Banzhaf random forests: Cooperative game theory based random forests with consistency

Jianyuan Sun [a], Guoqiang Zhong [a,*], Kaizhu Huang [b], Junyu Dong [a]

[a] Department of Computer Science and Technology, Ocean University of China, 238 Songling Road, Qingdao 266100, China
[b] Department of Electrical and Electronic Engineering, Xian Jiaotong-Liverpool University, SIP, Suzhou 215123, China

HIGHLIGHTS

- A novel random classification forests algorithm, called Banzhaf random forests (BRFs), is proposed.
- The Banzhaf power index is employed to evaluate the power of each feature by traversing possible feature coalitions.
- The consistency of BRFs is proved.

ARTICLE INFO

Article history:
Received 29 August 2017
Received in revised form 14 April 2018
Accepted 8 June 2018
Available online 28 June 2018

Keywords:
Random forests
Cooperative game
Banzhaf power index
Consistency

ABSTRACT

Random forests algorithms have been widely used in many classification and regression applications. However, the theory of random forests lags far behind their applications. In this paper, we propose a novel random forests classification algorithm based on cooperative game theory. The Banzhaf power index is employed to evaluate the power of each feature by traversing possible feature coalitions. Hence, we call the proposed algorithm Banzhaf random forests (BRFs). Unlike the previously used information gain ratio, which only measures the power of each feature for classification and pays less attention to the intrinsic structure of the feature variables, the Banzhaf power index can measure the importance of each feature by computing the dependency among the group of features. More importantly, we have proved the consistency of BRFs, which narrows the gap between the theory and applications of random forests. Extensive experiments on several UCI benchmark data sets and three real world applications show that BRFs perform significantly better than existing consistent random forests on classification accuracy, and better than or at least comparable with Breiman's random forests, support vector machines (SVMs) and k-nearest neighbors (KNNs) classifiers.

1. Introduction

Machine learning is an important sub-area of artificial intelligence. It includes many branches, such as ensemble leaning and deep learning. In recent years, a great deal of work shows the effectiveness of deep learning in solving practical problems (Amozegar & Khorasani, 2016; Bulo & Kontschieder, 2014; Kim, Jang, & Lee, 2016; Mcquoid, 1993; Pavel, Schulz, & Behnke, 2017; Roy & Todorovic, 2016; Scardapane & Di, 2017). However, in order to obtain better performance, it tends to combine the ideas of deep learning and ensemble learning. For instance, some work tries to improve the performance of deep learning models based on ensemble learning (Amozegar & Khorasani, 2016; Mcquoid, 1993; Scardapane & Di, 2017), while some uses deep learning models for feature extraction and ensemble learning models for classification or regression (Bulo & Kontschieder, 2014; Roy & Todorovic, 2016). In this paper, we focus on an ensemble learning method, random forest, which is mainly based on the combination of several independent decision trees (Breiman, 2001).

As a general classification and regression tool, the random forests algorithm and its variants (Ayerdi & Grana, 2014; Ristin, Guillaumin, Gall, & Van Gool, 2016; Zhang & Suganthan, 2014) have been successfully applied in many fields, such as computer vision (Dollar & Zitnick, 2014; Hallman & Fowlkes, 2015; Shotton, Sharp, Kipman, Fitzgibbon, Finocchio, Blake, Cook, & Moore, 2013; Zikic, Glocker, & Criminisi, 2013) and pattern recognition (Bosch, Zisserman, & Muoz, 2007; Shotton, Johnson, & Cipolla, 2008; Yin, Criminisi, Winn, & Essa, 2007). Nevertheless, the theory of random forests lags far behind their applications. Fortunately, Biau et al. have made a breakthrough in theoretical research of the random forests algorithms in recent years (Biau, 2012; Biau, Devroye, & Lugosi, 2008). However, as they mainly focus on the consistency of

* Corresponding author.
E-mail address: gqzhong@ouc.edu.cn (G. Zhong).

the random forests algorithms, the proposed algorithms generally perform not very well.

In this paper, we propose a new random classification forests algorithm based on the cooperative game theory, and call it Banzhaf random forests (BRFs). BRFs are formed with a number of Banzhaf decision trees (BDTs). For each BDTs, we adopt the Banzhaf power index to evaluate the "best" feature at each tree node. The Banzhaf power index is a method to calculate solution of the cooperative game, which can be used to explore the internal structure of the feature variables. More importantly, based on previous work on the consistency of classifiers (Biau, 2012; Biau et al., 2008), we have proved the consistency of BRFs.

The remainder of this paper is organized as follows. In Section 2, we review some related work on random forests algorithms. In Section 3, we describe two critical issues for constructing decision trees in random forests. In Section 4, we introduce the proposed BRFs algorithm in detail. Section 5 is devoted to the proof of the consistency of BRFs. In Section 6, we report the experimental results obtained by BRFs and the compared algorithms on several UCI data sets. Section 7 concludes this paper.

2. Related work

The original random forests algorithm (Breiman, 2001), which combines several classification and regression trees (CART) (Breiman, Friedman, Stone, & Olshen, 1984) or C4.5 decision trees (Salzberg, 1994) using bagging (Breiman, 1996), is proposed by Breiman. The construction of random forests algorithm is mainly based on three pieces of existing work: the feature selection work of Amit and Geman (1997), the random subspace method of Ho (1998) and the method of random split selection of Dietterich (2000). Later on, Criminisi et al. present a unified framework for random forests models (Criminisi, Shotton, & Konukoglu, 2012).

Random forests models have an excellent classification and regression performance. Therefore, they have been applied to a wide variety of real world applications (Criminisi & Shotton, 2013; Cutler, Edwards Jr, Beard, Cutler, Hess, Gibson, & Lawler, 2007; Prasad, Iverson, & Liaw, 2006; Svetnik, Liaw, Tong, Culberson, Sheridan, & Feuston, 2003). Although random forest algorithms have achieved great successes in practice, the mathematical properties behind them are difficult to be analyzed (Breiman, 2004). There are two main properties of the theory related to random forests models. The first is the consistency of the models, i.e. whether they can converge to an optimal solution as the data set grows infinitely large. The second is the rate of the convergence. In this work, we focus on the consistency aspect of the proposed BRFs algorithm. Note that, Biau et al. have showed that the consistency of Breiman's random forests cannot be theoretically guaranteed (Biau et al., 2008).

In recent years, many researchers have devoted efforts to the study of the consistency of random forests algorithms. Meinshausen has presented a consistent random regression forests algorithm, called quantile regression forests (Meinshausen, 2006). Ishwaran and Kogalur have demonstrated that their proposed algorithm, survival forests, possesses consistency (Ishwaran & Kogalur, 2010). In addition, Denil et al. have developed an online version of the random forests algorithms and proved its consistency (Denil, Matheson, & de Freitas, 2013b). Moreover, a new random regression forests algorithm was given by Denil, Matheson, and De Freitas (2013a). However, the consistency of these existing random forests algorithms only focuses on the online learning and regression problems. A significant contribution to the theory of random forests is the work by Biau et al. (2008), which proves the consistency of various randomized ensemble classifiers and investigates the consistency of bagging rules (Breiman, 1996). Biau et al. have also proposed two consistent random forests classifiers: the pure random forests and the scale-invariant version of the random forests (Biau et al., 2008). More importantly, Biau et al. suggest that various greedy random forest classifiers, including Breiman's random forests classifier, are inconsistent. To remedy the inconsistency of these random forests classifier, some techniques need to be employed (Györfi, Devroye, & Lugosi, 1996). For instance, the decision trees in random forests use different stopping rules instead of growing every decision tree down to nodes with a single data point.

In this paper, we propose an innovative random classification forests algorithm based on the cooperative game theory. It uses the Banzhaf power index to evaluate the power of each feature by traversing all possible feature coalitions, and the midpoint of the most powerful feature is used to split the node. According to the existing theoretical work, the consistency of the proposed random forests algorithm has been proved. Furthermore, extensive experiments show that the performance of the proposed random forests algorithm is significantly better than that of existing consistent random forests, and better than or at least comparable with Breiman's random forests and other state-of-the-art classifiers.

3. Two critical issues in random forests

In general, when some decision trees are employed in a random forests algorithm, two critical issues need to be considered. The first one is the method for splitting the tree nodes, and the second one is the method for injecting randomness into the trees. At the same time, the strategic choice of these two critical issues also determines whether a random forests algorithm is consistent.

