Applied Intelligence (2019) 49:2889–2898
https://doi.org/10.1007/s10489-019-01420-9

A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection

Yong Zhang[1] · Hai-Gang Li[1] · Qing Wang[1] · Chao Peng[1]

Published online: 12 February 2019

Abstract

Due to good exploration capability, particle swarm optimization (PSO) has shown advantages on solving supervised feature selection problems. Compared with supervised and semi-supervised cases, unsupervised feature selection becomes very difficult as a result of no label information. This paper studies a novel PSO-based unsupervised feature selection method, called filter-based bare-bone particle swarm optimization algorithm (FBPSO). Two filter-based strategies are proposed to speed up the convergence of the algorithm. One is a space reduction strategy based on average mutual information, which is used to remove irrelevant and weakly relevant features fast; another is a local filter search strategy based on feature redundancy, which is used to improve the exploitation capability of the swarm. And, a feature similarity-based evaluation function and a parameter-free update strategy of particle are introduced to enhance the performance of FBPSO. Experimental results on some typical datasets confirm superiority and effectiveness of the proposed FBPSO.

Keywords Particle swarm optimization · Feature selection · Unsupervised

1 Introduction

With the fast development of technologies such as big data, the number of attributes (or features) in data obtained by decision-makers is growing at an unprecedented rate. Because of lacking sufficient prior knowledge on real problems, original data usually includes many redundant or/and irrelevant features in order to prevent losing useful information. Obviously, those irrelevant and redundant features must increases storage pressure and cost of computation systems [29, 34]. Most of all, those features may reduce the performance of adopted learning algorithm.

Feature selection (FS) is an effectively dimensional reduction method. It can select a subset of features from all original ones, resulting in reducing the learning cost and maximizing the performance of classification [6, 11, 14, 27, 36]. Now it has been applied into various high-dimensional data [8]. Based on the proportion of labeled samples to unlabeled samples, existing FS approaches can be classed into three categories: supervised, semi-supervised and unsupervised. Supervised or semi-supervised FS methods mainly use class label information to search optimal feature subset. However, in the era of information explosion, not all data can be labeled because of unaffordable costs. Therefore, recently studying unsupervised feature selection (UFS) has received more and more attention [15].

Up to now, researchers have proposed a variety of UFS methods by introducing correlation, information theory, structure information and clustering techniques. He et al. [16] employed Laplacian score to evaluate the effectiveness of features, and proposed a Laplacian score-based algorithm (LS). Cai *et al.* [9] proposed another effective UFS algorithm, called multi-cluster feature selection algorithm (MCFS), by using the L1-norm minimization regularization term. Wang et al. [31] integrated unsupervised trace ratio formulation and structured sparsity-inducing norms regularization, and proposed an improved UFS method (TRACK). By preserving the local manifold structure of data, these algorithms can effectively reduce irrelevant or redundant features. But they did not consider the local discrimination information which has been demonstrated as an essential property for analyzing data [7]. Li *et* al. [23] proposed a nonnegative discriminative UFS algorithm (NDFS) by employing spectral clustering to guide feature selection directly. Mitra et al. [26] proposed an UFS algorithm,

✉ Yong Zhang
yongzh401@126.com

✉ Hai-Gang Li
haigangli@cumt.edu.cn

[1] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China

called unsupervised feature selection using feature similarity (UFSFS), by considering the similarity between features. These clustering-based algorithms do not need any exhaustive search technique, but generally needs a right parameter set to control the size of reduced features. Studying the hybrid mechanism of multiple strategies, Tang et al. [30] proposed a unified UFS framework via feature self-representation and robust graph regularization, and Hou et al. [17] proposed a hybrid algorithm with joint embedding learning and sparse regression. These algorithms all demonstrate competitive results, but their results may still include more redundancy features due to the lack of effective global search strategy.

Since the capability of seeking solutions using global search strategies, recently evolutionary algorithms (EA) have received much attention on solving feature selection problems [32, 35]. However, due to the lack of effective strategies on evaluating individuals, there are relatively few studies on the applications of EA in UFS. Tabakhi et al. [28] proposed a new UFS method, called UFSACO, by using ant colony optimization as a global search technology. Due to the iterative and parallel nature, UFSACO can search in a greater feature space and select a good feature subset. However, it needs larger computational cost to construct a fully connected weighted graph among features in advance. To shorten the computing time of an algorithm, recently Kimovski et al. [22] discussed the parallel implement of multi-objective evolutionary optimization approaches in solving high-dimensional UFS problems. Bhadra and Bandyopadhyay [7] studied the application of a recently developed differential evolution technique, called MoDE, in UFS, and proposed an improved differential evolution based UFS algorithm. Abualigah et al. [2] proposed a harmony search-based UFS technique to solve the text clustering problem. These algorithms improve significantly the search ability of UFS technique, but they still face the problem of "curse of dimensionality" as the number of features increases.

