Applied Intelligence
https://doi.org/10.1007/s10489-020-01637-z

Deep reinforcement learning for imbalanced classification

Enlu Lin[1] · **Qiong Chen**[1] · **Xiaoming Qi**[1]

Abstract

Data in real-world application often exhibit skewed class distribution which poses an intense challenge for machine learning. Conventional classification algorithms are not effective in case of imbalanced data distribution, and may fail when the data distribution is highly imbalanced. To address this issue, we propose a general imbalanced classification model based on deep reinforcement learning, in which we formulate the classification problem as a sequential decision-making process and solve it by a deep Q-learning network. In our model, the agent performs a classification action on one sample in each time step, and the environment evaluates the classification action and returns a reward to the agent. The reward from the minority class sample is larger, so the agent is more sensitive to the minority class. The agent finally finds an optimal classification policy in imbalanced data under the guidance of the specific reward function and beneficial simulated environment. Experiments have shown that our proposed model outperforms other imbalanced classification algorithms, and identifies more minority samples with better classification performance.

Keywords Imbalanced classification · Deep reinforcement learning · Reward function · Classification policy

1 Introduction

Imbalanced data classification has been widely researched in the field of machine learning [1–3]. In some real-world classification researches, such as abnormal detection, disease diagnosis, risk behavior recognition, etc., the distribution of data across different classes is highly skewed. The instances in one class (e.g., cancer patient) can be 1000 times less than that in another class (e.g., healthy people). Most machine learning algorithms are suitable for balanced training data set. When facing imbalanced scenarios, these models often provide good recognition results to the majority instances, whereas the minority instances are distorted. The instances in the minority class are difficult to detect because of their infrequency and casualness; however, misclassifying minority class instances will usually result in heavy costs.

A range of imbalanced data classification algorithms were developed during the past two decades. The methods to tackle these issues are generally divided into two groups [4]: the data level and the algorithmic level. The former group modifies the collection of instances to balance the class distribution by re-sampling the training data, which often represents as different types of data manipulation techniques. The latter group modifies the existing learners to alleviate their bias towards the majority class, which often assigns higher misclassification cost to the minority class. However, with the rapid developments of big data, a large amount of complex data with high imbalanced ratio is being generated which brings enormous challenges to imbalanced data classification. Conventional methods are inadequate to cope with more and more complex data so that novel deep learning approaches are becoming increasingly popular.

In recent years, deep reinforcement learning has been successfully applied to computer gaming, robot control, recommendation systems [5–7] and so on. For classification problems, deep reinforcement learning has served in eliminating noisy data and learning better features [8, 9], and has made great improvements in classification performance. However, there was seldom any research work on applying deep reinforcement learning to imbalanced data learning. In fact, deep reinforcement learning is

✉ Qiong Chen
csqchen@scut.edu.cn

Enlu Lin
linenus@outlook.com

Xiaoming Qi
qxmscut@126.com

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China

Published online: 10 March 2020

ideally suitable for imbalanced data classification as its learning mechanism and specific reward function. Deep reinforcement learning can pay more attention to the minority class by giving higher rewards to them. A deep Q-learning network (DQN) based model for imbalanced classification is proposed in this paper, in which the imbalanced classification problem is regarded as a guessing game that can be decomposed into a sequential decision-making process. In each time step, the agent receives an environment state which is represented by a training sample and then performs a classification action under the guidance of a policy. If the agent performs a correct classification action, it will be given a positive reward; otherwise, it will be given a negative reward. The reward from the minority class is higher than that of the majority class. The goal of the agent is to obtain as many more cumulative rewards as possible during the process of sequential decision-making, that is, to correctly recognize the samples as much as possible. This paper mainly focuses on the binary classification and the experimental data sets are binary-class ones.

The contributions of this paper can be summarized as follows: 1) To formulate the classification problem as a sequential decision-making process and propose a deep reinforcement learning framework for imbalanced classification. 2) To design and implement the DQN based imbalanced classification model (DQNimb), which mainly includes building the simulation environment, defining the interaction rules between agent and environment, and designing the specific reward function. 3) To study the performance of our model through experiments and compare it with other methods of imbalanced data learning.

