Artificial Bee Colony based Data Mining Algorithms for Classification Tasks

Mohd Afizi Mohd Shukran (Corresponding author)
Faculty of Science & Defense Technology, National Defense University of Malaysia (NDUM)
Sungai Besi Camp, 57000 Kuala Lumpur, Malaysia
Tel: 60-3-9051-3400 E-mail: afizi@upnm.edu.my

Yuk Ying Chung
School of Information Technologies, Faculty of Engineering & IT
University of Sydney, 2006 NSW, Australia
Tel: 61-2-9036-9109 E-mail: vchung@it.usyd.edu.au

Wei-Chang Yeh
Advanced Analytics Institute, Faculty of Engineering and Information Technology
University of Technology Sydney, PO Box 123 Broadway, NSW 2007, Australia
Tel: 61-2-9036-9109 E-mail: yeh@ieee.org

Noorhaniza Wahid
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat Johor
Tel: 60-7-453-8215 E-mail: nhaniza@uthm.edu.my

Ahmad Mujahid Ahmad Zaidi
Department of Mechanical Engineering, Faculty of Engineering
National Defense University of Malaysia (NDUM)
Sungai Besi Camp, 57000 Kuala Lumpur, Malaysia
Tel: 60-3-9051-3400 E-mail: mujahid@upnm.edu.my

Received: June 1, 2011 Accepted: June 26, 2011 doi:10.5539/mas.v5n4p217

Abstract

Artificial Bee Colony (ABC) algorithm is considered new and widely used in searching for optimum solutions. This is due to its uniqueness in problem-solving method where the solution for a problem emerges from intelligent behaviour of honeybee swarms. This paper proposes the use of the ABC algorithm as a new tool for Data Mining particularly in classification tasks. Moreover, the proposed ABC for Data Mining were implemented and tested against six traditional classification algorithms classifiers. From the obtained results, ABC proved to be a suitable candidate for classification tasks. This can be proved in the experimental result where the performance of the proposed ABC algorithm has been tested by doing the experiments using UCI datasets. The results obtained in these experiments indicate that ABC algorithm are competitive, not only with other evolutionary techniques, but also to industry standard algorithms such as PART, SOM, Naive Bayes, Classification Tree and Nearest Neighbour (kNN), and can be successfully applied to more demanding problem domains.

Keywords: Artificial bee colony algorithm, Data mining, Local search strategy

1. Introduction

As the size of digital information grows exponentially, large volumes of raw data need to be extracted. Nowadays, there are several methods to customise and manipulate data according to our needs. The most common method is to use Data Mining (DM). DM has been used in previous years for extracting implicit, valid, and potentially useful knowledge from large volumes of raw data (Sousa, Silva, & Neves). The extracted knowledge must be accurate, readable, comprehensible, and ease of understanding. Furthermore, the process of data mining is also called as the process of knowledge discovery which has been used in most new inter-disciplinary area such as database, artificial intelligence statistics, visualization, parallel computing and other fields (Wang & Huang, 2007). One new and extremely powerful algorithm used in DM is Evolutionary algorithms (EA) (Tan, Teoh, Yu, & Goh, 2009; Whitley, 2001). This includes biology inspired algorithms such as Genetic Algorithms (GA) (Sumida, Houston, McNamara, & Hamilton, 1990), Differential Evolution Algorithms (DE) (Storn & Price, 1997) and swarm based approaches like Ant Colonies (Colorni, Dorigo, & Maniezzo, 1991), and Particle Swarm Optimizations (PSO) (Kennedy, 2006). In this paper, we propose frameworks for DM as classifier using Artificial Bee Colony (ABC) algorithm (D. Karaboga, 2005). We have implemented the ABC algorithm and evaluate its performance by comparing to other most common classifiers such as PART, DPSO, SOM, Naive Bayes, Classification Tree and Nearest Neighbour (kNN).

The rest of this paper is organized as follows. Section 2 presents a brief overview of previous work on Optimization algorithm which relates to Data Mining. Section 3 describes the methodology of ABC from optimization perspectives. Section 4 and 5 are dedicated to describe the proposed ABC for Data Mining and presenting the experimental setup and results. Conclusions are given in Section 6.

2. Related Work

Many approaches, methods and goals have been tried out for DM. One approach is using GA in DM. GAs have been applied widely to data mining for classification. For example, GA algorithm has been used in oil collecting vehicle routing problem which is based on the insertion of local search and DM modules in order to discover good features in the best solutions found so far and to apply them in the generation of new solutions (Santos, Ochi, Marinho, & Drummond, 2006). Another example of applying GA in DM is Sikora and Piramuthu proposed doing both the tasks of mining and feature selection simultaneously by evolving a binary code alongside the chromosome structure used for evolving the rules in (Sikora & Piramuthu, 2007).

Another new evolution method used in DM is Particle swarm optimizer (PSO) (Kennedy, 2006). PSO is a new evolutionary algorithm, which simulates the coordinated motion in flocks of birds. Sousa, Silva, and Neves (Sousa, et al., 2004) proposed the use of PSO for data mining. PSO can achieve the rule discovery process. However, PSO has evolved in DM in which it can reduce complexity and speed up the data mining process. For instance, Yeh, Chang and Chung (Yeh, Chang, & Chung, 2009) have applied hybrid approach for mining breast cancer pattern using PSO. Moreover, each particle is coded in positive integer numbers and has a feasible system structure. The proposed DPSO from Chen and Hsu (Chen & Hsu, 2006) for data mining algorithm can overcome the drawbacks of GAs.

Besides that, Ant Colony Optimization (ACO) has been widely used in DM. ACO is a branch of newly developed form of artificial intelligence called Swarm Intelligence (SI). ACO algorithm is inspired by social behaviour of ant colonies. Marco Dorigo and colleagues introduced the first ACO algorithms in the early 1990's (Blum, 2005). The development of these algorithms was inspired by the observation of ant colonies. An example of ACO application is done via prediction of protein activity proposed by Nemati et al. (Nemati, Basiri, Ghasem-Aghaee, & Aghdam, 2009). The significant approach involves combining genetic algorithms (GA) and ant colony optimization (ACO) in prediction postsynaptic activity of proteins. The hybrid algorithm takes the advantages of both ACO and GA methods by complement each other for feature selection in protein function prediction. Furthermore, Nicholas (Nicholas & Alex, 2008) proposed a hybrid PSO/ACO algorithm for discovering classification rules. In PSO/ACO, the rule discovery process is divided into two separate phases. In the first phase, ACO discovers a rule containing nominal attributes only. In the second phase, PSO discovers the rule potentially extended with continuous attributes.

