Improved PSO for Feature Selection on High-Dimensional Datasets

Binh Tran, Bing Xue, and Mengjie Zhang

Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{binh.tran,bing.xue,mengjie.zhang}@ecs.vuw.ac.nz

Abstract. Classification on high-dimensional (i.e. thousands of dimensions) data typically requires feature selection (FS) as a pre-processing step to reduce the dimensionality. However, FS is a challenging task even on datasets with hundreds of features. This paper proposes a new particle swarm optimisation (PSO) based FS approach to classification problems with thousands or tens of thousands of features. The proposed algorithm is examined and compared with three other PSO based methods on five high-dimensional problems of varying difficulty. The results show that the proposed algorithm can successfully select a much smaller number of features and significantly increase the classification accuracy over using all features. The proposed algorithm outperforms the other three PSO methods in terms of both the classification performance and the number of features. Meanwhile, the proposed algorithm is computationally more efficient than the other three PSO methods because it selects a smaller number of features and employs a new fitness evaluation strategy.

Keywords: Particle swarm optimisation, Feature selection, Classification, High-dimensional data.

1 Introduction

Learning from examples is a successful approach in machine learning and data mining. Classification is a typical task of learning from training examples to predict the class labels of unseen examples/instances based on given attributes or features. Many classification algorithms have been successfully applied to automatically learn classifiers in a variety of problems, such as image classification, text categorisation and disease classification. Recently, there are more and more classification datasets with hundreds or even thousands features, which causes the "curse of dimensionality". This makes the classifier learning process become difficult because not all features are relevant to the class labels and often contain redundant information. Such data typically needs feature selection (FS) to remove irrelevant and redundant features [9].

Existing FS methods can be classified into wrapper approaches and filter approaches depending on whether a classification algorithm is used to evaluate the goodness of the feature subsets [5]. The criteria used in wrapper methods include the classification performance of a predefined classification algorithm using only the selected feature subsets. On the other hand, filter methods rely on various measures of the general characteristics of the training data, such as distance, information, dependence and consistency

G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, pp. 503–515, 2014.

measures [5] to evaluate the classification capability of the feature subsets. Filter methods are argued to be more general to different classification algorithms than wrapper methods. Wrapper approaches usually obtain feature subsets with higher classification accuracy than filter methods [14] because they directly use the classification performance to guide the search. However, wrapper approaches are computationally expensive because each evaluation involves a training process of the classification algorithm. This work focus mainly on developing a new wrapper FS algorithm.

An optimal feature subset is the smallest subset, which maximises the classification accuracy. However, finding the optimal feature subset is an NP-hard problem [14]. The size of the search space grows exponentially with the number of features in the dataset. Therefore, it is necessary to have an efficient global search technique to tackle FS problems. Evolutionary Computation (EC) techniques are well-known for their global search potential. Particle swarm optimisation (PSO) [13,22] is a relatively recent EC technique that has been successfully applied in many areas such as function optimisation [22] and feature selection [15,29,27,30,28]. Comparing to other EC techniques, PSO has some advantages such as simplicity, fewer parameters, lower computational cost, and fast convergence [8].

In PSO, a swarm of candidate solutions are encoded as particles. During the searching process, each particle remembers the best solution it obtained so far, i.e. the personal best called *pbest*. By sharing *pbest* with neighbours, each particle knows the best solution that the whole population has found so far, i.e. the global best called *gbest*. PSO searches for the optimal solutions based on the information from *pbest* and *gbest*. However, if *gbest* is near a local optimum, there is a high probability that the swarm will be stuck in this area, especially in high-dimensional/large-scale problems, e.g. FS on gene expression data, which has thousands or even more than ten thousands features.

To overcome this limitation, Chuang et al [4] proposed an improved PSO algorithm (PSO-RG) in which *gbest* will be restarted whenever it is not improved in a number of iterations. The proposed PSO achieved better performance than standard PSO. Resetting $gbest$ could avoid being stuck in local optima by encouraging the exploration of the search, but it may limit the algorithm further exploit the surroundings of the already found good solutions, i.e. $gbest$. Therefore, a new PSO algorithm is still needed to better solve high-dimensional feature selection problems on gene expression data.