The rest of this paper is organized as follows: The second section introduces the research methodology of imbalanced data classification and the applications of deep reinforcement learning on classification problems. The third section elaborates the proposed model and analyzes it theoretically. The fourth section shows the experimental results and evaluates the performance of our method compared with other methods. The last section summarizes the work of this paper and looks forward to future work.

2 Related work

In the past decades, many research efforts have been devoted to tackle the class imbalance problem. This section mainly introduces the previous work on imbalanced data classification and the deep reinforcement learning relative classification methods. These studies inspire us to extend reinforcement learning onto imbalanced data classification.

2.1 Imbalanced data classification

The previous researches in imbalanced data classification concentrate mainly on two levels: the data level [10–14] and the algorithmic level [15–25]. The data level methods aim to balance the class distribution by manipulating the training samples, including over-sampling the minority class, under-sampling the majority class and the combinations of the two above methods [13, 26]. SMOTE [11] is a well-known over-sampling method, which generates new samples by linear interpolation between adjacent minority samples. NearMiss [12] is a typical under-sample method based on the nearest neighbor algorithm. However, over-sampling can potentially lead to overfitting while under-sampling may lose valuable information on the majority class. The algorithmic level methods aim to lift the importance of the minority class by improving the existing algorithms, including cost-sensitive learning, ensemble learning and decision threshold adjustment. The cost-sensitive learning methods assign various misclassification costs to different classes by modifying the loss function, in which the misclassification cost of minority class is higher than that of the majority class. The ensemble learning based methods train multiple individual sub-classifiers, and then use voting or combining to get better results. The threshold-adjustment methods train the classifier in original imbalanced data and change the decision threshold in test time. A number of deep learning based methods have recently been proposed for imbalanced data classification [27–31]. Wang et al. [27] proposed a new loss function in deep neural network which can capture classification errors from both the majority class and minority class equally. Huang et al. [28] studied a method that learns more discriminative features of imbalanced data by maintaining both inter-cluster and inter-class margins. Yan et al. [29] used a bootstrapping sampling algorithm which ensures the training data in each mini-batch for convolutional neural network is balanced. A method to optimize the network parameters and the class-sensitive costs jointly was presented in [30]. In [31] Dong et al. mined hard samples in the minority classes and improved the algorithm by batch-wise optimization with class rectification loss function.

2.2 Reinforcement learning for classification problem

Deep reinforcement learning has recently achieved excellent results in classification tasks as it can assist classifiers to learn advantageous features or select high-quality instances from noisy data. In [32], the classification task was constructed into a sequential decision-making process, which uses multi-

ple agents to interact with the environment in order to learn the optimal classification policy. However, the intricate simulation between agents and the environment causes extremely high time complexity. Feng et al. [8] proposed a deep reinforcement learning based model to learn the relationship classification in noisy text data. The model is divided into instance selector and relational classifier. The instance selector selects a high-quality sentence from those noisy data under the guidance of the agent while the relational classifier learns better performance from selected clean data and feeds back a delayed reward to the instance selector. The model finally obtains a better classifier as well as a high-quality data set. The work in [26, 33–35] utilized deep reinforcement learning to learn advantageous features of training data in their respective applications. In general, the advantageous features improve the classifier while the better classifier feeds back a higher reward which encourages the agent to select more advantageous features. Martinez et al. [9] proposed a deep reinforcement learning framework for time series data classification in which the definition of specific reward function and the Markov process are clearly formulated. Researches in imbalanced data classification with reinforcement learning were quite limited. In [36] an ensemble pruning method was presented, which selects the best sub-classifier by using reinforcement learning. However, this method was merely suitable for traditional small data sets because it was inefficient to select classifiers when there were plenty of sub-classifiers. Incorporating classifiers with deep reinforcement learning has shown promising results in many applications. It is inspiring to explore the performances of deep reinforcement learning when applied on imbalanced classification. In this paper, we propose a deep Q-network based model for imbalanced classification, and test it on complex high-dimensional data such as images and texts.

