Building Decision Tree for Imbalanced Classification via Deep Reinforcement Learning

Guixuan Wen 201914131072@CQU.EDU.CN
College of Computer Science, Chongqing University, Chongqing, 400044 China

Kaigui Wu KAIGUIWU@CQU.EDU.CN
College of Computer Science, Chongqing University, Chongqing, 400044 China

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Data imbalance is prevalent in classification problems and tends to bias the classifier towards the majority of classes. This paper proposes a decision tree building method for imbalanced binary classification via deep reinforcement learning. First, the decision tree building process is regarded as a multi-step game and modeled as a Markov decision process. Then, the tree-based convolution is applied to extract state vectors from the tree structure, and each node is abstracted into a parameterized action. Next, the reward function is designed based on a range of evaluation metrics of imbalanced classification. Finally, a popular deep reinforcement learning algorithm called Multi-Pass DQN is employed to find an optimal decision tree building policy. The experiments on more than 15 imbalanced data sets indicate that our method outperforms the state-of-the-art methods.

Keywords: Decision tree; Imbalanced classification; Deep reinforcement learning; Tree-based convolution.

1. Introduction

For the last few years, machine learning methods are widely applied to practical issues and attain tremendous success. However, the data collected from some domains such as abnormal detection (Qin et al., 2020; Chen et al., 2020), disease diagnosis (Khalilia et al., 2011; Yildirim, 2017), risk behavior recognition (Chen et al., 2019; Chi et al., 2020), and so on usually is imbalanced, which is one of the thorniest tasks in the application. More importantly, the minority class is often more significant than the majority class. For example, the instances that belong to one class (e.g., cancer patient) can be 1000 times less than that in another class (e.g., healthy people) and the algorithm is to detect the minority one (i.e., cancer patient). Most machine learning algorithms are not proposed for the consideration of data skew. Therefore, though achieving sufficiently excellent performance on balanced data sets, they fail when faced with an imbalanced situation.

Numerous algorithms have been proposed for imbalanced data classification during the past two decades. They usually can be divided into two groups: the data level and the algorithmic level. The main idea of the former is to rebalance the distribution of data by different resampling techniques such as random undersampling (RUS), random oversampling (ROS), synthetic minority oversampling (SMOTE) (Chawla et al., 2002), and so on. In contrast, the latter group tries to adjust the original algorithms by assigning weights or

costs towards different classes or samples, reducing the bias caused by sample size between classes. This paper focuses on the decision tree (DT), which is one of the simplest machine learning algorithms and intuitively interpretable. Generally, the building of a decision tree can be considered a greedy algorithm. At each decision node, a locally best attribute is selected to split the data into child nodes. This process is repeated until a leaf node is reached, where further splitting is not possible. One of the most popular splitting criteria is Information Gain (IG) (Quinlan, 1986), an impurity-based splitting criterion. DT based on IG performs quite well for balanced data sets where the class distribution is uniform. However, as the prior probability of class is used to calculate a node's impurity degree, on an imbalanced dataset, IG becomes biased towards the majority class, which is also called skew sensitive. To improve standard DT performance in imbalanced classification domains, several splitting criteria are proposed to build DTs, such as Hellinger Distance (Cieslak and Chawla, 2008), Inter-node Hellinger Distance (Akash et al., 2019), and Class Confidence Proportion (Liu et al., 2010). Besides these, to deal with the class imbalance problem in Lazy DT building process, two skew insensitive split criteria based on Hellinger distance and K-L divergence are proposed in (Su and Cao, 2019). Furthermore, there are also some new ways to construct the DT model that do not belong to heuristic methods. For example, Pyeatt (2003) proposed a reinforcement learning approach to automatically search for splitting strategies in the global search space based on the evaluation of long-term payoff and Blake and Ntoutsi (2018) applied this method to data streams with concept drifts.

Unlike the methods mentioned above, this paper proposes a decision tree building method for imbalanced binary classification based on deep reinforcement learning (DRL). First, the decision tree building process is considered as a multi-step game that can be modeled as a Markov decision process (MDP) in deep reinforcement learning. Then, the tree-based convolution (Mou et al., 2016) is employed to extract state vector from tree structure and abstract each node into a parameterized action. Next, the reward function is designed based on a range of evaluation metrics of imbalanced classification. Finally, a popular DRL algorithm called Multi-Pass DQN (MP-DQN) (Bester et al., 2019) is used to find an optimal DT building policy. To verify our proposed method's performance, experiments on 18 data sets are conducted and compared them with the decision tree methods (Cieslak and Chawla, 2008; Akash et al., 2019; Liu et al., 2010). The results show that our method achieves the most excellent performance on imbalanced issues.

