

Imbalanced Remote Sensing Ship Image Classification

Sizhe Huang^{a,b,*}, Huosheng Xu^b, and Xuezhi Xia^b

^aCollege of Information Technology, Harbin Engineering University, Harbin, 150000, China

^bWuhan Digital Engineering Research Institute, Wuhan, 430000, China

Abstract

Aiming at the unbalanced classification problem of remote sensing ship image datasets in ship target classification and the problem that the traditional decision tree classification algorithm needs to rely on artificial construction features to realize classification, a weighted deep neural decision forest is proposed. This method combines deep learning with resampling. The results show that the method can achieve a better classification accuracy than the traditional decision tree on unbalanced classification of ship target.

Keywords: imbalanced ship classification; deep learning; decision tree

(Submitted on December 15, 2018; Revised on January 16, 2019; Accepted on February 11, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

As the key target of monitoring at sea and attacking in wartime, warship targets (aircraft carriers, cruisers, destroyers, frigates) can be quickly and accurately identified in sea battlefield that can provide support for commanders' decision-making. In modern naval warfare, in order to effectively identify the combat intentions of ship targets, it is the first task to accurately identify ship targets. In recent years, with the development of satellite remote sensing technology, optical remote sensing ship target recognition has gradually become possible and provides an extremely rich data source. However, for some special types of ship targets, there are few high resolution remote sensing images, and the training samples gathered in the ship target dataset are unbalanced, which makes it difficult to accurately identify the ship targets. Random Forest (RF) has natural advantages in dealing with unbalanced data [1]. The method itself belongs to the integration algorithm. The component classifier (tree classifier) is constructed by bootstrap selection of data. Because the final result is obtained by voting of each component classifier, the impact of unbalanced data can be reduced to a certain extent. Huang et al. [2] extracted the conventional features and visual saliency features of ships, and then classified the high resolution remote sensing ship images using random forest based on mutual information (MIRF), which got good results. Unfortunately, random forests lack an effective mechanism for learning internal representations to help capture the main factors of data change [3]. Peter Kontschieder et al. [4] put forward a new method of deep neural decision forest that unifies deep architectures and the principle of divide-and-conquer of decision trees, surpassing the state-of-the-art performance on ImageNet.

Based on this, this paper proposes a weighted deep neural decision forest (WDNF), which uses hybrid integration technology to fully mine the potential useful information in large class samples, to solve the problem of loss of data information in using the stochastic down-sampling method for multi-class imbalance remote sensing ship image data.

2. The Proposed Approach

In this section, we propose an approach WDNF based on deep neural decision forests and the resampling method, which drives the evolution of the deep neural decision forests model by iteratively updating the training data. Compared to Deep Neural Decision Forests, WDNF can greatly improve the performance on the imbalanced dataset. We will first present the general framework and the basic algorithm of our model and then propose two mechanisms to enhance the basic model.

* Corresponding author.

E-mail address: narahuangsizhe@qq.com

2.1. Data Preprocessing Stage

Vessel image dataset preprocessing stage is a preparation-stage before the vessel image data are processed.

(1) Vessel image dataset is defined as Γ , which is the unbalanced dataset of l class and n dimension. $\Gamma = \Gamma_r \cup \Gamma_e$, (Γ_r is train dataset, Γ_e is test dataset). The categories in the training set are sorted by the number of samples from fewer to more: $\Gamma_1, \Gamma_2, \dots, \Gamma_L$. $\Gamma_r = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \dots \cup \Gamma_i$, Γ_i is data samples of class i in the training dataset. The number of Γ_1 is denoted as Γ_{\min} .

(2) In each round, several subsets Γ_{kt} are obtained from independent and identically distributed random down sampling of the remaining large classes in the dataset. Each subset of large classes and small classes together constitute a balanced multi-class training set Γ_{r1} .

(3) Get a set of training sets ($\Gamma_{r1}, \Gamma_{r2}, \dots, \Gamma_{rT}$) after T round.

