Computer Engineering and Applications Vol. 7, No. 1, February 2018

A New Method to Improve Feature Selection with Meta-Heuristic Algorithm and Chaos Theory

Mohammad Masoud Javidi[1], Nasibeh Emami[2]

[1]*Dept. of IT, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran*
[2]*Dept. of Computer, Azad univ of Ghermi, Ghermi, Iran*
[1]*balafarila@tabrizu.ac.ir,* [2]*norzadeh.rouya@gmail.com*

ABSTRACT

Finding a subset of features from a large data set is a problem that arises in many fields of study. It is important to have an effective subset of features that is selected for the system to provide acceptable performance. This will lead us in a direction that to use meta-heuristic algorithms to find the optimal subset of features. The performance of evolutionary algorithms is dependent on many parameters which have significant impact on its performance, and these algorithms usually use a random process to set parameters. The nature of chaos is apparently random and unpredictable; however it also deterministic, it can suitable alternative instead of random process in meta-heuristic algorithms.

Keywords*:* Feature Selection, Classification, Meta-heuristic Algorithm, Binary Particle Swarm Optimization, Chaos Theory

1. INTRODUCTION

Feature selection is essential in analyzing large dataset, especially being a preprocessing step to reducing dimensionality, removing irrelevant features, reducing storage requirements and enhancing output comprehensibility (Mitra et al. 2012). Applications of feature selection can be noted pattern recognition (Kanan and Faez 2008; Wang et al. 2012; Huang and Aviyente 2006; Awaidah and Mahmoud 2009), machine learning (Sikonja and Kononenko 2003) and data mining (Patricia et al 2010). The term of feature selection is taken to refer to algorithms that the their input is feature set and output of them is a subset of input feature set (Jain and Zongker 1997). General procedure of feature selection algorithms is creating a subset, evaluate it, and loop until a stop criterion is satisfied. Then the subset extracted is validated by the classifier algorithm (Novaković et al. 2011; Chen et al. 2006).

Feature selection algorithms can be classified into two categories based on their evaluation procedure (Ferreira and Figueiredo 2012; Dash 1997):

Filter: the quality of a subset of features is determined by using characteristics of that subset, without use any learning algorithm.

Wrapper: To determining the adequacy of a subset of features, use learning algorithm and performance of learning algorithm is a measure to select subset or not.

In (Hall 1999) there is a good explanation of filter and wrapper methods. We describe them here:

Since wrapper methods use a learning algorithm to evaluate each feature subset; are expensive to run but give better results (predictive accuracy) than filters. Also these methods are less general than filters and must be re-run when switching from one learning algorithm to another. Filters don’t use learning algorithm they are many times faster than wrappers. Filters do not require re-execution of different learning algorithms. Filters can provide a good starting feature subset for a wrapper method. A process that is likely to result in a shorter, and hence faster, search for the wrapper. The Table 1 shows a summary comparison between the wrapper and filter methods.

TABLE 1.
Comparison between the wrapper and filter methods

Method	The need for learning algorithm	Predictive accuracy	Execute times
filter	No	low	fast
wrapper	Yes	high	slow

Search is an important issue in feature selection problem because the whole search space for optimization contains all possible subsets of features, the size of such space is2^d. Where d is the number of original features. Because of this space typically feature selection algorithms include heuristic or random search strategies to avoid this prohibitive complexity (Hosseinzadeh et al. 2009). Nevertheless development of a highly accurate and fast search algorithm for the selection of optimal feature subset is an open issue (Gheyas and Smith 2010).

In this paper we proposed a wrapper feature selection for classification. The proposed algorithm is based on one new binary particle swarm optimization and chaos inertia weight. We use the K-nearest neighbor (K-NN) method with leave-one-out cross-validation as a classifier for evaluating classification accuracies.

This paper organized in six sections: Section 2 reviews some previous studies in the area of feature selection, section 3 is preliminaries about proposed method. Proposed method will explain in section 4, implementation and result coming in section 5 and finally conclusion coming in section 6.

2. RELATED WORKS

In this Section, we review some feature selection techniques. Several common feature selection methods are named here. As we said in previous section feature selection methods generally fall into two categories: filter and wrapper. Some filter approaches are: t-test (Hua et al. 2008), chi-square test (Jin et al. 2006), Wilcoxon Mann–Whitney test (Liao et al. 2007), mutual information (Peng et al. 2005), Pearson correlation coefficients (Biesiada and Duch 2008) and principal component analysis (Rocchi 2004) Relief (Kira and Rendell 1992), Focus (Almuallim and Dietterich 1991), LVF (Liu and Setiono 1996), SCRAP (Raman and Ioerger 2002), EBR (Jensen and Shen 2001), FDR (Traina et al. 2000) and etc.