From above, we found that many optimization algorithms have been used for classification tasks. From the best of our knowledge, previous researches on ABC algorithm have only focused on optimization but none of them is for classification tasks. Therefore, this paper is the first time to apply the novel ABC algorithm for the classification applications.

3. Artificial Bee Colony Algorithm (ABC)

ABC algorithm is a new swarm intelligent algorithm, which proposed by Karabog in Erciyes University of Turkey in 2005 (D. Karaboga, 2005). Since ABC algorithm is simple in concept, easy to implement, and has fewer control parameters, it has been widely used in many optimization applications such as protein tertiary structures (H.A A. Bahamish, R. Abdullah, & R.A. Salam, 2009), digital IIR filters (N. Karaboga, 2009), artificial neural networks (D. Karaboga & Akay, 2005) and others.

The minimal model of foraging selection that leads to the emergence of collective intelligence of honey bee swarms consists of three essential components: food sources, employed foragers and unemployed foragers. There are two basic behaviours: recruitment to a food source and the abandonment of a food source (D. Karaboga, 2005).

1) Food sources: it represents a position of solution of optimization problem, the profitability of food source are expressed as fitness of the solution.

2) Unemployed foragers: there are two types of them, scouts and onlookers. Their main task is exploring and exploiting food source. At the beginning, there are two choices for the unemployed foragers: (i). it becomes a scout – randomly search new food sources around the nest; (ii). It becomes an onlooker – determine the nectar amount of food source after watching the waggle dances of employed bee, and select food source according to profitability.

3) Employed foragers: the honeybees found food source, which also known as the employed bees, are equal to the number of food sources. The employed bees store the food source information and share with others according to a certain probability. The employed bee will become a scout when food source has been exhausted.

Basically, there are two important function supports the algorithm:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad \text{Eq. (1.1)}$$

$$V_{ij} = X_{ij} + \Phi_{ij}(X_{ij} - X_{kj}) \qquad \text{Eq. (1.2)}$$

Where P_i is the probability value associated with i^{th} food source that calculated by the Eq.1.1. An onlooker bee selects a food source relying on P_i In this equation, fit_i represents i^{th} food source's nectar amounts, which is measured by employed bees and *SN* is the number of food source which is equal to the number of employed bees. (In the real-world problems, *SN* usually represents the number of possible solutions.)

In ABC algorithm, the artificial bees need to do the local search for finding the new possible solution. Eq.1.2 shows the local search strategy of the original ABC optimization algorithm. V_{ij} is the new candidate food position produced by this equation where k € [1,2,…,*SN*] and j € [1,2,…,*D*] are randomly chosen parameters, but k has to be different from j. Thus X_{ij} and X_{kj} represent the different old food source positions. The difference between these two positions is the distance from one food source to the other one. *SN* is the number of employed bees as the previous described and *D* is the number of optimization parameters. Φ_{ij} is a random number between [-1, 1] and controls the distance of a neighbour food source position around X_{ij}.

The most crucial factor in ABC is the processing of greedy selection which means if the new food has equal or better nectar than the previous source; it will replace the previous one in the memory. Otherwise, the previous one is retained. In other words, a greedy selection mechanism is employed as the selection operation between the previous and the current food sources.

Beside the *SN* control parameter, there are two other parameters used in the basic ABC algorithm: the limitation value of food source position and the maximum number of searching cycle. The users should carefully choose these two values, since the larger number will slow down the optimization process significantly, and the small value cannot find a good solution for the requirement.

4. Proposed ABC Algorithm for Data Mining

ABC algorithm is a swarm intelligence based algorithm simulating the foraging behaviour of a honey bee colony. Most previous research using ABC algorithm have been recently introduced for global optimization. For instance, Karboga (N. Karaboga, 2009) implements ABC algorithm to design digital Infinite Impulse Response (IIR) filters and its performance had compared with other conventional optimization algorithms (LSQ-nonlin) and particle swarm optimization (PSO). Moreover, Bahamish et al. (H. A. A. Bahamish, R. Abdullah, & R. A. Salam, 2009) also uses had applied the ABC algorithm to search the protein conformational search space to find the lowest free energy conformation which resulted the findings of the lowest free energy conformation for a test protein (i.e. Met enkephaline) using ECEPP/2 force fields. ABC algorithm had applied to solve the leaf-constrained minimum spanning tree problem in literature by (Singh, 2009).

Moreover, Basturk and Karaboga (D. Karaboga & Basturk, 2007; D. Karaboga & Basturk, 2008) compared the performance of ABC algorithm with those of GA, PSO and PS-EA; and Differential Evolution (DE), PSO and Evolutionary Algorithm (EA) on a limited number of test problems. Nevertheless, ABC algorithm also has been evolved through time by solving optimization problems in (D. Karaboga & Basturk, 2007). In neural networks, ABC has been used as optimization solution for training neural networks (D. Karaboga, Akay, & Ozturk, 2007). As ABC algorithm precedes the expectation in solving optimization solutions, new approach based on intelligent behaviour of honeybee swarm emerges. This can be found (Yang, 2005) and (D. Karaboga & Akay, 2005) in the literature. Moreover, researches took another step in developing ABC algorithm as optimization solutions in two-dimensional numeric functions (Yang, 2005). As for solving multi variable problems in optimization process, Karaboga proves ABC algorithm has the ability to optimize multi-variable and multi-modal numerical functions (D. Karaboga, 2005).

Based on previous research mentioned in the earlier paragraphs in this section, we found that ABC algorithm has only been used for the optimization purposes. Therefore, this paper aims at applying this novel ABC algorithm for classification applications. The accuracy of the ABC classifier will be compared with other common classifier such as PART, SOM, Naive Bayes, Classification Tree and Nearest Neighbour (kNN).

Firstly, the main components of the proposed ABC algorithm for classification tasks will be introduced as follows. There are six main components in this proposed algorithm. There are **(1) the rule format**, **(2) fitness function**, **(3) exchanged local search strategy**, **(4) rule discovery**, **(5) rule pruning** and **(6) prediction strategy**. These main components will be described in the following sections:

(1) Rule Format

In classification method, a classification rule for each attribute contains two parts: the antecedent and the consequent. The following format can be seen in Figure 1.

Where feature 1 to feature N are all attributes of the dataset. Each attribute has its lower bound which is the lowest value for this rule and upper bound which is the highest value for this rule. There are three other values associated with the classification rule: predictive class (Class X), the fitness value and the cover percentage of the rule. These three numbers have close relationship with fitness function and prediction strategy. This will be explained more detail in the following sections.

