Gaussian Based Particle Swarm Optimisation and Statistical Clustering for Feature Selection

Mitchell C. Lane[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, P.O. Box 600, Wellington 6140, New Zealand
{Mitchell.Lane,Bing.Xue,Mengjie.Zhang}@ecs.vuw.ac.nz,
Ivy.Liu@msor.vuw.ac.nz

Abstract. Feature selection is an important but difficult task in classification, which aims to reduce the number of features and maintain or even increase the classification accuracy. This paper proposes a new particle swarm optimisation (PSO) algorithm using statistical clustering information to solve feature selection problems. Based on Gaussian distribution, a new updating mechanism is developed to allow the use of the clustering information during the evolutionary process of PSO based on which a new algorithm (GPSO) is developed. The proposed algorithm is examined and compared with two traditional algorithms and a PSO based algorithm which does not use clustering information on eight benchmark datasets of varying difficulty. The results show that GPSO can be successfully used for feature selection to reduce the number of features and achieve similar or even better classification performance than using all features. Meanwhile, it achieves better performance than the two traditional feature selection algorithms. It maintains the classification performance achieved by the standard PSO for feature selection algorithm, but significantly reduces the number of features and the computational cost.

Keywords: Particle swarm optimisation, Gaussian distribution, Statistical clustering, Feature selection.

1 Introduction

Feature selection is a process of selecting a small subset of relevant features from the original large feature set, which can reduce the dimensionality of the data and increase the performance of a machine learning technique (e.g. a classification algorithm). It becomes increasingly important in data mining and machine learning because of the advances of data collection techniques, which increases the total number of features included in the dataset. Existing feature selection algorithms can be broadly classified into two categories: filter and wrapper approaches [1]. Filter approaches are independent of any learning algorithm while wrapper approaches include a classification/learning algorithm as part of the evaluation function. Therefore, wrapper approaches can often achieve better accuracy than filter approaches [1].

Feature selection is a challenging task, which has a large search space with 2^n possible points, where n is the total number of features in the dataset. This leads to the

C. Blum and G. Ochoa (Eds.): EvoCOP 2014, LNCS 8600, pp. 133–144, 2014.

problems of the high computational cost and stagnation in local optima in most existing feature selection approaches. Particle swarm optimisation (PSO) [2, 3] is a powerful global search technique, which is computationally less expensive than other evolutionary computation techniques such as genetic programming (GP) and genetic algorithms (GAs) [4]. Therefore, PSO has been successfully applied to many areas, including feature selection [5–7].

Feature interaction is a common and complex problem in classification [1], which also makes feature selection a hard problem. Feature interaction may change the relationship between a feature(s) and the class labels. Due to feature interaction, an individually relevant feature may become redundant and a weakly relevant feature may become highly useful when combining with other features. The removal or addition of some features needs to consider the appearance or absence of other features. Therefore, the optimal feature subset is a group of complementary features that working together can increase the classification performance.

Many statistical measures have been applied to form the evaluation function in feature selection algorithms [8]. However, all of them are used in filter approaches. Statistical clustering methods [9, 10] can group relatively homogeneous features together based on a statistical model. This method considers all features simultaneously and takes feature interaction into account. Features in the same cluster are similar and they are dissimilar to features in other clusters. Since feature interaction is an important factor in feature selection, the statistical feature interaction information found by the clustering method can be used to develop a good feature selection algorithm. However, this has not been seriously investigated to date.

1.1 Goals

The overall goal of this paper is to investigate the use of statistical clustering information in PSO for feature selection. To achieve this goal, a statistical clustering method is performed as a preprocessing step on part of the training set to group features into different clusters. A Gaussian based updating mechanism is developed to incorporate the clustering information during the evolutionary process of PSO. A new PSO based feature selection algorithm named GPSO is then developed to reduce the number of features and increase the classification accuracy. Specifically, we will investigate:

- whether GPSO with the developed Gaussian updating mechanism can successfully utilise the clustering information to select a small subset of features to achieve similar or even better classification performance than using all features;
- whether GPSO can achieve better performance than the standard PSO for feature selection without clustering information, and
- whether GPSO can outperform two traditional feature selection algorithms.

