An Evolutionary Multitasking-Based Feature Selection Method for High-Dimensional Classification

Ke Chen, Graduate Student Member , IEEE, Bing Xue , Member , IEEE, Mengjie Zhang , Fellow, IEEE, and Fengyu Zhou

Abstract

Feature selection (FS) is an important data preprocessing technique in data mining and machine learning, which aims to select a small subset of information features to increase the performance and reduce the dimensionality. Particle swarm optimization (PSO) has been successfully applied to FS due to being efficient and easy to implement. However, most of the existing PSO-based FS methods face the problems of trapping into local optima and computationally expensive high-dimensional data. Multifactorial optimization (MFO), as an effective evolutionary multitasking paradigm, has been widely used for solving complex problems through implicit knowledge transfer between related tasks. Inspired by MFO, this study proposes a novel PSO-based FS method to solve high-dimensional classification via information sharing between two related tasks generated from a dataset. To be specific, two related tasks about the target concept are established by evaluating the importance of features. A new crossover operator, called assortative mating, is applied to share information between these two related tasks. In addition, two mechanisms, which are variable-range strategy and subset updating mechanism, are also developed to reduce the search space and maintain the diversity of the population, respectively. The results show that the proposed FS method can achieve higher classification accuracy with a smaller feature subset in a reasonable time than the state-of-theart FS methods on the examined high-dimensional classification problems.

Keywords: Evolutionary multitasking, feature selec tion (FS), high-dimensional classification, particle swarm optimization (PSO).

1 Introduction

With the rapid development of data and knowledgemanagement technologies, the amount of data collected in many real-world applications is growing exponentially. Ideally, the features in these datasets are useful. However, these datasets frequently have a large number of irrelevant and redundant features, which significantly degrade the performance of learning algorithms[1]. Feature selection (FS) is a critical data preprocessing technique to facilitate data analysis and select relevant features of a dataset[2]. Due to its effectiveness, FS has been extensively applied to solve different problems, such as time-series forecasting[3]and job shop scheduling[4]. FS is a combinatorial optimization problem[5].A dataset has n original features, which has 2^n possible feature subset combinations for choosing one of them. However, the identification of relevant features is still a challenging problem in high-dimensional data due to its huge search space and complex interactions between features[6].

2 Related works

2.1 FS approaches to handle the classification problems

Researchers have presented many FS approaches to handle the classification problems, which can be categorized into filter-based and wrapper-based approaches[7]. Filterbased approaches use the intrinsic characteristics of the data (e.g., distance, dependence, consistency, and information measures) without involving any learning algorithm to search for a feature subset[8]. Wrapper-based approaches search through the feature space using the predicted accuracy from a learning algorithm to measure selected features. In general, wrapper-based approaches can achieve higher classification accuracy than filter-based approaches. However, these approaches are computationally expensive and normally specific to a particular wrapped learning algorithm[9]. On the contrary, filter-based approaches are typically computationally less expensive, but often achieve lower classification accuracy[10]. Hybrid methods have been presented to make use of the advantages of the filter-based and wrapper-based methods[11].

Since hybrid FS methods show great promise in dealing with high-dimensional classification problems, many hybrid FS methods are presented in[12] and[13]. In these methods, filter-based approaches are typically used to measure the importance of features, and wrapper-based methods are applied to determine the final feature subset based on the importance of features. In general, there are two ways to utilize the importance of features in the wrapper-based approaches. One is reducing the search space by eliminating unimportant features[12],[14]. These methods are effective because a large number of redundant and irrelevant features are removed before applying the wrapper-based approaches. However, it is not trivial to determine the number of unimportant features to remove in the filter operator without a certain degree of domain knowledge. The other is to use the correlation information between features to design some strategies for specific problems in the wrapper operator[13],[15]. These FS methods are effective because the strategy based on the relevance of features can avoid invalid search during the FS process. However, these methods usually require more time in strategy calculations, especially when the number of features is large. In addition, for these two ways of hybrid FS methods, some potentially useful information may be ignored for further investigation since the importance of features is measured in the filter operator individually (i.e., the interaction between features is not considered). This may take the risk of falling into local optima and loss the classification accuracy. Therefore, the hybrid FS method for the high-dimensional classification problems still requires further investigation.

2.2 Particle swarm optimization (PSO)

Particle swarm optimization (PSO)[16], as a populationbased search algorithm, has been widely applied for solving FS problems[17][18][19]. This is because PSO has the advantages of being easy to implement and has strong global searchability. Up to now, a large number of PSO-based hybrid FS approaches have been presented in the literature[20],[21]. For instance, a PSO-based hybrid FS method was developed in[20], where a filter algorithm is utilized to decrease the search space before using the minimum description length principle to generate potential cut-points, and then an adaptive PSO is applied to find a compact feature subset. In[21], a PSO-based unsupervised FS method was proposed, where two filter-based strategies are embedded in PSO to find a less correlated feature subset. However, most of the existing PSO-based hybrid FS methods still face the limitations of trapping in local optima and high computational cost during the search process, especially when the search space is large.

2.3 Evolutionary multitasking

Evolutionary multitasking[22] is an emerging research topic in the field of optimization. The success of evolutionary multitasking relies on knowledge transfer, which refers to sharing or disseminating of knowledge between different but related problems and providing assistance to problem-solving during the search process[23],[24]. Multifactorial optimization (MFO) is an evolutionary multitasking paradigm introduced by Gupta *et* al. in 2015[25], which has been successfully used for solving different optimization problems in practice, such as path optimization problems[26] and parameter estimation problems[27]. This is because MFO has the advantages of improving global search abilities and shortening computational time of complex problems with the help of information sharing between multiple related tasks. Multifactorial PSO (MFPSO)[28] was proposed under the MFO paradigm, which can effectively share knowledge between different related tasks through the operators of the assortative mating and vertical cultural transmission during the search process.

