PSO and Statistical Clustering for Feature Selection: A New Representation

Hoai Bach Nguyen[1], Bing Xue[1], Ivy Liu[2], and Mengjie Zhang[1]

[1] School of Engineering and Computer Science
[2] School of Mathematics, Statistics and Operations Research
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{nguyenhoai2,Bing.Xue,Mengjie.Zhang}@ecs.vuw.ac.nz,
Ivy.Liu@msor.vuw.ac.nz

Abstract. Classification tasks often involve a large number of features, where irrelevant or redundant features may reduce the classification performance. Such tasks typically requires a feature selection process to choose a small subset of relevant features for classification. This paper proposes a new representation in particle swarm optimisation (PSO) to utilise statistical clustering information to solve feature selection problems. The proposed algorithm is examined and compared with two conventional feature selection algorithms and two existing PSO based algorithms on eight benchmark datasets of varying difficulty. The experimental results show that the proposed algorithm can be successfully used for feature selection to considerably reduce the number of features and achieve similar or significantly higher classification accuracy than using all features. It achieves significantly better classification accuracy than one conventional method although the number of features is larger. Compared with the other conventional method and the two PSO methods, the proposed algorithm achieves better performance in terms of both the classification performance and the number of features.

Keywords: Particle swarm optimisation, Feature selection, Classification, Representation.

1 Introduction

In recent years, with the advances of data collection techniques, machine learning and data mining tasks such as classification often include a large number of features/variables. This causes the problem of "the curse of dimensionality" and leads to many issues, e.g. learning/classification algorithms fail to achieve satisfactory accuracy, the classification process is time-consuming, and the trained classifier is too complicated to understand/interpret. Feature selection can address these issues by removing irrelevant/redundant features and selecting only a small subset of relevant features for classification [8].

Feature selection is a challenging task due to the *large search space* and *feature interaction* problems. The size of the search space is 2^n for a dataset with n features [8]. Existing feature selection algorithms, such as greedy search based algorithms [11], suffer from stagnation in local optimal and/or high computational cost. Therefore, an efficient

G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, pp. 569–581, 2014.

global search technique is needed to address feature selection problems. Evolutionary computation (EC) techniques are a group of powerful "global" search algorithms and have been successfully applied to a variety of fields [9]. Particle swarm optimisation (PSO) [13, 19] is an EC technique based on social intelligence, which has fewer parameters and is computationally less expensive than other EC techniques, such as genetic programming (GP) and genetic algorithms (GAs). PSO has been recently used to address feature selection problems and shown a certain level of success [27].

Feature interaction is a common and complex problem in classification tasks [8]. Because of feature interaction, an individually relevant feature may become less useful or redundant when combined with other features. On the other hand, a weakly relevant feature may become highly useful when used together with other features. In an "optimal" subset, features are expected to be complementary to each other and can work together to increase the classification performance. Therefore, during the feature selection process, the removal or addition of features needs to consider the appearance or absence of other features, which increases the difficulty of feature selection tasks. Finding a way to cope with feature interaction problems is expected to increase the performance of a feature selection algorithm. Meanwhile, feature interaction is also an important issue being considered in statistical data analysis. We generalise the statistical clustering method [15, 17] by taking feature interaction into account to group relatively homogeneous features into clusters. Intuitively, these ideas could be useful to address feature interaction problems in feature selection, but this has not been seriously investigated. The main challenge is how to incorporate the statistical clustering information in the feature selection process.

1.1 Goals

The overall goal of this paper is to develop a new representation scheme to incorporate the statistical clustering information in PSO for feature selection. To achieve this goal, a statistical clustering method as a preprocessing step is performed on the training set to group features into different clusters. A new representation scheme is developed to utilise such statistical clustering information to improve the performance of PSO for feature selection. A new algorithm using the new representation is then developed and compared with two existing PSO based feature selection algorithms and two conventional algorithms on eight datasets with different numbers of features, classes and instances. Specifically, we will investigate:

- whether the new algorithm can be used to reduce the number of features and increase the classification performance,
- whether the new algorithm can utilise the statistical clustering information to achieve better performance than the two existing PSO based feature selection algorithms, and
- whether the new algorithm can achieve better performance than the two conventional feature selection algorithms.

