Pattern Recognition 48 (2015) 976–983

Contents lists available at ScienceDirect

Pattern Recognition

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

Disjunctive normal random forests

Mojtaba Seyedhosseini [a,b,*], Tolga Tasdizen [a,b]

[a] Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT 84112, USA
[b] Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, USA

ARTICLE INFO

Article history:
Received 17 January 2014
Received in revised form
1 July 2014
Accepted 23 August 2014
Available online 4 September 2014

Keywords:
Random forest
Decision tree
Classifier
Supervised learning
Disjunctive normal form

ABSTRACT

We develop a novel supervised learning/classification method, called *disjunctive normal random forest* (DNRF). A DNRF is an ensemble of randomly trained *disjunctive normal decision trees* (DNDT). To construct a DNDT, we formulate each decision tree in the random forest as a disjunction of rules, which are conjunctions of Boolean functions. We then approximate this disjunction of conjunctions with a differentiable function and approach the learning process as a risk minimization problem that incorporates the classification error into a single global objective function. The minimization problem is solved using gradient descent. DNRFs are able to learn complex decision boundaries and achieve low generalization error. We present experimental results demonstrating the improved performance of DNDTs and DNRFs over conventional decision trees and random forests. We also show the superior performance of DNRFs over state-of-the-art classification methods on benchmark datasets.

1. Introduction

Random forests became popular with Breiman's seminal paper [1] in 2001 due to their ease of use and good classification accuracy. The main idea of random forest classification is to grow an ensemble of decision trees such that the correlation between the trees remains as low as possible. This is achieved by injecting randomness into the forest using a different set of training samples for each tree. These sets are obtained by sampling the original training set with replacement, *i.e.*, bagging. Another source of randomness in random forests is the subset of features randomly selected to consider at each node as the splitting function. This parameter can directly control the correlation between the trees and also affect the accuracy performance of each individual tree. At test time, each tree in the random forest casts a unique vote for the given input and the most popular class among the trees is selected as the predicted label for that input. Random forests have been shown to be effective in many applications like image segmentation/classification [2,3], object detection [4], and biomedical image analysis [5,6].

Random forests have certain advantages over other widely used classification algorithms. For instance, support vector machines (SVMs) [7] offer good generalization performance due to the fact that they guarantee maximum margin, but choosing the kernel function and the kernel parameters can be time consuming. Boosting [8] is another popular classification approach, which trains a single strong classifier by combining multiple weak classifiers. However, convergence of the learning algorithm can be slow for problems with complex decision boundaries. Artificial neural networks (ANNs) [9] are powerful but slow at training due to the computational cost of backpropagation [10]. In addition to all the aforementioned shortcomings of ANNs, SVMs, and boosting methods, these techniques do not naturally handle multi-class problems [11–13]. On the other hand, random forests are fast to train and handle multi-class problems intrinsically [14]. Moreover, they perform consistently well for high dimensional problems [15].

The weak learner used at each node of the decision trees plays an important role in the behavior and performance of random forests. The conventional random forest exploits axis-aligned decision stumps, which partition the feature space with orthogonal hyperplanes. While this type of partitioning can be suitable for certain types of datasets, it results in overfitting and produces "blocky artifacts" in general datasets [14]. It has been shown that using linear discriminants that can be at any arbitrary orientation to the axes improves the performance of random forests [16]. Nonlinear weak learners like conic sections have also been proved successful in increasing the accuracy and generalization performance of random forests [14].

A lot of work has been put into improving the random forest, through the use of more powerful node models and less correlated trees. Rodriguez et al. [17] used PCA to make a linear combination of features at each node. Bernard et al. [18] focused on the number of features randomly selected at each node of the tree. They showed that

* Corresponding author at: Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, USA. Tel.: +1 801 585 1867; fax: +1 801 585 6513.
E-mail address: mseyed@sci.utah.edu (M. Seyedhosseini).

http://dx.doi.org/10.1016/j.patcog.2014.08.023
0031-3203/

using a random number of features, which can be different at each node, can improve the performance. Tripoliti et al. [19] improved the prediction peformance of random forests by modifying the node split function as well as the voting procedure in the forest.

In this paper, we propose a novel approach for learning linear discriminants of arbitrary orientation at each node of a decision tree. However, the main advantage of our approach over the above-mentioned methods such as [16,17] is that it learns all the weak learners of the decision tree in a unified framework. To be clear, unlike conventional decision trees and their variants that learn the splitting function at each node independently, our approach allows weak learners of different nodes to interact with each other during the training because it minimizes a single global objective function. To achieve this goal, we formulate each decision tree as a single disjunction of conjunctions [20] and approximate it with a differentiable function. Next, we use this approximation in a quadratic error cost to construct a single unified objective function. Finally, we minimize this objective function using the gradient descent rule to update the parameters of the discriminants in the decision tree. We call this type of decision tree a disjunctive normal decision tree (DNDT).