From the above explanation, every classification rule is designed as the structure below (C Language version):

```
Struct RuleSet{
    Double * lowb;//the lower bound for all attributes
    Double * upb;        //the upper bound for all attributes
    Char cName[50];      //Class X
    Double prec;         //Cover percentage
    Double * fitvalue;   //Fitness value
};
```

(2) Fitness Function

To evaluate the fitness value, the fitness function will be used for the classification instead of measuring the nectar amount. Its representation is defined as below:

$$\text{Fitness Value} = \frac{TP}{TP+FN} \times \frac{TN}{TN+FP} \qquad \text{Eq. (2.1)}$$

Where TP, FN, FP and TN are the number of different record types and representing of True Positives, False Negatives, False Positives and True Negatives associated with the rule respectively. Before introducing these four values, two important concepts will be explained:

- When the algorithm examines the type of the record, it will measure every feature in the record. If the value of a feature is between the lower bound and upper bound for this feature, it means the feature can be covered by the rule. If all features for a record can be covered by the rule, it means the record can be covered by the rule.
- If the class of the evaluated record is equal to the predictive class by the rule, this denote that the record has he class predicted by the rule.

i. True positives (TP): the number of records covered by the rule that have the class predicted by the rule;

ii. False negatives (FN): the number of records not covered by the rule but they have the class predicted by the rule;

iii. False positives (FP): the number of records covered by the rule but their class do not predicted by the rule;

iv. True negatives (TN): the number of records not covered by the rule and that do not have the class predicted by the rule.

(3) Exchanged Local Search Strategy

When an employed bee does not meet the requirement or reach the maximum cycle number, it needs to move to a new food source followed by the local search strategy. The original ABC algorithm used the Eq.1.2 to implement the local search method. However, it will consume an enormous amount of time when the classified dataset contains large number of data and is not suitable for the classification applications. In consideration of time is the significant factor of doing the classification and improving the accuracy, we propose a new simpler local search strategy which named "Exchanged" to replace the original local search strategy.

$$V_{ij} = X_{kj} \qquad \text{Eq. (2.2)}$$

Where V_{ij} represents the position of the new food source and X_{kj} stands for the neighbour of previous food source. The number of i and k are between 1 and SN, but k has to be different value from i. In addition, j is the number of dimension. In the classification task, the dimension of the dataset equals to the number of features in the dataset. k € $[1,2,\ldots,SN]$ and j € $[1,2,\ldots,D]$ are randomly chosen parameters. When we put this strategy into classification area, the Eq.2.2 is modified to the following functions:

V_{ij}'s lower bound = X_{k1j}'s lower bound Eq. (2.3)

V_{ij}'s upper bound = X_{k2j}'s upper bound Eq. (2.4)

Where K_1 and K_2 are two random numbers and do not equal to i.

To evaluate the performance of these two local search methods, we have used the UCI datasets for testing. The detail about UCI test datasets and configuration of control parameter will be explained in Section 3. The comparison result between the ABC with original and proposed "Exchanged" local search strategy has shown in Table 1.

From Table 1, we used the bold to highlight the best result between the original and "Exchanged" local search strategies. For the Breast Cancer, Zoo, Can and Monk datasets, the new local search can reach the better result in all four aspects, mean, standard deviation, maximum and minimum. There were only standard deviation and minimum in Soybean dataset and maximum in Irish dataset got the lower value for new strategy.

From Figure 2 testing result, we can conclude that the proposed "Exchanged" local search can outperform the original local search in ABC algorithm for data classification applications.

(4) Rule Discovery

The aim of classification rule mining is to find a set of rules which can identify the specific class from different groups. Thus, the rule discovery phase is the most important for classification algorithm, since rule sets is the outcome of this phase. The rule discovery on ABC algorithm has show in the flowchart in Figure 5.

In the intialization stage, we set lower bound value and upper bound value for every attribute. The procedure is defined in Eq.2.5 and Eq.2.6 as below:

Lower bound = $f - k_1$ x $(F_{max} - F_{min})$ Eq. (2.5)

Upper bound = $f + k_2$ x $(F_{max} - F_{min})$ Eq. (2.6)

For these two equations, F_{max} and F_{min} are, respectively, the maximum value and the minimum value of the feature respectively. The difference between them means the range of the feature. f is the original value of the feature. k_1 and k_2 are two random values between 0 and 1.

The classification rule mining algorithm can automatically discovers the rules for each class. For the selected class, it will find the rules iteratively until the rule set can cover all instances belong to that class. Every single rule abides by the rule structure and the rule set consists of many rules.

(5) Rule Pruning

After all classes have been processed and all rule sets have been generated, every rule will be put into the rule pruning procedure. The main goal of rule pruning is to remove the redundant feature limitation that might have been unnecessary included in the rule sets. Since some irrelative attributes will negatively influence the classification result, the rule pruning can increase the accuracy. The process of rule pruning has been shown in Figure 4. This process will be repeated until all rules in the rule set are evaluated.

(6) Prediction Strategy

The pruned rule set will be used to predict the new data which their classes are unknown. But sometime, one testing data record will be covered by more than one rule for different class. When this happened, the prediction strategy will determine which class should be predicted. There are three main steps for the prediction approach, they have specified as follow:

1) Calculate the prediction value for all rules which cover the test data record;
2) Accumulate these prediction value according to different possible class;
3) Select the class which has the highest prediction value as the final class.

After the procedure of prediction strategy, the core is the prediction function which is used to compute the prediction value for each rule. It is defined in Eq.2.7 as below:

$$\text{prediction value} = (\alpha \times \text{rule fitness value}) + (\beta \times \text{rule cover percentage}) \qquad \text{Eq. (2.7)}$$

Where α and β are two weighted parameters associated with rule fitness value and rule cover percentage respectively, $\alpha \in [0,1]$ and $\beta = (1-\alpha)$. The Eq.2.1 can calculate the fitness value for each rule. The rule cover percentage defines that the proportion of the records which covered by the rule that have the class predicted by the rule (TP). It is calculated by the expression shows in Eq.2.8:

$$\text{Cover percentage} = \frac{TP}{N} \qquad \text{Eq. (2.8)}$$

where N is the total number of the records which belong to the predicted class by the rules.

The prediction strategy balanced the effect of fitness value and cover percentage for the final predicted class. We need to choose the value of α and β carefully, since they will affect the classification accuracy.

Our new approach has provided a new mechanism of classification rule mining based on ABC optimization algorithm. The summary of the main procedure of proposed ABC data mining algorithm has shown in Figure 3.

