 和 T_{valid}, 而剩余的 $(H-1)$ 份数据(包括第 1 到第 $(h-1)$ 份和第 $(h+1)$ 份到第 H 份的数据及对应标签)被合并用作训练集 S_{train} 和 T_{train}。从而训练得到模型 G_h。基于 S_{valid} 和 T_{valid} 以及模型 G_h, 计算本次(第 h 次)划分的分类准确率 A_h, 计算过程为

$$A_h = \frac{1}{|S_{\text{valid}}|}\sum_{j=1}^{|S_{\text{valid}}|} I(G_h(S_{\text{valid},j}), T_{\text{valid},j}) \tag{5}$$

式中: $|S_{\text{valid}}|$为验证集 S_{valid} 的总样本数(也即第 h 份数据的样本数), $S_{\text{valid},j}$ 为 S_{valid} 中第 j 个样本数据, $T_{\text{valid},j}$ 为对应 S_{valid} 中第 j 个样本数据的分类标签, $G_h(S_{\text{valid},j})$ 为分类模型 G_h 对样本 $S_{\text{valid},j}$ 的预测分类标签, 函数 I 的定义为

$$I(a,b) = \begin{cases} 1, & a == b \\ 0, & \text{其他} \end{cases} \tag{6}$$

将 2)的过程执行 H 次, 得到 H 个准确率的结果。

3)计算 H 次准确率结果的均值作为粒子 $\boldsymbol{X}$ 的适应值 $f(\boldsymbol{X})$:

$$f(\boldsymbol{X}) = \frac{1}{H}\sum_{h=1}^{H} A_h \tag{7}$$

式中 A_h 表示第 h 次的分类准确率。适应值计算过程如算法 2 所示。

算法 2　适应值评估

输入　数据集 S, 标签数据 T, 交叉验证折数 H, 候选解 $\boldsymbol{X}$。

输出　候选解 $\boldsymbol{X}$ 的适应值。

1) 根据 $\boldsymbol{X}$ 将 S 中未被选中的特征列去掉;
2) 将数据集 S 和对应的标签数据 T 均分成 H 份;
3) FOR h=1 to H
4) 将第 h 份样本数据作为 S_{valid};
5) 将除第 h 份外的所有样本数据作为 S_{train};
6) 将第 h 份数据对应的标签作为 T_{valid};
7) 将除第 h 份外所有数据对应的标签作为 T_{train};
8) 基于 S_{train} 和 T_{train} 建立并训练分类器 G;
9) 式 (5) 计算 A_h;
10) END FOR
11) 公式 (7) 计算并输出 X 的适应值。

1.4　针对大规模特征选择的 PSO 算法

在求解大规模特征选择问题时, 传统的 PSO 经常面临着“维度诅咒”的挑战, 性能不佳。因此, 很多专家学者提出了针对大规模特征选择的 PSO 算法及其变种[26-27]。Xue 等[28] 提出基于自适应参数和策略的 PSO(Self-adaptive Parameter and Strategy based PSO, SPS-PSO), 用于对大规模特征选择问题进行自适应求解。Luo 等[29] 提出混合粗糙超立方体方法与 BPSO 的算法(Hybridization of the Rough Hypercuboid Approach and BPSO, RH-BPSO)来对大规模特征选择问题进行高效求解。

虽然国内外的专家学者们已经提出许多针对大规模特征选择的 PSO 算法及其变种, 但是这些算法都需要在有监督的条件下执行, 即算法搜索过程中需要用到具有真实标签的数据。然而, 在实际应用中, 监督信息(真实标签)的获取十分困难且成本很高, 很多情况下只有一部分的数据带有真实标签, 甚至没有数据带有真实标签。因此, 有监督算法在这些应用中很难取得良好的表现。与现有的有监督算法不同, 本文提出了自监督数据驱动的算法框架, 通过自监督学习来进行特征选择, 不用依赖于真实标签等监督信息。另外, 本文还提出了新型的离散区域搜索策略, 以辅助 PSO 对大规模特征选择问题进行求解。

2　自监督数据驱动粒子群优化算法

2.1　自监督数据驱动特征选择算法框架

为了实现不依赖标签等真实监督信息来进行高效的特征选择, 本文提出新型算法框架 SDFS。如图 1 所示, 该算法框架可利用自监督任务生成数据标签, 并使用生成的标签来驱动算法对特征进行选择。因此, 该框架不用依赖真实的数据标签亦可完成特征选择过程。

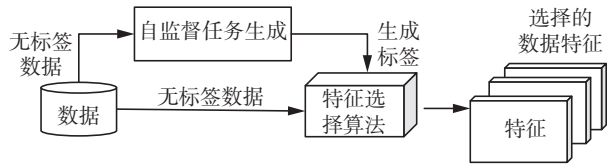


图 1　自监督数据驱动特征选择算法框架

Fig. 1　Selection Algorithm framework of self-supervised data-driven feature

然而，如何基于无监督的数据获取或构建合适的自监督信息是一个关键的难题。在现有的研究中，有许多学者研究和提出了不同的方法生成自监督分类任务，从而对数据赋予标签，例如基于数据重建方式生成分类任务（如图像着色和分辨率辨别）[35]，基于任务知识生成分类任务（如几何变换识别和噪声数据辨别）[36-37] 和自动任务生成（如使用聚类算法和矩阵分解）[38-41]。这些方法都可以用于生成自监督任务，从而对数据赋予标签。由于基于聚类的方法相比其他方法具有较好的通用性且较为常用[42]，本文也采用聚类的方式生成自监督分类任务，并赋予数据对应的标签。为此，本文提出了基于多模态聚类的任务生成方法（MCTG），对数据进行自动化的自监督分类任务生成，并基于生成任务对数据赋予标签。不同于现有采用单一聚类赋予数据标签的方法，MCTG 的创新之处在于它旨在利用多种聚类方法，对数据进行多种模态的分类，并通过对不同分类结果的关联综合结果作为生成的标签信息替代原来有监督搜索中所需的真实标签信息。