The rest of this paper is organized as follows: The second section introduces the research decision tree of imbalanced data classification and deep reinforcement learning applications on imbalanced classification problems. The details of our proposed method will be described in the third section. The fourth section shows the experimental results and evaluates the performance of our method compared with other methods. Conclusions and future work will be discussed in section five.

2. Related Work

2.1. Decision Tree for Imbalanced Classification

In 2008, Cieslak and Chawla (2008) proposed the Hellinger Distance Decision Tree (HDDT), which employs the Hellinger distance instead of information gain as the splitting criterion. Hellinger distance which is one kind of f-divergence can measure the similarity of two dis-

tributions. Because the normalized frequencies of all partitions are used instead of class probability, the Hellinger distance is skew insensitive, making the HDDT perform better than other decision tree methods based on information gain on imbalanced data sets. However, as HDDT tries to make pure leaves by capturing deviation between class conditionals which leads to smaller coverage, it performs poorly for more balanced class distribution. Besides, Liu et al. (2010) hold that some rules with high confidence generated by traditional decision tree algorithm may not be essential to split and lead to bias towards the majority classes. They introduced the Class Confidence Proportion(CCP) to replace the entropy to address this problem. By embedding CCP in information gain, the Class Confidence Proportion Decision Tree (CCPDT) is proposed. Thanks to the CCP not taking the class priors into account, CCPDT is also insensitive to data imbalance. However, when the information gain values of two splits are equal, CCPDT employs Hellinger distance to make a final bid for victory, which leads to poor performance like HDDT on more balanced data sets. In 2019, Akash et al. (2019) proposed the Inter-node Hellinger Decision Tree (iHD) and its weighted variant iHDw. iHD and iHDw utilize square Hellinger distance to evaluate class distributions' dissimilarity between parents and children instead of all partitions to generate mutually exclusive regions. Based on iHD, iHDw introduces the class's instance proportion to calculate the weight for the distance between parent and children, contributing to attaining purer child nodes.

2.2. Deep Reinforcement Learning

Deep reinforcement learning (DRL), which is the combination of reinforcement learning (RL) and deep learning, has attracted much attention from researchers and is mainly applied to address sequential decision-making problems. Inspired by the learning behaviors of animals, RL controls agents interact with the environment and get the rewards used to train the agent. Generally, RL problem is modeled as a Markov Decision Process (MDP) that can be represented as (S, A, R, T, γ), where S and A indicate the state space and action space respectively, T is the state transition probability, R denotes rewards from the environment and γ is the discount factor that used to calculate expected return. A complete interaction in RL can be described as follow: an agent firstly observes a state s_t from the environment at time step t, after taking an action a_t, it gets a reward r_t, and the environment transits to the next state s_{t+1} according to the probability $p = P(s_{t+1}|S = s_t, A = a_t)$. DRL aims to learn a policy network π to control the agent to maximize total reward during the interaction with environments. Thus according to different learning forms, DRL can be divided into three paradigms. The first is policy-based methods such as Policy Gradient (PG) (Mnih et al., 2015) and Proximal Policy Optimization (PPO) (Schulman et al., 2017). The second is value-based methods that contain Deep Q Network (DQN) (Mnih et al., 2015) and its variants Double DQN (van Hasselt et al., 2016), Dueling DQN (Wang et al., 2016), etc. The last but no less paradigm is actor-critic methods, including standard Actor-Critic (AC) (Thomas and Brunskill, 2017), Advantage Actor-Critic (A2C), and Asynchronous Advantage Actor-Critic (A3C) (Mnih et al., 2016), etc. Besides, from the perspective of action space, DRL can also be classified as discrete actions space methods, continuous actions space methods, and parameterized actions space methods. One of the most famous continuous action methods is Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016),

while Parameterized DQN (P-DQN) (Xiong et al., 2018) and Multi-Pass DQN (MP-DQN) (Bester et al., 2019) belong to parameterized actions space methods. This paper considers the action space of decision building as the parameterized action space paradigm.

2.3. DRL for Imbalanced Classification

In recent years, there are many excellent pieces of research about DRL focus on imbalanced classification. The recent work (Tan et al., 2018) introduces that the maximum likelihood in supervised machine learning is a particular case of a policy optimization framework. Therefore, Hu et al. (2019) put forward a new data manipulation method that can automatically handle different schemes(e.g., Augmentation & Weighting) by different parameter settings of reward function in DRL. Besides, Peng et al. (2019) viewed the data selected process as MDP and trained a data sampler via DRL. Unlike classical DRL architectures, they used a GRU unit to remember the sampling sequence information and directly applied imbalanced evaluation metrics (e.g., f-measures) as reward functions. However, Lin et al. (2020) directly considered the classification problem as a guessing game divided into a sequential MDP. They utilized an imbalanced ratio as a reward function that guides the agent to learn the optimal classification policy for imbalanced data. Unlike the methods mentioned above, in this paper, we think of the process of building a decision tree as an MDP and applying imbalanced evaluation metrics as reward functions similar to (Peng et al., 2019).