2.4 The proposed new algorithm PSO-EMT

Inspired by the effectiveness of information sharing in MFO for solving complex problems, this article proposes a novel hybrid FS method based on MFPSO for high-dimensional classification to improve the classification performance and shorten computational time via knowledge transfer between two related FS tasks. Different from the traditional usage of evolutionary multitasking that involves tasks of different problems/-datasets, two FS tasks in this study are composed from the same dataset. Specifically, one FS task is to select features from the entire feature set, and another FS task is to select features from a promising subset of the available features (e.g., determined using the developed knee point selection scheme). Since these two feature sets share some common features and have the same tasks of predicting the same set of class labels, FS tasks based on these two feature sets can be used to form a multitasking optimization problem. In this multitasking system, knowledge from the FS task with the entire feature set is used to help the FS task with the promising feature subset jump out of possible local optima to find a better feature subset.

The overall goal of this article is to develop an effective FS method based on evolutionary multitasking to improve the performance of PSO for FS on high-dimensional data. The proposed approach is expected to achieve high classification accuracy with a small feature subset. Specifically, the main contributions of this article are given as follows.

1. We proposed an effective FS method based on the idea of multitasking with knowledge transfer between different search space on high-dimensional data.
2. We designed a knee point selection scheme to distinguish promising features from all features to help generate related FS task for forming a multitasking FS system.
3. We proposed a novel variable-range strategy that defines the range of the search space based on the importance of features, which can effectively reduce the search space of the population.

4. We developed a subset updating mechanism, which can effectively maintain the diversity of the population during the evolutionary process.

The main novelty of the proposed method is to introduce evolutionary multitasking for FS on high-dimensional data. The knowledge transfer between the search space with the promising feature subset and the entire feature set cannot only focus on the exploitation in the small search space with promising features but also maintain all the feature information of the dataset during the FS process. To the best of our knowledge, this is the first work that uses the idea of multitasking to solve high-dimensional FS problems.

3 Method

This section provides a brief introduction of evolutionary multitasking, multifactorial particle swarm optimization, and related work on PSO for FS in high-dimensional classification.

3.1 Evolutionary Multitasking

Evolutionary multitasking is the study of how to use evolutionary computation algorithms to solve multiple tasks simultaneously via knowledge transfer between these tasks[22]. In a multitasking scenario, processing one task may help address other search tasks, because there often holds some useful knowledge in common with related problems. For example, a multitasking framework has K tasks. The k th task is denoted as T_k and its search space is X_k with a fitness function f_k. The multitasking framework can be described as

$$\{x_1^*, x_2^*, \ldots, x_K^*\}. \quad = \{\text{argmin}\, f_1\left(x_1\right), \text{argmin}\, f_2\left(x_2\right), \ldots, \text{argmin}\, f_K\left(x_K\right))\} \tag{1}$$

where x_k^* denotes the optimal global solution of T_k.

MFO[25] is a novel evolutionary multitasking paradigm that builds on the implicit parallelism of population-based search algorithms to address multiple problems simultaneously. Each task T_k in the multitasking framework can be considered as a factor affecting the search process of the individuals in the multitasking framework. In MFO, the uniform random key scheme[25] was used to encode the feasible search space of specific tasks into a unified representation space. Thus, each individual p_i in the population can be decoded into a task-specific solution for each task. Several definitions related to p_i in MFO are shown as follows[25].

Factorial Cost: The factorial cost of individual p_i on task T_k is the fitness value of solution p_i.

Factorial Rank: The factorial rank of p_i on T_k is the index of p_i in the list of individuals in the population sorted in an ascending order (from best to worse) with respect to the fitness function value f_k which is noted as r_k^i.

Skill Factor: The skill factor τ_i of p_i is defined by the index of T_k to decide which task is solved by $p_i.\tau_i$ of p_i is given by $\tau_i = \text{argmin}_k\left(r_k^i\right)$.

Scalar Fitness: The scalar fitness of p_i is given by $\varphi_i = 1/\min\left(r_k^i\right)$.

In MFO, there are two important factors. One is the scalar fitness that is used as the unified performance criterion in a multitasking framework. The other is the skill factor that is applied to allocate the task for a specific individual during the search process.

3.2 Multifactorial Particle Swarm Optimization

PSO[16],[29] is a population-based search algorithm. In PSO, each particle has its own position, which is a potential solution. During the updating process in the D dimensional problem space, the position of the i th particle can be represented by a vector $\boldsymbol{X}_i = [X_{i1}, X_{i2}, \ldots, X_{iD}]$, where $X_{\text{id}} \in [X_{\text{min}}, X_{\text{max}}]$, and the corresponding velocity is $\boldsymbol{V}_i = [V_{i1}, V_{i2}, \ldots, V_{iD}]$, where $V_{\text{id}} \in [V_{\text{min}}, V_{\text{max}}]$. The previously best position of the i th particle is represented by pbest $_i = [p \text{ best }_{i1}, p \text{ best }_{i2}, \ldots, p \text{ best }_{iD}]$, and the global best position so far is represented as gbest $=$ [gbest $_1$, gbest $_2, \ldots$, gbest $_D]$. pbest, gbest, and the previous velocity are used to update the i th particle velocity. Equations (2) and (3) are applied to update velocity and position, respectively

$$\begin{aligned} V_{\text{id}} =& \omega * V_{\text{id}} + c_1 * r_1 * (\text{ best }_{\text{id}} - X_{\text{id}}) \\ & + c_2 * r_2 * (\text{ beest }_d - X_{\text{id}}) \\ X_{\text{id}} =& X_{\text{id}} + V_{\text{id}} \end{aligned} \tag{2}$$

where ω denotes the inertia weight. c_1 and c_2 are acceleration constants. r_1 and r_2 are two uniformly distributed values in the range $[0, 1]$.