1.1 Goals

The overall goal of this paper is to develop a new PSO approach to feature selection on high-dimensional gene expression data to significantly reduce the number of features and increase the classification performance over using all features. To achieve this goal, the $gbest$ reset mechanism is used to encourage the global search (exploration) and a new local search strategy is proposed to facilitate the exploitation of the algorithm to further improve the performance. The local search is also designed to utilise the characteristics of a simple classification algorithm, k-Nearest-Neighbour (KNN), to avoid heavy computational cost. The proposed algorithm is examined and compared with standard PSO, PSO only using the proposed local search (PSO-LS), and PSO-RG only using the $gbest$ reset mechanism on five gene datasets with more than ten thousands of features. Specifically, we will investigate:

- whether the proposed algorithm can reduce the number of features and achieve similar or better classification performance than using all features,
- whether the proposed algorithm can outperform than standard PSO, PSO-LS and PSO-RG in terms of the number of features and the classification performance, and
- whether the proposed algorithm can be more efficient than standard PSO, PSO-LS and PSO-RG.

2 Background

2.1 Particle Swarm Optimisation (PSO)

PSO is an EC technique developed by Kennedy and Eberhart [13], which is inspired by social behaviours found in birds flocking or fish schooling. In PSO, a swarm consists of many individuals called particles communicating through iterations to search for optimal solutions when moving in the search space.

In PSO, each particle has a position and a velocity. The position is a candidate solution of the problem and is usually an *n*-dimension vector of numerical values. Velocity also has the same structure as position, which represents the speed and direction that the particle should move in the next iteration. In each iteration, the velocity of a particle is updated based on the personal best (*pbest*) which is the best position it has been explored so far and the global best (*gbest*) which is the best position it has been communicated from other particles. Formulae (1) and (2) are used to update the velocity and position of each particle.

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{1i} * (p_{id} - x_{id}^{t}) + c_2 * r_{2i} * (p_{gd} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where v_{id}^{t} and x_{id}^{t} are velocity and position of particle i in dimension d at time t, respectively. p_{gd} and g_{gd} are *pbest* and *gbest* positions in dimension d. c_1 and c_2 are acceleration constants, and r_1 and r_2 are random values. w is the inertia weight used to control the impact of the last velocity to the current velocity. The velocity values are usually limited by a predefined maximum velocity, v_{max} to the range $[-v_{max}, v_{max}]$.

2.2 Related Works on Feature Selection

Traditional Methods for Feature Selection. Two typical wrapper FS methods are sequential forward selection (SFS)[25] and sequential backward selection (SBS) [17], which employs a greedy search method. SFS (SBS) starts with an empty (full) feature subset, then gradually adds (removes) features until the classification accuracy is not improved. However, both methods suffer from the so-called "nesting effect" because a feature which is selected or removed cannot be removed or selected in later stage. As a compromise between these two approaches, "plus-l-take-away-r" [24] applies SFS l times and then SBS r times. This strategy can avoid nesting effect, but it is hard to determine appropriate values for l and r. To avoid this, Pudil et al [21] introduced two corresponding methods: sequential backward floating selection(SBFS) and sequential forward floating selection (SFFS). These floating search methods are claimed to be better than the static sequential methods, but they are still facing the problem of stagnation in local optima.

EC Techniques for Feature Selection. Many EC techniques have been applied to FS problems such as Genetic algorithms (GAs) [3], Genetic programming (GP) [19], and Ant colony optimisation (ACO)[7]. Among these, GA is probably the first popular EC technique that has been applied in FS. Guided by Darwinian evolution principles, GAs start with a population of candidate solutions, represented as chromosomes, and evolved better solutions by using genetic operators like crossover, mutation. Many GA based FS algorithms have been proposed either in filter or wrapper approaches. In the former, feature subsets are evaluated by using inconsistency rates [16], or fuzzy sets [3]; while in the latter, different approaches proposed with different classification algorithms for fitness evaluation, such as ID3 [2], and artificial neural network [20]. GA based hybrid FS algorithms that combine both filter and wrapper approaches have also been proposed to improve the performance [10].