In Figure 3, each rectangle represents the single in stage in classification data mining and the rhombus indicates the data structure or data set generated by the previous stage. In the initialization step, all control parameters are set, such as the number of colony size, the maximum cycle and the bound value limitation. We used K-fold cross-validation for assessing the accuracy of our testing results. In K-fold cross-validation, the original data set is randomly divided into K subsets. The training and testing period will execute K times. For each time, a single subset is used as the validation data for testing (Testing set), and the remaining (K-1) subsets are retained as training data. In addition, each of the K subsets is employed only once as the validation data. The 10-fold cross-validation is the most commonly used algorithm to test the accuracy of classification (McLachlan, Kim-Anh, & Christophe, 2004). After two data sets have separated, the training set will be used during the training period and the aim is to find the classification rule set. Finally, to evaluate the algorithm, we calculate the accuracy for each validation data set and compute the average of K time validations as the final accuracy.

5. Experimental Design and Result

For measuring the performance of the proposed ABC data mining algorithm, we have conducted experiment with different datasets that are selected from the UCI Machine Learning Repository. There are 187 data sets currently maintained on the homepage of UCI Machine Learning research group ("UC Irvine Machine Learning Repository,"). We have tested on the six popular data sets in the experiment: Breast Cancer, Iris, Zoo, Monk, Can and Soybean. To show the performance of ABC classification algorithm, we have compared it with five other classification algorithms from two data mining software tools which named Orange (J. & B., 2005) and Weka (Mark, et al., 2009). We have chosen KNN, Classification Tree, and NB from Orange 1.0 and PART and

SOM from Weka 3.6.1. These datasets and algorithms have been widely used in data mining research. The details about six UCI datasets and the configuration of computer system we used for the experiments are showed in Table 4 and Table 5 respectively:

5.1 ABC Settings

In ABC data mining algorithm, we have used the common value (10) for the number of fold to calculate the cross validation. During the rule discovery period, there are three basic control parameters: the number of food sources which is equal to the number of employed or onlooker bees (*SN*), the value of limit and the maximum cycle number. Furthermore, they are the most important parameters in this algorithm, since these three parameters will determine the time consuming of finding the rule set and the quality of rule set. Therefore, suitable values are needed to balance these two factors. The values of limit are automatically assigned to the maximum and the minimum value for each feature. With several times attempt, *SN* is set to 20 and the maximum cycle number set to 50.

After the final rule set has generated, two parameters are required by the class prediction, which are quality weight (α) and coverage weight (β). Both of them were set to 0.5 for the experiments in Section 5.2.

5.2 Experimental Result

When we did the testing of UCI dataset, we separated the UCI datasets experiment into two parts. First, we focus on whether the ABC classification algorithm can handle the UCI dataset. The result of first stage is shown in Table 2.

Where the 1 to 10 rows represent the training and testing result of each fold and the last row shows the average value of 10-fold cross validation experiment. According to Table 2 results, the proposed ABC classification algorithm can successfully classify the UCI datasets and give a good result.

For each dataset, we have done 10 times testing to analyse the result by using statistic method. We calculated mean, the lowest, the highest accuracy and the standard deviation. The testing results have been shown in Table 3.

Then, we compared ABC classification performance using the original local search and the proposed Exchanged local search with other popular classification algorithms that include PART, SOM, Naive Bayes, KNN and Classification Tree in Table 6.

In order to show the performance of the ABC classification algorithm in Table 6 more clearly, we have reproduced it in bar chart formats which have been shown from Figure 6 to Figure 11. From above results, we found that ABC classification algorithm can have the best performance for UCI datasets: Breast Cancer, Iris, Zoo, Monk and Soybean which out of all six experimental datasets we used in the experiments. Especially for the Soybean dataset, it is obviously that the accuracy of ABC data mining algorithm is higher than other 5 algorithms by at least 1%. For the only one dataset, “Can”, which did not get the top result, our proposed algorithm can obtain the second best performance.

The above testing results can clearly prove that the proposed ABC data mining algorithm can handle the data classification task successfully and obtain the superior result when compared with other data mining software tools such as “Orange” and “Weka”.

6. Conclusion

The Artificial Bee Colony (ABC) algorithm is a new searching algorithm under Swarm Intelligence technology. Many approaches, methods and goals have been tried out for DM. Biology inspired algorithms such as GA and swarm based approaches like Ant Colonies have been successfully used in DM. From the best of our knowledge, previous researches have never applied the ABC algorithm in DM. In this paper we propose a novel data mining approach based on ABC searching algorithm for classification task. For adapting classification task, our contributions for this new ABC data mining algorithm are in the following aspects: rule format, fitness value function, local search strategy and prediction strategy. In the local search part, we have proposed a novel strategy which named **Exchanged** local search instead of the original one in (D. Karaboga, 2005) for improving the classification accuracy and can obtain a superior result for testing datasets. For avoiding ambiguous situation happen, we designed two parameters: (1) quality weight and (2) coverage weight, for predication strategy. These two parameters will be used when a testing record can be covered by more than two rules for different classes. Furthermore, the proposed ABC algorithm has been tested using six UCI datasets. We have compared the proposed algorithm with other five data mining algorithms which selected from data mining softwares tools: “Orange” and “Weka”. It has been proved that the proposed ABC data mining algorithm can obtain the superior result for most of these (five out of six) UCI testing datasets. Therefore, we can conclude that the proposed ABC algorithm for DM can obtain competitive result against five traditional data mining algorithms and can be considered as useful and accurate classifier.

References

Bahamish, H. A. A., Abdullah, R., & Salam, R. A. (2009). Protein Tertiary Prediction Using Artificial Bee Colony Algorithm.

Bahamish, H. A. A., Abdullah, R., & Salam, R. A. (2009). *Protein Tertiary Structure Prediction Using Artificial Bee Colony Algorithm.* Paper presented at the Modelling & Simulation, 2009. AMS '09. Third Asia International Conference on Modelling & Simulation.

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews, 2*(4), 353-373.

Chen, T.-C., & Hsu, T.-C. (2006). A GAs based approach for mining breast cancer pattern. *Expert Systems with Applications, 30*(4), 674-681.

Colorni, A., Dorigo, M., & Maniezzo, V. (1991). *Distributed Optimization by Ant Colonies.* Paper presented at the European Conference on Artificial Life.

J., D., & B., Z. (2005). Orange: From Experimental Machine Learning to Interactive Data Mining. from www.ailab.si/orange

Karaboga, D. (2005). *An Idea Based on Honey Bee Swarm for Numerical Optimization* (TECHNICAL REPORT-TR06).

Karaboga, D., Akay, B., & Ozturk, C. (2007). Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks *Modeling Decisions for Artificial Intelligence* (pp. 318-329).

Karaboga, D., & Akay, B. B. (2005). *An artificial bee colony (abc) algorithm on training artificial neural networks* (Technical Report TR06): Erciyes University, Engineering Faculty, Computer Engineering Department.

Karaboga, D., & Basturk, B. (2007). Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems *Foundations of Fuzzy Logic and Soft Computing* (pp. 789-798).

