A Particle Swarm Optimisation Based Multi-objective Filter Approach to Feature Selection for Classification

Bing Xue[1,2], Liam Cervante[1], Lin Shang[2], and Mengjie Zhang[1]

[1] Victoria University of Wellington, P.O. Box 600, Wellington 6140, New Zealand
{Bing.Xue,Liam.Cervante,Mengjie.Zhang}@ecs.vuw.ac.nz
[2] State Key Laboratory of Novel Software Technology,
Nanjing University, Nanjing 210046, China
shanglin@nju.edu.cn

Abstract. Feature selection (FS) has two main objectives of minimising the number of features and maximising the classification performance. Based on binary particle swarm optimisation (BPSO), we develop a multi-objective FS framework for classification, which is *NSBPSO* based on multi-objective BPSO using the idea of non-dominated sorting. Two multi-objective FS algorithms are then developed by applying mutual information and entropy as two different filter evaluation criteria in the proposed framework. The two proposed multi-objective algorithms are examined and compared with two single objective FS methods on six benchmark datasets. A decision tree is employed to evaluate the classification accuracy. Experimental results show that the proposed multi-objective algorithms can automatically evolve a set of non-dominated solutions to reduce the number of features and improve the classification performance. Regardless of the evaluation criteria, NSBPSO achieves higher classification performance than the single objective algorithms. NSBPSO with entropy achieves better results than all other methods. This work represents the first study on multi-objective BPSO for filter FS in classification problems.

Keywords: Feature Selection, Particle Swarm Optimisation, Multi-Objective Optimisation, Filter Approaches.

1 Introduction

Feature selection (FS) is an important pre-processing technique for effective data analysis in many areas such as classification. In classification, without prior knowledge, relevant features are usually difficult to determine. Therefore, a large number of features are often involved, but irrelevant and redundant features may even reduce the classification performance due to the unnecessarily large search space. FS can address this problem by selecting only relevant features for classification. By eliminating/reducing irrelevant and redundant features, FS could reduce the number of features, shorten the training time, simplify the learned classifiers, and/or improve the classification performance [1].

FS algorithms explore the search space of different feature combinations to reduce the number of features and optimise the classification performance. They have two key factors: the evaluation criterion and the search strategy. Based on the evaluation criterion, existing FS approaches can be broadly classified into two categories: wrapper

P. Anthony, M. Ishizuka, and D. Lukose (Eds.): PRICAI 2012, LNAI 7458, pp. 673–685, 2012.

approaches and filter approaches. In wrapper approaches, a learning/classification algorithm is used as part of the evaluation function to determine the goodness of the selected feature subset. Wrappers can usually achieve better results than filters approaches, but the main drawbacks are their computational deficiency and loss of generality [2]. Filter approaches use statistical characteristics of the data for evaluation and the FS search process is independent of a learning/classification algorithm. Compared with wrappers, filter approaches are computationally less expensive and more general [1].

The search strategy is a key factor in FS because of the large search space (2^n for n features). In most situations, it is impractical to conduct an exhaustive search [2]. A variety of search strategies have been applied to FS. However, existing FS methods still suffer from different problems such as stagnation in local optima and high computational cost [3, 4]. Therefore, an efficient global search technique is needed to better address FS problems. Particle swarm optimisation (PSO) [5, 6] is one of the relatively recent evolutionary computation techniques, which are well-known for their global search ability. Compared with genetic algorithms (GAs) and genetic programming (GP), PSO is computationally less expensive and can converge more quickly. Therefore, PSO has been used as an effective technique in many fields, including FS in recent years [3, 4, 7].

Generally, FS has two main objectives of minimising both the classification error rate and the number of features. These two objectives are usually conflicting and the optimal decision needs to be made in the presence of a trade-off between them. However, most existing FS approaches are single objective algorithms and belong to wrapper approaches, which are less general and computationally more expensive than filter approaches. There has been no work conducted to use PSO to develop a multi-objective filter FS approach to date.