Using the same principles, however, instead of evolving bit strings, GP evolves computer programs to generate solutions. Each program is a tree consisting of internal nodes which are usually arithmetic operators, and leaf nodes which are constants or variables. When using GP for FS, the variables are chosen from the original features. Selected features are the ones used as leaf nodes of a GP tree. GP has been used in both filter FS methods [19] and wrapper FS methods [6]. ACO is another EC technique that stands in the same umbrella to PSO, swarm intelligence. ACO is inspired by the special communication system using pheromone between real ants about favorable paths to food. The shortest path will be the one that has most pheromone. When using ACO for FS, each feature is considered as a node, and paths between nodes represent the choices for next features. Many ACO algorithms are used for both filter [11,18] and wrapper [12] feature selection. PSO has recently gained more attention in addressing FS tasks, but most of the existing PSO based FS algorithms focus mainly on problems with a few hundreds of features [4,26].

3 Proposed Approach

In this section, a new PSO approach (named PSO-LSRG) is proposed for wrapper feature selection, where a new local search method is applied to $pbest$ to exploit better solutions and a reset mechanism is applied to $gbest$ to avoid stagnation in local optima. These two techniques are combined to see whether they can help PSO balance between global search and local search to improve the performance.

3.1 Overall Algorithm

PSO-LSRG mainly follows the basic steps of standard PSO. A particle represents a feature subset, where the dimensionality is the total number of features in the dataset. Each particle is encoded by a string of floating numbers in [0,1]. A threshold θ is used to determine whether a feature is selected or not. If the position value is larger than the threshold, the corresponding feature is selected. Otherwise, the corresponding feature is not selected. The fitness function of PSO-LSRG is to maximise the classification accuracy of the selected features, which is shown by Formula (3).

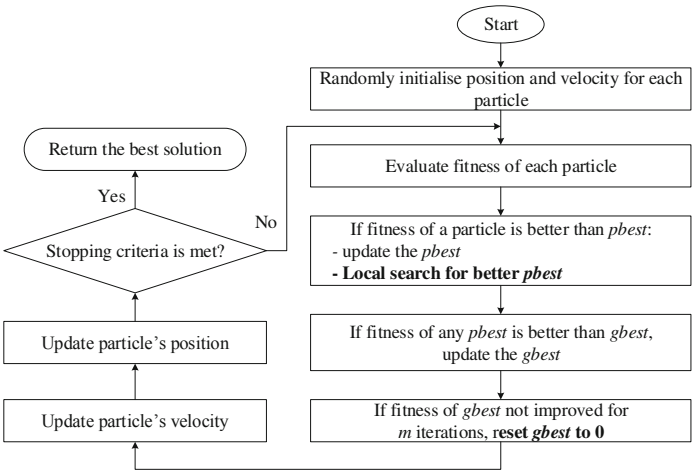

Fig. 1. Flowchart of the proposed algorithm

$$fitness = \frac{number\_of\_instances\_correctly\_classified}{Total\_number\_of\_instances} \tag{3}$$

Fig. 1 illustrates the overall steps of PSO-LSRG. The two techniques are highlighted in the figure and will be described in the remaining of this section.

3.2 Reset $gbest$

$gbest$ plays an important role in leading the search direction of all particles in the swarm. Resetting $gbest$ if it does not change for a number of iterations can help avoid premature convergence [4], which is a limitation of PSO. Following [4], the current $gbest$ is reset to a vector of all "0"s if it does not improve in a few iterations.

3.3 Local Search on $pbest$

The local search proposed in this algorithm is basically a loop in which a predefined number of dimensions in the *pbest* position will be flipped, i.e. a feature from being "selected" to "not selected" or from being "not selected" to "selected". To focus the search on the area surrounding current *pbest*, flipping is applied to a low percentage of the total dimensions and 2% is chosen in this work. After flipping, the new solution is evaluated. If it has better fitness, $pbest$ is updated. The search will stop after a predefined number of steps.

Fig. (2a) shows an example of the flipping procedure, where $pbest$ is converted to a binary string using the threshold θ, where "1" in the $pbest$ array means the corresponding feature is selected while "0" means the feature is not selected. Based on the randomly chosen flip dimensions positions, for example (1, 2, 5, 6, 9), the current $pbest$ can be flipped to obtain a new $pbest$ position (flipped $pbest$).