2 Background

2.1 Particle Swarm Optimisation (PSO)

Particle swarm optimisation (PSO) [13, 19] is an evolutionary computation method, which is inspired by social behaviours such as birds flocking and fish schooling. In PSO, candidate solutions are represented by a population or a swarm of particles. In order to find the optimal solutions, each particle moves around the search space by updating its position as well as its velocity. Particularly, the current position of particle i is represented by a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where D is the dimensionality of the search space. These positions are updated by using another vector, called velocity $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. During the search process, each particle maintains a record of the position of its previous best performance, called $pbest$. The best position of its neighbours is also recoreded, which is called $gbest$. The position and velocity of each particle are updated according to the following equations:

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{i1} * (p_{id} - x_{id}^{t}) + c_2 * r_{i2} * (p_{gd} - x_{id}^{t}) \quad (1)$$

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

where t denotes the t^{th} iteration in the search process, d is then d^{th} dimension in the search space, i is the index of particle, w is inertia weight balancing the global and local search abilities, c_1 and c_2 are acceleration constants, r_{i1} and r_{i2} are random values uniformly distributed in [0,1], p_{id} and p_{gd} represent the position value of $pbest$ and $gbest$ in the d^{th} dimension, respectively.

2.2 Related Work on Feature Selection

Existing feature selection algorithms can be generally classified into two categories, filter approaches and wrapper approaches [8, 28]. Their main difference is whether a classification/learning algorithm is used during the feature selection process. A wrapper algorithm typically includes a classification algorithm to measure the classification performance of the selected features to evaluate the goodness of the selected features. Filter approaches are independent of any classification algorithm. Filter approaches are argued to be computationally cheaper and more general than wrappers, but wrapper approaches can usually achieve better classification performance than filters due to the interaction between the selected features and the classification algorithm. This work focuses mainly on wrapper feature selection. In this section, typical wrapper feature selection algorithms and the use of statistics in feature selection are briefly reviewed.

Traditional Feature Selection Methods. Sequential forward selection (SFS) [22] and sequential backward selection (SBS) [14] are two commonly used wrapper feature selection algorithms. Both of them use a greedy hill-climbing search strategy to search for the optimal feature subset. However, both SFS and SBS suffer from the so-called nesting effect, which means that once a feature is selected (discarded) it cannot be discarded (selected) later. Therefore, both SFS and SBS are easily trapped in local optima. In addition, both SFS and SBS require long computational time when the number of

features is large. In order to avoid nesting effect, Stearns [20] proposed a "plus-l-take away-r" method in which SFS was applied l times forward and then SBS was applied for r back tracking steps. However, determining the best values of (l, r) is a challenging task.

Later, Pudil et al. [18] proposed two floating selection methods, sequential backward floating selection (SBFS) and sequential forward floating selection (SFFS) to automatically determine the values of (l, r). In addition, the values of (l, r) in SBFS and SFFS that denotes the number of forward and backtracking steps are dynamically controlled instead of being fixed in the "plus-l-take away-r" method. Although the floating methods are claimed to be at least as good as the best sequential method, they are still likely to become trapped in a local optima even the criterion function is monotonic and the scale of the problem is small. Meanwhile, based on the best-first algorithm and SFFS, Gutlein et al. [11] proposed a linear forward selection (LFS) in which the number of features considered in each step is restricted. Experiments show that LFS improves the computational efficiency of sequential forward methods while maintaining comparable accuracy of the selected feature subset. However, LFS starts with ranking all the individual features without considering the presence or absence of some other features, which in turn limits the performance of the LFS algorithm in problems where there are interactions between features.

EC Approaches to Feature Selection. EC algorithms have been applied to feature selection problems, such as PSO, GAs [31], GP [16], ant colony optimisation (ACO) [12] and differential evolution (DE) [1]. Zhu et al. [31] proposed a feature selection method using a memetic algorithm that is a combination of local search and GA. Experiments show that this algorithm outperforms GA alone and other algorithms. Neshatian and Zhang [16] proposed a GP relevance measure (GPRM) to evaluate and rank feature subsets in binary classification tasks. Experiments show that the proposed method detected subsets of relevant features in different situations, where other methods had difficulties. Based on ACO, Kanan and Faez [12] developed a wrapper feature selection algorithm, which outperforms GA and other ACO based algorithms on a face detection dataset, but its performance has not been tested on other problems. Al-Ani et al. [1] also proposed a DE based feature selection method, where features are distributed to a set of wheels and DE is employed to select features from each wheel. This algorithm can significantly reduce the number of features and improve the classification performance.