The overall goal of this paper is to develop a new PSO based multi-objective filter approach to FS for classification for finding a set of non-dominated solutions, which contain a small number of features and achieve similar or even better classification performance than using all features. To achieve this goal, we will develop a multi-objective binary PSO framework, *NSBPSO*, and apply two information measurements (mutual information and entropy) to the proposed framework. These proposed FS algorithms will be examined on six benchmark tasks/problems of varying difficulty. Specifically, we will investigate

- whether using single objective BPSO and the two information measurements can select a small number of features and improve classification performance over using all features;
- whether NSBPSO with mutual information can evolve a set of non-dominated solutions, which can outperform all features and the single objective BPSO with mutual information; and
- whether NSBPSO with entropy can outperform all other methods above.

2 Background

2.1 Particle Swarm Optimisation (PSO)

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995 [5, 6]. Candidate solutions in PSO are encoded as particles. Particles move in

the search space to search for the best solution by updating their positions according to the experience of a particle itself and its neighbours. $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ and $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$ represent the position and velocity of particle i, where D is the dimensionality of the search space. *pbest* represents the best previous position of a particle and *gbest* represents the best position obtained by the swarm so far. PSO starts with random initialisations of a population of particles and searches for the optimal solution by updating the velocity and the position of each particle according to the following equations:

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

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

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

PSO was originally proposed to address continuous problems [5]. Later, Kennedy and Eberhart [8] developed a binary PSO (BPSO) to solve discrete problems. In BPSO, x_{id}, p_{id} and p_{gd} are restricted to 1 or 0. v_{id}^{t} in BPSO indicates the probability of the corresponding element in the position vector taking value 1. A sigmoid function is used to transform v_{id} to the range of (0, 1). BPSO updates the position of each particle according to the following formula:

$$x_{id} = \begin{cases} 1, \text{ if } rand() < \frac{1}{1+e^{-v_{id}}} \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

where $rand()$ is a random number chosen from a uniform distribution in [0,1].

2.2 Entropy and Mutual Information

Information theory developed by Shannon [9] provides a way to measure the information of random variables with entropy and mutual information. The entropy is a measure of the uncertainty of random variables. Let X be a random variable with discrete values, its uncertainty can be measured by entropy $H(X)$:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \tag{4}$$

where $p(x) = Pr(X = x)$ is the probability density function of X.

For two discrete random variables X and Y with their probability density function $p(x, y)$, the joint entropy $H(X, Y)$ is defined as

$$H(X, Y) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 p(x, y) \tag{5}$$

When a variable is known and others are unknown, the remaining uncertainty is measured by the conditional entropy. Given Y, the conditional entropy $H(X|Y)$ of X with respect to Y is

$$H(X|Y) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 p(x|y) \tag{6}$$

where $p(x|y)$ is the posterior probabilities of X given Y. If X completely depends on Y, then $H(X|Y)$ is zero, which means that no more other information is required to describe X when Y is known. On the other hand, $H(X|Y) = H(X)$ denotes that knowing Y will do nothing to observe X.

The information shared between two random variables is defined as mutual information. Given variable X, mutual information $I(X;Y)$ is how much information one can gain about variable Y.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{7}$$

According to Equation 7, the mutual information $I(X;Y)$ will be large if X and Y are closely related. $I(X;Y) = 0$ if X and Y are totally unrelated.

2.3 Multi-objective Optimisation

Multi-objective optimisation involves minimising or maximising multiple conflicting objective functions. The formulae of a k-objective minimisation problem with multiple objective functions can be written as follows:

$$minimise \;\; F(x) = [f_1(x), f_2(x), \; ... \; , f_k(x)] \tag{8}$$

subject to:

$$g_i(x) \leqslant 0, \; (i = 1, 2, \; ... \; m) \qquad and \qquad h_i(x) = 0, (\; i = 1, 2, \; ... \; l) \tag{9}$$

where x is the vector of decision variables, $f_i(x)$ is a function of x, $g_i(x)$ and $h_i(x)$ are the constraint functions of the problem.