3. Method

Before the introduction of our method, some notations need to be explained, depicted in Table 1. State, action and reward are three vital elements when the decision tree building process is viewed as an MDP. In our method's framework, as shown in Fig 1, the state s in time step t can be extracted by a tree-based convolution layer embedding in the actor-network of MP-DQN. After selecting the k-th attribute with maximal q value, the agent acts an action containing a discrete attribute id and a threshold value. Because each node is abstracted into a parameterized action, the environment easily updates the tree T_t to T_{t+1}. Moreover, according to classification results of tree T_{t+1} and evaluation metrics, a reward function is designed. When state, action and reward are sure, a decision tree building policy can be found via DRL algorithms.

3.1. State

In the binary classification scenario, we assume that all the attributes are continued. Therefore each node in the decision tree can be represented as a real-value vector with two-part. The first part is attribute code, and the other is a threshold value, depicted in Fig 2 (a). To retain the structure information, the tree-based convolution layer contains a group of fixed-depth feature detectors is used to extract features from the decision tree at each time step, depicted in Fig 2 (b). After that, we will get a set of structural features that can synthesize a new tree similar to the original shape and size. Like convolution in image processing, a one-way pooling layer is utilized to pool all features to one vector.

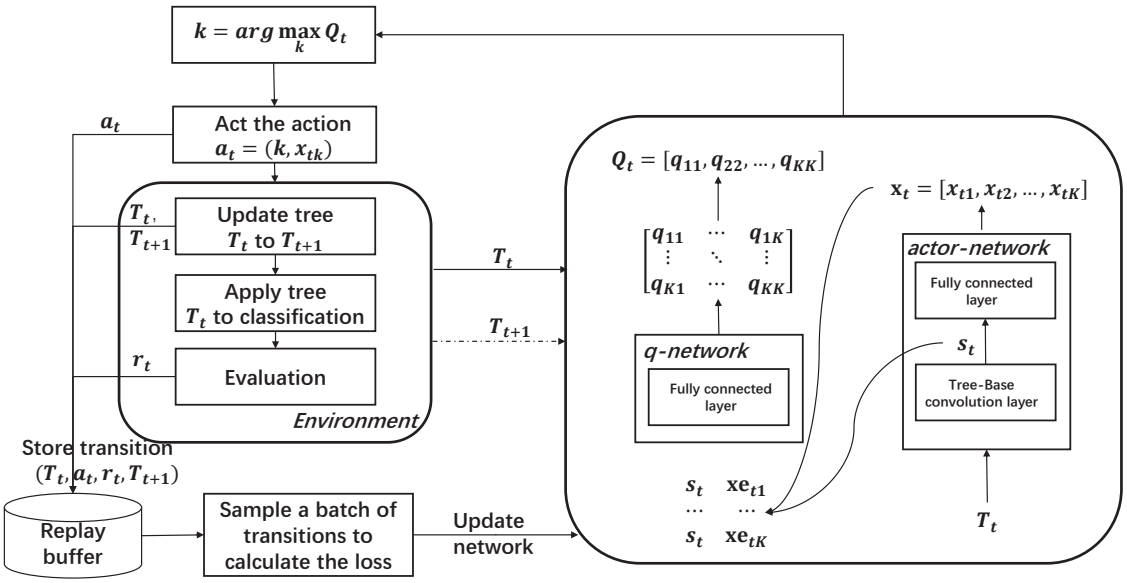

Figure 1: The framework of our method.

Table 1: Description of the notations.

Notations	Explanation
t	A time step
K	The number of attribute
T_t	The decision tree T at time step t
s_t	A state s at time step t
r_t	A reward r at time step t
X_k	The value space of the k-th attribute
x_t	The threshold vector at time step t
x_{tk}	The threshold value corresponds with k-th attribute at time step t
xe_{tk}	It is a vector where the k-th dimension is equal to x_{tk}, but everything else is zero
q_{kk}	The q value corresponds with k-th attribute at k-th pass
Q_t	The q value vector at time step t
$\mathbb{Q}_\mathrm{x}$	The actor-network
$\mathbb{Q}_\mathrm{q}$	The q-network
θ_x	The parameters correspond with actor-network
θ_q	The parameters correspond with q-network