2.2. Classifier Building Process with Weighted Deep Neural Random Forest

Let input space be denoted as X , output space as Y , decision node set of deep neural decision forests as N , and prediction node set as L . For each prediction node $l \in L$, there is a corresponding probability distribution π_l for Y . Each decision node has a decision function $d_n(\cdot; \Theta): X \rightarrow [0,1]$. Through parameter Θ , transfer X to $[0,1]$ interval. When a sample $x \in X$ reaches the decision node (split node n), it will assign the sample to the left subtree or the right subtree according to the output of $d_n(\cdot; \Theta)$. d_n is a binary value in the standard decision tree. In the deep neural decision forests [4], a probabilistic routing is considered, that is, the direction of routing is determined by the output of Bernoulli random variable mean $d_n(\cdot; \Theta)$. When a sample arrives at the leaf node (prediction node $l \in L$), the related tree prediction is given by the class label distribution π_l . In the case of random routing, the results of leaf prediction are determined by the mean of the sum of probabilities to each leaf node. To solve the imbalanced remote sensing ship images problem, we give larger weights to smaller classes and smaller weights to larger classes. The weight of each class is calculated as $w_i = |\Gamma_{k+1-i}|/|\Gamma_r|$.

Therefore, the probability Formula (1) of predicting that node sample X belongs to Y is given from tree T by decision nodes parameter Θ .

$$P_T[y | x, \Theta, \pi] = \sum_{l \in L} w_l \pi_l \mu_l(x | \Theta) \quad (1)$$

Where $\pi = (\pi_l)_{l \in L}$ and π_{ly} represents the probability of a sample x reaching the value y of leaf l . $\mu_l(x | \Theta)$ is the routing probability function that sample x can reach leaf l . Obviously, for all $x \in X$, there is $\sum_l \mu_l(x | \Theta) = 1$.

In order to express the routing probability function vividly, two binary relationships depending on tree structure are introduced: $l \swarrow n$ and $n \searrow l$. If l belongs to the left subtree of node n , then the value of $l \swarrow n$ is true. If l belongs to the right subtree of node n , then the value of $n \searrow l$ is true. Using these relations expression, μ_l can be defined as follows:

$$\mu_l(x | \Theta) = \prod_{n \in N} d_n(x; \Theta)^{l \swarrow n} \bar{d}_n(x; \Theta)^{l \searrow n} \quad (2)$$

Where $\bar{d}_n(x; \Theta) = 1 - d_n(x; \Theta)$ and $d_n(x, \Theta) = \sigma(f_n(x, \Theta))$. $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. $f_n(\cdot; \Theta): X \rightarrow R$ is a real-valued function depending on the sample and the parameterization Θ . 1_p represents the probability value of the function under condition P . Although Equation (2) multiplies all decision nodes, only those decision nodes that go along the root node to the leaf node l contribute to μ_l . This is because all other decision nodes $1_{l \swarrow n}$ and $1_{n \searrow l}$ are 0 (assuming $0^0 = 1$).

3. Learning Weighted Trees by Back-Propagation

In order to obtain the weighted deep neural decision tree model, it is necessary to evaluate the parameter Θ of decision nodes and the distribution π of leaf prediction. Under log-loss, the empirical minimum principle is used to estimate these two parameters for a given dataset $\Gamma \subset x \times y$, that is, to search the estimated parameters corresponding to the minimum value of the following risk items:

$$R(\Theta, \pi; \Gamma) = \frac{1}{|\Gamma|} \sum_{(x,y) \in \Gamma} L(\Theta, \pi; x, y) \quad (3)$$

Where $L(\Theta, \pi; x, y)$ is the log-loss term for the training sample $(x, y) \in \Gamma$, which is defined as follows:

$$L(\Theta, \pi; x, y) = -\log(P_T[y | x, \Theta, \pi]) \quad (4)$$

P_T is defined in Equation (1). In this paper, we consider a two-step optimization strategy to alternately update Θ and π to minimize Equation (4). The details are as follows.

