Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 3852 – 3861
© TÜBİTAK
doi:10.3906/elk-1404-220

# A hybrid search method of wrapper feature selection by chaos particle swarm optimization and local search

**Mohammad Masoud JAVIDI***, **Nasibeh EMAMI**
Department of Computer Science, Shahid Bahonar University, Kerman, Iran

**Received:** 13.04.2014 • **Accepted/Published Online:** 18.06.2015 • **Final Version:** 20.06.2016

**Abstract:** Finding a subset of features from a large dataset is a problem that arises in many fields of study. Since the increasing number of features has extended the computational cost of a system, it is necessary to design and implement a system with the least number of features. The purpose of feature selection is to find the best subset of features from the original ones. The result of the best selection is improving the computational cost and the accuracy of the prediction. A large number of algorithms have been proposed for feature subset selection. In this paper, we propose a wrapper feature selection algorithm for a classification that is based on chaos theory, binary particle swarm optimization, and local search. In the proposed algorithm, the nearest neighbor algorithm is used for the evaluation phase.

**Key words:** Feature selection problem, metaheuristic approach, local search, binary particle swarm optimization, chaos theory

## 1. Introduction

Feature selection is an issue about selecting the best subset of features among the primary features, the ones that show the best performance in a classified accuracy [1]. Since finding the best feature subset is in exponential space, the feature selection problem is intractable or NP-hard [2]. In order to overcome the intractable property of the feature selection problem, good search algorithms are required. A good search algorithm should provide [3]: 1) good global search, 2) rapid convergence to a near optimal solution, 3) the ability for good local search, 4) and high computational efficiency.

Feature selection has many applications in pattern recognition [4], machine learning [5], data mining [6], statistics [7], image processing [8], and signal processing [9]. The objective of feature selection is to identify important features in the dataset and remove any other irrelevant features and redundant information [10]. Four main reasons for using feature selection as a preprocessing step in many applications are [11]: 1) reducing the computational cost, 2) removing the irrelevant or redundant features and therefore saving the cost of measurement of nonselected features, 3) improving accuracy, and 4) providing the identity of the selected features that supplied the best insights into the nature of the problem at hand.

There are different categories in various sources for feature selection algorithms. Generally

they fall into 2 categories [12]. The first are filters and the second are wrappers. The filter-based methods have 3 phases [13]: Phase 1, feature set generation that generates a feature subset; Phase 2, measurement of the feature set that measures the information of the current feature set; and Phase 3, testing of the feature set by a learning algorithm.

*Correspondence: javidi@uk.ac.ir

Phase 1 and 2 run repeatedly while the result does not satisfy the stop criterion. The stop criterion could be a threshold of the measurement results. When a feature set reaches the threshold, filter methods enter Phase 3 and the goodness of the feature set is tested by a learning algorithm like support vector machine (SVM), K-nearest neighbors (KNN), neural network (NN), and so on. The phases of feature selection in wrapper methods are similar to filter methods except that Phase 2 is replaced by a learning algorithm.

The filters are fast due to using a simple measurement, but its result is not always satisfactory. The wrappers have good performance due to use of a learning algorithm [13], but they are very slow compared to filter methods [14].

In this paper, we introduce a combinational search method that can balance between discovering the new region and the local search. It tries to select a subset by a more accurate and low cardinality.

The rest of this paper is organized as follow: Section 2 is about a review of existing techniques in the area of feature selection. Preliminaries of the proposed method are given in Section 3, the proposed method is explained in Section 4, implementation and the results are provided in Section 5, and finally conclusions are given in Section 6.

