Artificial Bee Colony Programming for Feature Selected Cancer Data Classification

Sibel Arslan (Corresponding author)
Department of Computer Engineering
Erciyes University, Engineering Faculty Computer Engineering, Kayseri, Turkey
E-mail: sibel.arslan2@icisleri.gov.tr

Celal Ozturk
Department of Computer Engineering
Erciyes University, Engineering Faculty Computer Engineering, Kayseri, Turkey
E-mail: celal@erciyes.edu.tr

Abstract

Feature selection provides model extraction by using the necessary / related features and improves classification. Feature selection is a desired process at eliminating irrelevant and redundant features in data. Classification is used to distribute the data in a balanced way among the various classes defined in the data as a result of the feature selection. In this paper, we investigated the feature selected classification performance in cancer data of recently proposed Artificial Bee Colony Programming and widely used Genetic Programming. The experimental results are compared with our previous work show that Artificial Bee Colony Programming have better performance Genetic Programming.

Keywords: Feature Selection; Classification; Genetic Programming; Artificial Bee Colony Programming.

1. Introduction

Recently, learning from data and selecting relevant features has become increasingly popular in machine learning researches. Feature selection is a process of removing noisy and irrelevant features that are unnecessary to describe the output variables. As an important result of feature selection is high quality classification and successful modeling. Automatic programming is a computer programming technique based on features at a higher level that can be easily determined according to normal programming languages, and which automatically generates a program code [1].

One of the most widely used automatic programming methods is Genetic Programming. There are several works which related to solve Genetic Programming (GP) in feature selected classification problem [2-4]. Artificial Bee Colony Programming (ABCP) is a high level automatic programming technique improved on Artificial Bee Colony (ABC) algorithm. Since Artificial Bee Colony Programming (ABCP) is a recently proposed method, there are no works related to this field. In this paper, Artificial Bee Colony Programming was used in the problem of classifying the cancer data and its performance was compared with the results of GP results [5].

A feature is an individually measured characteristic of processes. The dependent output variables of the processes depend on a number of independent features including irrelevant and unnecessary features when collecting data makes it difficult to model processes. Various techniques have been developed to overcome the problem of reducing noisy or irrelevant features in data. Complexity of the model and computation time are reduced removing irrelevant features [6].

Feature selection provides a functional relation between output variables and input features by reducing noise. There are three categories of feature selection methods: filter methods, wrapper methods, and embedded methods [7-8]. Filter methods are fast methods that do not use the learning algorithm in the selection process. Wrapper methods use a black box to select features and achieve more successful results than other methods. Embedded methods include feature selection as part of the training process. In this paper, we evaluated the success of classification by selecting the features of GP and ABCP automatic programming methods using cancer data set.

In last decade, increasing interest in discovering potentially useful information has led to feature selection researches [3, 9]. In [3], it is proven that automatic programming methods can be used as a feature selector and cancer classifier. Selecting the discriminative genes of GP and expressing the relationships between the genes as mathematical equations were convincing proof that GP can be used in this field. Landry et al. compared k-NN with decision trees generated by GP over data sets with different class numbers [9]. GP was more reliable performance for feature selection and classification problems that the test sets and training sets have similarity with the feature. This paper is the first study of ABCP that applied to select the relevant features in cancer data.

Much work has been devoted to classification using Genetic Programming (GP) and Artificial Bee Colony (ABC) algorithm [10-13]. In [10], a new GP based feature selection method is proposed for feature selection and classification where the results are compared with other GP versions. Liu et al. designed a new GP based ensemble system to classify different cancer types [11]. ABC was applied to data clustering on benchmark problems and was compared with conventional classification techniques in [12]. Karaboga et al. applied ABC on training feed forward neural networks and classified different datasets [13].

The organization of the paper is as follows: ABCP is described in Section 2 and Experiment Design and Results is presented Section 3. The paper is concluded in Section 4 with remarking the future work.