As a relatively new evolutionary technique, particle swarm optimization (PSO) has advantages such as well global search, concise implementation, fast convergence. Up to now researchers have proposed many improved versions, and successfully applied them to various real problems, such as multi-objective optimization problems [25], classifier's parameter optimization [4], and robot navigation [38]. Recently a few researchers have attempted to study its application in UFS problems. Wang et al. [33] proposed an UFS algorithm based on Markov blanket and PSO (UFSPSO). This algorithm firstly filters out irrelevant features by using the maximum-entropy principle, and then combines PSO and Markov blanket to remove redundant features. Abualigah et al. [1] introduced a hybrid PSO with genetic operators for the unsupervised text feature selection problem. This method uses PSO to remove sparse and non-meaningful features from text. Iranmehr et al. [18] proposed a new PSO-based UFS algorithm for the problem of phonemes sound classification. Adeli and Broumandnia [3] studied a novel feature selection approach based on an adaptive inertia weight-based PSO, and applied successfully to image steganalysis problem. Those methods demonstrate the capability of PSO on solving UFS problems. However, there are still some disadvantages: (1) Most of those algorithms need the population to search optimal feature subset in the whole original feature space, so to face the problem of "Curse of Dimensionality" as the size of features increases; (2) most require decision-makers to set appropriate control parameters in advance, such as inertia weight and acceleration coefficients, for obtaining desirable solutions. But how to set those parameter values is still a challenge, because their values depend on individual applications or optimized problems.

Focused on these, this paper studies a novel unsupervised feature selection method by combining a control parameter-free PSO algorithm with two filter-based search strategies. Herein, the PSO algorithm is designed to find potentially optimal regions, while the two filter-based search strategies are used to improve the convergence speed of the proposed method. The highlights of this paper are as follows:

(1) By integrating the global search capacity of PSO with the fast local search capability of filter-based approach, a filter-based bare-bone particle swarm optimization algorithm is proposed for unsupervised feature selection problems.
(2) A filter-based strategy, called the space reduction strategy based on average mutual information, is given to remove irrelevant and weakly relevant features for reducing the search space of the swarm, as well as the cost of fitness evaluation.
(3) Focusing on exploiting potentially optimal regions obtained by the swarm, a local search strategy based on feature redundancy is proposed to improve the exploiting capability of the swarm.
(4) Moreover, a feature similarity-based fitness function and a parameter-free update strategy of particle are introduced to enhance the algorithm's performance and reduce the dependence of the algorithm on control parameters, respectively.

This paper is organized as follows: Section 2 shows related basic conceptions. Section 3 gives the proposed method, including the filter-based strategy based on average mutual information, and the improved PSO. Section 4 reports experimental results on several test datasets. Conclusions are presented in Section 5.

2 Related work

2.1 Problem formulation

Supposing that S is a data set which contains K samples and D features, F is a set of all features, then a UFS problem can be described as follows: to select d features ($d \leq D$) from all the features, so that appointed evaluation indicators (or objective function) $H(\cdot)$, such as the classification accuracy, are optimized. However, differing from supervised and semi-supervised cases that can directly use label information to evaluate selected features, generally UFS adopts implicit indicators, such as the proportion of predominant features to selected features [33] or the mean absolute difference [1], to evaluate selected features.

We adopt a binary string to encode a solution in UFS problems:

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

where $x_j = 1$ indicates the j-th feature is selected into the subset X; otherwise, it is not. So a UFS problem is formulated as follows:

$$\begin{array}{l} \max/\min H(X) \\ \text{s}.t.X = (x_1, x_2, ..., x_D), x_j \in \{0, 1\}, j = 1, 2, ..., D, \\ 1 \leq |X| \leq D, f(X) \in [0, 1]. \end{array} \tag{2}$$

2.2 Particle swarm optimization

As a swarm intelligence optimization technology, PSO is proposed by Kennedy and Eberhart by simulating the hunting behavior of bird [20]. Compared with other evolutionary algorithms (EAs), PSO abstracts individuals in the population into particles without mass and volume. Through information sharing and cooperation among the particles, the swarm can find optimal solutions from complex search spaces.

In traditional PSO, each particle has a velocity vector and a position vector, and the swarm looks for optimal solutions by constantly changing positions of all the particles. When PSO is applied to FS problem, each particle will represent a potential feature subset of the problem. Taking the i-th particle as example, let $X_i^t = (x_{i1}, x_{i1}, \cdots x_{iD})$ and $V_i^t = (v_{i1}, v_{i2}, \cdots v_{iD})$ are its position and velocity respectively, then this particle can be updated as follows:

$$v_{ij}^{t+1} = \omega v_{ij}^t + c_1 r_1 \left(Pb_{ij}^t - x_{ij}^t \right) + c_2 r_2 \left(Gb_{ij}^t - x_{ij}^t \right) \tag{3}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \tag{4}$$

Where, D represents the number of decision variables; $Pb_i^t = (Pb_{i1}, Pb_{i2}, \cdots, Pb_{iD})$ is the personal best position ($Pbest$) found by the i-th particle, and $Gb_i^t = (Gb_{i1}, Gb_{i2}, \cdots, Gb_{iD})$ is the global best position ($Gbest$) found by neighbors of the particle. r_1 and r_2 are random numbers uniformly distributed in [0, 1]. The inertia weight, ω, and the two acceleration coefficients, c_1 and c_2, are three control parameters, which are used to control the influences of the previous velocity, *Pbest* and *Gbest* on the search process of the swarm.

Traditional PSO was mainly designed to deal with continuous optimization problems. Focused on binary optimization problems, the literature [21] introduced a binary PSO algorithm (BPSO). In the BPSO, the eq. (3) is replaced as follows:

$$\begin{cases} x_{ij}^{t+1} = \begin{cases} 1 & othersise \\ 0 & if\, s\left(v_{ij}^{t+1}\right) < r_3 \end{cases} \\ s\left(v_{ij}^{t+1}\right) = 1/\left(1 + e^{-v_{ij}^{t+1}}\right) \end{cases} \tag{5}$$

Wherein, $s(\cdot)$ is a sigmoid function, $r_3$is a random values within [0,1].