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459-471.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8*(1), 687-697.

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital IIR filters. *Journal of the Franklin Institute, 346*(4), 328-348.

Kennedy, J. (2006). Swarm Intelligence (pp. 187-219).

Mark, H., Eibe, F., Geoffrey, H., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1).

McLachlan, G. J., Kim-Anh, D., & Christophe, A. (2004). Analyzing Microarray gene expression data

Nemati, S., Basiri, M. E., Ghasem-Aghaee, N., & Aghdam, M. H. (2009). A novel ACO-GA hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications, 36*(10), 12086-12094.

Nicholas, H., & Alex, A. F. (2008). A hybrid PSO/ACO algorithm for discovering classification rules in data mining. *J. Artif. Evol. App., 2008*, 1-11.

Santos, H. G., Ochi, L. S., Marinho, E. H., & Drummond, L. M. A. (2006). Combining an evolutionary algorithm with data mining to solve a single-vehicle routing problem. *Neurocomputing, 70*(1-3), 70-77.

Sikora, R., & Piramuthu, S. (2007). Framework for efficient feature selection in genetic algorithm based data mining. *European Journal of Operational Research, 180*(2), 723-737.

Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing, 9*(2), 625-631.

Sousa, T., Silva, A., & Neves, A. (2004). Particle Swarm based Data Mining Algorithms for classification tasks. *Parallel Computing, 30*(5-6), 767-783.

Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. [10.1023/A:1008202821328]. *Journal of Global Optimization, 11*(4), 341-359.

Sumida, B. H., Houston, A. I., McNamara, J. M., & Hamilton, W. D. (1990). Genetic algorithms and evolution. *Journal of Theoretical Biology, 147*(1), 59-84.

Tan, K. C., Teoh, E. J., Yu, Q., & Goh, K. C. (2009). A hybrid evolutionary algorithm for attribute selection in data mining. *Expert Systems with Applications, 36*(4), 8616-8630.

UC Irvine Machine Learning Repository. from http://archive.ics.uci.edu/ml/

Wang, G. Q., & Huang, D. (2007). The summary of the data mining technology. *Microcomputer Application Technology, 2*, 9-14.

Whitley, D. (2001). An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and Software Technology, 43*(14), 817-831.

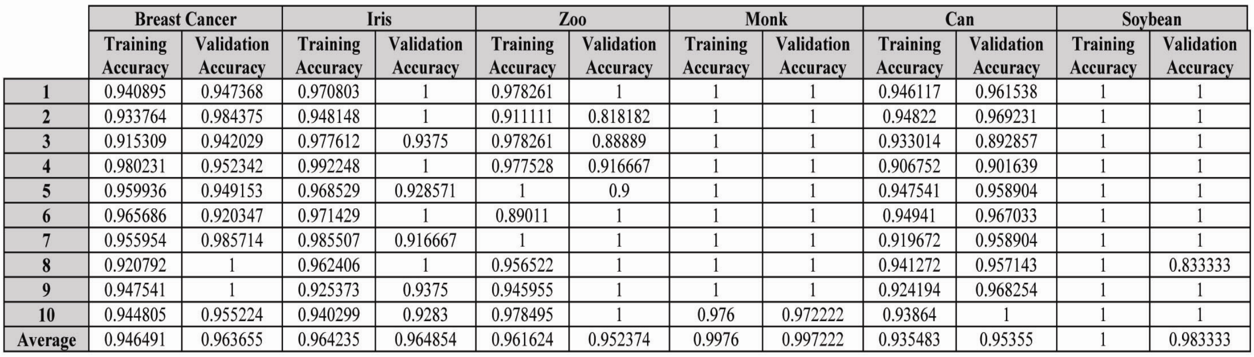
Yang, X. S. (2005). Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach* (pp. 317-323).

Yeh, W.-C., Chang, W.-W., & Chung, Y. Y. (2009). A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method. *Expert Systems with Applications, 36*(4), 8204-8211.

Table 1. Comparison result for UCI datasets for ABC (Original) and ABC (Exchanged)

	ABC (Original)				**ABC (Exchanged)**			
	AVG	**STD**	**MAX**	**MIN**	**AVG**	**STD**	**MAX**	**MIN**
Breast Cancer	0.9611694	0.00563988	0.967892	0.937451	**0.9636985**	**0.00427249**	**0.975482**	**0.955234**
Iris	0.944367	0.00900405	**0.970706**	0.905949	**0.94866478**	**0.00687913**	0.968432	**0.924398**
Zoo	0.91370382	0.0394821	0.953842	0.874361	**0.925383**	**0.02422355**	**0.955035**	**0.876995**
Can	0.96352811	0.00712822	0.972072	0.946644	**0.9642988**	**0.00506**	**0.972183**	**0.957563**
Monk	0.99847567	0.004573	**1**	0.986281	**0.999972**	**8.85E-05**	**1**	**0.99972**
Soybean	0.95560489	**0.01903092**	0.98	**0.925**	**0.9559464**	0.03565	**0.991351**	0.888095

Table 2. Comparison result for UCI datasets for ABC (Original) and ABC (Exchanged)



	Breast Cancer		Iris		Zoo		Monk		Can		Soybean	
	Training Accuracy	Validation Accuracy	Training Accuracy	Validation Accuracy	Training Accuracy	Validation Accuracy	Training Accuracy	Validation Accuracy	Training Accuracy	Validation Accuracy	Training Accuracy	Validation Accuracy
1	0.940895	0.947368	0.970803	1	0.978261	1	1	1	0.946117	0.961538	1	1
2	0.933764	0.984375	0.948148	1	0.911111	0.818182	1	1	0.94822	0.969231	1	1
3	0.915309	0.942029	0.977612	0.9375	0.978261	0.88889	1	1	0.933014	0.892857	1	1
4	0.980231	0.952342	0.992248	1	0.977528	0.916667	1	1	0.906752	0.901639	1	1
5	0.959936	0.949153	0.968529	0.928571	1	0.9	1	1	0.947541	0.958904	1	1
6	0.965686	0.920347	0.971429	1	0.89011	1	1	1	0.94941	0.967033	1	1
7	0.955954	0.985714	0.985507	0.916667	1	1	1	1	0.919672	0.958904	1	1
8	0.920792	1	0.962406	1	0.956522	1	1	1	0.941272	0.957143	1	0.833333
9	0.947541	1	0.925373	0.9375	0.945955	1	1	1	0.924194	0.968254	1	1
10	0.944805	0.955224	0.940299	0.9283	0.978495	1	0.976	0.972222	0.93864	1	1	1
Average	0.946491	0.963655	0.964235	0.964854	0.961624	0.952374	0.9976	0.997222	0.935483	0.95355	1	0.983333