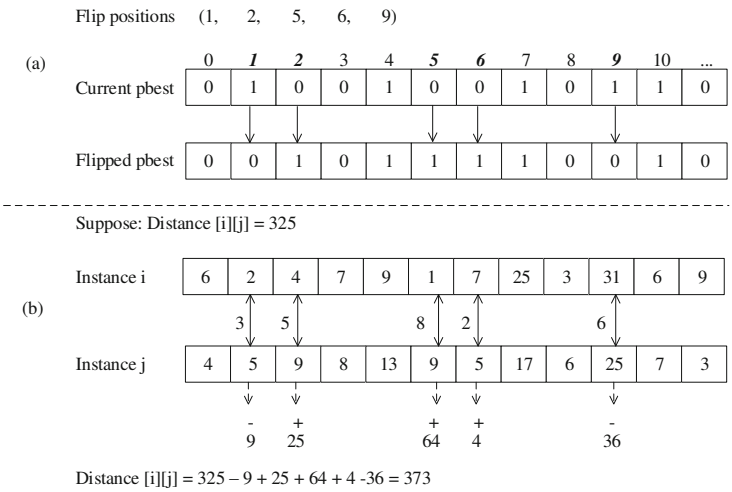

Fig. 2. An example of one local search step and re-calculating instances' distance

Local search usually brings more computation to the algorithm because of more fitness evaluations. Moreover, each evaluation in a wrapper FS method is usually expensive because it involves a training process of a classification algorithm. To avoid this problem, we proposed a new strategy to calculate the distances in KNN to find the nearest neighbours. This strategy utilises the characteristics of the local search and KNN. In each local search step, just a small percentage of the total dimensions (2%) will be changed and 98% of them in the flipped $pbest$ remain the same as in the original $pbest$. In standard KNN, to calculate the distance between two instances, their differences (e.g. the squared difference) in all dimensions need to be calculated, where 98% of the calculation is repeated because 98% of the dimensions in the flipped $pbest$ is the same as in the original $pbest$. Therefore, in the new evaluation strategy, only 2% of the dimensions are calculated. To achieve this, at the beginning of each local search run, all the cross distances between instances will be calculated regarding to the features selected in this given *pbest* and stored in a square matrix (distance$[i][j]$). Since this matrix is symmetric, with m instances in the dataset, the algorithm actually calculates $\frac{m(m+1)}{2}$ times. By using the distances stored in the matrix, it can speed up the computation of finding the nearest neighbours of a certain instance by calculating only 2% of the dimensions in each evaluation.

Fig. (2b) shows an example of distance re-calculation between two illustrated instances: $instance_i$ and $instance_j$, where the distance is determined by summing the squared difference of $instance_i$ and $instance_j$ in all dimensions. Suppose the distance between $instance_i$ and $instance_j$ regarding to the original $pbest$ ($distance[i][j]$) is 325, the new distance regarding to the flipped $pbest$ can be re-calculated as follows.

Table 1. Datasets

Data set	Number of Features	Number of Instances	Number of Classes
DLBCL	5469	77	2
9 Tumors	5726	60	9
Prostate Tumor	10509	102	2
11 Tumors	12533	174	11
Lung Cancer	12600	203	5

With 5 chosen flipped dimensions, it only need to subtract from 325 the square differences of the features' values that were selected (2 features: 1 and 9) and add those that were not selected (3 features: 2, 5 and 6). The new distance therefore is 373. Having all distances calculated, the algorithm can quickly find the new nearest neighbours for a given instance. As a result, the proposed strategy can save a significant amount of time.

4 Experimental Design

To examine the proposed approach, four different PSO algorithms are used for feature selection, which are the standard PSO (PSO), PSO with reset $gbest$ only (PSO-RG) [4], PSO with local search on $pbest$ only (PSO-LS) and PSO with both the reset $gbest$ strategy and the local search on $pbest$ (PSO-LSRG). Five datasets (Table 1) with a large number of features are chosen to test the performance of the algorithms, which are gene expressions profiles download from http://www.gems-system.org.