Furthermore, Kennedy removed the three control parameters of the traditional PSO, and proposed a Gaussian sampling-based method to update the position of a particle, called bare-bones PSO (BBPSO) [19]. In details, the formula of updating particle is as follows:

$$x_{ij}^{t+1} = N\left(\frac{Pb_{ij}^t + Gb_{ij}^t}{2}, |Pb_{ij}^t - Gb_{ij}^t| \right) \tag{6}$$

In addition, Kennedy proposed other update formula as follows:

$$x_{ij}^{t+1} = \begin{cases} N\left(\frac{Pb_{ij}^t + Gb_{ij}^t}{2}, |Pb_{ij}^t - Gb_{ij}^t| \right) & r_4 < 0.5 \\ Pb_{ij}^t & otherwise \end{cases} \tag{7}$$

Compared with the traditional PSO presented by eqs. (3) and (4), BBPSO does not require the three control parameters, so it is more compact and more practical. Recently, Zhang et al. [37] proposed an improved BBPSO algorithm and applied it to supervised FS problems. The proposed BBPSO algorithm employs two strategies, the uniform mutation and the intensive memory strategy, to improve the search capability of BBPSO.

3 The proposed FBPSO algorithm

The proposed filter-based bare-bone particle swarm optimization algorithm (FBPSO) is introduced in this section. Firstly, a relevance measure is defined by means of mutual information to evaluate the correlation between features. Based on it, a space reduction strategy based on average mutual information is given. After that, the unsupervised feature selection based on BBPSO is introduced by integrating several new operators, i. e., the local search strategy based on feature redundancy and the feature similarity-based fitness function, together with several established techniques such as the parameter-free update strategy of particle, and the real encoding strategy.

3.1 Space reduction strategy based on average mutual information

For a supervised FS problem, the correlation between a feature and class labels can be directly calculated by some measures. After that, optimal feature subsets are determined by using those correlation values. However, in the unsupervised issues, there are no class labels to be employed directly. Therefore, how to evaluate the correlation between a feature and underlying classes needs to define suitable approaches.

Mutual information can be used to evaluate the interdependence between two features. The higher the relevant degree between two features is, the larger the mutual information between them is. Wang et al. [33] used the average mutual information (*AMI*) to evaluate the relevance of a feature to all the rest features, and stated that the greater the average mutual information of a feature, the higher the relevance of this feature in the dataset. In view of this, the average mutual information can partly reflect the correlation between a feature and potential classes. Therefore, this paper uses it to delete irrelevant and weakly relevant features in advance.

Supposing that $F = \{f_1, f_2, \cdots f_D\}$ represents a feature set, $Y_i = (y_{i1}, y_{i2}, \cdots y_{in})$ is sample value of the *i*-th feature f_i, the average mutual information of f_i is calculated as follows:

$$AMI(f_i) = H(f_i) - \sum_{j=1}^{D} \frac{H\left(f_i|f_j\right)}{D-1} \tag{8}$$

Where, $H(f_i)$ is the information entropy of f_i,

$$H(f_i) = -\sum_{y_{ij}\in Y_i} p\left(y_{ij}\right)\log_2 p\left(y_{ij}\right) \tag{9}$$

$H(f_i|f_j)$ is the conditional entropy f_i aboutf_j,

$$H\left(f_i|f_j\right) = H(f_i) + H\left(f_j\right) - MI\left(f_i, f_j\right) \tag{10}$$

$MI(f_i, f_j)$ is the mutual information (MI) between two features f_i and f_j,

$$MI\left(f_i, f_j\right) = \sum\sum p\left(f_i, f_j\right)\log_2 \frac{p\left(f_i, f_j\right)}{p(f_i)p\left(f_j\right)} \tag{11}$$

According to the average *MI* defined in Eq. (8), the correlation between a feature and potential classes can be calculated. Based on those correlation values, all the features can be divided into three categories: strongly relevant features, irrelevant features and weakly relevant features. Since irrelevant and weakly relevant features can enlarge the search space of PSO and increase its computational cost, it is necessary to remove these features before implementing PSO. Specifically, in this paper, a threshold θ is used. If the average mutual information of a feature is smaller than θ, the feature is regarded as an irrelevant or weakly relevant feature, and is removed. After removing those features, the search space of PSO used in Subsection 3.2 can be reduced significantly. Furthermore, Algorithm 1 shows pseudo code of the proposed space reduction strategy.

***Algorithm 1*：** The space reduction strategy based on AMI

```
Input:   The original feature set $F$; the threshold $\theta$ .
Output： The feature set after removing irrelevant and weakly relevant features, $F'$
Begin
While  $|F| > 0$      %$|F|$ represents the number of features among the set $F$
    Select randomly a feature from the set $F$, marked as $f_i$
    Calculate the $AMI$ value of the feature, $f_i$ , in the set $F$ by using Equation (8);
    If $AMI(f_i) \le \theta$ , delete the feature, $f_i$ , from the set $F$, i.e., $F = F / \{f_i\}$ ;
End While
Output  $F$
```

3.2 The improved BBPSO based on local filter search

By running the space reduction strategy above, we get a new reduced feature set which only contains strongly relevant features. To further remove redundant features from the set, this section presents an improved BBPSO-based feature selection approach with few control parameters. Compared with the standard PSO, BBPSO does not require the three control parameters including inertia weight and two acceleration coefficients, and is more compact and more practical. However, the standard BBPSO proposed in [19] still has disadvantages including premature convergence, especially when the *Pbest* of a particle happens to be close to *Gbest*. Focused on this, an improved BBPSO based on local filter search is presented to deal with UFS problems.