具体地，MCTG 使用 Q 个不同的聚类方法，其中每个聚类方法都将数据分成两类，类别记为 0 或 1。记对第 i 个样本数据的第 q 个分类结果为 $t_{i,q}$ ($t_{i,q}\in\{0,1\}$)，则对于该样本的自监督任务分类标签 T_i 为

$$T_i=\sum_{q=1}^{Q}(2^{Q-1}\cdot t_{i,q}) \tag{8}$$

可见，生成的任务最终共有 2^Q 个类别（T_i 的取值有 2^Q 种），即类别 0, 1, …, 2^Q–1。因此，每个数据都将属于 2^Q 个类别中的一个类别，这些类别标签可用于后续的特征选择过程。不失一般性，本文在算法实现中采用了 3 种（Q=3）较为常见的聚类方法来进行任务生成，分别是基于欧式距离、基于曼哈顿距离（绝对值距离）和基于余弦值的 K-means 聚类方法。具体的 MCTG 伪代码如算法 3 所示。基于 MCTG，可以用生成的标签评估特征组合的优劣（即候选解的适应值），该计算过程如前文的算法 2 所示，其中 T 为生成的分类标签而非真实分类标签。

算法 3　MCTG

1) FOR q=1 to Q

2)使用第 q 个聚类方法将样本数据集 S 聚成两类

3) FOR i=1 to |S| //|S|为数据集 S 的总样本数

4) IF S_i 被分到第一类 THEN

5) $t_{i,q}=0$

6) ELSE

7) $t_{i,q}=1$

8) END IF

9) END FOR

10) END FOR

11) 根据式 (8) 计算每个样本数据的 T 结果作为标签。

2.2　离散区域编码的搜索策略

近期的研究工作发现，基于区域编码的 PSO 在求解大规模连续优化问题中有很好的效果[30]。受此启发，本文提出 DRES 来提升 PSO 求解大规模特征选择优化问题的性能。

首先，在区域编码的 PSO 中，每个粒子不仅仅表示一个点，还表示以该点为中心，r 为半径的覆盖区域。然而，针对离散问题，粒子覆盖的区域需要重新定义。因此，首先定义粒子的离散区域编码。给定区域半径 r，每个粒子 $\boldsymbol{X}_i$ 所对应的一个离散区域集合，记为 $\kappa(\boldsymbol{X}_i, r)$，可表示为

$$\kappa(\boldsymbol{X}_i,r)=\{U\,|\gamma(U,\boldsymbol{X}_i)\leqslant r\} \tag{9}$$

其中 γ 计算两个向量的汉明距离。使用 $\kappa(\boldsymbol{X}_i, r)$ 中的随机个体替换当前个体位置，从而增加搜索的多样性。综上，DRES 的伪代码如算法 4 所示。

算法 4　DRES

1) FOR 现有种群中的每个个体 $\boldsymbol{X}_i$;

2) 任意生成一个属于 $\kappa(\boldsymbol{X}_i, r)$ 的个体，记为 $\boldsymbol{U}_i$;

3) 用 $\boldsymbol{U}_i$ 替换 $\boldsymbol{X}_i$;

4) END FOR。

2.3　完整算法

基于上述方法和策略，并结合 BPSO 中的速度（式 (2)）和位置更新（式 (3)），本文提出了完整 SDPSO 算法。SDPSO 的算法伪代码如算法 5 所示。算法 5 主要分为 2 个部分。第 1 部分是算法 5 的第 1 行，对于无标签数据通过 MCTG（即算法 3）生成数据的标签，以进行后续的特征选择。第 2 部分是基于自监督数据驱动的特征选择优化，即算法 5 的第 2~15 行。该过程包括粒子的初始化（算法 5 的第 2~6 行）和演化搜索（算法 5 的第 7~15 行）。其中，第 10 行会执行 DRES（算法 4）以增强算法搜索的多样性和效率。第 7~15 行将不断迭代执行直到满足算法停止条件（此处为适

应值评估次数 Z 达到预设的最大次数 $Z_{\max}$)。最后, 算法输出找到的最优解 $\boldsymbol{X}^{\text{gbest}}$。

算法 5　SDPSO

输入　种群规模 N, 特征数目(问题维度) D, 样本数据 S, 最大的适应值评估次数 $Z_{\max}$。

输出　搜索到的最优特征选择方案 $\boldsymbol{X}^{\text{gbest}}$。

1) 对 S 中的数据生成标签 T; //算法 3

2) 随机初始化 N 个粒子的速度 $\boldsymbol{V}$ 和位置 $\boldsymbol{X}$;

3) 计算 N 个粒子的适应值; //算法 2, 基于生成的标签 T

4) Z=N; //已消耗 N 次适应值评估次数

5) 记录每个个体的历史最优 $\boldsymbol{X}^{\text{pbest}}$;

6) 记录全局最优 $\boldsymbol{X}^{\text{gbest}}$;

7) WHILE(Z <$Z_{\max}$)DO

8) 根据式 (2) 更新粒子速度 $\boldsymbol{V}$;

9) 根据式 (3) 更新粒子位置 $\boldsymbol{X}$;

10) 执行 DRES 得到新的粒子位置 $\boldsymbol{X}$; //算法 4

11) 计算新粒子的适应值; //算法 2, 基于生成的标签 T

12) Z= Z +N;

13) 更新每个个体的历史最优 $\boldsymbol{X}^{\text{pbest}}$;

14) 更新全局最优 $\boldsymbol{X}^{\text{gbest}}$;

15) END WHILE

16) 输出 $\boldsymbol{X}^{\text{gbest}}$。

本节对 SDPSO(即算法 5)的时间复杂度进行分析。由上文可知, 算法 5 主要有 2 个部分, 第 1 部分为算法 5 的第 1 行, 第 2 部分算法 5 的第 2~15 行。

在算法 5 的第 1 部分, SDPSO 通过算法 3 对数据进行标签生成。算法 3 的第 1~10 行需执行 Q 次循环。每次循环中, 算法 3 的第 2 行执行一次 Kmeans 算法。Kmeans 算法的时间复杂度为 $O(G\times t\times|S|\times D)$, 其中 G 为聚类的迭代次数, t 为聚类的类别数, $|S|$为 S 中样本数, D 为样本特征数。由于算法 3 中 Kmeans 聚类类别均为 2 类(即 t=2), 执行的 Kmeans 算法的时间复杂度可简化为 $O(G\times|S|\times D)$。算法 3 的第 3~9 行执行$|S|$次循环, 每次循环执行一次判断和一次赋值, 因此 $|S|$次循环的时间复杂度为 $O(|S|)$。因此, 算法 3 第 2~9 行的时间复杂度合并为 $O(G\times|S|\times D)$。由于算法 3 第 1~10 行需执行 Q 次循环, 因此其时间复杂度为 $O(Q\times G\times|S|\times D)$。算法 3 的 11 行计算式 (8), 由于式 (8) 包含 Q 次乘法和加法, 因此该行的时间复杂度为 $O(Q)$。综上, 算法 3 的总时间复杂度合并为 $O(Q\times G\times|S|\times D)$。由于 Q 为常数参数且本文中 Q=3, 因此算法 3 的时间复杂度可简化为 $O(G\times|S|\times D)$。

