*Send Orders for Reprints to reprints@benthamscience.ae*

*The Open Automation and Control Systems Journal*, 2014, *6*, 747-753




# A Novel Naive Bayes Classification Algorithm Based on Particle Swarm Optimization

Jun Li[1,2,3,*], Lixin Ding[1,2] and Bo Li[3]

[1]*State Key Laboratory of Software Engineering, Wuhan University, Wuhan, 430072, China*

[2] *School of Computer, Wuhan University, Wuhan, 430072, China*

[3]*College of Computer Science and Technology, Wuhan University of Science and Technology, WUST, Wuhan, 430065, China*

**Abstract:** Naive Bayes (NB) classifier is a simple and efficient classifier, but the independent assumption of its attribute limits the application of the actual data. This paper presents an approach called particle swarm optimization-naive Bayes (PSO-NB) which takes advantage of combination particle swarm optimization with naive Bayes for attribute selection to improve naive Bayes classifier. This method applies PSO firstly to search out an optimal subset of attributes reduction in the original attribute space, and then constructs a naive Bayes classifier on the gotten subset of the attributes reduction. Nineteen experimental results on UCI datasets distinctly show that compared with Cfs-BestFirst algorithm, NB algorithm, Decision Tree(C4.5) algorithm, K-neighbor(KNN) algorithm, the proposed algorithm has higher classification accuracy.

**Keywords:** Naive Bayes, particle swarm optimization algorithm, feature selection, attribute subset, classification accuracy

## 1. INTRODUCTION

Classified prediction is an important branch of data mining. Classification is to identify a set of data collection that can describe typical characteristics of the model to make predictions of the unknown variables or categories. The core part of the classification algorithm is to construct a classifier. Because of the efficient calculation, the high accuracy and the solid theoretical foundation, naive Bayes classifier has been widely used. But Naive Bayes assumes that for the given class all the attributes of instance are independent of each other, which is called Naive Bayes assumption. Owing to the independence between attributes, the parameters of each attribute can be estimated separately, which simplifies the calculation greatly; making it especially suitable for the classification problems with a very large number of attributes (the number of its attributes is usually ranging several thousand to tens of thousands). However, in the real classification problem，this assumption is often untenable. So in many documents, Naive Bayes classifier's performance is not perfect. In order to relax the limitation of attribute independence assumption, many scholars have done a lot of research to improve its performance.

Relevant work is divided into the following three categories:

1. Structure extension: Originally, each attribute variable of Naive Bayes only has one classified variable as its parent node, which is not conducive to express dependencies between attribute variables. So this structure of augmented Naive Bayes uses directed edge expressing dependencies between attributes. Friedman etc. propose a Tree Augmented Naive Bayes classification method, TAN [1]. In TAN classifier, each attribute variables can have at most one other attribute variables except for categorical attribute variables as the parent node, which extends the limitation that each attribute variables only have one categorical attribute as the parent node in Naive Bayes. On the basis of TAN, Friedman proposes Bayesian Augmented Naive Bayes classifier, BAN [2]. In BAN, categorical attribute variables are the parent node of all the attribute variables, any directed acyclic graph can be constructed between attribute variables. Jin Zhe, etc. propose a Bayesian Augmented Naive Bayes classifier GBAN in his paper [3]. GBAN classification derives from genetic algorithms. GBAN classification model not only satisfies BAN model for limiting the network structure, but also includes the following features: Besides the categorical variable as its parent node, each attribute variable has no more than m parent nodes (usually $m \leq 4$).

2. Attribute weighting: Originally, Naive Bayes considered each attribute variable is equally important for classification, so the weights are 1. But in fact it's not always the case. Therefore, when constructing Bayesian, attributes are assigned different weights. Qin Feng etc. propose the Attribute Weighted Naive Bayes improved algorithm whose weighted parameters are learned directly from the training data [4]. Weights can be quantified in order to measure the degree of correlation between condition attribute variables and categorical attributes variables. The prior probability of

1874-4443/14 

weighting adjustments replaces the prior probability of original Naive Bayes to calculate so as to improve the classification performance. Zhang Wen, etc. propose the Weighted Integration Bayesian classification algorithm based on attribute correlation [5], introducing the correlation coefficient in mathematics, considering it as the impact of a property on this category, giving attributes with high or low relevancy to lager or less weight and using attribute weighting method based on the integration learning of AdaBoost algorithm.