MFPSO[28] is an effective and efficient implementation of MFO that incorporates PSO into the MFO paradigm. The basic procedure of MFPSO is described in Algorithm 1. The scalar fitness is adopted to evaluate the performance of individuals in MFPSO. Assortative mating and vertical cultural transmission are the two critical operators in MFPSO for implementing multitasking with the information transfer between individuals. Specifically, assortative mating (line 7) is applied to guide the implicit information transfer among tasks. Algorithm 2 presents the procedure of the assortative mating. A random mating probability, called rmp, is defined to control this procedure. If a random number is less than rmp, (4) will be used to update the velocity. Otherwise, the velocity will be updated using (2). The vertical cultural transmission (line 9) is used to assign the skill factor for each generated individual in the population during the search process. In vertical cultural transmission, the generated individual imitates one task randomly from its previous individuals, thereby realizing exchanging solutions across the tasks. With vertical cultural transmission, the tasks in the multitasking system can find better solutions through continuously achieving promising solutions from other tasks. More details about MFPSO can be found in[28]

$$\begin{aligned} V_{\text{id}} &= \omega * V_{\text{id}} + c_1 * r_1 * (\text{ peest }_{\text{id}} - X_{\text{id}}) \\ &+ c_2 * r_2 * (\text{ gest }_d - X_{\text{id}}) + c_3 * r_3 * (\text{ best }'_d - X_{\text{id}}) \end{aligned} \tag{4}$$

Algorithm 1: Assortative Mating

Input: c_1, c_2, c_3, three acceleration constants; w, inertia weight; rmp, random mating probability.

Output: The velocity of each particle.

Generate a random number rand between 0 and 1.

if rand $< rmp$ then

Update the velocity of particle using (4).

end

else

Update the velocity of particle using (2).

end

where gbest$'_d$ denotes the global best position so far which has a different skill factor from particle i. In other words, g best$_d$ comes from other tasks.

4 Implementation details

4.1 Comparing with released source codes

No related source codes are available to my work. So here are the details of my own work, creative additions, noticeable improvements and/or new features.

4.2 Experimental environment setup

We use the platform named Matlab R2021b for model training. Ten gene expression datasets with thousands of features are used in the experimental analyses. They are open sources on https://ckzixf.github.io/dataset.html. The distribution of data is highly unbalanced. The classification on such datasets is a challenging task.

4.3 Main contributions

In this section, we propose a novel evolutionary multitasking-based FS method for the high-dimensional classification problems. Specifically, two important steps for successfully applying evolutionary multitasking are described first, followed by the details of two designed mechanisms. Finally, it presents the overall method of PSO-EMT.

4.3.1 Generating Two Related Tasks

When applying evolutionary multitasking for handling FS on the high-dimensional classification problems, the relevance of generated tasks is one of the essential factors affecting the classification performance. In a multitasking system, the tasks should satisfy a certain degree of commonality or complementarity in terms of the optimal solution or function landscape.

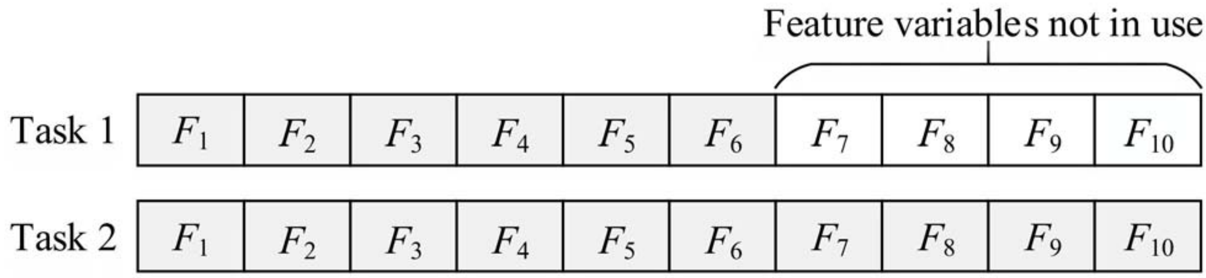

Figure 1: Example is used to illustrate the dimensions of feature space of Task 1 and Task 2.

In this study, two tasks with different search spaces based on different feature sets (e.g., a subset of the available features and the entire feature set) are developed. FS Task 1 is to select features from a subset of the original features, and FS Task 2 is to select features from all the original features.

Fig. 1 is an example to illustrate the dimensions of feature space, where the dimensions of Task 1 and Task 2 are six and ten, respectively. For Task 1, the first six feature variables of the individuals are used to indicate the solutions, while all feature variables are used in Task 2. Since Task 1 and Task 2 share some common features and have the same tasks of predicting the same set of class labels, these two FS tasks have a certain degree of commonality in terms of the optimal solution. According to the conditions of evolutionary multitasking mentioned above, FS from the subset of the available features and FS from the entire feature set are related tasks in multitasking. During the FS process, Task 1 focuses on more promising search space (i.e., important features), while Task 2 can ensure that all the original features have a chance to be considered. These two tasks can help each other by sharing information between different search spaces.

However, how to decide the promising feature subset (i.e., a subset includes important features) is not trivial. Since ReliefF [39] can better determine the nonlinear relationship between variables than many other feature ranking methods, this study uses ReliefF to measure the importance of features through assigning different weights to these features. The weights are determined based on the distinguishing ability of features to the close-range samples. The pseudocode of using ReliefF to calculate the weights of features is presented in Algorithm 3. In Algorithm 3, one sample R_i is first randomly selected from the training dataset (line 3), and it then finds h nearest neighbors of R_i from the same and different classes, which are denoted by $H_j(j=1,2,\ldots,h)$ and $M_j(c)(j=1,2,\ldots,h)$, respectively (from lines 4 to 5). Next, (5) is used to calculate the weight of each feature (line 7). Finally, a feature weight vector W is returned (line 10). Note that the larger the weight, the more important the feature

$$\begin{aligned} W_a^{u+1} =& W_a^u - \sum_{j=1}^{h} f\left(a, R_i, H_j\right)/(u*h) \\ &+ \sum_{c \notin s(R)} \left[\frac{p(c)}{1-p(s(R))} * \sum_{j=1}^{h} f\left(a, R_i, M_j(c)\right)\right]/(u*h) \end{aligned} \tag{5}$$

where W_a is the weight of feature a, u is the number of iterations, and h is the number of nearest neighbor samples. $s(R)$ is the class label of the sample R, and $p(c)$ is the prior probability of class label c. $f\left(a, R_i, R_j\right)$ means the difference between the values of the feature a for samples R_i and R_j.

