New Representations in PSO for Feature Construction in Classification

Yan Dai, Bing Xue(✉), and Mengjie Zhang

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

Abstract. Feature construction can improve the classification performance by constructing high-level features using the original low-level features and function operators. Particle swarm optimisation (PSO) is an powerful global search technique, but it cannot be directly used for feature construction because of its representation scheme. This paper proposes two new representations, pair representation and array representation, which allow PSO to direct evolve function operators. Two PSO based feature construction algorithms (PSOFCPair and PSOFCArray) are then developed. The two new algorithms are examined and compared with the first PSO based feature construction algorithm (PSOFC), which employs an inner loop to select function operators. Experimental results show that both PSOFCPair and PSOFCArray can increase the classification performance by constructing a new high-level feature. PSOFCArray outperforms PSOFCPair and achieves similar results to PSOFC, but uses significantly shorter computational time. This paper represents the first work on using PSO to directly evolve function operators for feature construction.

Keywords: Particle swarm optimisation · Feature construction · Classification

1 Introduction

In classification, the quality of the data that is defined by a set of features is an important factor. A classification algorithm usually can not achieve good classification performance using the original feature set. Therefore, feature manipulation techniques are proposed to improve the quality of the feature space, two of which are feature selection and feature construction [11]. Feature selection is to select a subset of original features to reduce the dimensionality and improve the classification performance [7]. Feature construction is a means of enhancing the quality of feature space by constructing new high-level features [6,7]. The constructed feature(s) should be able to discover the hidden relationship between the original low-level features, which is particularly useful when the original features could not provide enough information for classification. This work will mainly focus on feature construction for classification.

© Springer-Verlag Berlin Heidelberg 2014
A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2014, LNCS 8602, pp. 476–488, 2014.
DOI: 10.1007/978-3-662-45523-4_39

A constructed feature is usually a function of original low-level features and mathematical operators. Therefore, the selection of the original features and function operators is the key issue in feature construction, but it is a difficult problem due mainly to the large search space. The size of the search space grows exponentially with the number of original features and the candidate operators. As a result, feature construction approaches often suffer from the problem of being stagnation in local optima and computationally expensive. Therefore, a global search technique is needed to develop an effective and efficient feature construction algorithm.

Evolutionary computation (EC) techniques are a group of powerful arguably global search algorithms, which have been successfully applied to many areas [3]. Most of the EC based feature construction approaches rely on genetic programming (GP) due to its tree-like representation [6,9,10]. Particle swarm optimisation (PSO) is a powerful EC technique and is argued to be computationally less expensive than GP [3]. PSO has been used for feature selection [3,14,16], but there is only one work successfully using PSO for feature construction [17]. However, since the original representation in PSO does not allow it to evolve nominal values, the function operators in [17] are selected by a time-consuming inner loop rather than evolved by PSO itself. Therefore, in order to further investigate the use of PSO for feature construction, a new representation scheme needs to be developed to allow PSO itself to select function operators during the evolutionary process.

1.1 Goals

The overall goal of this paper is to propose a new representation scheme in PSO to develop a PSO based feature construction approach to binary classification. To achieve this goal, we develop two new representations named pair representation and array representation, based on which two PSO based feature construction algorithms are developed. We expect each new algorithm to construct a single high-level feature, which can benefit the classification performance either being used solely or combined with the original features. The two proposed algorithms are examined and compared with the first PSO based feature construction approach (PSOFC) [17] on seven commonly used binary classification problems. Specifically, we will investigate:

- whether PSO using the pair representation can automatically construct a new high-level feature to improve the classification performance either by the new feature itself or combined with the original features;
- whether PSO using the array representation can successfully construct a new high-level feature to improve the classification performance and outperforms the pair representation; and
- whether the two new algorithms can use a shorter computational time to achieve similar classification performance to PSOFC.

