An Improved Random Forest Algorithm

Ying Yu, Lingyun Wang, Hao Huang and Wenwen Yang

Department of Mechanical and Electrical Engineering and Automation,
University of Shanghai, room 808, No.333 Nan Chen Road, Baoshan District, Shanghai, China
Email: squarey@shu.edu.cn

Abstract. A random forest algorithm with an improved decision tree and an optimal tree integration is proposed. First, an improved ID3 decision tree is proposed. Next, when the trees are integrated, each decision tree is assigned a voting right consistent with the out-of-bag accuracy of the decision tree. Finally, the experimental results show that the improved random forest algorithm is efficient and the classification performance is improved compared with the traditional random forest algorithm.

1. Introduction
Data mining extracts potentially valuable knowledge via data mining methods. Random forest algorithm is a common method in data mining analysis. Leo Breiman proposed the random forest algorithm for the first time in 2001[1]. Robnik-Sikonja et al. improved the voting method of traditional random forest in the process of integrating decision trees[2]. Gall et al. introduced the Hough transform into the random forest algorithm and obtained the Hough forest algorithm[3]. Ishwaran et al. proposed a randomly generated forest algorithm[4].

In this paper, a random forest algorithm combining improved decision tree and optimized integration is proposed, and the efficiency of the algorithm is verified by using the common data set.

2. Structure of the Improved Random Forest Algorithm

2.1. Structure of the Improved Decision Tree
Different from the traditional ID3 algorithm, with two thresholds(including a confidence level threshold and a residual data threshold), the improved ID3 algorithm can prune and develop into leaves before exhausting all the training data. In addition, the improved ID3 algorithm can also control the size of the decision tree generation by adjusting the confidence level threshold and the residual data threshold. In this way, the possibility of over-learning could be reduced greatly.

The construction module of the improved ID3 decision tree is shown below:
2.2. Process of the Improved Decision Tree

The process of the recursion function $\text{ModifyCreatModelID3}(A, C, R, \beta, \theta)$ is as follows:

- **Input:**
  - $A$: the current set of properties to be selected is the set of all properties which used to build the decision tree, initial $A = \{A_1, A_2, \ldots, A_l\}$, $l$ represents the number of attribute in the set $A$, where $A_i = \{a_{i1}, a_{i2}, \ldots, a_{ik}\}$, where $a_{ik}$ represents the $k$th attribute value of $A_i$;
  - $C$: represents the set of decision values $C = \{c_1, c_2, \ldots, c_m\}$, $m$ represents the number of decision attributes;
  - $R$: a set of records for classification;
  - $\beta$: Confidence level threshold, $\beta \in (0, 1]$;
  - $\theta$: Residual data threshold, $\theta \in (0, 1]$;
  - $S$: represents the sample size of the data record set.

- **Output:**
  - an improved ID3 decision tree with node as its root

**Symbols in the algorithm:**
- $\text{RecA}(i, a_{ij}, R)$: a record set in $R$, where the $j$th attribute value in $A_i = a_{ij}$;
- $\text{RecC}(c_k, R)$: a record set in $R$, where the decision value $= c_k$. 

---

**Figure 1.** The construction module of the improved ID3 decision tree.
1. node ← ∅  
2. flag ← $\text{newIsLeaf}(A,R,β,θ,\text{node})$ (To check if it is identified as a leaf node)  
3. if (flag=true) (To check if it is a leaf node)  
   4. node.Leafnode ← true  
   5. node ← creatLeafnode(node, $C,R,θ$) (Generate a leaf node)  
   6. return node  
   7. else  
      8. node.Leafnode ← false  
   9. end if  
10. maxGain ← 0 (maxGain represents the maximum information gain)  
11. $l_{max} ← 0$ (is the ordinal number of the attribute with the maximum information gain)  
12. for $i = 1$ to $l$  
13.   if (Gain $(A_i) > \text{maxGain}$)  
14.      maxGain ← Gain $(A_i)$  
15.      $l_{max} ← i$  
16.  end if  
17. end for  
18. node.name ← $A_{l_{max}}$ (node.name is the name of the current node)  
19. $A' ← A - \{A_{l_{max}}\}$ (Remove $A_{l_{max}}$ from the set of attributes to be selected)  
20. $A ← A'$  
21. Define the node.attribute[] array, length of $|A_{l_{max}}|$ (node.attribute[] is the branch name reserved for the current node)  
22. Define the node.attribute[] array, length of $|A_{l_{max}}|$ (node.childnode[] represents the collection of children for the current node)  
23. $k ← l_{max}$  
24. for $i = 1$ to $|A_{l_{max}}|$  
25.   node.attribute[$i$] ← $a_{ki}$  
26. $R' ← \text{RecA}(k,a_{ki},R)$  
27. $R ← R - R'$  
28. node.childnode[$i$] = ModifyCreatModelID3$(A,C,R',β,θ)$  
29. end for  
return node

In the process, function newIsLeaf() is to check whether the inputted node is a leaf, and function creatLeafnode() is to create a new leaf node.