The similarity of filter method is that ranking the features by a metric and eliminate all features that do not achieve an adequate score (Chen et al. 2013). In wrapper approach since exhaustive search is not computationally feasible, the wrapper methods employ a search algorithm to search for an optimal feature subset. In General Wrapper methods can be classified into two categories based on search strategy (Gheyas and Smith 2010), Greedy and Randomized/stochastic.

Greedy wrapper approaches use less computer time than other wrapper methods. Sequential forward selection (SFS) (Peng et al. 2003; Guan et al. 2004), is to start the search process with an empty set and successfully add features; and Sequential backward selection (SBS) (Gasca et al. 2006; Hsu et al. 2002), is to start with a full set and successfully remove features; are the two most commonly used wrapper methods that use a greedy search strategy. The disadvantage of SFS and SBS is that they can easily be fall into local minima (Gheyas and Smith 2010).

Stochastic algorithms developed for solving wrapper feature selection such as Ant Colony Optimization (ACO) (Kabira et al. 2012; Sivagaminathan and Ramakrishnan 2007), Genetic Algorithm (GA) (Tsai et al. 2013; Yang et al. 2011), Particle Swarm Optimization (PSO) (Sahu and Mishra 2012; Wang et al. 2007). They are global search and cannot easily be trapped into local minima. They can produce the best solution by heuristic information but these algorithms are computationally expensive (Gheyas and Smith 2010; Chen 2013).

In this paper we will introduce a wrapper feature selection method to search in exhausted feature space and find an optimal feature subset for classifier task. In the next section we introduce preliminaries of the proposed method.

3. PRELIMINARIES

In proposed algorithm we used a new version of the Binary Particle Swarm Optimization with chaotic inertia weight. So the following is a more detailed description of Particle Swarm Optimization, Binary Particle Swarm Optimization, New Binary Particle Swarm Optimization, Chaos theory for setting inertia weight.

3.1 PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) was first suggested by Kennedy and Eberhart in 1995(Kennedy J, Eberhart 1995). PSO is a global optimization that is inspired by the social behavior of birds. It is a population based optimization technique, where a population is called a swarm (Thangavel et al. 2012). A swarm consists of N particles moving around in a d-dimensional search space. The position of the ith particle can be represented by:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \quad i = 1,2, \dots, N$$

And for represented velocity of each particle we have:

$$v_i = (v_{i1}, v_{i2}, \dots, v_{id}) \quad i = 1,2, \dots, N$$

The positions and velocities of the particles are confined within $[X_{min}, X_{max}]^d$ and $[V_{min}, V_{max}]^d$ respectively. Each particle has a memory that keeps its previous best position:

$$P\_best_i = \left(P_{best\ i1}, P_{best\ i2}, \dots, P_{best\ id}\right) \quad i = 1,2,\dots,N$$

In PSO, we have global best concept that it is the best position among all the particles in the population and can be represented by:

$$G_{best} = \left(G_{best\ 1}, G_{best\ 2}, \dots, G_{best\ d}\right)$$

At each iteration, the velocity and the position of each particle are updated according to its previous best position (*P_best*) and the global best position (*G_best*). Redefined formula are:

$$v_{ij}(T+1) = wv_{ij}(T) + C_1Rand1\left(P\_best_i - x_{ij}(T)\right) + C_2Rand_2\left(G_{best} - x_{ij}(T)\right) \qquad (1)$$

$$x_{ij}(T+1) = x_{ij}(T) + v_{ij}(T+1) \qquad (2)$$

where j=1,2,…,d, *w* is the inertia coefficient between [0, 1], *C1*, *C2* are the acceleration constants, *Rand1* and *Rand2* are random number between [0, 1]. $v_{ij}(T+1)$ and $v_{ij}(T)$ are velocities of the updated particle and the particle before being updated, respectively $x_{ij}(T)$ is the original particle position, and $x_{ij}(T+1)$ is the updated particle position (Chuang et al. 2011).

PSO was presented to solve problems in continuous space; in discrete space problems Kennedy and Eberhart proposed binary version of PSO (BPSO) (Kennedy J and Eberhart 2007). In BPSO the position of a particle is represented as the binary string and is randomly generated. In feature selection problem zero bit means unselected feature and bit with one value means that selected feature. The initial velocities are probabilities limited to a range of [0, 1] and velocity update by Eq (1) (Chuang et al. 2011). If the velocity after updating in each dimension exceed V_{max} then the velocity of that dimension is limited to V_{max} (Eq. (3)). Both V_{max} and V_{min} are user-specified parameters (Chuang et al. 2011).

$$If\, v_{ij}(T+1) \notin (V_{min}, V_{max})$$

$$then\ v_{ij}(T+1) =$$

$$\max\left(min\left(V_{max}, v_{ij}(T+1)\right), V_{min}\right)$$

$$j = 1,2,\dots,d \qquad (3)$$

In order to update position of each particle, we should first transform the velocity vector into a probability vector through a sigmoid function (Unler and Murat 2010). Figure 1 shows a sigmoid function.