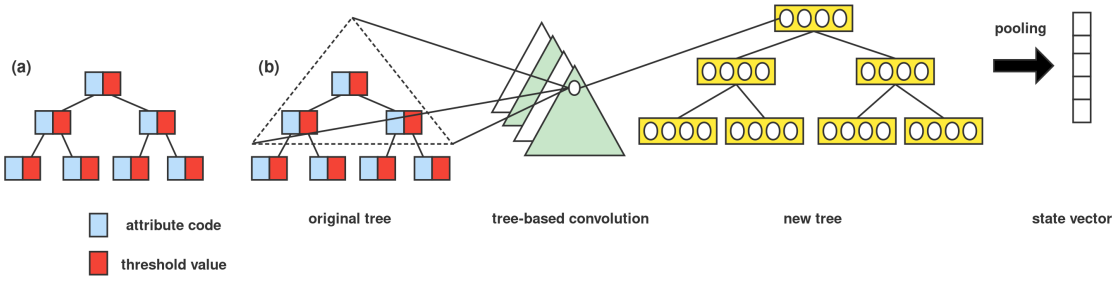

Figure 2: (a) the vector representation of the nodes of a decision tree. (b) the transformation process from decision tree to state vector.

3.2. Action

Parameterized action spaces (Masson et al., 2016) consist of a set of discrete actions, $A_d = [K] = \{k_1, k_2, k_{3,\ldots,}k_n\}$, where n is the number of actions and each k has a corresponding m dimensional continues action-parameter $x_k \in X_k \subseteq R^{m_k}$, where the X_k is the continues action space of action k. This can be written as (1). We consider the continues attributes in DT in binary classification. Thus the attributes can be represented as discrete action A_d and the threshold values are the corresponding continues action-parameter $x_k \in X_k \subseteq R$. Each node in DT corresponds with a action $a_k = (k,\ x_k)$.

$$A = \bigcup_{k \in [K]} \{a_k = (k, x_k) \,|x_k \subseteq X_k\} \tag{1}$$

3.3. Reward

Intuitively we can apply the generated but incomplete decision tree to classify on the training set at step t. Next, the predicted results $\hat{Y}_t$ and the truth Y_t are used to calculate a score sc_t based on arbitrary evaluation metrics, such as F-Measure and G-Mean. Finally the reward r_t is easily obtained according to (2). That is to say, the positive r_t means action a_t improves the performance of the decision tree while negative reward r_t decreases it. Note that the sc_0 is usually set to 0 or 0.5. The difference with (Peng et al., 2019) is that we employ the decision tree to classify at every step t instead of the terminal. In other words, if the initial sc_0 is set to zero, the total reward R is equivalent to the evaluation score of the final classifier. Similarly, setting sc_0 at 0.5 is equivalent to adding a baseline.

$$r_t = sc_t - sc_{t-1} \tag{2}$$

3.4. Training Details

According to the form of action, we can easily apply the recent DRL algorithm with parameterized action spaces in our training, such as P-DQN and MP-DQN. Consider that MP-DQN is more theoretical and has better performance than P-DQN, the former is adopted by us.

Algorithm 1

Require:

Data Set D, max nodes number N, max training episode M, exploration parameter ε, minibatch size B, replay buffer L, discount rate γ

Ensure:

for $m = 0$ to $M - 1$ **do**
  Initialize full binary tree T_0
  for $t = 0$ to $N - 1$ **do**
    Compute an action parameters and state vetor : $\mathrm{x}_t, s_t \leftarrow \mathbb{Q}_{\mathrm{x}}(T_t; \theta_{\mathrm{x}})$
    Choose an action based ϵ-greedy policy:

$$a_t = \begin{cases} random\ sample & probability\ \epsilon \\ (k_t, x_{tk})\ \ such\ that\ k_t = arg\max\limits_{k \in [K]} \mathbb{Q}_{\mathrm{q}}(s_t, \mathrm{xe}_{tk}; \theta_{\mathrm{q}}) & probability\ 1 - \epsilon \end{cases}$$

    where $\mathrm{xe}_{tk} = (0, 0, ..., x_{tk}, 0, ..., 0)$
    Take action a_t, change T_t into T_{t+1}
    Apply T_{t+1} to classify on D and observe reward r_t
    Store transition (T_t, a_t, r_t, T_{t+1}) into L
    Sample B transitions $(T_b, a_b, r_b, T_{b+1})_{b \in [B]}$ from L randomly
    Set the target

$$y_b = \begin{cases} r_b & if\ T_{b+1}\ is\ the\ terminal \\ r_b + \gamma \max\limits_{k \in [K]} \mathbb{Q}_{\mathrm{q}}(s_{b+1}, \mathrm{xe}_{b+1,k}; \theta_{\mathrm{q}}) & otherwise \end{cases}$$

    where $s_{b+1}, \mathrm{x}_{b+1} = \mathbb{Q}_{\mathrm{x}}(T_b, \theta_{\mathrm{x}})$, $\mathrm{xe}_{b+1,k} \in \mathrm{x}_{b+1}$
    Perform a gradient descent step on $L_{\mathrm{q}}(\theta_{\mathrm{q}})$ and $L_{\mathrm{x}}(\theta_{\mathrm{x}})$