2 Background

2.1 Particle Swarm Optimisation (PSO)

PSO stimulates social behaviours of birds flocking and fish schooling [5,13]. In PSO, each candidate solution is encoded as a particle. A PSO algorithm starts with randomly initialising a population or swarm of particles. During the evolution of PSO, all the particles move or "fly" 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 $v_i = (v_{i1}, v_{i2}, ..., v_{iD},)$ represents its velocity, where D is the dimensionality of the search space. During the search process, each particle can remember its best position visited so far called personal best (denoted by *pbest*), and the best previous position visited so far by the whole swarm called global best (denoted by *gbest*). Based on *pbest* and *gbest*, PSO iteratively updates the x_i and v_i of particle i to search for the optimal solutions according to Equations 1 and 2.

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

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

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 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 $v_{id}^{t+1} \in [-v_{max}, v_{max}]$.

2.2 Related Work on Feature Construction

Feature construction has a long research history and a large number of feature construction approaches have been developed [7]. Based on whether a classification algorithm is included in the evaluation procedure, existing feature construction methods can be broadly divided into two categories, which are wrapper approaches and filter approaches [7]. In wrapper approaches, a classification algorithm is used to evaluate the classification performance of the constructed features. A filter feature construction process is a separate, independent preprocessing stage and the new features are constructed before the classification algorithm is applied to build the classifier [6]. Different filter and wrapper feature construction methods have been developed and more details can be seen in [6,7,11]. Due to the page limit, this section will briefly review typical evolutionary feature construction approaches only.

In evolutionary approaches to feature construction, most of the work relies on GP due to its tree-based representation, which can naturally evolve functions of features and mathematical expressions [6]. Muharram and Smith [9] developed two fitness functions in GP for feature construction, which are based on information gain and gini index, respectively. Experimental results show that the classification performance of four different classification algorithms can be be improved by using the constructed features. Krawiec [6] extends the standard

GP for feature construction framework aiming to preserve the valuable components in GP individuals, which may be destructed by mutation or crossover operators. Neshatian at al. [12] develop a GP based filter feature construction algorithm, where the class dispersion and entropy are used to form the fitness function. Experiments show that these algorithms can improve the classification performance by constructing new high-level features. Later, Neshatian at al. [10] develop a GP based filter system to construct multiple high-level features. New features are constructed by GP with an entropy-based fitness function to maximise the purity of class intervals. Constructing multiple features is achieved by using a decomposable objective function. The experiments show that the constructed features can significantly increase the classification performance.

2.3 PSO for Feature Manipulation

PSO has been used to solve problems in many areas [3,14–16]. In terms of feature manipulation, PSO has been successfully used for feature selection, but there is only one existing work on PSO for feature construction [17]. Typical PSO based manipulation methods will be reviewed in this section.

Marinakis et al. [8] propose a wrapper feature selection approach based on PSO and K-nearest neighbour (KNN) for a real-world medical diagnosis problem called Pap-smear cell classification. The results show that this method removes around half of the features and achieves good classification performance. Azevedo et al. [1] proposed a wrapper feature selection algorithm using PSO and support vector machine (SVM) for personal identification in a keystroke dynamic system. However, the proposed algorithm obtained a relatively high false acceptance rate, which should be low in most identification systems. Unler and Murat [14] develop a modified PSO algorithm for feature selection. In the proposed algorithm, whether a feature is chosen or not depends on two criteria, which are the *likelihood* calculated by PSO and the relevance of the feature to the already selected features. The experiments show that the proposed algorithm achieves better performance than scatter search and tabu search algorithms. Xue et a. [16] proposed a PSO based multi-objective feature selection approach. Experimental results show that the proposed approach outperforms other three well-known EC based multi-objective feature selection algorithms.