Since the datasets include a small number of instances, KNN (K=1) with leave one out cross validation (LOOCV) is used to calculate the classification accuracy, which is the same as in [4]. The acceleration coefficients are set as are $c_1 = c_2 = 2.0$ and inertia weight linearly decreases from 0.9 to 0.4 [23]. The swarm consists of 30 particles. The maximum number of iterations is 70 and fully connected communication topology is used here. Maximum velocity is 6.0. The threshold $\theta = 0.6$ is used to determine the selection of features. Whenever the local search is applied on a given *pbest*, it will try 100 times to find better *pbest*. In each time, 2% of the dimensions will be flipped to create a new candidate solution. Meanwhile, if *gbest* is not improved for three iterations, it is reset to all 0, which is the same as in [4] for comparison purposes. The experiment is conducted for 30 independent runs with different random seeds. A statistical significance test, pairwise Student's T-test, is performed between the classification performance of different algorithms, where the significance level is set as 0.05.

5 Results and Discussions

Table 2 show the experimental results of the four PSO algorithms: PSO, PSO-RG, PSO-LS, PSO-LSRG. In this table, "Ave-Size" means the average number of features selected by each method over the 30 runs. "Best", "Mean" and "StdDev" respectively are the best, the average and standard deviation of the classification accuracies returned by 1NN with LOOCV in the 30 independent runs. The "All" row shows the original number of features and its classification accuracy when using all features. The highest average accuracies and the smallest size of all methods in each dataset are the bold ones.

Table 2. Experimental Results

The more "-", the better PSO-RG, PSO-LS or PSOLSRG.

Dataset	Method	Ave-Size	Best	Mean±StdDev	T_{RG}	T_{LS}	T_{LSRG}
DLBCL	All	5469	87.01		–	–	–
	PSO	2625.70	98.70	97.70±1.01	–	–	–
	PSO-RG	1766.53	98.70	98.44±0.53		=	=
	PSO-LS	2094.70	98.70	98.40±0.56			–
	PSO-LSRG	**1690.13**	98.70	**98.66**±0.24			
9 Tumors	All	5726	53.33		–	–	–
	PSO	2808.20	78.33	72.72±2.78	–	–	–
	PSO-RG	2720.63	78.33	74.67±2.68		–	–
	PSO-LS	2139.10	86.67	80.72±2.13			=
	PSO-LSRG	**2114.57**	86.67	**81.39**±1.76			
Prostate Tumor	All	10509	76.47		–	–	–
	PSO	5143.20	91.18	88.53±1.77	–	–	–
	PSO-RG	2353.17	98.04	92.42±2.74		=	–
	PSO-LS	3825.17	95.10	92.09±1.06			–
	PSO-LSRG	**2148.47**	98.04	**94.94**±1.18			
11 Tumors	All	12533	84.48		–	–	–
	PSO	6138.33	92.53	90.92±0.90	–	–	–
	PSO-RG	5623.87	95.40	91.51±1.08		–	–
	PSO-LS	4671.07	95.40	**93.87**±1.03			=
	PSO-LSRG	**4293.63**	95.40	93.79±0.70			
Lung Cancer	All	12600	90.15		–	–	–
	PSO	6144.03	96.55	95.78±0.50	–	–	–
	PSO-RG	4792.83	97.54	96.40±0.61		–	–
	PSO-LS	4641.17	97.54	97.03±0.50			=
	PSO-LSRG	**3426.43**	98.03	**97.19**±0.43			

The "T_{RG}" column shows the results of T-tests between PSO-RG and other methods, where "+" ("–") means the corresponding method achieves significantly better (worse) classification performance than PSO-RG. "=" means they are similar. Similarly, "T_{LS}" or "T_{LSRG}" are the results of T-tests comparing the classification performance achieved by other methods and PSO-LS or PSO-LSRG, respectively.

5.1 The Standard PSO

As shown in Table 2, the standard PSO obtained feature subsets with higher classification performance and smaller size than all features on all the five datasets. Using feature subsets evolved by PSO, the 1NN classifier increases its average classification accuracy about 20% on the 9 Tumor dataset, 10% on the DLBCL and the Prostate Tumors, and 5% on the other two datasets. The number of features selected by PSO is about 50% of the original number of features on all the five datasets. The results show that PSO is a suitable tool for FS problems. For the rest of this work, we will consider PSO as a baseline to compare with other methods.

