Single Feature Ranking and Binary Particle Swarm Optimisation Based Feature Subset Ranking for Feature Selection

Bing Xue **Mengjie Zhang** **Will N. Browne**

School of Engineering and Computer Science
Victoria University of Wellington, Wellington, New Zealand
Email: (Bing.Xue, Mengjie.Zhang, Will.Browne)@ecs.vuw.ac.nz

Abstract

This paper proposes two wrapper based feature selection approaches, which are single feature ranking and binary particle swarm optimisation (BPSO) based feature subset ranking. In the first approach, individual features are ranked according to the classification accuracy so that feature selection can be accomplished by using only a few top-ranked features for classification. In the second approach, BPSO is applied to feature subset ranking to search different feature subsets. K-nearest neighbour (KNN) with n-fold cross-validation is employed to evaluate the classification accuracy on eight datasets in the experiments. Experimental results show that using a relatively small number of the top-ranked features obtained from the first approach or one of the top-ranked feature subsets obtained from the second approach can achieve better classification performance than using all features. BPSO could efficiently search for subsets of complementary features to avoid redundancy and noise. Compared with linear forward selection (LFS) and greedy stepwise backward selection (GSBS), in almost all cases, the two proposed approaches could achieve better performance in terms of classification accuracy and the number of features. The BPSO based approach outperforms single feature ranking approach for all the datasets.

Keywords: Feature selection, Particle swarm optimisation, Single feature ranking, Feature subset ranking

1 Introduction

In many fields such as classification, a large number of features may be contained in the datasets, but not all of them are useful for classification. Redundant or irrelevant features may even reduce the classification performance. Feature selection aims to pick a subset of relevant features that are sufficient to describe the target classes. By eliminating noisy and unnecessary features, feature selection could improve classification performance, make learning and executing processes faster, and simplify the structure of the learned models (Dash & Liu 1997).

The existing feature selection approaches can be broadly classified into two categories: filter approaches and wrapper approaches. The search process in filter approaches is independent of a learning algorithm and they are argued to be computationally less expensive and more general than wrapper approaches (Dash & Liu 1997). On the other hand, wrapper approaches conduct a search for the best feature subset using the learning algorithm itself as part of the evaluation function. In a wrapper model, a feature selection algorithm exists as a wrapper around a learning algorithm and the learning algorithm is used as a "black box" by the feature selection algorithm. By considering the performance of the selected feature subset on a particular learning algorithm, wrappers can usually achieve better results than filter approaches (Kohavi & John 1997).

A feature selection algorithm explores the search space of different feature combinations to optimise the classification performance. The size of search space for n features is 2^n, so it is impractical to search the whole space exhaustively in most situations (Kohavi & John 1997). Single feature ranking is a relaxed version of feature selection, which only requires the computation of the relative importance of the features and subsequently sorting them (Guyon et al. 2003). Feature selection can be accomplished by using only the few top-ranked features for classification. However, not much work has been done on wrapper based single feature ranking (Neshatian & Zhang 2009). Single feature ranking is computationally cheap, but the combination of the top-ranked features may be a redundant subset. The performance obtained by this subset could possibly be achieved by a smaller subset of complementary features.

In order to avoid exhaustive search, greedy algorithms are introduced to solve feature selection problems such as sequential forward selection (SFS) (Whitney 1971) and sequential backward selection (SBS) (Marill & Green 1963). They are the two most commonly used greedy search algorithms that are computationally less expensive than other approaches. Thus they are used as the basis for bench-

mark techniques to test novel approaches. Existing feature selection approaches, such as greedy search algorithms, suffer from a variety of problems, such as stagnation in local optima and high computational cost. Therefore, an efficient global search technique is needed to address feature selection problems. Particle swarm optimisation (PSO) is such a global search technique, which is computationally less expensive, easier to implement, has fewer parameters and can converge more quickly than other techniques, such as genetic algorithms (GAs) and genetic programming (GP). PSO has been successfully applied in many areas and it has been shown to be a promising method for feature selection problems (Yang et al. 2008, Unler & Murat 2010, Yang et al. 2008). However, PSO has never been applied to feature subset ranking (See Section 4), which is expected to obtain many feature subsets to meet different requirements in real-world applications.

1.1 Goals

This paper aims to develop a new approach to feature subset ranking for feature selection in classification problems with the goal of using a small number of features to achieve better classification performance. To achieve this goal, we will develop two new algorithms for finding a subset of features for classification. The two algorithms will be examined and compared with conventional feature selection approaches on eight benchmark datasets with different numbers of features and instances. Specifically, we will

- develop a simple wrapper based single feature ranking algorithm and investigate whether the combination of some top-ranked features generated by this algorithm can achieve better performance than using all features and can outperform conventional approaches; and
- develop a feature subset ranking algorithm using BPSO with heuristic search and investigate whether this algorithm can outperform the method of using all features, conventional approaches and the single feature ranking algorithm.

