Applied Soft Computing Journal 74 (2019) 634–642

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Feature selection based on artificial bee colony and gradient boosting decision tree

Haidi Rao [a,b], Xianzhang Shi [a,b], Ahoussou Kouassi Rodrigue [a,b], Juanjuan Feng [a], Yingchun Xia [a], Mohamed Elhoseny [d], Xiaohui Yuan [c,*], Lichuan Gu [a,b,**]

[a] *College of Computer and Information, Anhui Agricultural University, Hefei, 230036, China*
[b] *Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture, Hefei, 230036, China*
[c] *Department of Computer Science and Engineering, University of North Texas, TX, 76203, USA*
[d] *Mansoura University, Mansoura, 35516, Egypt*

HIGHLIGHTS

- A novel method for feature selection based on bee colony and decision tree.
- The proposed method improves efficiency and informative quality of the selected features.
- Experiments conducted with breast cancer datasets demonstrate superior performance.

ARTICLE INFO

Article history:
Received 20 July 2018
Received in revised form 10 October 2018
Accepted 16 October 2018
Available online 5 November 2018

Keywords:
Bee colony algorithm
Decision tree
Feature selection
Dimensionality reduction

ABSTRACT

Data from many real-world applications can be high dimensional and features of such data are usually highly redundant. Identifying informative features has become an important step for data mining to not only circumvent the curse of dimensionality but to reduce the amount of data for processing. In this paper, we propose a novel feature selection method based on bee colony and gradient boosting decision tree aiming at addressing problems such as efficiency and informative quality of the selected features. Our method achieves global optimization of the inputs of the decision tree using the bee colony algorithm to identify the informative features. The method initializes the feature space spanned by the dataset. Less relevant features are suppressed according to the information they contribute to the decision making using an artificial bee colony algorithm. Experiments are conducted with two breast cancer datasets and six datasets from the public data repository. Experimental results demonstrate that the proposed method effectively reduces the dimensions of the dataset and achieves superior classification accuracy using the selected features.

1. Introduction

Data from real-world applications can be of high dimensional. This is particularly true for the applications in the fields of medicine [1–4] and remote sensing [5]. For instance, mass spectrometry is a promising diagnostic and cancer biomarker discovery tool for cancers such as prostate, ovarian, breast, and bladder cancers [6]. The superior sensitivities and specificities in contrast to the classical cancer biomarkers attract great attention of medical professionals. However, thousands of features of mass spectrometric data make feature selection a necessary step for effective processing and analysis in computer-aided diagnosis. Features in the mass spectrometric data are usually highly redundant, which is the cause of the well-known curse of dimensionality problem in machine learning [7]. Identifying informative features has become an important step for data mining not only to circumvent the curse of dimension but to reduce the amount of data for processing. In general, feature selection reduces the number of features while keeping the same or even better learning performance [8]. Its advantages have been demonstrated in various data mining and machine learning applications [9,10]. When redundant, irrelevant, noisy features are removed from the training dataset, the efficiency of the learning process is usually improved as well.

Feature selection is responsible for selecting a subset of features, which can be described as a search process in a state space. There have been many methods developed for feature selection [11]. Alickovic et al. [3] proposed a decision-making system of breast cancer diagnosis. In this method, genetic algorithms are used to remove insignificant features and multiple classifiers are

[*] Corresponding author.
[**] Corresponding author at: College of Computer and Information, Anhui Agricultural University, Hefei, 230036, China.
E-mail address: xiaohui.yuan@unt.edu (X. Yuan).

https://doi.org/10.1016/j.asoc.2018.10.036
1568-4946/© 2018 Elsevier B.V. All rights reserved.

employed to classify breast cancer. Zhu et al. [12] proposed an unsupervised spectral feature selection method to preserve both the local and global structures of the features when removing irrelevant ones. Mafarja et al. [13] proposed three improvements based on Whale Optimization Algorithm (WOA) to optimize features in a dataset. When a simple mutation operator is used, the performance of the algorithm becomes better. Wan et al. [4] evaluated hierarchical feature selection methods for aging-related gene datasets. Wang et al. [14] construct the primary features of user comments about items and select features using Gradient Boosting Decision Tree (GBDT). GBDT [15] was developed to identify the primary features of users' comments about items and could be efficient for feature selection. However, GBDT has a high demand for initial input when building trees. Redundant initial inputs could pose a significant challenge to the efficiency in both time and space. On the other hand, Artificial Bee Colony (ABC) algorithm has demonstrated great efficiency to convergence, which compliments the disadvantage of GBDT.

To address the open issues in feature selection of high dimensional data, we propose a method that is based on the coherent integration of artificial bee colony and gradient boosting decision tree algorithm (ABCoDT). To improve the feature selection using GBDT, initial inputs with a high dimensionality is optimized. We employ the accuracy of classification by GBDT to evaluate the quality of the inputs, which minimize the potential of ABC being trapped in a local optimum. Hence, the proposed algorithm achieves a global optimization and identifies the most informative features. The general idea is that the method initializes the feature space spanned by the dataset. Less relevant features are suppressed according to the information each feature contributes to the decision making using an artificial bee colony algorithm. Our method reduces the initial input of a gradient boosted decision tree algorithm and removes the features with a low correlation.

The main contributions of this paper are two-fold: (1) Feature selection is closely coupled with the decision process by integrating the gradient boosting decision tree in the loop, and hence an end-to-end solution is devised. (2) The combinatorial optimization problem is addressed via a swarm intelligence method to reach the global optimum. The idea extends the deep learning theories by integrating the identification of the informative features of the learning process in a coherent manner. This provides a framework to allow the employment of various classification methods for a deep learning.