2 Background

2.1 Particle Swarm Optimisation (PSO)

PSO is an evolutionary computation (EC) technique, which imitates the social behaviours of birds flocking and fish schooling [2, 3]. PSO uses a swarm of particles

to search for the optimal solution, where each particle represents a possible solution in the search space. Each particle has a position vector, $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$, and a velocity vector, $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$, where D is the dimensionality. During the evolutionary process, each particle remembers its previous best position ($pbest$) and the best position found so far by the swarm ($gbest$). In binary PSO (BPSO)[11], each element in the position vector is a binary value. The velocity represents the probability of an element in the position taking value 1. To achieve this, a sigmoid function $s(v_{id})$ is used to transform v_{id} to $(0, 1)$. BPSO updates the position and velocity of each particle according to Equations 1 and 2.

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

$$x_{id} = \begin{cases} 1, \text{ if } rand() < s(v_{id}) \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

where

$$s(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \tag{3}$$

where t denotes the tth iteration in the search process. d denotes the dth dimension in the search space. w is the inertia weight. c_1 and c_2 are acceleration constants. r_{1i}, r_{2i} and $rand()$ are random values uniformly distributed in $[0, 1]$. p_{id} and p_{gd} represent the value of $pbest$ and $gbest$ in the dth dimension, respectively. v_{id}^{t} is limited by a predefined maximum velocity v_{max}, where $v_{id}^{t} \in [-v_{max}, v_{max}]$.

When using BPSO for feature selection, the dimensionality of the search space is the total number of features in the dataset. "1" in the position vector means the corresponding feature is selected and "0" otherwise [5].

2.2 Related Work on Feature Selection

A number of feature selection algorithms have been proposed, which can be seen in [1, 5, 12]. Due to the page limit, only typical EC based feature selection algorithms and the role of statistics are reviewed here.

EC Approaches for Feature Selection. Zhu et al. [13] 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 et al. [14] proposed a feature ranking method for feature selection, where each feature is assigned a score according to the frequency of its appearance in a collection of GP trees and the fitness of those trees. Feature selection can be achieved by using the top-ranked features for classification. Based on ant colony optimisation (ACO), Kanan and Faez [15] 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. He et al. [16] applied a binary differential evolution (BDE) algorithm to filter feature selection with a mutual information based fitness function. However, the proposed algorithm is not compared with any other algorithm and the datasets used include a relatively small number (maximum 56) of features. Al-Ani et al. [17] also proposed a DE based 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.

Chuang et al. [5] proposed a PSO based algorithm that resets $gbest$ if it maintains the same value after several iterations. The experiments on cancer-related gene expression datasets show that the proposed algorithm can select a small number of features to improve the classification performance. Xue et al. [18] 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. Wang et al. [19] redefined the velocity in BPSO as the number of elements that should be changed in the position. The experiments show that the proposed approach is computationally less expensive than GA. Fdhila et al. [20] 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. Yang et al. [21] proposed two PSO based feature selection approaches based on two inertia weight setting methods. The results show that the two algorithms can outperform sequential forward search, sequential forward floating search, sequential GA and different hybrid GAs. Xue et al. [12, 22] also proposed a PSO based multi-objective approach for feature selection, which shows that the PSO based approach outperforms three other commonly used EC based multi-objective algorithms, i.e. NSGAII, SPEA2, and PAES.

Javani et al. [23] applied PSO for feature selection and clustering in machine learning, where each particle is used to optimise the weights for all features and cluster center values. feature selection is achieved by omitting features with a low weight. However, features with a low weight may be useful because of feature interaction and the removal may reduce the performance of the feature subset. **Note** that the clustering problem here is a machine learning task which aims to group **instances** into different clusters. This is different from the statistical clustering used in this paper, which aims to group **features** into different clusters.