Specifying a method for splitting tree nodes needs to select the shapes of candidate splits and a criterion for evaluating the quality of each candidate split point or feature. Typical choices are to use axis aligned splits, where sample data are routed to sub-trees depending on whether or not they exceed a threshold value of a chosen feature; or linear splits, where a linear combination of features is compared with a threshold to make a decision (Breiman, 2001). The threshold value in either case can be chosen randomly or by optimizing a function of the data in the tree nodes.

For the used split point, a simple method is to choose among the candidate split points at random, such as a random forests algorithm of Biau et al. (2008). A more common method is to choose the "best" split points, which optimizes a purity function to travel the possible candidate split point of each feature in a tree node. A typical choice is to maximize the Gini index or the information gain ratio (Breiman, 2001; Friedman, 2001; Geurts, Ernst, & Wehenkel, 2006). Recently, Biau (2012) proposed a new method to split the tree node in their random forests model (called Biau12, in our paper). They had not used the traversal way to compute the "best" split point. Alternatively, the midpoint of the most important feature was used to split the node. In this way, Biau12 algorithm can achieve consistency. In our paper, based on the theoretical results of Biau et al., we proved the consistency of the proposed BRFs algorithm. In particular, to ensure that the proposed BRFs algorithm has consistency, the midpoint value of the "best" feature was used as split point to expand tree nodes.

In addition, Breiman claims that randomness can help to reduce the correlation between tree classifiers in a random forests algorithm, and meanwhile, maintain reasonable strength of each tree (Breiman, 2001). Therefore, injecting randomness is very important to improve the performance of random forests models. In general, injecting randomness into each tree can be achieved in several ways. Which feature to be chosen for splitting the tree node can be random, and the split point can be chosen either randomly or by optimization over some or all of the data at each tree node. In this work, we randomly select some data samples and features to construct the decision trees.

4. Banzhaf random forests

In this section, we present the construction rules of Banzhaf random forests (BRFs). Banzhaf random forests are formed by combining the prediction of several Banzhaf decision trees (BDTs). The idea of constructing the BDTs mainly motivated by Banzhaf power index that comes from the cooperative game theory. Therefore, first, we start with some basic concepts of the cooperative game theory and the Banzhaf power index. Second, we introduce the method of constructing the randomized BDTs. Third, the construction rules of BRFs are given. Finally, we discuss the computation issue and present the prediction method of BRFs.

4.1. Some basic concepts of the cooperative game theory

Cooperative game is a game where groups of players ("coalitions") may enforce cooperative behavior, while the game is a competition between coalitions of players rather than between individual players. Cooperative game theory mainly searches for an 'acceptable' way to distribute gains to each individual player in the cooperative game (Chalkiadakis, Elkind, & Wooldridge, 2011). The mathematical definition of cooperative games (Chalkiadakis et al., 2011) can be read as follows.

Formally, the cooperative game $\Gamma = (\mathcal{N}, \gamma)$ consists of a finite set of player $\mathcal{N} = \{1, 2, \ldots, n\}$, called the grand coalition, and a characteristic function $\gamma : 2^{\mathcal{N}} \rightarrow \mathbf{R}$. For each subset $\mathcal{S} \subseteq \mathcal{N}$, $\gamma(\mathcal{S})$ represents the profit achieved by the players of $\mathcal{S} \subseteq \mathcal{N}$ by accomplishing the task together. In general, the goal in a cooperative game is to distribute the total gains $\gamma(\mathcal{N})$ to each player $i (i = 1, 2, \ldots, n)$ who belongs to the grand coalition $\mathcal{N}$ in a fair and reasonable way.

Obviously, the grand player set $\mathcal{N}$ gets profit more than that of any player subset $\mathcal{S} \subset \mathcal{N}$ in a cooperative game $\Gamma = (\mathcal{N}, \gamma)$, i.e., $\gamma(\mathcal{N}) > \gamma(\mathcal{S})$. Only meeting this condition, the players in $\mathcal{N}$ are willing to cooperate. Otherwise, no cooperation is necessary. In particular, a game is a cooperative game, which need to satisfy the superadditivity (Saad, Han, Debbah, & Hjorungnes, 2009). Superadditivity implies that, given any two disjoint player subsets $\mathcal{S}_1$ and $\mathcal{S}_2$, if coalition $\mathcal{S}_1 \bigcup \mathcal{S}_1$ forms, then it can guarantee at least the profit that is obtained by the disjoint coalitions separately. i.e., $\gamma(\mathcal{S}_1) \bigcup \gamma(\mathcal{S}_2) \geq \gamma(\mathcal{S}_1) + \gamma(\mathcal{S}_2)$, $\forall \mathcal{S}_1 \subset \mathcal{N}, \mathcal{S}_2 \subset \mathcal{N}$ and $\mathcal{S}_1 \cap \mathcal{S}_2 = \varnothing$. Therefore, how much profit to be obtained and how to distribute the gains are an important factor to the players in the cooperation.

Moreover, for the distribution of total gains, different requirements of fairness and rationality yield different solution concepts for the cooperative game, such as 'the nucleolus', 'the Shapley value', 'Banzhaf power index' and other concepts. Among these solution concepts, the nucleolus and the Shapley value focus on providing the expected payment for each player in the coalition, while the Banzhaf power index focuses on evaluating the power or importance of each player in the coalition (Feltkamp, 1995). In particular, the Banzhaf power index focuses more on the fairness of the distribution gains. For the proposed BRFs algorithm, we try to search the node feature with the strongest discriminative ability for classification tasks, according to the intricate and intrinsic interrelation among candidate features. That is, the most powerful or important feature is selected as the split feature at each tree node. Therefore, in this work, we use the Banzhaf power index to evaluate the power or importance of the candidate features at each tree node.

The definition of Banzhaf power index is described in Banzhaf III (1964). The original Banzhaf power index is used to evaluate the power of players in a simple game. The simple game is a cooperative game due to satisfying the superadditivity (Saad et al., 2009). A simple game $\Gamma = (\mathcal{N}, \gamma)$ consisting of a player set $\mathcal{N} = \{1, 2, \ldots, n\}$ with $|\mathcal{N}| = n$, the coalition $\mathcal{S}$ with value 1 is considered to be 'winning', and that with value 0 is considered to be 'losing', i.e. $\forall \mathcal{S} \subseteq \mathcal{N}$, $\gamma(\mathcal{S}) = 1$ and $\gamma(\mathcal{S}) = 0$, respectively. The phenomenon that coalition $\mathcal{S} \cup \{i\}$ wins but $\mathcal{S}$ loses is called a swing of player $i \in \mathcal{N}$, because the player i in the coalition $\mathcal{S} \cup \{i\}$ is crucial to its 'winning'. In fact, the Banzhaf power index of a player $i \in \mathcal{N}$ is the probability of swings of player i. Here, the symbol $\beta_i(\Gamma)$ is used to represent the Banzhaf power index of player $i \in \mathcal{N}$, and it is given by

$$\beta_i(\Gamma) = \frac{1}{2^{n-1}} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}} \Delta_i(\mathcal{S}), \tag{1}$$

where 2^{n-1} represents the total number of player subsets $\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}$ and $\Delta_i(\mathcal{S})$ is the marginal contribution of player i, i.e. $\Delta_i(\mathcal{S}) = \gamma(\mathcal{S} \cup \{i\}) - \gamma(\mathcal{S})$.

From Eq. (1), the Banzhaf power index of player $i (i \in \mathcal{N})$ is used to count the number of 'winning' coalitions when the player i joins in the 'losing' coalitions $\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}$. To find the most powerful player i that can make the majority of coalitions 'winning', the normalized Banzhaf power index $\zeta_i(\Gamma)$ is defined as

$$\zeta_i(\Gamma) = \frac{\beta_i(\Gamma)}{\sum_{i \in \mathcal{N}} \beta_i(\Gamma)}, \tag{2}$$

where $\sum_{i \in \mathcal{N}} \beta_i(\Gamma)$ is the total number of 'winning' coalitions of all players in the game.

Banzhaf power index has a particularly attractive interpretation—it measures the power of each player in a cooperative game, i.e. the probability that this player yields a good or bad result to a game. Some famous and vivid examples in Banzhaf III (1964) may help the readers for a better understanding of the Banzhaf power index. In this paper, in order to construct each BDT, we employ the Banzhaf power index to evaluate the power of the candidate features at each tree node.

4.2. Construction of BDTs

BRFs are composed of several Banzhaf decision trees (BDTs). Fig. 1 shows the structure of a single BDTs. For the root node, the "best" feature is selected by the criterion of the information gain ratio. For all of the other nodes except the leaves, the "best" features (players in the cooperative game) are selected by the Banzhaf power index. At each step, the midpoint value of the "best" feature is selected as the split point, which can be considered as a threshold ε. If there is only one data point in each tree node or a predefined number of cuts has been reached, then the tree stops growing.