2. Artificial Bee Colony Programming

ABCP is a recently proposed automatic programming method based on the Artificial Bee Colony (ABC) algorithm [14]. In ABC, the positions of food sources represent solutions. Each solution is represented by fixed-size arrays. The positions of food sources in ABCP (solutions), are expressed in tree structure and consist of different combinations of terminals and functions, which are defined specifically for problems. Fig. 1 shows a parse tree that can represent solutions in ABCP. The mathematical relationship of the solution in Fig. 1 is expressed in Eq. (1). In these notations *'x_1, x_2, x_3, x_4'* are used to represent the independent variable, and *'$f(x)$'* is used to represent the dependent variable.

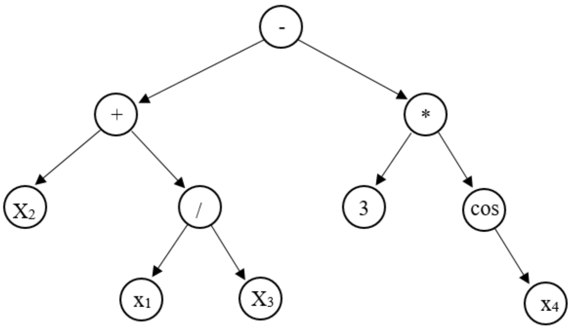


Figure 1. Representation of solutions in ABCP

$$f(x) = x_2 + \left(\frac{x_1}{x_3}\right) - 3 * \cos(x_4) \qquad (1)$$

In ABCP, each employed bee is responsible for only one food source. Evaluating the production and quality of food sources is similar to GP. A food source is represented by a parse tree. Each component in the tree is called a node. Programmer terminals as predetermined constants or variables (such as x_1, y, 8) and functions (+, - /, sin, cos arithmetic operators) create the nodes.

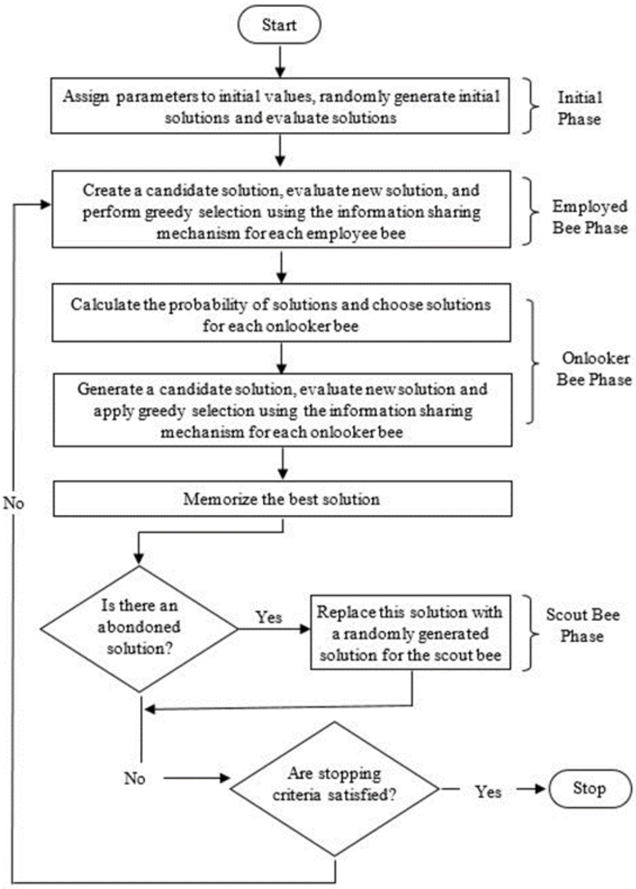


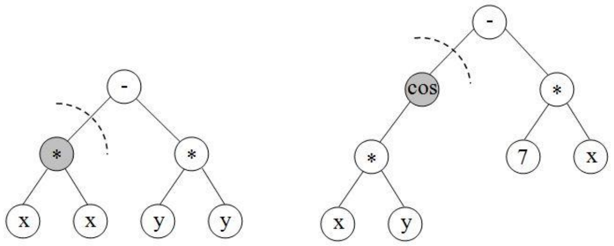
Figure 2. The flow chart of ABCP

The ABCP flow diagram is given in Fig. 2 [15]. At the initial phase; all the bees are scout bees and randomly assigned food sources. Bees assigned food supplies continue their lives as employed bees. Employed bees share their knowledge with the onlooker bees after completing the research process. Source selection is based on the probability values that are calculated based on the nectar quantities (quality), of the sources which the information obtained by the employed bees.