算法 5 的第 2 部分包括其第 2~15 行。第 2 行初始化 N 个粒子个体的速度和位置, 时间复杂度为 $O(N\times D)$。第 3 行计算 N 个个体的适应值。由算法 2 可见, 适应值计算的时间复杂度与所采用的分类器相关。因此, 为了不失一般性, 此处假设每次适应值评估的时间复杂度为 $O(E)$, 则第 3 行的时间复杂度为 $O(N\times E)$。第 4 行为赋值操作, 时间复杂度为 $O(1)$, 而第 5 和第 6 行是寻找并记录最优个体, 因此时间复杂度为 $O(N\times D)$。因此, 第 3~6 行的时间复杂度为 $O(N\times(E+D))$。在第 7~15 行的 WHILE 循环中, 第 8、9 行更新粒子的速度和位置, 时间复杂度为 $O(N\times D)$; 第 10 行执行算法 4, 对每个个体进行 DRES, 时间复杂度也为 $O(N\times D)$; 第 11~14 行与第 3~6 行相同, 因此时间复杂度也为 $O(N\times(E+D))$。综上, 第 8~14 行的时间复杂度可合并为 $O(N\times(E+D))$。由于 WHILE 循环将循环执行第 8~14 行 v=($Z_{\max}$/N–1) 次, 其中 $Z_{\max}$ 为最大的适应值评估次数, WHILE 循环的时间复杂度为 $O(v\times N\times(E+D))$。则算法 5 第 2 部分(第 2~14 行)的总时间复杂度为 $O(N\times(E+D)+v\times N\times(E+D))$= $O(Z_{\max}\times(E+D))$。

综合算法 5 的第 1 部分和第 2 部分, 算法 5 的总时间复杂度为 $O(G\times|S|\times D+ Z_{\max}\times(E+D))$。

3　实验分析

3.1　实验设置

为了对算法性能进行测试, 本文采用了领域内常见的 6 个具有大规模特征的数据集进行实验。这 6 个数据集分别是人脸数据集 Yale [31] 和 ORL [32], 物品识别数据集 COIL20 [33] 以及 3 个基因数据集 Leukemia [34]、DLBCL [34]、Braintumor [34]。这些数据集已被研究者收集整理, 其中前 3 个数据集在文献 [43] 中公开, 而后 3 个数据集在文献 [34] 中公开。表 1 提供了这些数据集的相关信息。此外, 图 2 也给出了 3 个图片数据集的部分样例图片, 以供参考。

为了对比 SDPSO 的性能, 本文使用目前主流的代表性有监督算法 BPSO[25] 和 CSO[26] 作为对比算法。对于参数设置, 为了公平起见, SDPSO 与 BPSO 采用相同的参数, 即种群规模为 20, 速度的最大值和最小值分别为 6 和−6, 加速因子 c_1 和 c_2 均为 2.01, 惯性权重 w 为 1。上述这些参数也是 BPSO 论文的推荐值[25]。对于 SDPSO 中 DRES 的设置, 因为在大规模问题中, $\kappa(\boldsymbol{X}_i, r)$ 包含的候选解会随着 r 增大急剧增多, 所以 DRES 中

的 r 设置为较小的值，即 r=1。CSO 的参数设置也按其论文的推荐进行设置[26]。此外，本文也将全选所有特征的方法（简记为 FULL 方法）和随机分类的方法（简记为 RAND）作为基准结果与提出的算法进行对比。

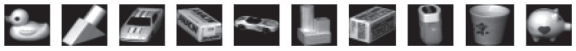
表 1　6 个数据集的具体信息

Table 1　Information of the six datasets

数据集名称	样本数	类别数	特征数	源数据类型
Yale[31]	165	15	1 024	人脸图片数据
ORL[32]	400	40	1 024	人脸图片数据
COIL20[33]	1 440	20	1 024	物品图片数据
Leukemia[34]	72	3	5 327	基因数据
DLBCL[34]	77	2	5 469	基因数据
Braintumor[34]	90	5	5 920	基因数据

(a) Yale 数据集样例

(b) ORL 数据集样例



(c) COIL20 数据集样例

图 2　3 个图片数据集的样例

Fig. 2　Samples of the three image datasets

所有进行特征选择的算法在优化过程中的最大评估次数被设定为 Z_{max}=5 000。在每个数据集上，每个算法独立运行 30 次，采用平均结果（在测试集上的分类准确率）作为最终结果。为了保持公平性，每个数据集也进行 30 次的随机划分，每次将 90% 的样本作为特征选择数据集和 10% 的样本作为测试集。因此，每个算法在第 i 次运行时使用第 i 次划分的数据集。即使用第 i 次划分的特征选择数据集进行特征选择，然后将选择的特征在第 i 次划分的测试集上进行测试。另外，在特征选择阶段，SDPSO、BPSO 和 CSO 的每个粒子使用常用的 H=5 的 H–折交叉验证法来获得基于选择特征的平均分类准确率作为适应值。为了体现本文提出方法的实用性，在实验中特征选择数据集中只有 20% 的数据带有真实分类标签。由于 SDPSO 不需要依赖真实标签，因此 SDPSO 可以在整个特征数据集上进行特征选择，而 BPSO 与 CSO 只能使用特征选择数据集中带真实标签的数据（即 20% 的数据）进行特征选择。此外，为了更准确地评估选出的特征的效果，实验在测试集上使用留一验证法计算分类的准确率。其中，留一验证法是 H 设置为数据样本数的 H–折交叉验证法，即有多少个数据，就需计算多少次分类准确率再求平均。因此，相比于 H 更小的交叉验证法版本（如 H=5），留一验证法计算误差更小，但因为需要计算更多次的准确率（即 A 值）以取平均，所以计算量更大。另外，由于 FULL 和 RAND 算法不需要进行特征选择，因此在计算其第 i 次结果的时候，只需直接使用第 i 次划分的测试集通过留一验证法来计算分类准确率。在实验中，本文选择 K 最邻近分类方法作为分类器来计算实验中所选特征的分类准确率，其中参数 K 被设置为 5。此外，当得到的分类准确率相近时，只需更少特征数量的特征选择组合比需要更多特征数量的特征选择组合更优。