We apply Eq. (1) to the process of constructing BDTs. At each tree node, the Banzhaf power index of feature f_i is computed as follows.

Each step of constructing the BDTs can be modeled as a simple game $\Gamma = (\mathcal{N}, \gamma)$, which consists of a feature player set $\mathcal{N} = \{f_1, f_2, \ldots, f_n\}$. i.e., $\gamma(S) \in \{0, 1\}$, $\forall S \subseteq N$ and $\gamma(N) = 1$. In particular, the simple game satisfies the superadditivity (Saad et al., 2009). Therefore, the problem of computing the Banzhaf power index of features at each node of BDTs is a cooperative game. Then, let the coalition $\mathcal{S}$ be a subset of the features and $f_i (f_i \notin \mathcal{S})$ be a feature to be estimated.

Whether feature f_i leads a coalition $\mathcal{S}$ to enter the 'winning' state can be measured by the ratio $\sigma = \mu_{f_i}(\mathcal{S}) / \rho_{f_i}(\mathcal{S})$, where $\mu_{f_i}(\mathcal{S})$ is the number of features (belonging to the coalition $\mathcal{S}$) interdependent with the features $f_i (f_i \notin \mathcal{S})$, and $\rho_{f_i}(\mathcal{S})$ is the total number of features in the coalition $\mathcal{S}$. For convenience, the symbol τ is used to represent the splitting threshold for the ratio σ, and we set

$\tau = 1/2$. If $\sigma < \tau$, the coalition $\mathcal{S} \cup \{f_i\}$ is 'losing', otherwise, it is 'winning', i.e.

$$\Delta_{f_i}(\mathcal{S} \cup f_i) = \begin{cases} 1 & \sigma \geq \tau; \\ 0 & \sigma < \tau. \end{cases}$$

The threshold $\tau = 1/2$ means, if more than half of the features of a coalition $\mathcal{S}$ are interdependent with f_i, then f_i joining can make coalition $\mathcal{S}$ enter the 'winning' state. Hence, for simplicity of the computation, we specify $\Delta_i(\mathcal{S})$ for a single coalition $\mathcal{S} \subseteq \mathcal{N} \setminus \{f_i\}$ in Eq. (1) as

$$\Delta_i(\mathcal{S}) = \gamma(\mathcal{S} \cup \{i\}) - \gamma(\mathcal{S}) = \begin{cases} 1 & \sigma \geq \tau; \\ 0 & \sigma < \tau. \end{cases} \tag{3}$$

In this work, we use the conditional mutual information to evaluate the interdependence between a single feature $f_i \in \mathcal{N} \setminus \mathcal{S}$ and the feature player $f_j \in \mathcal{S} \subseteq \mathcal{N}$. In general, the conditional mutual information $I(X; Y|Z)$ is defined as information shared by random variables X and Y when variable Z is given. It can be formally defined as

$$I(X; Y|Z) = \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} p(x, y, z) \log \frac{p(x, y|z)}{p(x|z)p(y|z)}.$$

Conditional mutual information is also widely used to measure the reduction of the uncertainty of X due to the knowledge of Y when Z is given.

In our paper, the conditional mutual information is used to measure the interdependency between a single feature player $f_i \notin \mathcal{S}$ and the target class label $\mathbf{y}$, given feature player $f_j \in \mathcal{S}$. It is defined by

$$I(f_i; \mathbf{y}|f_j) = p(f_i, f_j, \mathbf{y}) \log \frac{p(f_i, \mathbf{y}|f_j)}{p(f_i|f_j)p(\mathbf{y}|f_j)}. \tag{4}$$

Two feature variables f_i and f_j are interdependent on each other, if the relevance between f_i and the target class $\mathbf{y}$ increases given f_j, i.e. $I(f_i; \mathbf{y}) \leq I(f_i; \mathbf{y}|f_j)$, where $I(f_i; \mathbf{y})$ is the mutual information between the feature f_i and the class label $\mathbf{y}$.

By Eqs. (1), (3), (2) and (4), we can obtain the Banzhaf power index of each candidate feature for each tree node. More concretely, this computation process is described in Algorithm 1.

Algorithm 1: Selecting the best feature using the Banzhaf power index

Input: A tree node contains data set D_n with feature space $\mathcal{F}$ and the data labels $\mathbf{y}$.
Output: β: Banzhaf power index vector of $\mathcal{F}$.

```
β = 0, τ = 1/2;
For each feature i ∈ F do
  Create coalitions set {S_1, ..., S_t} over F \ {i};
    For each feature S_j ∈ {S_1, ..., S_t} do
    Calculate marginal function Δ_i(S_j) using Eq. (3) and Eq. (4);
     End
  Calculate the Banzhaf power index β_i using Eq. (1);
End
Normalized the value β_i using Eq. (2) to obtain ζ_i;
Choose the best feature with max_i ζ_i for this tree node.
```

For clarity, we give an example to demonstrate how the Banzhaf power index can be used to evaluate the power of a feature. Given a cooperative feature game $\Gamma = (\mathcal{N}, \gamma)$ with the feature player set $\mathcal{N} = \{f_1, f_2, f_3, f_4\}$. Our goal is to calculate the Banzhaf power index of the feature f_4. For all coalitions $\mathcal{S} \subseteq \mathcal{N} \setminus \{f_4\}$, the total number of possible coalitions of feature subsets $\mathcal{N} \setminus \{f_4\}$ is 8. Assume that the feature f_4 can make the coalitions $\{f_2\}$, $\{f_2, f_3\}$ and $\{f_1, f_2\}$ enter the 'winning' state, i.e. each coalition has half of its features interdependent with feature f_4. Then the Banzhaf power index of f_4 can be computed as

$$\beta_{f_4}(\Gamma) = \frac{1}{2^{4-1}} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{f_4\}} \Delta_{f_4}(\mathcal{S}) = 3/8.$$

where $\sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{f_4\}} \Delta_{f_4}(\mathcal{S}) = \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{f_4\}} (\gamma(\mathcal{S} \cup \{f_4\}) - \gamma(\mathcal{S})) = 1 + 1 + 1 = 3$. Similarly, the Banzhaf power index of other features can be computed in a similar way.

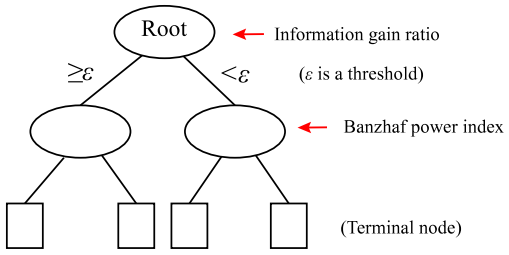

Fig. 1. A Banzhaf decision tree.

4.3. Banzhaf random forests algorithm

Given a training data set $D_n = (\mathbf{X}_i, Y_i)_{i=1}^n$, including n samples with dimensionality M, the learning of BRFs is based on the general technique of bootstrap aggregating, or bagging, and the constructed BDTs. The BRFs algorithm can be formally described as follows.

- Using the bagging method to generate *ntree* subsets, $\{d_1, d_2, \ldots, d_{ntree}\}$, where *ntree* is the number of trees in BRFs. Concretely, for each tree, randomly sampling D_n for n times with replacement.
- For each data set d_i, a Banzhaf decision tree is built. Before the decision tree construction, randomly sample a subspace of $h = round(log2(M) + C)$ $(h \ll M)$ features from the available features presented in d_i, where $C \in \mathbf{R}$ is a parameter. For the root node, the "best" feature is selected by the criterion of information gain ratio. For the other nodes except the terminal ones, the "best" feature is selected among the h features based on the Banzhaf power index. The midpoint value of the "best" feature is used as a split threshold to generate left and right child nodes. Repeat this process until reaching the user-set limit (i.e. the percentage of incorrect points or a minimal number of samples at a node).
- Integrating *ntree* BDTs $h_1(d_1), h_2(d_2), \ldots, h_{ntree}(d_{ntree})$ to form a Banzhaf random forest (BRF), and this BRF uses the majority votes of *ntree* BDTs to obtain the class prediction for a new sample.

It is easy to see that the BRFs algorithm is similar to the original random forests algorithm by Breiman (2001). Both of them use bootstrap aggregating, i.e. the bagging ensemble method. The main difference between them is the way of selecting the split point (threshold) for each tree node. For the original random forests, the information gain ratio is employed to traverse each possible split point at each candidate feature, accordingly, the "best" split point and the corresponding feature is obtained. While BRFs algorithm employs a different way to obtain the split point at each tree node. BRFs first use the Banzhaf power index to evaluate the "best" feature, and then the midpoint value of the "best" feature is selected as the split point.