## 2. A review of existing techniques

The algorithm used for feature selection depends on the cardinality of the original feature set. The feature selection problem based on cardinality feature set falls into 3 categories [15]: cardinality between )0, 19( (small-scale), cardinality between )20, 49( (medium-scale), and cardinality $\geq$50 (large-scale). There are 2 groups of algorithms in each scale: wrapper and filter-based ones. Two famous wrapper-based methods that use a greedy search strategy are sequential forward selection (SFS) [16] and sequential backward selection (SBS) [17]. SFS starts with the empty set and adds the most rewarding features among the unselected ones in each iterative. SBS starts with all features and removes the least rewarding features among the selected ones in each iteration until the stop criterion is satisfied [18]. The main disadvantage of the SFS method is that when a feature is added to the feature subset it cannot be removed in the future, and another disadvantage of SBS is high computation cost since the criterion function evaluates larger sets of features [19]. In addition, both of them fall into local optima easily [3]. The stochastic search strategy has been developed for solving large-scale problems and the most famous of them are [3]: the genetic algorithm [20], ant colony optimization [21], particle swarm optimization [22], and simulated annealing [23]. These methods are stochastic optimization approaches that are inspired by nature and try to achieve better solutions by referencing the feedback and heuristic information [24]. The advantage of these algorithms is that they efficiently capture feature redundancy and the disadvantage of these algorithms is that they are computationally expensive while being a less exhaustive search [3]. Well-known filter methods include the t-test [25], chi-square test [26], Wilcoxon test [27], mutual information [28], Pearson correlation coefficients [29], and principal component analysis [30]. Filtering techniques are very efficient in selecting, but they are unstable when used on wide feature sets [13].

## 3. Preliminaries

In this section, we discuss the preliminaries of the search algorithm and of classifiers that will be used in this proposed method. These preliminary involve particle swarm optimization, binary particle swarm optimization, chaos theory, local search, and 1-nearest neighbor as classifiers.

### 3.1. Standard particle swarm optimization (PSO)

PSO is an optimization algorithm that is inspired by nature. Kennedy and Eberhart proposed it for the first time to solve the problem in continuous space [31]. It uses a number of particles that constitute a swarm. Each particle traverses the search space looking for the global minimum (or maximum) [32]. Each particle in the swarm has its own position, velocity, and best position represented by $x_k^t$, $v_k^t$, and $bestp_k^t$. In the optimization problem the goal is to find the global optimization result represented by Gbest (namely the global best position of all particles). Each iteration of the algorithm's position and velocity of each particle is updated by:

$$v_{kd}^{t+1} = w\times v_{kd}^{t}+c_1\times r_1\times \left(bestp_{kd}^{t}-x_{kd}^{t}\right)+c_2\times r_2\times \left(Gbest_{kd}^{t}-x_{kd}^{t}\right)_{\max} \tag{1}$$

$$x_{kd}^{t+1}=x_{kd}^{t}+v_{kd}^{t+1} \tag{2}$$

where $c_1$ and $c_2$ are positive constants, called acceleration coefficients, and $r_1$ and $r_2$ are two random numbers in the range of [0,1]. $W$ is the inertia weight that controls the influence of the previous velocity on the current velocity [33]. The velocity vector is kept within a predefined interval as in [$V_{\min}$, $V_{\max}$] [34].

### 3.2. Binary particle swarm optimization (BPSO)

For solving the binary problem Kennedy and Eberhart proposed a binary version of PSO

[35]. For solving the feature selection problem by BPSO, the particle's position in each dimension is 1 (selected feature) or 0 (nonselected feature). Initial velocities in particles are in the range of [0, 1] [36]. In this version of PSO, velocity of each particle is updated by:

$$v_{kd}^{t+1} = w\times v_{kd}^{t}+c_1\times r_1\times \left(bestp_{kd}^{t}-x_{kd}^{t}\right)+c_2\times r_2\times \left(Gbest_{kd}^{t}-x_{kd}^{t}\right) \tag{3}$$

After the velocity update, it is necessary to examine velocity to be in the range [$V_{\min}$, $V_{\max}$]. If it is not in the range we should use the following formula to place it in the range:

$$v_{kd}^{t+1}=\max\left(\min\left(V_{\max},v_{kd}^{t+1}\right),V_{\min}\right) \tag{4}$$

The position will then be updated by the following formula:

$$S\left(v_{kd}^{t+1}\right)=\frac{1}{1+e^{-v_{kd}^{t+1}}} \tag{5}$$

$$x_{kd}^{t+1}=\begin{cases} 1 & \text{if rand} < S\left(v_{kd}^{t+1}\right) \\ 0 & O.W \end{cases} \tag{6}$$

$S$ is a sigmoid function in terms of velocity.

### 3.3. BPSO with chaotic inertia weight

PSO and BPSO are sensitive to the parameters, and especially the inertia weight. Inertia weight can adjust the global search facility and the local search facility. Larger inertia weight means that PSO and BPSO tend to do a global search, while smaller inertia weight means that PSO and BPSO tend to do local searches [37].