Recently, BPSO has been applied to feature selection problems. Yang et al. [30] proposed two BPSO based wrapper feature selection approaches based on two inertia weight setting methods. The results show that the two algorithms can outperform SFS, SFFS, sequential GA and different hybrid GAs. Fdhila et al. [10] applied a multi-swarm PSO algorithm to solve feature selection problems. However, the computational cost of the proposed algorithm is high because it involves parallel evolutionary processes and multiple sub-swarms with a relative large number of particles. Xue et al. [27] proposed a PSO based two-stage feature selection algorithm to optimise the classification performance in the first stage and consider the number of features in the second stage. Chuang et al. [7] applied the so-called catfish effect to PSO for feature selection, which is to introduce new particles into the swarm by re-initialising the worst particles when $gbest$ has not improved for a number of iterations. The authors claimed that the introduced

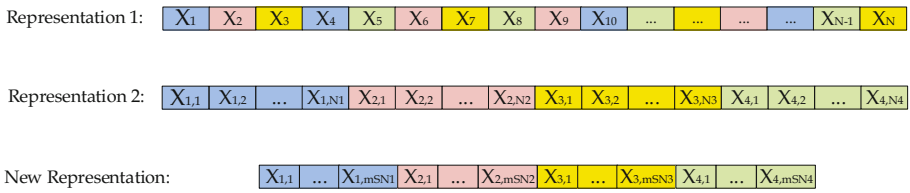

Fig. 1. Example of N features that are grouped into 4 clusters with N_1, N_2, N_3 and N_4 features, respectively, then $N = N_1 + N_2 + N_3 + N_4$. mSN_j is the predefined maximum number of features selected from cluster j and $mSN_1 < N_1, ..., mSN_4 < N_4$.

catfish particles could help PSO avoid premature convergence and lead to better results than sequential GA, SFS, SFFS and other methods. Xue et al. [29] developed new initialisation and $pbest$ and $gbest$ updating mechanisms in PSO for feature selection, which can increase the classification accuracy and reduce both the number of features and the computational time. Other PSO based feature selection methods can be found from [4–6, 24–26, 29].

Many statistical methods can be used to reduce the dimensionality of a dataset, such as principal component analysis, linear discriminant analysis, or canonical correlation analysis [3], but most of them are not feature selection approaches because they create new features. Clustering analysis is an important topic in statistics which aims to group features/variables to a number of clusters. We use the statistical clustering method [15, 17] to find relatively homogeneous feature groups by taking feature interactions into account. Therefore, the statistical grouping information could be used to develop a good feature selection algorithm.

3 Proposed Algorithm

In this section, a new representation scheme is proposed in PSO for feature selection to utilise the statistical clustering information to reduce the number of features selected and increase the classification performance. A newly developed clustering method based on statistical models proposed by Pledger and Arnold [17] and Matechou et. al. [15] is used to group features into different clusters. Features in the same cluster are considered similar and features in different clusters are dissimilar to each other. The technical detail of statistical clustering methods is not described here due to the page limit and the scope of this paper.

Fig. 1 shows three different types of representations, where a dataset with N features which can be grouped into 4 clusters is used as an example. N_1, N_2, N_3 and N_4 show the numbers of features in the 4 clusters, respectively. Representation 1 shows the traditional way of using PSO for features selection without considering the feature clustering information. Representation 2 and the proposed new representation consider the feature clustering information. Representation 2 is different from Representation 1 by putting features in the same cluster together. The new representation is different from Representations 1 and 2 in two main aspects. The first is the dimensionality of

the particles (search space). In Representations 1 and 2, the dimensionality equals to the total number of features, although Representation 2 considers the feature clustering information. In the new representation, the dimensionality equals to $\sum_{1 \leq j \leq 4} mSN_j$, where mSN_j shows the predefined maximum number of features selected from the j^{th} cluster. The second difference is the meaning of each element in the position vector. In Representations 1 and 2, each element (e.g. x_i or $x_{j,k}$) determines whether the corresponding feature is selected or not. In the new representation, each element shows which feature is selected from the corresponding cluster. Therefore, in this new representation, two important tasks are how to determine the value of mSN for each cluster and how to determine which features are selected from a cluster. They will be described as follows.