4.4. Computational issue

To evaluate the power of each feature, it is necessary to calculate the proportion of the 'winning' coalitions. Theoretically, calculating the Banzhaf power index requires summing over all possible feature subsets, which may lead to high computational complexity. However, empirically, it is unnecessary to consider empty-set and large coalitions. In most cases, there is a small probability that a single feature f_i results in a large coalition to be 'winning'. Hence, we set a bound ϖ for the coalition size. To this end, Eq. (1) can be redefined as

$$\beta_i(\Gamma) = \frac{1}{|\Pi_\varpi|} \sum_{\mathcal{S} \subseteq \Pi_\varpi} \Delta_i(\mathcal{S}),$$

where Π_ϖ is the subset of the feature set $\mathcal{F} \setminus \{f_i\}$ (except $\varnothing$), with a number of elements less than or equal to ϖ.

Moreover, in our proposed method, whether a coalition win or not depends on the number of features increasing (or reducing) its associate with the target class when the condition is given. Therefore, at each node of BDTs, the number of winning coalitions containing only one member (denoted as M_1) that can be calculated with time complexity $O(n)$, where n denotes the number of features at the corresponding tree node. Then, we can calculate the number of winning coalitions that including more than one member based on M_1 according to the knowledge of combinatorial mathematics and dynamic programming technique. In particular, dynamic programming is an efficient programming technique for solving the combinatorial problems (Cormen, Leiserson, Rivest, & Stein, 2009). For example, M_2 can be calculated as $M_2 = C^2_{M_1} + C^1_{M_1} \times C^1_{n-M_1}$, where C represents the number of combinations. In this way, each BDTs in BRFs can be constructed with low computational complexity.

More specifically, to determine the value of ϖ for most application, we used 5-fold cross-validation to choose the value of ϖ for all of the experiments (shown in Section 6). The experimental results demonstrated that, when $\varpi \in [3, 5]$, the performance of BRFs is satisfactory. Thus, we suggest a range of $[3, 5]$ for ϖ in most of the applications. Based on the above discussion, the computational complexity of constructing BRFs is acceptable for real world applications.

4.5. Prediction

We denote a Banzhaf decision tree (BDT) created in the BRFs algorithm as g_n. To make a prediction for a query point $\mathbf{x}$, each BDT computes, for each class k,

$$\eta^k_n(\mathbf{x}) = \frac{1}{N(A_n(\mathbf{x}))} \sum_{(\mathbf{X}_i, Y_i) \in A_n(\mathbf{x})} \delta(Y_i = k),$$

where $(\mathbf{X}_i, Y_i)$ is i.i.d. pairs of random variables, $\mathbf{X}$ (the feature vector) takes its value in $\mathbf{R}^{(M+1)}$, Y (the label) is a multi-class random variable, $A_n(\mathbf{x})$ denotes the leaf node of the tree containing $\mathbf{x}$, and $N(A_n(\mathbf{x}))$ is the number of points that are located in $A_n(\mathbf{x})$. The tree prediction is then the class that maximizes the value η^k_n, i.e.

$$g_n(\mathbf{x}) = \arg\max_k \{\eta^k_n(\mathbf{x})\}.$$

BRFs predict the class that receives the most votes from the individual BDT.

5. Consistency of BRFs

In this section, we discuss and present the theoretical results about the consistency of Banzhaf random forests (BRFs). In particular, under some assumptions, we prove that BRFs are consistent.

5.1. Theoretical results

We denote the Banzhaf decision tree (BDT) created by the BRFs algorithm from n data points as g_n. As n varies, a sequence of BDT classifiers can be obtained, i.e. $\{g_n\}$. Then, we focus on showing that the sequence $\{g_n\}$ is consistent. According to the work of Devroye et al. (Györfi et al., 1996), a sequence $\{g_n\}$ of BDT classifiers is consistent, when the probability of error of g_n converges to the Bayes risk L^*, i.e.

$$L(g_n) = \mathbb{P}(g_n(\mathbf{X}, \theta, D_n) \neq Y) \to L^*,$$

as $n \to \infty$, where $(\mathbf{X}, Y)$ is a random test data point, θ represents the randomness of constructing BDT, such as randomly selecting a group of features to evaluate the "best" features for each node, and D_n is the training data set. The Bayes risk L^* is the minimum of the prediction error of the Bayes classifier for the distribution of $(\mathbf{X}, Y)$, which makes predictions by choosing the class with the highest posterior probability, $g(\mathbf{x}) = \arg\max_k \mathbb{P}(Y = k|\mathbf{X} = \mathbf{x})$. For more explanation about this setting, please refer to the work by Györfi et al. (1996).

In order to reduce the complexity of the problem, we try to reduce the problem of proving the consistency of multi-class classifier $\{g_n\}$ to prove the consistency of the transformed two class classification problems. Inspired by the work of Denil et al. (2013b), we give the lemma as follows.

Lemma 1. *Suppose the probability estimates, $\eta^k_n(\mathbf{x})$, for each class posterior $\eta^k(\mathbf{x}) = \mathbb{P}(Y = k|\mathbf{X} = \mathbf{x})$, is consistent (as $n \to \infty$, tends to the class Bayes probability). Then the classifier*

$$g_n(\mathbf{x}) = \arg\max_k \{\eta^k_n(\mathbf{x})\}$$

is consistent for the corresponding multi-class classification problem as $n \to \infty$.

Proof. In the case, where each class posterior $\eta^k(\mathbf{x})$ is equal, there is nothing to prove, since all results have the same probability of error. So, suppose there is at least one k such that $\eta^k(\mathbf{x}) < \eta^{g(\mathbf{x})}(\mathbf{x})(g(\mathbf{x}) = \arg\max_k \{\eta^k(\mathbf{x})\})$ and define

$$m(\mathbf{x}) = \eta^{g(\mathbf{x})}(\mathbf{x}) - \max_k \{\eta^k(\mathbf{x}) | \eta^k(\mathbf{x}) < \eta^{g(\mathbf{x})}(\mathbf{x})\},$$

$$m_n(\mathbf{x}) = \eta^{g(\mathbf{x})}_n(\mathbf{x}) - \max_k \{\eta^k_n(\mathbf{x}) | \eta^k(\mathbf{x}) < \eta^{g(\mathbf{x})}(\mathbf{x})\},$$

where $m(\mathbf{x})$ is the margin function for each class which measures how much better the best result is than the second best result. By assumption, the probability estimates of each class posterior is consistent, so $m(\mathbf{x}) \geq 0$. Similarly, the function $m_n(\mathbf{x})$ measures the margin for multi-class $g_n(\mathbf{x})$. If $m_n(\mathbf{x}) > 0$, then $g_n(\mathbf{x})$ has the same probability of error as the Bayes classifier. Accordingly, we just need to prove $m_n(\mathbf{x}) > 0$.

Based on the above guarantees, there is some ϵ such that $m(\mathbf{x}) > \epsilon$. Because the probability estimates $\eta^k_n(\mathbf{x})$ of each class posterior η^k is consistent, by making n large it can satisfy

$$\mathbb{P}(|\eta^k_n(\mathbf{X}) - \eta^k(\mathbf{X})| < \varepsilon/2) \geq 1 - \delta,$$

where $\delta \in (0, 1)$ is arbitrary, then we have

$$\begin{aligned} m_n(\mathbf{X}) &= \eta^{g(\mathbf{X})}_n - \max_k \{\eta^k_n(\mathbf{X}) | \eta^k(\mathbf{X}) < \eta^{g(\mathbf{X})}(\mathbf{X})\} \\ &\geq (\eta^{g(\mathbf{X})} - \epsilon/2) - \max_k \{\eta^k_n(\mathbf{X}) + \epsilon/2 | \eta^k(\mathbf{X}) < \eta^{g(\mathbf{x})}(\mathbf{X})\} \\ &= \eta^{g(\mathbf{X})} - \max_k \{\eta^k(\mathbf{X}) | \eta^k(\mathbf{X}) \langle \eta^{g(x)}(\mathbf{X})\} - \epsilon) 0. \end{aligned}$$

Since $\delta \in (0, 1)$ is arbitrary, this means that the risk of $g_n(\mathbf{x})$ converges in probability to the Bayes risk. □

Lemma 1 allows us to transform the proof of the consistency of multi-class tree classifier to prove the consistency of the corresponding two class classifiers, i.e. given a set of classes $\{1, \ldots, c\}$, we can re-assign the labels by using the map $(\mathbf{X}, Y) \mapsto (\mathbf{X}, \mathcal{I}(Y = k))$ for any $k \in \{1, \ldots, c\}$. We solve a two class classification problem, $\eta^k(\mathbf{x}) = \mathbb{P}(Y = 1|\mathbf{X} = \mathbf{x})$ is equal to learn $\eta_n^k(\mathbf{x})$ in the original multi-class classification problem. Then we only need to show that the sequence $\{g_n\}$ of BDT classifiers is consistent for the corresponding two-class problem. In addition, according to the work of Biau et al. (2008), for the two class random forests classifier, if the number of *ntree* is large, the random forests classifier takes a majority vote to obtain the classification result, which can be well approximated by the averaged classifier. As shown in Biau et al. (2008) that consistency of a random forests classifier is preserved by averaging, we have Lemma 2.