3. Feature selection: Select an independence assumption attributes subset which is approximately satisfying the conditions from the entire attribute space and construct Naive Bayes classifier on the new subset of attributes. This article will describe these methods in detail.

Feather subset selection improved Naive Bayes by removing redundant or irrelevant attributes from the training data intensively and only selecting those most useful attributes in the learning stage [6]. In fact, such improved algorithm, only using a subset of the specified attribute to predict, is a variant of Naive Bayes. Now, researchers have proposed a number of feature subset selection algorithms and proved that these algorithms have considerably improved Naive Bayes. Yang Guangzu etc. proposes a Bayesian algorithm feature selection algorithms, firstly sorting the attributes in accordance with the level of information gain value of attributes, and then choosing the properties, so it is possible to improve the performance of classification [7]. Zhang Jing and Wang Jianmin put forward an new attribute reduction based on the attribute correlation [8]. In the paper authors proposed a new definition of attribute relevance based on rough set theory firstly, and then construct attribute reduction algorithm from two aspects. That is, for one thing, the greater correlation between the subset and classification attribute the better; for another, the smaller correlation between attributes subset the better. Eventually, the algorithm find redundant attributes from the attribute set effectively, resulting in ideal reduction of the subset of attributes.

In addition, many common feature subset selection algorithms are not only used to improve Naive Bayes, also used for a variety of classification methods. S.Casale and A. Russo applied joint use of property assessment methods CFSSubsetEval and search strategies BestFirst in the paper [9]. CFSSubsetEval-BestFirst method can also be embedded in WEKA [10] (Waikato Environment for Knowledge Analysis) software. Ratanamahatana and Gunopulos proposed a method for feature subset selection by constructing a decision tree in the paper [11].

Kennedy and Eberhart put forward the particle swarm optimization (PSO) [12], which is a method of optimizing numerical functions on integrating group behavior, human decision and simulation of birds' flight to forage behavior. And it then evolved into a random search method of the optimal solution for feature selection [13, 14].

In the paper, the authors present an algorithm called PSO-NB, which applies PSO to NB. The remainder of this paper is organized as follows. The basic principle of the naive Bayes is given in section 2. Section 3 introduces the particle swarm optimization algorithm. Section 4 elaborates the proposed PSO-NB method. Section 5 gives the experimental result of comparison of the proposed method with the four other methods. Section 6 summaries this paper.

## 2. NAIVE BAYES CLASSIFICATION

Naive Bayes classifier uses the probabilistic method to predict a class for every instance of data set. The specific working process of Naive bayes is as follows [15-17]:

Let T as the training sample set. Each sample has category labels. Sample set has a total of m classes: C1, C2, ..., Cm. Each sample is represented by an n-dimensional vector X={x1, x2, ..., xn}, and each vector describes n attributes A1, A2, ..., An.

1. Given a simple X, the classifier will predict that X belongs to the highest posterior probability of class. If and only if $P(C_i|X)>P(C_j|X)$, $1\le i, j\le m$, X is predicted to belong to class $C_i$. According to the bayes' theorem, $P(C_i|X)=P(X|C_i)P(C_i)/ P(X)$. Because P(X) is the same for all classes, it only need to find the largest $P(X|C_i)P(C_i)$. The prior probability of class $C_i$ can be calculated. $P(C_i)=s_i/s$, $s_i$ is the number of training samples of class $C_i$, and s is the total number of training samples. If the prior probability of class $C_i$ is unknown, it is usually assumed that the probability of these classes are equal, then $P(C_1)= P(C_2)= \ldots = P(C_m)$, therefore the problem is transformed into how to get maximum $P(X|C_i)$.

