PSO for Feature Construction and Binary Classification

Bing Xue, Mengjie Zhang, Yan Dai, and Will N. Browne

School of Engineering and Computer Science, Victoria University of Wellington
PO Box 600, Wellington 6140, New Zealand
{Bing.Xue, Mengjie.Zhang, Yan.Dai, Will.Browne}@ecs.vuw.ac.nz

ABSTRACT

In classification, the quality of the data representation significantly influences the performance of a classification algorithm. Feature construction can improve the data representation by constructing new high-level features. Particle swarm optimisation (PSO) is a powerful search technique, but has never been applied to feature construction. This paper proposes a PSO based feature construction approach (PSOFC) to constructing a single new high-level feature using original low-level features and *directly* addressing binary classification problems without using any classification algorithm. Experiments have been conducted on seven datasets of varying difficulty. Three classification algorithms (decision trees, naïve bayes, and k-nearest neighbours) are used to evaluate the performance of the constructed feature on test set. Experimental results show that a classification algorithm using the single constructed feature often achieves similar (or even better) classification performance than using all the original features, and in almost all cases, adding the constructed feature to the original features significantly improves its classification performance. In most cases, PSOFC as a classification algorithm (using the constructed feature only) achieves better classification performance than a classification algorithm using all the original features, but needs much less computational cost. This paper represents the first study on using PSO for feature construction in classification.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence

General Terms

Algorithm, performance

Keywords

Particle swarm optimisation, feature construction, binary classification

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.

1. INTRODUCTION

Classification is an important task in the areas of machine learning and data mining. However, many classification/learning algorithms cannot achieve adequate classification performance when facing difficult problems. One of the main reasons is the poor quality of the data space representation, which is defined by the features in the dataset. Feature construction as a feature transformation method is a typical solution for this deficiency. Feature construction aims to discover hidden relationships between original features to construct new high-level features to improve the quality of the representation. The constructed feature usually act as functions of the original low-level features [19]. By transforming the input space using a set of one or more constructed features, feature construction can improve the quality of the representation, reduce its complexity, and increase the classification performance [25].

Feature construction is a difficult combinatorial problem. The size of the search space or the possible number of constructed features (i.e. the combinations of the original low-level features and function operators) grows exponentially along with the total number of original features and the potential function operators. Exhaustively search the possible solutions is impractical in most situations, especially when the total number of original features is large. Due to the large search space, feature construction approaches are often easy to become stagnated in local optima and suffer from the problem of computational expensive. Therefore, to develop a good feature construction method, an effective and efficient global search technique is needed.

Evolutionary computation techniques are global heuristic search techniques, which have been widely used in many areas [6]. Genetic programming (GP) has been successfully used for feature construction [19, 13, 20] due mainly to its capability in building programs and expression. Particle swarm optimisation (PSO) is relative recent evolutionary computation technique [11], which is inspired by social behaviour, such as birds flocking and fish schooling. Compared with other techniques, such as genetic algorithms (GAs) and GP, PSO is computationally less expensive, easier to implement, has fewer parameters and can converge more quickly [6]. As a powerful search technique, PSO has been successfully implemented in many combinatorial problems, such as process planning and scheduling [9], vehicle routing problem [2], electric power systems [3] and feature selection problems [28, 26]. However, PSO has never been used for feature construction. The main reason is that the encoding scheme and the updating mechanisms in PSO do not allow the function

operators to be included in the evolutionary process. Meanwhile, PSO is traditionally used as an optimisation technique and it has seldom been used directly for classification. This paper will start the first study on using PSO for feature construction and will also investigate the use of PSO as a general method for classification.

1.1 Goals

The overall goal of this research is to propose a PSO based approach to feature construction and binary classification, which is expected to automatically construct one new high-level feature using original low-level features and directly address binary classification problems. To achieve this goal, we propose a new approach which employs PSO as a global search technique to select low-level features, a local search to select function operators to construct a new high-level feature, and directly addresses binary classification problems without using any classification/learning algorithm. Specifically, we will investigate:

- whether a classification algorithm using the constructed high-level feature only can achieve better classification performance than using all the original features,
- whether adding the constructed high-level feature to the original feature set can improve the classification performance of a classification algorithm, and
- whether the proposed algorithm using the constructed high-level feature only can achieve better classification performance than a classification algorithm using all the original features.

In feature construction, when the original input space has a quantitative representative (when the feature values are numerical), the constructed features are a set of mathematical formulas. In case of nominal values in the original input space, the constructed features are a combination of logical and mathematical expressions. In this research, classification problems with numerical features are used and the constructed feature is a function of mathematical expressions and low-level features.