5.2 Effect of Reset $gbest$ Technique (PSO-RG)

From all the "–"symbols in T_{RG} column of Table 2, we can state that the classification accuracies of feature subsets selected by PSO-RG is significantly better than using all features and those of PSO on all the five datasets. Furthermore, the subset size of the PSO-RG solutions is also smaller than that of PSO. It is about half on two datasets, which are the 9 Tumor and the 11 Tumor datasets, and one-third on the DLBCL and

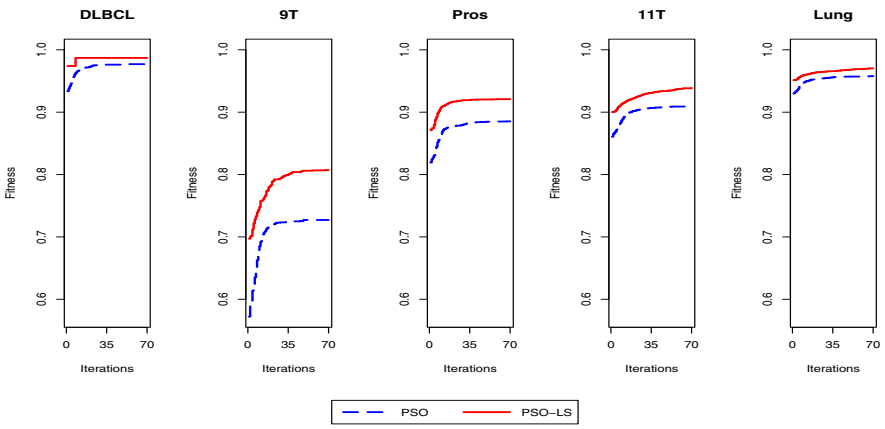

Fig. 3. Average *gbest* fitness of PSO and PSO-LS on five datasets

Lung Cancer datasets. On the Prostate Tumor dataset, this number even further reduces, which is less than a quarter of the original feature set size. This is possibly because setting *gbest* to 0 attracts all other particles moving toward this direction, resulting to smaller subsets. The results indicate that the reset $gbest$ technique is useful in directing particles to other promising areas when they seems to get stuck in a local optimum.

5.3 Effect of Local Search on $pbest$ (PSO-LS)

According to the results of "PSO-LS" in Table 2, the feature subsets evolved by PSO with local search on $pbest$ can achieve significantly higher classification performance than using all features and standard PSO. PSO-LS obtained significantly better classification performance than PSO-RG on three datasets and they are similar on the other two datasets, which are the DLBCL and the Prostate Tumor. The overall results show that the proposed local search technique on $pbest$ gives particles more chances to reach better positions in their local areas.

To have a better view of the local search effect on $pbest$, it is worth to observe how $gbest$ changes during the searching process. Fig. 3 contains five graphs for the five datasets, where each graph shows the average fitness value of *gbest* over the 30 runs in each iteration. Each graph has a dashed line and a solid line representing PSO and PSO-LS, respectively. The figure shows that just after the first iteration, the average value of $gbest$ of PSO-LS shows a significant improvement comparing to the standard PSO algorithm on all the five datasets. This indicates that the local search helps the population reach better solution regions and obtain feature subsets with higher classification performance.

Table 3. Average computation time (in minutes)

	PSO	PSO-RG	PSO-LS	PSO-LSRG
DLBCL	34.81	23.37	28.23	**18.99**
9 Tumors	23.64	19.79	18.85	**17.40**
Prostate Tumor	160.13	99.85	134.50	**83.77**
11 Tumors	452.14	361.68	352.26	**310.06**
Lung Cancer	558.17	435.01	446.47	**363.59**

5.4 Combination of Local Search and Reset $gbest$ in PSO (PSO-LSRG)

As can be seen from Table 2, PSO-LSRG selected the smallest number of features on all the five datasets, which is only around 20% of the original features on the Prostate Tumor dataset. Meanwhile, PSO-LSRG achieved the highest classification accuracies on four of the five datasets. The results of the significance T-tests in T_{LSRG} show that PSO-LSRG outperformed PSO-RG and PSO-LS on four and three of the five datasets, respectively. For the remaining datasets, they achieved similar results.