3.2.1 Encoding of the particles

In most papers, the binary string is usually used to represent a particle. In the string, if the value of a bit is equal to "1", then its corresponding feature is selected into feature subset; on the contrary, '0' indicates not. But this kind of encoding strategy needs the sigmoid function $s(\cdot)$ to transform the real velocity of a particle to a binary value.

Differing from the binary coding, this paper adopts a more direct method, i.e., real encoding strategy. This strategy uses the probability that each feature is selected as the coding element of a particle. Taking a data set with D features, the i-th particle is expressed as:

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{iD}) \tag{12}$$

Where, x_{ij} denotes the probability that the j-th feature is selected. Further, a threshold value 0.5 is defined to decide whether a feature is remained or not in the current set. $x_{ij} > 0.5$ indicates that the j-th feature is selected; otherwise, not selected.

3.2.2 Feature similarity-based evaluation function

All features can be divided into two parts: selected features (SF) and non-selected features (*NSF*), where $X_i = SF \cup NSF$ and $SF \cap NSF = \varnothing$. In this paper, we consider both the dissimilarity of selected features, fit', and the similarity of non-selected features, fit'', to evaluate the fitness of a particle.

The dissimilarity of selected features To calculate the dissimilarity of selected features, fit', the average of the maximal mutual information of each selected feature to remaining selected features is used as follows:

$$fit' = \frac{1}{|SF|}\sum_{i=1}^{|SF|} \max\_NMI(f_i) \tag{13}$$

Where, $\max\_NMI(f_i)$ is the maximal mutual information of the feature f_i to remaining selected features, i.e.,

$$\max\_NMI(f_i) = \max\left\{NMI\left(f_i, f_j\right) | f_j \in SF, f_i \neq f_j\right\} \tag{14}$$

$NMI(f_i, f_j)$ is the normalized mutual information,

$$NMI\left(f_i, f_j\right) = \frac{MI\left(f_i, f_j\right)}{\sqrt{H(f_i)H\left(f_j\right)}} \tag{15}$$

The smaller the value of fit' is, the smaller the redundancy between selected featues is.

The similarity of non-selected features To calculate the similarity of non-selected features,fit'', the average mutual information between each non-selected feature and its neighbors among selected features is used as follows:

$$fit'' = \frac{1}{|NSF|}\sum_{i=1}^{|NSF|} NMI(f_i, f_{\min}) \quad (f_i \in NSF, f_{\min} \in SF) \tag{16}$$

Where, $f_{\min}$ is the feature among the set SF which is closest to the non-selected feature f_i. A higher value of fit'' signifies that each non-selected feature can be represented by one among SF in some well manner.

Combining the similarity and dissimilarity items to ensure the representativeness of selected features, the fitness funtion is defined as the following problem:

$$fit = fit'' - fit' \tag{17}$$

The smaller the value of fit, the better the quality of selected features.

3.2.3 Local filter search based feature redundancy

To improve the local exploitation performance of the swarm, this subsection proposes a local filter search strategy based on feature redundancy. This strategy includes mainly two operators: the deleting operator and the adding operator. The deleting operator removes redundant features from a feature subset, while the adding operator inserts missing key features into a feature subset.

For the global best position, *Gbest*, found by the swarm, supposing that the feature set after removing irrelevant and weakly relevant features by the method in Section 3.1 is F', and the feature subset determined by the position *Gbest* is *Fset*, the two operators are described as follows:

The deleting operator Firstly, this operator selects randomly two features from the set *Fset*, marked as f_{l1} and f_{l2}; Second, for each feature among $\{f_{l1}, f_{l2}\}$, run the k-Nearest Neighbor algorithm to select its k nearest neighbors from the set *Fset,* marked their neighbors as $kNN(f_{l1})$ and $kNN(f_{l2})$ respectively; Next, calculate the average normalized mutual information (A_NMI) between this feature and its neighbors as follows:

$$A\_NMI(f_i) = \sum_{f_j \in kNN(f_i)} \frac{NMI\left(f_i, f_j\right)}{k}, f_i \in \{f_{l1}, f_{l2}\} \tag{18}$$

After that, the one with larger A_NMI value among $\{f_{l1}, f_{l2}\}$ is removed from *Fset*.

The adding operator First, this operator selects randomly two features from the set $F'/Fset$, marked as f_{l1}' and f_{l2}'; After that, using the above method, calculate the average normalized mutual information (A_NMI) between each feature and its neighbors; and then the one with smaller A_NMI value among $\left\{f_{l1}', f_{l2}'\right\}$ is added into the set *Fset*.

Furthermore, Algorithm 2 shows steps of the proposed local search strategy. First, **Step 1** implements the deleting operator, and deletes the redundant one among $\{f_{l1}, f_{l2}\}$ from *Fset*. Second, **Step 2** implements the adding operator, and adds the important one among $\left\{f_{l1}', f_{l2}'\right\}$ into the set *Fset*. Next, **Step 3** generates a new position, $Gbest'$, by using the features included in the set *Fset*. For each feature among *Fset,* set its corresponding feature bit on $Gbest'$to 1, and set all the rest feature bits to 0. Finally, **Step 4** updates the global best position *Gbest*. If $Gbest'$ is better than *Gbest*, the *Gbest* is replaced by $Gbest'$; otherwise, we use the position $Gbest'$to replace the worst particle among the swarm.