When using ReliefF for FS, it suffers from a problem that ReliefF requires domain knowledge to set an appropriate threshold weight value to choose features from the ranked original features, which is very hard to determine and datasetdependent. The use of the largest distance to the extreme line to characterize knee point was first proposed by Das [40], which can detect the largest jump in the curve. Due to its effectiveness, the knee point has been successfully used as a demarcation point in many optimization problems. Inspired by this idea, we design a knee point selection scheme based on the weights of features to automatically determine the number of features selected from the original features ranked by ReliefF without losing much information about the class labels. In the knee point selection scheme, we sort features in a descending order according to their ReliefF weights first, and a curve (the blue curve) based on the weights is achieved. Second, a line is generated by connecting the point with the largest feature weight and the point with the smallest feature weight. Third, we calculate the distance between the line and each point on the curve. The point that has the largest distance (d) to the line is chosen as the knee point, and the features whose weights are larger than (larger weights, better features) that of the knee point are selected as the promising feature subset. It is noted that this scheme does not need to determine the threshold in advance, because it can automatically generate the promising feature subset by detecting the knee point for each dataset based on the characteristics of the data itself.

4.3.2 Applying Evolutionary Multitasking for FS

In step 2, there are two important parts. One is the knowledge transfer that is designed to share information between two related FS tasks. The other is the fitness function that is used to evaluate the quality of selected features in the multitasking framework.

1. Knowledge Transfer:

   Introducing the idea of knowledge transfer for FS on high-dimensional data implies that multiple related tasks need to be executed in parallel, and the search process involves information sharing among these related tasks.In this situation, we adopt a multipopulation framework to transfer knowledge between Task 1 and Task 2. During the search process, a skill factor is used to allocate the particles for each FS task. In other words, if a particle whose skill factor value is 1 , the particle is assigned to solve Task 1; otherwise, it is used to address Task 2. Vertical cultural transmission is another important operator in PSOEMT, which can change the value of skill factor to implement transfer solutions between Task 1 and Task 2. The way of transferring knowledge plays an important role in knowledge transfer, which has a great influence on the quality of the final selected feature subset. The crossover is an effective information-sharing operator, which can occur between the different or same subpopulations.

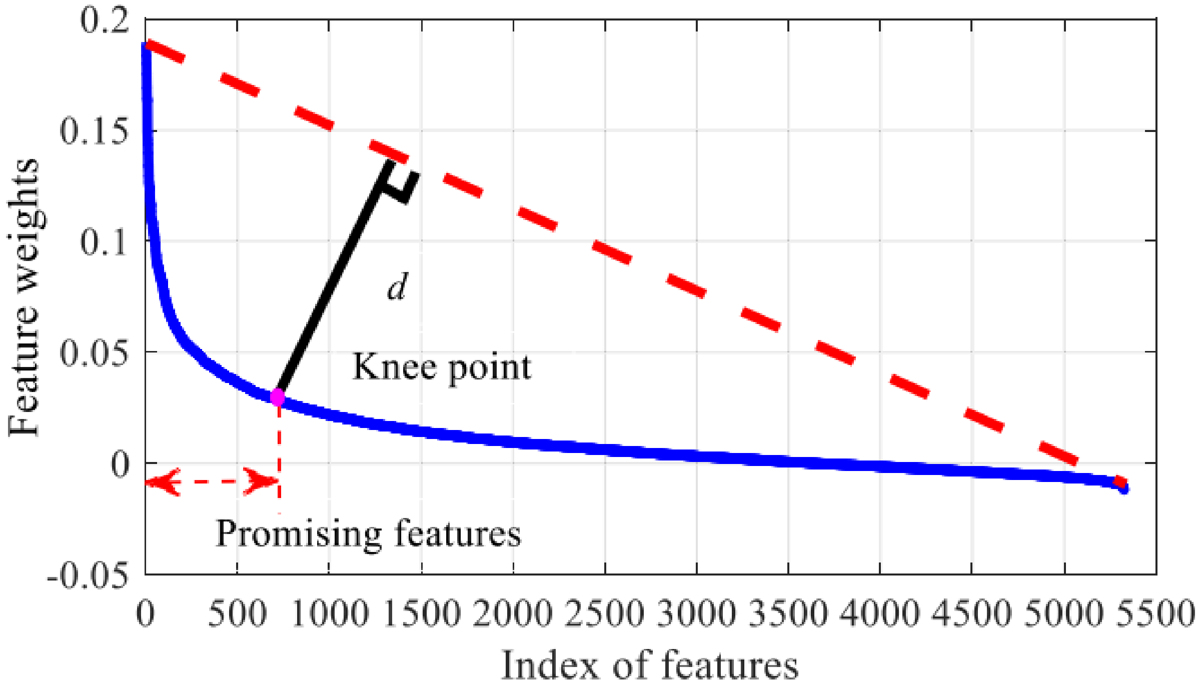

Figure 2: Example is used to select promising features based on the knee point

In this study, the crossover on the particle's velocity called assortative mating is applied to transfer knowledge between Task 1 and Task 2. Sharing knowledge by assortative mating not only can enhance the diversity for each subpopulation (task) but also can provide assistance to find better feature subsets. During the evolutionary process, a random mating probability (rmp) is defined to control when to transfer knowledge from another task. At each generation, if a random number rand is less than or equal to rmp, (4) is adopted to update particle's velocity; otherwise, (2) is applied to update particle's velocity.