3 Methodology

The imbalanced data classification task can be resolved into a sequential decision-making problem. This section describes the details of the Imbalanced Classification Markov Decision Process (ICMDP) framework and formulated the DQN-based imbalanced classification model in theory. In the model, different reward values are given to the minority class and the majority class, the classification policy is learned by deep Q-learning network.

3.1 Imbalanced classification Markov decision process

Reinforcement learning algorithms that incorporate deep learning have defeated world champions at the game of Go as well as human experts playing numerous Atari video games. We now regard classification problem as a guessing game, in which the agent receives a sample in each time step and guesses (classifies) which category the sample belongs to, and then the environment returns an immediate reward as well as the next sample, as shown in Fig. 1. A positive reward is given to the agent by the environment when the agent correctly guesses the category of sample; otherwise a negative reward is given. When the agent learns an optimal behavior from its interaction with the environment in order to get the maximum accumulative rewards, it will gradually be able to correctly classify samples as much as possible.

Now we formalize the Imbalanced Classification Markov Decision Process framework into a sequential decision-making problem. Assume that the imbalanced training data set is $D = \{(x_1, l_1), (x_2, l_2), ..., (x_n, l_n)\}$ where x_i is the ith sample and l_i is the label of the ith sample. We propose to train a classifier as an agent evolving in ICMDP where:

- **State** $\mathcal{S}$: The state of environment is determined by the training sample. At the beginning of training, the agent receives the first sample x_1 as its initial state s_1. The state s_t of environment at each time step corresponds to the sample x_t. When a new episode begins, environment shuffles the order of samples in training data set.
- **Action** $\mathcal{A}$: The action of agent is associated with the label of the training data set. The action a_t taken by agent is to predict a class label. For binary classification problem, $\mathcal{A} = \{0, 1\}$ where 0 represents the minority class and 1 represents the majority class.

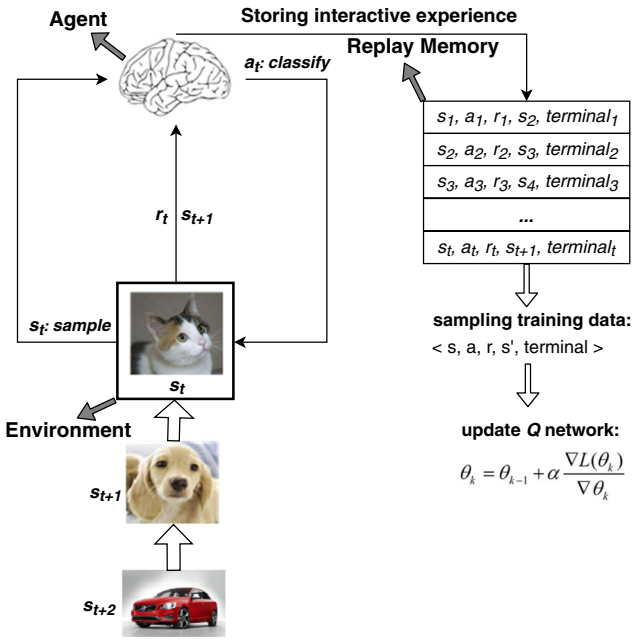

Fig. 1 Overall process of ICMDP

- **Reward** $\mathcal{R}$: A reward r_t is the feedback from the environment through which we measure the success or failure of an agent's actions. In order to guide the agent to learn the optimal classification policy in imbalanced data, the absolute reward value of sample in the minority class is higher than that in the majority class. That is, when the agent correctly or incorrectly recognizes the minority class sample, the environment will feed back to the agent with a larger reward or punishment.
- **Transition probability** $\mathcal{P}$: Transition probability $p(s_{t+1}|s_t, a_t)$ in ICMDP is deterministic. The agent moves from the current state s_t to the next state s_{t+1} according to the order of samples in the training data set.
- **Discount factor** γ: $\gamma \in [0, 1]$ is to balance the immediate and future reward.
- **Episode**: Episode in reinforcement learning is a transition trajectory from the initial state to the terminal state $\{s_1, a_1, r_1, s_2, a_2, r_2, ..., s_t, a_t, r_t\}$. An episode ends when all samples in training data set are classified or when the agent misclassifies a sample from the minority class.
- **Policy** π_θ: The policy π_θ is a mapping function $\pi : \mathcal{S} \rightarrow \mathcal{A}$ where $\pi_\theta(s_t)$ denotes the action a_t performed by the agent in state s_t. The policy π_θ in ICMDP can be considered as a classifier with the parameter θ.