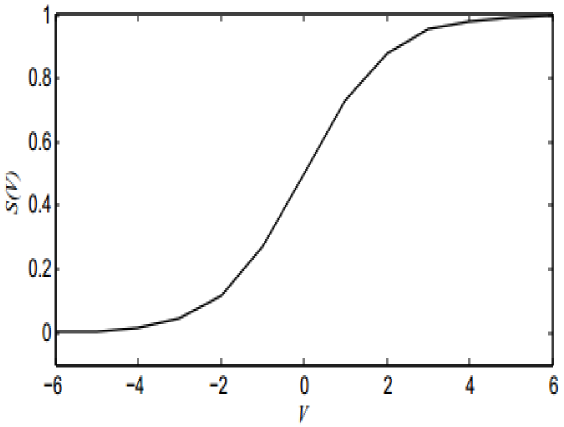


FIGURE 1. Sigmoid function (Rostami and Nezamabadi 2006)

So Equation (4) and (5) use for update position of each particle.

$$S\left(v_{ij}(T+1)\right)=\frac{1}{1+e^{-v_{ij}(T+1)}}$$

$$j=1,2,\dots,d \qquad (4)$$

$$x_{ij}(T+1) \quad =\begin{cases}1 \ if\ rand < S\left(v_{ij}(T+1)\right) \\ 0 \qquad\qquad\qquad O.W\end{cases}$$

$$j=1,2,\dots,d \qquad (5)$$

3.2 NEW BINARY PARTICLE SWARM OPTIMIZATION

In original BPSO the new position of each particle is based on the likelihood function (sigmoid function) that v_ij (T+1) passes of the sigmoid function. Because of use this function in original BPSO, Rostami and Nezamabadi in (2006) Objections were made on the original BPSO.

When the particle velocity is close to zero for a specified dimension, it means that the particle is in a good position and the position of the particle shouldn’t change. But with sigmoid function, the probability of the particle’s position be changed and be zero or one is equal. So Rostami and Nezamabadi in (2006) present a new likelihood function. Figure 2 shows new likelihood function.

In Equation (4) previous position of the particle to calculate the next position of the particle's position is not considered.

To eliminate the disadvantage of BPSO, they proposed Equation (6) and (7):

$$S'\left(v_{ij}(T+1)\right) =$$
$$2 * abs\left(S\left(v_{ij}(T+1)\right) - 0.5\right)$$
$$j = 1,2,\ldots,d \qquad (6)$$

$$If\ \left(rand < S'\left(v_{ij}(T+1)\right)\right)$$
$$then\ x_{ij}(T+1) = complement\left(x_{ij}(T)\right)$$
$$else\ x_{ij}(T+1) = x_{ij}(T)$$
$$j = 1,2,\ldots,d \qquad (7)$$

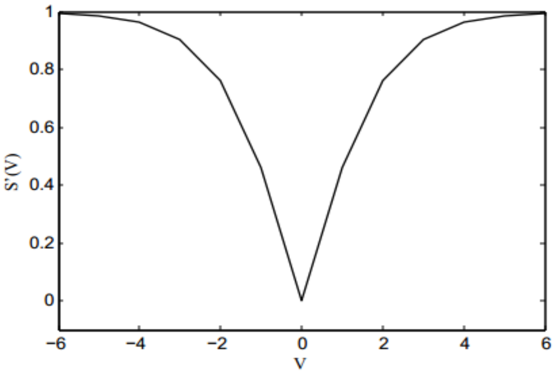


FIGURE 2. $S'(v_{id})$function [44]

3.3 CHAOTIC SEQUENCES FOR INERTIA WEIGHT

The inertia weight as a PSO's parameters make a balance between the exploration and exploitation. Inertia weight with a large value provides a global search while inertia weight with a small value provides a local search (Nickabadi et al. 2011). PSO or BPSO have prematurely convergent problem and trap into local minimum. To solve above problem, some improved measures are proposed such as embedded crossover operation in algorithm or use chaos theory (Shen et al. 2009).

Chaos is highly sensitive to the initial values and thus it provides great diversity based on the ergodic property, which allows transiting states without repetition in certain ranges. Chaos is usually highly sensitive to the initial values and thus provides great diversity based on the ergodic property of the chaos phase, which transits every state without repetition in certain ranges. Because of these characteristics, chaos theory can be applied in optimization (Chuang wt al. 2011).

One application of chaos system is in determining of the inertia weight for BPSO based on logistic map; to prevent early convergence, and thus achieve superior classification results in wrapper feature selection (Chuang et al. 2011). The logistic map can be described by the Equation (8):

$$w_{(T+1)} = 4 \times w_{(T)} \times \left(1 - w_{(T)}\right) \quad w_{(T)} \in (0,1) \tag{8}$$

In this equation, $w_{(T)}$ is the T th chaotic number where T denotes the iteration number.