Existing works have shown that PSO can be successfully used for feature selection. However, there is only one existing work to investigate the use of PSO for feature construction [17]. Xue et al. [17] apply PSO to feature construction (PSOFC) to construct a high-level feature, where PSO is used to select original features and a inner loop is used to exhaustively evaluate all the candidate operators to search for an operator for each of the selected features. The experiments have shown that PSOFC can successfully construct a high-level feature to improve the classification performance of three different classification algorithms, i.e. KNN, decision trees (DT), and naïve bayes (NB). However, the operators are not evolved by PSO itself, but selected by the inner loop, which is computationally expensive, especially when the number of features is large. This is due mainly to the major limitation of PSO in feature construction, i.e.

the standard representation does not allow PSO to evolve function operators. Therefore, a new representation scheme is needed in PSO to evolve function operators to further investigate its potential in feature construction.

3 Proposed Approaches

In order to address the major problem in PSO for feature construction, we propose two new representations, which are the pair representation and the array representation. These two new representations allow PSO to directly evolve function operators for feature construction.

3.1 Pair Representation

In this representation, the position shows a candidate solution of the problem, i.e. a constructed feature. The dimensionality of each particle/search space is n, where n is the total number of features in the dataset. Different from the traditional representation in PSO, the meaning/function of each dimension in the pair representation is two-folded. The first one is the probability of a feature being selected and the second one is the operator chosen for this feature if it is selected. By using the pair representation, a PSO based feature construction algorithm is proposed and named PSOFCPair.

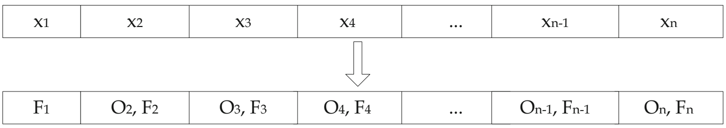

Fig. 1. Pair Representation

Fig. 1 shows a particle in the pair representation. x_i is the value of a particle in the ith dimension with $i \in [1, n]$. F_i represents feature i and O_i represents the operator for feature i. $x_i \in [0, 1]$ represents the probability of F_i being selected. A position determines the selected features and operators, which is regarded as a constructed feature. The selected features and operators are read from left to right and used as input to the feature construction function. The function starts with the first selected feature, followed by a number of pairs of an operator and a selected feature, and ends with the last selected feature. For example, a constructed feature can $F = F_1 * F_3 + F_5 - F_{10}$. Since there is no need to put any operator before the first selected feature, x_1 in the position only determines whether F_1 is selected or not. Note that the order of features in the dataset will not significantly effect the performance of the constructed feature because PSO is expected to automatically evolve the solutions during the evolutionary process and overcome the influence of the features being ordered.

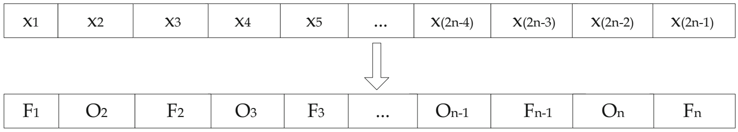

Fig. 2. Array Representation

To determine whether a feature is selected or not, a threshold $\theta \in [0, 1]$ is used here. If $x_i > \theta$, F_i is selected. Otherwise, F_i is not selected. If F_i is selected, an operator is needed to select for F_i according to the value of x_i. Given F_i being selected, $\theta < x_i \leq 1$. According to the number of candidate operators, the interval of $[\theta, 1]$ can be divided into a number of sub intervals. The operator is selected according to which sub interval x_i belongs to. For example, if there are four candidate operators, three numbers $(\alpha_1, \alpha_2, \alpha_3)$ can be used here to divide $[\theta, 1]$ into three sub intervals. If $\theta < x_i < \alpha_1$, the first operator is selected. If $\alpha_1 \leq x_i < \alpha_2$, the second operator is selected. If $\alpha_2 \leq x_i < \alpha_3$, the third operator is selected. If $\alpha_3 \leq x_i \leq 1$, the fourth operator is selected.