1.2 Organisation

The remainder of the paper is organised as follow. Background information is provided in Section 2. Section 3 describes the proposed wrapper based single feature ranking algorithm. The BPSO based feature subset ranking algorithm is proposed in Section 4. Section 5 describes experimental design and Section 6 presents experimental results with discussions. Section 7 provides conclusions and future work.

2 Background

2.1 Particle Swarm Optimisation (PSO)

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995 (Kennedy & Eberhart 1995). In PSO, each solution can be represented as a particle in the search space. A vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ presents the position of particle i, where D is the dimensionality of the search space. The velocity of particle i is represented as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The best previous position of each particle is recorded as the personal best called Pbest and the best position obtained thus far is called Gbest. The swarm is initialised with a population of random solutions and searches for the best solution by updating the velocity and the position of each particle according to the following equations:

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

$$\begin{aligned} v_{id}^{t+1} &= w * v_{id}^{t} + c_1 * r_1 * (p_{id} - x_{id}^{t}) \\ &+ c_2 * r_2 * (p_{gd} - x_{id}^{t}) \end{aligned} \tag{2}$$

where t denotes iteration t in the search process. c_1 and c_2 are acceleration constants. r_1 and r_2 are random values uniformly distributed in [0, 1]. p_{id} presents the Pbest and p_{gd} stands for the Gbest. w is inertia weight. The velocity v_{id}^{t} is limited by a predefined maximum velocity, v_{max} and $v_{id}^{t} \in [-v_{max}, v_{max}]$.

PSO was originally introduced as an optimization technique for real-number search spaces. However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and between levels of variables. To extend the implementation of the original PSO, Kennedy and Eberhart (Kennedy & Eberhart 1997) developed a binary particle swarm optimisation (BPSO) for discrete problems. The velocity in BPSO represents the probability of element in the particle taking value 1 or 0. Equation (2) is still applied to update the velocity while x_{id}, p_{id} and p_{gd} are integers of 1 or 0. A sigmoid function $s(v_{id})$ is introduced to transform v_{id} to the range of (0, 1). BPSO updates the position of each particle according to the following formulae:

$$x_{id} = \begin{cases} 1, \text{ if } rand() < s(v_{id}) \\ 0, \ otherwise \end{cases} \tag{3}$$

where

$$s(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \tag{4}$$

where $s(v_{id})$ is a sigmoid limiting transformation. $rand()$ is a random number selected from a uniform distribution in [0,1].

2.2 BPSO for Feature Selection

Generally, when using BPSO to solve feature selection problems (Unler & Murat 2010, Yang et al. 2008), the representation of a particle is a n-bit binary string,

where n is the number of features and the dimensionality of the search space. The feature mask is in Boolean such that "1" represents the feature will be selected and "0" otherwise. Many BPSO based filter and wrapper feature selection approaches have been proposed in recent years.

Chakraborty (2008) compares the performance of BPSO with that of GA in a filter feature selection approach with fuzzy sets based fitness function. The results show that BPSO performs better than GA in terms of classification accuracy.

Inertia weight can improve the performance of BPSO by properly balancing its local search and global search. Yang et al. (2008) propose two strategies to determine the inertia weight of BPSO. Experiments on a wrapper feature selection model suggest that the two proposed BPSOs outperform other methods, including sequential forward search, plus and take away, sequential forward floating search, sequential GA and different hybrid GAs. In order to avoid the particles converging at local optima, Yang et al. (2008) propose a strategy to renew the Gbest during the search process to keep the diversity of the population in BPSO. In the proposed algorithm, when Gbest is identical after three generations, a Boolean operator 'and(.)' will 'and' each bit of the Pbest of all particles in an attempt to create a new Gbest. Experimental results illustrate that the proposed method usually achieves higher classification accuracy with fewer features than GA and standard BPSO.

Chuang et al. (2008) also develop a strategy for Gbest in BPSO for feature selection in which Gbest will be reset to zero if it maintains the same value after several iterations. Experiments with cancer-related human gene expression datasets show that the proposed BPSO outperforms the algorithm proposed by Yang et al. (2008) in most cases.

Wang et al. (2007) propose an improved BPSO by defining the velocity as the number of elements that should be changed. The performance of the improved BPSO is compared with that of GA in a filter feature selection model based on rough sets theories. Experimental results show that the improved BPSO is computationally less expensive than GA in terms of both memory and running time. They also conclude that most of the running time is consumed by the computation of the rough sets, which is a drawback of using rough sets to solve the feature selection problems.

Unler & Murat (2010) modify the standard BPSO by extending social learning to update the velocity of the particles. Meanwhile, an adaptive feature subset selection strategy is developed, where the features are selected not only according to the likelihood calculated by BPSO, but also according to their contribution to the subset of features already selected. The improved BPSO is applied to a wrapper feature selection model for binary classification problems. Experimental results indicate that the proposed BPSO method outperforms the tabu search and scatter search algorithms.

Alba et al. (2007) combine a geometric BPSO with a support vector machine (SVM) algorithm for feature selection, where the current position, Pbest and Gbest of a particle are used as three parents in a three-parent mask-based crossover operator to create a new position for the particle instead of using the position update equation. Experiments on high dimensional microarray problems show that the proposed algorithm could achieve slightly higher accuracy than GA with SVM in most cases. Meanwhile, experiments also show that the initialisation of the BPSO had a great influence in the performance since it introduces an early subset of acceptable solutions in the evolutionary process.