3.1 How to Determine mSN_j

Since the features from the same cluster are similar features, a small proportion of these features can be used as the representatives of this cluster. However, it is difficult to determine how many features should be selected from each cluster. Selecting a large number of features may contain redundant information while selecting a small number of features may deteriorate the classification performance. Therefore, in the new representation, we propose the use of mSN_j, which means the maximum number of features selected from the j^{th} cluster, to limit the number of features selected. The algorithm is expected to search for a feature subset which contains fewer than mSN_j features from cluster j, but can achieve better performance than using all features in cluster j. Since the sizes of clusters are usually different, the value of mSN_j should vary in different clusters.

$$mSN_j = \sqrt{N_j} \tag{3}$$

Fig. 2 compares three different ways to determine mSN_j, which are a square root function of N_j shown as Eq. 3, a constant value, and a linear function of N_j. As can be seen from the figure, Eq. 3 allows selecting more features from a cluster that contains a larger number of features, which cannot be done by the constant function. On the other hand, Eq. 3 is preferred over the linear scaling function, since it leads to a smaller number of selected features from large feature clusters, which is more likely to include redundant features. The smaller mSN_j in Eq. 3 may reduce the chance of selecting those redundant features. Therefore, in this work, Eq. 3 is used to determine the value of mSN_j.

3.2 How to Select Features

In traditional representation, the position value determine whether a feature is selected or not, which is usually determined by a threshold. If the position value is larger than the threshold, the corresponding feature is selected. Otherwise, it is not selected. In the new representation, the position value in a dimension determines which feature is

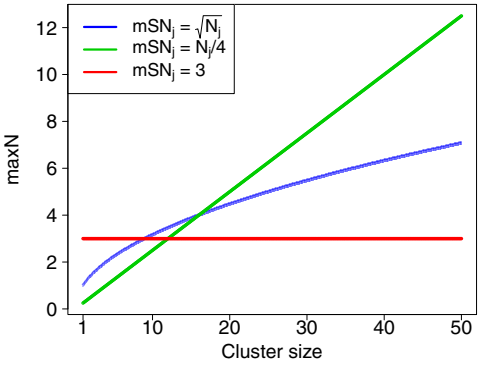

Fig. 2. Three different ways of determining mSN_j

selected from a certain cluster. To achieve this, the position value is limited to [0,1]. For the dimensions corresponding to the j^{th} cluster, [0, 1] is equally divided into $(N_j + 1)$ intervals, where N_j is the total number of features in the j^{th} cluster. Each interval corresponds to one feature in the cluster, which ensures that features in the same cluster has the equal chance to be selected. A feature is selected if the position value falls into its corresponding interval. There are $(N_j + 1)$ intervals rather than N_j intervals because a virtual feature, called "Null" feature, is introduced to each cluster. The "Null" feature allows the selection of zero feature from a cluster if all features in that cluster are irrelevant or redundant.

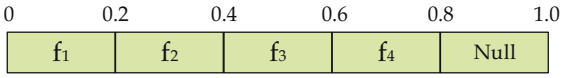

Fig. 3. Intervals for selecting features (Not PSO positions)

Fig. 3 takes a cluster with four features (f_1, f_2, f_3, f_4) showing the intervals for selecting features. As can been seen in Fig. 3, the interval [0,1] is further divided into five intervals, where four of them corresponds to the four features while the last interval corresponds to the "Null" feature, i.e. no feature is selected. Suppose that its $mSN_1 = 2$ and the position values are $\{x_{1,1} = 0.5,\ x_{1,2} = 0.96\}$. As $x_{1,1} \in [0.4, 0.6]$, which is the interval of Feature f_3, f_3 will be selected. Similarly, $x_{1,2} \in [0.8, 1.0]$ that belongs to Null feature, which means that no feature is selected. So the values are interpreted as selecting only feature f_3 from the cluster. Eq. 4 shows a general case of how to determine which feature or no feature is selected from cluster j, where x is the position value in a dimension.

$$Feature = \begin{cases} f_k, \text{if } \ x \in [\frac{k-1}{N_j+1}, \frac{k}{N_j+1}], \text{where } k \in [1, N_j] \\ \text{Null feature, if } \ [\frac{N_j}{N_j+1}, 1] \end{cases} \tag{4}$$