As the quality of a source increases, the probability of selecting that source increases. After the sources are selected, the onlooker bees begin to look for new sources by acting like employed bees. The quality of newly found source is checked by evaluating it with the objective function. If new source is more qualified, the new source is taken into memory and old source is deleted from the memory. Thus, like the employed bees, the onlooker bees have also made a greedy selection.

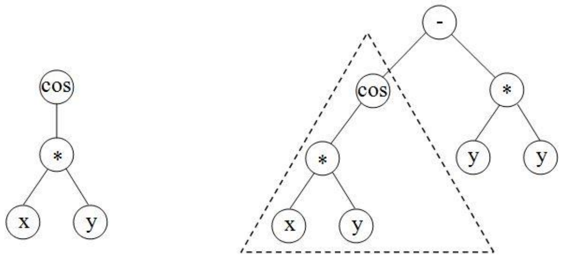
Once the employed bees and the onlooker bees have completed the search process in each cycle, the penalty points of the respective resources are increased by one if they can't find better sources. In a stage where a better source of any source is found, the penalty point of this source is reset. If the penalty score exceeds the value expressed by the *"limit"* parameter, the employed bee abandoned the source behaves as a scout bee and generates a new source randomly, rather than an abandoned source.

The mathematical model of the sources is found by analyzing the whole tree according to a predetermined objective function [16]. Information sharing mechanism, which is the most fundamental difference between ABC and ABCP algorithms, is being used to improve the quality of sources as shown in Fig. 3. When a candidate solution (v_i) is generated, the neighbor node solution x_k, taken from the tree, is randomly selected considering the predetermined probability p_{ip}. The node selected from the neighbor solution x_k determines what information will be shared with the current solution and how much it will be shared. Then node x_i, which presents the current solution in the tree that determines how to use the neighboring node, is randomly selected in the probability distribution p_{ip}. The candidate solution v_i is produced by replacing the nodes of the current solution node x_i and the neighbor solution node x_k. Fig. 3a and 3b are: node x_i representing the current solution and neighbor node x_k taken from the tree, respectively, neighboring information and the generated candidate solution are given in Fig. 3c and Fig. 3d. After the candidate solution is generated, a greedy selection process is applied between the node x_i expressing the current solution and the candidate solution v_i. Candidate solution is evaluated and greedy selection is used for each employed bee.



(a)
$x^2 - y^2$

(b)
$\cos(xy) - 7x$



(c)
$\cos(xy)$

(d)
$\cos(xy) - y^2$

Figure 3. Example of information sharing mechanism in ABCP

3. Experimental Design & Results

In this paper, the experiments are conducted on Wisconsin breast cancer data set [17]. The dataset consists of 30 input parameters that determine whether the breast cancer tumor of 569 patients is good or malignant. When the data set is examined, it is observed that approximately 60% of them are benign and remainder of the tumors were malignant. The paper aimed to identify the tumor as benign or malignant using 30 input parameters presented in the dataset. In the data set, malignant tumor 1 is defined as benign tumor 0. Input parameters include parameters belonging to 10 units. These input parameters are given as radius, texture, circumference, area, fluency, density, concavity, concavity points, symmetry and fractal. There is an average, standard error, and worst error value for each record. Thus, there are totally 30 input parameters. The number of features, training instances and test instances of WDBC is shown in Table 1. Data is almost split with 70% of instances randomly selected from the datasets for training and the other 30% instances forms test set.

Table 1. Characteristics of WDBC

Metrics	**Value**
Features	30
Total Instances	569
Training Instances	427
Test Instances	142
Output Classes	2

Defining Fitness Function

The output variables estimated from the data input parameters in the range [-∞, ∞] take real values. In order to be able to identify the discrete class values (class 0, class 1, etc.) of the output variables, it is first drawn with Eq. (2) to a range defined previously and containing the total number of classes.

$$(N_c - 1)\left(\frac{1}{(1+\exp(-g_o))}\right) \tag{2}$$

In the equation, N_c is the number of output classes, and g_o is the result of the current solution. For example, for a problem of class 5, the output of Eq. (2) is in the range [0-4]. By rounding the found real numbers to the nearest integer value, the solution class value is drawn to the discrete class range as '0', '1', '2', '3', '4'. In our work, a two-class problem is mentioned which is similar to benign class 0, malignant class 1. The result obtained from Equation 2 is in the range [0,1] and the outputs are rounded to the nearest integer value and the class value is estimated as '0' or '1'.