4. K-NEAREST NEIGHBOR CLASSIFICATION (KNN)

K- nearest neighbor is one of the none parametric learning approaches mainly used for classification (Pedrycz and Chen 2015). In application of classification an ith instance is represented by a feature vector namely:

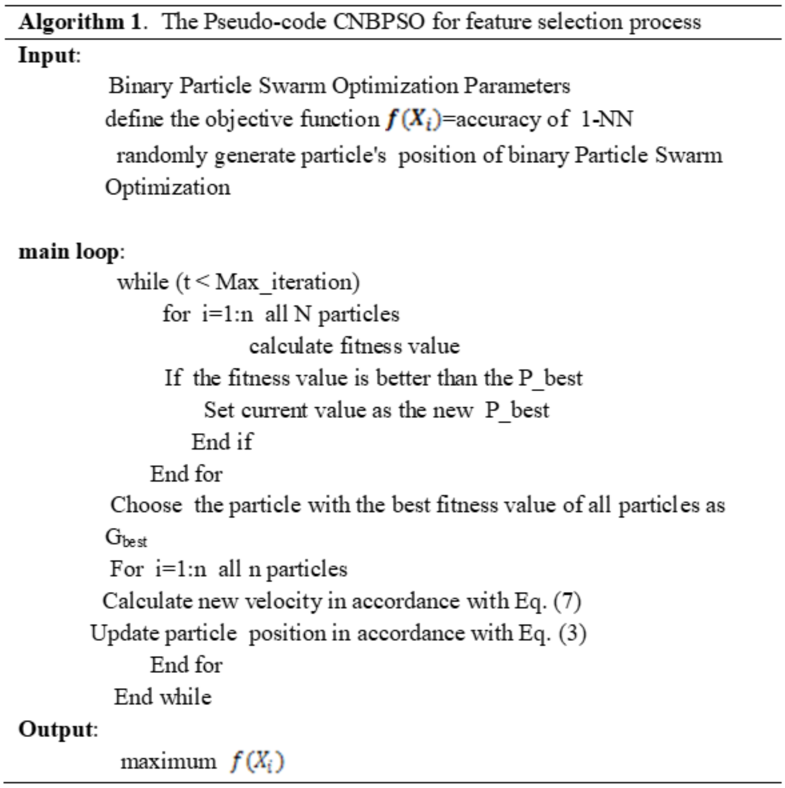
$$X_i = (< x_{i1}, x_{i2}, \dots, x_{id} >, C),$$

where x_{id} denotes the value of the i^{th} feature, and C denote the class variable. K nearest neighbor is a famous classifier that based on the distance function as a measure the difference or similarity between two instances. The standard Euclidean distance between two instance X and Y is often used as the distance function (Jiang et al. 2007). To predictive class majority voting among the data records in the neighborhood is usually used to decide (Wu et al. 2008).

5. PROPOSED METHOD

In this paper; we present Chaotic New Binary Particle Swarm Optimization (CNBPSO) for wrapper feature selection. The Position of each particle is a binary string; if it has 1 in each dimension means selected feature and 0 means that unselected feature. At first, binary strings or subsets, as a candidate solutions, produce randomly then evaluated by the evaluator function. The accuracy of 1-Nearest Neighborhood with leave one out cross validation is the criteria for evaluation solution. In each iteration position and velocity of each particle update by Equation (3) and (7) respectively. Proposed method enter stop phase after specific number iteration.

In proposed method, Binary Particle Swarm Optimization with new likelihood function capable to have good exploration of new regions of the feature space by improving of BPSO's and CBPSO's disadvantage. That is, when the particle has a proper position, the position of the particle should not be changed. in order to probability of changing reach to zero at the zero velocity, in the new probability function, the sigmoid function is mapped as much as 0.5. On the other hand, increasing the velocity of the particle in both the positive and negative directions means increasing the probability of changing the position of the particle, so that at the beginning and the end of the interval, the magnitude of the probability function must be equal to one. Therefore, multiplication 2 is used in Equation (6). Also in proposed method chaos logistic map used to determine the inertia weight that prevents early convergence. So it helps to produce a better quality solution. Flowchart of a proposed search method is in Figure 3.



```
Algorithm 1.  The Pseudo-code CNBPSO for feature selection process
Input:
        Binary Particle Swarm Optimization Parameters
        define the objective function $f(X_i)$=accuracy of 1-NN
        randomly generate particle's position of binary Particle Swarm
        Optimization

main loop:
        while (t < Max_iteration)
            for i=1:n all N particles
                calculate fitness value
            If the fitness value is better than the P_best
                Set current value as the new P_best
            End if
            End for
        Choose the particle with the best fitness value of all particles as
        $G_{best}$
        For i=1:n all n particles
        Calculate new velocity in accordance with Eq. (7)
        Update particle position in accordance with Eq. (3)
            End for
        End while
Output:
        maximum $f(X_i)$
```

FIGURE 3. Algorithm 1 is the Pseudo-code of CNBPSO for feature selection process.