The rest of this article is organized as follows. Section 2 reviews the related work of feature selection. Section 3 presents our proposed method in detail and gives a brief review of the artificial bee colony algorithm. Section 4 discusses the experimental results from six public datasets. Section 5 concludes this paper with a summary of our proposed method and findings.

2. Related work

Dash et al. [16] presented a general framework of feature selection that includes four phases: generating a subset of features according to rules, the candidate subset is evaluated, repeating the subset generation if the stop rule is not satisfied, validate the final subset of features. Feature selection methods can be categorized into three types: filter based methods, wrapper based methods, and embedded based methods [17]. Filter based methods analyze the statistical performance of sample data to select features. It is independent of the classification algorithm. There are many methods such as Variance Threshold (VT), SelectKBest (SKB) and information gain. The multivariate methods evaluate the dependencies of features and attempt to minimize the relevance among features [18]. In general, filter based methods are highly efficient in selecting features. However, the accuracy of the following classification process using the selected feature is relatively low.

Wrapper-based methods evaluate attribute sets using a machine learning method via an iterative search process. The results of each iteration are used as a heuristic for this search. Exemplar methods of wrapper based methods include ant colony optimization (ACO) [19], genetic algorithm (GA) [20], random mutation hill-climbing [21], simulated annealing (SA) [22] and ABC [23]. The wrapper based methods leverage the feedback from learning algorithms and usually outperform the filter based methods in term of accuracy. On the other hand, the employment of a learning algorithm in the search process is time-consuming and computationally expensive, especially for high-dimensional, large datasets. They also face the risk of overfitting.

Embedded methods attempt to reduce the computation cost of reclassifying different subsets that are performed in wrapper methods. The methods integrate feature selection into the process of building model and are closely coupled with a specific learning model. The selected features are achieved by optimizing the learning objective function [8]. Decision trees, such as CART, have a built-in mechanism to perform variable selection [8]. Bi et al. [24] use l1-norm SVMs, without iterative multiplicative updates, which takes in the context of least-square regression and eliminates features by setting their weights to zero. The number of variables can be further reduced by backward elimination.

To enhance search performance, Shunmugapriya et al. [25] optimize the abandoned solutions of food sources by giving weight to food sources of employed bees based on ABC. Gu et al. [26] exploited competitive swarm optimizer (CSO) to optimize large-scale data. It selects a much smaller number of features. Uzer et al. [27] optimize feature selection by using ABC and classify a sample dataset by using Support Vector Machine (SVM). The method has high accuracy but just reduces fewer features.

To overcome the aforementioned problems and meet the demand for feature selection from high dimensional data, we develop a novel method that coherently integrates ABC and GBDT algorithms. ABC usually converges to the global optimum efficiently and consumes less time for feature selection. It provides a reasonably good initialization for building trees using GBDT, which is slow when data is of high dimension. In contrast, GBDT achieves a high accuracy and hence a promising method for feature selection.

3. Methodology

3.1. Gradient boosting decision tree

The basic idea of the gradient boosting decision tree is combining a series of weak base classifiers into a strong one. Different from the traditional boosting methods that weight positive and negative samples, GBDT makes global convergence of algorithm by following the direction of the negative gradient [14,28].

Let $\{x_i, y_i\}_{i=1}^{n}$ denotes the dataset. Softmax is the loss function. Gradient descent algorithm is used to ensure the convergence of the GBDT. The basic learner is $h(x)$, where $x_i = (x_{1i}, x_{2i}, \dots, x_{pi})$. p is the number of the predicted variables. y_i is the predicted label. The steps of GBDT [29,30] are presented as follows:

Step 1: The initial constant value of the model β is given:

$$F_0(x) = \arg\min_{\beta} \sum_{i=1}^{N} L(y_i, \beta) \tag{1}$$

Step 2: For the number of iterations $m = 1 : M$ (M is the times of iteration), the gradient direction of residuals are calculated.

$$y_i^* = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)-F_{\mathrm{m}-1(x)}}, \quad i = \{1, 2, \dots, N\} \tag{2}$$

Step 3: The basic classifiers are used to fit sample data and get the initial model. According to the least square approach, parameter a_m of the model is obtained and the model $h(x_i; a_m)$ is fitted.

$$a_m = \arg\min_{\alpha,\beta} \sum_{i=1}^{N} [y_i^* - \beta h(x_i; a)]^2 \tag{3}$$

Step 4: Loss function is minimized. According to Eq. (4), a new step size of the model, namely the current model weight, is calculated.

$$\beta_m = \arg\min_{\alpha,\beta} \sum_{i=1}^{N} L(y_i, F_{m-1}(x) + \beta h(x_i; a)) \tag{4}$$

Step 5: the model is updated as follows:

$$F_m(x) = F_{m-1}(x) + \beta_m h(x_i; a) \tag{5}$$

However, limited to the dimension and size of the sample data, information gain of feature branch points are needed to be calculated multiple times when raw data is input into GBDT to be analyzed. It leads to an increase of the iteration number and slows the speed of convergence and update. In this paper, we propose to optimize initial data which is input into GBDT by using ABC. The proposed method reduces forcibly the initial feature dimensions of sample data and generates a decision tree rapidly to obtain the weight of features while guaranteeing the accuracy and efficiency of GBDT.