A fitness function is required to correctly classify the classes predicted from Equation 2. In this work, the fitness function is found to be the weighted sum of the ratios of the total classes in the data set of correctly estimated classes. This function, which is defined as Sensivity Fitness Function (SFF), is the weighted sum of the portion of class ‘0’ cases which it correctly predicts as being class ‘0’ and the portion of class ‘1’ cases which it correctly predicts as being class ‘1’. SFF is given in Eq. (3) [18].

$$SFF = w\frac{n_c(i,0)}{n_a(i,0)} + (1-w)\frac{n_c(i,1)}{n_a(i,1)} \tag{3}$$

$n_c(i,k)$ is the number of correctly estimated states when compared to the k class in data set from the class k for the i. th solution, $n_a(i,k)$ the number of all records in class k in the data set. The weight value (w) is generally used equally to ensure that the ratio distribution for each class is balanced. In some cases, different values can be set to avoid weight classification misclassification in order to provide a balance between classes in the unbalanced data sets. In order to evaluate the models obtained from the solutions, the penalty value can be added to the fitness function defined in Eq. (3) as expressed in Eq. (4). p is defined penalty point and N_o is the total number of nodes in the individual.

$$SFF_n = \left(\sum_{J=0}^{n-1} w_j * \frac{n_c(i,j)}{n_a(i,j)}\right) - pN \tag{4}$$

The complexity of solutions that are proportional to the depth of the tree and the number of nodes is computed as in Eq. (5). C is tree complexity, d is the depth of the solution tree, and n is the number of nodes at that depth.

$$C = \sum_{k=1}^{d} n * k \tag{5}$$

Parameters

The control parameters used by the GP and ABCP are given in Table 2 and they are chosen same as [5]. The weight value is defined in proportion to the number of classes in the output of each data set. Each class is equal importance.

Table 2. Parameters

Parameters	**Values**
Colony Size	50
Generation	100
Maximum Tree Depth	12
Functions	*+,-, *, tan, sin, cos, square, maxx minx, exp, if then else*
w	0.5

The *maxx* function specifies the maximum value of vector, the *minx* function specifies the minimum value of vector, and the *ifbte* and the *iflte* indicates the condition of the nodes. Eq. (6) and Eq. (7) describe how the functions operate condition expressions.

$$X = ifbte(A, B, C, D) \\ if(A \geq B) then\ X = C\ else\ X = D \tag{6}$$

$$X = iflte(A, B, C, D) \\ if(A < B) then\ X = C\ else\ X = D \tag{7}$$

Simulation Results

ABCP run 30 times according to configuration in Table 2. The classification success of the methods are given in Table 3 in terms of mean and best values.

As shown in Table 3, ABCP is found to be more successful than the GP. As a result of running, the best SFF_{train} was found 0,94 in ABCP algorithm. The function of the best classifier in train at ABCP is determined as in Eq. (8).

$$2\cos(x_{27} + if(x_7 \geq x_3\ then\ x_{11}\ else\ (x_1 * x_{25} - \cos(x_{11})))) - 2x_{11} \tag{8}$$

As expressed in Equation 8, the best classifier in ABCP consists of six different features. These parameters are defined in the data set as x_1 radius (average), x_3 circumference (average), x_7 concavity (average), x_{11} radius (standard error), x_{25} fluency (worst error), x_{27} concavity (worst error). The tree-like representation of Eq. (8) is given in Fig. 4.