此外，本文采用 Wilcoxon 符号秩检验（显著性水平 α=0.05）来检验 SDPSO 与其他对比算法之间的显著差异。在实验统计结果中，符号 “+”表示 SDPSO 明显优于对比算法，符号“–”表示 SDPSO 明显劣于对比算法，符号“=”表示 SDPSO 与对比算法在当前显著水平上没有明显差异。所有的算法实验均在配置 Intel Core i7-7700F CPU @ 3.60 GHz 和总内存为 8 GB 的集群服务器上运行。

3.2　与主流经典方法的实验对比

表 2 给出了 SDPSO 与主流经典算法的对比结果。对比结果显示 SDPSO 能取得与使用了真实标签的有监督算法相当的结果。根据 Wilcoxon 符号秩检验，SDPSO 在 Yale 数据集上取得了比 CSO 显著好的结果，并在剩下的数据集上取得与 BPSO 和 CSO 相当的结果。除此之外，SDPSO 在 2 个数据集上取得最优的均值（已在表 2 中加粗），在数量上与 BPSO 和 CSO 相当。而在选择出来的特征数量方面，SDPSO 和对比的 BPSO 和 CSO 在 6 个数据集上都没有明显差别。可见，SDPSO 具有与基于真实标签的算法相当，甚至更好的性能。而且，SDPSO 的运行并不需要真实的标签。因此，相比于基于真实标签的算法，SDPSO 具有更加通用、成本更低（获取真实标签的成本）的优势。这些都表明了 SDPSO 的优越性。

此外，跟全选所有特征的 FULL 方法相比，SDPSO 在 Leukemia 数据集上取得显著好的结果，并在剩下 5 个数据集上与 FULL 表现相当。同时，SDPSO 在这 6 个数据集上选择的特征数量约占 FULL 的 50%。即，SDPSO 只用 50% 的特征就能取得与 FULL 相当，甚至更好的分类结果。可见，SDPSO 具有较强的特征选择能力。另外，与 RAND 相比，SDPSO 在全部的数据集上都得到显著更优的结果，即 SDPSO 取得了比随机分类显著更高的准确率，甚至在 COIL20 数据集上的准确率比随

机分类的准确率超过 13 倍, 表明了特征选择进行分类的有效性。

综上所述, 对比实验结果表明了 SDPSO 不依赖真实标签进行特征选择的性能。

表 2　SDPSO 与经典算法的实验结果

Table 2　The experimental results of SDPSO and classical algorithms

数据集	SDPSO (特征选择不需真实标签)		BPSO (特征选择基于真实标签)		CSO (特征选择基于真实标签)		FULL (不进行特征选择)		RAND (随机进行分类)
	准确率	特征数	准确率	特征数	准确率	特征数	准确率	特征数	准确率
Yale	**0.150±0.142**	511.6	**0.150±0.133(=)**	510.1	0.138±0.141(+)	510.6	0.144±0.134(=)	1 024	0.070±0.004(+)
ORL	0.151±0.081	509.0	0.149±0.090(=)	511.0	0.143±0.086(=)	508.3	**0.154±0.085(=)**	1 024	0.025±0.001(+)
COIL20	0.721±0.035	512.1	0.722±0.042(=)	508.9	**0.723±0.042(=)**	511.2	0.722±0.041(=)	1 024	0.050±0.001(+)
Leukemia	0.462±0.285	2 661.2	0.463±0.320(=)	2 664.3	**0.468±0.265(=)**	2 669.0	0.452±0.293(+)	5 327	0.333±0.002(+)
DLBCL	**0.671±0.229**	2 735.8	0.663±0.222(=)	2 720.2	0.670±0.226(=)	2 736.2	**0.671±0.229(=)**	5 469	0.500±0.002(+)
Braintumor	0.652±0.184	2 960.2	**0.671±0.157(=)**	2 965.7	0.662±0.186(=)	2 965.0	0.648±0.187(=)	5 920	0.200±0.001(+)
+/=/−	NA		0/6/0		1/5/0		1/5/0		6/0/0

3.3　SDFS 框架的作用分析

为了对 SDFS 框架的作用进行分析, 本文将 SDPSO 与不使用 SDFS 框架的 SDPSO 变种(简记为 SDPSO-w/o-SDFS)进行对比。具体地, SDPSO-w/o-SDFS 使用真实的标签数据代替 SDFS 得到的生成标签数据。除此之外, SDPSO-w/o-SDFS 在设置上与 SDPSO 一致。表 3 给出了对比实验的结果。由表 3 可知, SDPSO 在 Leukemia 数据集上得到的结果显著优于 SDPSO-w/o-SDFS。此外, 在另外 5 个数据集上 SDPSO 与 SDPSO-w/o-SDFS 取得的结果没有显著性差异。同时, SDPSO 与 SDPSO-w/o-SDFS 在每个数据集上所选择的特征数量也基本相当。这说明 SDFS 能够免除算法对真实标签的依赖, 同时对算法的效果有一定程度的提升, 表明了 SDFS 的有效性。

表 3　SDPSO 与 SDPSO-w/o-SDFS 的实验结果

Table 3　Experimental results of SDPSO and SDPSO-w/o-SDFS

数据集	SDPSO (不需真实标签)		SDPSO-w/o-SDFS (基于真实标签)	
	准确率	特征数	准确率	特征数
Yale	**0.150±0.142**	511.6	0.143±0.143(=)	512.4
ORL	**0.151±0.081**	509.0	0.143±0.091(=)	510.1
COIL20	**0.721±0.035**	512.1	0.720±0.040(=)	508.8
Leukemia	**0.462±0.285**	2 661.2	0.443±0.278(+)	2 664.4
DLBCL	**0.671±0.229**	2 735.8	0.663±0.245(=)	2 741.5
Braintumor	**0.652±0.184**	2 960.2	**0.652±0.184(=)**	2 959.4
+/=/-	NA		1/5/0	