Algorithm 1. Pseudo-code of PSOR

```
begin
    indexing features in each cluster;
    define mSN for each cluster according to Eq. 3;
    randomly initialise the position and velocity of each particle;
    while Maximum iterations is not reached do
        Collect the features selected by each particle;
        evaluate the fitness of each particle according to its classification accuracy;
        for i = 1 to Population size do
            update pbest and gbest of particle i;
        for i = 1 to Population size do
            update v_i of particle i according to Eq. 1;
            update x_i of particle i according to Eq. 2;
    calculate the training and testing classification accuracy of the selected feature
    subset on the test set;
    return the position of gbest, the training and testing classification accuracies;
```

3.3 Pseudo-code of the Algorithm

By using the proposed representation, a new feature selection algorithm is proposed, which is named PSOR. The pseudo-code of PSOR is shown in Algorithm 1. The fitness function of PSOR is to maximise the classification accuracy of the selected features.

Table 1. Datasets

Dataset	NO. of features	**NO. of clusters**	NO. of classes	No of instances
Wine	13	**6**	3	178
Vehicle	18	**6**	4	846
Ionosphere	34	**11**	2	351
Sonar	60	**12**	2	208
Musk1	166	**14**	2	476
Arrhythmia	279	**15**	16	452
Madelon	500	**11**	2	4400
Multiple Features	649	**15**	10	2000

4 Experimental Design

To examine the performance of the proposed algorithm PSOR, two traditional feature selection methods, which are linear forward selection (LFS) [11] and greedy stepwise backward selection (GSBS), and two existing PSO based feature selection algorithms (PSOFS [27] and PSO42 [29]) are used for comparison purposes in the experiments. LFS and GSBS were derived from two typical feature selection algorithms, i.e. sequential forward selection (SFS) and sequential backward selection (SBS), respectively. LFS [11] restricts the number of features that are considered in each step of the forward selection. The greedy stepwise feature selection algorithm implemented in Weka [23] can move either forward or backward. Given that LFS performs a forward selection, a backward search is chosen in greedy stepwise search to form a greedy stepwise backward

Table 2. Experimental Results

Dataset	Method	Ave-Size	Best	Ave-Test-Acc	Std-Test-Acc	T
	All	13	76.54			-
Wine	PSOFS	7.93	98.77	95.6	1.7953	-
	PSO42	6.73	98.77	94.86	1.8628	-
	PSOR	4.75	100	96.70	3.10	
	All	18	83.86			-
Vehicle	PSOFS	9.5	87.01	85.03	0.8899	=
	PSO42	10.33	87.01	85.44	0.8372	+
	PSOR	5.87	86.22	84.72	0.8720	
	All	34	83.81			-
Ionosphere	PSOFS	12.47	93.33	88.41	2.3079	=
	PSO42	3.13	91.43	86.69	1.6444	-
	PSOR	9.7	91.43	88.63	1.6765	
	All	60	76.19			-
Sonar	PSOFS	26.1	84.13	77.3	3.5765	-
	PSO42	11.23	84.13	77.94	3.2104	=
	PSOR	14.33	84.13	78.94	4.0185	
	All	166	83.92			=
Musk1	PSOFS	85.93	88.81	84.61	2.0568	=
	PSO42	77.3	89.51	84.87	2.7042	=
	PSOR	35.03	90.21	83.12	3.4196	
	All	279	94.46			-
Arrhythmia	PSOFS	118.73	95.14	94.56	0.3517	=
	PSO42	69.77	95.59	94.77	0.4495	-
	PSOR	44.17	95.59	94.96	0.38	
	All	500	70.9			-
Madelon	PSOFS	259.07	78.97	76.35	1.0909	-
	PSO42	206.57	84.23	78.81	3.1171	-
	PSOR	54.39	85.13	83.40	2.0368	
	All	649	98.63			-
Multiple features	PSOFS	297.07	99.2	99.0	0.0934	+
	PSO42	314.5	99.2	99.0	0.0935	+
	PSOR	51.07	99.23	98.84	0.1751	

selection (GSBS). The algorithm PSOFS [27] selects features by using continuous PSO. The other PSO based algorithm, PSO42 [29], introduced a new initialisation strategy and $pbest$ and $gbest$ updating mechanism.