In multi-objective optimisation, the quality of a solution is explained in terms of trade-offs between the conflicting objectives. Let y and z be two solutions of the above k-objective minimisation problem. If the following conditions are met, one can say y dominates z:

$$\forall i : f_i(y) \leqslant f_i(z) \quad and \quad \exists j : f_j(y) < f_j(z) \tag{10}$$

where $i, j \in \{1, 2, ..., k\}$. When y is not dominated by any other solutions, y is referred as a Pareto-optimal solution. The set of all Pareto-optimal solutions forms the trade-off surface in the search space, the Pareto front. A multi-objective algorithm is designed to search for a set of non-dominated solutions.

2.4 Related Work on FS

A number of FS algorithms have been recently proposed [1] and typical FS algorithms are reviewed in this section.

Traditional FS Approaches. The Relief algorithm [10] is a classical filter FS algorithm. Relief assigns a weight to each feature to denote the relevance of the feature to the target concept. However, Relief does not deal with redundant features, because it attempts to find all relevant features regardless of the redundancy between them. The FOCUS algorithm [11] exhaustively examines all possible feature subsets, then selects the smallest feature subset. However, it is computationally inefficient because of the exhaustive search.

Two commonly used wrapper FS methods are sequential forward selection (SFS) [12] and sequential backward selection (SBS) [13]. SFS (SBS) starts with no features (all features), then candidate features are sequentially added to (removed from) the initial feature subset until the further addition (removal) does not increase the classification performance. The limitation of SFS and SBS is that once a feature is selected (eliminated) it cannot be eliminated (selected) later, which is so-called nesting effect. Stearns addressed this limitation by proposing the "plus-l-take away-r" to perform l times forward selection followed by r times backward elimination [14]. However, the optimal values of (l, r) are difficult to determine.

Evolutionary Computation Algorithms (Non-PSO) for FS. Recently, evolutionary computation techniques have been applied to FS problems. Based on GAs, Chakraborty [15] proposes a FS algorithm using a fuzzy-set based fitness function. However, BPSO with the same fitness function achieves better performance than this GA based algorithm. Hamdani et al. [16] develop a multi-objective FS algorithm using non-dominated sorting based multi-objective genetic algorithm II (NSGAII), but its performance has not been compared with any other FS algorithm.

Muni et al. [17] develop a multi-tree GP algorithm for FS (GPmtfs) to simultaneously select a feature subset and design a classifier using the selected features. For a c-class problem, each classifier in GPmtfs has c trees. Comparisons suggest GPmtfs achieves better results than SFS, SBS and other methods. However, the number of features selected increases when there are noisy features. Kourosh and Zhang [18] propose a GP relevance measure (GPRM) to evaluate and rank subsets of features in binary classification tasks, and GPRM is also efficient in terms of FS.

PSO Based FS Approaches. PSO has recently gained more attention for solving FS problems. Wang et al. [19] propose a filter FS algorithm based on an improved binary PSO and rough sets. Each particle is evaluated by the dependency degree between class labels and selected features, which is measured by rough sets. This work also shows that the computation of the rough sets consumes most of the running time, which is a drawback of using rough sets in FS problems.

Azevedo et al. [20] propose a FS algorithm using PSO and support vector machines (SVM) for personal identification in a keystroke dynamic system. However, the proposed algorithm obtains a relatively high false acceptance rate, which should be low in most identification systems. Mohemmed et al. [7] propose a FS method (PSOAdaBoost)

based on PSO and an AdaBoost framework. PSOAdaBoost simultaneously searches for the best feature subset and determines the decision thresholds of AdaBoost. Liu et al. [4] introduce multi-swarm PSO to search for the optimal feature subset and optimise the parameters of SVM simultaneously. The proposed FS method achieved better classification accuracy than grid search, standard PSO and GA. However, it is computationally more expensive than the other three methods because of the large population size and complicated communication rules between different subswarms.