Talbi et al. (2008) propose a geometric BPSO and compare it with GA using SVM for the feature selection in high dimensional microarray data. They conclude that the performance of the proposed BPSO is superior to GA in terms of accuracy. Liu et al. (2011) propose a multiple swarm BPSO (MSPSO) to search for the best feature subset and optimise the parameters of SVM. Experimental results show that the proposed feature selection methods could achieve higher classification accuracy with a smaller subset of features than grid search, standard BPSO and GA. However, the proposed MSPSO is computationally more expensive than other three methods because of the large population size and complicated communication rules between different subswarms.

Huang & Dun (2008) develop a wrapper feature selection method based on BPSO and SVM, which uses BPSO to search for the best feature subset and continuous PSO to simultaneously optimise the parameters in the kernel function of SVM, respectively. Experiments show that the proposed algorithm could determine the parameters, search for the optimal feature subset simultaneously and also achieve high classification accuracy.

Many studies have shown that BPSO is an efficient search technique for feature selection. Therefore, it is selected as the basic tool for developing new feature subset ranking algorithms in this paper.

3 Wrapper Based Single Feature Ranking

We propose a wrapper based single feature ranking approach, where the relative importance of each feature is measured by its classification accuracy.

Algorithm 1 shows the pseudo-code of the proposed wrapper based single feature ranking approach. In this approach, each dataset is divided into two sets: a training set and a test set. In both the training set and the test set, K-nearest neighbour (KNN) with n-fold cross-validation is employed to evaluate the classification accuracy. A detailed discussion of why and how n-fold cross-validation is applied in this way is given by Kohavi & John (1997). In this algorithm, firstly, in order to make sure n-fold cross-validation is always performed on the n fixed folds, both the training set and the test set are divided into n folds when

Algorithm 1: The wrapper based single feature ranking algorithm

```
1  begin
2      divide the training set to n folds;                          // n-fold cross-validation
3      divide the test set to n folds;
4      for d=1 to number of features do
5          keep feature d and remove all the other features from training set ;   // training set only
           contains feature d
6          use KNN with n-fold cross-validation to evaluate the classification accuracy of feature d for the
           training set;
7      end
8      rank the features according to the classification accuracy;
9      for d=1 to number of features do
10         keep d top-ranked features and remove the others from the test set;
11         use KNN with n-fold cross-validation to evaluate the classification accuracy of d top-ranked
           features for the test set;
12     end
13     return classification accuracy achieved by each feature;
14     return the order of features;
15     return the classification accuracies achieved by the successive numbers of the top-ranked features;
16 end
```

Algorithm 2: The BPSO based feature subset ranking algorithm

```
1  begin
2      divide the training set to n folds                           // n-fold cross-validation
3      divide the test set to n folds;
4      initialise a feature subset S by randomly selecting 1 feature;
5      for d=1 to number of features do
6          initialise half of the swarm in BPSO with S;
7          initialise the other half of the swarm with a subset randomly selecting d features;
8          while maximum iteration or fitness=1 is not met do
9              for p=1 to number of particles do
10                 calculate sum (number of the selected features by particle p);
11                 if sum > d then
12                     randomly exclude (sum − d) features;
13                 end
14                 else if sum < d then
15                     randomly include (d − sum) features;
16                 end
17                 use KNN with n-fold cross-validation to evaluate the fitness of particle p
                   // classification accuracy of d features selected by particle p for the
                   training set
18             end
19             for p=1 to number of particles do
20                 update Pbest_p and Gbest;
21             end
22             for p=1 to number of particles do
23                 update the velocity of particle p (Equation 2);
24                 update the position of particle p (Equations 3 and 4);
25             end
26         end
27         record the evolved feature subset and the corresponding classification accuracy;
28         S ← the recorded feature subset in Line 27;
29     end
30     rank the learnt feature subsets;
31     use KNN with n-fold cross-validation to calculate the classification accuracy of the ranked feature
       subsets for the test set;
32     return the order of feature subsets and classification accuracies;
33 end
```

the algorithms starts. Secondly, every feature is used for classification in the training set individually and its classification accuracy is calculated by a loop of n-fold cross-validation on the fixed n folds of training data (from Line 4 to Line 7 in Algorithm 1). Thirdly, the features are ranked according to the classification accuracies they achieve. Finally, based on the order of the ranked features, successive numbers of the top-ranked features are selected for classification to show the utility of single feature ranking in feature selection and the classification accuracy is calculated by KNN with n-fold cross-validation on the fixed n folds of the test data (from Line 9 to Line 12 in Algorithm 1).

The proposed algorithm is simple and easy to implement (around 20 lines of code). In each dataset, the aim is to determine the number of successive top-ranked features that can achieve classification accuracy close to or even better than the classifier with all features.