1.2 Organisation

The remainder of the paper is organised as follows. Section 2 provides background information. Section 3 describes the proposed PSO based feature construction and classification approach. Section 4 describes the design of experiments and Section 5 presents experimental results with discussions. Section 6 provides conclusions and future work.

2. BACKGROUND

2.1 Particle Swarm Optimisation (PSO)

Particle Swarm Optimisation (PSO) [11, 22] is an evolutionary computation technique. PSO is based on swarm intelligence and simulates the social behaviours of birds flocking and fish schooling. In PSO, a candidate solution is encoded as a particle in the swarm. PSO starts with a population of randomly generated particles. All the particles move in the search space to find the optimal solutions. For any particle i, a vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ is used to represent its position and a vector $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$ is used to represent its velocity, where D is the dimensionality of the search space. During the evolutionary process, particles can remember the best position visited so far, which is called personal best *pbest*. The best position obtained by the whole swarm thus far is called *gbest*, based on which a particle shares information with its neighbours. PSO iteratively updates the position and velocity of each particle to search for the optimal solutions based on *pbest* and *gbest* according to the following equations:

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

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{i1} * (p_{id} - x_{id}^{t}) + c_2 * r_{i2} * (p_{gd} - x_{id}^{t}) \quad (2)$$

where t shows the tth iteration. $d \in D$ shows the dth dimension. w is the inertia weight, which can balance the local search and global search abilities of PSO. c_1 and c_2 are acceleration constants. r_{i1} and r_{i2} are random constants uniformly distributed in [0, 1]. p_{id} and p_{gd} denote the values of *pbest* and *gbest* in the dth dimension. v_{id}^{t+1} is limited by a predefined maximum velocity, v_{max} and $v_{id}^{t+1} \in [-v_{max}, v_{max}]$.

PSO was originally proposed to address problems in continuous search spaces. To extend PSO to address discrete problems, Kennedy and Eberhart [12] developed a binary particle swarm optimisation (BPSO), where x_{id}, p_{id} and p_{gd} are restricted to 1 or 0. The velocity in BPSO represents the probability of the corresponding position taking value of 1. Therefore, a sigmoid function iss used to transfer the velocity value to (0,1). BPSO updates the position of each particle according to the following equations:

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

where

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

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

2.2 Related Work on Feature Construction

Based on whether a learning/classification algorithm is included in the feature construction process or not, feature construction approaches can be roughly divided into two categories, which are wrapper approaches and filter (non-wrapper) approaches [14].

In wrapper approaches, feature construction process includes a learning/classification algorithm and the classification performance is used to evaluate the goodness of the constructed feature(s). Murthy et al. [17] propose a system which constructs new features by linearly combining the original features in the process of learning a decision tree. Due to the capability of GP in building programs and expression, it has been used as an efficient technique for feature construction. Krawiec [13] proposes a GP based feature construction method, where C4.5 is used as a classification algorithm to evaluate the goodness of the constructed features.

Vafaie and De Jong [25] propose a feature selection and construction method, where the traditional binary string representation in GA is used for feature selection and the tree-based representation in GP is used for feature construction. C4.5 is used to evaluate the classification accuracy in the fitness function. By repeatedly performing a sequence

of feature selection and feature construct, the propsed algorithm can substantially increase the classification accuracy and/or reduce the number of features. Smith and Bull [23] use GP and GA for feature construction and feature selection, where GP is used to construct new features using original features and GA is used to select good features from the constructed features. C4.5 is also used to evaluate the classification performance of the constructed features during the evolutionary training process. Such wrapper feature construction approaches can successfully construct new features to improve the representation of the data space. However, wrapper approaches usually have the disadvantages of losing generality and long computational time [21].

In filter approaches, the process of feature construction is a separate, independent preprocessing stage and the new features are constructed before the classification algorithm is applied to build the classifier. Recently, more feature construction methods fall into this category than wrapper category because of computational efficiency and generality of filter approaches. Otero et al. [21] use GP to construct new features with information gain ratio as the fitness function. C4.5 classifier is applied to examine the classification performance. Experimental results show that by using both the newly constructed features and the original features, the classification performance or C4.5 is increased. This improvement may be because both C4.5 classifier and GP use information gain ratio as fitness function and it is unknown whether a different classifier still could benefit as much as C4.5. Muharram and Smith [16] use the similar method in [21] for feature construction and evaluate the performance of the newly constructed features using four different classifiers. However, adding the newly constructed features into the feature space could not improve the performance of the four classifiers on two of the five datasets.