A variety of FS approaches have been proposed, but most of them treat FS as a single objective problem. Although Hamdani et al. [16] develop a NSGAII based multi-objective algorithm, there is no comparison to test its performance. Studies have shown that PSO is an efficient technique for FS, but the use of PSO for multi-objective FS has never been investigated. Moreover, most existing approaches are wrappers, which are computationally expensive and less general than filter approaches. Therefore, investigation of a PSO based multi-objective filter FS approach is still an open issue and we make an effort in this paper.

3 Proposed Multi-objective FS Algorithms

Two filter measurements based on mutual information and entropy [21] are firstly described. Then we propose a new multi-objective BPSO framework, which forms two new algorithms to address FS problems.

3.1 Mutual Information and Entropy for FS

Mutual information can be used in FS to evaluate the relevance between a feature and the class labels and the redundancy between two features. In [21], we proposed a BPSO based filter FS algorithm (PSOfsMI) using mutual information to evaluate the relevance and redundancy in the fitness function (Equation 11). The objectives are to maximise the relevance between features and class labels to improve the classification performance, and to minimise the redundancy among features to reduce the number of features.

$$Fitness_1 = D_1 - R_1 \tag{11}$$

where

$$D_1 = \sum_{x \in X} I(x;c), \quad and \quad R_1 = \sum_{x_i, x_j \in X} I(x_i, x_j).$$

where X is the set of selected features and c is the class labels. Each selected feature and the class labels are treated as discrete random variables. D_1 calculates the mutual information between each feature and the class labels, which determine the relevance of the selected feature subset to the class labels. R_1 evaluates the mutual information shared by each pair of selected features, which indicates the redundancy contained in the selected feature subset.

Mutual information can find the two-way relevance and redundancy in FS, but could not handle multi-way complex feature interaction, which is one of the challenges in FS. Therefore, a group evaluation using entropy was proposed in [21] to discover multi-way relevance and redundancy among features. A single objective filter FS algorithm

(PSOfsE) [21] was then developed based on the group evaluation and BPSO, where Equation 12 was used as the fitness function.

$$Fitness_2 = D_2 - R_2 \tag{12}$$

where

$$D_2 = IG(c|X) \quad and \quad R_2 = \frac{1}{|X|} \sum_{x \in X} IG(x|\{X/x\})$$

where X and c have the same meaning as in Equation 11. D_2 evaluates the information gain in c given information of the features in X, which show the relevance between the selected feature subset and the class labels. R_2 evaluates the joint entropy of all the features in X, which indicates the redundancy in the selected feature subset. Detailed calculation of D_2 and R_2 is given in [21].

The representation of a particle in PSOfsMI and PSOfsE is a n-bit binary string, where n is the number of available features in the dataset and also the dimensionality of the search space. In the binary string, "1" represents that the feature is selected and "0" otherwise.

3.2 New Algorithms: NSfsMI and NSfsE

PSOfsMI and PSOfsE [21] have shown that mutual information or entropy can be an effective measurement for filter FS. Therefore, we develop a multi-objective filter FS approach based on BPSO and mutual information (or entropy) with the objectives of minimising the number of features and maximising the relevance between features and class labels. Standard PSO could not be directly used to address multi-objective problems because it was originally proposed for single objective optimisation. In order to use PSO to develop a multi-objective FS algorithm, one of the most important tasks is to determine a good leader ($gbest$) for each particle from a set of potential non-dominated solutions. NSGAII is one of the most popular evolutionary multi-objective techniques [22]. Li [23] introduces the idea of NSGAII into PSO to develop a multi-objective PSO algorithm and achieves promising results on several benchmark functions.