3.2 Array Representation

The pair representation could allow PSO to be directly used for feature construction without increasing the dimensionality of the search space, but using one variable to determine the selection of both features and operators may limit the search of the their best combination. Therefore, we propose an array representation, where the feature selection and operator selection are determined separately. By using the pair representation, a PSO based feature construction algorithm is proposed and named PSOFCArry.

Fig. 2 shows the position of a particle in the proposed array representation. The dimensionality of the particle is $2n - 1$, where n is the total number of features in the dataset. A dimension is used to determine the selection of either the feature or the operator. The $(2 * i - 1)$th dimension determines whether F_i is selected or not, where $i \in [1, n]$. The $(2 * i - 2)$th dimension determines which operator is selected for F_i, where $i \in [2, n]$ since the first feature does not need any operator. Meanwhile, the operator i is selected only when F_i is selected.

The threshold θ is also used in the $(2*i-1)$th dimension to determine whether F_i is selected or not. θ performs the same way as in the pair representation. According to the number of candidate operators, the interval [0,1] is divided into a number of sub intervals. An operators is selected according to which sub interval x_i in the $(2 * i - 2)$th dimension belongs to, which is the same as in the pair representation.

3.3 Pesuode Code of the Proposed Approaches

Both PSOFCPair and PSOFCArry follow the basic steps in PSO and each of them produces a single high-level feature. An important step in PSOFCPair and

Algorithm 1. Pseudo-code of PSOFCArry and PSOFCPair

```
begin
    split the instances into a Training and a Test set;
    initialise x and v of each particle;
    while Maximum Iterations has been not met do
        construct a new high-level feature for each particle according to the Pair
        or Array representation;
        calculate the classification performance of the constructed high-level
        feature;
        for i=1 to Swarm Size do
            update the personal best (pbest) of particle i;
            update the global best (gbest) of particle i;
        for i=1 to Swarm Size do
            for d=1 to Dimensionality do
                calculate v_i according to Equation 2
                calculate x_i according to Equation 1
    calculate the classification performance of the constructed feature on the
    test set using 0 as the threshold or using other classification algorithms;
    return gbest, the training and testing classification performance.
```

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
Sonar	60	2	208
Hillvalley	100	2	606
Musk1	166	2	476
Madelon	500	2	4400

PSOFCArry is the evaluation of a particle, which is shown in Line 1. In both PSOFCPair and PSOFCArry, the algorithm first constructs a new high-level feature according to the low-level features and the operators selected by the particle. The fitness of the particle is evaluated by the classification performance of the newly constructed high-level feature. Since binary classification problems are considered here, we use 0 as the threshold for the constructed feature to determine an instance to be class 1 or class 2. The purpose of using 0 as the threshold for classification rather than using a classification algorithm is to speed up the classification (i.e. the fitness evaluation) process by avoiding a complex process to train a classifier.

4 Design of Experiments

A set of experiments have been conducted to examine the performance of PSOFC-Pair and PSOFCArry on seven binary datasets (see Table 1) chosen from the UCI machine learning repository [4]. The seven datasets are chosen to have different numbers of features and instances. On each dataset, 70% of the instances

Table 2. Operator Selection

PSOFCPair		PSOFCArry	
Interval	Operator	Interval	Operator
[0.5, 0.625)	+	[0.0, 0.25)	+
[0.625, 0.7)	-	[0.25, 0.5)	-
[0.7, 0.825)	*	[0.5, 0.75)	*
[0.825, 1]	/	[0.75, 1]	/

are randomly selected as training examples and the rest 30% are used as the testing set, following the settings in [17] to make a fair comparison.