3.1. Learning of Decision Nodes

In updating Θ , we use the re-sampling and training set partitions methods. All decision functions depend on the common parameter Θ , which in turn parameterizes each function f_n . f_n is one of the output units of the deep neural network, and Θ can receive all the parameters of the deep neural network. It will not be a difficult and huge problem to optimize the risk function with respect to Θ when π is known. Therefore, the stochastic gradient algorithm can be used to minimize the value of the risk function with Θ as a variable, just as in the deep neural network to minimize the risk function.

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta \frac{\partial R}{\partial \Theta}(\Theta^{(t)}, \pi; B) = \Theta^{(t)} - \frac{\eta}{|B|} \sum_{(x,y) \in B} \frac{\partial L}{\partial \Theta}(\Theta^{(t)}, \pi; x, y) \quad (5)$$

Where η is the learning rate, which is a value greater than 0, and $B \subseteq \Gamma$ is a random sample subset of the training set. According to the chain rule, the gradient of loss can be decomposed as follows:

$$\frac{\partial L}{\partial \Theta}(\Theta, \pi; x, y) = \sum_{n \in N} \frac{\partial L(\Theta, \pi; x, y)}{\partial f_n(x; \Theta)} \frac{\partial f_n(x; \Theta)}{\partial \Theta} \quad (6)$$

Here, the gradient term depending on the decision tree (see [4]) is

$$\frac{\partial L(\Theta, \pi; x, y)}{\partial f_n(x; \Theta)} = d_n(x; \Theta) A_{n_r} - \bar{d}_n(x; \Theta) A_{n_l} \quad (7)$$

Where n_l and n_r represent the left subtree and the right subtree of node n respectively. For a node $m \in N$, A_m can be defined as follows:

$$A_m = \frac{\sum_{l \in L_m} \pi_l \mu_l(x | \Theta)}{P_T[y | x, \Theta, \pi]} \quad (8)$$

Where $L_m \subseteq L$ represents the leaf subset of a subtree with node m as its root node.

Finally, considering the optimization process of stochastic gradient descent, i.e., the elastic backward propagation method, this method can automatically adjust the learning rate of each parameter according to the change of the sign of the risk function's partial derivative in the last iteration.

3.2. Learning Prediction Nodes

The method of updating parameter Θ is given above. Now, we consider the problem of minimizing Equation (4) with respect to π when Θ is determined, that is, the problem of minimizing Equation (4) with respect to π .

$$\min_{\pi} R(\Theta, \pi; T) \quad (9)$$

This is a convex optimization problem, and a global optimal solution can be obtained. Such similar problems can be encountered in some decision trees [5], but they are usually limited to a single node. Here, all decision-making nodes will be taken into account, and the predicted values of all leaf nodes will be estimated jointly. In order to calculate the global minimum of Equation (9), the following solutions are adopted in this paper:

$$\pi_{ly}^{(t+1)} = \frac{1}{Z_l^{(t)}} \sum_{(x, y') \in \Gamma} \frac{I_{y=y'} \pi_{ly}^{(t)} \mu_l(x | \Theta)}{P_T[y | x, \Theta, \pi^{(t)}]} \quad (10)$$

Where $P_T[y | x, \Theta, \pi^{(t)}] = \sum_{l \in L} w_l \pi_{ly}^{(t)} \mu_l(x | \Theta)$, $l \in L$, $y \in Y$, and $Z_l^{(t)}$ is the normalization factor guaranteeing $\sum_y \pi_{ly}^{(t+1)} = 1$. The initial value of $\pi^{(0)}$ can be any value greater than 0. A typical choice is to take the uniform distribution of all leaf nodes as the starting value, $\pi_{ly}^{(0)} = |Y|^{-1}$. The step size of each iteration in the updating rule of formula (10) is arbitrary. Each iteration updating ensures that the function decreases strictly before the value of the risk reaches a fixed value (the proof of Equation (10) is shown in [4]).