3.4　基于离散区域编码的搜索策略的作用分析

为了对 DRES 进行性能分析, 本文将 SDPSO 与不使用 DRES 的 SDPSO 变种(简记为 SDPSO-w/o-DRES)进行对比。表 4 给出了 SDPSO 与 SDPSO-w/o-DRES 的对比结果。从表 4 可知, SDPSO 在 Yale 数据集和 Leukemia 数据集上得到的结果显著优于 SDPSO-w/o-DRES。在另外的 4 个数据集上, SDPSO 取得的结果也与 SDPSO-w/o-DRES 相当。即, 在总体上 SDPSO 有比 SDPSO-w/o-DRES 更好的性能。这说明了 DRES 有助于算法在大规模搜索空间中找到更好的候选解。

表 4　SDPSO 与 SDPSO-w/o-DRES 的实验结果

Table 4　Experimental results of SDPSO and SDPSO-w/o-DRES

数据集	SDPSO		SDPSO-w/o-DRES	
	准确率	特征数	准确率	特征数
Yale	**0.150±0.142**	511.6	0.132±0.142(+)	511.8
ORL	**0.151±0.081**	509.0	0.150±0.084(=)	505.9
COIL20	**0.721±0.035**	512.1	**0.721±0.032(=)**	509.9
Leukemia	**0.462±0.285**	2 661.2	0.445±0.295(+)	2 662.6
DLBCL	0.671±0.229	2 735.8	**0.678±0.230(=)**	2 735.2
Braintumor	**0.652±0.184**	2 960.2	0.647±0.195(=)	2 963.0
+/=/-	NA		2/4/0	

3.5　真实标签比例对特征选择的影响

为了探究真实标签比例对特征选择的影响, 通过实验测试对比 SDPSO 与其他算法在不同情况下的优化结果。测试了 SDPSO、BPSO 和 CSO

在特征选择数据集中真实标签数据比例变化时的特征选择效果。由于 SDPSO 在特征选择过程中不需要依赖真实标签数据，因此真实标签数据比例的变化不会影响 SDPSO 的结果。

3 个算法在不同真实标签数据比例情况下的结果如图 3 所示。由图 3 可见，在真实标签数据比例较小时（如不大于 40%），SDPSO 在大部分问题上都能取得比 BPSO 和 CSO 更高的准确率，说明在真实标签数据样本较少时，SDPSO 相比 BPSO 与 CSO 具有总体更优越的性能。而且，在真实标签数据比例变化的情况下，SDPSO 的结果波动比 BPSO 与 CSO 更小，说明了 SDPSO 具有更好的稳定性和通用性。综上所述，真实标签对 SDPSO 的影响要比对 BPSO 和 CSO 的影响小得多。而且，基于 SDPSO 的特征选择在真实标签不足的情况下仍能有很好的表现，说明了 SDPSO 的通用性。

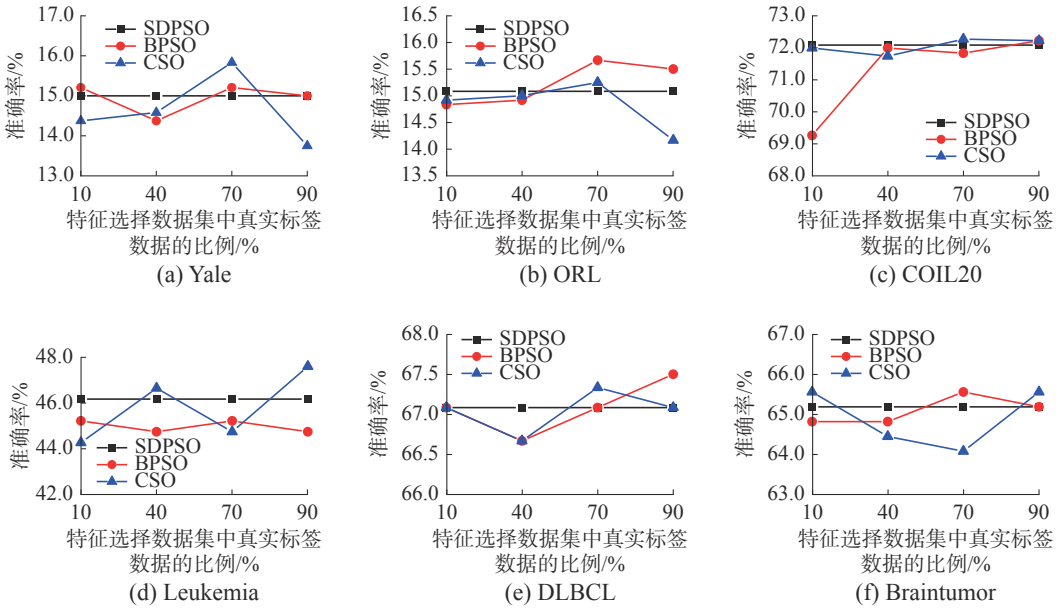


图 3　不同真实标签数据比例情况下的结果变化图

Fig. 3　Results with different percentages of real label data

3.6　与前沿方法的实验对比

为了进一步测试 SDPSO 的性能优势，本节将 SDPSO 与 6 个前沿的特征选择方法进行实验对比。这 6 个前沿方法分别为：引言中提到的 SPS-PSO[28] 和 RH-BPSO[29]，结合凸非负矩阵分解和自适应图约束的特征选择方法 (convex non-negative matrix factorization with an adaptive graph constraint feature selection, CNAFS)[38]、多组自适应图表示方法 (multi-group adaptive graph representation, MGAGR)[39]、基于嵌入图学习和约束的无监督特征选择方法 (unsupervised feature selection via embedded graph learning and constraint method, EGCFS)[40] 和结合可分性的无监督特征选择方法 (unsupervised feature selection with separability, UFS^2)[41]。其中，CNAFS 和 UFS^2 主要基于生成标签进行无监督特征选择，MGAGR 和 EGCFS 主要利用图学习与几何约束进行无监督特征选择。由于 CNAFS、MGAGR、EGCFS 和 UFS^2 方法均需预先确定需要选择的特征数，而现有文献通常采用穷举法确定最佳选择特征数[38-41]，本文也采用穷举法确定这 4 种方法的最佳选择特征数。在穷举法过程中，需要对不同特征数设置下的选取特征进行评估和比较，以确定最佳特征数。因此，在实验中，采用基于真实标签的算法 2 作为穷举法过程中的特征评估方式。另外，由于 SPS-PSO 和 RH-BPSO 需要标签信息进行特征搜索，为了公平比较算法的搜索效率，SPS-PSO 和 RH-BPSO 也使用与 SDPSO 相同的生成标签，并使用算法 2 作为搜索过程中粒子的适应值评估方式。所有对比算法的参数均采用其对应论文的推荐设置。每个算法独立运行 30 次并使用平均结果进行比较。