2. Fitness Function:

In this study, the k-nearest neighbor (KNN) algorithm [32] is adopted as an evaluator to assess the selected subset. Equation (6) represents the fitness function that considers both the number of selected features and the classification accuracy of using the subset for classification

$$\text{fitness } = \alpha * \gamma_R(D) + (1-\alpha) * \frac{|S|}{|N|} \tag{6}$$

where $\gamma_R(D)$ denotes the classification error rate of the learning algorithm, $|S|$ represents the number of selected features, and $|N|$ indicates the total number of available features in the dataset. α is a parameter to reflect the role of the classification error rate and the number of selected feature subsets, where $\alpha \in [0, 1]$ according to[30]. In the multitasking system, since Task 1 is to select features from a subset of the available features and Task 2 is to select features from the entire set of the available features, the number of features in Task 1 is much smaller than that in Task 2. In order to ensure that the classification accuracy is always more important than the number of selected features in Task 1, we set $\alpha = 0.999999$ in this study. For Task 2 , we set $\alpha = 0.9$ to increase the impact of the number of selected features on subset evaluation, which can effectively speed up the removal of redundant features from the entire feature set. To handle unbalanced data in the FS process, a balanced accuracy[31] is used to determine the accuracy of the learning algorithm. Equation (7) shows the classification error rate

$$\gamma_R(D) = 1 - \frac{1}{c} * \sum_{i=1}^{c} TPR_i \tag{7}$$

where c denotes the number of classes in a classification problem, and TPR_i represents the proportion of correctly identified instances in class i. Since the balanced accuracy is no bias to each class in the classification problem, the weight for each class is set to $1/c$.

3. Final Selected Features:

It is noted that the final goal in this article is to obtain a feature subset that can achieve high classification accuracy in high-dimensional classification problems. However, two feature subsets will be produced when applying evolutionary multitasking in step 2, which are from Task 1 and Task 2, respectively. We will choose the selected features from Task 1 as the final feature subset to return, and the reasons are as follows. During the search process, Task 1 focuses on the promising areas (i.e., the features in Task 1 are important about the class labels) and continuously obtains useful information from Task 2 to jump out of potential local optima. It is difficult that Task 2 finds a subset with high accuracy due to its huge search space and complex interactions between the large number of features. The feature subset obtained from Task 1 is potentially better than that from Task 2 .

4. Variable-Range Strategy:

When implementing PSO for FS, a particle represents a potential feature subset, and each position value is within a fixed range (i.e., $[0, 1]$), which indicates whether the corresponding feature should be reserved or not by using a user-defined threshold (e.g., 0.6). Since the importance of each feature in the dataset is different, using a fix range for each feature in Task 2 is difficult to find a good feature subset due to the huge search space. A variable-range strategy based on the importance of features is proposed, which aims to reduce the search space by adapting the search range of each feature. The proposed strategy can effectively ensure that the features with higher (lower) importance have a higher (lower) probability to be selected.

The search range of each dimension is determined based on two demarcation values (e.g., the weight of the knee point and point 0). The knee point was described in Section III-A. If the weights of the features are greater than that of the knee point, their search ranges of the corresponding dimensions are set to $[0, 1]$. If the feature weights are less than 0 , these features are less helpful for solving problems. However, these features may become useful when combined with other features due to interactions between features. Therefore, we limit the dimensions corresponding to these features in the search space to a smaller search space, that is, these features are less likely to be selected. In the variable-range strategy, the search ranges of these features are set to $[0, \delta]$, where δ denotes the upper bound value for the dimensions corresponding to these features. If the feature weights are between the weight of the knee point and 0 , the search ranges for the dimensions corresponding to these features are linearly reduced from $[0, 1]$ to $[0, \delta]$ based on the weights of features.

5. Subset Updating Mechanism:

To help PSO escape from potential local optima, we design a promising subset updating mechanism in Task 1. During the evolutionary process, if gbest does not change within the given iterations (m), we modify the feature set in Task 1 . We randomly select some features from the unselected features in the first step to replace the same number of features in Task 1 (i.e., keep the size unchanged). Equation (8) is used to determine the number of features updated. Since the promising feature subset may be updated during the search process, the combination of potentially complementary features in Task 1 will likely be enhanced to some extent.

$$\text{numChange } = \rho * \text{ numSelect} \tag{8}$$

where numChange and numSelect represent the number of features updated and the number of features in Task 1, respectively. The scale factor ρ is used to control the number of updates. A high ρ value provided by the user may result in a lot of useful information to be lost, whereas a low ρ value does not achieve the objective of jumping out of possible local optima. Therefore, a sensitive analysis is conducted in Section IV-C to determine the appropriate values for ρ and δ.

4.3.3 Proposed Method

In Fig. 3, two important steps are designed for successfully applying evolutionary multitasking for FS. The purpose of step 1 is to generate two related FS tasks that can help each other to form a multitasking system. The input of step 1 is the training data. The outputs are two related FS tasks, which are based on the promising feature subset and the entire feature set, respectively. In this step, the proposed knee point selection scheme is used to determine the promising feature subset. The purpose of step 2 is to select a feature subset through knowledge transfer between these two related tasks in step 1. The inputs of step 2 are two related FS tasks. The output is a feature subset, which has a higher discriminating ability about the class labels. In this step, the proposed variable-range strategy and the subset updating mechanism are applied to reduce the search space of PSO and maintain the diversity of the population, respectively. Note that these two FS tasks are solved simultaneously in step 2.