4 BPSO Based Feature Subset Ranking

The top-ranked feature set resulting from the single feature ranking algorithm might contain potential redundancy. For example, the combination of the two top-ranked features might not perform as well as the combination of one top-ranked feature and a low-ranked feature if the two top-ranked features are highly dependent (redundant). To overcome this problem, we propose a feature subset ranking algorithm based on BPSO. Different feature subsets are evolved and ranked according to the classification accuracy on the training set.

Algorithm 2 shows the pseudo-code of BPSO for feature subset ranking. In this approach, each dataset is firstly divided into two sets: a training set and a test set. KNN with n-fold cross-validation is employed to evaluate the classification accuracy (Kohavi & John 1997) in both of the training set and the test set, which are divided into n folds, respectively. If a dataset includes D features, D feature subsets will be evolved and ranked. The feature subsets search process starts from finding the best subset including 1 feature and ends with the feature subset with D features. The dth feature subset includes d features, where d is a positive integer from 1 to D. There are many combinations for a feature subset with a particular number of features, and we use the dth feature subset to represent the best combination with d features in this method.

The process of selecting a certain feature subset is one step in this approach. For a dataset including D features, D feature subsets will be evolved and D steps are needed. Each step can be regarded as a process of using BPSO to select a certain number of the most relevant features (from Line 8 to Line 26 in Algorithm 2). The dth step is actually the process of using BPSO to search for the d most relevant features and the fitness function of BPSO is to maximise the classification accuracy. During the search process of BPSO, if a particle selects more than d features, a deletion strategy is employed to randomly exclude some features to reduce the number of features to d. On the other hand, if the number of selected features is smaller than d, an addition strategy is applied to randomly include some features to increase the number of the selected features to d.

During the search process, when searching for the dth feature subset, half of the population in BPSO is initialised with the $(d-1)$th feature subset achieved in the $(d-1)$th step. This is due to the expection that some of the features in the $(d-1)$th subset are useful and should be retained in the dth subset. Meanwhile, each particle in the other half of the population is initialised with a feature subset that randomly selects d features to ensure the diversity of the swarm.

All the evolved feature subsets are ranked according to the classification accuracy on the training set and then their classification performance are evaluated by KNN with n-fold cross-validation on the test set. In each dataset, the aim is to determine the number of top-ranked feature subsets that can can achieve classification accuracy close to or even better than the classifier with all features.

Table 1: Datasets

Dataset	Number of features	Number of classes	Number of instances
Vowel	10	11	990
Wine	13	3	178
Australian	14	2	690
Zoo	17	7	101
Vehicle	18	4	846
German	24	2	1000
WBCD	30	2	569
Sonar	60	2	208

5 Experimental Design

5.1 Datasets and Parameter Settings

Eight datasets chosen from the UCI machine learning repository (Frank & Asuncion 2010) are used in the experiments, which are shown in Table 1. The eight datasets were selected to have different numbers of features, classes and instances as the representative samples of the problems that the two proposed approaches could address. For two proposed approaches, in each dataset, the instances are divided into two sets: 70% as the training set and 30% as the test set. Classification accuracy is evaluated by 5NN with 10-fold cross-validation implemented in Java machine learning library (Java-ML) (Abeel et al. 2009). The classification accuracy is determined according to Equation 5:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where TP, TN, FP and FN stand for true positives, true negatives, false positives and false negatives, respectively.

The parameters of BPSO are set as follows: inertia weight $w = 0.768$, acceleration constants $c_1 = c_2 = 1.49618$, maximum velocity $v_{max} = 6.0$, population size $P = 30$, maximum iteration $T = 100$. The fully connected topology is applied in BPSO. These values are chosen based on the common settings in the literature (Van Den Bergh 2002).

For BPSO based feature subset ranking, the experiment has been conducted for 30 independent runs. The results achieved in different runs are similar to each other in terms of the classification accuracy of the evolved feature subsets. Therefore, the results from a typical run and the best results from 30 independent runs are shown in Section 6.

5.2 Benchmark Techniques

Two conventional wrapper feature selection methods, linear forward selection (LFS) and greedy stepwise backward selection (GSBS), are used as benchmark techniques to examine the performance of the two proposed approaches. They were derived from SFS and SBS, respectively.

LFS (Gutlein & Frank 2009) is an extension of best first algorithm. The search direction can be forward, or floating forward selection (with optional backward search steps). In LFS, the number of features considered in each step is restricted so that it does not exceed a certain user-specified constant. More details can be seen in the literature (Gutlein & Frank 2009).

Greedy stepwise (Caruana & Freitag 1994), implemented in Waikato Environment for Knowledge Analysis (Weka) (Witten & Frank 2005), is a steepest ascent search. It can move either forward or backward through the search space. Given that LFS performs a forward selection, a backward search is chosen in greedy stepwise to conduct a greedy stepwise backward selection. GSBS begins with all features and stops when the deletion of any remaining attribute results in a decrease in evaluation, i.e. the accuracy of classification.