Since feature construction process usually implies searching a very large space of possibilities and is often computationally demanding, Mierswa and Wurst [15] propose a case based feature construction approach to speed up the construction of new features by developing a new representation model for learning tasks and a corresponding distance measure. Neshatian at al. [20] develop a GP based feature construction algorithm using a fitness function based on the class dispersion and entropy to improve the classification performance. Neshatian and Zhang [18] develop a GP based feature construction method using a variable terminal pool which is constructed by the class-wise orthogonal transformations of the original features. Experimental results show that the proposed GP based algorithm outperforms the principle component analysis (PCA) method in terms of classification performance and dimensionality reduction. Later, Neshatian at al. [19] develop another GP based feature construction approach in which the constructed features are evolved by GP using an entropy-based fitness function to maximise the purity of class intervals. A decomposable objective function is proposed so that the system is able to construct multiple high-level features. Experimental results show that the constructed features are highly effective in increasing the classification performance, but may lack intelligibility.

2.3 PSO for Classification

PSO has been successfully used in many areas [9, 27, 4], including classification [5, 30, 29]. Typical works are reviewed in this section.

Eberhart and Shi [5] apply PSO to evolve a multi-layer perceptron artificial neural network (ANN), where PSO is used to search for the best values for the weights in ANN. The evolved ANN is then used for classification. Zhang et al. [30] apply an adaptive chaotic PSO algorithm to evolve a forward neural network (FNN) for the classification of magnetic resonance (MR) brain images. The proposed system starts with using a wavelet transform to extract features from images, and then PCA is used to reduce the number of features. The remained features are fed to FNN and PSO is used to find the best parameters for FNN. Experimental results show that the proposed algorithm can achieve good classification performance for MR brain image in a short time.

Tan et al. [24] apply PSO to a classification problem of Spam detection, where PSO is used to optimise a number of parameters. Two of the parameters are used in the proposed concentration function to generate features and others parameters are related to a classification algorithm. For two benchmark problems, the proposed algorithm can achieve good classification performance in a short time. Note that the proposed algorithm in [24] is different from the approach we propose in this paper, mainly because this algorithm aims to generate features according to the predefined concentration function instead of constructing new high-level features using existing low-level features in our approach, and this algorithm needs a classification algorithm while PSO is directly used for classification in our approach.

Sousa et al. [1] apply and compare three different versions of PSO with GA in evolving a set of classification rules (similar to classifiers). Experimental results show that the classification rules evolved by PSO achieve competitive classification performance to the rules evolved by GA and other classification algorithms. However, the proposed algorithm needs multiple evolutionary processes to generate a set of classification rules, which may be computationally expensive. Holden and Freitas [10] propose a hybrid PSO and ant colony optimisation algorithm (PSO/ACO) for the discovery of classification rules. Compared with the algorithm in [1], PSO/ACO achieves better classification performance and the discovered set of rules are more comprehensible. Xue et al. [28] propose a PSO based multi-objective feature selection algorithm to select a small number of original features and improve the classification performance. Experimental results show that the proposed PSO based algorithm outperforms other traditional feature selection algorithms. Later, Xue et al. [26] combine PSO with information theory for feature selection in classification, which can reduce the dimensionality and achieve better classification performance than other methods.

DeFalco et al. [7] directly use PSO to classification problems, where PSO is used to search for the best c centroids for a c-class problem. As each centroid is represented by all the original features, e.g. n (n is the number of features), the dimensionality of the search space for is $c * n$. This is a potential limitation because the search space is large, especially when n is large. Experimental results show that on average, the proposed algorithm achieves competitive performance with other nine classification algorithms, but in most cases, the proposed algorithm outperforms all other nine algorithms on the tested binary classification problems, which

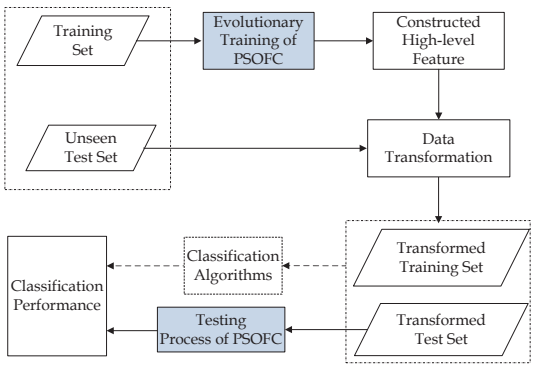

Figure 1: Overall structure.

shows that PSO has a potential to be good at addressing binary classification problems. However, the largest number of features in the datasets used in the experiments is only 58. Therefore, in this work, we will further investigate and improve the performance of PSO for binary classification on datasets with a larger number of features.