各个算法在 6 个数据集上的平均准确率如表 5 所示。在 6 个不同数据集的特征选择问题上，SDPSO 能在 Yale、ORL、COIL20、Leukemia 和 Braintumor 这 5 个不同数据集上取得比其他对比算法都更好的结果，表明了 SDPSO 具有比其他算法更好的综合性能。而且，Yale、ORL 和 COIL20 是样本类别数分别为 15、40 和 20 的多类别数据集，要对样本实现准确分类十分困难。然而，SDPSO 在这 3 个数据集上都能得到比其他算法更好的分

类准确率, 说明 SDPSO 选择的特征能够应对复杂的分类任务。综上所述, 实验对比结果表明了 SDPSO 进行特征选择的优越性。

本节也对算法的运行时间进行比较。其中, CNAFS、MGAGR、EGCFS 和 UFS^2 方法的运行时间包括使用穷举法选取最优特征数所需的时间。各个算法的平均运行时间如表 6 所示。由表 6 可见, 在特征数较小的数据集上, 如 Yale、ORL 和 COIL20, CNAFS 的运行时间最短, 而在特征数较多的数据集上, 如 Leukemia, DLBCL 和 Braintumor, SDPSO 算法所需的运行时间最短。尽管在 Yale、ORL 和 COIL20 上, SDPSO 的运行时间多于 CNAFS, 但 SDPSO 得到的分类准确率明显优于 CNAFS(如表 5 所示)。此外, 与 SPS-PSO 和 RH-BPSO 相比, SDPSO 使用简单且高效的 DRES 进行高效的搜索, 因此在大规模问题上只需更少的平均运行时间。与 CNAFS、MGAGR、EGCFS 和 UFS^2 相比, SDPSO 能够在搜索最优特征组合的同时确定最优特征数, 而 CNAFS、MGAGR、EGCFS 和 UFS^2 在确定特征的优劣和选择顺序之后, 还需再寻找最优的选择特征数。因此, 随着问题特征数不断增加, SDPSO 的运行时间效率优势更加明显。综上所述, 算法运行时间的比较结果表明了 SDPSO 的高效性。

表 5　SDPSO 算法与前沿算法的平均准确率结果

Table 5　Average accuracy results of SDPSO and state-of-the-art algorithms

数据集	SDPSO	SPS-PSO	RH-BPSO	CNAFS	MGAGR	EGCFS	UFS^2
Yale	**0.150**	0.145	0.142	0.124	0.117	0.121	0.113
ORL	**0.151**	0.148	0.144	0.115	0.129	0.123	0.136
COIL20	**0.721**	0.658	0.658	0.633	0.643	0.660	0.568
Leukemia	**0.462**	0.437	0.410	0.430	0.423	0.437	0.417
DLBCL	0.671	0.693	0.687	0.683	0.687	0.683	**0.723**
Braintumor	**0.652**	0.648	0.640	0.652	0.623	0.643	0.623
最优结果数	**5**	0	0	0	0	0	1

表 6　SDPSO 算法与前沿算法的平均运行时间

Table 6　Average running time of SDPSO and state-of-the-art algorithms　　s

数据集(总特征数)	SDPSO	SPS-PSO	RH-BPSO	CNAFS	MGAGR	EGCFS	UFS^2
Yale(1 024)	108.376	107.471	107.123	**22.756**	110.726	43.912	372.168
ORL(1 024)	22.385	22.950	22.446	**4.817**	20.156	24.775	179.095
COIL20(1 024)	950.269	926.013	926.361	**202.682**	1 490.019	212.838	1 530.091
Leukemia(5 327)	**59.313**	61.276	59.451	66.717	275.612	2 357.183	5 982.512
DLBCL(5 469)	**64.378**	66.325	64.779	75.146	343.659	4 311.511	6 527.624
Braintumor(5 920)	**83.871**	85.479	84.113	105.956	538.061	5 460.928	7 882.529

4　结束语

本文提出了面向大规模特征选择问题的 SDPSO 算法。首先, SDPSO 算法基于本文提出 SDFS 框架和 MCTG 方法, 可以在不依赖真实标签的情况下进行特征选择。其次, SDPSO 使用了本文提出的 DRES, 能够高效地在大规模搜索空间中找到更好的候选解。实验结果表明, 本文提出的 SDPSO 能在不使用真实标签的情况下依然与使用真实标签的对比算法表现相当, 并比前沿无监督算法具有更高的特征选择效率。

在未来工作中, 我们希望进一步增强 SDPSO 算法及 SDFS 框架在复杂特征选择问题上的求解性能, 包括带有多目标、多任务、小样本和数据不平衡等特点的特征选择问题。而且, 如何基于无监督的数据获取或构建合适的自监督信息也是未来值得研究的科学问题。此外, 我们也会将相关算法应用到智慧城市的复杂智能应用问题。

参考文献:

[1]　XUE Bing, ZHANG Mengjie, BROWNE W N, et al. A survey on evolutionary computation approaches to fea-

ture selection[J]. IEEE transactions on evolutionary computation, 2016, 20(4): 606–626.

[2] ZHAN Zhihui, SHI Lin, TAN K C, et al. A survey on evolutionary computation for complex continuous optimization[J]. Artificial intelligence review, 2022, 55(1): 59–110.

[3] WANG Peng, XUE Bing, LIANG Jing, et al. Differential evolution based feature selection: a niching-based multi-objective approach[J]. IEEE transactions on evolutionary computation, PP(99): 1.