6. IMPLEMENTATION AND RESULT EVALUATION

6.1 DATASET

The dataset in this paper is coming from UCI (https://archive.ics.uci.edu/ml/datasets.html). Data sets selected such that cover medium and large scale of the feature selection problem. Data sets with number of features between 20, 49 are medium scale and greater than 50 are large (Tahir and Smith 2010). Table 2 shows selected data set from UCI and their characteristic. For controlling of domain values of each feature, Features are normal in the range of 0 and 1 (except Libras dataset that are between 0, 1) normalization formula is as follows:

$$x = \frac{x - min_x}{max_x - min_x} \qquad (9)$$

In Equation (9), x is the value of feature, min_x is minimum and max_x is maximum value of each feature.

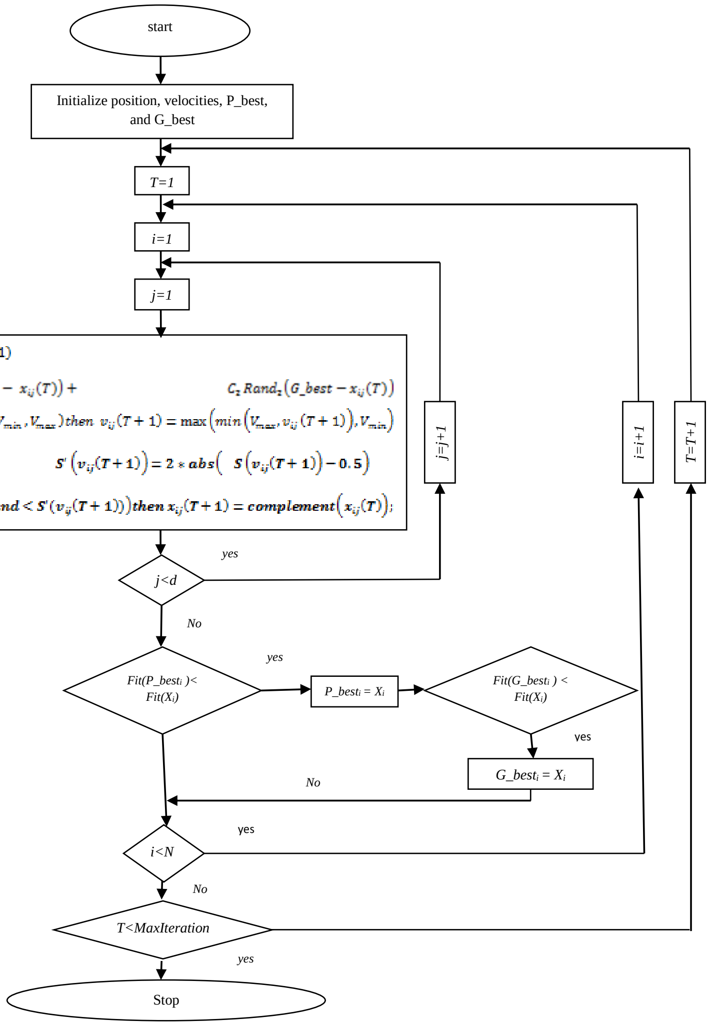

start
Initialize position, velocities, P_best, and G_best
T=1
i=1
j=1
$=v_{ij}(T+1)$
$nd_1(P\_best_i - x_{ij}(T)) +$
$C_2 Rand_2(G\_best - x_{ij}(T))$
$(T+1) \notin (V_{min}, V_{max}) then\ v_{ij}(T+1) = \max(min(V_{max}, v_{ij}(T+1)), V_{min})$
$S'(v_{ij}(T+1)) = 2 * abs(\ S(v_{ij}(T+1)) - 0.5)$
$If\ (rand < S'(v_{ij}(T+1)))\, then\ x_{ij}(T+1) = complement(x_{ij}(T));$
j=j+1
i=i+1
T=T+1
j<d
yes
No
Fit(P_besti)< Fit(Xi)
yes
P_besti = Xi
Fit(G_besti) < Fit(Xi)
yes
G_besti = Xi
No
i<N
yes
No
T<MaxIteration
yes
Stop

FIGURE 4. Flowchart of proposed method

TABLE 2.
Dataset

No.	Datasets	Features	Sample	Classes
1	*ionosphere*	34	351	2
2	*Chess (* King-Rook vs. King-Pawn*)*	36	3196	2
3	*spectf*	44	267	2
4	*lung cancer*	57	32	3
5	*sonar*	60	208	2
6	Libras Movement Data Set	91	360	15
7	*Musk(version 1)*	166	476	2