3. PROPOSED APPROACH

3.1 Overall Algorithm

In this section, we propose a PSO based feature construction and classification approach (PSOFC). The overall structure of the training and testing process of PSOFC is shown in Figure 1. As can be seen from Figure 1, PSOFC first runs on the training set of the dataset to construct a new high-level feature. Based on the constructed feature, the training set is then transformed into a new training set and the test set is transformed into a new test set. Based on the transformed test set, PSOFC itself can directly address binary classification problems without using any classification algorithm (details can be seen in Section 3.4). Meanwhile, a classification algorithm also can be used in the testing process to evaluate the classification performance of the constructed feature. The classification algorithm can be trained on the transformed training set, and the learnt classifier is then applied to the transformed test set to obtain the final classification results. Meanwhile, PSOFC itself also can directly address binary classification problems without using any classification algorithm (details can be seen in Section 3.4).

Figure 2 shows the evolutionary training process of PSOFC. PSOFC has three new components, which are the selection of low-level features, the selection of function operators to construct a new high-level feature and binary classification. Detailed description of these three components are presented in the following subsections.

3.2 Selection of Low-level Features

In the proposed algorithm, PSO is used as a global search technique to select a number of low-level original features, which are used to construct a new high-level feature.

As a low-level feature is either selected or not selected, BPSO is used in the proposed algorithm. In PSOFC, each particle is encoded as a n-bit binary string, where n is the total number of original features in the dataset. In the binary

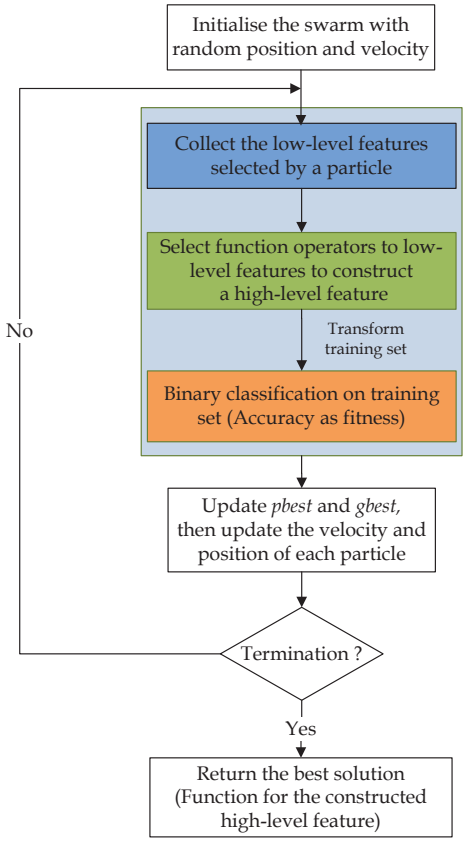

Figure 2: Evolutionary training process of PSOFC.

string, "1" means the corresponding feature is used for the construction of the new high-level feature while "0" means it is not used.

In most PSO based algorithms, each particle represents an individual solution of the problem. In the proposed algorithm (the situation of feature construction), each particle represents part of the solution, which includes a number of low-level original features. In order to construct a high-level feature, a set of function operators need to be selected and optimised together with the selected low-level features. This is the main challenge of using PSO for feature construction because the function operators can not be directly evolved by PSO.

3.3 Selection of Function Operators

To address the above problem in PSO for feature construction, we firstly define a function set, just like in GP, which includes different function operators. Then a local search is performed to select operators to construct a new high-level feature.

For each particle i, a number (m) of low-level original features are selected. For the m selected features, at least m function operators are needed. Exhaustively searching the best combination of function operators and the selected low-level features can take a relative long time. To speed up this process, we use a local search to find a near-optimal (or optimal) combination. In the proposed algorithm, the function operator for the first feature is always set as "+". Then the second feature is combined with the first feature by a function operator to form a function of the first and

the second features, which is used as a constructed feature (f_c). To choose the function operator for the second feature, every operator in the function set is firstly tested in in f_c. The operator with which f_c achieves better classification performance (Details in Section 3.4) than with others is chosen as the function operator for the second feature. Accordingly, f_c is updated. The following low-level features are sequentially added in to the function to update f_c. The function operators for these low-level features are selected in the same way as the selection of the function operator for the second feature. After selecting the function operator for the mth feature and updating f_c, f_c is a complete solution for a feature construction problem and its classification performance is used as the fitness value of particle i in PSO.

3.4 Binary Classification

In the proposed algorithm, the classification performance is used as the fitness value for a constructed feature. Since binary classification problems (with class 1 and class 2) are considered in this work, the proposed algorithm uses "0" as the threshold. An instance is classified to class 1 if the value of the constructed feature f_c is larger than 0. Otherwise, it is classified to class 2.