Statistics in Feature Selection. Many statistical methods can be used to reduce the dimensionality of a dataset [8], such as principal component analysis, linear discriminant analysis, or canonical correlation analysis. However, most of them are not feature selection approaches because they create new features. Some researchers introduce statistical measures to evaluate the relationship between a feature and the class labels, which are then used in feature selection to evaluate the goodness of the selected features. Based on a statistical discrepancy measure, Jakub Segen [24] developed a feature selection method, which starts with the feature that best distinguishes the classes, and iteratively adds features which in combination with the chosen features improve the classification discrimination. Relief [25] uses a statistical method to select the relevant features, where each feature has a score indicating its relevance to the class labels. Relief selects all the relevant features. However, the selected features may still have redundancy because Relief does not consider the redundancy between the relevant features. Many other statistical measures such as Pearson's correlation and least square regression error, have been used in feature selection to score the significance of features in class separability.

Clustering analysis is an important class of statistical techniques that can be applied to group features/variables to a number of clusters. A statistical clustering method can group relatively homogeneous features together taking feature interactions into account [9, 10]. A statistical clustering method usually considers feature interaction in the dataset. Therefore, the statistical feature interaction information found by a statistical clustering method can be used to develop a good feature selection algorithm, but this has not been seriously investigated.

Based on PSO and a statistical clustering method [9, 10] that groups features into different clusters and similar features to the same cluster, Lane et al. [6] proposed a feature selection algorithm, which uses PSO to select one feature from each cluster. The results show that by selecting a representative feature from each cluster, the proposed algorithm can significantly reduce the number of features and increase the classification performance. This shows the the statistical clustering information (i.e. feature clusters) can provide useful information in feature selection. Therefore, this work will also utilise such information to further develop the new approach.

3 The Proposed Approach

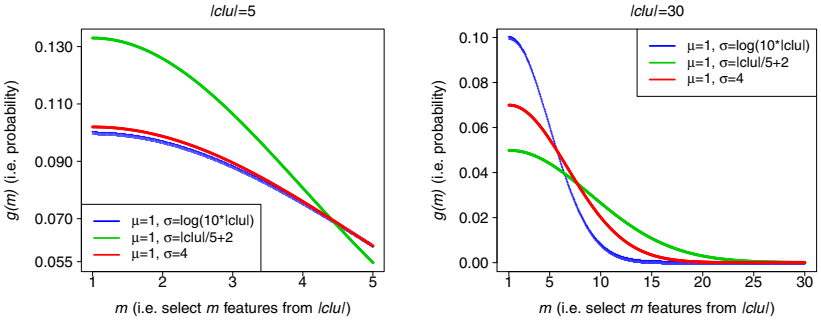
We use a newly developed clustering method based on statistical models proposed by Pledger and Arnold [9] and Matechou et. al. [10] to group features into different clusters. Due to the page limit, it is not described here. The statistical clustering method is performed as a preprocessing step on a small number of training instances to group features into different clusters.

Features in the same cluster are considered as similar features. Therefore, to use statistical clustering information for feature selection, on one hand, a single feature can be selected as a representative of its associated cluster. On the other hand, features from the same cluster might still be complementary to each other, which means that multiple features may be needed from a single feature cluster. Therefore, we want to consider feature clustering and feature interaction information to develop a new PSO approach to selecting features based on the obtained feature clusters, which is different from the traditional PSO based approach that selects features based on the whole feature set. The new approach is expected to encourage the selection of a *single* feature from each cluster, but when needed, it can also select multiple features from the same cluster. However, the original updating mechanism in PSO does not consider clustering information. Therefore, a new position updating mechanism is needed.

In PSO for feature selection, the position of a particle represents one feature subset, but the traditional position updating mechanism PSO does not consider the clustering information. Based on a Gaussian distribution (i.e. normal distribution) function, the new position updating mechanism is proposed to consider the clustering information, which first determines the number of features that will be selected from a cluster, and then determines the selection of individual features from that cluster.

3.1 Determine the Number of Features Selected

Since a small number of features is preferred, there should be a relatively large (small) probability to select a small (large) number of features from a given feature cluster.