***Algorithm 2*：** The proposed local filter search based feature redundancy

```
Input:   The position, $Gbest$; the feature subset, $Fset$, determined by $Gbest$; the feature set $F'$
Output:  The new position of $Gbest$
Step1:  Implement the deleting operator.
        Select randomly two features from $Fset$, marked as $f_{l1}$ and $f_{l2}$;
        Calculate their $A\_NMI$ value, $A\_NMI(f_{l1})$ and $A\_NMI(f_{l2})$;
        Delete the redundant one among $\{f_{l1}, f_{l2}\}$ from $Fset$.
Step 2:  Implement the adding operator.
        Select randomly two features from the set $F' / Fset$, marked as $f_{l1}'$ and $f_{l2}'$;
        Calculate their $A\_NMI$ value, $A\_NMI(f_{l1}')$ and $A\_NMI(f_{l2}')$;
        Add the one with smaller $A\_NMI$ value among $\{f_{l1}', f_{l2}'\}$ into the set $Fset$.
Step 3: Generate a new position, $Gbest'$, based on the set $Fset$.
Step 4: Updates the position $Gbest$. If $fit(Gbest') \leq fit(Gbest)$, set $Gbest = Gbest'$; otherwise, keep
        $Gbest$ unchanged, and replace the worst particle among the swarm by $Gbest'$.
```

3.3 The steps of the proposed FBPSO

According to the above work, we describe steps of the proposed unsupervised feature selection algorithm as follows:

- **Step 1:** Implement the space reduction strategy. Calculate the average mutual information of all features according to Eq. (8), and remove irrelevant and weakly relevant features based on the method in Subsection 3.1;
- **Step 2:** Initialize all the particles, and evaluate their fitness value by Eq. (17). For each particle, the *Pbest* is initialized as its oneself position, and the *Gbest* is set to the particle with the most fitness value;
- **Step 3:** For every particle in the swarm, implement the following steps circularly:

Table 1 The selected data sets

datasets	# of features	# of samples	# of classes	type of data
Musk1	166	476	2	biological data
Arrhythmia	279	452	2	biological data
AR10P	2400	130	10	image data
ORL10P	10,304	100	40	image data
PCMAC	3289	1943	2	text data
BASEHOCK	4862	1993	2	text data

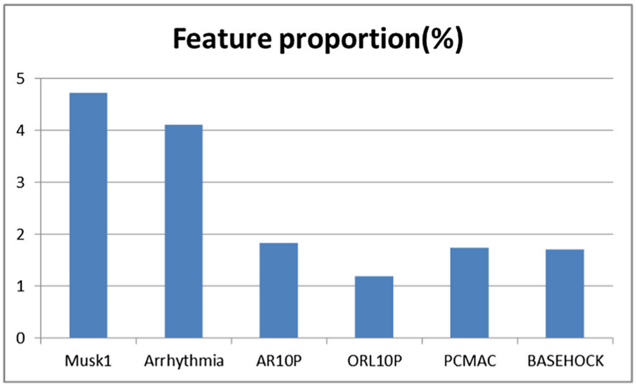

Fig. 1 The PRO values obtained by FBPSO

- **Step 3.1:** Evaluate the fitness values of each particle by Eq. (17).
- **Step 3.2:** Update the *Pbest* and *Gbest* of each particle by the following standard strategy. For a particle, if its new position is better than the current *Pbest*, the current *Pbest* is replaced by the new position. If the new *Pbest* is better than the current *Gbest*, *Pbest* replaces the current *Gbest*.
- **Step 3.3:** Update the position of each particle by Eq. (7);
- **Step 3.3:** Run the local filter search proposed in Subsection 3.2;
- **Step 3.4:** Loop to Step 3.1 until the termination condition is met.

4 Experiments

This section verifies the performance of the FBPSO algorithm, by comparing with some existing unsupervised feature selection algorithms on several frequently-used data sets.

4.1 Experiment settings

The proposed FBPSO was tested on six data sets, including the biological data Musk1 and Arrhythmia, the image data AR10P and ORL10P, the text data PAMAC and BASEHOCK. The first two data sets are from the UCI repository [10], the rest come from the ASU repository [39]. Table 1 gives general information about these data sets. These data sets have a wide range of spans and representations, with sample sizes ranging from 100 to 2000, and feature numbers from 166 to 10,304. Therefore, they can provide comprehensive and detailed testing for the proposed algorithm under different conditions.

We compare the proposed FBPSO algorithm with MCFS [9], TRACK [31], UFSACO [28] and UFSPSO [33]. Where, MCFS and TRACK belong to non-heuristic approaches, and they have been used in many literatures such as [7, 9, 15, 30, 33]. In our experiments, the two algorithms are mainly used to prove the superiority of EA-based approaches compared with non-heuristic approaches. UFSACO and UFSPSO are two representations of existing EA-based approaches. As we all know, ACO and PSO are two very universal swarm intelligence optimized algorithms. The two algorithms are mainly used to validate the effectiveness of our improved PSO-based algorithm, compared with existing EA-based approaches.

Since all data sets have different feature size, the maximum correlation entropy in the FBPSO and UFSPSO method is set to the entropy relevance value of the $\lceil D\log_2 D\rceil - th$ ranked feature for each dataset, and 10-fold cross-validation is used, where $\lceil . \rceil$ is an upward integral function. We set the swarm/population size to 40, the maximal iteration times to 100. The 1-nearest neighbor (1NN) and the decision tree classifier (C4.5) are introduced to calculate the classification accuracy of a feature subset. In order to reduce the impact of randomness, all algorithms are run 20 times on each data set to obtain their statistical results.