The disadvantages of PSO and BPSO are being prematurely convergent and trapping into local minimum. To overcome these disadvantages, some improved measures are proposed, such as embedded crossover operation or use of a chaotic sequence [38]. Chaotic sequences have apparently complicated behavior and sounds that have random movement; they are nonlinear systems that are sensitive to initial conditions. Due to the nonrepetition of chaos, it can carry out overall searches at higher speeds than stochastic searches, which depend on probabilities. The application of chaotic sequences instead of random sequences in PSO is a powerful strategy to improve the PSO's performance in preventing premature convergence to local minima [39]. In [36], chaotic sequence by logistic maps was used to determine the inertia weight value in BPSO. Each iteration is used to prevent the premature convergence to local minima and to improve classification results. The chaotic sequence by logistic maps for changing the inertia weight is:

$$w\,(t+1) = 4 \times w\,(t) \times (1 - w\,(t))\, w\,(t) \in (0,1) \tag{7}$$

When the inertia weight value is close to 1, it means that BPSO tends to do the global search. When the inertia weight value is close to 0, it means that BPSO tends to do the local search.

In summary, BPSO with chaotic inertia weight is called chaotic binary particle swarm optimization (CBPSO).

## 4. Local search

The local search starts from an initial solution (in our case, the feature subset of end points to the global search) and attempts to achieve better and better solutions. For doing this task (finding better feature subsets), it evaluates the neighbor solution, and if it is a better solution than the current one it moves to the neighbor solution. It does that until the stop condition is satisfied. The local search often needs to start from an initialized solution, and therefore it is recommended to configure a construction heuristic solver phase before it.

In the proposed method we will use the local search to find a better feature subset. As we said, a local search algorithm is recommended to configure a construction heuristic solver phase before it. Thus, the initial solution in our local search is global, the best of BPSO, and then the local search does the following tasks:

- Best = global best
- Best fit = global best fit
- C = sum (global best) such that C is cardinality of global best
- Randomly select a number between [1, D] such that D is dimension of data
- New position = invert value of select position
- C' = sum (new position) such that C' is the cardinality of new position
- New position fit = evaluate fitness of new position by classifier.
- If New position fit >Best fit
  - Best = New position
  - Best fit = New position fit
    - Else if New position fit = Best fit & C' <C
    - Best = New position

### 4.1. K-nearest neighbors

K-nearest neighbors are familiar classifiers in data mining, machine learning, and pattern recognition that use distance metrics to predict those classes of instances that still are not seen. There are 3 key elements of this classifier [40]: 1) a set of labeled objects, 2) distance or similarity metrics, and 3) the number of nearest neighbors (value of K). In order to predicate a class of new instances, A, we first compute the distance between A and every object in the training set, and then K, the closest training object to A, is selected. The predicated class, for instance Z, is specified by majority voting on K-nearest neighbors.

## 5. Proposed method: combination of CBPSO and local search (CBPSOL) for feature subset selection

We present a hybrid algorithm (CBPSOL) for selecting optimal feature subsets efficiently. This algorithm is based on CBPSO and local search. The 1-nearest neighbor (1-NN) method with leave-one-out cross-validation as a classifier is used for evaluating classification accuracies [36]. CBPSO is a global search algorithm that is useful for exploration of new regions. The location of each particle is a binary string. If it has a value of one, it means to select the feature, and if it has a value of zero, it means the feature is not chosen. After a certain number of iterations of CBPSO, the local search algorithm starts to work from the point where CBPSO finished. The neighbor of the current solution in the local search algorithm differs in only one bit. We have used following innovative method: If the accuracy of the current solution neighborhood equals the accuracy of the current solution but has fewer number of features, we consider it as the current solution, so it has good local search ability and makes improvements to the solution. The combination of these two algorithms achieves high accuracy and reduces the number of features. The Figure shows all stages of the proposed method.