3.2. Artificial bee colony optimization

Artificial bee colony optimization method is inspired by the behavior of bees gathering nectar [31]. The global optimum is obtained by neighborhood search optimization of each bee [32,33]. To be self-content, we include the main steps of the ABC optimization method as follows:

Step 1: Initialize food source. A number of feasible solutions (denoted with SN) are randomly generated. According to Eq. (6), the profit value of the food sources is calculated.

$$x_{i,j} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{6}$$

where $x_i(i = 1, 2, \dots, SN)$ is D dimensions vector, D is the number of parameters in the optimization problem. The profit value is defined as the following:

Definition 1 (The Profit Value of Food Sources). Assume D dimensions vector x_i is an arbitrary food source. D dimensional vector $x_{centeroid}$ is the center point of SN food sources. $fit(x_i)$ is the profit value of a food source x_i .

$$fit(x_i) = \sqrt{\sum_{i=1}^{D} (x_i - x_{centroid})^2} \tag{7}$$

Step 2: Employed bees log itself an optimum value and carry out searching in the neighborhood of current food sources. The Eq. of food sources search is given below:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{8}$$

Step 3: According to the greedy strategy, employed bees choose food sources. The probability that an onlooker bee chooses an employed bee is calculated:

$$p_i = \frac{fit(x_i)}{\sum_{n=1}^{SN} fit(x_n)} \tag{9}$$

where SN is the number of feasible solutions.

Step 4: According topp_i, onlookers choose food sources. Employed bees search new food sources by Eq. (8) and calculate the profit values.

Step 5: When a food source has no improvement after a number of iterations (denoted with *limit*), it will be given up and replaced by a new food source, which is generated randomly.

Step 6: Record the best result.

Comparing with other biological heuristic algorithms, ABC has many advantages [23], such as a more simple structure, fewer control parameters and easy to be realized. Because of its strong ability and a wide range in search, it had received wide attention and researched when once proposed. Thus, ABC is chosen to select features preliminary and preprocess raw dataset. Thereby, the initial input of GBDT is optimized by greatly reducing dimensions of features in the raw dataset.

3.3. Structure of solution

Assume D is the dimensions of the dataset to be optimized, i.e., D features. Solutions can be defined as follows:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{iD}), \quad i = \{1, 2, \dots, SN\} \tag{10}$$

where feasible solutions $x_{ij}(j \in \{1, 2, \dots, D\})$ correspond to the features of the raw dataset in the solution space and x_{i1} is the first feature in the dataset [34].

Let D be the length of the solution array, which is the number of dimensions of the dataset to be optimized. Position in the j^{th} dimension of the i^{th} bee is located in the j^{th} column. Array $x_{ij} \in \{0, 1\}$. After initialization, employed bees search and traverse food sources from an initial food source. If an employed bee is chosen by an onlooker bee, then the position of the employed bee in the array is 1, otherwise, the position is 0. For example, a feasible solution expresses a candidate features subset, such as $x_{i1} = (1, 0, 1, 0, 0, 1, 0, 0)$ represents that the 1^{th}, 3^{th} and 6^{th} features in the raw dataset are chosen and others are given up.

Considering that the feature optimization problem is discrete classification and combination problem, this paper chooses the classification and regression tree as a basic classifier. The coefficient GINI of the tree is used to determine whether branching [35]. Let k, $k = \{1, 2, \dots, D\}$, represent the class, where D is a total number of classes in the dataset. The Gini coefficient of a node A is computed as follows:

$$Gini(A) = 1 - \sum_{k=1}^{D} p_k^2 \tag{11}$$

where p_k is the probability that the sample node belongs to the k^{th} class.

When basic classifiers carry out branching every time, all of the features are traversed and the gains of the splitting threshold for each feature are calculated. The maximum gain of all of feature split points is chosen as the first split points. Branching until the calibration value of the sample on each leaf satisfies unique or default termination condition (for example, the number of leaf reaches the upper limit or information gain after the split is negative).

3.4. Proposed algorithm

Fig. 1 illustrates the flowchart of the proposed method. First, the initial dataset is optimized by using ABC to reduce irrelevant features. The optimized dataset is the input of GBDT. Then, the features of the dataset are further reduced by using GBDT.

The descriptive steps are given as follows:

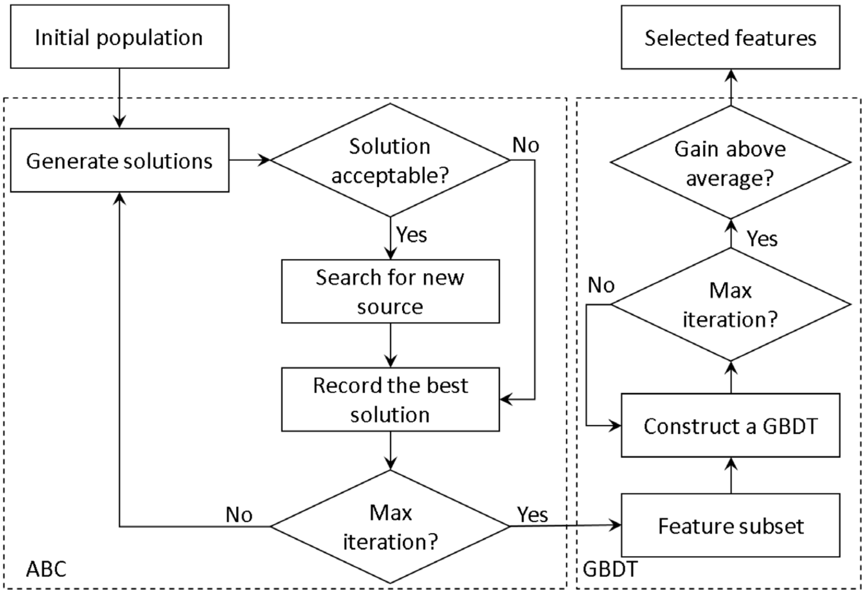


Fig. 1. The flowchart of the proposed ABCoDT method.