Lemma 2. *Assume that the sequence $\{g_n\}$ of tree classifiers is consistent for a certain distribution of $(\mathbf{X}, Y)$. Then the voting random forests classifier $g_n^{(ntree)}$ (for any value of ntree) and the averaged forests classifier $\overline{g}_n$ are also consistent.*

Proof. See that for Proposition 1 in Biau et al. (2008). □

According to Lemma 2, we can state the main theoretical result about BRFs. In addition, to prove the consistency of BRFs, it is sufficient to prove the consistency of the base tree classifier (BDT) g_n based on Lemma 2. Before giving our main theorem, we recall the construction method of the Banzhaf decision trees (BDTs) and give some mild assumptions as follows.

All nodes of the individual BDT can be seen as associated with rectangular cells, such that at each step of the construction of the individual BDT, the collection of cells associated with the external nodes of the BDT forms a partition of $d_n \in [0, 1]^h \subseteq D_n$. Then, the root node of BDT is $[0, 1]^h$ itself. In addition, the Banzhaf power index is used to evaluate the power of candidate features for each tree node. In fact, the Banzhaf power index of each candidate feature can be seen as a probability that each candidate feature may be chosen at each node, i.e. at each node, each candidate feature X_j is chosen according to the value of a Banzhaf power index $\zeta_{nj} \in (0, 1)$ (refer to Eq. (2)), in particular, $\sum_{j=1}^{h} \zeta_{nj} = 1$. Furthermore, because BDT chooses the midpoint of feature as the split point to expand tree at each node, then we assume that each BDT has $2^{\lceil \log_2 k_n \rceil} (\approx k_n)$ terminal (leaf) nodes, and let $K_{nj}(\mathbf{X}, \theta)$ denote the number of times the leaf node $A_n(\mathbf{X}, \theta)$ is split on the $j-th$ feature $(j = 1, \ldots, h)$. Then, conditioned on $\mathbf{X}$, $K_{nj}(\mathbf{X}, \theta)$ has a binomial distribution $\mathcal{B}(\lceil \log_2 k_n \rceil, \zeta_{nj})$. Inspired by the work of Biau et al. (Biau, 2012) and based on the above assumptions, our main theorem is given as follows.

Theorem 1. *Assume that the distribution of $\mathbf{X}$ has support on $[0, 1]^h \subseteq D_n$. Then the BRFs estimate $\overline{g}_n$ is consistent whenever $\log_2 k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$.*

Proof. By Lemma 1, the consistency of multi-class classifier can be transformed to the consistency of the corresponding two-class classifiers. In addition, for the two-class situation, Lemma 2 allows us to transform proving the consistency of BRFs into proving the consistency of base tree classifier g_n. Therefore, we only need to prove that the individual BDT classifier is consistent.

To prove the consistency of BDT, we recall a general consistency theorem for partitioning tree classifiers proved in Györfi et al. (1996, Theorem 6.1). According to this theorem, the BDT classifier g_n is consistent if both $diam(A_n(\mathbf{X}, \theta)) \to 0$ in probability and $N_n(\mathbf{X}, \theta) \to \infty$ in probability, where $A_n(\mathbf{X}, \theta)$ is the rectangular cell (node) of the tree partition containing $\mathbf{X}$ and

$$N_n(\mathbf{X}, \theta) = \sum_{i=1}^{n} \mathbb{I}_{\{\mathbf{X}_i \in A_n(\mathbf{X}, \theta)\}}$$

is the number of data point falling in the same cell as $\mathbf{X}$.

First, we prove that $N_n(\mathbf{X}, \theta) \to \infty$ in probability, where θ denotes the partition of BDT. Assume that the number of partition is $\lceil \log_2 k_n \rceil$, then a single BDT has exactly $2^{\lceil \log_2 k_n \rceil}$ cells (nodes), i.e. $A_1, \ldots, A_{2^{\lceil \log_2 k_n \rceil}}$. Let $N_1, \ldots, N_{2^{\lceil \log_2 k_n \rceil}}$ denote the number of data points among $\mathbf{X}, \mathbf{X}_1, \ldots, \mathbf{X}_n$ falling in these $2^{\lceil \log_2 k_n \rceil}$ cells. Since the data points $\mathbf{X}, \mathbf{X}_1, \ldots, \mathbf{X}_n$ are independent and identically distributed, fixing θ, the conditional probability of data point $\mathbf{X}$ falls in the ith cell is equal to $N_i/(n + 1)$. Thus, for every fixed $t \geq 0$,

$$\begin{aligned}
\mathbb{P}(N_n(\mathbf{X}, \theta) \leq t) &= \mathbb{E}[\mathbb{P}(N_n(\mathbf{X}, \theta) \leq t|\theta] \\
&= \mathbb{E}[\sum_{i=1,\ldots,2^{\lceil \log_2 k_n \rceil}: N_i < t} \frac{N_i}{n+1}] \\
&\leq \frac{t 2^{\lceil \log_2 k_n \rceil}}{n+1} \\
&\leq \frac{2tk_n}{n+1},
\end{aligned}$$

which converges to 0 by assumption on $k_n/n \to 0$.

Next, we show that $diam(A_n(\mathbf{X}, \theta)) \to 0$ in probability. To this aim, it is sufficient to show that the size of the each dimension feature of the rectangular cell containing $\mathbf{X}$ converges to 0. Let $V_{nj}(\mathbf{X}, \theta)$ denote the size of the $j - th$ $(j = 1, \ldots, h)$ dimension feature. Then, we only need to show that $V_{nj}(\mathbf{X}, \theta) \to 0$ in probability for all $j = 1, \ldots, h$. In addition, we have

$$V_{nj}(\mathbf{X}, \theta) = 2^{-K_{nj}(\mathbf{X}, \theta)},$$

where $K_{nj}(\mathbf{X}, \theta)$ denotes the number of times the cell containing $\mathbf{X}$ is split on the j-th coordinate, and conditionally on $\mathbf{X}$, $K_{nj}(\mathbf{X}, \theta)$ has a binomial $\mathcal{B}(\lceil \log_2 k_n \rceil, \zeta_{nj})$ distribution according to the construction of BDT. Therefore

$$\begin{aligned}
\mathbb{E}[V_{nj}(\mathbf{X}, \theta)] &= \mathbb{E}[2^{-K_{nj}(\mathbf{X}, \theta)}] \\
&= \mathbb{E}[\mathbb{E}[2^{-K_{nj}(\mathbf{X}, \theta)}|\mathbf{X}]] \\
&= (1 - \zeta_{nj}/2)^{\lceil \log_2 k_n \rceil},
\end{aligned}$$

which tends to 0 as $\log_2 k_n \to \infty$. □

By Lemmas 1, 2 and Theorem 1, the consistency of the multi-class classifier BRFs has been proved. Note that, in BRFs, we use the bagging method to generate many bootstrap samples from the original data set. Each of the BDTs is grown based on an independent bootstrap sample. According to the work of Biau et al. (Biau, 2012; Biau et al., 2008), the bagging classifiers $\overline{g}_n$ (random forests) are consistent, when the base tree classifier g_n is consistent (refer to Biau et al., 2008, Theorem 6). Therefore, in BRFs, using the bagging method does not affect the consistency of BRFs.