The classification process is independent of any classification algorithm. The advantages of such a process are to speed up the classification process and construct a general feature. The classification process can be speed up mainly because it does not involves any training and testing processes and just simply compare the single value of the constructed feature f_c with the threshold 0. The constructed high-level feature is general to different classification algorithms mainly because its classification performance (fitness) is not calculated specifically according to any classification algorithm. This is very simple and we would like to investigate whether such a simple method can work well for binary classification.

Table 1: Datasets

Dataset	No. of Features	No. of Classes	No. of Instances
Australian	14	2	690
Ionosphere	34	2	351
WBCD	30	2	569
Hillvalley	100	2	606
Musk1	166	2	476
Semeion	256	2	1593
Madelon	500	2	4400

4. DESIGN OF EXPERIMENTS

In order to examine the performance of the proposed feature construction algorithm, a set of experiments have been conducted on different binary benchmark datasets chosen from the UCI machine learning repository [8]. Due to the page limit, the results of seven typical datasets listed in Table 1 are presented in this paper. These seven datasets were chosen to have a variety numbers of features (from 14 to 500) and different numbers of instances (from 351 to 4400). They are used as representatives of different types of binary classification problems that the proposed algorithm will be test on. The instances in each dataset are randomly divided into two sets [19, 26] 70% as the training set and 30% as the test set.

The parameters of the proposed algorithm are set as follows: $w = 0.7298$, $c_1 = c_2 = 1.49618$, $v_{max} = 6.0$, population size is 30, and the maximum iteration is 100. The fully connected topology is used here. These values are chosen based on the common settings in [22]. For each dataset, the proposed algorithm has been conducted for 50 independent runs and one new high-level feature is obtained from one run. The function set used in this work includes $+$, $-$, $*$ and $'/'$ (protected division).

To examine the classification performance of the constructed high-level feature, three different classification/learning algorithms are used in the experiments, which are naïve bayes (NB), K-nearest neighbour (KNN) K=5 (5NN), and decision trees (DT). As the feature construction process is independent of any classification algorithms, using three classification algorithms (instead of only one) also can test the generality of the constructed feature.

In the experiments, in each dataset, all the original features are firstly used for classification, which is used as the baseline to compare with the classification performance of including the conducted high-level feature. Then only the single constructed high-level feature is used for classification to see whether one single constructed feature can achieve good classification performance or not. The constructed high-level feature is then used together all the original features to see whether including the constructed feature can improve the classification performance. Finally, the classification performance of the proposed algorithm is examined by comparing it (using the constructed feature only) with a classification algorithm (using all the original features).

A statistical significance test, Student's T-test (Z-test), is performed between their classification performances. The significance level in the T-tests was selected as 0.05 (or confidence interval is 95%).

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

Table 2 shows the experimental results of the proposed feature construction algorithm, where DT, 5NN and NB are used as the classification algorithm. In the table, "# Features" shows the total number of original features in each dataset. "All" means all the original features are used for classification. "CF" means only the single constructed high-level feature is used for classification. "CFOrg" means that the constructed feature is used together with all the original features for classification. "Best", "Mean" and "StdDev" show the best, the average and the standard deviation of the classification performance. "T-test" means the results of T-test, where a "+" ("-") means the classification performance of the of the feature (set) is significantly better (worse) than using all the original features. "=" means there are not significant difference between them.

5.1 Experimental Results of Only Using the Constructed feature

According to Table 2, it can be seen that only use the single constructed high-level feature, a classification algorithm (DT, 5NN or NB) achieved reasonably good classification performance in almost all cases. In many cases (around half), a classification algorithm using only the single con-

Table 2: Results of using DT, 5NN and NB as the classification algorithm.