4.2 Experimental analysis

We evaluate the proposed FBPSO algorithm by the following two aspects: the proportion of selected features to all features (PRO) and the classification accuracy (ACC) [40]. Fig. 1 shows the PRO value obtained by FBPSO. It can be seen that FBPSO significantly reduces the number of features, where all the PRO values locate within the range 4.72% to 1.19%. It should be noted that the bigger the data size, the smaller the PRO value obtained by FBPSO.

Compared FBPSO with other four feature selection methods, Tables 2 and 3 report their ACC values by using

Table 2 Average ACC values found by all the algorithms (1NN)

Datasets	All Feature	TRACK	MCFS	UFSACO	UFSPSO	FBPSO
Musk1	80.76-	**81.50-**	79.20=	80.22=	79.83=	79.85
Arrhythmia	64.29+	69.20+	68.19+	69.75+	70.42+	**70.74**
AR10P	66.92+	69.60+	66.15+	71.79+	73.00+	**73.31**
ORL10P	88.00+	89.81=	**90.00-**	89.27=	89.50=	89.53
PCMAC	76.36+	76.50+	76.60+	76.06+	77.98=	**78.09**
BASEHOCK	81.15+	81.09+	81.02+	81.41+	81.47+	**82.47**
Average accuracy	76.24	77.95	76.86	78.08	**78.70**	**79.01**

Table 3 Average classification accuracy obtained by all the algorithms (C4.5)

Datasets	All Feature	TRACK	MCFS	UFSACO	UFSPSO	FBPSO
Musk1	77.04+	77.88+	76.11+	78.56+	79.62=	**79.67**
Arrhythmia	66.37+	66.45+	63.56+	66.49+	67.04=	**67.15**
AR10P	69.31+	70.00+	69.62+	70.05+	70.40=	**70.58**
ORL10P	81.67=	**84.00-**	**84.00-**	81.33+	81.40+	81.72
PCMAC	77.51+	77.84+	77.72+	77.90+	78.18+	**78.38**
BASEHOCK	**85.08-**	82.44+	82.31+	82.07+	84.84=	84.62
Average accuracy	76.16	76.44	75.55	76.07	76.91	**77.02**

two classifiers and their statistical results, respectively. In the two Tables, the baseline column is the ACC values obtained by the classifier when all features are selected, other columns are average ACC values obtained by different methods, and the last row is the average classification accuracy of an algorithm on different data sets. The best classification results are highlighted at blackbody for the corresponding feature selection method. Moreover, according to the suggestion in [12], the Wilcoxon rank sum test with the significant level of 0.05 is used to show the statistical significance of results. Here the symbol "+" indicates that the null hypothesis (i.e., the median difference between two algorithms is zero) cannot be rejected at the 5% level, and FBPSO is significantly better than the compared one. The symbol "-" indicates that the null hypothesis cannot also be rejected at the 5% level, but FBPSO is significantly worse than the compared one. The symbol "=" indicates that the null hypothesis can be rejected at the 5% level, and means that the differences between FBPSO and the compared one are not significant.

From the 1NN-based classification results given in Table 2, it can be seen that for the data sets Arrhythmia, AR10P and BASEHOCK, the ACC values of FBPSO is significantly better than that obtained by the four comparison algorithms, i.e., TRACK, MCFS, UFSACO, and UFSPSO. For the data sets, Arrhythmia, AR10P, PCMAC and BASEHOCK, the FBPSO algorithm obtained the highest ACC values. As the last line of Table 2 shown, the FBPSO algorithm obtained the best average ACC values for all the six data sets. UFSPSO also obtained good ACC values similar to FBPSO for the data sets, Musk1, ORL10P and PCMAC, and achieved the second best values in terms of the average classification accuracy, as shown in the last line of Table 2. Compared with the baseline method, the classification accuracies of TRACK, MCFS, UFSACO, UFSPSO and FBPSO are increased by 1.71%, 0.62%, 1.84%, 2.46%, and 2.77%, respectively.

Table 3 shows the ACC values of different algorithms under the C4.5 classifier. We can see that for all the six data sets, the performance of FBPSO is significantly better than the three comparison algorithms, TRACK, MCFS, UFSACO, in terms of the ACC values. UFSPSO also obtained good ACC values similar to FBPSO for the data sets, Musk1, Arrhythmia, AR10P and BASEHOCK, but its performance is significantly worse than FBPSO on the data sets ORL10P and PCMAC. Moreover, on four out of the six data sets, i.e., Musk1, Arrhythmia, AR10P and PCMAC, the FBPSO algorithm has higher ACC values than the other four comparison methods. Compared with the baseline method, TRACK, MCFS and UFSACO, the classification accuracies of FBPSO are increased by 0.86%, 0.58%, 1.47% and 0.95%, respectively. Therefore, the performance of FBPSO is also better than all the four comparison algorithms.

Overall, we can see from the above results that the proposed FBPSO algorithm can optimize the feature selection process and improve the classification accuracy of data. It is a competitive data pre-processing tool.

4.3 Further discussion

This subsection evaluates the effectiveness of our proposed local filter search strategy. Here, FBPSO without the local filter search is denoted as FBPSO/LS. Fig. 2 shows experimental results of both FBPSO and FBPSO/LS. We can see that for all the datasets, under the help of the local search operator, FBPSO obtains higher classification accuracies than FBPSO/LS. Taking the dataset AR10P as example, FBPSO achieves a better classification accuracy value, 73.24, which is 1.42 points higher than FBPSO/LS. Overall, the local search strategy plays a key role in improving the performance of FBPSO.