5.2. Discussion

According to the work of Biau et al. (2008), the original random forests classifier proposed by Breiman does not have consistency. To verify this fact, a two dimensional example was provided by Biau et al. (2008). Similarly, inspired by Biau et al., we give a two dimensional example to illustrate the inconsistency of the original random forests classifier and the consistency of the BRFs algorithm. That is, consider the joint distribution of $(\mathbf{X}, Y)$ sketched in Fig. 2, $\mathbf{X}$ has a uniform distribution on $[0, 1] \times [2, 3] \cup [1, 2] \times [1, 2] \cup [2, 3] \times [0, 1]$. Y is a function of $\mathbf{X}$, that is $\eta(\mathbf{x}) \in \{0, 1\}$ and $L^* = 0$. The upper left square $[0, 1] \times [2, 3]$ is divided into countably infinitely

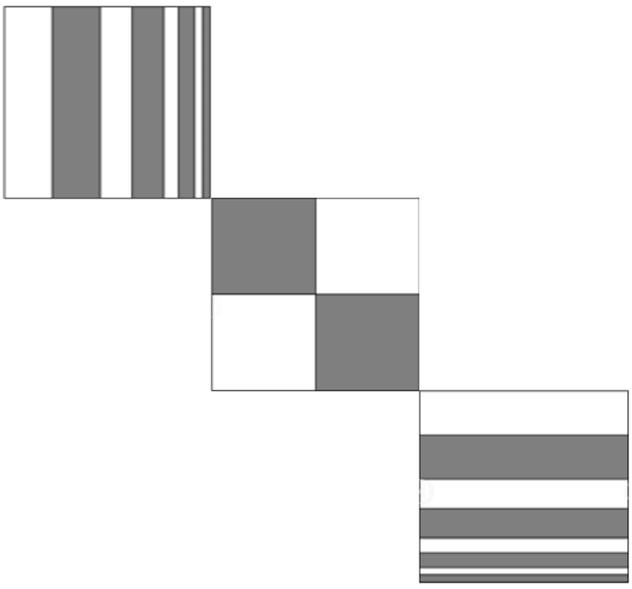

Fig. 2. An example of a distribution for which Breiman's random forests classifier is inconsistent.

many vertical stripes in which the stripes with $\eta(\mathbf{x}) = 0$ and $\eta(\mathbf{x}) = 1$ alternate. The lower right square $[2, 3] \times [0, 1]$ is divided similarly into horizontal stripes. The middle rectangle $[1, 2] \times [1, 2]$ is a 2×2 checkerboard. For simplicity, we consider the original random forests classifier when each node is split by minimizing the empirical probability error instead of maximizing the information gain ratio to achieve the growth of each tree. In addition, each tree is grown until every tree node cell contains just one data point. Then, it is not hard to find that no matter what the sequence of random selection of split directions is and no matter how long each tree is grown, no tree will ever cut the middle rectangle, and therefore the probability of error of the original random forests classifier is at least 1/6. However, in BRFs, the midpoint of the features is selected as the split point at each node for each tree. Thus, in any case, it is clear that BRFs can definitely cut the middle rectangle.

6. Experiments

In order to evaluate the effectiveness and robustness of the proposed BRFs algorithm, we conducted extensive experiments on 12 data sets from the UCI machine learning repository and three real world applications, including handwritten digits recognition (usps), face recognition (Yale) and text classification (newsgroups). Furthermore, to demonstrate the effect of the Banzhaf power index in construction of random forests, we compared it with the information gain ratio. The basic information of the used data sets was shown in Table 1.

6.1. Classification performance of BRFs

To assess the classification accuracy of BRFs, we compared BRFs to the existing random forests algorithms and the well-known classifiers—support vector machines (SVMs) (Chang & Lin, 2007) and k-nearest neighbors (KNNs), where SVMs with radial basis function (RBF) kernel was used in our experiments. In particular, Breiman's random forests (RFs) (Breiman, 2001) and the consistent random classification forests (Biau12) (Biau, 2012) were used in the experiments to compare with BRFs.

Table 1
Summary of the used UCI data sets.

Data sets	No.examples	No.features	No.classes
ionosphere	35	34	2
wine	178	13	3
sonar	208	20	2
housing	506	13	2
dermatology	366	34	6
pima	768	8	2
vehicle	846	18	4
waveform40	5000	40	3
newsgroups	16242	100	4
satimage	6435	36	6
musk2	6598	166	2
shuttle	14516	9	7
usps	9298	256	10
Yale	165	1024	15
isolet	6238	617	26

For RFs and Biau12, we employed the information gain ratio as the splitting criterion to grow each binary tree of the forests. Although the Gini index criterion was also implemented in RFs (Breiman, 2001), the information gain ratio based split criterion occupied the dominant position in the construction of random forests. Thus, in this paper, we do not consider to use the Gini index as split criterion. For Biau12, we first evaluated the importance of each candidate feature by using the information gain ratio to compute the candidate split point (i.e. the midpoint value of the each feature) based on the training data set, then the tree nodes in Biau12 were expanded by selecting a fixed number of random candidate features (without replacement). If the selected features were all weak ones, then chose one at random and splitted at the midpoint value. If more than one strong features were selected, chose one at random and cutted at the midpoint value. For all the random forests algorithms, we empirically set $ntree = 100$. At the same time, to construct the decision trees, $h = round(log_2(M) + C)$ features were randomly selected, where M was the dimensionality of the data samples and $C \in \mathbf{R}$ was a parameter. For all the forests, SVMs and KNNs, 5-fold cross-validation was applied to select the parameters. For all the data sets, the sample features were scaled to [0, 1].

In our experiments, all of the classification results were obtained by averaging over 5-fold cross-validation except for the isolet data set. For the isolet data set, we simply followed the training and test partition given a priori. Table 2 showed the results obtained by SVMs, KNNs, RFs, Biau12 and BRFs. The best classification accuracy was shown in boldface.

Following the suggestions of Demsar (2006), we implemented the Friedman and Nemenyi statistical test at 95% confidence level to show the performance difference between BRFs and the compared algorithms. The computed meanrank (the less is it, the better is the corresponding algorithm) is shown in the last row of Table 2. The critical difference was computed as $CD = 1.5750$. It was easy to see that BRFs performed significantly better than Biau12, and at least comparable with RFs, KNNs and SVMs.

The reason for BRFs performed comparable with RFs was that BRFs employed a simple way to choose the midpoint of the "best" feature as the split point, while RFs took a traverse way to compute each possible split point for each feature. From the view point of structure, our BRFs algorithm was much rougher than RFs. However, choosing the midpoint of the "best" feature as the split point, which can guarantee the consistency of BRFs algorithm. Note that, RFs algorithm does not have consistency. In addition, Biau12 also employed a simple way to choose the midpoint of the "best" feature as the split point, but the performance of Biau12 was significantly worse than that of BRFs. The reason for this result was that BRFs employed Banzhaf power index to evaluate the importance of candidate feature variable at each tree node,

Table 2
Mean classification accuracy and standard deviation obtained by the compared SVMs, KNNs and random forests algorithms. Algorithms with the best accuracy are shown in boldface.

Data sets	SVMs	KNNs	RFs	Biau12	BRFs
ionosphere	**0.9401 ± 0.0370**	0.8375 ± 0.0562	0.9315 ± 0.0384	0.8972 ± 0.0469	0.9315 ± 0.0530
wine	0.8964 ± 0.0477	0.9423 ± 0.0474	0.9658 ± 0.0251	0.8715 ± 0.0909	**0.9706 ± 0.0509**
sonar	0.5687 ± 0.1192	0.5908 ± 0.1847	0.6840 ± 0.0914	0.5933 ± 0.0723	**0.7088 ± 0.1361**
housing	0.7605 ± 0.1151	**0.8132 ± 0.1230**	0.6418 ± 0.0645	0.6715 ± 0.1052	0.7964 ± 0.1230
dermatology	0.9540 ± 0.0130	0.9656 ± 0.0246	0.9530 ± 0.0167	0.8777 ± 0.0643	**0.9730 ± 0.0128**
pima	0.7605 ± 0.0132	0.7593 ± 0.0218	0.7461 ± 0.0482	0.6382 ± 0.0440	**0.7617 ± 0.0250**
vehicle	0.6728 ± 0.0470	0.7359 ± 0.0220	0.6490 ± 0.0076	0.6532 ± 0.0340	**0.7513 ± 0.0420**
waveform40	**0.8652 ± 0.0086**	0.7724 ± 0.0148	0.7490 ± 0.0076	0.7012 ± 0.1121	0.7790 ± 0.0082
newsgroups	**0.7872 ± 0.0580**	0.7386 ± 0.0554	0.7729 ± 0.0579	0.5776 ± 0.0323	0.6973 ± 0.0144
satimage	0.8645 ± 0.0123	0.8850 ± 0.0091	**0.8970 ± 0.0110**	0.7048 ± 0.0478	0.8735 ± 0.0206
musk2	0.8508 ± 0.0747	0.7227 ± 0.0636	0.8546 ± 0.1204	0.6202 ± 0.0292	**0.8960 ± 0.0503**
shuttle	0.9752 ± 0.0042	0.9951 ± 0.0035	**0.9983 ± 0.0011**	0.8468 ± 0.0137	0.9968 ± 0.0013
usps	0.9251 ± 0.0132	**0.9450 ± 0.0125**	0.9041 ± 0.0183	0.8968 ± 0.0126	0.9032 ± 0.0115
Yale	**0.7400 ± 0.1362**	0.6911 ± 0.1811	0.5156 ± 0.0183	0.5067 ± 0.0147	0.4986 ± 0.0182
isolet	**0.9628 ± 0.0000**	0.9256 ± 0.0000	0.9529 ± 0.0000	0.9415 ± 0.0000	**0.9628 ± 0.0000**
meanrank	2.4333	2.9333	2.9000	4.4667	**2.2667**