2. If the data set has many attribues, the workload of calculating P(X|Ci) is very high. In order to reduce the computational overhead of P(X|Ci), simple assumptions that under certain condition attribute characteristic value is independent of each other. Mathematics is expressed as:

$$P(X \mid C_i) \approx \prod_{k=1}^{n} P(x_k \mid C_i) \tag{1}$$

3. Probability $P(x_1|C_i)$、$P(x_2|C_i)$, …, $P(x_n|C_i)$ can be calculated from the training set. Here $x_k$ refer to the attribute $A_k$ of sample X.

4. For each class, calculating $P(X|C_i)P(C_i)$. If and only if $P(X|C_i)P(C_i)$ is maximum, the classifier prediction sample X belongs to class $C_i$.

## 3. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart integrate group behavior, human decision and simulation of birds' flight to forage behavior, and propose particle swarm optimization algorithm [12]. In this algorithm, each solution of the optimization problem is searching a bird in the space, which is called as the "particle". All particles have a fitness value that is measured by fitness function, and have a speed to decide the direction and distance of particle flight.

Like other evolutionary algorithms, particle swarm optimization algorithm also uses concepts such as the group and

evolution. And the algorithm operates according to individual fitness value. But the particle swarm algorithm does not like other evolutionary algorithms using evolutionary operators for individual. Instead each individual is seen as in n-dimensional search space without weight and volume of the particles, and at a certain flight speed in the search space. According to the individual flight experience and group flight experience, the flight speed can be adjusted dynamically. The PSO algorithm is initialized to a group of random particle (random solutions), and then through the iteration to find the optimal solution. In each generation, particles update themselves by tracking two extremums. The first extremum is the optimal solution for the particle it has ever experienced. This solution is known as individual extremum.

Another extreme is optimal solution for the whole population they have ever experienced at present. This extremum is known as global extremum. Also a part of the whole population can be selected as the particle's neighbors, the extremum that in all neighbors called local extremum. This algorithm makes use of information sharing mechanism in birds, and it is the whole population movement from disorderly to orderly evolution process in problem solving space, and can obtain the optimal solution. Assume the number of particles in group is s. The best position visited by all particles in the group is $P_g(t)$, $P_g(t) \in \{P_0(t), P_1(t), \cdots, P_s(t)\}$,

$$f(P_g(t))=\min\{f(P_0(t)), f(P_1(t)),\cdots, f(P_s(t))\} \quad (2)$$

The evolution equation of particle swarm optimization algorithm can be described as

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(p_{gj}(t) - x_{ij}(t)) \quad (3)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t) + 1 \quad (4)$$

In the equation (3) and (4), the subscript j represents the j dimensional of particle, the subscript irepresents particle i, t represents the t generation. $C_1$ is a cognitive learning factor, $C_2$ is a social learning factor, and they are usually set from 0 to 2.0. When the learning factor $C_1$ and $C_2$ is set to the same value,it means that the particles in the search for itself and social aspects have the same proportion, $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are two independent random function.

When the maximum generation is reached or the designated value of the fitness is achieved, iterations of the PSO are terminated. The PSO is also called continuous PSO in our study.

For dealing with combination optimization problem, Kennedy proposed BPSO(binary particle swarm optimization) algorithm, the value of $x_{id}$ is "0" or "1"(i=1,2, …, n). The value is determined according to the following sigmoid function:

$$S(v_{id}) = \frac{1}{1+e^{-v_{id}}} \quad (5)$$

$$if\,(rand() < S(v_{id}))\ then\ x_{id} = 1\ else\ x_{id} = 0$$

Where $x_{id}=1$ (d=1,2,…,n) represents that the d th feature is selected otherwise this d th feature is not selected at particle i. Rand() represent a random value between 0 and 1. In our study, the number n represents all hyperspectral feature number. Therefore, feature selection problem can be implemented by BPSO (Binary PSO). BPSO is used to select optimal feature subset.

## 4. PSO-NB

After a PSO search process, an attribute subset is selected in the PSO-NB algorithm. Naive Bayes'classification accuracy is used as the fitness function to evaluate alternative subsets of attributes. And after several generations of evolution, the individual with the highest classification accuracy is selected. Fig. (**1**) shows the flow chart of the developed PSO-NB model.