Step 1: Initialize the colony. SN feasible solutions are generated randomly in the search space and profit values $fit(x_i)$ are calculated. The greedy strategy is used to select better solutions.

Step 2: Search food sources. According to Eq. (6), employed bees carry out a search in their current field of food sources and determine whether receive new food source based on Eq. (9).

Step 3: Calculate profit values. Scout bees choose employed bees by using roulette. According to Eq. (8), feasible values are searched and profit values are calculated. The greedy strategy is used to select better solutions.

Step 4: Replace old food source. When a food source has no improvement after iterations, it is replaced with a new food source generated randomly using Eq. (6).

Step 5: Obtain feature subset θ. Steps 2 to 5 are repeated. When reaching maximum phylogeny number, the preliminary feature subset θ is obtained.

Step 6: Initial GBDT model. Initial value $F_0(X) = 0$ is given. Classification And Regression Trees (CART) is chosen as a basic classifier and the objective function *softmax* is loss function.

Step 7: Fit the model. For the number of iterations $m = 1 : M$ (M is the number of iteration), the gradient direction of residuals y_i^* are calculated by Eq. (2). The basic classifiers are used to fit the new decision model.

Step 8: Compute new step size β_m . According to Eq. (4), the new step size β_m of the model is obtained.

Step 9: Build a regression tree model. Step 7 to Step 8 are repeated and the complete gradient progressive regression tree model is built in M iterations.

Step 10: Compute maximum feature subset. The average values *ave* of branch points gain are calculated and the branch points, of which the gain value is more than *ave* are selected to construct final solution space. Thereby, the maximum feature subset is obtained.

We take the Glass dataset as an example. The dataset contains eight features, including refractive index (RI), Sodium (Na), Magnesium (Mg), Aluminum (Al), Silicon (Si), Potassium (K), Calcium (Ca), Bavium (Ba), Iron (Fe). After preliminary feature selection by ABC, four irrelevant features are removed and features RI, Al, K, Ba are left. Let the optimized subset be the input of GBDT, the final feature subset is obtained. The diagram is shown in Fig. 2.

ABCoDT is an integration algorithm based on decision trees. The model is trained to get the degree of importance of features used in the model and distinguish those features which have an effect to results of the model. In GBDT, Friedman proposed a global important degree of the feature *j*, which is measured by the average importance degree in a single tree:

$$J_j^2 = \frac{1}{M}\sum_{m=1}^{M} J_j^2(T_m) \tag{12}$$

where M is the number of trees. Assume that each tree is binary. The importance degree of a feature J in a single tree is computed as follows

$$J_j^2(T) = \sum_{t=1}^{L-1} i_j^2 1(v_t = j) \tag{13}$$

where L is the number of leaf nodes, $L-1$ is the number of the non-leaf node, v_t is the feature associated with the node t . i_t^2 is a loss of square after splitting node t .

The importance degree of features is available for measuring the weight of an optimized feature set. Let ε_j be the weight of feature j. It is calculated as the average loss of square error:

$$\varepsilon_j = \frac{1}{M}\sum_{m=1}^{M} i_j^2(T_m) \tag{14}$$

where M is the number of features of network-wide behaviors. The weight of features will be changed with the change of feature combinations. Algorithm 1 shows the pseudo-code of the proposed approach.

4. Experimental results

4.1. Experimental datasets and settings

As shown in Table 1, we select eight UCI datasets [36] as an experimental set, including WDBC (Wisconsin Diagnostic Breast Cancer), Habeman (Habeman's Survival), Wine, Contraceptive, Glass, ULC (Urban Land Cover). The selected UCI datasets have no missing

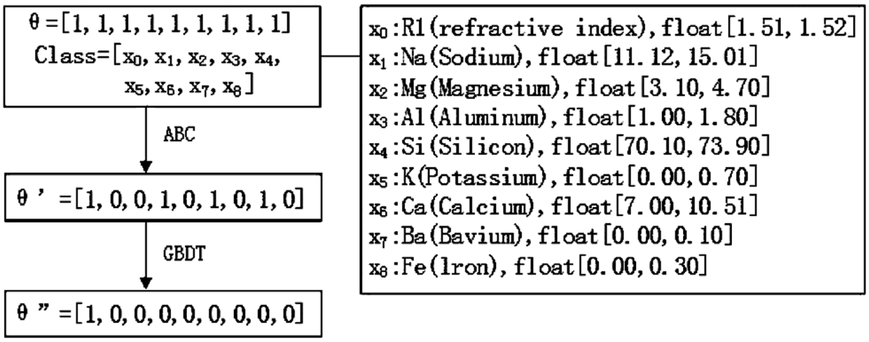

θ =[1, 1, 1, 1, 1, 1, 1, 1, 1]
Class=[x0, x1, x2, x3, x4, x5, x6, x7, x8]
ABC
θ ’ =[1, 0, 0, 1, 0, 1, 0, 1, 0]
GBDT
θ ” =[1, 0, 0, 0, 0, 0, 0, 0, 0]
x0 :R1(refractive index), float[1.51, 1.52]
x1 :Na(Sodium), float[11.12, 15.01]
x2 :Mg(Magnesium), float[3.10, 4.70]
x3 :Al(Aluminum), float[1.00, 1.80]
x4 :Si(Silicon), float[70.10, 73.90]
x5 :K(Potassium), float[0.00, 0.70]
x6 :Ca(Calcium), float[7.00, 10.51]
x7 :Ba(Bavium), float[0.00, 0.10]
x8 :Fe(Iron), float[0.00, 0.30]

Fig. 2. Example of feature selection for the Glass dataset.