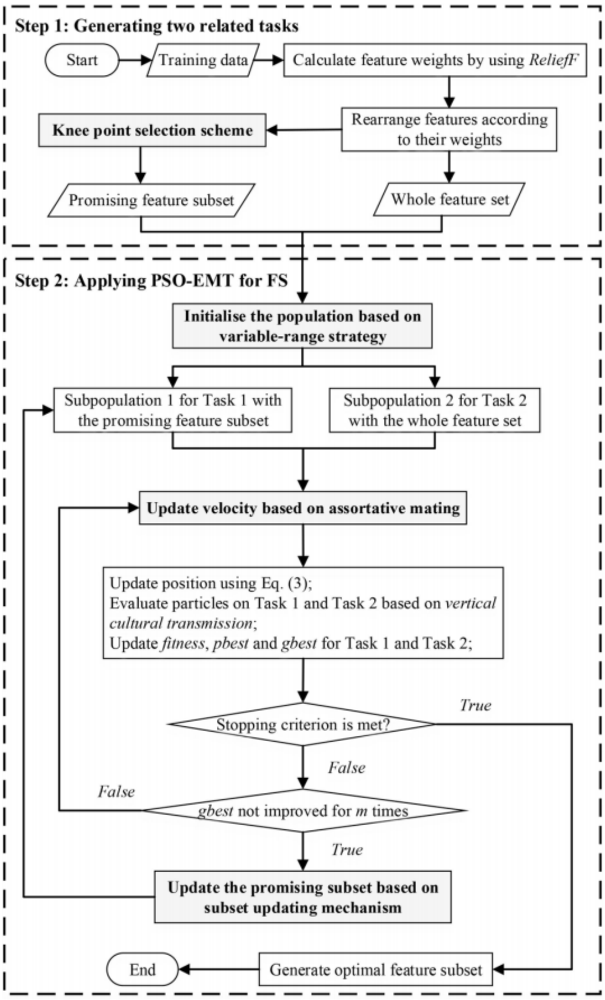

Figure 3: Framework of the PSO-EMT.

5 Results and analysis

5.1 Experimental Design

This section describes the details of the experimental setup, including the investigated datasets, the compared methods, and the parameter settings.

5.1.1 Datasets

Table 1: DATASETS

Dataset	#Features	#Instances	#Classes	%Smallest class	%Largest class
Leukemia 1	5,327	72	3	13	53
DLBCL	5,469	77	2	25	75
9Tumor	5,726	60	9	3	15

Ten gene expression datasets with thousands of features are used in the experimental analyses. They are open sources on https://ckzixf.github.io/dataset.html. Tabl I shows the key characteristics of these datasets. The distribution of data is highly unbalanced. The classification on such datasets is a challenging task.

5.1.2 Parameter setting

Table 2: PARAMETER SETTING

Parameters	Setting
Population Size	100
Maximum iterations	100
$c_1 = c_2 = c_3$	1.49445
Threshold for selected feature	0.6
ω	$0.9 - 0.5$*(iter/max_iter)
Communication topology	Fully connected (PSO)
The upper bound value (δ)	0.7
Scale factor (ρ)	0.05
Random mating probability (rmp)	0.6
Max iterations for subset updating (m)	10

Table II shows the parameter settings for the experiments. Since the number of features in the ten classification datasets surpasses 5000 , the population size is set as 100 to avoid high computational cost and to maintain PSO efficiency during the FS process. The maximum number of iterations is set to 100 . The threshold for selecting features is set to 0.6 so that the methods start with a small number of features[32]. Several different values for the upper bound value δ (i.e., from 0.6 to 1.0) and the scale factor ρ (i.e., from 0.01 to 0.10) are conducted to determine the appropriate values. The results of $\delta = 0.7$ and $\rho = 0.05$ are better than the other values. Therefore, δ and ρ are set to 0.7 and 0.05, respectively. The fully connected topology is used for particle communication in the population.

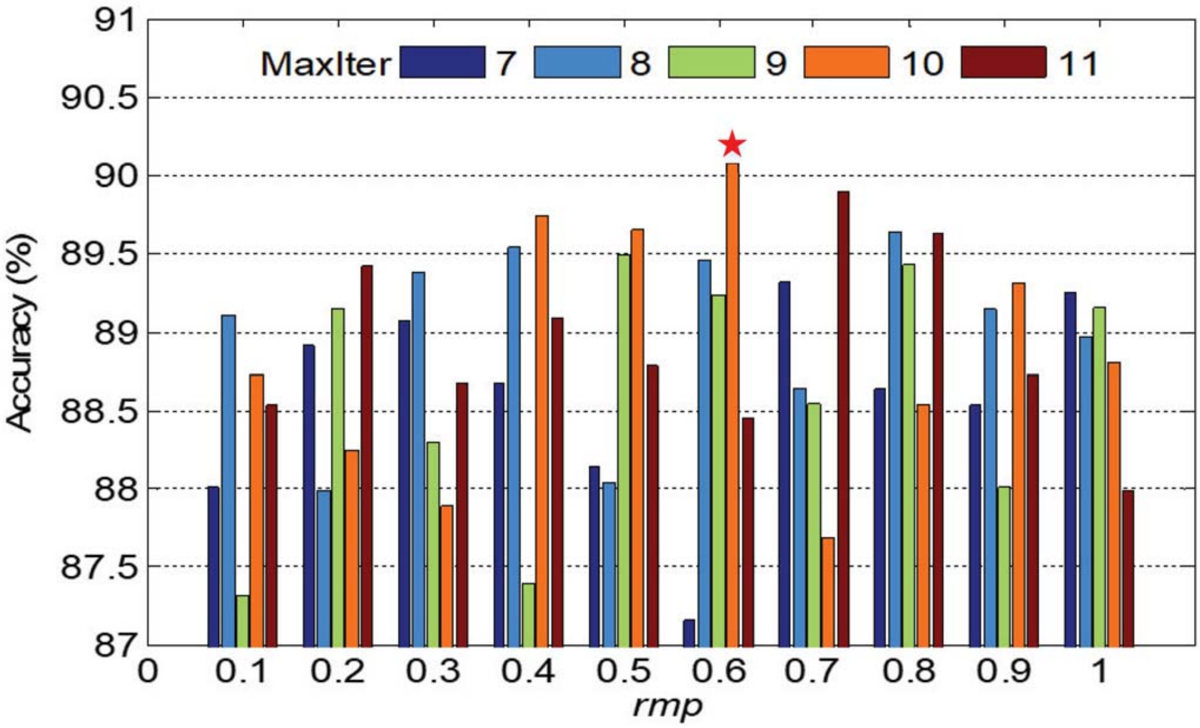

Figure 4: Average results of 50 combinations of two parameters (random mating probability and maximum iterations to update the promising feature subset).