The parameters in PSOFCPair and PSOFCArry are set as follows [2]: $w = 0.7298$, $c_1 = c_2 = 1.49618$. The swarm size is 30 and the fully connected topology is used. The maximum number of iterations is 100. θ in both PSOFCPair and PSOFCArry is set as 0.5, which means each original feature has 50% probability to be selected for constructing the new high-level feature. Four commonly used function operators in GP for feature construction [11] are used in both PSOFCPair and PSOFCArry, which are "+", "-", "*" and "/" (protected division). The operators are selected according to which interval the corresponding position value falls into and details can be seen in Table 2. The four operators are considered equally important. Therefore, the four intervals in PSOFCPair or PSOFCArry have the same range to ensure that the four operators have the same probability to be selected.

Both PSOFCPair and PSOFCArry are run 50 independent times on each dataset. To test the generality of the constructed feature, three different learning algorithms are used to test its classification performance on the testing set. The three classifiers are DT, KNN with K = 5 and NB. To further test the performance of PSOFCPair and PSOFCArry, they are compared with the first and the only existing PSO based feature construction algorithm (PSOFC) [17].

5 Results and Discussions

The results of PSOFCPair and PSOFCArry are shown in Tables 3 and 4. In the tables, "Org" means all the original features are used for the classification. "CF" means only the single constructed feature is used for the classification. "OrgCF" means the constructed feature and the original features are combined together for classification. "#Fea" represents the total number of features in the datasets. "Best", "Avg" and "Std" represent the best, the average and the standard deviation of the testing classification performance obtained from the 50 runs.

5.1 Results of the PSOFCPair

As can be seen from Table 3, by using only the single constructed feature for classification, DT, KNN and NB can achieve similar or better classification performance than using all the original features on a few datasets only. The results suggest that the simple pair representation in PSO has potential to construct

Table 3. Result of PSOFCPair

Dataset	#Fea	Method	DT			KNN			NB		
			Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
Australian	14	Org	85.99			70.05			85.51		
		CF	77.29	65	4.13	74.4	61.99	4.28	59.9	53.93	1.84
		OrgCF	85.99	85.99	0	74.88	69.05	3.01	85.99	85.34	45.1E-2
WBCD	30	Org	92.98			92.98			90.64		
		CF	95.32	86.5	8.66	95.32	85.94	9.06	61.99	61.41	8.26E-2
		OrgCF	97.08	93.2	92.1E-2	95.91	92.13	4.12	90.64	90.64	0
Ionosphere	34	Org	86.67			83.81			28.57		
		CF	84.76	75.47	5.65	84.76	73.45	5.66	84.76	80.86	2.36
		OrgCF	89.52	86.88	93.8E-2	89.52	84.7	1.57	28.57	28.57	0
Sonar	60	Org	71.43			76.19			53.97		
		CF	69.84	53.14	7.7	65.08	53.05	7.23	47.62	47.62	0
		OrgCF	73.02	71.11	1.42	79.37	66.25	11.9	53.97	53.97	0
Musk1	166	Org	71.33			83.92			42.66		
		CF	67.13	58.95	4.17	64.34	55.37	5.09	60.14	59.38	36.6E-2
		OrgCF	75.52	71.41	58.7E-2	85.31	62.69	13.7	72.73	72.73	0
Hillvalley	100	Org	62.09			56.59			52.2		
		CF	83.79	54.56	7.99	83.52	53.38	7.84	47.8	47.8	0
		OrgCF	85.99	63.47	5.84	83.79	53.56	7.95	52.2	52.2	0
Madelon	500	Org	76.79			70.9			49.49		
		CF	57.31	50.98	2.34	54.23	50.27	1.69	49.49	49.49	0
		OrgCF	77.69	76.79	16E-2	72.44	52.24	5.85	55.51	55.51	1.84E-2

a high-level feature to provide useful information for classification and using only the single constructed needs much less computational time than using the original full set of features. However, the limitation of PSOFCPair is that a feature and its operator share the same value to determine whether the feature is selected or not and which operator is chosen. During the evolution, the shared dimension may not reach the ideal value for both feature and operator selection. Therefore, only using the constructed feature could not improve the classification performance on most cases, but adding it to the original feature set may increase the classification accuracy.