Table 1
Properties of the datasets used in our experiments.

Dataset	# of examples	# of features	Classes
WDBC	569	30	2
Haberman	306	3	2
Glass	214	9	7
Contraceptive	1,473	9	3
Wine	178	13	3
ULC	675	148	9
p53	16,772	5,408	2
Arcene	900	10,000	2

Table 3
Results of feature selection.

Dataset	# of the original	# of the selected	Reduction rate (%)
WDBC	30	2	93.3
Haberman	3	1	66.7
Glass	9	1	88.9
Contraceptive	9	2	77.8
Wine	13	1	92.3
ULC	147	5	96.6
p53	5,408	565	89.5
Arcene	10,000	2103	80.0

in property values. When testing the algorithm, the number of classes is not considered.

The proposed method is implemented with Python 3.5 and the experiments are conducted on a computer with Intel Core i5-4590 CPU at 3.30 GHz and 16G of memory using 64 bit Windows 7 operating system. The parameter settings used in our experiments are listed in Table 2. In our evaluation, six iteration numbers are used to gain an understanding of the optimization results using different numbers of repetitions. To deal with multiple classifications, the loss function is chosen as the objective function.

In this paper, sklearn class library is used to carry out the algorithm experiment. The parameters of the VT algorithm are as follows: threshold of variance is 0. The parameters of the SKB method are: calculating correlation coefficient function is chi-square, step length is 1, optimal display. The base model is a logistic regression with L1 and L2 penalty terms, and the threshold value of the weight coefficient is 0.5.

4.2. Feature reduction

Table 3 shows the number of features retained by our proposed ABCoDT method. The reduction ratio of selected features is computed as $(\frac{n-v}{n}) \times 100\%$. The minimum percentage of reduction ratio is 66.7% and the maximum reduction ratio is 96.6%. Although the amount of reduction varies, it is clear that the ratio of feature reduction is more than 60%. When the number of original features is greater than 10 and lower than 150, the reduction of retained features is more than 90%. When the number of original features is oversize, the reduction of retained features decline. It can be seen that our method could remove the majority of the irrelevant information in different datasets, of which the number of original features keeps within a certain range while keeping the same or even better learning performance.

Fig. 3 depicts the weights of the selected features for all the test cases. The comparisons between the weight of original features and selected features of the eight datasets are shown. The weight of original features is marked in purple color and the weight of selected features using ABC is marked in red color. The figures show that the weight of important features are increased and weight of redundant features are reduced to zero after using ABC. For example, in Fig. 3(a), there are three features in the original dataset Haberman and two features in the optimized dataset. In other figures, it can be seen that the number of features in the optimized dataset is less than the original dataset. The features selected by ABC are more informative; yet, most of the information of the datasets is reserved.

4.3. Classification accuracy and comparison study

We combine xgboost [30] with 10-fold cross-validation to evaluate the accuracy of classification with feature selections. Table 4

Table 2
Experimental parameters and settings.

Method	Parameter	Setting
ABC	The number of nectar sources The maximum phylogeny number MEN Limit	Depending on the dataset dimension 1000 [50, 100, 200, 400,800, 1000]
GBDT	Limit of Iteration times Loss function Tree depth The sampling ratio of samples Regularization parameter The sampling ratio of features Zoom factor	50, 100, 200, 500,800, 1000 Softmax 12 0.7 2 1 0.1

Algorithm 1: ABCoDT Algorithm.

```
Input: $\{x_i, y_i\}_{i=1}^{n}$
Output: $\theta$
Initialize:
$x_i, i = \{1, \ldots, SN\}$ following $x_{ij} = x_{min,j} + rand\,(0, 1)\left(x_{max,j} - x_{min,j}\right)$
Compute $fit(x_i)$ following $fit(x_i) = \sqrt{\sum_{j=1}^{D}\left(x_{ij} - x_{centroid}\right)^2}$
FES = SN
Employed Bee:
for i=1 to SN do:
    $v_{ij} \leftarrow x_{ij} + \varphi_{ik}\left(x_{ij} - x_{kj}\right)$
    $fit(v_{ij}) = \sqrt{\sum_{j=1}^{D}\left(x_{ij} - x_{centroid}\right)^2}$,FES=FES+1
if $fit(v_{ij}) < fit(x_i)$
    $x_i \leftarrow v_{ij}, trial_i = 1$
else
    $trial_i = trial_i + 1$
    $p_i \leftarrow \frac{fit(x_i)}{\sum_{n=1}^{SN} fit(x_n)}, t = 0, i = 1$
Scout Bee:
while $t \leq SN$
    if $rand\,(0, 1) < p_t$
        $x_i = v_{ij}, trial_i = 1$
    else
        $trial_i = trial_i + 1$
        $t = t + 1$
    else
        $i = i + 1$
    if $i = SN$,i=1
Onlooker :
    ifmax($trial_i$) > lim $it$
        $x_{ij} = x_{min,j} + rand\,(0, 1)\left(x_{max,j} - x_{min,j}\right)$
    if $FES >= MEN$
        Return $\theta.append(v_i)$
    else
        Go to Employed Bee
        $F_m\,(x) = 0$,i=1,N
    for m =1 to M do:
        $F_0\,(\mathrm{x}) = argmin_\beta \sum_{i=1}^{N} L\,(y_i, \beta)$
        for i =1 to N do:
            $y_i^* = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}, i = 1, 2, n, N$
            $a_m = argmin_{\mathrm{a},\beta} \sum_{i=1}^{N}\left[y_i^* - \beta h\,(x_i; a)\right]^2$
            $\beta_m = argmin_{\mathrm{a},\beta} \sum_{i=1}^{N} L\,(y_i, F_{m-1}\,(x_i) + \beta h\,(x_i; a_m))$
        end for
    end for
**return** $\theta$
```

Table 4
Classification Accuracy (%) of ABCoDT using the original dataset and the selected features.