Weka (Witten & Frank 2005) is used to run the experiments when using LFS and GSBS for feature selection. During the feature selection process, 5NN with 10-fold cross-validation in Weka is employed to evaluate the classification accuracy. In order to make fair comparisons, all the feature subsets selected by LFS, GSBS and two proposed methods are tested by 5NN with 10-fold cross-validation in Java-ML on the test sets.

When using Weka to run the experiments, all the settings are kept to the defaults except that backward search is chosen in the greedy stepwise approach to perform GSBS for feature selection and 5NN with 10-fold cross-validation is selected to evaluate the classification accuracy in both LFS and GSBS.

6 Results

Figure 1 shows the classification accuracy of each feature achieved by the wrapper based single feature ranking on the training set. The eight charts correspond to the eight datasets used in the experiments. In each chart, the horizontal axis shows the feature index in the corresponding dataset. The vertical axis shows the classification accuracy.

Figure 2 compares the classification performance of the two proposed methods, LFS and GSBS on the *test set*. Each plot corresponds to one of the eight datasets. In each plot, the horizontal axis shows the number of features used for classification and the vertical axis shows the classification accuracy. "SFR" in the figure stands for the results achieved by the successive numbers of top-ranked features in the wrapper based single feature ranking. For the BPSO based feature subset ranking, "FSR-Best" shows the best results in 30 independent runs and "FSR" shows the results achieved in a typical run. Both LFS and GSBS produce a unique feature subset, so have a single result for each test set. The red star denotes the classification accuracy achieved by LFS and the blue dot presents the result of GSBS. In addition, the red star and the blue dot in the plot of Vowel dataset are in the same position, which means both methods selected the same number of features and achieved the same classification accuracy.

6.1 Results of Wrapper Based Single Feature Ranking

According to Figure 1, classification accuracy achieved by each feature varies considerably, which means that they are not equally important for classification. In most cases, the difference between the highest classification accuracy and the lowest one is more than 20%, but it varies with the datasets. For example, the difference in the WBCD dataset is around 50% while the difference is only about 3% in the Vowel dataset. This is caused by the different characteristics in different datasets.

According to the results denoted by "SFR" in Figure 2, a selection of a small number of top-ranked features achieves better results than using all features in all the datasets. In almost all cases, using more top-ranked features, not only does not increase the performance, but actually causes a deterioration, especially for the Wine and Zoo datasets. The results suggest that there are interactions between some features, so the relevance level of a feature changes in the presence or absence of some other features.

6.2 Results of BPSO Based Feature Subset Ranking

According to the results ("FSR" and "FSR-Best") in Figure 2, in all the eight datasets, with many of the feature subsets evolved by BPSO the classifier can achieve higher classification accuracy than with all