According to Table 3, by adding the constructed feature to the original feature set, the classification performance of all the three classification algorithms (DT, KNN and NB) can be increased. Specifically, the average accuracy of DT is increased on four of the seven datasets and similar on the other three datasets. The best accuracy is higher than using only the original features on six of the seven datasets and the same on one dataset. The performance of using both the constructed feature and the original features on KNN and NB shows a similar pattern to DT, where the classification performance is increased in most cases. These results indicate that adding the constructed feature can provide useful information to the feature set to achieve better classification performance than using only the original features, but the computational time cost by adding only one feature can be safely ignored. Although there is a preprocessing step to constructed the new feature, its computation time is very short (details can be seen in Section 5.3).

5.2 Results of PSOFCArry

According to Table 4, it can be seen that when using only the single constructed high-level feature for classification, the best classification performance of DT is

Table 4. Results of PSOFCArry

Dataset	#Fea	Method	DT			KNN			NB		
			Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
Australian	14	Org	85.99			70.05			85.51		
		CF	88.41	85.05	1.63	87.44	66.59	17.3	76.33	55.21	5.02
		OrgCF	87.92	85.87	98.8E-2	87.92	79.54	4.17	88.89	86.59	67.4E-2
WBCD	30	Org	92.98			92.98			90.64		
		CF	95.91	91.47	2.36	95.91	90.98	2.49	61.4	61.4	0
		OrgCF	97.08	93.45	1.07	95.91	92.65	1.5	90.64	90.64	0
Ionosphere	34	Org	86.67			83.81			28.57		
		CF	83.81	76.71	4.79	85.71	76.32	5.07	87.62	82.32	1.56
		OrgCF	92.38	85.96	3.27	87.62	84.46	1.18	28.57	28.57	0
Sonar	60	Org	71.43			76.19			53.97		
		CF	73.02	63.33	6.14	74.6	61.27	5.45	47.62	47.62	0
		OrgCF	76.19	68.67	4.17	80.95	71.08	6.42	53.97	53.97	0
Musk1	166	Org	71.33			83.92			42.66		
		CF	67.13	58.77	5.29	67.13	57.86	4.39	60.14	59.36	43.5E-2
		OrgCF	73.43	71.32	58.6E-2	84.62	66.99	12.7	72.73	72.73	0
Hillvalley	100	Org	62.09			56.59			52.2		
		CF	99.45	96.98	1.71	99.45	96.87	1.85	50	47.86	31.6E-2
		OrgCF	99.45	97.15	1.51	76.92	62.86	5.34	52.47	52.21	3.81E-2
Madelon	500	Org	76.79			70.9			49.49		
		CF	64.36	57.28	4.16	58.33	53.33	2.34	49.49	49.49	0
		OrgCF	77.05	76.77	24.6E-2	70.9	66.4	8.08	49.49	49.49	0

better than when using all the original features on four of the seven datasets. For example, on the Hillvalley dataset, the classification performance of DT using all the 100 original features is 62.09%. By using only the single constructed features, DT achieved the average classification performance of 96.98% and the best accuracy of 99.45%. The best classification performance of KNN and NB using only the constructed feature is better than using all the original low-level features on most datasets. The results suggests that PSOFCArry can effectively evolve a number of original low-level features and function operators to construct a single high-level feature, which is possible to achieve better classification performance than using all the original features.

According to Table 4, it can be observed that when combing the single constructed feature with the original features, the best classification performance of DT is better than using only the original features on all the seven datasets. The average classification accuracy is similar or better than using only the original features on almost all datasets. KNN and NB shows a similar patter to DT, which is the average accuracy is better or similar on most datasets and the best accuracy is higher than using only the original features on most cases. The results suggest that adding the constructed feature to the original features brings useful information to the feature set, which can help a classification algorithm (DT, KNN or NB) to achieve better classification performance than using only the original features.