Eight datasets (Table 1) chosen from the UCI machine learning repository [2] are used in the experiments. These datasets have a different number of fetures, classes and instances. For each dataset, all instaces are randomly divided into a training set and a test set, which contains 70% and 30% of the instances, respectively. Up to 500 training instances are used in the statistical clustering method to group features into different clusters, where the number of clusters are listed in the second column in Table 1. In the experiments, the classification/learning algorithm is K-nearest neighbour (KNN) where K = 5. The parameters of PSO are set as follows [21]: w =0.7298, c_1 = c_2 = 1.49618, v_{max} = 6.0, population size is 30, the maximum number of iterations is 100. The fully connected topology is used. All the PSO based algorithms have been run for 30 independent times on each dataset. A statistical significance test, Wilcoxon signed-rank test, is performed to compare the classification accuracies of different algorithms. The significance level was set as 0.05.

5 Experimental Results

Table 2 shows the experimental results of the PSO based algorithms, where "All" means that all the available features are used for classification. "Ave-size" shows the average number of selected features over the 30 runs. "Best", "Ave-Test-Acc", "Std-Test-Acc" illustrate the best, average and standard deviation of the testing accuracies over the 30 independent runs. T shows the results of the statistical significance tests between the accuracy of PSOR and other algorithms. "+" or "-" means that the algorithm achieved significantly better or worse classification performance than PSOR (the more "-", the better PSOR is). "=" means there is no significant difference between them.

Table 3. Results of GSBS and LFS

Method	Wine		Vehicle		Ionosphere		Sonar	
	# Features	Accuracy(%)	# Features	Accuracy(%)	# Features	Accuracy(%)	# Features	Accuracy(%)
GSBS	8	85.19	16	75.79	30	78.1	48	68.25
LFS	7	74.07	9	83.07	4	86.67	3	77.78

Method	Musk1		Arrhythmia		Madelon		Multiple Features	
	# Features	Accuracy(%)	# Features	Accuracy(%)	# Features	Accuracy(%)	# Features	Accuracy(%)
GSBS	122	76.22	130	93.55	489	51.28	-	-
LFS	10	85.31	11	94.46	7	64.62	18	99.0

From Table 2, it can be observed that the number of features selected by PSOR is significantly smaller than the total number of features, but using the selected features only, the 5NN classification algorithm achieved significantly better or similar classification accuracy. For example, on the Multiple Features dataset, PSOR selected on average 51 features from the original 649 features, but significantly increased the classification accuracy. The results suggest that PSOR can be successfully used for feature selection to reduce the dimensionality of the data and significantly increase the classification performance over using all features.

Comparing PSOR with PSOFS, the feature subsets selected by PSOR are smaller than that of PSOFS on all the eight datasets. In terms of the classification performance, PSOR achieved similar or significantly better classification accuracy than PSOFS on seven of the eight datasets. Comparing PSOR with PSO42, it can be observed that PSOR selected smaller feature subsets and achieved similar and significantly better classification performance than PSO42 on six of the eight datasets. The results suggest that PSOR using the new representation can effectively utilising the statistical clustering information to improve the classification performance over PSOFS and PSO42 and further reduce the number of features.

5.1 Further Comparisons with Traditional Methods

The results of LFS and GSBS are shown in Table 3. Since LFS and GSBS are deterministic algorithms, each of them produces only a single solution on each dataset. Since the experiment of using GSBS on the Multiple Features dataset cannot finish within two days, the results are not listed in the table.

Comparing the results of PSOR in Table 2 with the results in Table 3, it can be seen that LFS selected a smaller number of features than PSOR, but achieved significantly worse classification accuracy than PSOR. PSOR outperformed GSBS in terms of both the number of features and the classification performance on all datasets. The results show that PSOR, which is based on PSO and the feature clustering information, can better explore the solution space to obtain better feature subsets than LFS and GSBS.

6 Conclusions and Future Work

The goal of this paper was to develop a new approach to using the statistical clustering information in PSO for feature selection. The goal was successfully achieved by developing a new representation scheme in PSO. By using the new representation, the dimensionality of the search space is reduced over the traditional representation scheme and the statistical clustering information can be incorporated in the feature selection process. We have conducted the experiments to compare the new algorithm with two conventional methods and two existing PSO algorithms without using statistical clustering information on eight datasets of varying difficulty. The results show that the proposed algorithm can effectively utilise the statistical clustering information in PSO for feature selection, which results in smaller feature subsets and better classification accuracy than the existing methods.

In future work, new search mechanisms will be investigated in PSO and statistical clustering for feature selection to further increase the classification accuracy and reduce the number of features. Meanwhile, it will be interesting to split the data multiple times to test the stability of the feature selection algorithms.