[4] ZHAN Zhihui, LI Jianyu, ZHANG Jun. Evolutionary deep learning: a survey[J]. Neurocomputing, 2022, 483: 42–58.

[5] CHENG Fan, CUI Junjie, WANG Qijun, et al. A variable granularity search based multi-objective feature selection algorithm for high-dimensional data classification[EB/OL]. (2022–03–18)[2022–06–01].https://ieeexplore.ieee.org/abstract/document/9737335.

[6] LI Jianyu, ZHAN Zhihui, XU Jin, et al. Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks[EB/OL]. (2021–09–20)[2022–06–01].https://ieeexplore.ieee.org/document/9540902.

[7] SONG Xianfang, ZHANG Yong, GONG Dunwei, et al. Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data[EB/OL]. (2022–05–18)[2022–06–01].https://ieeexplore.ieee.org/abstract/document/9775183

[8] 李永豪, 胡亮, 高万夫. 基于稀疏系数矩阵重构的多标记特征选择 [J]. 计算机学报, 2022, 45(9): 1827–1841.
LI Yonghao, HU Liang, GAO Wanfu. Multi-label feature selection based on sparse coefficient matrix reconstruction[J]. Chinese journal of computers, 2022, 45(9): 1827–1841.

[9] LI Junyu, CHEN Jiazhou, QI Fei, et al. Two-dimensional unsupervised feature selection via sparse feature filter [EB/OL]. (2022–04–11)[2022–06–01].https://ieeexplore.ieee.org/abstract/document/9754711.

[10] WANG Peng, XUE Bing, LIANG Jing, et al. Multiobjective differential evolution for feature selection in classification[EB/OL]. (2021–12–07) [2022–06–01].https://pubmed.ncbi.nlm.nih.gov/34874881.

[11] 陈彤, 陈秀宏. 特征自表达和图正则化的鲁棒无监督特征选择 [J]. 智能系统学报, 2022, 17(2): 286–294.
CHEN Tong, CHEN Xiuhong. Feature self-representation and graph regularization for robust unsupervised feature selection[J]. CAAI transactions on intelligent systems, 2022, 17(2): 286–294.

[12] LI Xiaoping, WANG Yadi, RUIZ R. A survey on sparse learning models for feature selection[J]. IEEE transactions on cybernetics, 2022, 52(3): 1642–1660.

[13] 李顺勇, 王改变. 一种新的最大相关最小冗余特征选择算法 [J]. 智能系统学报, 2021, 16(4): 649–661.
LI Shunyong, WANG Gaibian. New MRMR feature selection algorithm[J]. CAAI transactions on intelligent systems, 2021, 16(4): 649–661.

[14] LIU Shulei, WANG Handing, PENG Wei, et al. A surrogate-assisted evolutionary feature selection algorithm with parallel random grouping for high-dimensional classification[J]. IEEE transactions on evolutionary computation, 2022, 26(5): 1087–1101.

[15] 曾毓菁, 姜勇. 一种融入注意力和预测的特征选择 SLAM 算法 [J]. 智能系统学报, 2021, 16(6): 1039–1044.
ZENG Yujing, JIANG Yong. Feature selection simultaneous localization and mapping algorithm incorporating attention and anticipation[J]. CAAI transactions on intelligent systems, 2021, 16(6): 1039–1044.

[16] ZHANG N, GUPTA A, CHEN Zefeng, et al. Evolutionary machine learning with minions: a case study in feature selection[J]. IEEE transactions on evolutionary computation, 2022, 26(1): 130–144.

[17] YANG Jiaquan, CHEN Chunhua, LI Jianyu, et al. Compressed-encoding particle swarm optimization with fuzzy learning for large-scale feature selection[J]. Symmetry, 2022, 14(6): 1142.

[18] CHEN Ke, XUE Bing, ZHANG Mengjie, et al. Correlation-guided updating strategy for feature selection in classification with surrogate-assisted particle swarm optimisation[EB/OL]. (2021–12–13)[2022–06–01].https://ieeexplore.ieee.org/abstract/document/9647020.

[19] WANG Zijia, JIAN Junrong, ZHAN Zhihui, et al. Gene targeting differential evolution: a simple and efficient method for large scale optimization[EB/OL]. (2022–06–23)[2022–06–23].https://ieeexplore.ieee.org/abstract/document/9804806.

[20] ZHANG Yong, WANG Yanhu, GONG Dunwei, et al. Clustering-guided particle swarm feature selection algorithm for high-dimensional imbalanced data with missing values[J]. IEEE transactions on evolutionary computation, 2022, 26(4): 616–630.

[21] WANG Yequn, LI Jianyu, CHEN Chunhua, et al. Scale adaptive fitness evaluation-based particle swarm optim-

isation for hyperparameter and architecture optimisation in neural networks and deep learning[EB/OL]. (2022–06–02)[2022–06–02].https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/cit2.12106.

[22] HE Chunlin, ZHANG Yong, GONG Dunwei, et al. A multi-task bee colony band selection algorithm with variable-size clustering for hyperspectral images[EB/OL]. (2022–03–14) [2022–06–02].https://ieeexplore.ieee.org/abstract/document/9733922.

[23] 陈宗淦, 詹志辉. 面向多峰优化问题的双层协同差分进化算法 [J]. 计算机学报, 2021, 44(9): 1806–1823.
CHEN Zonggan, ZHAN Zhihui. Two-layer collaborative differential evolution algorithm for multimodal optimization problems[J]. Chinese journal of computers, 2021, 44(9): 1806–1823.

[24] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95-International Conference on Neural Networks. Perth: IEEE, 1995: 1942–1948.

[25] KENNEDY J, EBERHART R C. A discrete binary version of the particle swarm algorithm[C]//1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. Orlando: IEEE, 1997: 4104–4108.

[26] GU Shenkai, CHENG Ran, JIN Yaochu. Feature selection for high-dimensional classification using a competitive swarm optimizer[J]. Soft computing, 2018, 22(3): 811–822.

[27] TRAN Binh, XUE Bing, ZHANG Mengjie. Variable-length particle swarm optimization for feature selection on high-dimensional classification[J]. IEEE transactions on evolutionary computation, 2019, 23(3): 473–487.