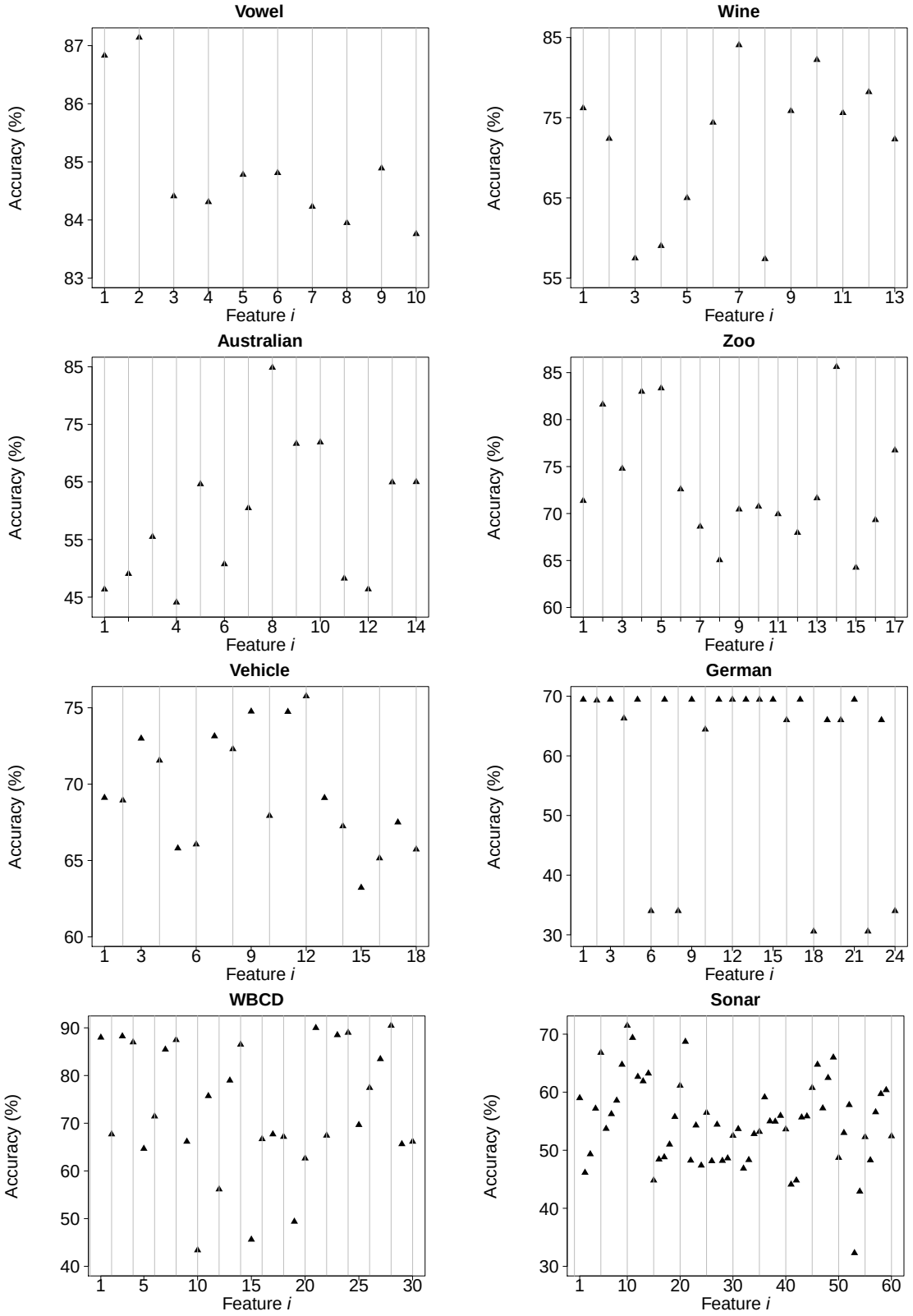

Figure 1: Results of single feature ranking

features. In most cases, the feature subset with which the classifier achieves the best performance contains a small number of features. For example, in the Australian dataset, the second feature subset evolved by BPSO only includes two features, but achieves the highest classification accuracy. This suggests that BPSO can select the relevant features and eliminate some noisy and irrelevant ones.

6.3 Comparisons Between Two Proposed Methods

Comparing the two proposed methods for feature selection, leads to the following observations. Firstly, using all features could not achieve the best performance in all the eight datasets. The two proposed methods could select a relatively small number of features with which the classifier could achieve higher