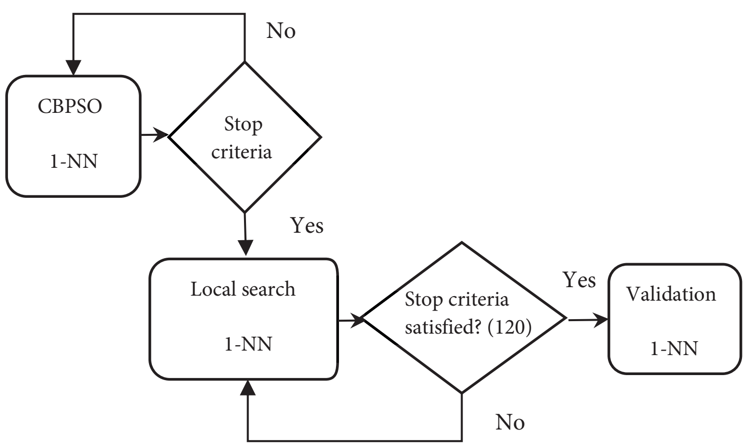


**Figure.** All stages of proposed algorithm.

## 6. Implementation and result evaluation

The proposed algorithm is implemented in MATLAB. This method is applied to 2 groups of 8 datasets for real and synthetic datasets, of low, medium, and high dimensionality. The 8 datasets that are real are in the UCI repository and the rest of them are synthetic datasets. In order to produce synthetic datasets, we make a random permutation of existing real datasets so the synthetic datasets have an equal size and range as the existing real datasets in this work. Table 1 shows the characteristics of the databases used. The initial setups for the proposed method are given in Table 2. For controlling the domain values of each feature, features are normal in the range of 0 and 1. The normalization formula is as follows:

**Table 1.** Datasets.

| No. | Datasets | Samples | Features | Classes |
|---|---|---|---|---|
| 1 | Wine | 178 | 113 | 3 |
| 2 | Hepatitis | 155 | 19 | 2 |
| 3 | Wisconsin Diagnostic Breast Cancer (WDBC) | 569 | 30 | 2 |
| 4 | Dermatology | 366 | 34 | 6 |
| 5 | Ionosphere | 351 | 33 | 2 |
| 6 | Spectf | 267 | 44 | 2 |
| 7 | Sonar | 208 | 60 | 2 |
| 8 | Musk | 476 | 166 | 2 |
| 9 | Synthetic 1 | 178 | 113 | 3 |
| 10 | Synthetic 2 | 155 | 19 | 2 |
| 11 | Synthetic 3 | 569 | 30 | 2 |
| 12 | Synthetic 4 | 366 | 34 | 6 |
| 13 | Synthetic 5 | 351 | 33 | 2 |
| 14 | Synthetic 6 | 267 | 44 | 2 |
| 15 | Synthetic 7 | 208 | 60 | 2 |
| 16 | Synthetic 8 | 476 | 166 | 2 |

**Table 2.** Initial setups.

| CBPSOL | | | | | | |
|---|---|---|---|---|---|---|
| CBPSO | | | | | Local search | |
| #Particles | W(0) | $C_1$ | $C_2$ | Termination criteria | Start point | Termination criteria |
| 20 | 0.86 | 1.49 | 1.49 | 80 iterations | gbest | 120 iterations |

$$x = \frac{x - \min_x}{\max_x - \min_x} \tag{8}$$

Table 3 shows the performance of the best and average results of the proposed method on a database presented. Because the proposed method is a method of a random search algorithm, it was run 10 times and the means of accuracy ??were considered as the final result. In order to evaluate the proposed method, this algorithm is compared with the proposed local search, BPSO, and CBPSO [35] methods. The parameters of BPSO and CBPSO are similar to CBPSOL. The only difference is that the weight is constant in BPSO; the inertia weight $w$for BPSO was fixed at 0.48 [36]. The termination criterion is t iterations, with t set to 200. The comparison results are given in Table 4. As Table 4 shows, the proposed algorithm is on average a set of better results in terms of classification accuracy for real and high-dimensional synthetic datasets. For the Wine dataset, BPSO, CBPSO, and CBPSOL produced equal accuracy of 99.44%. For the Dermatology dataset the average accuracy of the proposed method is the same as BPSO but the average number of features has increased. In the Spectf dataset the average accuracy of the proposed method is the same as BPSO but the average number of features has decreased. The rest of the real datasets show increasing classification accuracy coming with decreasing number of features. In the case of lower-dimensional synthetic datasets, CBPSO has a good performance in terms of accuracy and small number of features. There is a different result for the medium-dimensional synthetic datasets. For synthetic dataset 3 BPSO obtained the best accuracy, for synthetic datasets 4 and 6 CBPSO has the best accuracy, and CBPSOL obtained a good accuracy and minimum number of features for synthetic dataset 5. For all of the datasets, both real and synthetic, local search obtained the worst performance.