$$L_{\mathrm{q}}(\theta_{\mathrm{q}}) = E\left(y_b - \mathbb{Q}_{\mathrm{q}}(s_b, \mathrm{xe}_{bk}; \theta_{\mathrm{q}})\right)^2$$

$$L_x(\theta_{\mathrm{x}}) = E\left(-\sum_{k=1}^{K} \mathbb{Q}_{\mathrm{q}}(s_b, \mathrm{xe}_{bk}; \theta_{\mathrm{q}})\right)$$

    if $r_t < 0$ **then**
      Break
    end if
  end for
end for

As indicated in Algorithm 1, we should first build a complete binary tree T_0 with N nodes and set a default attribute and a threshold value for each node. At the same time, the parameters of actor-network and q-network θ_{x} and θ_{q} are initialized. After that, a complete interaction can be described as the following steps: agent observes the state s_t, representing tree T_t at time step t. It takes an action a_t containing a discrete attribute k and a continues threshold value x_k. Next, the environment updates the attribute and threshold value of the t-th node of tree T_t into k and x_k. Note that nodes are numbered according to the level traversal of the tree. Last, the T_t goes to T_{t+1}, which is viewed as a classifier on the training set. Finally, the reward r_t will be calculated based on the classification result according (2). Like DQN, MP-DQN also uses an experience replay mechanism to decrease the temporal correlations during the update and improve rare experience efficiency. Thus, the agent must store the transition (s_t, a_t, r_t, s_{t+1}) in the replay buffer and sample a batch of transitions to update the actor-network q-network in each round of interaction. The interaction stops when the agent receives a negative reward or traversed the last node of the tree.

4. Experiment

4.1. Data Sets

Table 2: Description of the data sets. #I, #F denote the number of instances, attributes respectively

Data Sets	#I	#F	IR
ecoli-0-1-vs-2-3-5	244	7	9.17
ecoli-0-1-4-6-vs-5	187	6	13.0
ecoli-0-1-4-7-vs-2-3-5-6	336	7	10.59
ecoli-0-6-7-vs-5	220	6	10.0
ecoli2	336	7	5.46
haberman	306	3	2.78
new-thyroid1	215	5	5.14
new-thyroid2	215	5	5.14
vehicle3	846	18	3.0
winequality-red-4	1599	11	29.17
wisconsin	683	9	1.86
yeast-0-2-5-6-vs-3-7-8-9	1004	8	9.14
yeast1	1484	8	2.46
glass0	214	9	2.06
glass6	214	9	6.38
pima	768	8	1.87
africa recession	486	53	11.9
insurance	382154	10	5.1

Over 15 data sets from reality are described in TABLE 2, which contains the number of instances and attributes. Besides, the imbalance ratio (IR) that measures the degree of

imbalance between the classes of majority and minority is also provided. These data sets are collected from three well-known public sources called UCI Machine Learning Repository (Asuncion and Newman, 2007), KEEL Imbalanced Data Sets (Alcalá-Fdez et al., 2011) as well as Kaggle.

4.2. Evaluation Metrics

It is unreasonable to take the accuracy as the metric to evaluate classifier performance in imbalanced classification. At present, the commonly used imbalanced classification evaluation metrics include G-Mean and the area under the ROC curve (AUC) that are applied in our experiments.

To make the experiment more convincing, we not only take 10-fold cross-validation but also conduct a Friedman test and Nemenyi test. The null hypothesis of the Friedman test is that all the methods are equivalent. Precisely, assuming there are k methods, N data sets, and average rank r_i corresponding to each method m_i, we should compute two essential statistics χ_F^2 and F_F that calculated as (3) and (4), then compare the value of F_F with the critical value of given significance level α. If the null hypothesis is rejected, we need to take a Nemenyi test for further comparison. For given significance level α, the critical value CD can be calculated as (5) on the Nemenyi test. If the average rank difference between the two methods is greater than CD, it is believed that the two methods have different performances.

$$\chi_F^2 = \frac{12N}{k\,(k+1)} \left(\sum_{i=1}^{k} r_i^2 - \frac{k\,(k+1)^2}{4} \right) \tag{3}$$

$$F_F = \frac{(N-1)\,\chi_F^2}{N\,(k-1) - \chi_F^2} \tag{4}$$

$$CD = q_\alpha \sqrt{\frac{k\,(k+1)}{6N}} \tag{5}$$

4.3. Result

The comparison of the six methods can be clearly seen in TABLE 3 and TABLE 4. We conduct the experiments over sixteen data sets and make a Friedman test for validation.