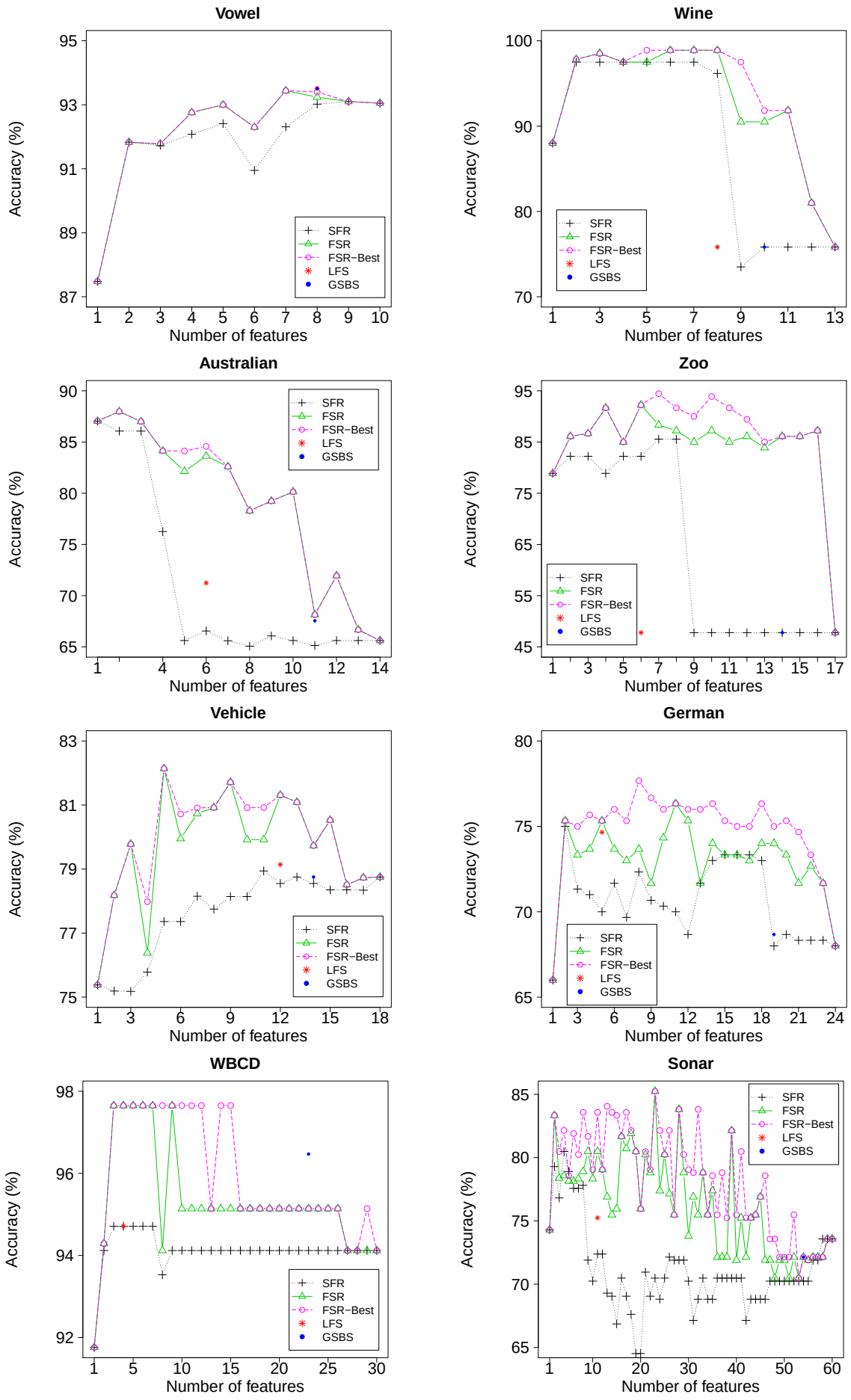

Figure 2: Comparisons between single feature ranking (SFR), feature subset ranking (FSR), the best results of FSR in 30 runs, linear forward selection (LFS) and greedy stepwise backward selection (GSBS)

classification accuracy than with all features. Secondly, in most cases, combining top-ranked features could not achieve the best performance because this combination still has redundancy. Thirdly, feature subset ranking provides an effective way for feature selection. Using the same number of features, BPSO based feature subset ranking can achieve higher classification accuracy than wrapper based single feature ranking. This suggests that BPSO could find a subset of complementary features to improve the classification performance.

6.4 Further Analysis

Results in Figure 2 show that in almost all cases, the feature subset evolved by BPSO is not the combination of the top-ranked features, but a subset of complementary ones.

Considering the Australian dataset as an example, as can be seen in Figure 1, the order of the ranked features is F8, F10, F9, F14, F13, F5, F7, F3, F6, F2, F11, F12, F1, F4, where Fi denotes the ith feature in the dataset. The second feature subset evolved by BPSO includes F8 and F12, which are not the two top-ranked features (F8 and F10). According to Figure 2, although with F8 and F10 the classifier can achieve higher classification accuracy than with all features, with F8 and F12 it can obtain better results than with F8 and F10. This suggests that the combination of the two top-ranked features is redundant while the combination of a top-ranked feature (F8) and a low-ranked feature (F12) is a subset of complementary features. Meanwhile, the other 11 (from the 3th to the 13th) feature subsets evolved by BPSO are also not the combinations of the top-ranked features. These results suggest that the BPSO based subset ranking algorithm has great potential to avoid redundant and/or noisy features and reduce the dimensionality of the classifier.

6.5 Comparisons Between Proposed Methods and Benchmark Techniques

The red star and blue dot in Figure 2 show that the number of features selected by LFS is smaller than that of GSBS, but the classification accuracy achieved by LFS is close to or better than that of GSBS in most cases. This suggests that LFS starting with an empty feature subset is more likely to obtain some optimality of the small feature subsets than backward selection methods, but does not guarantee finding the larger feature subsets. GSBS starts with all features and a feature is removed only when its removal can improve the classification performance. The redundant features that do not influence the classification accuracy will not be removed. Therefore, the feature subset selected by GSBS is usually larger than the feature subset selected by LFS because of the redundant features.

Comparing the proposed wrapper based single feature ranking with the two conventional techniques, it can be observed that using the same number of features, LFS and GSBS could achieve higher classification accuracy than single feature ranking in most cases. This suggests that the combination of top-ranked features could not achieve the best performance because it contains redundancy or noise. However, in most cases, combing a relatively small number of top-ranked features could obtain higher accuracy than LFS and GSBS. The reseason might be that the feature subsets selected by LFS and GSBS still have redundancy.

Figure 2 shows that BPSO based feature subset ranking outperforms LFS and GSBS. In seven of the eight datasets, feature subsets obtained by feature subset ranking can achieve higher classification accuracy than the subsets obtained by LFS and GSBS (in the eighth one, the Vowel dataset, the results are almost the same). This suggests that BPSO could find subsets of complementary features that could achieve better classification performance than other combinations of features.

7 Conclusions

The goal of this paper was to investigate a feature subset ranking approach to feature selection for classification. This goal was successfully achieved by developing two new wrapper based algorithms, namely a single feature ranking algorithm and a BPSO based feature subset ranking algorithm. The two algorithms were examined and compared with the corresponding method using all features, LFS and GSBS on eight problems of varying difficulty.

The results suggest that both methods can substantially improve the classification performance over the same classifier using all features. In almost all cases, the two proposed approaches could achieve higher classification accuracy whilst using fewer features than LFS and GSBS. The BPSO based feature subset ranking algorithm outperforms the simple single feature ranking algorithm on all the datasets regarding the classification performance. The results also show that on all the eight problems investigated here, it was always possible to find a subset with a small number of features that can achieve substantially better performance than using all features.

The proposed BPSO based algorithm has one limitation, that is, the evolutionary training time is relatively long. While this is usually not a problem as many situations allow offline training (as the test time is shorter using a subset of features than using all features), it might not be suitable for online (real-time) applications. We will investigate efficient feature subset ranking methods for effectively selecting good features in the future.