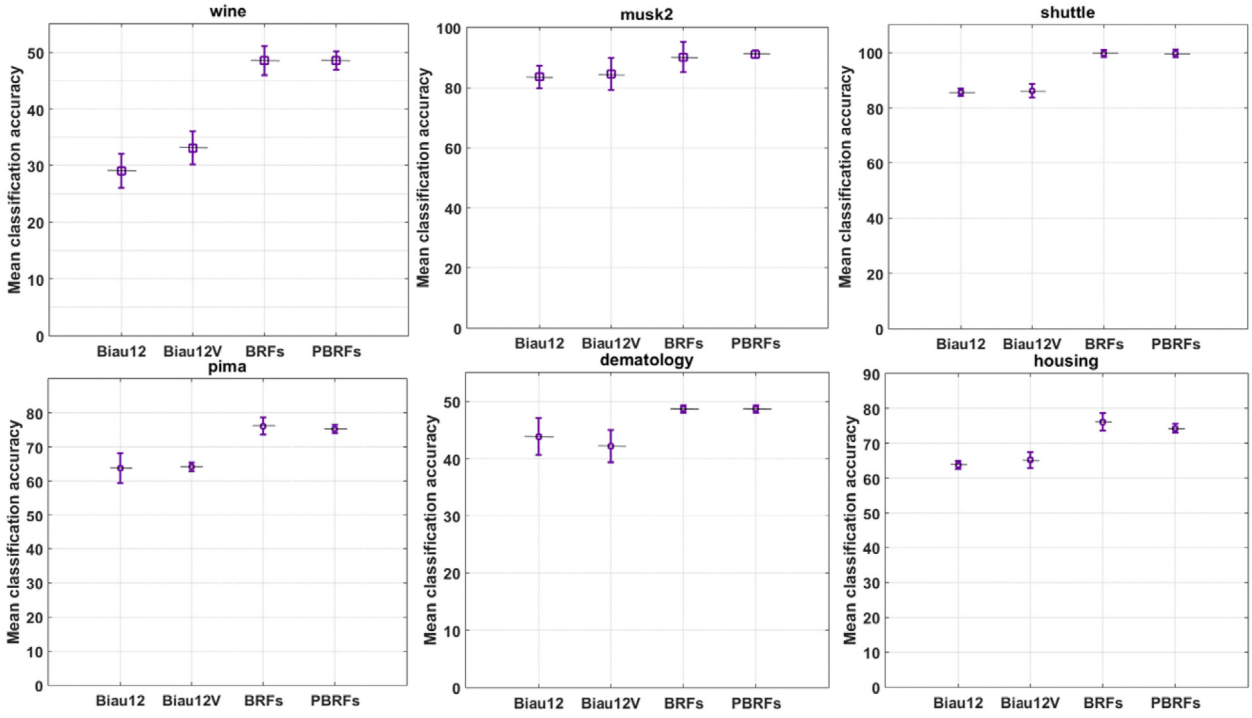

Fig. 3. Classification accuracy for different random forests algorithms on several data sets. In these charts, the *y*-axis shows the classification accuracy and the *x*-axis indicates different algorithms.

which could find some features with good discriminative ability as a group for the construction of a single decision trees. Biau12 first used the information gain ratio to evaluate the importance of each feature variable in the training data set. The information gain ratio often pays less attention to the intrinsic structure of feature variables, then any combination of predicting feature variables which presents a much stronger prediction may therefore be missed. In the following subsection, we will give a more detailed experimental analysis to show the difference between the Banzhaf power index and the information gain ratio.

6.2. Banzhaf power index vs. information gain ratio

To further illustrate the effectiveness of the Banzhaf power index in improving the performance of random forests algorithm, comparison experiments for Biau12, Biau12V (a variant of Biau12), BRFs and PBRFs (a variant of BRFs) were performed on several data sets from the UCI machine learning repository. In detail, Biau12V used the Banzhaf power index to evaluate the importance of the sample features instead of the information gain ratio in the original Biau12 algorithm, while PBRFs employed the Banzhaf power index to select the feature of the root node instead of the information gain ratio in the original BRFs algorithm.

For comparison, 5-fold cross-validation was applied to select the parameter, i.e., the number of candidate features for each node. The classification performances of BRFs, PBRFs, Biau12 and Biau12V are shown in Fig. 3. It is easy to see that the performance of BRFs was comparable to that of PBRFs, while both of them outperformed Biau12 and Biau12V. In addition, Biau12V generally performed better than Biau12. These results indicated that in a practical sense it was the feature evaluation strategy that accounted for most of the improvement of BRFs and Biau12V over Biau12. Furthermore, we can see that the performances of BRFs and PBRFs were certainly competitive, but we found that the computation speed of PBRFs was generally slower than that of BRFs. In view of this, we recommend the BRFs algorithm to

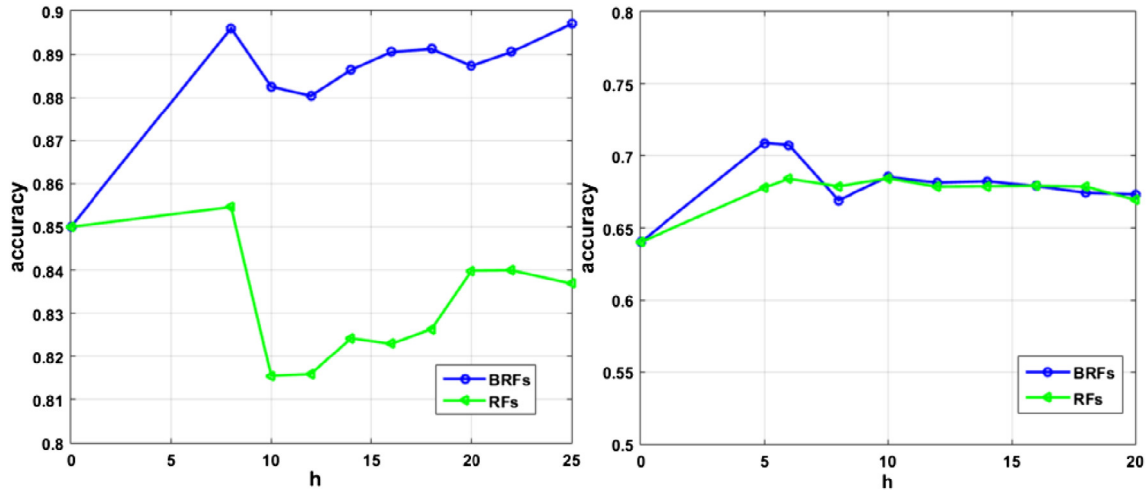

Fig. 4. The performance of BRFs and RFs with different "h" parameters for the "musk2" and "sonar" data sets. In the two charts, the y-axis shows the accuracy and x-axis shows the different "h" values.

the practical applications. According to the experimental results, it is sufficient to show that the effectiveness of the Banzhaf power index in improving the performance of random forests algorithm, in particular, for the consistent random forests algorithm.

6.3. Robustness analysis

In order to investigate the robustness of BRFs, the robustness analysis experiments were performed. Fig. 4 shows the performance of BRFs and Breiman's Random forests (RFs) (Breiman, 2001) with different parameter h which controls the size of feature subsets for each tree on two data sets (musk2 and sonar). Fig. 5 shows how the number of trees *ntree* affects the performance of BRFs on the pima, wine, sonar and ionosphere data sets. The curves of the other data sets told a similar story.

From Fig. 4, we can see after h approached to 10 all the random forests algorithms lead to satisfactory results for both the "musk2" and "sonar" data sets, which demonstrated that the BRFs algorithm was fairly robust to the parameter h (though the results tend to have small fluctuation).