Two parameters, which are random mating probability ((rmp) and max iterations for subset updating (m), are set according to the sensitive analysis. The analyses are conducted based on the Leukemia 2 dataset because it has a medium number of features compared with other classification problems. PSOEMT is executed with ten different values (i.e., from 0.1 to 1) for rmp, and five different values (i.e., from 7 to 11) for m. The number of combinations with these two parameters is 50 . In this experiment, each combination is carried out 30 times independently. Fig. 4 shows the average classification accuracy for each combination and the best result is marked with a five-pointed star ($\star$). As can be seen from Fig. 4, $\{rmp$ of $0.6, m$ of $10\}$ is the best combination. Therefore, this study uses the $\{rmp$ of $0.6, m$ of $10\}$ as the parameter settings in PSO-EMT for all the examined datasets.

The k-nearest neighbor (KNN)[33] is a commonly used classification method due to its effectiveness and efficiency. In this study, KNN is used to evaluate the selected feature subsets. According to previous studies[17],[34], the number of nearest neighbors is set to 1 in this study to maintain KNN efficiency and avoid noisy instances.

5.2 Results and analysis

In this section, two experiments are conducted to illustrate the effectiveness and efficiency of the proposed approach. The purpose of the first comparison is to verify whether the evolutionary multitasking-based FS method outperforms other state-of-the-art PSO-based FS approaches. The purpose of the second comparison is to verify whether the reproduced PSO-EMT method is better than the original PSO-EMT method.

Table 3: Average Test Results

Dataset	Method	Best	Mean±Std	Size	Time(s)
Leukemia1	FULL	79.48		5327.00	
	PSO	87.36	80.60±2.55	2615.50	41.20
	CSO	95.89	90.79±2.88	170.12	247.29
	AMSO	97.64	94.01±1.58	**51.49**	6.80
	VLPSO	**97.92**	93.31±2.34	54.70	**6.09**
	PSO-EMT(original)	95.71	91.11±2.79	198.40	9.28
	PSO-EMT(reproduced)	97.321	**94.594±9.2797**	67.2533	25.9758
DLBCL	FULL	82.33		5469.00	
	PSO	86.33	83.67±1.52	2681.00	47.59
	CSO	100.00	**94.60±3.26**	**30.08**	389.67
	AMSO	96.67	94.10±1.95	50.56	8.34
	VLPSO	93.33	86.51±2.88	48.14	7.18
	PSO-EMT(original)	**100.00**	93.76±2.80	83.55	**7.02**
	PSO-EMT(reproduced)	97.5	94.395±8.5093	84.2367	25.869
9Tumor	FULL	35.94		5726.00	
	PSO	45.00	42.72±1.42	2811.90	39.18
	CSO	**68.33**	**59.78±3.55**	220.34	370.40
	AMSO	56.67	50.11±3.61	52.16	**5.52**
	VLPSO	61.67	54.94±4.80	**47.05**	5.65
	PSO-EMT(original)	66.67	58.00±4.02	263.09	8.09
	PSO-EMT(reproduced)	67.5	59.632±18.783	154.0633	23.4274

5.2.1 Performance of original PSO-EMT

We first compare the original PSO-EMT algorithm with the reproduced PSO-EMT algorithm, and then compare the original PSO-EMT algorithm with other PSO algorithms.

In Leuk data set, the feature number, accuracy and time are better than two algorithms, and equal to one algorithm

In DLBCL data set, it is better than two algorithms in terms of feature number, but it is better than four algorithms in terms of accuracy, and it is optimal in terms of time. In general, it is better than three algorithms and equal to one algorithm

In the 9Tumor data set, it is better than the two algorithms in terms of feature number, but better than the four algorithms in terms of accuracy, better than the two methods in terms of time, and better than the four algorithms in general

Therefore, the experimental results show that on most data sets, the PSO-EMT method can obtain better classification accuracy with fewer features than the existing FS method. This is because the variable range strategy and the inflection point selection strategy are used in the training of search space control strategy, PSO-EMT method in most cases requires less computing time than other methods. In addition, the subset update strategy can maintain the diversity of the population in the search process, help the particle swarm optimization algorithm avoid the local optimal in the FS process, and explore the global optimal solution, so as to improve the classification accuracy

5.2.2 Performance of reproduced PSO-EMT

We compare the original PSO-EMT algorithm with the reproduced PSO-EMT algorithm.

Comparing the original PSO-EMT algorithm with the reproduced PSO-EMT algorithm, the average classification accuracy of the reproduced algorithm is improved in all data sets, and the optimal accuracy is higher in two of the three data sets than the original PSO-EMT algorithm.

The average classification accuracy of the reproduced PSO-EMT algorithm is increased by 2.44% on average;The optimal classification accuracy was increased by 0.14% on average;In dimension reduction, the number of original features was reduced by 35.57% on average.

It shows that the reproduced PSO-EMT algorithm can also remove redundant or irrelevant features and improve the classification accuracy. The results are similar to the original PSO-EMT algorithm, but due to the randomness of the evolutionary algorithm, the value of the results will fluctuate slightly.

In terms of running time, I think it is due to the different machine configuration or running environment that the time becomes longer. Because the same algorithm will also have very different results under different machine conditions, the running time of the original PSO

6 Conclusion and future work

The objective of this study is to use an effective comprehensive method to classify high-dimensional features. A feature selection method based on particle swarm optimization and multi-task optimization is designed to effectively select feature subsets.

Experimental results show that on most data sets, both the original PSO-EMT method and the reproduced

PSO-EMT method can obtain better classification accuracy with fewer features than the existing FS method.

PSO-EMT requires less computing time than other methods in most cases because of the variable range strategy and the inflection point selection strategy used in the training.In addition, the subset updating mechanism can keep the diversity of the population in the search process, and help the particle swarm optimization algorithm avoid the local optimal in the FS process and explore the global optimal solution.

For the follow-up research, this paper can also be improved in the feature correlation calculation, classification prediction and other aspects

References

[1] YU Z, ZHANG Y, YOU J, et al. Adaptive semi-supervised classifier ensemble for high dimensional data classification[J]. IEEE transactions on cybernetics, 2017, 49(2): 366-379.