The outline of the proposed algorithm lists as follows:

Step 1. Initialize population X by binary code, each particle is composed of a string of feature selection bit.

Step 2. Remove attributes which are not selected from the training samples attribute to get the training data T, according to the feature of each particle selection bit.

Step 3. Calculate priori probability $P(y_i)$ of each class of training data.

Step 4. Calculate conditional probability $P(x|y_i)$ of each attribute' division.

Step 5. Calculate priori probability $P(y_i)*P(x|y_i)$ of each class.

Step 6. Select the maximum priori probability $P(y_i)*P(x|y_i)$ as the class x belongs to.

Step 7. Calculate the entire sample classification accuracy as the classification accuracy BestAccuracy and its corresponding feature selection Bestf.

Step 8. Determine whether the current accuracy and number of iterations reaches the end of the condition, if reached, go to step 13, otherwise the next step go into the next generation iterative process.

Step 9. For each particle, compare its classification accuracy of current position with the classification accuracy of the best position Bestpi (feature selection bit) which it experienced. If the former is better than the latter, then Bestpi equals the current position.

Step 10. For each particle, compare its classification accuracy of its best position Bestpi with global best position Bestpg. If the former is better than the latter, then Bestpg equals the current position.

Step 11. Update position and velocity of each particle, then go to step3.

Step 12. Repeat step 2 to step 7 to get the current best accuracy BestAccuracy_temp and its corresponding subset of feature selection. If BestAccuracy_temp> BestAccuracy, then BestAccuracy=BestAccuracy _temp, Bestf =f_temp.