The obtained classification accuracy results vs. the number of trees in BRFs was shown in Fig. 5. We can see that BRFs are basically robust with the number of trees. With the increasing of the number of trees, the classification accuracy increases gradually. When the number of trees is within [100, 1000], are quite robust. Hence, for simplicity, we chose $ntree = 100$ in our experiments. Moreover, Fig. 5 shows that BRFs will not incur over-fitting.

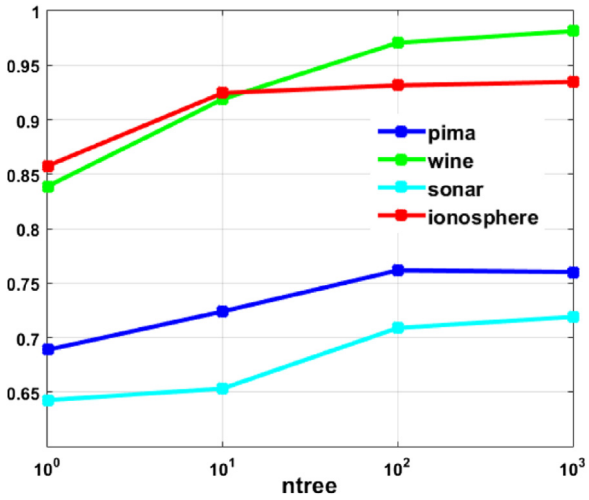

Fig. 5. The performance of BRFs is influenced by the number of trees *ntree* on four data sets; in this plot, the y-axis shows the classification accuracy and the number of trees are shown along the x-axis.

7. Conclusion and future work

Random forests play an important role in areas related to machine learning. At present, there are many random forests algorithms. Among these forests models, we rarely find one that has both significant practical performance and a complete theoretical guarantee. The structure of the original random forests is mainly based on information theory. However, information theory pays less attention to the intrinsic structure of candidate feature variables. Alternatively, the Banzhaf power index can capture this structure information between feature variables. Motivated by this fact, in this paper, we propose a novel random forests model called Banzhaf random forests (BRFs) and give the proof of its consistency. We have tested BRFs on several UCI data sets and some real world applications. The experimental results demonstrate that BRFs perform significantly better than existing consistent random forests, and better than or at least comparable with the original random forests, support vector machines (SVMs) and k-nearest neighbors (KNNs).

Our work is an innovation utilizing the combination of cooperative game theory and the random forests algorithm in machine learning. In the future, we will try to combine the game theory and the existing neural networks based on the research results discussed in this paper, so that the neural networks can explore better features for target tasks. In fact, there has already been some studies that combined Game Theory and the neural networks (Fung & Liu, 2003; Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, & Bengio, 2014; He, Yu, Huang, Li, & Li, 2014). Typically, Generative Adversarial Networks (GANs) are built upon the two person zero-sum game (two-player game) in the Game Theory. In addition, the efficiency of existing network models can be improved by combining the proposed random forest algorithm. Moreover, we will try to combine other solution concepts of cooperative game theory (i.e. the Shapley value and the nucleolus) with random forest algorithms or existing neural networks to develop new learning models.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No. 2016YFC1401004, the National Natural Science Foundation Of China (NSFC) under Grant No. 61473236 and 61403353, the International Science & Technology Cooperation Program of China (ISTCP) under Grant No. 2014DFA10410, the Science and Technology Program of Qingdao under Grant No. 17-3-3-20-nsh, the Suzhou Science and Technology Program under grant no. SYG201712 and SZS201613, the CERNET Innovation Project under Grant No. NGII20170416, the Key Program Special Fund in XJTLU under Grant No. KSF-A-01, and the Fundamental Research Funds for the Central Universities of China.

References

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, *9*(7), 1545–1588.

Amozegar, M., & Khorasani, K. (2016). An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines. *Neural Networks*, *76*, 106–121.

Ayerdi, B., & Grana, M. (2014). Hybrid extreme rotation forest. *Neural Networks*, *52*, 33–42.

Banzhaf III, J. F. (1964). Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, *19*, 317–343.

Biau, G. (2012). Analysis of a random forests model. *Journal of Machine Learning Research (JMLR)*, *13*(1), 1063–1095.

Biau, G., Devroye, L., & Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research (JMLR)*, *9*, 2015–2033.

Bosch, A., Zisserman, A., & Muoz, X. (2007). Image classification using random forests and ferns. In *IEEE conference on computer vision* (pp. 1–8). IEEE.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2004). Consistency for a simple model of random forests. Technical report 670.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press. 40(3), 582–588.

Bulo, S. R., & Kontschieder, P. (2014). Neural decision forests for semantic image labelling. In *Computer vision and pattern recognition* (pp. 81–88).

Chalkiadakis, G., Elkind, E., & Wooldridge, M. (2011). Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *5*(6), 1–168.

Chang, C. C., & Lin, C. J. (2007). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 389–396.

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). The MIT Press.

Criminisi, A., & Shotton, J. (2013). *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media. 273–293.

Criminisi, A., Shotton, J., & Konukoglu, E. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, *7*, 81–227.

Cutler, D. R., Edwards Jr., T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, *88*(11), 2783–2792.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research (JMLR)*, *7*(1), 1–30.

Denil, M., Matheson, D., & De Freitas, N. (2013a). Narrowing the gap: random forests in theory and in practice. In *International conference on machine learning* (pp. 665–673).

Denil, M., Matheson, D., & de Freitas, N. (2013b). Consistency of online random forests. In *International conference on machine learning* (pp. 1256–1264).

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, *40*(2), 139–157.

Dollar, P., & Zitnick, C. L. (2014). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(1), 1558–1570.

Feltkamp, V. (1995). Alternative axiomatic characterizations of the Shapley and Banzhaf values. *International Journal of Game Theory*, *24*(2), 179–186.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Fung, W. K., & Liu, Y. H. (2003). Adaptive categorization of ART networks in robot behavior learning using game-theoretic formulation. *Neural Networks*, *16*(10), 1403.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. *Advances in Neural Information Processing Systems*, *3*, 2672–2680.

Györfi, L., Devroye, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer-Verlag. 63(279): 194–199.

Hallman, S., & Fowlkes, C. C. (2015). Oriented edge forests for boundary detection. In *IEEE conference on computer vision and pattern recognition* (pp. 1732–1740).

He, X., Yu, J., Huang, T., Li, C., & Li, C. (2014). Neural network for solving nash equilibrium problem in application of multiuser power control. *Neural Networks*, *57*(9), 73–78.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Ishwaran, H., & Kogalur, U. B. (2010). Consistency of random survival forests. *Statistics & Probability Letters*, *80*(13), 1056–1064.

Kim, J., Jang, G. J., & Lee, M. (2016). Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. *Neural Networks*, *87*, 109–121.

Mcquoid, M. R. J. (1993). Neural ensembles: Simultaneous recognition of multiple 2-D visual objects. *Neural Networks*, *6*(7), 907–917.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research (JMLR)*, *7*, 983–999.

Pavel, M. S., Schulz, H., & Behnke, S. (2017). Object class segmentation of RGB-D video using recurrent convolutional neural networks. *Neural Networks*, *88*, 105–113.

Prasad, A. M., Iverson, L. R., & Liaw, A. (2006). Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, *9*(2), 181–199.

Ristin, M., Guillaumin, M., Gall, J., & Van Gool, L. (2016). Incremental learning of random forests for large-scale image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(3), 490–503.

Roy, A., & Todorovic, S. (2016). Monocular depth estimation using neural regression forest. In *Computer vision and pattern recognition* (pp. 5506–5514).

Saad, W., Han, Z., Debbah, M., & Hjorungnes, A. (2009). Coalitional game theory for communication networks. *Signal Processing Magazine IEEE*, *26*(5), 77–97.

Salzberg, S. L. (1994). C4.5: programs for machine learning by J. Ross Quinlan. Morgan kaufmann Publishers, Inc., 1993. *Machine Learning*, *16*(3), 235–240.

Scardapane, S., & Di, L. P. (2017). A framework for parallel and distributed training of neural networks. *Neural Networks*, *91*, 42–54.

Shotton, J., Johnson, M., & Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., & Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, *56*(1), 116–124.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of Chemical Information and Computer Sciences*, *43*(6), 1947–1958.

Yin, P., Criminisi, A., Winn, J., & Essa, M. S. (2007). Tree-based classifiers for bilayer video segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 18–23). IEEE.

Zhang, L., & Suganthan, P. N. (2014). Random forests with ensemble of feature spaces. *Pattern Recognition*, *47*(10), 3429–3437.

Zikic, D., Glocker, B., & Criminisi, A. (2013). Atlas encoding by randomized forests for efficient label propagation. In *Medical image computing and computer-assisted intervention* (pp. 66–73). Springer.