Many researchers have proposed converting decision trees into a differentiable form and performing some global parameter tuning to make a smooth decision boundary with high generalization performance. For example [21–24] propose to convert decision trees into artificial neural networks (ANN) and use back-propagation to fine tune the weights and improve the performance. These methods speed up the training of ANNs by using decision trees to initialize the weights of ANNs. However, it would be hard to generalize these methods to random forest framework due to the slowness of back-propagation. Our approach is different from these methods in the sense that unlike the neural networks that have at least two layers of adaptive weights, our disjunctive normal form has only one adaptive layer and thus is faster than back-propagation. Moreover, we will show that DNDTs outperform ANNs.

Fuzzy/soft decision trees are another technique that have been developed to improve the performance of decision trees. Olaru and Wehenkel [25] build a decision tree by introducing a third state at each node. The samples which fall in the third state go to both children nodes. Using this strategy, a sample might contribute to the final decision through multiple paths. Irsoy et al. [26] also propose a soft decision tree that uses a gate function to redirect each sample to all the children with a certain probability. This strategy results in more accurate and simpler trees. The fundamental difference of our approach with soft decision trees is that we propose a global objective function and learn all the splits simultaneously. We will show that DNDTs outperform soft decision trees.

We follow the idea of random forests and use DNDTs as building blocks of a new random forest, called a disjunctive normal random forest (DNRF). While DNRFs have all the advantages of conventional random forests, they outperform them due to their stronger building blocks, *i.e.*, DNDTs. Fig. 1 demonstrates the superior performance of DNRF over conventional random forest with artificial examples. We observe that conventional random forest results in box-like decision boundaries and overfits to the training data while DNRF produces a smooth boundary with lower generalization error. In the results section, we show that, similar to random forests, DNRFs are able to handle multi-class

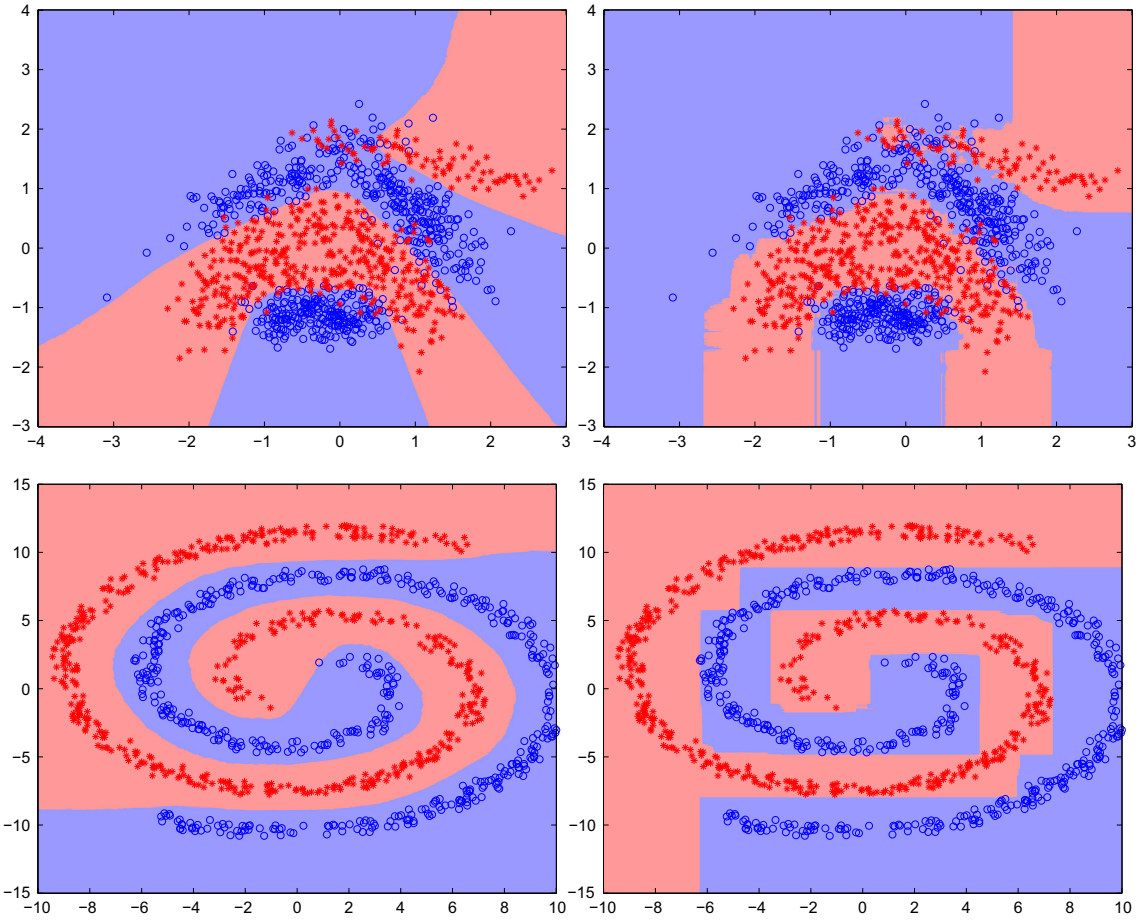

Fig. 1. Comparison of DNRF (left panel) with random forest (right panel) on the banana dataset [29] (upper panel) and two-spiral dataset (lower panel). DNRF results in a smoother decision boundary and, unlike random forests, does not overtrain.