The average ranking of G-Mean shows that our method has the best performance. To further verify this conclusion's reliability, Friedman's χ_F^2 statistic and Iman's F_F statistic are calculated as 26.54 and 7.11, respectively. With six methods and sixteen data sets, Iman's F_F statistic follows the F distribution with degrees of freedom of 5 and 85. At the 95% confidence level, it is easy to say that $F_F = 7.11$ is greater than $F_{0.05} = 2.322$. Thus we should reject the null hypothesis that all the methods have the same performance. Next, we compute the critical difference CD, which is 1.434 on the Nemenyi test. It can be concluded that our method surpasses all the other five at a 95% confidence level.

Similarly, Friedman's χ_F^2 statistic and Iman's F_F statistic are 26.13 and 6.95 based on AUC results. Obviously, $F_F = 6.95$ is grater than $F_{0.05} = 2.322$, which means all methods have different performances. Furthermore, according to the difference CD, our method also outperforms others.

Table 3: G-Mean of six methods on eighteen data sets

Data Sets	C4.5	HDDT	CCPDT	iHD	iHDw	ours
ecoli-0-1-vs-2-3-5	0.848	0.866	0.860	0.854	0.848	0.965
ecoli-0-1-4-6-vs-5	0.892	0.788	0.793	0.793	0.892	0.913
ecoli-0-1-4-7-vs-2-3-5-6	0.686	0.759	0.603	0.783	0.783	0.783
ecoli-0-6-7-vs-5	0.819	0.897	0.919	0.655	0.756	0.926
ecoli2	0.856	0.865	0.903	0.870	0.836	0.889
haberman	0.520	0.542	0.507	0.515	0.511	0.515
new-thyroid1	0.925	0.887	0.872	0.872	0.872	0.894
new-thyroid2	0.913	1.000	0.859	0.866	0.913	0.957
vehicle3	0.696	0.740	0.703	0.751	0.765	0.769
winequality-red-4	0.224	0.226	0.322	0.319	0.226	0.394
wisconsin	0.921	0.914	0.917	0.914	0.925	0.970
yeast-0-2-5-6-vs-3-7-8-9	0.731	0.717	0.713	0.724	0.690	0.864
yeast1	0.615	0.598	0.645	0.607	0.619	0.654
glass0	0.827	0.737	0.739	0.739	0.728	0.826
glass6	0.830	0.830	0.830	0.823	0.816	0.880
pima	0.675	0.693	0.715	0.726	0.685	0.715
africa recession	0.526	0.458	0.455	0.521	0.367	0.667
insurance	0.614	0.614	0.628	0.593	0.593	0.792
Avg. G-Mean	0.729	0.730	0.721	0.718	0.713	0.799
Avg. Rank	3.444	3.722	3.667	3.889	4.667	1.611

4.4. Complexity and Hyper-parameters

For a dataset D, its size $|D|$ only matters in the classification and evaluation stage which complexity is $O(|D| \times \log T)$, where T is the height of DT. Besides, the number of attribute of D only affects the input and output layer of q-network and output layer of actor-network. The q-network and the second stage of actor-network are both kinds of multi-layer fully connected neural network, which complexity is $O\left(\sum_{l=1}^{L}(S_{l-1} \times S_l)\right)$, where L is the number of hidden layer, S_{l-1} is the size of the $l-1$th layer and S_l is the size of the lth layer. The first stage of actor-network is a tree-based convolution layer which is implemented by the 1D convolution and its complexity is $O\left(\sum_{l=1}^{L}(M_l * K_l * C_l * C_{l-1})\right)$, where L is the number of layer, C_{l-1} is the input chanels, C_l is the output chanels, M is the size of feature map and K is the kernel size. In our experiments, the hyper-parameters ε affects the agent's exploration of the action space. The initial value of ε is set to 1 and monotonically decreases with the number of iterations, and its lower limit is set to 0.1. The agent's exploration of action spaces of different sizes (of attributes) can be controlled by adjusting the rate of ε's descent. The discount factor is set to 0.9 that makes the agent pay more attention to future rewards.