**Table 3.** Performance of the proposed method (CBPSOL).

| No. | Dataset | Without feature selection | | The results of the best answer (accuracy) | | The results of the mean answer (accuracy) | |
|---|---|---|---|---|---|---|---|
| | | #Features | Acc. (%) | #Features | Acc. (%) | #Features | Acc. (%) |
| 1 | Wine | 13 | 94.94 | 8 | 99.44 | 8 | 99.44 |
| 2 | Hepatitis | 19 | 59.3548 | 13 | 79.3548 | 11.9 | 77.4838 |
| 3 | WDBC | 30 | 95.25 | 14 | 98.24 | 15.4 | 97.80 |
| 4 | Dermatology | 34 | 95.63 | 21 | 98.63 | 19.9 | 98.36 |
| 5 | Ionosphere | 33 | 86.89 | 11 | 94.59 | 12.2 | 94.02 |
| 6 | Spectf | 44 | 69.29 | 22 | 84.64 | 19.2 | 83.15 |
| 7 | Sonar | 60 | 87.5 | 25 | 93.75 | 28.9 | 93.60 |
| 8 | Musk | 166 | 85.92 | 72 | 94.12 | 79.3 | 92.38 |
| 9 | Synthetic 1 | 13 | 36.52 | 6 | 48.31 | 7 | 46.8 |
| 10 | Synthetic 2 | 19 | 49.03 | 8 | 67.10 | 8.6 | 65.55 |
| 11 | Synthetic 3 | 30 | 47.02 | 9 | 60.18 | 14.3 | 58.54 |
| 12 | Synthetic 4 | 34 | 19.67 | 18 | 29.78 | 14.9 | 28.96 |
| 13 | Synthetic 5 | 33 | 52.42 | 16 | 67.52 | 13.8 | 66.30 |
| 14 | Synthetic 6 | 44 | 64.79 | 22 | 79.78 | 22.9 | 77.12 |
| 15 | Synthetic 7 | 60 | 57.69 | 31 | 70.67 | 30.9 | 68.22 |
| 16 | Synthetic 8 | 166 | 51.05 | 84 | 62.40 | 86.5 | 60.78 |

**Table 4.** Average classification accuracies.

| No. | Dataset | Local search | | BPSO | | CBPSO | | CBPSOL | |
|---|---|---|---|---|---|---|---|---|---|
| | | #Features | Acc. (%) | #Features | Acc. (%) | #Features | Acc. (%) | #Features | Acc. (%) |
| 1 | Wine | 6 | 97.75 | 8 | 99.44 | 8 | 99.44 | 8 | 99.44 |
| 2 | Hepatitis | 18 | 66.45 | 12.9 | 77.42 | 11.9 | 76.97 | 11.9 | 77.48 |
| No. | Dataset | Local search | | BPSO | | CBPSO | | CBPSOL | |
| | | #Features | Acc. (%) | #Features | Acc. (%) | #Features | Acc. (%) | #Features | Acc. (%) |
| 3 | WDBC | 15.5 | 95.84 | 18.34 | 97.68 | 16 | 97.75 | 15.4 | 97.80 |
| 4 | Dermatology | 19 | 94.97 | 19.8 | 98.36 | 20.2 | 98.33 | 19.9 | 98.36 |
| 5 | Ionosphere | 17.9 | 89.92 | 14.5 | 93.48 | 13.2 | 93.82 | 12.2 | 94.02 |
| 6 | Spectf | 22 | 75.62 | 22.06 | 83.15 | 22.4 | 83.11 | 19.2 | 83.15 |
| 7 | Sonar | 31 | 88.14 | 30 | 93.13 | 29.7 | 92.98 | 28.9 | 93.60 |
| 8 | Musk | 85.1 | 86.70 | 81 | 91.74 | 85.2 | 91.93 | 79.3 | 92.38 |
| 9 | Synthetic 1 | 6.5 | 40.39 | 5.6 | 48.15 | 6 | 48.31 | 7 | 46.8 |
| 10 | Synthetic 2 | 18 | 56.77 | 8 | 66.45 | 7.9 | 66.97 | 8.6 | 65.55 |
| 11 | Synthetic 3 | 29 | 52.11 | 12.6 | 59.23 | 12.4 | 59.05 | 14.3 | 58.54 |
| 12 | Synthetic 4 | 33 | 22.13 | 16.3 | 28.03 | 14.6 | 29.04 | 14.9 | 28.96 |
| 13 | Synthetic 5 | 32 | 55.84 | 14 | 65.41 | 14.1 | 66.01 | 13.8 | 66.30 |
| 14 | Synthetic 6 | 43 | 67.42 | 20.8 | 77.19 | 20.2 | 77.45 | 22.9 | 77.12 |
| 15 | Synthetic 7 | 59 | 61.54 | 32.4 | 66.39 | 30.5 | 66.44 | 30.9 | 68.22 |
| 16 | Synthetic 8 | 165 | 52.92 | 81.5 | 60.44 | 82.7 | 60.29 | 86.5 | 60.78 |