(a) $|clu| = 5$ (b) $|clu| = 30$

Fig. 1. The effects of the standard deviation functions upon two Gaussian distributions (colour)

Gaussian distribution is used here to determine the probability of selecting a certain number (m) of features. Gaussian distribution is typically shown by $N(\mu, \sigma)$, where μ is the mean and σ is the standard deviation. The output of the Gaussian function is used here as the probability of selecting m of features from a cluster. In Gaussian function, the output value is the largest when using μ as input. Since selecting only 1 feature from each cluster is the ideal case, which should have the largest probability, $\mu = 1$ is used here. σ determining the change of the probability is a key factor, which should be defined according to the feature cluster size, i.e. the number of features in this cluster. A logarithmic function using the cluster size ($|clu|$) as the input variable, $\sigma = log(10 \times |clu|)$, is used to determine σ.

Based on $\mu = 1$ and $\sigma = log(10 \times |clu|)$, the Gaussian distribution function is built to calculate the probability of selecting m ($1 \leq m \leq |clu|$) features from a given cluster, which is shown by Equation 4.

$$g(m) = \frac{\exp(-\frac{(m-1)^2}{2log^2(10\times|clu|)})}{\sqrt{2\pi}\log(10 \times |clu|)} \tag{4}$$

Fig. 1 plots the Gaussian function shown by Equation 4, where $|clu| = 5$ in Fig. 1(a) is used as a representative of a small feature cluster, and $|clu| = 30$ in Fig. 1(b) is used as a representative of a large feature cluster. Fig. 1 also plots the Gaussian distribution of using a constant $\sigma = 4$ and and a linearly changing σ ($\sigma = |clu|/5+2$), which are used for comparison purposes to explain why $\sigma = log(10 \times |clu|)$ is chosen here. From Fig. 1(a), it can be seen that $\sigma = log(10 \times |clu|)$ provides a chance of selecting 1, 2, 3 or 4 features that is more even than the linear function, which favors selecting 1 or 2 features from the small cluster. From Fig. 1(b), it can be observed that $\sigma = log(10 \times |clu|)$ provides a much smaller chance for selecting more than 10 features than the other two standard deviation functions. Therefore, fewer redundant features will be introduced when using $\sigma = log(10 \times |clu|)$ since features are similar within a cluster.

For a given feature cluster, a desired feature list is formed by adding the features if a random value is smaller than $s(v_{id})$. If there are $|DF|$ features, the sum of all the possible $g(m)$ values should be 1. Therefore, $g(m)$ is normalised to make sure

$\sum_{m=1}^{|DF|} g(m) = 1$. Based on the normalised $g(m)$ values, a "roulette wheel selection" is performed here to determine the value of m. Note that the "roulette wheel selection" is performed on features within a cluster (not on individuals within a swarm/population). It is used here to ensure that the large $g(m)$ will have a large chance to be selected, but the small $g(m)$ will also have a chance to be selected (not completely ruled out).

3.2 How to Select Features

When using PSO for feature selection, each feature corresponds to one dimension in the position and velocity. "1" in the position means the corresponding feature is selected. Selecting m features from a cluster means m dimensions in the position are updated to "1" and all other dimensions in the same cluster are updated to "0".

In the proposed algorithm, m features are chosen based on the maximum probability mechanism, where the motivation is that the velocity in PSO represents the probability of the corresponding dimension taking value "1" [11]. In terms of feature selection, the velocity represents the probability of a feature being selected. Therefore, the m features with the highest velocity in a certain cluster should have the largest probability to be selected.

3.3 An Example

Taking a cluster with 30 features as an example, the following steps show the process of the proposed Gaussian position updating procedure. The elements in the position that correspond to other clusters are updated in the same way.

- Step 1: Build the Gaussian function $g(m)$ using $\mu = 1$ and $\sigma = log(10 \times 30)$;
- Step 2: Build a set of desired features DF: add feature i to the desired feature list if a random value is smaller than $\frac{1}{1+e^{-v_i d}}$;
- Step 3: Calculate the $g(m)$ values with $m = 1, 2, 3, ...|DF|$ and normalise them;
- Step 4: Based on the normalised $g(m)$ values, the "roulette wheel selection" is performed to determine the value of m;
- Step 5: Update the position of the m dimensions with the largest velocities to "1" and all other dimensions in the same cluster to "0".