Table 4: AUC of six methods on eighteen data sets

Data Sets	C4.5	HDDT	CCPDT	iHD	iHDw	ours
ecoli-0-1-vs-2-3-5	0.854	0.875	0.868	0.861	0.854	0.966
ecoli-0-1-4-6-vs-5	0.894	0.799	0.805	0.805	0.894	0.917
ecoli-0-1-4-7-vs-2-3-5-6	0.721	0.774	0.673	0.803	0.803	0.803
ecoli-0-6-7-vs-5	0.827	0.898	0.921	0.714	0.786	0.929
ecoli2	0.861	0.872	0.906	0.877	0.848	0.891
haberman	0.555	0.588	0.561	0.574	0.568	0.574
new-thyroid1	0.925	0.892	0.875	0.875	0.875	0.900
new-thyroid2	0.917	1.000	0.867	0.875	0.917	0.958
vehicle3	0.705	0.750	0.714	0.754	0.768	0.770
winequality-red-4	0.503	0.512	0.544	0.537	0.514	0.570
wisconsin	0.922	0.916	0.919	0.916	0.925	0.970
yeast-0-2-5-6-vs-3-7-8-9	0.759	0.750	0.745	0.749	0.726	0.864
yeast1	0.629	0.614	0.655	0.626	0.638	0.655
glass0	0.827	0.739	0.747	0.747	0.744	0.834
glass6	0.842	0.842	0.842	0.834	0.825	0.884
pima	0.677	0.695	0.715	0.727	0.686	0.715
africa recession	0.611	0.580	0.597	0.615	0.525	0.670
insurance	0.667	0.654	0.670	0.645	0.645	0.810
Avg. AUC	0.761	0.764	0.757	0.752	0.752	0.816
Avg. Rank	3.778	3.833	3.500	3.833	4.500	1.556

4.5. Discussion

There are two main properties of DT+DRL that make the combo appealing for imbalance data setting. First, the reward function can be easy to design. For example, in Lin et al. (2020), the reward function was desgined based on the imbalance ratio of datasets. Besides, in our method, the reward function was desgined based on the evaluation metrics in imbalance classification, which can guide the agent correctly during the training process. Second, unlike other methods that use DRL to solve imbalance classification problem, our method can shorten the length of Markov's decision chain by introducing the decision tree structure as a classifier. Because the maximum number of nodes in the decision tree can be specified manually, the Markov decision chain is no longer dependent on the number of samples. The same approach still be applicable to balance data, which only requires the use of accuracy to replace the imbalanced classification evaluation metrics to modify the reward function.

5. Conclusion

This paper comes up with a new decision tree building method via deep reinforcement learning for imbalanced binary classification. First, we apply the tree-based convolution to extract the state information from a tree structure. Second, the attributes and thresholds

are combined into a parameterized action. Third, the typical evaluation metrics of imbalanced classification are used to calculate the reward. Finally, we utilize a prevalent deep reinforcement algorithm named Multi-Pass DQN to find an optimal DT building policy. To compare the proposed method with that of the most advanced decision tree methods, we conduct experiments on more than 15 imbalanced data sets. The experiment results indicate that our method has better performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61662083

References

Pritom Saha Akash, Md. Eusha Kadir, Amin Ahsan Ali, and Mohammad Shoyaib. Inter-node hellinger distance based decision tree. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1967–1973. ijcai.org, 2019. doi: 10.24963/ijcai.2019/272. URL https://doi.org/10.24963/ijcai.2019/272.

Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, and Salvador García. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Multiple Valued Log. Soft Comput.*, 17(2-3):255–287, 2011. URL http://www.oldcitypublishing.com/journals/mvlsc-home/mvlsc-issue-contents/mvlsc-volume-17-number-2-3-2011/mvlsc-17-2-3-p-255-287/.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Craig J. Bester, Steven D. James, and George Dimitri Konidaris. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces. *CoRR*, abs/1905.04388, 2019. URL http://arxiv.org/abs/1905.04388.

Christopher Blake and Eirini Ntoutsi. Reinforcement learning based decision tree induction over data streams with concept drifts. In Xindong Wu, Yew-Soon Ong, Charu C. Aggarwal, and Huanhuan Chen, editors, *2018 IEEE International Conference on Big Knowledge, ICBK 2018, Singapore, November 17-18, 2018*, pages 328–335. IEEE Computer Society, 2018. doi: 10.1109/ICBK.2018.00051. URL https://doi.org/10.1109/ICBK.2018.00051.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002. doi: 10.1613/jair.953. URL https://doi.org/10.1613/jair.953.

Jie Chen, ZhongCheng Wu, and Jun Zhang. Driving safety risk prediction using cost-sensitive with nonnegativity-constrained autoencoders based on imbalanced naturalistic driving data. *IEEE Trans. Intell. Transp. Syst.*, 20(12):4450–4465, 2019. doi: 10.1109/TITS.2018.2886280. URL https://doi.org/10.1109/TITS.2018.2886280.

Qing Chen, Anguo Zhang, Tingwen Huang, Qianping He, and Yongduan Song. Imbalanced dataset-based echo state networks for anomaly detection. *Neural Comput. Appl.*, 32(8): 3685–3694, 2020. doi: 10.1007/s00521-018-3747-z. URL https://doi.org/10.1007/s00521-018-3747-z.