## 7. Conclusion

Finding the best subset requires a good search algorithm to search among $2^d$ possible cases. A classification wrapper feature selection mechanism was proposed and tested in this paper. The metaheuristic search works along with the chaos inertia weight and local search in order to find effective features for classification. The metaheuristic searching algorithm with chaos inertia weight prevents premature convergence to local minima and ends at a good point. This point is the start point of the local search and it increases the accuracy and decreases the number of features. The aim was to increase the classification accuracy and decrease the number of effective features with equal accuracy. We compared its performance with the other feature selection methods. CBPSOL has a strong search capability in the problem space and can efficiently find minimal feature subsets in real datasets and the high-dimensional synthetic datasets presented in this work. Experimental results demonstrate a competitive performance.

## 8. Future work

Some suggestions for future work are:

- Using local search inside CBPSO and proposing a mimetic algorithm for feature selection problems.
- A filter-based method can be used before CBPSOL.
- Using other global searches (GA, ACO, etc.) and other classification methods (ANN, SVM, etc.) can be an avenue for more research.

## References

[1] Jain A, Zongker D. Feature selection: evaluation, application, and small sample performance. IEEE T Pattern Anal 1997; 19: 153-158.

[2] Mitra S, Kundu PP, Pedrycz W. Feature selection using structural similarity. Inform Sciences 2012; 198: 48-61.

[3] Gheyas IA, Smith LS. Feature subset selection in large dimensionality domains. Pattern Recogn 2010; 43: 5-13.

[4] Wang Y, Dahnoun N, Achim AA. Novel system for robust lane detection and tracking. Signal Process 2010; 92: 319-334.

[5] Sikonja MR, Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF. Mach Learn 2003; 53: 23-69.

[6] Lutu PEN, Engelbrecht AP. A decision rule based method for feature selection in predictive data mining. Expert Syst Appl 2010; 37: 602-609.

[7] Miller A. Subset Selection in Regression. 2nd ed. Boca Raton, FL, USA: CRC Press, 2002.

[8] Tsai JS, Huang WB, Kuo YH, Horng MF. Joint robustness and security enhancement for feature based image watermarking using invariant feature regions. Signal Process 2012; 92: 1431-1445.

[9] Choi SI, Oh J, Choi CH, Kim C. Input variable selection for feature extraction in classification problem. Signal Process 2012; 92: 636-648.

[10] Kabira MM, Shahjahan M, Murase K. A new hybrid ant colony optimization algorithm for feature selection. Expert Syst Appl 2012; 39: 3747-3763.

[11] Navot A. On the role of feature selection in machine learning. PhD, Hebrew University, Jerusalem, Israel, 2006.

[12] Unler A, Murat A. A discrete particle swarm optimization method for feature selection in binary classification problems. Eur J Oper Res 2010; 206: 528-539.

[13] Hsu HH, Hsieh CW, Lu MD. Hybrid feature selection by combining filters and wrappers. Expert Syst Appl 2011; 38: 8144-8150.

[14] Vieira SM, Sousa JMC, Kaymak U. Fuzzy criteria for feature selection. Fuzzy Set Syst 2012; 189: 1-18.

[15] Tahir MA, Smith J. Creating diverse nearest-neighbor ensembles using simultaneous metaheuristic feature selection. Pattern Recogn Lett 2010; 31: 1470-1480.

[16] Whitney AW. A direct method of nonparametric measurement selection. IEEE T Comput 1971; 20: 1100-1103.