5.3 Further Comparisons

Table 5 shows of the classification performance of PSOFCPair, PSOFCArray and PSOFC using DT as the classification algorithm, where“CF” means DT using only the constructed high-level feature and “CFOrg” means the combination of

Table 5. Results of PSOFCPair,PSOFCArray and PSOFC using DT

Feature	Method	Australian		WBCD		Ionosphere		Sonar	
		Ave±Std	Test	Ave±Std	Test	Ave±Std	Test	Ave±Std	Test
CF	PSOFC	85.35±1.13		93.31±1.07		81.01±2.85		63.11±2.26	
	PSOFCPair	65±4.13	-	86.5±8.66	-	75.47±5.65	-	53.14±7.7	-
	PSOFCArray	85.05±1.63	=	91.47±2.36	-	76.71±4.79	-	63.33±6.14	=
	Method	Musk1		Hillvalley		Madelon			
		Ave±Std	Test	Ave±Std	Test	Ave±Std	Test		
	PSOFC	65.96±3.24		85.93±12.5		53.97±4.26E-14			
	PSOFCPair	58.95±4.17	-	54.56±7.99	-	50.98±2.34	-		
	PSOFCArray	58.77±5.29	-	96.98±1.71	+	57.28±4.16	+		

Feature	Method	Australian		WBCD		Ionosphere		Sonar	
		Ave±Std	Test	Ave±Std	Test	Ave±Std	Test	Ave±Std	Test
CFOrg	PSOFC	85.93±66E-2		93.81±1.1		86.55±3.13		71.43±2.8E-14	
	PSOFCPair	85.99±9.9E-14	=	93.2±92.1E-2	-	86.88±93.8E-2	=	71.11±1.42	=
	PSOFCArray	85.87±98.8E-2	=	93.45±1.07	=	85.96±3.27	=	68.67±4.17	-
	Method	Musk1		Hillvalley		Madelon			
		Ave±Std	Test	Ave±Std	Test	Ave±Std	Test		
	PSOFC	71.54±3.09		85.63±12.4		76.79±8.53E-14			
	PSOFCPair	71.41±58.7E-2	=	63.47±5.84	-	76.79±16E-2	=		
	PSOFCArray	71.32±58.5E-2	=	97.15±1.51	+	76.77±24.6E-2	=		

the constructed feature and original features. "Test" shows the results of the statistical significant T-test (Z-test) comparing the classification performance achieved by PSOFC and PSOFCPair(or PSOFCArray). The results of using KNN or NB as the classification algorithm show a similar patter to DT and the results are not listed here due to the page limit. The average computational time (in seconds) of the three algorithms in each run is shown in Table 6.

According to Table 5, when DT using only the constructed feature for classification, PSOFC achieved better performance than PSOFCPair in all cases, better than PSOFCArray in three cases and worse than PSOFCArray in two cases. When using the combination of the constructed feature and the original features, PSOFC achieved slightly better performance than PSOFCPair and similar performance to PSOFCArray. The main reason is that PSOFC using a inner loop for operator selection, which conducts an exhaustive search of all the candidates operators to find the optimal operator for each feature, can obtain a better set of operators. PSOFCPair has a potential limitation due to the use of one dimension for both features and operators. However, the inner loop in PSOFC is time-consuming. From Table 6, it can be observed that the time used by PSOFCPair and PSOFCArray is around 100 times shorter than that of PSOFC. The main reason is that the inner loop in PSOFC causes a much larger number of evaluations than PSOFCPair and PSOFCArray. The operators in PSOFCPair and PSOFCArray are evolved by PSO itself and not need extra calculations. Since the dimensionality of PSOFCArray is higher than PSOFCPair, the computational time used by PSOFCArray is slightly larger than PSOFCPair, but still around 100 times shorter than PSOFC.