From Table 2, we can see an interesting pattern shown in the bold pair of T-test values in T_{LS} and T_{LSRG} on each dataset. The pattern of **"– ="** shows that for those datasets where PSO-LS outperformed PSO-RG, which are the 9 Tumors, 11 Tumors and Lung Cancer datasets, PSO-LSRG achieved similar classification performance to PSO-LS. On the other hand, the **"= –"** shows that for those datasets where PSO-LS only evolved similar results to PSO-RG, which are the DLBCL and Prostate Tumors, PSO-LSRG made an improvement. From this observation, we can conclude that the combination of reset $gbest$ and local search on $pbest$ can overcome the limitations of the two techniques and help PSO balance its exploration and exploitation abilities. Therefore, the feature subsets selected by PSO-LSRG that combines two techniques generally have a smaller size and equal or better classification performance than the case where only one technique is applied.

5.5 Computational Time

Table 3 shows the average CPU time used by each method in the 30 independent runs on the five datasets, where the numbers are expressed in minutes.

From Table 3, it can be seen that PSO-RG consumes less time than the standard PSO. Since all these PSO methods have the same settings in term of the number of particles and iterations, they have the same number of evaluations in one run. Therefore, the factor makes their computation time different is that the evolved feature subsets in the former are smaller than the latter. This again confirms the big influence of the feature subset size on the computation time in wrapper FS approaches. As a consequence, in PSO-LS and PSO-LSRG, although the local search part adds more running time to the standard PSO in every update of *pbest*, it does not make the total computation time of one run longer. By contrast, by reaching solutions with smaller subsets, it can significantly reduce the total computation time. Another important factor is the computation time saved by using the cross distance matrix of the instances to evaluate a new *pbest*, which was explained in Section 3.3. This new strategy successfully reduces the running time of KNN classifier. The quick evaluation time and selecting smaller subsets make the computation time of PSO-LSRG be the shortest in all the five different datasets.

5.6 Further Discussions

Note that in the experimental design, this paper uses the re-substitution estimator to evaluate the performance of the feature subsets, which is the same as in [4] and many other existing papers. The re-substitution estimator, in other words, means the whole dataset is used during the evolutionary feature selection process. There is no separated unseen data to test the generality of the selected features. There is a feature selection bias here, so we cannot claim that the selected features can be used for future unseen data for classification.

According to Ambroise et al [1], the feature selection bias effect can be reduced by using cross validation or bootstrap estimators in which a fraction of the original dataset are held out for testing the performance of the selected features. We will further investigate this in our future work.

6 Conclusions and Future Work

The goal of this paper was to develop a new PSO approach to feature selection on high-dimensional gene datasets with thousands of features. The goal has been successfully achieved by developing a new efficient local search on $pbest$ and applying a reset $gbest$ mechanism in PSO for feature selection, where KNN with LOOCV was used to evaluate the classification performance. The performance of the new PSO approach with both local search and reset $gbest$ (PSO-LSRG) is examined and compared with standard PSO, PSO with the proposed local search only (PSO-LS), and PSO with the reset $gbest$ mechanism only (PSO-RG). The experiments on five gene datasets of varying difficulty show that the feature subsets returned by PSO-LS are smaller and achieved better classification performance than those of PSO, and better or at least similar to those of PSO-RG. The change of *gbest* during searching processes also indicates that the proposed local search on *pbest* is an effective strategy for PSO to improve its search ability. The results of PSO-RG show that the reset $gbest$ technique helped particles divert their search to other promising regions when they seem to get stuck in a near local optima. However, the *gbest* reset mechanism might also prevent particles to better exploit their findings. Meanwhile, applying local search on *pbest* gives particles more chances to obtain better solutions. Therefore, the combination of these two techniques in PSO-LSRG further increased the performance. The results confirm that PSO-LSRG could overcome their limitations to achieve even better results than both PSO-LS and PSO-RG in terms of the classification performance and the number of features. Meanwhile, the proposed strategy for the fitness evaluation in local search has successfully saved the computation time for KNN, enabling PSO-LSRG being more efficient than the other three methods.

The proposed algorithm significantly reduced the size of the feature set, but it can be seen that the numbers are still large. Further reducing the number of features is still an important and challenging task. In future work, we intend to develop a new EC approach to further reduce the number of features and improve the classification performance without increasing the computational cost.