Table 3. The statistic result of UCI dataset for ABC algorithm

	AVG	**MAX**	**MIN**	**STD**
Breast Cancer	0.963699	0.975482	0.955234	0.004272
Iris	0.948665	0.968432	0.924398	0.006879
Zoo	0.925383	0.955035	0.876995	0.024224
Can	0.964299	0.972183	0.957563	0.00506
Monk	0.999972	1	0.99972	8.85E-05
Soybean	0.955946	0.991351	0.888095	0.03565

Table 4. Six UCI Datasets Used in the Experiments

Data Set	**Features**	**Instances**	**Class**
Breast Cancer	9	683	2
Iris	4	150	3
Zoo	17	101	7
Monk	6	432	2
Can	4	683	2
Soybean	35	47	4

Table 5. Computer Configuration

	Configuration
CPU	Intel® Core™2 P7350 @ 2.00GHz
Memory	4.00GB DDR2 SDRAM
Hard-Disk	320GB (5400RPM)
operating system	Windows Vista™ Home Premium

Table 6. the result of comparison

	ABC (Original)	**ABC (Exchanged)**	**PART**	**SOM**	**Naive Bayes**	**Classification Tree**	**KNN**
Breast Cancer	0.967892	**0.975482**	0.9487	0.9707	0.9722	0.9457	0.9721
Iris	**0.97071**	0.968432	0.953	0.9597	0.9329	0.9531	0.9597
Zoo	0.953842	**0.965035**	0.93	0.92	0.91	0.91	0.37
Monk	**1**	**1**	**1**	0.8051	0.9722	0.9722	0.4987
Can	0.972072	0.972183	0.9575	0.9721	**0.9737**	0.9531	0.9663
Soybean	0.98	**0.991351**	0.9583	0.9792	0.955	0.9534	0.98

Feature 1		Feature 2		……	Feature N	
Lower Bound	Upper Bound	Lower Bound	Upper Bound		Lower Bound	Upper Bound

Figure 1. the format of classification rule

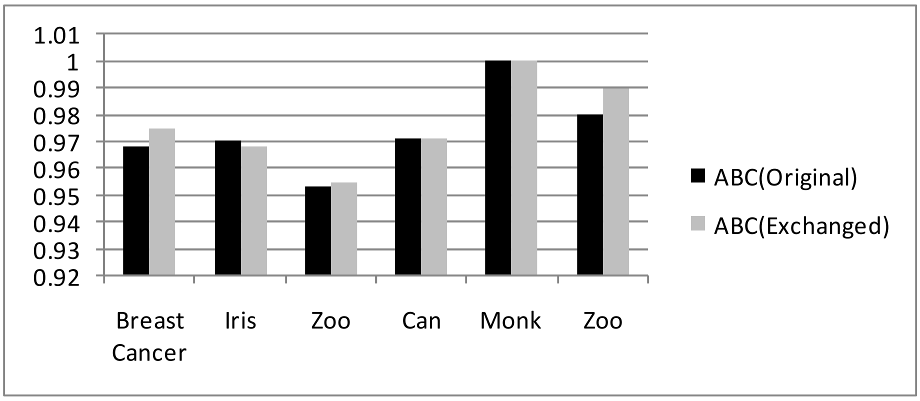


Figure 2. Comparison of two different local search strategies

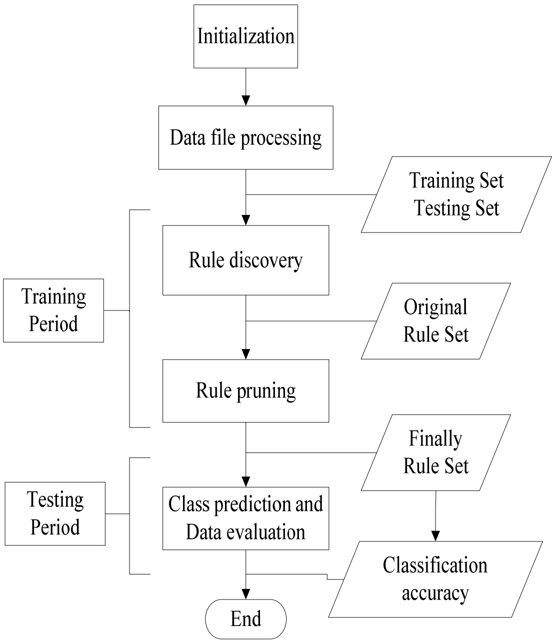


Figure 3. the flowchart of classification data mining

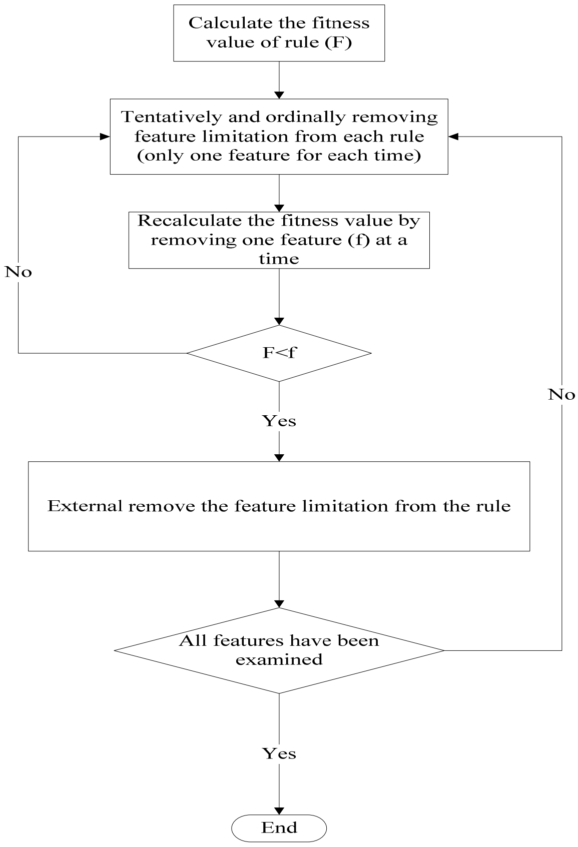

Calculate the fitness value of rule (F)
Tentatively and ordinally removing feature limitation from each rule (only one feature for each time)
Recalculate the fitness value by removing one feature (f) at a time
No
F<f
Yes
No
External remove the feature limitation from the rule
All features have been examined
Yes
End