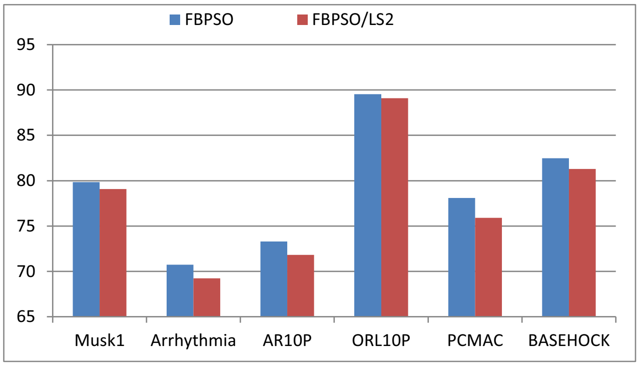

Fig. 2 Average classification accuracy obtained by FBPSO and FBPSO/LS

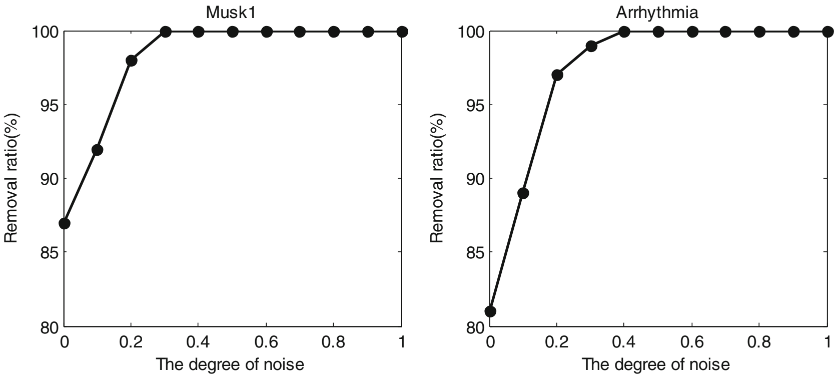

Fig. 3 The removal ratios on new data sets with differently level noises

Moreover, this paper uses a space reduction strategy based on average mutual information to remove irrelevant and weakly relevant features, as shown in Subsection 3.1. Now we design another experiment to test the effectiveness of this strategy. In this experiment, 20 new data sets are constructed based on the two data sets, Musk1 and Arrhythmia. Taking Musk1 as example, first we select randomly its 100 features, and then add a certain degree of noise to these features by the following method: $v(x_i) = (1 + \lambda \times rand) \times v(x_i)$, $i = 1, 2\cdots$, 100. Here, $v(x_i)$ represents the value of the i-th selected feature, λ is the degree of noise, *rand* is a random number within [−1,1]. In our experiment, we set λ to {0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1} respectively to generate ten new data sets.

Fig. 3 shows the removal ratios of features from the 100 features by implementing the space reduction strategy on new data sets. We can see that for the two original data sets (i.e., Musk1 and Arrhythmia when $\lambda = 0$), only 87 and 81 out of the 100 features are removed respectively. The main reason is that the 100 features include strongly relevant features. However, as the noise degree increases, the relevant degrees of all the 100 features become gradually weak. Correspondingly, the removal ratios increase rapidly under the help of the space reduction strategy. When the noise degree is equal to 0.2, the removed ratios have already achieved 98% and 97% for the two data sets, Musk1 and Arrhythmia, respectively. When $\lambda \geq$ 0.4, all the 100 new features are removed because these features have become totally irrelevant under the action of strong noise. Therefore, the space reduction strategy has good capability on removing irrelevant and weakly relevant features.

5 Conclusion

This paper presented a new unsupervised feature selection algorithm, the filter-based bare-bone particle swarm optimization algorithm, FBPSO. The space reduction strategy based on average mutual information, the problem-specific local search strategy based on feature redundancy and the feature similarity-based evaluation function proposed in this paper, together with several established techniques such as the parameter-free update strategy of particle, and the real encoding strategy, all have made the FBPSO algorithm more effective in dealing with UFS problems. Finally, the experimental results on six datasets showed that the proposed FBPSO algorithm can not only ensure the classification accuracy, but also significantly reduce the number of selected features, and it is a highly competitive unsupervised feature selection method.

Generally, a feature selection problem contains two main objectives, i.e., the number of features and the classification accuracy. An important topic for further research is to study the applications of typical multi-objective particle swarm optimization technologies such as vortex multi-objective PSO [24] in feature selection problems. Another venue of research is to apply the developed algorithms to various real feature selection problems presented in cancer diagnosis [5], image recognition [13], and other practical application areas.

Acknowledgements This work was jointly supported by the National Natural Science Foundation of China (No. 61876185), and Six Talents Peaks Project of Jiangsu Province (No. DZXX-053).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73(11):4773–4795
2. Abualigah LM, Khader AT, Al-Betar MA (2016) Unsupervised feature selection technique based on harmony search algorithm for improving the text clustering. In: the 7th International Conference on Computer Science and Information Technology (pp.1–6)