In this study, we develop a binary multi-objective PSO framework (NSBPSO) for filter FS based on the idea of non-dominated sorting. Two filter multi-objective FS algorithms are then developed based on NSBPSO, which are NSfsMI using D_1 to evaluate the relevance between features and class labels, and NSfsE using D_2 to measure the relevance. Algorithm 1 shows the pseudo-code of NSfsMI and NSfsE. The main idea is to use non-dominated sorting to select a $gbest$ for each particle and update the swarm in the evolutionary process. As shown in Algorithm 1, in each iteration, the algorithm firstly identifies the non-dominated solutions in the swarm and calculates the crowding distance, then all the non-dominated solutions are sorted according to the crowding distance. For each particle, a $gbest$ is randomly selected from the highest ranked part of the sorted non-dominated solutions, which are the least crowded solutions. After determining the $gbest$ and $pbest$ for each particle, the new velocity and the new position of each particle are calculated according to the equations. The old positions (solutions) and the new positions of all particles are combined into one union. The non-dominated solutions in the union are called the first non-dominated front, which are excluded from the union. Then the non-dominated solutions in the new union are called the second

Algorithm 1. Pseudo-Code of NSfsMI and NSfsE

```
begin
    divide Dataset into a Training set and a Test set, initialise the swarm (Swarm);
    while Maximum Iterations is not met do
        evaluate two objective values of each particle;   /* number of features
        and the relevance (D1 in NSfsMI and D2 in NSfsE) on
        the Training set */
        identify the particles (nonDomS) (non-dominated solutions in Swarm);
        calculate crowding distance particles in nonDomS and then sort them;
        for i=1 to Population Size (P) do
            update the pbest of particle i;
            randomly select a gbest for particle i from the highest ranked solutions in
            nonDomS;
            update the velocity and the position of particle i;
        add the original particles Swarm and the updated particles to Union;
        identify different levels of non-dominated fronts F = (F1, F2, F3, ...) in
        Union;
        empty the Swarm for the next iteration;
        i = 1;
        while |Swarm| < P do
            if (|Swarm| + |Fi| ≤ P) then
                calculate crowding distance of each particle in Fi;
                add Fi to Swarm;
                i = i + 1;
            if (|Swarm| + |Fi| > P) then
                calculate crowding distance of each particle in Fi;
                sort particles in Fi;
                add the (P − |Swarm|) least crowded particles to Swarm;
    calculate the classification error rate of the solutions (feature subsets) in the F1 on the
    test set;
    return the solutions in F1 and their testing classification error rates;
```

non-dominated front. The following levels of non-dominated fronts are identified by repeating this procedure. For the next iteration, solutions (particles) are selected from the top levels of the non-dominated fronts, starting from the first front.

4 Experimental Design

Table 1 shows the six datasets used in the experiments, which are chosen from the UCI machine learning repository [24]. The six datasets were selected to have different numbers of features, classes and instances and they are used as representative samples of the problems that the proposed algorithms will address. In the experiments, the instances in each dataset are randomly divided into two sets: 70% as the training set and 30% as the test set. All FS algorithms firstly run on the training set to select feature subsets and then the classification performance of the selected features will be calculated on the test set by a learning algorithm. There are many learning algorithms that can be used here,

Table 1. Datasets

Dataset	Type of the Data	#Features	#Classes	#Instances
Lymphography (Lymph)	Categorical	18	4	148
Mushroom	Categorical	22	2	5644
Spect	Categorical	22	2	267
Leddisplay	Categorical	24	10	1000
Soybean Large	Categorical	35	19	307
Connect4	Categorical	42	3	44473

such as K-nearest neighbour (KNN), NB, and DT. A DT learning algorithm is selected in this study to calculate the classification accuracy.

In all FS algorithms, the fully connected topology is used, $v_{max} = 6.0$, the population size is 30 and the maximum iteration is 500. $w = 0.7298$, $c_1 = c_2 = 1.49618$. These values are chosen based on the common settings in the literature [6]. Each algorithm has been conducted for 40 independent runs on each dataset.