[2] WANG C, HU Q, WANG X, et al. Feature selection based on neighborhood discrimination index[J]. IEEE transactions on neural networks and learning systems, 2017, 29(7): 2986-2999.

[3] GONZALEZ-VIDAL A, JIMENEZ F, GOMEZ-SKARMETA A F. A methodology for energy multivariate time series forecasting in smart buildings based on feature selection[J]. Energy and Buildings, 2019, 196: 71-82.

[4] ZHANG F, MEI Y, NGUYEN S, et al. Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling[J]. ieee transactions on cybernetics, 2020, 51(4): 1797-1811.

[5] CHEN K, ZHOU F, XUE B. Particle swarm optimization for feature selection with adaptive mechanism and new updating strategy[C]//Australasian Joint Conference on Artificial Intelligence. 2018: 419-431.

[6] POURPANAH F, SHI Y, LIM C P, et al. Feature selection based on brain storm optimization for data classification[J]. Applied Soft Computing, 2019, 80: 761-775.

[7] BRANKOVIC A, FALSONE A, PRANDINI M, et al. A feature selection and classification algorithm based on randomized extraction of model populations[J]. IEEE transactions on cybernetics, 2017, 48(4): 1151-1162.

[8] DASH M, LIU H. Consistency-based search in feature selection[J]. Artificial intelligence, 2003, 151(1-2): 155-176.

[9] XUE B, ZHANG M, BROWNE W N, et al. A survey on evolutionary computation approaches to feature selection[J]. IEEE Transactions on Evolutionary Computation, 2015, 20(4): 606-626.

[10] MAFARJA M, ALJARAH I, FARIS H, et al. Binary grasshopper optimisation algorithm approaches for feature selection problems[J]. Expert Systems with Applications, 2019, 117: 267-286.

[11] MORADI P, GHOLAMPOUR M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy[J]. Applied Soft Computing, 2016, 43: 117-130.

[12] HUANG Z, YANG C, ZHOU X, et al. A hybrid feature selection method based on binary state transition algorithm and ReliefF[J]. IEEE journal of biomedical and health informatics, 2018, 23(5): 1888-1898.

[13] MAFARJA M M, MIRJALILI S. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection[J]. Soft Computing, 2019, 23(15): 6249-6265.

[14] MAFARJA M M, MIRJALILI S. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection[J]. Soft Computing, 2019, 23(15): 6249-6265.

[15] SOLORIO-FERNÁNDEZ S, CARRASCO-OCHOA J A, MARTÍNEZ-TRINIDAD J F. A new hybrid filter–wrapper feature selection method for clustering based on ranking[J]. Neurocomputing, 2016, 214: 866-880.

[16] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//MHS'95. Proceedings of the sixth international symposium on micro machine and human science. 1995: 39-43.

[17] TRAN B, XUE B, ZHANG M. A new representation in PSO for discretization-based feature selection [J]. IEEE Transactions on Cybernetics, 2017, 48(6): 1733-1746.

[18] NGUYEN B H, XUE B, ANDREAE P, et al. A new binary particle swarm optimization approach: momentum and dynamic balance between exploration and exploitation[J]. IEEE transactions on cybernetics, 2019, 51(2): 589-603.

[19] LIU H, CAI Z, WANG Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization[J]. Applied Soft Computing, 2010, 10(2): 629-640.

[20] HUANG X, CHI Y, ZHOU Y. Feature selection of high dimensional data by adaptive potential particle swarm optimization[C]//2019 IEEE Congress on Evolutionary Computation (CEC). 2019: 1052-1059.

[21] ZHANG Y, LI H G, WANG Q, et al. A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection[J]. Applied Intelligence, 2019, 49(8): 2889-2898.

[22] GUPTA A, ONG Y S, FENG L, et al. Multiobjective multifactorial optimization in evolutionary multitasking[J]. IEEE transactions on cybernetics, 2016, 47(7): 1652-1665.

[23] GUPTA A, ONG Y S, FENG L. Insights on transfer optimization: Because experience is the best teacher [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 2(1): 51-64.

[24] ZHANG F, MEI Y, NGUYEN S, et al. A preliminary approach to evolutionary multitasking for dynamic flexible job shop scheduling via genetic programming[C]//Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. 2020: 107-108.

[25] GUPTA A, ONG Y S, FENG L. Multifactorial evolution: toward evolutionary multitasking[J]. IEEE Transactions on Evolutionary Computation, 2015, 20(3): 343-357.

[26] THANH P D, BINH H T T, TRUNG T B. An efficient strategy for using multifactorial optimization to solve the clustered shortest path tree problem[J]. Applied Intelligence, 2020, 50(4): 1233-1258.

[27] BALI K K, ONG Y S, GUPTA A, et al. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II[J]. IEEE Transactions on Evolutionary Computation, 2019, 24(1): 69-83.

[28] FENG L, ZHOU W, ZHOU L, et al. An empirical study of multifactorial PSO and multifactorial DE[C]//2017 IEEE Congress on Evolutionary Computation (CEC). 2017: 921-928.

[29] CHEN K, ZHOU F, LIU A. Chaotic dynamic weight particle swarm optimization for numerical function optimization[J]. Knowledge-Based Systems, 2018, 139: 23-40.

[30] XUE B, ZHANG M, BROWNE W N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms[J]. Applied soft computing, 2014, 18: 261-276.

[31] PATTERSON G, ZHANG M. Fitness functions in genetic programming for classification with unbalanced data[C]//Australasian Joint Conference on Artificial Intelligence. 2007: 769-775.

[32] AZEVEDO G L, CAVALCANTI G D, CARVALHO FILHO E C. An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting[C]//2007 IEEE Congress on Evolutionary Computation. 2007: 3577-3584.

[33] CHEN K, ZHOU F Y, YUAN X F. Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection[J]. Expert Systems with Applications, 2019, 128: 140-156.

[34] CHUANG L Y, YANG C H, LI J C. Chaotic maps based on binary particle swarm optimization for feature selection[J]. Applied Soft Computing, 2011, 11(1): 239-248.