6.2 INITIAL PARAMETERS SETTING UP

CNBPSO such as every version of original BPSO have parameters must be adjusted. This parameter includes number of particles, acceleration constants, inertia weight setting up and stopping criteria. In our application the number of particles is 20, acceleration constants are 1.49, for setting up inertia weight; we use logistic map chaotic sequence to start point 0.86. The stopping criterion of CNBPSO is after 200 iterations. The minimum and maximum velocity are -6 and 6 respectively. This value is almost ubiquitously adopted in PSO research [41].

6.3 EXPERIMENTAL EVALUATION

In this section, we have evaluated the effectiveness of the proposed method on datasets that introduced in section 5.1. The proposed model is implemented in MATLAB software and on computer using Intel core i7. We tried to have diversity dataset; especially in terms of number of features. We compare the results of the proposed method (CNBPSO) with BPSO and CBPSO (Chuang et al. 2011).

All algorithms have the same parameters and used 1-nearest neighbor by leave one out cross validation to select an optimal subset, just only inertia weight for BPSO is constant, namely 0.86. Due to the nature of randomizing of algorithms; we run them ten times and we report average classification accuracy too. Our result adjusts in three tables in terms of average of accuracy, the best accuracy and smallest feature subset between ten times run. The result in Table 3 shows that in case of average; CNBPSO has better performance (in terms of accuracy) than BPSO and CBPSO to find optimal subset. But this performance is associated with average number of feature increased. Obtained the best accuracy in during oftentimes run of BPSO, CBPSO and CNBPSO shows in Table 4.

In terms of the best accuracy, the proposed method has better result (accuracy) than CBPSO and BPSO, but associated with increasing number of features except Ionosphere and Musk. In following the smallest Feature subset is coming in Table 5 in during of 10 times run algorithms.

TABLE 3.

Average accuracy

No	Data set	Without feature selection		BPSO[41]		CBPSO[41]		Proposed method(CNBPSO)	
		#feature	acc	#feature	acc	#feature	acc	#feature	acc
1	*ionosphere*	34	86.89	14.5	93.48	13.2	93.82	12.6	94.04
2	*chess*	36	83.76	21.4	97.66	22.9	97.86	22.4	98.18
3	*spectf*	44	69.29	22.06	83.15	22.4	83.11	24.22	84.27
4	*lungcancer*	57	43.75	28.1	75.94	26.6	77.19	28	81.87
5	*sonar*	60	87.5	30	93.13	29.7	92.98	31.6	94.28
6	*libras*	91	87.22	42.7	89.58	41.9	89.75	44.4	90.33
7	*musk*	166	85.92	81	91.74	85.2	91.93	85.2	94.45

#feature= average of feature numbers, acc=Average of accuracy

TABLE 4.
The best accuracy

No	Data set	BPSO[41]		CBPSO[41]		Proposed method(CNBPSO)	
		#feature	acc	#feature	acc	#feature	acc
1	*ionosphere*	13	94.30	12	94.02	9	95.16
2	*Chess*	23	98.06	23	98.25	24	98.44
3	*Spectf*	18	84.64	23	83.89	20	86.14
4	*lung cancer*	26	78.12	18	81.25	24	87.50
5	*Sonar*	32	95.19	30	93.75	30	96.15
6	*Libras*	40	90	40	90.28	50	91.11
7	*Musk*	86	92.44	76	93.28	74	96.22

TABLE 5.
The smallest feature subset

No	Data set	BPSO[41]		CBPSO[41]		Proposed method(CNBPSO)	
		#feature	acc	#feature	acc	#feature	acc
1	*ionosphere*	11	93.45	12	94.02	9	95.19
2	*chess*	17	97.62	19	97.78	21	98.25
3	*spectf*	18	84.64	15	83.15	20	86.14
4	*lungcancer*	22	71.87	18	81.25	22	84.37
5	*sonar*	26	93.27	26	93.27	27	95.19
6	*libras*	37	89.44	35	90	33	90.83
7	*musk*	67	91.81	76	93.28	66	96

#feature= minimum feature numbers, acc= accuracy

To have Quick comparison between algorithms, you can see the results in the Figure 4 and 5. Figure 4 shows datasets versus average accuracy of each algorithm and Figure 5 shows datasets versus best accuracy of each algorithm. Y axis is the percent of Accuracy and X axis is dataset that used in our paper.

7. CONCLUSION

Feature selection is an important preprocessing technique in many applications. Due to be intractable of problems, search is a key issue. In this paper, we have presented a new way of wrapper feature selection for classification tasks. The proposed method (CNBPSO) by using new likelihood function and chaotic logistic map for inertia weight; attempt to find the best feature subset such that accuracy of classification increase. In fact with this modification, proposed method avoid falling in local minima and as the results show, produce better result than BPSO and CBPSO.