References

1. Al-Ani, A., Alsukker, A., Khushaba, R.N.: Feature subset selection using differential evolution and a wheel based search strategy. Swarm and Evolutionary Computation 9, 15–26 (2013)
2. Asuncion, A., Newman, D.: Uci machine learning repository (2007)
3. Bach, F.R., Jordan, M.I.: A probabilistic interpretation of canonical correlation analysis. Tech. rep. (2005)
4. Cervante, L., Xue, B., Shang, L., Zhang, M.: Binary particle swarm optimisation and rough set theory for dimension reduction in classification. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 2428–2435 (2013)
5. Cervante, L., Xue, B., Shang, L., Zhang, M.: A multi-objective feature selection approach based on binary pso and rough set theory. In: Middendorf, M., Blum, C. (eds.) EvoCOP 2013. LNCS, vol. 7832, pp. 25–36. Springer, Heidelberg (2013)
6. Cervante, L., Xue, B., Zhang, M., Shang, L.: Binary particle swarm optimisation for feature selection: A filter based approach. In: IEEE Congress on Evolutionary Computation (CEC 2012), pp. 881–888 (2012)
7. Chuang, L.Y., Tsai, S.W., Yang, C.H.: Improved binary particle swarm optimization using catfish effect for feature selection. Expert Systems with Applications 38, 12699–12707 (2011)

8. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(4), 131–156 (1997)
9. Engelbrecht, A.P.: Computational intelligence: an introduction, 2nd edn. Wiley (2007)
10. Fdhila, R., Hamdani, T.M., Alimi, A.M.: Distributed mopso with a new population subdivision technique for the feature selection. In: International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII 2011), pp. 81–86 (2011)
11. Gutlein, M., Frank, E., Hall, M., Karwath, A.: Large-scale attribute selection using wrappers. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2009), pp. 332–339. IEEE (2009)
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)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
14. Marill, T., Green, D.: On the effectiveness of receptors in recognition systems. IEEE Transactions on Information Theory 9(1), 11–17 (1963)
15. Matechou, E., Liu, I., Pledger, S., Arnold, R.: Biclustering models for ordinal data. Presentation at the NZ Statistical Assn. Annual Conference, University of Auckland (2011)
16. Neshatian, K., Zhang, M.: Genetic programming for feature subset ranking in binary classification problems. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (eds.) EuroGP 2009. LNCS, vol. 5481, pp. 121–132. Springer, Heidelberg (2009)
17. Pledger, S., Arnold, R.: Multivariate methods using mixtures: correspondence analysis, scaling and pattern detection. Computational Statistics and Data Analysis (2013), http://dx.doi.org/10.1016/j.csda.2013.05.013
18. Pudil, P., Novovicova, J., Kittler, J.V.: Floating search methods in feature selection. Pattern Recognition Letters 15(11), 1119–1125 (1994)
19. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation (CEC 1998), pp. 69–73 (1998)
20. Stearns, S.: On selecting features for pattern classifier. In: Proceedings of the 3rd International Conference on Pattern Recognition, pp. 71–75. IEEE Press, Coronado (1976)
21. Van Den Bergh, F.: An analysis of particle swarm optimizers. Ph.D. thesis, University of Pretoria (2006)
22. Whitney, A.: A direct method of nonparametric measurement selection. IEEE Transactions on Computers C-20(9), 1100–1103 (1971)
23. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)
24. 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)
25. 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)
26. Xue, B., Zhang, M., Browne, W.N.: Multi-objective particle swarm optimisation (PSO) for feature selection. In: Genetic and Evolutionary Computation Conference (GECCO 2012), Philadelphia, PA, USA, pp. 81–88. ACM (2012)
27. Xue, B., Zhang, M., Browne, W.N.: New fitness functions in binary particle swarm optimisation for feature selection. In: IEEE CEC 2012, pp. 2145–2152 (2012)
28. 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)

29. 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)
30. Yang, C.S., Chuang, L.Y., Li, J.C.: Chaotic maps in binary particle swarm optimization for feature selection. In: IEEE Conference on Soft Computing in Industrial Applications (SMCIA 2008), pp. 107–112 (2008)
31. Zhu, Z.X., Ong, Y.S., Dash, M.: Wrapper-filter feature selection algorithm using a memetic framework. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37(1), 70–76 (2007)