Dataset	# Features	Method	DT				5NN				NB			
			Best	Mean	StdDev	T-test	Best	Mean	StdDev	T-test	Best	Mean	StdDev	T-test
Australian	14	All	85.99				70.05				85.51			
		CF	86.96	85.35	1.13E0	-	86.96	55.16	14.3E0	-	86.47	62.97	10.2E0	-
		CFOrg	87.44	85.93	66E-2	=	80.19	73.29	2.28E0	+	88.41	86.97	43.7E-2	+
Ionosphere	34	All	86.67				83.81				28.57			
		CF	87.62	81.14	3.16E0	-	87.62	80.53	3.7E0	-	83.81	77.56	1.71E0	+
		CFOrg	92.38	86.29	3.45E0	=	91.43	85.54	2.27E0	+	28.57	28.57	14.3E-4	=
WBCD	30	All	92.98				92.98				90.64			
		CF	95.32	93.31	1.07E0	+	95.91	92.96	1.36E0	=	61.4	61.4	35.1E-4	-
		CFOrg	97.08	93.81	1.1E0	+	94.15	92.99	21.9E-2	=	90.64	90.64	32.7E-4	=
Hillvalley	100	All	62.09				56.59				52.2			
		CF	99.45	99.41	9.53E-2	+	99.45	99.4	11E-2	+	49.45	48.5	48.6E-2	-
		CFOrg	99.45	99.41	9.53E-2	+	57.42	56.99	25.9E-2	+	53.02	52.25	19E-2	+
Musk1	166	All	71.33				83.92				42.66			
		CF	76.22	65.96	3.24E0	-	72.03	63.9	3.15E0	-	59.44	58.43	62.8E-2	+
		CFOrg	77.62	71.54	3.09E0	=	88.81	70.11	7.95E0	-	72.73	72.73	27.3E-4	+
Semeion	256	All	93.31				96.44				90.79			
		CF	100	100	0E0	+	89.96	89.96	18.4E-4	-	100	100	0E0	+
		CFOrg	100	100	0E0	+	96.44	96.44	35.1E-4	=	94.56	94.56	6.69E-4	+
Madelon	500	All	76.79				70.9				49.49			
		CF	53.97	53.97	43.6E-4	-	49.23	49.23	7.69E-4	-	49.1	49.1	25.6E-4	-
		CFOrg	76.79	76.79	48.7E-4	=	72.05	72.05	12.8E-4	+	55.38	55.38	46.2E-4	+

structed feature achieved better classification performance than using all the original features. For example, when using DT or NB as the classification algorithm, in three of the seven datasets, DT or NB using the single constructed feature achieved significantly better classification performance than using all the original features. Meanwhile, in most cases, the best classification performance of using the single constructed feature is better than using all the available features.

The results suggest that PSOFC can discover the hidden information contained by the low-level original features. The proposed algorithm can effectively select a small number of original features and use a few function operators to construct the selected low-level features to a new, high-level and (more) informative feature. In most cases, the constructed feature usually contain at least as much information as all the original features. Therefore, the average (best) classification performance of a classification algorithm using only the constructed feature is better than using all the original feature.

Another important issue is that the dimensionality, which is the number of features, is significantly reduced by PSOFC. The original dimensionality of a dataset is the total number of original features, which can be a few *hundreds* (e.g. 265 in the Semeion dataset and 500 in the Madelon dataset). After feature construction, the dimensionality of the all the datasets is reduced to only *one* by PSOFC, and the classification performance is significantly increased in many cases although slightly reduced in some cases. The best classification performance of using the single constructed feature only is better than using all available features in most cases. The reduction of the dimensionality can significantly reduce the testing time and simplify the learnt classifiers.

Meanwhile, from Table 2, we also can observe that the classification performance of the single constructed feature is general to the three different classification algorithms. For example, in the Semeion dataset, using the single constructed feature only, two of the three classification algorithms, DT and NB, achieved a classification accuracy of 100% and only 5NN achieved a slightly worse accuracy of 89.96%. The main reason is that the feature construction process of PSOFC is independent of any classification algorithm. Therefore, the achieved new constructed feature is usually general to different classification algorithms.

5.2 Combining the Constructed Feature with All Original Features

From Table 2, we can observe that in almost all cases, the classification performance of a classification algorithm (DT, 5NN or NB) using both the constructed feature and the original features is similar or better than only using the original features. In all cases, the best classification performance is at least as good as only using the original features. In this cases, the dimensionality of the problem increases from n to $n+1$, where the n is the total number of original features in the dataset. However, this can safely be ignored, especially in the datasets where n is relatively large.

The results suggest that combining the constructed high-level feature with the original features brings useful information for classification (at least no deterioration). The underlying information discovered by the constructed feature may be complementary to that of the original features without significantly increasing the dimensionality of the problem. Therefore, the classification performance is increased in most cases. In some cases, the constructed feature might also be redundant to some of the original features. In such cases, combining them together does not significantly increase the classification performance, but not decrease the classification performance either. As the construction process of the high-level feature is independent of any classification algorithm, all the three classification algorithms can achieve better classification performance when combining it with the original features, which is a major achievement.

5.3 Performance of PSOFC as a Classification Algorithm

The proposed algorithm also involves directly solving binary classification problems without using any classification algorithm. It simply uses "0" as the threshold, where a instance is classified to class 1 if the value of the constructed feature is larger than 0. Otherwise, it is classified to class 0. To test the classification performance of PSOFC, the test-

ing accuracy of PSOFC (using the constructed feature only) is compared with that of the three classification algorithms using all the original features.