Table 3. Classification Results

WDBC	**GP [5]**				**ABCP**			
	SFF$_{train}$	**Train Success Percentage**	**SFF$_{test}$**	**Test Success Percentage**	**SFF$_{train}$**	**Train Success Percentage**	**SFF$_{test}$**	**Test Success Percentage**
Mean	0,87	87,94	0,86	87,54	0,91	92,85	0,90	91,60
Standard Deviation	0,03	3,04	0,02	2,70	0,02	2,34	0,04	3,62
Best	0,91	92,04	0,90	91,55	0,94	96,02	0,96	97,18

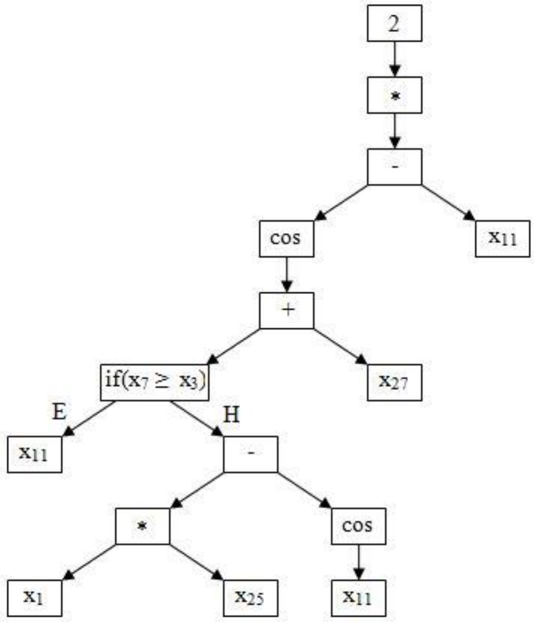


Figure 4. Best classifier tree at ABCP

The characteristics of the best classifier tree in ABCP are given in Table 4.

Table 4. Best Classifier Tree Information In ABCP

Metrics	**Values**
Total number of nodes	15
Depth of the best solution tree	9
Best solution tree complexity	88

The maximum input features evolved by mathematical models at the end of each run are presented in Fig. 5 in ABCP.

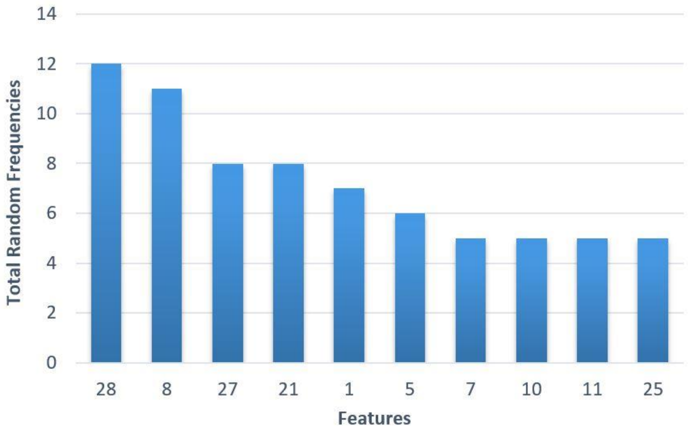


Figure 5. Maximum Features Evolved Models in ABCP

As shown in Fig. 5, the most common feature is x_{28}. Other frequently features are x_8, x_{27}, x_{21}, x_1, x_5. In our previous work, the most common features were mentioned in the models extracted by the GP [5]. Comparison of the top ten important features by the two methods on WDBC are shown in Table 5 in a descending order of importance.

Table 5. Features Selected By the Methods

Metrics	**Mean**	**Top 10 Important Features**	**Features in Both GP and ABCP**
ABCP	3,53	x_{28}, x_8, x_{27}, x_{21}, x_1, x_5, x_7, x_{10}, x_{11}, x_{25}	x_{28}, x_8, x_{27}, x_1, x_7, x_{11}, x_{25}
GP [5]	3,3	x_{25}, x_1, x_7, x_8, x_{28}, x_9, x_{22}, x_{27}, x_{11}, x_4	

As a result of 30 runs, the average number of features in the evolved models for both methods are close to each other in GP and ABCP. Seven features are also found to be common in the methods. 5 of the 6 features (x_1, x_7, x_{11}, x_{25}, x_{27}) in the best classifier of the ABCP (Eq. (8)) are among the top ten important features in both GP and ABCP.