	Accuracy				Improvement	
	Original		Selected		Rate	
Dataset	Max	Ave.	Max	Ave.	Max	Ave.
WDBC	95.1	93.7	97.9	92.8	2.8	−0.9
Haberman	84.2	73	85.5	74.3	1.3	1.3
Glass	74.1	62.6	75.9	58. 3	1.8	−4.3
Contraceptive	62.3	55.5	62	56.3	−0.3	0.8
Wine	95.5	92.0	97.7	93.6	2.2	1.6
ULC	88.8	83.5	89.9	83.2	1.1	−0.3
p53	74.8	76.2	78.1	76.5	3.3	0.3
Arcene	89.1	83.5	87.2	85.6	−1.9	2.1

Table 5
Number of selected features of ABCoDT versus six state-of-the-art methods.

Dataset	ABCoDT	GBDT	VT	SKB	RFE	L12	PCA
Haberman	1	2	3	1	3	1	1
WDBC	2	8	15	1	4	20	7
Glass	1	4	9	4	4	1	4
Contraceptive	2	2	9	4	4	9	3
Wine	1	3	13	4	4	13	3
ULC	5	45	147	1	4	147	10
p53	565	843	1216	753	4	565	874
Arcene	2103	3240	8102	41	4	2213	3380

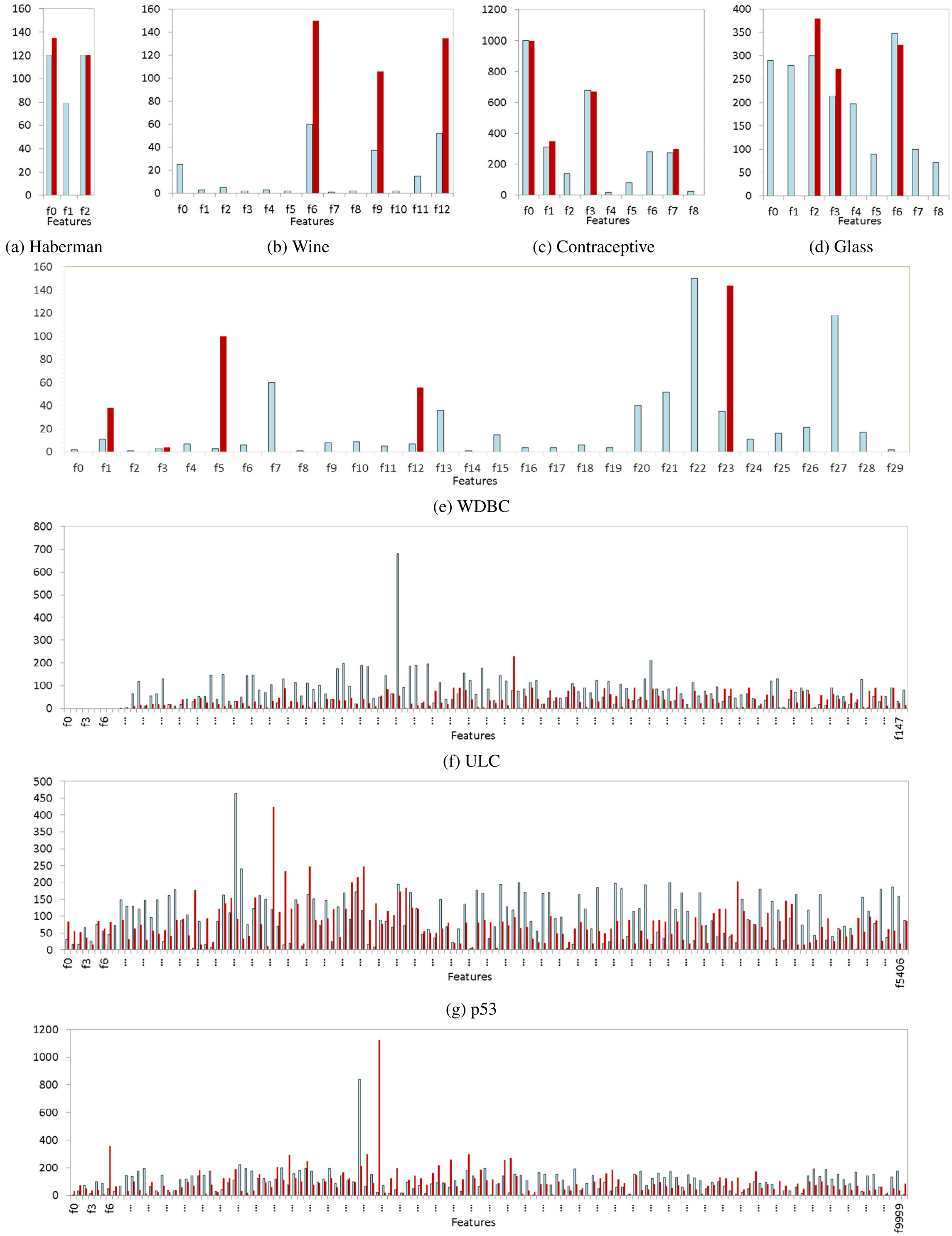
lists the classification accuracy with and without feature selection. Both maximum accuracy and average accuracy of the cross-validation are reported. The maximum classification accuracy from original dataset are 95.1%, 84.2%, 74.1%, 62.3%, 95.5%, 88.8%, 74.8, and 89.1 for the eight datasets. The maximum classification accuracy from features subset are 97.9%, 85.5%, 75.9%, 62%, 97.7%, 89.9%, 78.1%, and 87.2% for the eight datasets. After selecting features using ABCoDT, the number of features of five out of six datasets is reduced and the classification accuracy is improved. Although the classification accuracy of the dataset Contraceptive is lower after feature selection, the efficiency of the classifier is greatly improved as shown in Table 7. The comparison results show that features subset selected by our method could save most of the effective information of original datasets and our method is useful for improving learning performance.