For each dataset, PSOfsMI and PSOfsE obtain a single solution in each of the 40 runs. NSfsMI and NSfsE obtain a set of non-dominated solutions in each run. In order to compare these two kinds of results, 40 solutions in PSOfsMI and PSOfsE are presented in the next section. 40 sets of feature subsets achieved by each multi-objective algorithm are firstly combined into one union set. In the union set, for the feature subsets including the same number of features (e.g. m), their classification error rates are averaged. Therefore, a set of average solutions is obtained by using the average classification error rates and the corresponding number of features (e.g. m). The set of average solutions is called the *average* Pareto front and presented in the next section. Besides the average Pareto front, the non-dominated solutions in the union set are also presented in the next section.

5 Results and Discussions

Figures 1 and 2 show the results of NSfsMI and PSOfsMI, NSfsE and PSOfsE. On the top of each chart, the numbers in the brackets show the number of available features and the classification error rate using all features. In each chart, the horizontal axis shows the number of features selected and the vertical axis shows the classification error rate. In the figures, "-A" stands for the average Pareto front and "-B" represents the non-dominated solutions resulted from NSfsMI and NSfsE in the 40 independent runs. "PSOfsMI" and "PSOfsE" show the 40 solutions achieved by PSOfsMI and PSOfsE.

In some datasets, PSOfsMI or PSOfsE may evolve the same feature subset in different runs and they are shown in the same point in the chart. Therefore, although 40 results are presented, there may be fewer than 40 distinct points shown in a chart. For "-B", each of these non-dominated solution sets may also have duplicate feature subsets, which are shown in the same point in a chart.

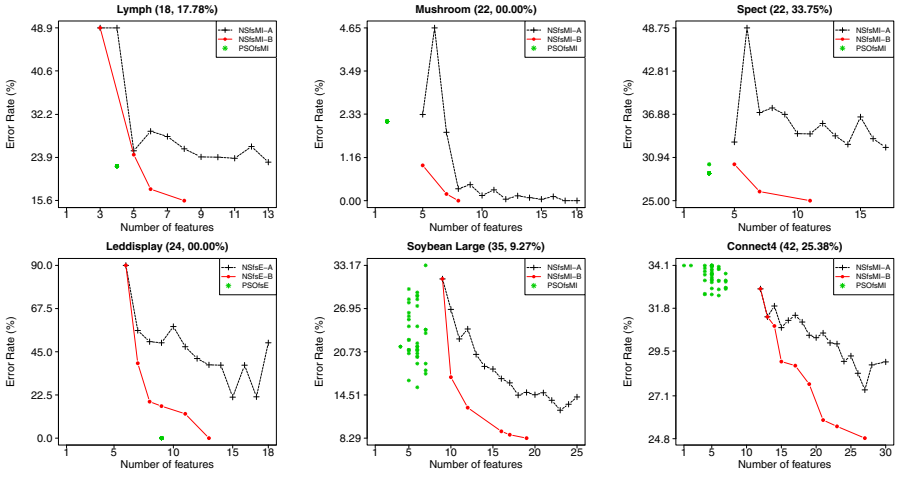

Fig. 1. Experimental Results of PSOfsMI and NSfsMI

5.1 Results of NSfsMI

According Figures 1, it can be seen that on average, PSOfsMI reduced around 75% of the available features in most cases although the classification error rates are slightly higher than using all features in some datasets.

Figure 1 shows that in three datasets, the average Pareto fronts of NSfsMI (NSfsMI-A) include two or more solutions, which selected a smaller number of features and achieved a lower classification error rate than using all features. For the same number of features, there are a variety of combinations of features with different classification performances. The feature subsets obtained in different runs may include the same number of features but different classification error rates. Therefore, although the solutions obtained in each run are non-dominated, some solutions in the average Pareto front may dominate others. This also happens in NSfsE. In almost all datasets, the non-dominated solutions (NSfsMI-B) include one or more feature subsets, which selected less than 50% of the available features and achieved better classification performance than using all features. For example, in the Spect dataset, one non-dominated solution selected 11 features from 22 available features and the classification error rate was decreased from 33.75% to 25.00%. The results suggests that NSfsMI as a multi-objective algorithm can automatically evolve a set of feature subsets to reduce the number of features and improve the classification performance.