CONCLUSIONS

In this paper, breast cancer is classified in the performance analysis of the ABCP. When the results are evaluated, ABCP compared with GP is effective in finding important features which have been obtained by evaluating the evolved models along with the presence of redundant features. Future work will compare ABCP with other traditional methods and evaluate their performance.

REFERENCES

[1] Biermann, A.W., Automatic Programming: A Tutorial on Formal Methodologies, J. Symbolic Computation, pp. 119-142, 1985.

[2] Nag, K., Pal, N. R., A Multiobjective Genetic Programming Based Ensemble For Simultaneous Feature Selection And Classification, IEEE Trans. Cybern., Vol. 46, No. 2, pp. 499–510, 2016.

[3] Yu, J., Yu, J., Almal, A.A., Dhanasekaran, S.M., Ghosh, D., Worzel, W.P., Chinnaiyan, A.M, Feature Selection and Molecular Classification Of Cancer Using Genetic Programming, Neoplasia, Vol. 9, No: 4, pp. 292 – 303, 2007.

[4] Zhang, Y., Rockett, P.I., Domain Independent Approaches To Optimise Feature Extraction for Multiclassification Using Multi-Objective Genetic Programming, Technical Report, Department of Electronic and Electrical Engineering, University of Sheffield, 2007.

[5] Arslan, S., Ozturk, C., Feature Selected Cancer Data Classification with Genetic Programming, 12. National Meeting of Biomedical Engineering BİYOMUT 2017, 2017.

[6] Cai, R., Hao, Z., Yang, X., Wen, W., An Efficient Gene Selection Algorithm Based On Mutual Information, Neurocomputing, Vol 72, pp. 991-999, 2009.

[7] Xue, B., Zhang, M., Browne, W. N., Yao, X., A Survey On Evolutionary Computa-Tion Approaches To Feature Selection, IEEE Trans. Evol. Comput., Vol. 20, No. 4, pp. 606–626, Aug. 2016.

[8] Guyon, I, Elisseeff, A., An Introduction To Variable And Feature Selection, J. Mach. Learn. Res., Vol. 3, pp. 1157–1182, Mar. 2003.

[9] Landry, J.A., Costa, L.D., Bernier, T., Discriminant Feature Selection By Genetic Programming: Towards A Domain Independent Multiclass Object Detection System, Systemics Cybernetics and Informatics, 3(1), pp. 7681, 2006.

[10] Awuley, A., Ross, B.J., Feature Selection And Classification Using Age Layered Population Structure Genetic Programming. CEC 2016, Vancouver BC, 2016.

[11] Liu, K.H., Tong, M., Xie, ST., Yee, V.T., Genetic Programming Based Ensemble System for Microarray Data Classification, Hindawi Publishing Corporation Computational and Mathematical Methods in Medicine, Vol: 2015, Article ID 193406, 2015.

[12] Karaboga, D., Ozturk, C., A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, Vol: 11, pp. 652-657. (2011)

[13] Karaboga, D., Ozturk, C., Neural Networks Training by Artificial Bee Colony Algorithm on Pattern Classification. Neural Network World, International Journal on Neural and Mass Parallel Computing and Information Systems, Prague, Vol. 19, Iss. 3, pp. 279-292, 2009.

[14] Karaboga, D., Ozturk, C., Karaboga, N., Gorkemli, B., Artificial Bee Colony Programming For Symbolic Regression, Information Sciences 209, Pages: 1–15, 2012.

[15] Gorkemli, B., Work of Artificial Bee Colony Programming (ABCP) to Symbolic Regression Problems, PhD Thesis, Erciyes University, Engineering Faculty, Computer Engineering Department, 2015.

[16] Gan, Z., Chow, T.W.S., Chau, W.N. Clone selection programming and its application to symbolic regression, Expert Systems with Applications 36, 3996–4005, 2009.

[17] UCI Machine Learning Repository Breast Cancer Wisconsin Data Set. [Online] Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic), (March 10, 2018)

[18] Morrison, G. A., Searson, D.P., Willis, M.J., Using Genetic Programming to Evolve a Team of Data Classifiers, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 4, No:12.