We analyze the optimization results and classification accuracy of five feature selection methods. Fig. 3 shows that ABCoDT reduces more feature dimensions than GBDT, VT, RFE, and L12. The features reduction ratio of ABCoDT is much higher than that of GBDT. Besides, the classification accuracy of ABCoDT is greater than SKB while they have similar feature selections ratio.

We also compare our proposed ABCoDT method with GBDT and five feature selection methods including VT, SKB, RFE, L12, and PCA on the eight datasets by combining xgboost and 10-fold cross-validation. The feature retention results and the best classification accuracy are reported in Tables 5 and 6. The best results for each dataset are highlighted with boldface font. Classification accuracy of ABCoDT are 85.52%, 97.18%, 70.37%, 56.52%, 90.91%, 80.47%, 73.51%, 80.2%, respectively. Comparing to GBDT, VT, RFE, L12 and PCA, our method reduces more features than other feature selection algorithms and results in the six smallest subsets of selected features among the eight datasets. In the other two cases, the number of retained features is close to the best performers. When the number of original features in cases is within a certain range, the accuracy of our method is also superior to that of the other methods. When the number of features is oversize, the accuracy of classification is decreased but still higher than most of the other methods. In comparison to the second best cases, our method improves the accuracy of Haberman dataset after feature selection to 3.9%. The minimum improvement is about 1%.

4.4. Efficiency

Besides classification accuracy, execution time is another important indicator of the algorithm performance. Table 7 lists the running time (in seconds) of our method and six state-of-the-art feature selection methods, which are evaluated with different datasets. It is evident that our method used much less time on average in comparison to the other methods. This is even clearly demonstrated when the size of the dataset becomes larger (as shown for the datasets p53 and Arcene).



(a) Haberman (b) Wine (c) Contraceptive (d) Glass

(e) WDBC

(f) ULC

(g) p53

(h) Arcene

Fig. 3. Weights of features in the experimental datasets.

5. Conclusion

Data from real-world applications can be high dimensional and features of such data are usually highly redundant. Identifying informative features has become an important step for data mining to not only circumvent the curse of dimensionality but to reduce the amount of data for processing. To reduce the problem of high dimensionality, we propose a novel feature selection method based on bee colony and gradient boosting decision tree. The method initializes the feature space spanned by the dataset. Less relevant features are suppressed according to the information each feature contributes to the decision making using an artificial bee colony algorithm. Our method reduces the initial input of a gradient boosted decision tree algorithm and removes the features with a low correlation.

Table 6
The best accuracy of ABCoDT versus six state-of-the-art methods.

Dataset	ABCoDT	GBDT	VT	SKB	RFE	L12	PCA
Haberman	85.52	73.68	72.37	71.05	72.37	82.34	75.64
WDBC	97.18	96.48	93.66	71.83	88.03	90.85	91.34
Glass	70.37	68.51	59.26	37.37	14.81	20.37	65.3
Contraceptive	56.52	54.35	54.61	39.67	47.01	54.61	55.79
Wine	90.91	93.18	90.91	72.51	43.18	90.91	91.23
ULC	80.47	77.51	79.29	15.38	1.77	79.29	78.38
p53	73.51	73.56	68.15	73.21	69.38	75.10	77.28
Arcene	80.20	84.31	79.34	58.25	60.32	84.31	67.34

Table 7
Running time of feature selection algorithms (in second).

Dataset	ABCoDT	GBDT	VT	SKB	RFE	L12	PCA
Haberman	0.1401	0.4060	0.1860	0.2780	0.3620	0.3821	0.1245
WDBC	0.3020	0.1740	0.2810	0.1370	0.4510	0.2350	0.2987
Glass	0.1850	0.3560	0.3160	0.2470	0.2860	0.1960	0.2640
Contraceptive	0.2250	0.4190	0.3720	0.9481	0.2870	0.4040	0.3076
Wine	0.1360	0.1990	0.1710	0.1850	0.1990	0.9120	0.1290
ULC	0.3930	0.7370	1.2681	0.2900	5.0983	1.8641	1.3498
p53	14.3478	27.1368	15.2540	56.3870	18.3654	27.8420	24.5024
Arcene	23.6870	36.8810	84.2570	113.5642	28.6472	94.2360	45.0981

Experiments are conducted with two breast cancer datasets and six datasets from the public data repository. The feature reduction ratio of our proposed method is more than 60% for all cases. By applying our proposed ABCoDT method to the datasets, it is demonstrated that the number of features is successfully reduced without sacrificing the classification accuracy. In comparison to the state-of-the-art methods, ABCoDT improves the accuracy by up to 3.9% with respect to the second best cases. The minimum improvement is about 1%. Among the six datasets, our method results in the four smallest subsets of selected features. In the other two cases, the number of retained features is close to the best performers. In the evaluation of the efficiency, the proposed method outperformed the other methods in most cases. With a selected subset of data, the efficiency is much improved.