Comparing NSfsMI with PSOfsMI, it can be seen that in most cases, NSfsMI achieved better classification performance than PSOfsMI although the number of features are slightly larger. Comparisons show that with *mutual information* as the evaluation criterion, the proposed multi-objective FS algorithms, NSfsMI can outperform single objective FS algorithm (PSOfsMI).

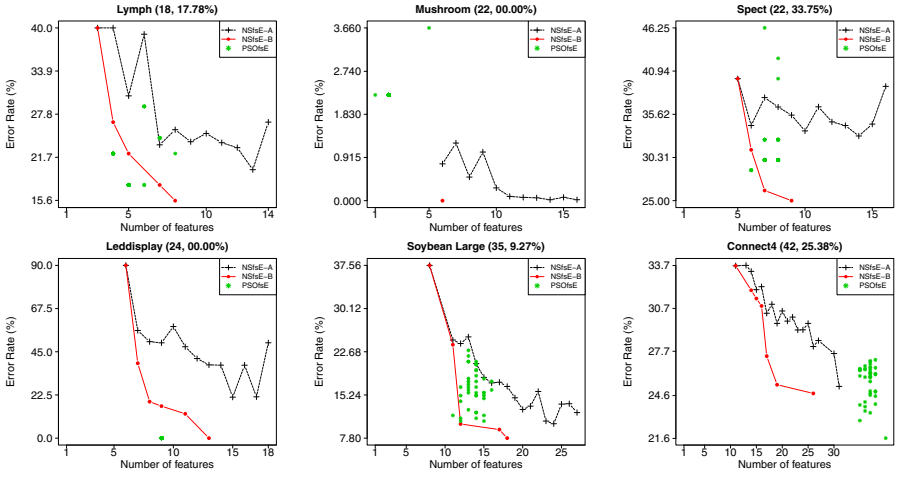

Fig. 2. Experimental Results of PSOfsE and NSfsE.

5.2 Results of NSfsE

Figure 2 shows that PSOfsE selected around half of the available features and achieved similar or even better classification performance than using all features in most cases.

Figure 2 shows that in most cases, NSfsE-A contains more than one solution that selected a smaller number of features and achieved better classification performance than using all features. In almost all datasets, NSfsMI-B reduced the classification error rate by only selecting around half of available features. Take the Spect dataset as an example, NSfsE reduced the classification error rate from 33.75% to 25.00% by selecting only 9 features from the 22 available features. The results suggest that the proposed NSfsE with *entropy* as the evaluation criterion can evolve a set of feature subsets to simultaneously improve the classification performance and reduce the number of features.

Comparing NSfsE with PSOfsE, it can be observed that NSfsE outperformed PSOfsE because NSfsE achieved better classification performance than PSOfsE in all datasets although NSfsE selected slightly more features than PSOfsE in most cases. Comparisons show that with *entropy* as the evaluation criterion, the proposed multi-objective FS algorithms (NSfsE) can achieve better solutions than single objective FS algorithm (PSOfsE).

5.3 Further Comparisons

Comparing *mutual information* and *entropy*, Figures 1 and 2 show that PSOfsE and NSfsE using entropy usually achieved better classification performance than PSOfsMI and NSfsMI using mutual information. PSOfsMI using mutual information usually selected a smaller number of features than PSOfsE using entropy. The proposed multi-objective algorithms, NSfsE usually evolved a smaller number of features and achieved better classification performance than NSfsMI. The comparisons suggest that the algorithms

with entropy as the evaluation criterion can discover the multiple-way relevancy and redundancy among a group of features to further increase the classification performance. Because the evaluation is based on a group of features (instead of a pair of features), the number of features involved is usually larger in PSOfsE than PSOfsMI. However, the number of features in the proposed multi-objective algorithms is always small because they can explore the search space more effectively to minimise the number of features. Moreover, NSfsE can utilise the discovered multiple-way relevancy to simultaneously increase the classification performance.