### 3. Numerical Examples

#### 3.1. Data Set
5 groups of data sets including Blood, Haberman, Iris, Seeds and Wine are selected from the UCI data set[5].

About 75% of the data in the data set were used as the training set, and the remaining 25% were used as the testing set.
3.2. Performance Evaluation Indexes of the Classification Model
The confusion matrix of binary classification data, which includes TP, TN, FP, FN, is used to evaluate the classification performance. Moreover, Accuracy, Precision, Recall and $F_{score}$ are used as the performance evaluation index. For the multiple classification problem, only Accuracy is used as the performance evaluation index in this paper.

3.3. Results and Discussion
Set the parameters for the Iris data set as follows: the decision tree number $\text{TreeNumber} = 100$, the residual data threshold $\theta = 0$, threshold for tree integration $= 0.9$. The classification accuracy of the proposed algorithm with varying confidence level threshold for the Iris data set is shown in Figure 2:

![Figure 2. Classification accuracy of the proposed algorithm with varying confidence level (For Iris data set).](image)

As can be seen from Figure 2, when the confidence level threshold is 0.9 or 1.0, the classification accuracy is the highest.

In the Iris data set, set the number of decision tree $\text{TreeNumber} = 100$, the confidence level threshold $\theta = 0.9$, threshold for tree integration $= 0.9$. The classification accuracy of the proposed algorithm with varying residual data threshold is shown in Figure 3:

![Figure 3. Classification accuracy of the proposed algorithm with varying residual data threshold (For Iris data set).](image)
According to Figure 3, when the residual threshold is 0 or 5, the proposed algorithm has the highest classification accuracy.

Classification accuracy comparison between the common random forest algorithm (RF) and proposed improved random forest algorithm for different data sets are given in Table 1. The tree number of $\text{RF} = 100$. Parameters of the proposed algorithm are as follows: $\text{TreeNumber} = 100$, the confidence level threshold $= 0.9$, threshold for tree integration $= 0.9$, the residual data threshold $\theta = 5$.

| Data set | RF       | The proposed algorithm |
|----------|----------|------------------------|
| Blood    | 75.94%   | 76.38%                 |
| Haberman | 72.37%   | 73.90%                 |
| Iris     | 89.19%   | 94.59%                 |
| Seeds    | 71.15%   | 75.64%                 |
| Wine     | 86.36%   | 92.05%                 |

More comparative results are shown in Table 2 and Table 3 for data set Blood and Haberman respectively.

| Performance indicators | Algorithm      | RF       | the proposed algorithm |
|------------------------|----------------|----------|------------------------|
| Accuracy               | 75.94%         | 76.38%   |
| Precision              | 98.59%         | 100%     |
| Recall                 | 76.50%         | 76.34%   |
| $F_{score}$            | 86.15%         | 86.59%   |

| Performance indicators | Algorithm      | RF       | the proposed algorithm |
|------------------------|----------------|----------|------------------------|
| Accuracy               | 72.37%         | 73.90%   |
| Precision              | 98.21%         | 85.71%   |
| Recall                 | 73.33%         | 80.00%   |
| $F_{score}$            | 83.97%         | 82.76%   |

From the above tables, we could see that in most cases, the proposed algorithm achieve a superior performance compared with $\text{RF}$. 
4. Summary
An improved random forest algorithm is proposed in this paper, where an improved ID3 algorithm with pruning effect is introduced. The algorithm can not only control the complex size of the decision tree, but also avoid over-fitting, thus improve the classification effect of the algorithm.

The comparative results show that the improved random forest algorithm is superior than RF.

5. References
[1] Breiman L. Random Forests [J]. Machine Learning, Vol.45, No.1, 2001, pp. 5-32.
[2] Robnik-Sikonja M. Improving Random Forests [J]. Vol. 3201, 2004, pp. 359-370.
[3] Gall Juergen, Yao Angela, Razavi Nima, Van Gool Luc, Lempitsky Victor. Hough forests for object detection, tracking, and action recognition [J]. IEEE transactions on pattern analysis and machine intelligence, Vol. 33, No.11, 2011, pp. 188-202.
[4] Ishwaran H, Kogalur U B, Chen X, et al. Random survival forests for high-dimensional data [J]. Statistical Analysis & Data Mining, Vol.4, No.1, 2011, pp. 115-132.
[5] http://archive.ics.uci.edu/ml/datasets.php