Figure 4. the flowchart of rule discovery procedure

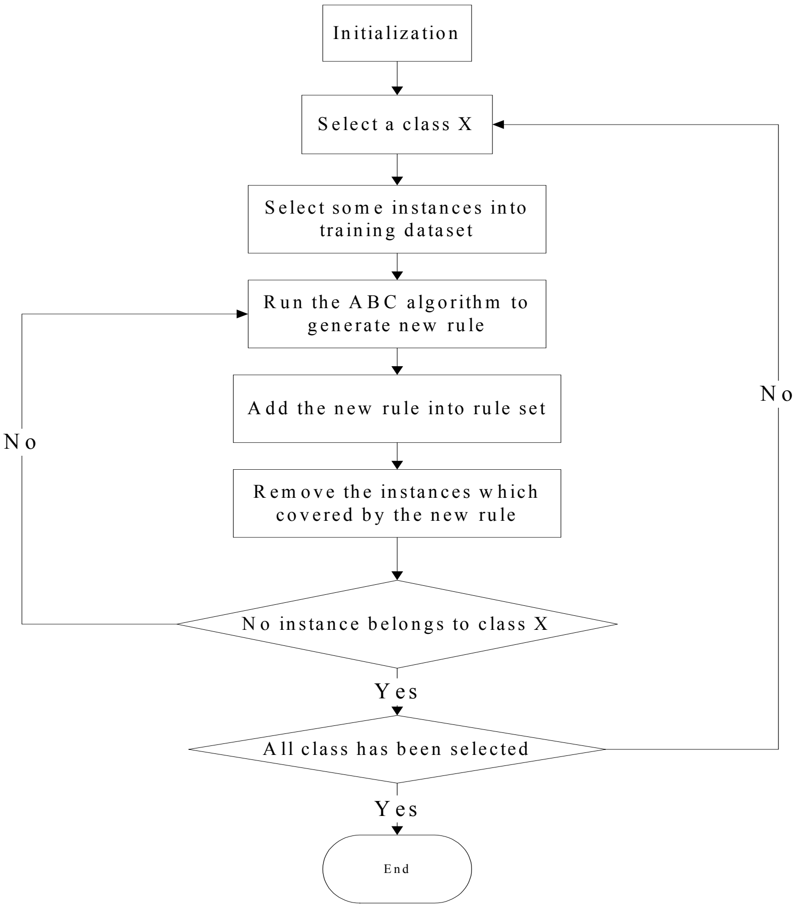

Initialization
Select a class X
Select some instances into training dataset
Run the ABC algorithm to generate new rule
Add the new rule into rule set
Remove the instances which covered by the new rule
No instance belongs to class X
No
No
Yes
All class has been selected
Yes
End