Tables 5 and 6 suggest that the new representations in PSOFCPair and PSOFCArray can effectively evolve operators to construct a new high-level feature to achieve similar classification performance to PSOFC, but use significantly shorter computational time.

Table 6. Computation Time used by PSOFCPair,PSOFCArray and PSOFC

Method	Australian	Ionosphere	WBCD	Hillvalley	Musk1	Semeion	Madelon
PSOFCPair	94.6E-2	84.2E-2	54.4E-2	41.6E-2	1.46	2.68	18.8
PSOFCArray	93.3E-2	88E-2	59E-2	44.5E-2	1.93	3.45	25.6
PSOFC	31.4	65.4	47.1	65.6	7.2E2	8.2E2	6.1E4

6 Conclusion and Future Work

The goal of this research was to develop a new representation scheme in PSO for feature construction to construct a high-level feature to improve the classification performance. The goal was successfully achieved by proposing two new representations, which are the pair representation (PSOFCPair) and the array representation (PSOFCArray). PSOFCPair and PSOFCArray were examined and compared with the first and only existing PSO based feature construction algorithm (PSOFC) on seven benchmark datasets. The experimental results show that PSOFCPair increased the classification performance in most cases by adding the constructed feature to the original feature set, but it has a limitation because of using one dimension in the particle for both the feature selection and operator selection. By using a larger dimensionality, PSOFCArray could increase the classification performance by using only the constructed feature and increase the classification performance in almost all cases by adding the constructed feature to the original feature set. PSOFCArray achieved similar classification performance to PSOFC, but used significantly shorter computational time.

This paper is the first work that uses PSO to automatically select original low-level features and function operators for feature construction. In the future, we will further investigate the use of PSO for feature construction and compare its performance with GP based feature construction approaches.

References

1. 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)
2. Clerc, M., Kennedy, J.: The particle swarm- explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation **6**(1), 58–73 (2002)
3. Engelbrecht, A.P.: Computational intelligence: an introduction, 2. ed. Wiley (2007)
4. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. IEEE International Conference on Neural Networks. **4**, 1942–1948 (1995)
6. Krawiec, K.: Genetic programming-based construction of features for machine learning and knowledge discovery tasks. Genetic Programming and Evolvable Machines **3**(4), 329–343 (2002)
7. Liu, H., Motada, H. (eds.): Feature extraction, construction and selection: A data mining perspective. Kluwer Academic Publishers, Norwell (1998)
8. Marinakis, Y., Marinaki, M., Dounias, G.: Particle swarm optimization for pap-smear diagnosis. Expert Systems with Applications **35**(4), 1645–1656 (2008)

9. Muharram, M.A., Smith, G.D.: Evolutionary Feature Construction Using Information Gain and Gini Index. In: Keijzer, M., O'Reilly, U.-M., Lucas, S., Costa, E., Soule, T. (eds.) EuroGP 2004. LNCS, vol. 3003, pp. 379–388. Springer, Heidelberg (2004)
10. Neshatian, K., Zhang, M., Andreae, P.: A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. IEEE Transactions on Evolutionary Computation **16**(5), 645–661 (2012)
11. Neshatian, K.: Feature Manipulation with Genetic Programming. PhD thesis, Victoria University of Wellington, Wellington, New Zealand (2010)
12. Neshatian, K., Zhang, M., Johnston, M.: Feature Construction and Dimension Reduction Using Genetic Programming. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 160–170. Springer, Heidelberg (2007)
13. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation (CEC 1998), pp. 69–73 (1998)
14. 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)
15. Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: A multi-objective particle swarm optimisation for filter based feature selection in classification problems. Connection Science (2012)
16. Xue, B., Zhang, M., Browne, W.: Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics **43**(6), 1656–1671 (2013)
17. Xue, B., Zhang, M., Dai, Y., Browne, W.N.: PSO for feature construction and binary classification. In: Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference, GECCO 2013, 137–144 (2013)