The success of ABCoDT provides a framework to integrate a classification method with a feature selection process. In our future work, we plan to explore the learning theories that are different from what is employed by decision trees. For axis-parallel decision trees, features are evaluated one by one in the tree construction process, which makes the evaluation of a natural consequence. If such an integrated system can be adopted by other learning algorithms requires further investigation.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant (3177167, 31371533, 31671589), the Anhui Foundation for Science and Technology Major Project, China under Grant (16030701092), the Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture of China under Grant (AEC2018003, AEC2018006), the 2016 Anhui Foundation for Natural Science Major Project, China of the Higher Education Institutions under Grant (kJ2016A836).

References

[1] X. Yuan, L. Xie, M. Abouelenien, A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data, Pattern Recognit. 77 (2018) 160–172.

[2] L. Gu, Y. Han, C. Wang, W. Chen, J. Jiao, X. Yuan, Module overlapping structure detection in ppi using an improved link similarity-based markov clustering algorithms, Neural Comput. Appl. (2018) 1–10, http://dx.doi.org/10.1007/s00521-018-3508-z (in press).

[3] E. Aličković, A. Subasi, Breast cancer diagnosis using ga feature selection and rotation forest, Neural Comput. Appl. 28 (4) (2017) 753–763.

[4] A.A. Wan, Cenan Freitas, An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features, Artif. Intell. Rev. 50 (2) (2018) 201–240.

[5] Y. Shen, X. Liu, X. Yuan, Fractal dimension of irregular region of interest application to corn phenology characterization, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (4) (2017) 1402–1412.

[6] K.D. Rodland, Proteomics and cancer diagnosis: the potential of mass spectrometry, Clin. Biochem. 37 (7) (2004) 579–583.

[7] A. Alsaffar, N. Omar, Study on feature selection and machine learning algorithms for malay sentiment classification, in: International Conference on Information Technology and Multimedia, IEEE, 2014, pp. 270–275.

[8] Y. Liu, F. Tang, Z. Zeng, Feature selection based on dependency margin, IEEE Trans. Cybern. 45 (6) (2015) 1209–1221.

[9] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar) (2003) 1157–1182.

[10] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (6) (2017) 94.

[11] M. Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP J. Image Video Process. 2013 (1) (2013) 47.

[12] X. Zhu, S. Zhang, R. Hu, Y. Zhu, J. Song, Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Trans. Knowl. Data Eng. 30 (3) (2018) 517–529.

[13] M.M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Applied Soft Computing 62 (2017) 441–453.

[14] H. Wang, Y. Meng, P. Yin, J. Hua, A model-driven method for quality reviews detection: an ensemble model of feature selection, in: Wuhan International Conference On E-Business, 2016, p. 2.

[15] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. statist. (2001) 1189–1232.

[16] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1–2) (2003) 155–176.

[17] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (3) (2013) 483–519.

[18] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.

[19] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Syst. Appl. 36 (3) (2009) 6843–6853.

[20] R. Sikora, S. Piramuthu, Framework for efficient feature selection in genetic algorithm based data mining, European J. Oper. Res. 180 (2) (2007) 723–737.

[21] M.E. Farmer, S. Bapna, A.K. Jain, Large scale feature selection using modified random mutation hill climbing, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004.

[22] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, European J. Oper. Res. 171 (3) (2006) 842–858.

[23] M. Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP J. Image Video Process. 2013 (1) (2013) 47.

[24] J. Bi, K. Bennett, M. Embrechts, C. Breneman, M. Song, Dimensionality reduction via sparse support vector machines, J. Mach. Learn. Res. 3 (Mar) (2003) 1229–1243.

[25] P. Shunmugapriya, S. Kanmani, R. Supraja, K. Saranya, et al., Feature selection optimization through enhanced artificial bee colony algorithm, in: International Conference on Recent Trends in Information Technology, Chennai, India, 2013, pp. 56–61.

[26] S. Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, Soft Comput. 22 (3) (2018) 811–822.

[27] M.S. Uzer, N. Yilmaz, O. Inan, Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification, Sci. World J. 2013 (2013) 419187:1–10.

[28] X. Yuan, M. Abouelenien, A multi-class boosting method for learning from imbalanced data, Int. J. Granul. Comput. Rough Sets Intell. Syst. 4 (1) (2015) 13–29.

[29] Z. Xu, G. Huang, K.Q. Weinberger, A.X. Zheng, Gradient boosted feature selection, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, 2014, pp. 522–531.

[30] E. Tuv, A. Borisov, G. Runger, K. Torkkola, Feature selection with ensembles, artificial variables, and redundancy elimination, J. Mach. Learn. Res. 10 (Jul) (2009) 1341–1366.

[31] J. Ramos-González, D. López-Sánchez, J.A. Castellanos-Garzón, J.F. d. Paz, J.M. Corchado, A cbr framework with gradient boosting based feature selection for lung cancer subtype classification, Comput. Biol. Med. 86 (2017) 98–106.

[32] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Global Optim. 39 (3) (2007) 459–471.

[33] B. Wu, C.H. Qian, Differential artificial bee colony algorithm for global numerical optimization, J. Comput. 6 (5) (2011) 841–848.

[34] M. Cheng, Z. NIZW, Lifecycle-based binary ant colony optimization algorithm, Pattern Recognit. Artif. Intell. 27 (11) (2014) 1005–1014.

[35] S. Suthaharan, Machine Learning Models and Algorithms for Big Data Classification, Springer, 2016.

[36] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, 2016, pp. 785–794.