Based on the proposed Gaussian updating mechanism, we develop a new PSO approach (named GPSO) to incorporate the statistical clustering information to address feature selection problems.

4 Experimental Design

A set of experiments have been conducted to examine the performance of the proposed algorithm (GPSO). Eight benchmark datasets shown in Table 1 were chosen from the UCI machine learning repository [26], which have different numbers of features, classes and instances. The instances in each dataset are split randomly into a training set (70%) and a test set (30%). The statistical clustering method used here was taken from a recently developed algorithm [9, 10], which is not described here due to the page limit.

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

A small number (less than 500) of training instances are used in the statistical clustering method to speed up the clustering process, which is part of the training set on datasets such as Madelon. The number of clusters obtained are listed in the third column of Table 1.

A standard BPSO based feature selection algorithm (PSOFS), which does not consider the statistical clustering information as GPSO, is used as a baseline algorithm to test the performance of GPSO. In all the two PSO based methods, K-Nearest Neighbour (KNN) with K=5 is used in the fitness function to evaluate the classification accuracy of the selected features. The parameters are set as follows [3]: w = 0.7298, c_1 = c_2 = 1.49618, v_{max} = 6.0, the population size is 30, the maximum number of iterations is 100 and the fully connected topology is used. The algorithms have been conducted for 40 independent runs on each dataset. The non-parametric statistical significance test, Wilcoxon test, is performed between the testing classification performance of a PSO algorithm and all features. The significance level is selected as 0.05 (or confidence interval is 95%).

To further examine the performance of the proposed algorithms, we also compare them with two traditional feature selection methods, which are linear forward selection (LFS) [27] and greedy stepwise backward selection (GSBS). LFS and GSBS were derived from two typical feature selection algorithms, i.e. sequential forward selection (SFS) and sequential backward selection (SBS), respectively. LFS [27] restricts the number of features that are considered in each step of the forward selection, which can reduce the number of evaluations. Therefore, LFS is computationally less expensive than SFS and can obtain good results. The greedy stepwise feature selection algorithm implemented in Weka [28] 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 selection (GSBS). GSBS starts with all available features and stops when the deletion of any remaining feature reduces the classification accuracy.

5 Results and Discussions

Table 2 shows the experimental results of PSOFS, GPSO, where "All" means that all of the available features are used for classification, "AveSize" shows the average number of features selected in the 40 independent runs, "AveAcc", "BestAcc" and "StdAcc" shows the average, the best and the standard deviation of the 40 testing accuracies. "Test" shows the results of the Wilson significance tests, where "+" (-) means PSOFS,

Table 2. Experimental Results

Dataset	Method	AveSize	BestAcc	AveAcc ± StdAcc	Test	Time
Wine	All	13	76.54			
	PSOFS	8.32	97.53	95.96 ± 1.8725	+	0.25
	GPSO	5.38	98.77	96.7 ± 2.7521	+	0.18
Vehicle	All	18	83.86			
	PSOFS	9.28	85.83	84.3 ± 0.6194	+	8.13
	GPSO	8.92	85.24	84.26 ± 0.5962	+	4.51
Ionosphere	All	34	83.81			
	PSOFS	10.38	93.33	89.05 ± 1.8444	+	1.36
	GPSO	7.5	94.29	89.26 ± 1.6631	+	0.92
Sonar	All	60	76.19			
	PSOFS	24.72	87.3	79.52 ± 2.9222	+	0.75
	GPSO	17.75	87.3	78.49 ± 3.7217	+	0.68
Musk1	All	166	83.92			
	PSOFS	83.6	89.51	85.65 ± 2.102	+	10.09
	GPSO	39.6	89.51	84.91 ± 2.5641	=	3.56
Arrhythmia	All	279	94.46			
	PSOFS	119.35	95.14	94.57 ± 0.3351	=	11.82
	GPSO	45.9	95.7	94.86 ± 0.355	+	3.83
Madelon	All	500	70.9			
	PSOFS	244.68	78.85	76.83 ± 1.2334	+	866.47
	GPSO	36.25	87.82	85.61 ± 1.0066	+	137.67
Multiple Features	All	649	98.63			
	PSOFS	295.52	99.2	99 ± 0.0962	+	726.19
	GPSO	92.25	99.27	99.02 ± 0.1258	+	112.94

GPSO is significantly better (or worse) than "All", and "=" means they are similar (no significant difference). The last column shows the average computational time used by the two PSO algorithms in a single run, which is expressed in minutes.