Table 3 shows the classification performance of PSOFC and that of three different classification algorithms, DT, 5NN and NB using all the original features. Note that "+" ("-") in the T-test means that the classification algorithm using all the original features achieved significantly better (worse) classification performance than PSOFC. "=" means they are similar.

From Table 3, we can observe that for the seven datasets, PSOFC using the single constructed feature outperformed at least one of the three classification algorithms (DT, KNN and NB) using all the original features. In the WBCD, Hillvalley and Semeion datasets, PSOFC achieved better class performance than all the three classification algorithms. Regardless of the classification algorithm and datasets, in 13 of the 21 T-tests (around 2/3) show that PSOFC achieved significantly better classification performance than a classification algorithm using all the original features and only 7 T-tests show that PSOFC achieved worse classification performance than the classification algorithm using all the original features. The results suggest that PSOFC can be successfully used as a classification algorithm. PSOFC outperforms the three classification algorithms due mainly to the high-level information provided by the constructed feature.

Meanwhile, PSOFC as a binary classification algorithm considerably reduce the computational time. The main reasons are that PSOFC only uses one feature instead of the total number of available features and PSOFC uses "0" as the threshold for binary classification, which does not involve the time of training a classifier. The potential disadvantage is that the feature construction (evolutionary training) process is slightly long if a large number of low-level features are selected for construction of the new feature, although it does not influence the testing classification time (being short). We will address this issue in our future work.

Table 3: Classification Performance of PSOFC

Dataset	Method	Best	Mean	T-test
Australian	PSOFC	85.99	85.63	
	DT	85.99		+
	KNN	70.05		-
	NB	85.51		=
Ionosphere	PSOFC	85.71	81.71	
	DT	86.67		+
	KNN	83.81		+
	NB	28.57		-
WBCD	PSOFC	95.32	93.31	
	DT	92.98		-
	KNN	92.98		-
	NB	90.64		-
Hillvalley	PSOFC	99.73	99.68	
	DT	62.09		-
	KNN	56.59		-
	NB	52.2		-
Musk1	PSOFC	76.92	66.32	
	DT	71.33		+
	KNN	83.92		+
	NB	42.66		-
Semeion	PSOFC	100	100	
	DT	93.31		-
	KNN	96.44		-
	NB	90.79		-
Madelon	PSOFC	56.54	56.54	
	DT	76.79		+
	KNN	70.9		+
	NB	49.49		-

6. CONCLUSIONS AND FUTURE WORK

This paper represents the first work on PSO for feature construction in classification. We propose a PSO based feature construction approach to construct a high-level feature using the original low-level features and automatically address binary classification problems. To test the performance of the proposed algorithm, a set of experiments have been conducted on different binary classification problems with different numbers of features and instances. Experimental results show that in many cases, a classification/learning algorithm (DT, 5NN and NB) using the single constructed feature achieved similar or better classification performance than using all the original features, and in almost all cases, adding the constructed feature to the original features significantly improved their classification performance. The proposed algorithm using the constructed high-level feature directly addressed binary classification problems (without using any classification algorithm), and in most cases, achieved better classification performance than a classification algorithm using all the original features, but used much less testing computational time because the number of features is only one. Although the proposed algorithm employs the classification performance as the evaluation criterion, the constructed feature is general to different classification algorithms because the construction process is independent of any of them.

This works starts the research direction of using PSO for feature construction, which utilises the advantages of PSO as a global search technique and relatively cheap computational cost. It also contributes the use of PSO as a potential general method directly used for classification without using any classification algorithm, instead of its traditional role of being an optimisation technique only. This work provides solutions to meet two typical real-world requirements. On one hand, if users need a fast classification process and relatively good classification performance (similar to using all features), they can use the constructed feature only for classification. On the other hand, if the classification performance is more important than the computational time, users can add the constructed high-level feature to the original features to improve the classification accuracy.

In the future, we will further investigate the use of PSO for feature construction and classification. Our future research directions can be summarised as follows. We will further analysis the constructed feature to better understand why it can achieve good performance. We will also work on using PSO to construct multiple high-level features to see whether constructing multiple features can further improve the classification performance and whether using only the constructed features can achieve better performance than using both the constructed features and all the original features, which also achieve the goal of dimension reduction. PSO is directly used for binary classification in this work. We intend to work on the use of PSO to address multiple classification tasks. Evolving function operators is still difficult for PSO. Therefore, we will work on using PSO to evolve different function operators (like in GP) to construct high-level features and more function operators will be used for constructing new features. The evolutionary training efficiency is also an important issue. We intend to improve the efficiency of the PSO based feature construction algorithm without decreasing its classification performance.