References

Abeel, T., de Peer, Y. V. & Saeys, Y. (2009), 'Java-ML: A Machine Learning Library', *Journal of Ma-*

chine Learning Research **10**, 931–934.

Alba, E., Garcia-Nieto, J., Jourdan, L. & Talbi, E.G. (2007), Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, *in* 'IEEE Congress on Evolutionary Computation', pp. 284–290.

Azevedo, G.L.F., Cavalcanti, G.D.C. & Filho, E.C.B. (2007), An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting, *in* 'IEEE Congress on Evolutionary Computation', pp. 3577–3584.

Caruana, R. & Freitag, D. (1994), Greedy Attribute Selection, *in* 'In Proceedings of International Conference on Machine Learning', pp. 28–36.

Chakraborty, B. (2008), Feature subset selection by particle swarm optimization with fuzzy fitness function, *in* '3rd International Conference on Intelligent System and Knowledge Engineering', Vol. 1, pp. 1038–1042.

Chuang, L.Y., Chang, H.W., Tu, C.J.& Yang, C.H. (2008), 'Improved binary PSO for feature selection using gene expression data', *Computational Biology and Chemistry* **32**(29), 29–38.

Clerc, M. & Kennedy, J. (2002), The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *in* 'IEEE Congress on Evolutionary Computation', Vol. 6, pp. 58–73.

Dash, M. & Liu, H. (1997), 'Feature selection for classification', *Intelligent Data Analysis* **1**, 131–156.

Frank, A. & Asuncion, A. (2010), *UCI Machine Learning Repository*, [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Gutlein, M., Frank, E., Hall, M. & Karwath, A. (2009), Large-scale attribute selection using wrappers, *in* 'IEEE Symposium on Computational Intelligence and Data Mining', pp. 332–339.

Guyon, I., Elisseeff, A. & Liu, H. (2003), 'An introduction to variable and feature selection', *The Journal of Machine Learning Research* **3**, 1157–1182.

Huang, C.J. & Dun, J.F. (2008), 'A distributed PSO-SVM hybrid system with feature selection and parameter optimization', *Applied Soft Computing* **8**(4), 1381–1391.

Kennedy, J. & Eberhart, R. (1995), Particle swarm optimization, *in* 'IEEE International Conference on Neural Networks',Vol. 4, pp. 1942–1948.

Kennedy, J. & Eberhart, R. (1997), A discrete binary version of the particle swarm algorithm, *in* 'IEEE International Conference on Systems, Man, and Cybernetics',Vol. 5, pp. 4104–4108.

Kennedy, J. & Spears, W.M. (1998), Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, *in* 'IEEE Congress on Evolutionary Computation',pp. 78–83.

Kohavi, R. & John, G.H. (1997), 'Wrappers for feature subset selection', *Artificial Intelligence* **97**, 315–333.

Langley, P. (1994), Selection of relevant features in machine learning, *in* 'Proceedings of the AAAI Fall symposium on relevance',pp. 127–131.

Liu, Y.N., Wang, G., Chen, H.L., & Dong, H. (2011), 'An Improved Particle Swarm Optimization for Feature Selection', *Journal of Bionic Engineering* **8**(2), 191–200.

Marill, T., & Green, D.M. (1963), 'On the effectiveness of receptors in recognition systems', *IEEE Transactions on Information Theory* **9**(1), 11–17.

Neshatian, K. & Zhang, M.J. (2009), Genetic Programming for Feature Subset Ranking in Binary Classification Problems, *in* 'European Conference on Genetic Programming', pp. 121–132.

Talbi, E.G., Jourdan, L., Garcia-Nieto, J. & Alba, E. (2008), Comparison of population based metaheuristics for feature selection: Application to microarray data classification, *in* 'ACS/IEEE International Conference on Computer Systems and Applications', pp. 45–52.

Unler, A.& Murat, A. (2010), 'A discrete particle swarm optimization method for feature selection in binary classification problems', *European Journal of Operational Research* **206**, 528–539.

Van Den Bergh, F. (2002), An analysis of particle swarm optimizers, Ph.D., University of Pretoria, South Africa.

Wang, X.Y., Yang, J., Teng, X.L. & Xia, W.J. (2007), 'Feature selection based on rough sets and particle swarm optimization', *Pattern Recognition Letters* **28**(4), 459–471.

Whitney, A.W. (2007), 'A direct method of nonparametric measurement selection', *IEEE Transactions on Computers* **20**(4), 1100–1103.

Witten I.H. & Frank E. (2005), *Data Mining: Practical Machine Learning Tools and Techniques.*, 2nd Edition, Morgan Kaufmann, San Francisco.

Yang, C.S., Chuang, L.Y. & Ke, C.H. (2008), Boolean binary particle swarm optimization for feature selection, *in* 'IEEE Congress on Evolutionary Computation', pp. 2093–2098.

Yang, C.S., Chuang, L.Y. & Li, J.C. (2008), Chaotic maps in binary particle swarm optimization for feature selection, *in* 'IEEE Conference on Soft Computing in Industrial Applications', pp. 107–112.