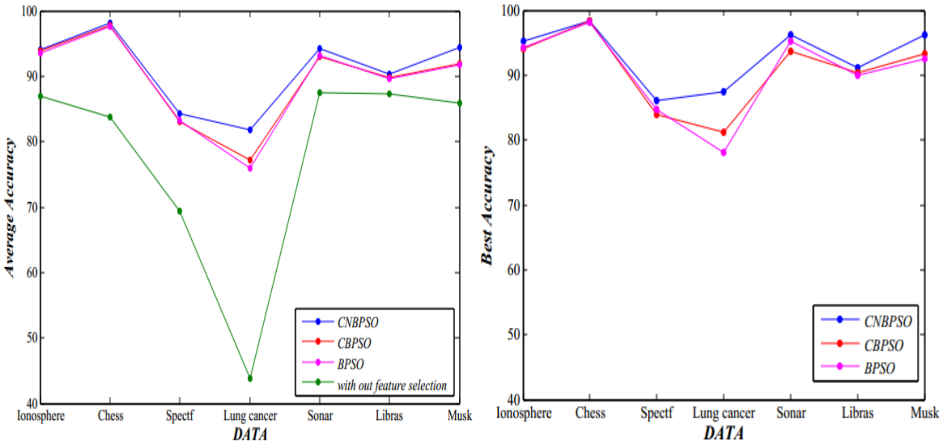


FIGURE 5. Average accuracy

REFERENCES

[1] Almuallim H, Dietterich T G (1991) Learning with many irrelevant features. In: Proceedings of Ninth National Conference on Artificial Intelligence 547–552.

[2] Awaidah S M, Mahmoud S A (2009) A multiple feature/resolution schemeto Arabic (Indian) numerals recognition using hidden Markov models. Signal Process 89: 1176–1184.

[3] Biesiada J, Duch W (2008) Feature selection for high-dimensional data—a Pearson redundancy based filter. Advances in Soft Computing 45:242–249.

[4] Chen B, Chen L, Chen Y (2013) Efficient ant colony optimization for image feature selection. Signal Process 93:1566–1576.

[5] Chen Y, Li Y, Cheng X, Guo L (2006) Survey and taxonomy of feature selection algorithms in intrusion detection system. LECT NOTES COMPUT SC 4318: 153-167.

[6] Chuang L Y, Hsiao C J, Yang C H (2011) Chaotic particle swarm optimization for data clustering. EXPERT SYST APPL 38: 14555–14563.

[7] Chuang L, Yang C, Li J C (2011) Chaotic maps based on binary particle swarm optimization for feature selection. Applied Soft Computing 11: 239–248.

[8] Dash M, Liu H (1997) Feature selection for classification. Intelligent Data Analysis 1:131–156.

[9] Ferreira A J, Figueiredo M A T (2012) Efficient feature selection filters for high-dimensional data. PATTERN RECOGN LETT 33: 1794–1804.

[10] Gasca E, Sanchez J S, Alonso R (2006) Eliminating redundancy and irrelevance using a new MLP-based feature selection method. PATTERN RECOGN 39: 313–315.

[11] Gheyas I A, Smith L S (2010) Feature subset selection in large dimensionality domains. PATTERN RECOGN 43:5-13.

[12] Guan S, Liu J, Qi Y (2004) An incremental approach to contribution based feature selection. Journal of Intelligence Systems 13.

[13] Hall M A (1999) Correlation-based feature selection for machine learning. thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at The University of Waikato.

[14] Hosseinzadeh Aghdam M, Ghasem Aghaee N, Basiri M E (2009) Text feature selection using ant colony optimization. EXPERT SYST APPL 36: 6843–6853.

[15] Hsu C, Huang H, Schuschel D (2002) The ANNIGMA-wrapper approach to fast feature selection for neural nets. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 32:207–212.

[16] Hua J, Tembe W, Dougherty E R. (2008) Feature selection in the classification of high-dimension data. IEEE International Workshop on Genomic Signal Processing and Statistics 1–2.

[17] Huang K, Aviyente S (2006)Information-theoretic wavelet packet sub-band selection for texture classification. Signal Process 86:1410–1420.

[18] Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence 19:153 - 158.

[19] Jensen R, Shen Q(2001) A rough set aided system for sorting WWW bookmarks. Web Intelligence: Research and Development 95–105.

[20] Jiang L, Cai Z, Wang D, Jiang S(2007) Survey of improving k-nearest-neighbor for classification. In: Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery , , 1, pp. 679 – 683.

[21] Jin X, Xu A, Bie R, Guo P(2006) Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles, LECT NOTES COMPUT SC 3916: 106–115.