3. Adeli A, Broumandnia A (2018) Image steganalysis using improved particle swarm optimization based feature selection. Appl Intell 48(6):1609–1622
4. Arun V, Krishna M, Arunkumar BV, Padma SK, Shyam V (2018) Exploratory boosted feature selection and neural network framework for depression classification. International Journal of Interactive Multimedia and Artificial Intelligence 5(3):61–71
5. Bagwari P, Saxena B, Balodhi M, Bijalwan V (2017) Comparison of feedforward network and radial basis function to detect leukemia. International Journal of Interactive Multimedia and Artificial Intelligence 4(5):55–57
6. Barani F, Mirhosseini M, Nezamabadi-pour H (2017) Application of binary quantum-inspired gravitational search algorithm in feature subset selection. Appl Intell 47(2):304–318
7. Bhadra T, Bandyopadhyay S (2015) Unsupervised feature selection using an improved version of Differential Evolution. Expert Syst Appl 42(8):4042–4053
8. Bi N, Tan J, Lai JH, Suen CY (2018) High-dimensional supervised feature selection via optimized kernel mutual information. Expert Syst Appl 108:81–95
9. Cai D, Zhang C, He X (2010) Unsupervised feature selection for multi-cluster data. In: ACM International Conference on Knowledge Discovery and Data Mining (pp.333-342)
10. Dua D, Karra Taniskidou E (2017) UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml
11. Gao WF, Hu L, Zhang P, Wang F (2018) Feature selection by integrating two groups of feature evaluation criteria. Expert Syst Appl 110:11–19
12. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heuristics 15(6):617–644
13. Gong DW, Liu K (2018) A multi-objective optimization model and its evolution-based solutions for the fingertip localization problem. Pattern Recogn 74:385–405
14. Hancer E, Xue B, Zhang MJ, Karaboga D, Akay B (2018) Pareto front feature selection based on artificial bee colony optimization. Inf Sci 422:462–479
15. He JR, Bi YZ, Ding LX, Li ZK, Wang SW (2017) Unsupervised feature selection based on decision graph. Neural Comput & Applic 28(10):3047–3059
16. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. In: International Conference on Neural Information Processing Systems (pp. 507–514)
17. Hou C, Nie F, Li X, Yi D, Wu Y (2014) Joint embedding learning and sparse regression: a framework for unsupervised feature selection. IEEE Transactions on Cybernetics 44(6):793–804
18. Iranmehr E, Shouraki SB, Faraji MM (2017) Unsupervised feature selection for phoneme sound classification using particle swarm optimization. In: Iranian Joint Congress on Fuzzy and Intelligent Systems (pp. 86–90)
19. Kennedy J (2003) Bare-bones particle swarms. In: Proceedings of the Swarm Intelligence Symposium (pp. 80–87)
20. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE International Conference on Neural Networks (pp.1–7)
21. Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. IEEE International Conference on Systems 5:4104–4108
22. Kimovski D, Ortega J, Ortiz A, Banos R (2015) Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection. Expert Syst Appl 42(9):4239–4252
23. Li Z, Yang Y, Liu J, Zhou X, Lu H (2012) Unsupervised feature selection using nonnegative spectral analysis. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (pp.1026–1032)
24. Meza J, Espitia H, Montenegro C, Giménez E, González R (2017) MOVPSO: vortex multi-objective particle swarm optimization. Appl Soft Comput 52:1042–1057
25. Meza J, Espitia H, Montenegro C, Crespo RG (2016) Statistical analysis of a multi-objective optimization algorithm based on a model of particles with vorticity behavior. Soft Comput 20(9): 3521–3536
26. Mitra P, Murthy CA, Pal SK (2002) Unsupervised feature selection using feature similarity. IEEE Transactions on Pattern Analysis & Machine Intelligence 24(3):301–312
27. Sheikhpour R, Sarram MA, Sheikhpour E (2018) Semi-supervised sparse feature selection via graph Laplacian based scatter matrix for regression problems. Inf Sci 468:14–28
28. Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell 32(6):112–123
29. Talukdar U, Hazarika SM, Gan JQ (2018) A Kernel Partial least square based feature selection method. Pattern Recogn 83:91–106
30. Tang C, Zhu XZ, Chen JJ, Wang PC, Liu XW, Tian J (2018) Robust graph regularized unsupervised feature selection. Expert Syst Appl 96:64–76
31. Wang D, Nie F, Huang H (2014) Unsupervised feature selection via unified trace ratio formulation and k-means clustering. In: Proceeding of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 306–321)
32. Wang Y, Feng LZ, Zhu JM (2018) Novel artificial bee colony based feature selection method for filtering redundant information. Appl Intell 48(4):868–885
33. Wang Y, Wang J, Liao H (2017) Unsupervised feature selection based on Markov blanket and particle swarm optimization. J Syst Eng Electron 28(1):151–161
34. Zhai Y, Ong YS, Tsang IW (2014) The emerging "Big Dimensionality". IEEE Comput Intell Mag 9(3):14–26
35. Zhang Y, Song XF, Gong DW (2017) A return-cost-based binary firefly algorithm for feature selection. Inf Sci 418:561–574
36. Zhang Y, Gong DW, Cheng J (2017) Multi-objective particle swarm optimization approach for cost-based feature selection in classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics 14(1):64–75
37. Zhang Y, Gong DW, Hu Y (2015) Feature selection algorithm based on bare-bones particle swarm optimization. Neurocomputing 148(1):150–157
38. Zhang Y, Gong DW, Zhang JH (2013) Robotic path planning in uncertain environment using multi-objective particle swarm optimization. Neurocomputing 103:172–185
39. Zhao Z, Morstatter F, Sharma S, Anand A, Liu H (2010) Advancing feature selection research. Arizona State University, Phoehix
40. Zhu PF, Xu Q, Hu QH, Zhang CQ (2018) Co-regularized unsupervised feature selection. Neurocomputing 275:2855–2863