When updating π with Equation (10), any non-optimal estimation result of π will have a great impact on the final prediction, so this paper does not randomly divide training samples into several batch datasets at each iteration. Instead, it directly applies all samples to Equation (11) to upgrade π at one time, so as to obtain a more reliable π estimation, which is different from batch gradient updating used in estimating Θ . Moreover, as mentioned above, in each iteration, the staggered updates of π , Θ , and π will only be estimated once, while Θ will be estimated according to the sample batch T , as shown in Figure 1.

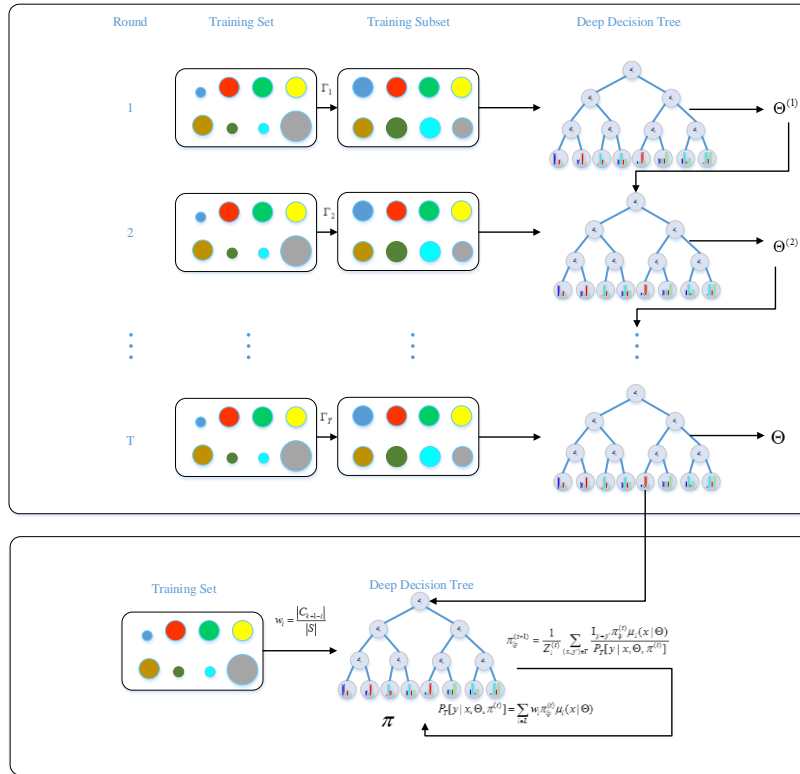


Figure 1. Training algorithm of the proposed method

3.3. Training Process

The proposed training algorithm is described below.

Algorithm 1: Proposed method

Inputs: Training dataset Γ_r , testing dataset Γ_e , n epochs, iteration T

random initialization Θ

Outputs: Equilibrium weighted deep neural decision forests

Training Stage:

Step 1: For training dataset Γ_r , categories are sorted from fewer to more samples: $\Gamma_{r1}, \Gamma_{r2}, \dots, \Gamma_{rT}$. The number of Γ_{r1} class samples is denoted as Γ_{\min}

Step 2: For $t = 1, 2, \dots, T$

 Bootstrap sampling for class Γ_{r1} to get P samples

 Several subsets Γ_{kt} are obtained from independent and identically distributed random down-sampling of the remaining large classes in training dataset Γ_r . $\Gamma_{kt} = \Gamma_{\min}$

End for

Step 3: Get a set of training sets $(\Gamma_{r1}, \Gamma_{r2}, \dots, \Gamma_{rT})$ after T rounds

Step 4: The weight of each class is calculated as $w_i = \frac{|\Gamma_{k+1-i}|}{|\Gamma_r|}$