[17] Marill T, Green DM. On the effectiveness of receptors in recognition systems. IEEE T Inform Theory 1963; 9: 11-17.

[18] Breaban M, Luchian H. A unifying criterion for unsupervised clustering and feature selection. Pattern Recog 2011; 44: 854-865.

[19] Webb A. Statistical Pattern Recognition. 2nd ed. New York, NY, USA: Wiley, 2002.

[20] Li YM, Zhang SJ, Zeng XP. Research of multi population agent genetic algorithm for feature selection. Expert Syst Appl 2009; 36: 11570-11581.

[21] Vieira SM, Sousa JMC, Runkler TA. Two cooperative ant colonies for feature selection using fuzzy models. Expert Syst Appl 2010; 37: 2714-2723.

[22] Wang X, Yang J, Teng X, Xia W, Jensen R. Feature selection based on rough sets and particle swarm optimization. Pattern Recogn Lett 2007; 28: 459-471.

[23] Ronen M, Jacob Z. Using simulated annealing to optimize feature selection problem in marketing applications. Eur J Oper Res 2006; 71: 842-858.

[24] Chen B, Chen L, Chen Y. Efficient ant colony optimization for image feature selection. Signal Process 2013; 93: 1566-1576.

[25] Hua J, Tembe W, Dougherty ER. Feature selection in the classification of high dimension data. In: IEEE 2008 Genomic Signal Processing and Statistics; 8–10 June 2008; Phoenix, AZ, USA. pp. 1-2.

[26] Jin X, Xu A, Bie R, Guo P. Machine learning techniques and chi square feature selection for cancer classification using SAGE gene expression profiles. Lect Notes Comp Sci 2006; 3916: 106-115.

[27] Liao C, Li S, Luo Z. Gene selection using Wilcoxon rank sum test and support vector machine for cancer. Lect Notes Comp Sci 2007; 4456: 57-66.

[28] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max dependency, max relevance and min redundancy. IEEE T Pattern Anal 2005; 27: 1226-1238.

[29] Biesiada J, Duch W. Feature selection for high dimensional data: a Pearson redundancy based filter. Adv Soft Comp 2008; 45: 242-249.

[30] Rocchi L, Chiari L, Cappello A. Feature selection of stabilometric parameters based on principal component analysis. Med Biol Eng Comput 2004; 42: 71–79.

[31] Kennedy J, Eberhart RC. Particle swarm optimization. In: IEEE 1995 Neural Networks; November–December 1995; Perth, Australia. pp. 1942-1948.

[32] Muneender E, Vinod KDM. Optimal real and reactive power dispatch for zonal congestion management problem for multi congestion case using adaptive fuzzy PSO. In: IEEE 2009 TENCON; 23–26 January 2009; Singapore. pp. 1-6.

[33] Lee S, Soak S, Oh S, Pedrycz W, Jeon M. Modified binary particle swarm optimization. Prog Nat Sci 2008; 18: 1161-1166.

[34] Camci F. Analysis of velocity calculation methods in binary pso on maintenance scheduling. In: IEEE 2008 Applications of Digital Information and Web Technologies; 4–6 August 2008; Ostrava, Czech Republic. pp. 12-17.

[35] Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. In: IEEE 1997 Systems, Man, and Cybernetics; 12–15 October 1997; Orlando, FL, USA. pp. 4104-4109.

[36] Chuang LY, Yang CH, Li JC. Chaotic maps based on binary particle swarm optimization for feature selection. Appl Soft Comput 2011; 11: 239-248.

[37] Yue LG, Xiao HA, Junmin L. A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. In: IEEE 2008 Computational Intelligence and Security; 13–17 December 2008; Suzhou, China. pp. 61-65.

[38] Shen Y, Bu Y, Yuan M. A novel chaos particle swarm optimization (PSO) and its application in pavement maintance decision. In: IEEE 2009 Industrial Electronics and Applications; 25–27 May 2009; Xi'an, China. pp. 3521-3526.

[39] Hao Z, Hong SJ, Tienan Z, Yang L. An improved chaotic particle swarm optimization and its application in investment. In: IEEE 2008 Computational Intelligence and Design; 17–18 October 2008; Wuhan, China. pp. 124-128.

[40] Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Yu PS et al. Top 10 algorithms in data mining. Knowl Inf Syst 2008; 14: 1-37.