5.1 Results of GPSO

According to Table 2, it can be seen that the feature subsets selected by GPSO achieved significantly higher classification accuracy than using all features on **all** datasets. Furthermore, on **all** datasets, GPSO selected fewer than half of the original features, which is less than 20% on the datasets with a large number of features, i.e. the Arrhythmia, Madelon and Multiple Features datasets. For example, on the Madelon dataset, GPSO selected on average only around 7.2% of the original features (36.08 out of 500) and increased the classification accuracy from 70.9% to on average 85.61%.

Compared with PSOFS which does not use the statistical clustering information, it can be seen that GPSO achieved similar or even better classification performance than PSOFS, but the average number of features selected by GPSO is smaller or much smaller than PSOFS in **all** datasets. On the three datasets with more than 200 features, i.e. Arrhythmia, Madelon and Multiple Features, GPSO further reduced more than 60% of the feature selected by PSOFS, but still achieved slightly better classification performance than PSOFS. The reason is that on such large datasets, GPSO further removed redundant and irrelevant features, which reduced the complexity of the problem and increased the classification performance on unseen test data.

The results suggest that by developing the new Gaussian based updating mechanism in PSO, GPSO can successfully use the statistical clustering information to address feature selection problems. GPSO reduced the dimensionality of the datasets and

Table 3. Further Comparisons

Method	Wine		Vehicle		Ionosphere		Sonar	
	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy
LFS	7	74.07	9	83.07	4	86.67	3	77.78
GSBS	8	85.19	16	75.79	30	78.1	48	68.25
Method	Musk1		Arrhythmia		Madelon		Multiple Features	
	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy
LFS	10	85.31	11	94.46	7	64.62	18	99.0
GSBS	122	76.22	130	93.55	489	51.28		

simultaneously increased the classification performance in **all** cases, and also outperformed the standard PSO based feature selection algorithm, PSOFS.

5.2 Comparisons on Computational Time

According to Table 2, it can be seen that GPSO finished the evolutionary training process within 6 minutes in almost all cases, except on the Madelon and Multiple Features datasets, where a large number of features and instances are involved. Since the number of features selected by GPSO is much smaller than all the original features, the testing classification time will also be significantly reduced over using all the original features.

GPSO used a much shorter time than PSOFS on **all** datasets. The main reason is that as wrapper approaches, their computational time was mainly spent on evaluating the classification performance of the selected features, where a small number of features used a shorter time than a large number of features. GPSO selected a much smaller number of features than PSOFS, so its evaluations are much faster than PSOFS, especially on the large datasets. **Note** that although GPSO involves the statistical clustering process, this process is very fast since it is only performed on a part of the training examples. The computational time used by PSOFS is longer than the total time used by the statistical clustering method and GPSO.

5.3 Further Comparisons with Traditional Methods

Both LFS and GSBS are deterministic algorithms and only a single solution is obtained on each dataset, where the results are shown in Table 3. The results of using GSBS on the Multiple Features dataset are not available because the dataset is too big and the training process took too long time to finish.

Comparing Table 3 with Table 2, it can be seen that the number of features selected by LFS is usually smaller than GPSO, but GPSO achieved significantly better classification performance than LFS on almost all datasets. GPSO outperformed GSBS in terms of both the number of features and the classification performance on **all** datasets.

The results show that GPSO based on PSO and the feature clustering information can better explore the solution space to obtain better feature subsets than LFS and GSBS. In terms of the computational time, GPSO is slower than LFS because LFS selected a smaller number of features, but it is faster than GSBS on datasets with a relative large number of features.