Step 5: For all $i \in \{1, \dots, nEpochs\}$ do

 For all $\Gamma_{r1}, \Gamma_{r2}, \dots, \Gamma_{rT}$

 Updating Θ by SGD

$$\begin{aligned}\Theta^{(t+1)} &= \Theta^{(t)} - \eta \frac{\partial R}{\partial \Theta}(\Theta^{(t)}, \pi; B) \\ &= \Theta^{(t)} - \frac{\eta}{|B|} \sum_{(x,y) \in B} \frac{\partial L}{\partial \Theta}(\Theta^{(t)}, \pi; x, y)\end{aligned}$$

end for

 Calculating π , by iterating $\pi_{ly}^{(t+1)} = \frac{1}{Z_l^{(t)}} \sum_{(x,y) \in \Gamma} \frac{I_{y=y} \cdot \pi_{ly}^{(t)} \mu_l(x|\Theta)}{P_T[y/x, \Theta, \pi^{(t)}]}$, where

$$P_T[y|x, \Theta, \pi^{(t)}] = \sum_{l \in L} w_l \pi_{ly}^{(t)} \mu_l(x|\Theta)$$

end for

Test Stage:

$$P_F[y|x] = \frac{1}{k} \sum_{h=1}^k P_{F_h}[y|x]$$

4. Experimental Results

The experimental data are real remote sensing images. First, land-sea segmentation and cloud mask are preprocessed. Then, the suspected target areas are obtained by the connected operator [6]. Finally, the target slice is obtained by the GPAC image segmentation algorithm [7], as shown in Figure 2. We fill each image slice to 170×170 and then use bilinear interpolation to scale the image to 45×45 . The experimental dataset of ship classification is a total of 2160 images, which are divided into six ship classes, each of which has 360 pieces. The initial ship dataset V_{org} is processed into unbalanced dataset V_{ub} by sampling without replacement, as shown in Table 1. We compared the classification performance of the proposed method against those of six representative ship classification methods to observe whether the proposed method can effectively solve the problem of imbalanced data on the premise of guaranteeing the classification performance. 70% of the data is allocated as the training set, and the remaining 30% of the data is used as the testing set.

1) Recognition performance comparisons on two datasets

In order to verify the validity of the proposed method, we compare the WDNF with five representative ship classification methods: the k-nearest neighbor (KNN) [8] method, support vector machine (SVM) [9] method, affinity propagation (AP) [10], entropy-based hierarchical discriminant regression (E-HDR) [11] method, deep neural decision forest (dNDF), and MIRF [2]. dNDF [4] and WDNF adopt the same structure.

As shown in Table 2, the performance of WDNF on ship unbalanced datasets is impressive and has significant

advantages over other classification methods. WDNF is based on the combination of boosting and deep random forest. The performance of WDNF is better than those of the other six methods for two reasons. First, the structure of the deep random forest can extract useful information. It can actively select useful training samples and significantly reduce the error rate compared with other classifiers.

Table 1. Initial vessel dataset and unbalanced vessel dataset

	Carrier	Frigate	Destroyer	Container	Cargoship	Cargoship	Total
V_{org} (Initial ship dataset)	360	360	360	360	360	360	2160
V_{ub} (Unbalanced dataset)	360	300	280	260	200	64	1464

Table 2. Classification accuracy on vessel dataset

	SVM	KNN	HDR	PGM	RF	MIRF	dNDF	WDNF
V_{org} (Initial ship dataset)	72.3	69.4	70.4	70.8	71.3	72.1	91.5	90.9
V_{ub} (Unbalanced dataset)	69.4	68.6	60.4	58.4	64.5	65.3	85.7	88.2

2) Effects of resampling and weighting on deep neural decision forest

In order to test the effects of resampling and weighting in proportion to the unbalanced dataset V_{ub} on the deep neural decision forest, we compared WDRF against the deep neural decision forest without resampling and the deep neural decision forest without weighted (the three methods have the same structure). The deep neural decision forest without resampling and deep neural decision forest without weighted are respectively represented as "WDNF_B" and "WDNF_W" in Figure 3. Compared with the WDNF method, the WDNF_B and WDNF_W methods do not perform well, but the weighting is more important for unbalanced datasets, because this method can pay more attention to the classes with fewer samples.