6 Conclusions

This paper aimed to propose a filter multi-objective FS approach based on BPSO to search for a small number of features and achieve high classification performance. The goal was successfully achieved by developing two multi-objective FS algorithms (NSfsMI and NSfsE) based on two multi-objective BPSO (NSBPSO) and two information evaluation criteria (mutual information and entropy). The proposed algorithms were examined and compared with two BPSO based single objective FS algorithms, namely PSOfsMI and PSOfsE, based on mutual information and entropy on six benchmark datasets. The experimental results show that in almost all cases, the proposed multi-objective algorithms are able to automatically evolve a Pareto front of feature subsets, which included a small number of features and achieved better classification performance than using all features. NSfsMI and NSfsE achieved better classification performance than BPSOfsMI and BPSOfsE in most cases.

The proposed multi-objective FS algorithms can achieve a set of good feature subsets, but it is unknown whether the achieved Pareto fronts can be improved or not. In the future, we will further investigate the multi-objective PSO based filter FS approach to better address FS problems.

Acknowledgments. This work is supported in part by the National Science Foundation of China (NSFC No. 61170180) and the Marsden Fund of New Zealand (VUW0806).

References

[1] Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(4), 131–156 (1997)
[2] Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
[3] Unler, A., Murat, A.: A discrete particle swarm optimization method for feature selection in binary classification problems. European Journal of Operational Research 206(3), 528–539 (2010)
[4] Liu, Y., Wang, G., Chen, H., Dong, H.: An improved particle swarm optimization for feature selection. Journal of Bionic Engineering 8(2), 191–200 (2011)
[5] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
[6] Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, CEC 1998, pp. 69–73 (1998)

[7] Mohemmed, A., Zhang, M., Johnston, M.: Particle swarm optimization based adaboost for face detection. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 2494–2501 (2009)
[8] 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)
[9] Shannon, C., Weaver, W.: The Mathematical Theory of Communication. The University of Illinois Press, Urbana (1949)
[10] Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Assorted Conferences and Workshops, pp. 249–256 (1992)
[11] Almuallim, H., Dietterich, T.G.: Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence 69, 279–305 (1994)
[12] Whitney, A.: A direct method of nonparametric measurement selection. IEEE Transactions on Computers C-20(9), 1100–1103 (1971)
[13] Marill, T., Green, D.: On the effectiveness of receptors in recognition systems. IEEE Transactions on Information Theory 9(1), 11–17 (1963)
[14] Stearns, S.: On selecting features for pattern classifier. In: Proceedings of the 3rd International Conference on Pattern Recognition, Coronado, CA, pp. 71–75 (1976)
[15] Chakraborty, B.: Genetic algorithm with fuzzy fitness function for feature selection. In: IEEE International Symposium on Industrial Electronics, ISIE 2002, vol. 1, pp. 315–319 (2002)
[16] Hamdani, T.M., Won, J.-M., Alimi, M.A.M., Karray, F.: Multi-objective Feature Selection with NSGA II. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007, Part I. LNCS, vol. 4431, pp. 240–247. Springer, Heidelberg (2007)
[17] Muni, D., Pal, N., Das, J.: Genetic programming for simultaneous feature selection and classifier design. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(1), 106–117 (2006)
[18] 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)
[19] Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature selection based on rough sets and particle swarm optimization. Pattern Recognition Letters 28(4), 459–471 (2007)
[20] Azevedo, G., Cavalcanti, G., Filho, E.: An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 3577–3584 (2007)
[21] Cervante, L., Xue, B., Zhang, M., Lin, S.: Binary particle swarm optimisation for feature selection: A filter based approach. In: IEEE Congress on Evolutionary Computation, CEC 2012, pp. 889–896 (2012)
[22] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
[23] Li, X.: A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003, Part I. LNCS, vol. 2723, pp. 37–48. Springer, Heidelberg (2003)
[24] Frank, A., Asuncion, A.: UCI machine learning repository (2010)