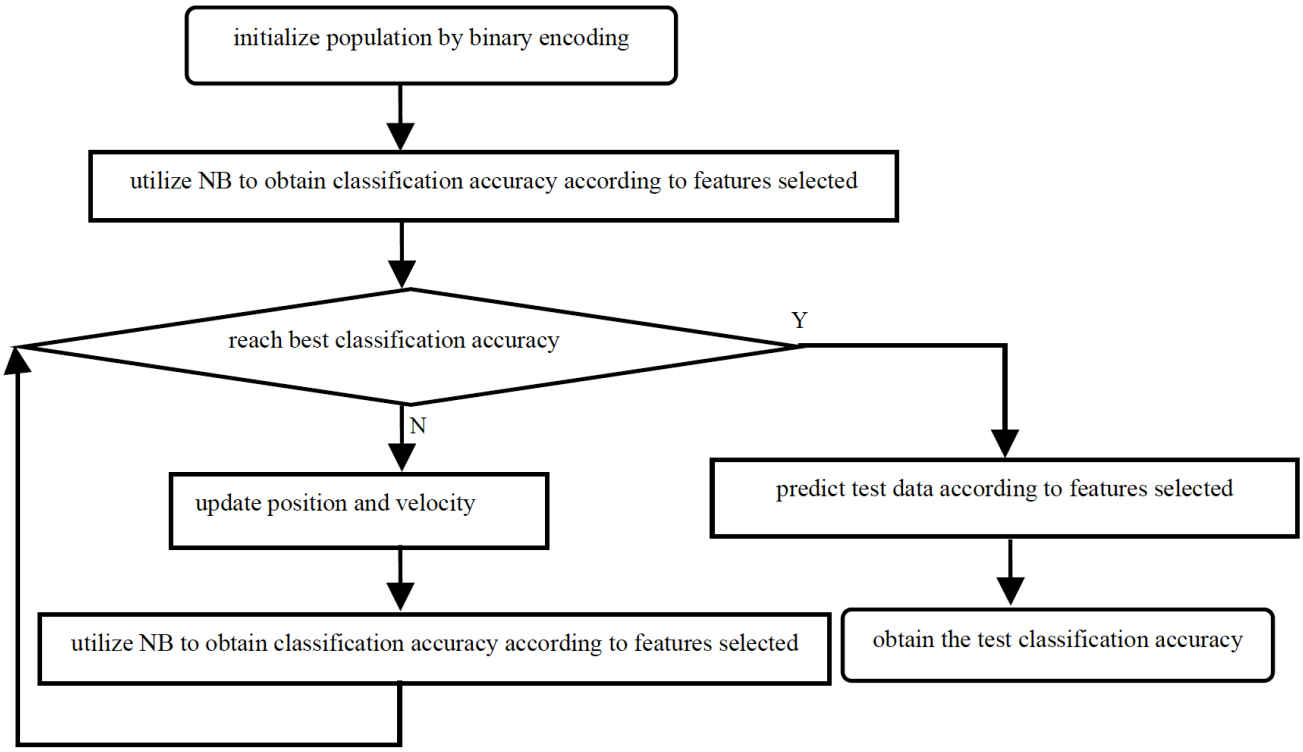


**Fig. (1).** The fiow chat of the PSO-NB model.

Step 13. Repeat step 2 to step 7 according to training data selected by Bestf attributes to get final classified accuracy.

## 5. EXPERIMENT RESULTS

The software platform adopted to develop the PSO-NB algorithm is Matlab2010a.

In order to measure the performance of the developed PSO-NB approach, the following datasets in UC Irvine Machine Learning Repository [18] are used: balance, blood, diabets, glass, haberman, iris, kr_V_kp, Libras_Movement, liver, liverdisorder, p_gene, parkinsons, sonar_all_data, Soybean, splice, tic_tac_toc, wdbc, wine, zoo. Table **1** presents the properties of these datasets.

To guarantee valid results for making predictions regarding new data, the dataset is further randomly partitioned into training sets and independent test sets via a k-fold cross validation [19]. This study used k=10. Each of the 10 subset is used as test data sets in turn, so the program runs 10 times. The final classification accuracy is expressed in the form "mean±standard deviation".

In the PSO-NB algorithm, the parameters are initially set as follows. Iterations GEN is 200, population size NP is 40, $C_1$ and $C_2$ are generated randomly in the interval $[0,+\infty]$.

To verify the excellence of PSO-NB for parameters optimization, the authors design an experiment to compare the PSO-NB algorithm, NB algorithm, C4.5 algorithm, KNN algorithm and Cfs-BestFirst algorithm (that is an embedded feature selection algorithm in WEKA. In WEKA's AttributeSelectedClassifier, evaluator is CfsSubsetEval, search method is BestFirst. And classification algorithm is naive bayes. So, it shorts for Cfs-Best First). The experimental result is shown in Table **2**.

The experiment result is illustrated as follows:

It is obviously seen that the PSO-NB classification performance is much better than Cfs-BestFirst algorithm, NB algorithm, C4.5 algorithm, KNN algorithm, etc. Among the 19 testing data sets, 12 test results of PSO-NB rank first (* is put at the end of the test results). Its average classification accuracy is also significantly higher than that of other data sets.

When comparing PSO-NB algorithm, Cfs-BestFirst algorithm, NB algorithm, it is found that the average classification accuracy of Cfs-BestFirst (77.78) is slightly less than that of NB (78.07). But when compared to PSO-NB average classification accuracy, both of them are inferior, with the gap about 6%. Among the whole 19 testing sets, 16 test results of PSO-NB rank first, thus proves that its classification performance is much better than the former two.

The obtained results clearly confirm the superiority of the PSO-NB algorithm compared to Cfs-BestFirst algorithm, NB algorithm, C4.5 algorithm, KNN algorithm, etc.

**Table 1. Datasets from the UCI repository.**

| Dataset | Number of Instances | Number of Features | Number of Classes | Numeric |
|---|---|---|---|---|
| balance | 625 | 4 | 3 | Y |
| blood | 748 | 4 | 2 | Y |
| diabetes | 768 | 8 | 2 | N |
| glass | 214 | 9 | 6 | N |
| haberman | 306 | 3 | 2 | Y |
| iris | 150 | 4 | 3 | N |
| kr_V_kp | 3196 | 36 | 2 | Y |
| Libras_Movement | 360 | 89 | 15 | N |
| liver | 327 | 6 | 2 | N |
| liverdisorder | 345 | 6 | 2 | Y |
| p_gene | 106 | 57 | 2 | Y |
| parkinsons | 195 | 22 | 2 | N |
| sonar_all_data | 208 | 60 | 2 | N |
| Soybean | 47 | 35 | 4 | Y |
| splice | 1000 | 60 | 2 | Y |
| tic_tac_toc | 958 | 9 | 2 | Y |
| wdbc | 569 | 30 | 2 | N |
| wine | 178 | 12 | 3 | N |
| zoo | 101 | 16 | 7 | Y |