[22] Kabira M M, Shahjahan M, Murase K(2012) A new hybrid ant colony optimization algorithm for feature selection. EXPERT SYST APPL 39:3747–3763.

[23] Kennedy J, Eberhart R C (1997) A discrete binary version of the particle swarm algorithm. In: Proceedings of the 1997 Conference on Systems, Man, and Cybernetics 4104–4109.

[24] Kennedy J, Eberhart R C (1995) Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks 4:1942–1948.

[25] Kira K, Rendell L A (1992) The feature selection problem: traditional methods and a new algorithm. In: Proceedings of Ninth National Conference on Artificial Intelligence 129–134.

[26] Liao C, Li S, Luo Z (2007) Gene selection using Wilcoxon rank sum test and support vector machine for cancer. LECT NOTES COMPUT SC 4456:57–66.

[27] Liu H, Setiono R (1996) A probabilistic approach to feature selection – a filter solution. In: Proceedings of Ninth International Conference on Industrial and Engineering Applications of AI and ES 284–292.

[28] Mitra S, Kundu P, Pedrycz W (2012) Feature selection using structural similarity. INFORM SCIENCES 198: 48–61.

[29] Kanan H R, Faez K (2008) An improved feature selection method based on ant colony optimization (aco) evaluated on face recognition system. APPL MATH COMPUT 205: 716–725.

[30] Nickabadi A, Ebadzadeh M M, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. Applied Soft Computing 11: 3658–3670.

[31] Novaković J, Strbac P, Bulatović D(2011) Toward optimal feature selection using ranking methods and classification algorithms. Yugoslav Journal of Operations Research 21: 119-135.

[32] Patricia E N, Andries L, Engelbrecht P(2010) A decision rule-based method for feature selection in predictive data mining. EXPERT SYST APPL 37:602–609.

[33] Pedrycz W , Chen S. M (2015) Information Granularity, Big Data, and computational Intelligence. Springer, Heidelberg, Germany.

[34] Peng H, Long F, Ding C (I2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min redundancy. IEEE T PATTERN ANAL 27:1226–1238.

[35] Peng H, Long F, Ding C(2003) Over fitting in making comparisons between variable selection methods. Journal of Machine Learning Research 3: 1371–1382.

[36] Raman B, Ioerger T R (2002) Instance based filter for feature selection, Journal of Machine Learning Research 1: 1–23.

[37] Rocchi L, Chiari L, Cappello A(2004) Feature selection of stabilometric parameters based on principal component analysis. Medical and Biological Engineering and Computing 42: 71–79.

[38] Rostami N, Nezamabadi H (2006) A new method for binary PSO. In: proceedings of the international Conference on Electronic Engineering (in Persian).

[39] Sahu B, Mishra D (2012) A novel feature selection algorithm using particle swarm optimization for cancer microarray data. Procedia Engineering 38:27-31.

[40] Shen Yi, Bu Y, Yuan M (2009) A novel chaos particle swarm optimization (pso) and its application in pavement maintance decision. In: Proceedings on fourth IEEE Conference on Industrial Electronics and Applications 3521-3526.

[41] Sikonja M R, Kononenko I (2003) Theoretical and empirical analysis of Relief and Relieff. Machine Learning 53: 23–69.

[42] Sivagaminathan R K, Ramakrishnan S(2007) A hybrid approach for feature subset selection using neural networks and ant colony optimization. EXPERT SYST APPL 33: 49–60.

[43] Tahir M A, Smith J(2010) Creating diverse nearest-neighbor ensembles using simultaneous metaheuristic feature selection. PATTERN RECOGN LETT 31: 1470–1480.

[44] Thangavel K, Bagyamani J, Rathipriya R (2012) Novel hybrid pso-sa model for biclustering of expression data. Procedia Engineering 30: 1048 – 1055.

[45] Traina C, Traina A, Wu L, Faloutsos C (2000) Fast feature selection using the fractal dimension, In: Proceedings of the fifteenth Brazilian Symposium on Databases (SBBD) 158–171.

[46] Tsai C F, Eberle W, Chu C Y(2013)Genetic algorithms in feature and instance selection, Knowledge Based Systems 39: 240-247.

[47] Unler A, Murat A(2010) A discrete particle swarm optimization method for feature selection in binary classification problems. EUR J OPER RES 206: 528–539.

[48] Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. PATTERN RECOGN LETT 28: 459–471.

[49] Wang Y, Dahnoun N, Achim A(2012)A novel system for robust lane detection and tracking. Signal Process 92: 319–334.

[50] Wu X, et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14:1–37.

[51] Yang W, Li D, Zhu L (2011) An improved genetic algorithm for optimal feature subset selection from multi-character feature set. EXPERT SYST APPL 38: 2733–2740.