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 Gaussian based updating mechanism to propose a new algorithm named GPSO. GPSO was examined and compared with two traditional feature selection algorithms (LFS and GSBS) and a standard PSO based feature selection algorithm (PSOFS) on eight benchmark datasets of varying difficulty. The results show that GPSO can successfully use the statistical clustering information to select a small subset of features and achieve similar or significantly better classification performance than using all features on **all** the eight datasets. GPSO achieved significantly better classification performance than LFS, although the number of features is slightly larger. It outperformed GSBS in terms of both the number of features and classification accuracy. GPSO achieved similar classification performance to PSOFS, but selected a much smaller number of features and used a much shorter time. Compared with the original features, GPSO achieved significantly better classification performance, and reduced the number of features to an order of magnitude on the large datasets.

This work shows that statistical clustering information can be successfully used to improve the performance of a PSO based feature selection algorithm. The successes of GPSO provides motivations to further explore the use of statistical methods with evolutionary computation techniques to solve feature selection problems. For example, statistical clustering information and PSO can be used for multi-objective feature selection or for feature construction.

References

1. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(4), 131–156 (1997)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
3. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation (CEC 1998), pp. 69–73 (1998)
4. Engelbrecht, A.P.: Computational intelligence: An introduction, 2nd edn. Wiley (2007)
5. Chuang, L.Y., Chang, H.W.: Improved binary PSO for feature selection using gene expression data. Computational Biology and Chemistry 32(29), 29–38 (2008)
6. Lane, M., Xue, B., Liu, I., Zhang, M.: Particle swarm optimisation and statistical clustering for feature selection. In: Cranefield, S., Nayak, A. (eds.) AI 2013. LNCS, vol. 8272, pp. 214–220. Springer, Heidelberg (2013)
7. 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)
8. Bach, F.R., Jordan, M.I.: A probabilistic interpretation of canonical correlation analysis. Technical report (2005)
9. 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
10. 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)

11. Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, vol. 5, pp. 4104–4108 (1997)
12. Xue, B., Zhang, M., Browne, W.: Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics 43(6), 1656–1671 (2013)
13. 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)
14. Neshatian, K., Zhang, M., Andreae, P.: Genetic programming for feature ranking in classification problems. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 544–554. Springer, Heidelberg (2008)
15. Kanan, H.R., Faez, K.: An improved feature selection method based on ant colony optimization evaluated on face recognition system. Applied Mathematics and Computation 205(2), 716–725 (2008)
16. He, X., Zhang, Q., Sun, N., Dong, Y.: Feature selection with discrete binary differential evolution. In: International Conference on Artificial Intelligence and Computational Intelligence (AICI 2009), vol. 4, pp. 327–330 (2009)
17. 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)
18. Xue, B., Zhang, M., Browne, W.: Novel initialisation and updating mechanisms in pso for feature selection in classification. In: Esparcia-Alcázar, A.I. (ed.) EvoApplications 2013. LNCS, vol. 7835, pp. 428–438. Springer, Heidelberg (2013)
19. Wang, X., Yang, J., Teng, X., Xia, W.: Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters 28(4), 459–471 (2007)
20. Fdhila, R., Hamdani, T., Alimi, A.: Distributed mopso with a new population subdivision technique for the feature selection. In: International Symposium on Computational Intelligence and Intelligent Informatics, pp. 81–86 (2011)
21. 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)
22. 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)
23. Javani, M., Faez, K., Aghlmandi, D.: Clustering and feature selection via pso algorithm. In: International Symposium on Artificial Intelligence and Signal Processing, pp. 71–76 (2011)
24. Jakub Segen, J.: Feature selection and constructive inference. In: Proceedings of Seventh International Conference on Pattern Recognition, pp. 1344–1346 (1984)
25. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Assorted Conferences and Workshops, pp. 249–256 (1992)
26. Bache, K., Lichman, M.: UCI Machine Learning Repository (2013)
27. 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 (2009)
28. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)