Jianfeng Chi, Guanxiong Zeng, Qiwei Zhong, Ting Liang, Jinghua Feng, Xiang Ao, and Jiayu Tang. Learning to undersampling for class imbalanced credit risk forecasting. In Claudia Plant, Haixun Wang, Alfredo Cuzzocrea, Carlo Zaniolo, and Xindong Wu, editors, *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pages 72–81. IEEE, 2020. doi: 10.1109/ICDM50108.2020.00016. URL https://doi.org/10.1109/ICDM50108.2020.00016.

David A. Cieslak and Nitesh V. Chawla. Learning decision trees for unbalanced data. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part I*, volume 5211 of *Lecture Notes in Computer Science*, pages 241–256. Springer, 2008. doi: 10.1007/978-3-540-87479-9_34. URL https://doi.org/10.1007/978-3-540-87479-9_34.

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom M. Mitchell, and Eric P. Xing. Learning data manipulation for augmentation and weighting. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15738–15749, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/671f0311e2754fcdd37f70a8550379bc-Abstract.html.

Mohammad Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics Decis. Mak.*, 11:51, 2011. doi: 10.1186/1472-6947-11-51. URL https://doi.org/10.1186/1472-6947-11-51.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.02971.

Enlu Lin, Qiong Chen, and Xiaoming Qi. Deep reinforcement learning for imbalanced classification. *Appl. Intell.*, 50(8):2488–2502, 2020. doi: 10.1007/s10489-020-01637-z. URL https://doi.org/10.1007/s10489-020-01637-z.

Wei Liu, Sanjay Chawla, David A. Cieslak, and Nitesh V. Chawla. A robust decision tree algorithm for imbalanced data sets. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 766–777. SIAM, 2010. doi: 10.1137/1.9781611972801.67. URL https://doi.org/10.1137/1.9781611972801.67.

Warwick Masson, Pravesh Ranchod, and George Dimitri Konidaris. Reinforcement learning with parameterized actions. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1934–1940. AAAI Press, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11981.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/mniha16.html.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-2022. URL https://doi.org/10.18653/v1/p16-2022.

Minlong Peng, Qi Zhang, Xiaoyu Xing, Tao Gui, Xuanjing Huang, Yu-Gang Jiang, Keyu Ding, and Zhigang Chen. Trainable undersampling for class-imbalance learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4707–4714. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33014707. URL https://doi.org/10.1609/aaai.v33i01.33014707.

Larry D. Pyeatt. Reinforcement learning with decision trees. In M. H. Hamza, editor, *The 21st IASTED International Multi-Conference on Applied Informatics (AI 2003), February 10-13, 2003, Innsbruck, Austria*, pages 26–31. IASTED/ACTA Press, 2003.

Hongyun Qin, Houpan Zhou, and Jiuwen Cao. Imbalanced learning algorithm based intelligent abnormal electricity consumption detection. *Neurocomputing*, 402:112–123, 2020. doi: 10.1016/j.neucom.2020.03.085. URL https://doi.org/10.1016/j.neucom.2020.03.085.

J. Ross Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986. doi: 10.1023/A:1022643204877. URL https://doi.org/10.1023/A:1022643204877.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Chong Su and Jie Cao. Improving lazy decision tree for imbalanced classification by using skew-insensitive criteria. *Appl. Intell.*, 49(3):1127–1145, 2019. doi: 10.1007/s10489-018-1314-z. URL https://doi.org/10.1007/s10489-018-1314-z.

Bowen Tan, Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Connecting the dots between mle and rl for sequence prediction. *arXiv preprint arXiv:1811.09740*, 2018.

Philip S. Thomas and Emma Brunskill. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *CoRR*, abs/1706.06643, 2017. URL http://arxiv.org/abs/1706.06643.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1995–2003. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/wangf16.html.

Jiechao Xiong, Qing Wang, Zhuoran Yang, Peng Sun, Lei Han, Yang Zheng, Haobo Fu, Tong Zhang, Ji Liu, and Han Liu. Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space. *CoRR*, abs/1810.06394, 2018. URL http://arxiv.org/abs/1810.06394.

Pinar Yildirim. Chronic kidney disease prediction on imbalanced data by multilayer perceptron: Chronic kidney disease prediction. In Sorel Reisman, Sheikh Iqbal Ahamed, Claudio Demartini, Thomas M. Conte, Ling Liu, William R. Claycomb, Motonori Nakamura, Edmundo Tovar, Stelvio Cimato, Chung-Horng Lung, Hiroki Takakura, Ji-Jiang Yang, Toyokazu Akiyama, Zhiyong Zhang, and Kamrul Hasan, editors, *41st IEEE Annual Computer Software and Applications Conference, COMPSAC 2017, Turin, Italy, July 4-8, 2017. Volume 2*, pages 193–198. IEEE Computer Society, 2017. doi: 10.1109/COMPSAC.2017.84. URL https://doi.org/10.1109/COMPSAC.2017.84.