[28] XUE Yu, TANG Tao, PANG Wei, et al. Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers[J]. Applied soft computing, 2020, 88: 106031.

[29] LUO Chuan, WANG Sizhao, LI Tianrui, et al. Large-scale meta-heuristic feature selection based on BPSO assisted rough hypercuboid approach[EB/OL]. (2022–05–12) [2022–06–02].https://ieeexplore.ieee.org/abstract/document/9773310.

[30] JIAN Junrong, CHEN Zonggan, ZHAN Zhihui, et al. Region encoding helps evolutionary computation evolve faster: a new solution encoding scheme in particle swarm for large-scale optimization[J]. IEEE transactions on evolutionary computation, 2021, 25(4): 779–793.

[31] HE Xiaofei, YAN Shuicheng, HU Yuxiao, et al. Face recognition using Laplacianfaces[J]. IEEE transactions on pattern analysis and machine intelligence, 2005, 27(3): 328–340.

[32] CAI Deng, HE Xiaofei, HAN Jiawei, et al. Orthogonal laplacianfaces for face recognition[J]. IEEE transactions on image processing, 2006, 15(11): 3608–3614.

[33] CAI Deng, HE Xiaofei, HAN Jiawei, et al. Graph regularized nonnegative matrix factorization for data representation[J]. IEEE transactions on pattern analysis and machine intelligence, 2011, 33(8): 1548–1560.

[34] CHEN Ke, XUE Bing, ZHANG Mengjie, et al. An evolutionary multitasking-based feature selection method for high-dimensional classification[J]. IEEE transactions on cybernetics, 2022, 52(7): 7172–7186.

[35] JING Longlong, TIAN Yingli. Self-supervised visual feature learning with deep neural networks: a survey[J]. IEEE transactions on pattern analysis and machine intelligence, 2021, 43(11): 4037–4058.

[36] SARKAR P, ETEMAD A. Self-supervised ECG representation learning for emotion recognition[J]. IEEE transactions on affective computing, 2022, 13(3): 1541–1554.

[37] HSU W N, BOLTE B, TSAI Y H H, et al. HuBERT: self-supervised speech representation learning by masked prediction of hidden units[J]. IEEE/ACM transactions on audio, speech, and language processing, 2021, 29: 3451–3460.

[38] YUAN Aihong, YOU Mengbo, HE Dongjian, et al. Convex non-negative matrix factorization with adaptive graph for unsupervised feature selection[J]. IEEE transactions on cybernetics, 2022, 52(6): 5522–5534.

[39] YOU Mengbo, YUAN Aihong, ZOU Min, et al. Robust unsupervised feature selection via multi-group adaptive graph representation[EB/OL]. (2021–11–08)[2022–06–1]. https://ieeexplore.ieee.org/abstract/document/9606609.

[40] ZHANG Rui, ZHANG Yunxing, LI Xuelong. Unsupervised feature selection *via* adaptive graph learning and constraint[J]. IEEE transactions on neural networks and learning systems, 2022, 33(3): 1355–1362.

[41] CHANG Heng, GUO Jun, ZHU Wenwu. Rethinking embedded unsupervised feature selection: a simple joint approach[EB/OL]. (2022–05–30)[2022–06–01].https://ieeexplore.ieee.org/abstract/document/9784919.

[42] SOLORIO-FERNÁNDEZ S, CARRASCO-OCHOA J A, MARTÍNEZ-TRINIDAD J F. A review of unsupervised feature selection methods[J]. Artificial intelligence review, 2020, 53(2): 907–948.

[43] LI Jundong, CHENG Kewei, WANG Suhang, et al. Feature selection: a data perspective[J]. ACM computing surveys, 2018, 50(6): 94.

作者简介：

黎建宇，博士研究生，主要研究方向为人工智能、进化计算、群体智能、知识学习与数据驱动。

詹志辉，教授，博士生导师，Elsevier 中国高被引学者，主要研究方向为人工智能、进化计算、群体智能、云计算和大数据。荣获吴文俊人工智能优秀青年奖和 IEEE 计算智能学会杰出青年奖。目前已在国际期刊和国际会议发表（录用）论文共 150 余篇，其中 IEEE Transactions 系列的计算机领域顶尖国际期刊论文 60 余篇。论文近被国际同行引用超过一万次（Google Scholar），其中 SCI 引用超过 5000 次。11 篇论文先后入选 ESI 高被引（全球影响力排名前百分之一）论文，包括 1 篇 ESI 热点（全球影响力排名前千分之一）论文。

第 19 届中国智能系统会议 (CISC 2023)

中国智能系统会议是由中国人工智能学会智能空天系统专业委员会发起的系列学术会议，其宗旨是为本领域的专家学者、工程技术人员以及研究生提供一个学术交流平台，以推动我国智能系统相关理论、技术与应用的发展。第 19 届中国智能系统会议 (CISC 2023) 将于 2023 年 10 月 14~15 日在浙江省宁波市召开。

本次会议由中国人工智能学会主办，中国人工智能学会智能空天系统专业委员会与中国仿真学会人工智能仿真技术专业委员会协办，中科院宁波材料技术与工程研究所、北京精密机电控制设备研究所、航天伺服驱动与传动技术实验室与北京航空航天大学联合承办。会议论文集将由 Springer 出版社在 Lecture Notes in Electrical Engineering 系列正式出版，EI 收录。热忱欢迎海内外广大同仁踊跃投稿并出席本届会议，交流学术成果。

征文范围包括：多智能体系统、智能机器人、复杂网络与复杂系统、无人系统与集群行为、事件与数据驱动控制、拟人系统与人工生命、鲁棒与自适应控制、大数据与脑科学、过程控制、非线性系统与控制、智能传感器与检测技术、嵌入式系统与无线传感网络、智能交通与控制、深度学习与学习控制、信息物理系统、人工智能仿真技术、智能制造与云制造、伺服驱动与传动技术、电力系统及其自动化、模糊系统与神经网络 、航天智能发射系统、5G 与工业互联网等。论文采用网上投稿，投稿详情请浏览以下网址：https://easychair.org/conferences/?conf=cisc2023。

重要日期：

论文投稿截止日期：2023 年 4 月 30 日

论文录用通知日期：2023 年 6 月 20 日

会议注册/终稿提交截止日期：2023 年 7 月 10 日