Figure 5 the flowchart of rule discovery for ABC data mining

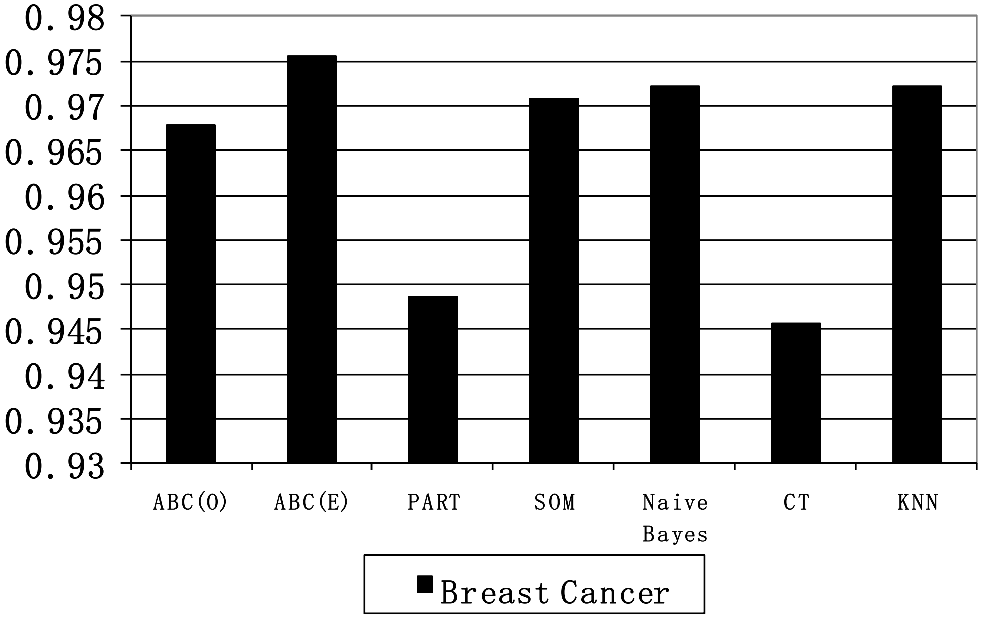


Figure 6. the accuracy of Breast Cancer dataset

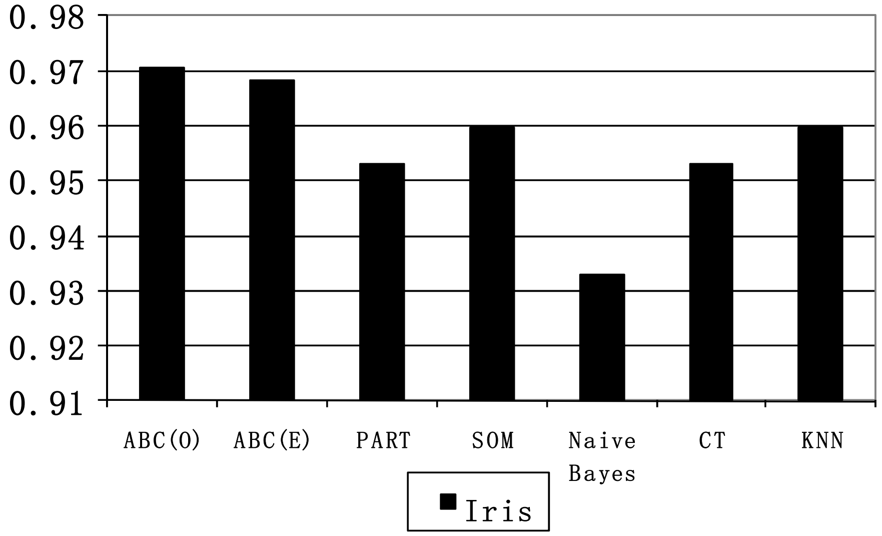


Figure 7. the accuracy of Iris dataset

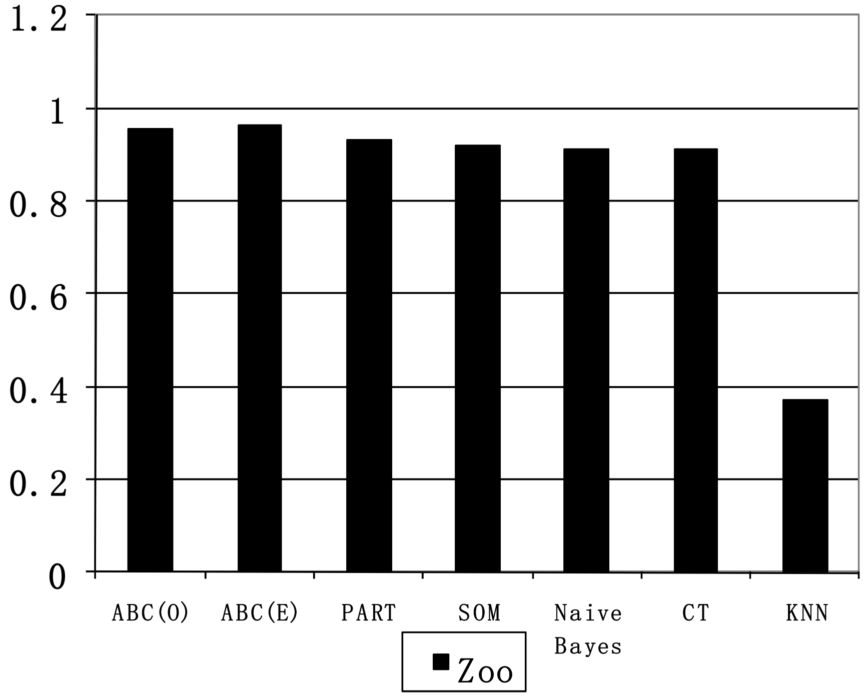


Figure 8. the accuracy of Zoo dataset

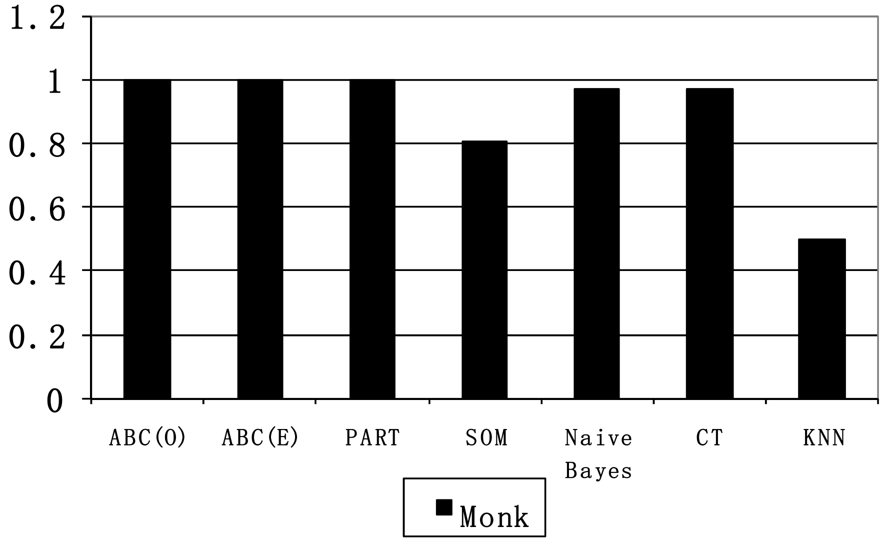


Figure 9. the accuracy of Monk dataset

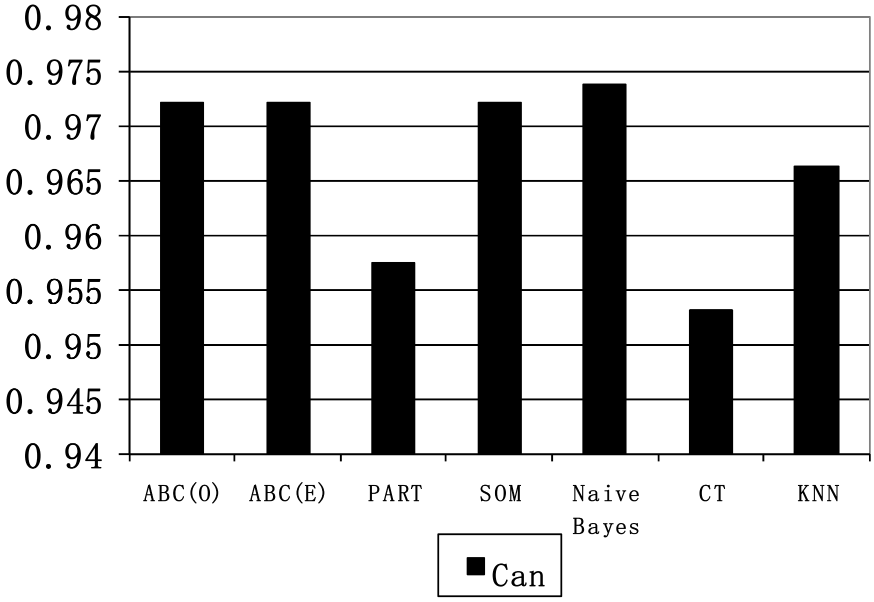


Figure 10. the accuracy of Can dataset

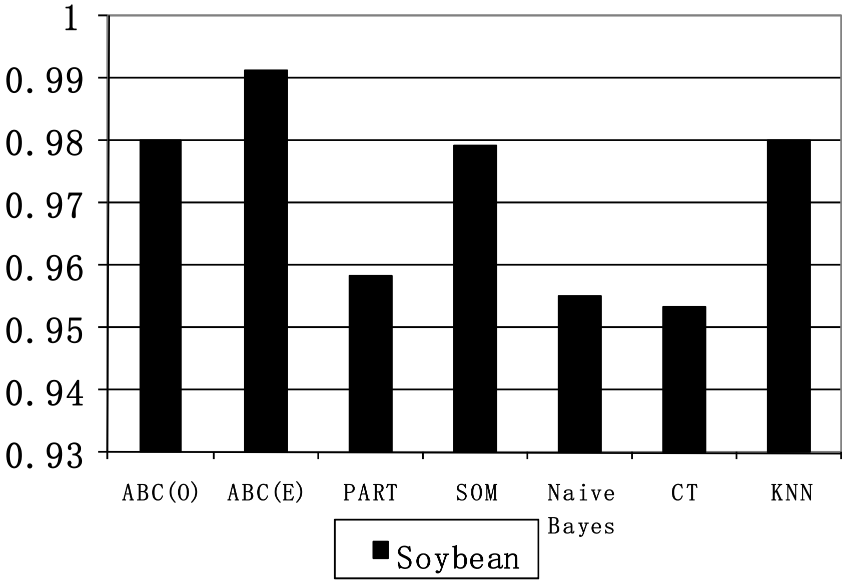


Figure 11. the accuracy of Soybean dataset