7. REFERENCES

[1] Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*, 30:767 – 783, 2004.

[2] T. J. Ai and V. Kachitvichyanukul. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers and Industrial Engineering*, 56(1):380–387, 2009.

[3] M. AlRashidi and M. El-Hawary. A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation*, 13(4):913–918, 2009.

[4] L. Cervante, B. Xue, M. Zhang, and L. Shang. Binary particle swarm optimisation for feature selection: A filter based approach. In *IEEE Congress on Evolutionary Computation (CEC'2012)*, pages 881–888, 2012.

[5] R. Eberhart and Y. Shi. Evolving artificial neural networks. In *Proceedings of the International Conference on Neural Networks and Brain*, pages 27–30, 1998.

[6] A. P. Engelbrecht. *Computational intelligence: an introduction (2. ed.)*. Wiley, 2007.

[7] I. D. Falco, A. D. Cioppa, and E. Tarantino. Facing classification problems with particle swarm optimization. *Applied Soft Computing*, 7(3):652 – 658, 2007.

[8] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[9] Y. Guo, W. Li, A. Mileham, and G. Owen. Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, 25(2):280–288, 2009.

[10] N. Holden and A. Freitas. A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data. In *Swarm Intelligence Symposium*, number 1-8, pages 100 – 107, 2005.

[11] J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[12] J. Kennedy and R. Eberhart. A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation.*, volume 5, pages 4104–4108, 1997.

[13] K. Krawiec. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*, 3(4):329–343, 2002.

[14] H. Liu and H. Motada, editors. *Feature extraction, construction and selection: A data mining perspective*. Kluwer Academic Publishers, Norwell, MA, 1998.

[15] I. Mierswa and M. Wurst. Efficient case based feature construction. In *European Conference on Machine Learning (ECML)*, pages 641–648. Springer Berlin / Heidelberg, 2005.

[16] M. A. Muharram and G. D. Smith. The effect of evolved attributes on classification algorithms. In *Australian Conference on Artificial Intelligence (AI'03)*, volume 2903 of *Lecture Notes in Computer Science*, pages 933–941. Springer, 2003.

[17] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[18] K. Neshatian and M. Zhang. Genetic programming for performance improvement and dimensionality reduction of classification problems. In *IEEE Congress on Evolutionary Computation (CEC'08)*, pages 2811–2818, 2008.

[19] K. Neshatian, M. Zhang, and P. Andreae. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Transactions on Evolutionary Computation*, 16(5):645 –661, 2012.

[20] K. Neshatian, M. Zhang, and M. Johnston. Feature construction and dimension reduction using genetic programming. In *Australian Conference on Artificial Intelligence (AI'07)*, volume 4830 of *Lecture Notes in Computer Science*, pages 160–170. Springer, 2007.

[21] F. Otero, M. Silva, A. Freitas, and J. Nievola. Genetic programming for attribute construction in data mining. In *Genetic Programming*, volume 2610 of *Lecture Notes in Computer Science*, pages 101–121. Springer Berlin/Heidelberg, 2003.

[22] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation (CEC'98)*, pages 69–73, 1998.

[23] M. G. Smith and L. Bull. Genetic programming with a genetic algorithm for feature construction and selection. *Genetic Programming and Evolvable Machines*, 6(3):265–281, 2005.

[24] Y. Tan, C. Deng, and G. Ruan. Concentration based feature construction approach for spam detection. In *International Joint Conference on Neural Networks (IJCNN 2009)*, pages 3088 –3093, 2009.

[25] H. Vafaie and K. DeJong. Feature space transformation using genetic algorithms. *IEEE Intelligent Systems*, 13(2):57–65, 1998.

[26] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang. A multi-objective particle swarm optimisation for filter based feature selection in classification problems. *Connection Science*, 24(2-3):91–116, 2012.

[27] B. Xue, M. Zhang, and W. Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, (99):1–16, 2012.

[28] B. Xue, M. Zhang, and W. N. Browne. Multi-objective particle swarm optimisation (pso) for feature selection. In *Genetic and Evolutionary Computation Conference (GECCO '12)*, pages 81–88. ACM, 2012.

[29] B. Xue, M. Zhang, and W. N. Browne. New fitness functions in binary particle swarm optimisation for feature selection. In *IEEE Congress on Evolutionary Computation (CEC'2012)*, pages 2145–2152, 2012.

[30] Y. Zhang, S. Wang, and L. Wu. A novel method for magnetic resonance brain image classification based on adaptive chaotic pso. *Progress In Electromagnetics Research*, 109:325–343, 2010.