References

1. Ambroise, C., McLachlan, G.J.: Selection bias in gene extraction on the basis of microarray gene-expression data. Proceedings of the National Academy of Sciences 99(10), 6562–6566 (2002)
2. Bala, J., Huang, J., Vafaie, H., Dejong, K., Wechsler, H.: Hybrid learning using genetic algorithms and decision trees for pattern classification. In: The 14th International Joint Conference on Artificial Intelligence, vol. 1
3. Chakraborty, B.: Genetic algorithm with fuzzy fitness function for feature selection. In: IEEE International Symposium on Industrial Electronics (ISIE 2002), vol. 1, pp. 315–319 (2002)
4. Chuang, L.Y., Chang, H.W., Tu, C.J., Yang, C.H.: Improved binary pso for feature selection using gene expression data. Computational Biology and Chemistry 32(1), 29–38 (2008)
5. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1, 131–156 (1997)
6. Davis, R.A., Charlton, A.J., Oehlschlager, S., Wilson, J.C.: Novel feature selection method for genetic programming using metabolomic 1h NMR data. Chemometrics and Intelligent Laboratory Systems 81(1), 50–59 (2006)
7. Dorigo, M., Di Caro, G.: Ant colony optimization: a new meta-heuristic. In: IEEE Congress on Evolutionary Computation, vol. 2, pp. 1470–1477 (1999)
8. Engelbrecht, A.P.: Computational intelligence: an introduction, 2nd edn. Wiley (2007)
9. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
10. Huang, J., Cai, Y., Xu, X.: A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recognition Letters 28(13), 1825–1844 (2007)
11. Jensen, R., Shen, Q.: Finding rough set reducts with ant colony optimization. In: Proceedings of the 2003 UK Workshop on Computational Intelligence, pp. 15–22 (2003)
12. Kanan, H.R., Faez, K.: An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. Applied Mathematics and Computation 205(2), 716–725 (2008), Special Issue on Advanced Intelligent Computing Theory and Methodology in Applied Mathematics and Computation
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
14. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997), relevance
15. Lane, M.C., Xue, B., Liu, I., Zhang, M.: Gaussian based particle swarm optimisation and statistical clustering for feature selection. In: Blum, C., Ochoa, G. (eds.) EvoCOP 2014. LNCS, vol. 8600, pp. 133–144. Springer, Heidelberg (2014)
16. Lanzi, P.L.: Fast feature selection with genetic algorithms: a filter approach. In: IEEE International Conference on Evolutionary Computation, pp. 537–540 (1997)
17. Marill, T., Green, D.M.: On the effectiveness of receptors in recognition systems. IEEE Transactions on Information Theory 9(1), 11–17 (1963)
18. Ming, H.: A rough set based hybrid method to feature selection. In: International Symposium on Knowledge Acquisition and Modeling, KAM 2008, pp. 585–588 (December 2008)
19. Neshatian, K., Zhang, M.: Pareto front feature selection: Using genetic programming to explore feature space. In: The 11th Annual Conference on Genetic and Evolutionary Computation, GECCO 2009, pp. 1027–1034 (2009)
20. Oliveira, L., Sabourin, R., Bortolozzi, F., Suen, C.: Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In: 16th International Conference on Pattern Recognition (ICPR 2002), vol. 1, pp. 568–571 (2002)

21. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature selection. Pattern Recogn. Lett. 15(11), 1119–1125 (1994)
22. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation (CEC 1998), pp. 69–73 (1998)
23. Shi, Y., Eberhart, R.: Empirical study of particle swarm optimization. In: IEEE Congress on Evolutionary Computation (CEC 1999), vol. 3, pp. 1945–1950 (1999)
24. Stearns, S.D.: On selecting features for pattern classifiers. In: Proceedings of the 3rd International Conference on Pattern Recognition (ICPR 1976), Coronado, CA, pp. 71–75 (1976)
25. Whitney, A.: A direct method of nonparametric measurement selection. IEEE Transactions on Computers C-20(9), 1100–1103 (1971)
26. Xue, B.: Particle Swarm Optimisation for Feature Selection in Classification. Ph.D. thesis, Victoria University of Wellington, Wellington, New Zealand (2014)
27. Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: A multi-objective particle swarm optimisation for filter based feature selection in classification problems. Connection Science 24(2-3), 91–116 (2012)
28. Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: Binary PSO and rough set theory for feature selection: A multi-objective filter based approach. International Journal of Computational Intelligence and Applications 13(02), 1450009 (2014)
29. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics 43(6), 1656–1671 (2013)
30. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing 18, 261–276 (2014)