**Table 2. Experimental results.**

| Dataset | PSO-NB(%) | Cfs-BestFirst(%) | NB(%) | C4.5(%) | KNN(%) |
|---|---|---|---|---|---|
| balance | 70.81±5.71 | 63.52±4.97 | 90.62±1.34 | 77.76±3.85 | 86.99±2.83 |
| blood | 70.27±5.14 | 76.27±3.08 | 75.28±3.47 | 78.20±3.71 | 77.18±3.45 |
| diabetes | 80.13±4.41* | 77.06±4.70 | 75.75±5.32 | 74.49±5.27 | 73.86±4.61 |
| glass | 68.10±8.41* | 48.11±9.97 | 47.84±8.74 | 68.08±9.28 | 66.18±8.22 |
| haberman | 70.67±6.99 | 74.21±5.48 | 75.06±5.42 | 71.05±5.20 | 70.49±5.17 |
| iris | 97.33±3.44* | 96.20±4.26 | 94.87±5.26 | 94.73±5.30 | 95.73±4.60 |
| kr_V_kp | 93.20±1.00 | 92.97±1.43 | 84.00±1.96 | 99.34±0.41 | 95.83±1.10 |
| Libras_Movement | 66.39±10.51* | 66.94±7.56 | 64.42±7.67 | 45.72±7.92 | 63.56±7.39 |

Table **2**. contd..

| Dataset | PSO-NB(%) | Cfs-BestFirst(%) | NB(%) | C4.5(%) | KNN(%) |
|---|---|---|---|---|---|
| liver | 74.06±8.08* | 62.54±7.32 | 63.00±7.75 | 63.28±7.57 | 65.32±7.61 |
| liverdisorder | 71.47±5.55* | 56.15±6.48 | 54.89±8.83 | 65.84±7.40 | 60.48±7.92 |
| p_gene | 98.00±4.22* | 85.35±10.57 | 81.10±11.76 | 80.05±12.04 | 72.34±13.14 |
| parkinsons | 94.74±3.51* | 77.83±8.92 | 70.14±9.24 | 84.69±7.96 | 92.73±5.27 |
| sonar_all_data | 90.50±4.97* | 67.62± 9.26 | 67.71±8.66 | 73.61±9.34 | 82.28±9.12 |
| Soybean | 100.00±0.00* | 100.00±0.00 | 98.00±6.03 | 97.65±7.12 | 100.00±0.00 |
| splice | 91.70±1.42 | 82.32±3.61 | 83.48±3.33 | 94.44±2.54 | 69.13±3.78 |
| tic_tac_toc | 77.16±4.21 | 65.40±0.50 | 71.28±2.45 | 89.15±3.93 | 83.69±2.66 |
| wdbc | 96.79±1.84 | 94.55±3.03 | 93.31±3.30 | 93.27±3.55 | 96.88±2.25 |
| wine | 97.65±3.04* | 94.78±4.58 | 95.60±4.29 | 92.98±5.77 | 94.21±4.90 |
| zoo | 99.00±3.16* | 95.95±5.09 | 96.95±4.75 | 92.61±7.33 | 95.05±6.70 |
| mean | 84.63±4.51 | 77.78±5.31 | 78.07±5.77 | 80.89±6.08 | 81.05±5.30 |

## CONCLUSION

In this paper, aiming at the shortcomings of the independent assumption of naive Bayes classification, combined with particle swarm optimization, a novel pso-nb algorithm is proposed. It selects an attribute subset through the whole space of attributes by carrying out a pso search process. Simulation results show that the proposed algorithm greatly enhances classification accuracy of naive Bayes.

## CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

## ACKNOWLEDGEMENTS

This research is supported by National Natural Science Foundation of China under Grant NO. 61273303. The authors thank professor Jiang Liangxiao for his very useful comments and suggestions.

## REFERENCES

[1] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning* vol. 29, no. 2-3, pp. 131-163, 1997.

[2] J. Cheng, and R. Greiner, "Comparing Bayesian network classifiers," In: *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence,* Morgan Kaufmann Publishers Inc., pp. 101-108, 1999.

[3] J. Zhe, "Research and Implementation of Genetic Algorithm Based BN Augmented Naïve-Bayes Classifier," Jilin University, 2006.

[4] Q. Feng, R. Shiliu, and C. Zekai, "Attribute weighted Naïve Bayes classification," *Computer Engineering and Applications,* vol. 44, no. 6, pp. 107-109, 2008.

[5] Z. Wen, and Z. Huaxiang, "Naïve Bayesian ensemble classifier using attribute weighting," *Computer Engineering and Applications,* vol. 46, no. 29, pp. 144-146, 2010.

[6] J. Liangxiao, Z. Harry, and C. Zhihua, "A Novel Bayes Model: Hidden Naive Bayes," *IEEE Transactions On Knowledge And Data Engineering,* vol. 21, no. 10, pp. 1361-1371, 2009.

[7] Y. Guangzu, and W. Guojun, "New Selective Naïve Bayes Algorithm," *Science Technology and Engineering,* vol. 4, pp. 978-980, 2009.

[8] Z. Jing, W. Jianmin, and H. Huacan, "A novel feature reduct algorithm based on feature correlation," *Computer Engineering and Applications,* vol. 41, no. 28, pp. 55-57, 2006.

[9] S. Casale, A. Russo, and S. Serrano, "Analysis of robustness of attributes selection applied to speech emotion recognition," In: *Proceedings of the 18th European Signal Processing Conference (EUSIPCO'10). EURASIP, Aalborg, Denmark,* 2010.

[10] M. Hall, E. Frank, and G. Holmes, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter,* vol. 11, no. 1, pp. 10-18, 2009.

[11] C. A. Ratanamahatana, and D. Gunopulos, "Scaling up the naive Bayesian classifier: Using decision trees for feature selection," In: *Proceedings of Workshop on Data Cleaning and Preprocessing (DCAP), at IEEE International Conference on Data Mining (ICDM 2002), Maebashi, Japan*. 2002.

[12] Y. J. Kenned, and R. Eberhart, "Particle swarm optimization," In: *IEEE International Conference on Neural Networks, Perth: IEEE Neural Networks Society,* pp.1942-1948, 1995.

[13] L. Yuanning, W. Gang, and C. Huiling, "An improved particle swarm optimization for feature selection," *Journal of Bionic Engineering*, vol. 8, pp. 191-200, 2011.

[14] E. Correa, A. Freitas, and C. Johnson, "Particle swarm for attribute selection in Bayesian classification: an application to protein function prediction," *Journal of Artificial Evolution and Applications,* vol. 2008, pp. 1-12, 2008.

[15] M. Kantardzic, *Data Mining-Concepts, Models, Methods, and Algorithms*, IEEE Press, Wiley-Interscience, 2003.

[16] H. Jiawei, and K. Micheline, *Data Mining: Concepts and Techniques,* Elsevier: USA, 2006.

[17] J. Lin, and J. Yu, "Weighted Naive Bayes classification algorithm based on particle swarm optimization," In: *Proceedings of Communication Software and Networks (ICCSN),* pp. 444-447, 2011.

[18] S. Hettich, C. Blake, and C. Merz, UCI repository of machine learning databases , Department of Information and Computer Science,University of California, Irvine, CA. 1998. (http//www.ics.uci.edu/~mlearn /MLRepository.html).

[19] C. L. Huang, and J. F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing,* vol. 8, no. 4, pp. 1381-1391, 2008.

Received: November 28, 2014 Revised: January 09, 2015 Accepted: January 20, 2015