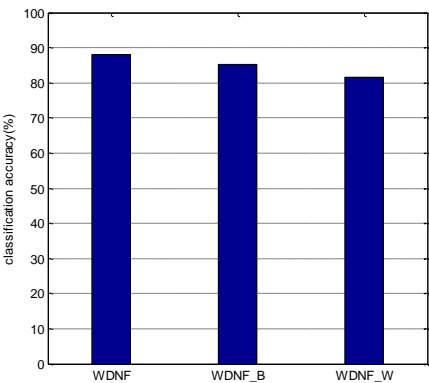
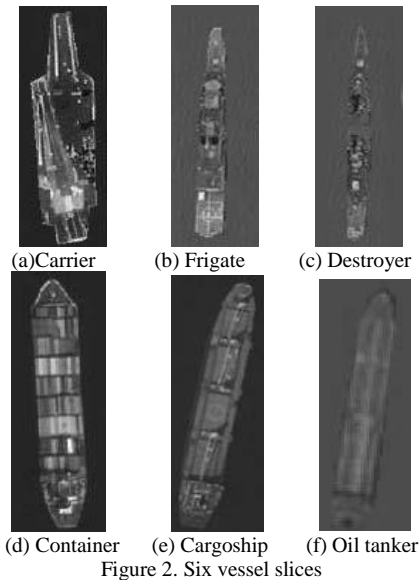


Figure 3. Classification accuracy of WDNF, WDNF_B, WDNF_W

5. Conclusions

In this paper, a weight-based classification training model for deep neural decision tree is discussed. The traditional decision tree cannot learn data features and train massive data, and its application effect on unbalanced datasets is not significant. In order to overcome this problem, an unbalanced classification method is added to the deep decision forest and applied to the classification of ship targets. The results show that it has better performance than the traditional decision tree classifier and deep neural network in ship target classification.

Acknowledgements

The authors would like to acknowledge Wuhan Digital Engineering Research Institute for providing materials and equipment for this study.

References

1. M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for Class Imbalance Problem: Bagging, Boosting and Hybrid Based Approaches," *IEEE Transactions on Systems*, Vol. 42, No. 4, pp. 463-484, 2012
2. S. Huang, X. Xia, and H. Xu, "A Remote Sensing Ship Recognition using Random Forest," in *Proceedings of the 4th International Conference on Information Science and Cloud Computing (ISCC2015)*, Guangzhou, China, December 2015
3. Y. Bengio, O. Delalleau, and C. Simard, "Decision Trees do not Generalize to New Variations," *Computational Intelligence*, Vol. 26, No. 4, pp. 449-467, 2010
4. P. Kotschieder, M. Fiterau, A. Crimisi, and S. Bulò, "Deep Neural Decision Forests," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 1467-1475, 2015
5. S. Bulò and P. Kotschieder, "Neural Decision Forests for Semantic Image Labelling," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 81-88, 2014
6. C. Corbane, L. Najman, and E. Pecoul, "A Complete Processing Chain for Ship Detection using Optical Satellite Imagery," *International Journal of Remote Sensing*, Vol. 31, No. 22, pp. 5837-5854, 2010
7. B. S. Manjunath and B. Sumengen, "Graph Partitioning Active Contours (GPAC) for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 509-521, 2006
8. J. X. Sun, "Modern Pattern Recognition," 2nd Edition, Higher Education Publishing Company, Beijing, 2008
9. G. C. Anagnostopoulos, "SVM-based Target Recognition from Synthetic Aperture Radar Images using Target Region Outline Descriptors," *Nonlinear Anal*, Vol. 71, No. 12, pp. 2934-2939, 2009
10. B. J. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points," *Science*, Vol. 315, No. 5814, pp. 972-976, 2007
11. W. Guo, X. Xia, and X. Wang, "A Remote Sensing Ship Recognition Method of Entropy-based Hierarchical Discriminant Regression," *Optik-International Journal for Light and Electron Optics*, Vol. 126, No. 20, pp. 2300-2307, 2015