With the definitions and notations above, the imbalanced classification problem is formally defined as finding an optimal classification policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$, which maximizes the cumulative rewards in ICMDP.

3.2 Reward function for imbalanced data classification

The minority class samples are difficult to be identified correctly in the imbalance data set. In order to better recognize the minority class samples, the algorithm should be more sensitive to the minority class. A large reward or punishment is returned to the agent when it meets a minority sample, the reward function is defined as follows:

$$R(s_t, a_t, l_t) = \begin{cases} +1, & a_t = l_t \text{ and } s_t \in D_P \\ -1, & a_t \neq l_t \text{ and } s_t \in D_P \\ \lambda, & a_t = l_t \text{ and } s_t \in D_N \\ -\lambda, & a_t \neq l_t \text{ and} s_t \in D_N \end{cases} \tag{1}$$

where $\lambda \in [0, 1]$, D_P is the minority class sample set, D_N is the majority class sample set, l_t is the class label of the sample in state s_t. Assume the reward value be 1 or -1 when the agent correctly or incorrectly classifies a minority class sample, be λ or $-\lambda$ when the agent correctly or incorrectly classifies a majority class sample.

The value of reward function is the prediction cost of the agent. For imbalanced data set, the prediction cost values of the minority class are higher than that of the majority class, so the agent is more sensitive to the minority class. If the class distribution of training data set is balanced, then $\lambda = 1$, the prediction cost values are the same for all classes. In fact, λ is a trade-off parameter to adjust the importance of the majority class. Our model achieves the best performance in experiment when λ is equal to the imbalanced ratio $\rho = \frac{|D_P|}{|D_N|}$. We will discuss it in Section 4.8.

3.3 DQN based imbalanced classification algorithm

3.3.1 Deep *Q*-learning for ICMDP

In ICMDP, the classification policy π is a function which receives a sample and returns the probabilities of all labels.

$$\pi(a|s) = P(a_t = a|s_t = s) \tag{2}$$

The goal of the classifier agent is to correctly recognize the samples in the training data as much as possible. As the classifier agent can get a positive reward when it correctly recognizes a sample, it can achieve its goal by maximizing the cumulative rewards g_t:

$$g_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{3}$$

In reinforcement learning, there is a function that calculates the quality of a state-action combination, called the Q function:

$$Q^\pi(s, a) = E_\pi[g_t|s_t = s, a_t = a] \tag{4}$$

According to the Bellman equation [37], the Q function can be expressed as:

$$Q^\pi(s, a) = E_\pi[r_t + \gamma Q^\pi(s_{t+1}, a_{t+1})|s_t = s, a_t = a] \tag{5}$$

The classifier agent can maximize the cumulative rewards by solving the optimal Q^* function, and the greedy policy under the optimal Q^* function is the optimal classification policy π^* for ICMDP.

$$\pi^*(a|s) = \begin{cases} 1, & \text{if } a = \arg\max_a Q^*(s, a) \\ 0, & \text{else} \end{cases} \tag{6}$$

$$Q^*(s, a) = E_\pi[r_t + \gamma \max_a Q^*(s_{t+1}, a_{t+1})|s_t = s, a_t = a] \tag{7}$$

In the low-dimensional finite state space, Q functions are recorded by a table. However, in the high-dimensional continuous state space, Q functions cannot be resolved until deep Q-learning algorithm was proposed, which fits the Q function with a deep neural network. In deep Q-learning algorithm, the interaction data (s, a, r, s') obtained from (7) are stored in the experience replay memory M. The agent randomly samples a mini-batch of transitions B from M and performs a gradient descent step on the Deep Q network according to the loss function as follow:

$$L(\theta_k) = \sum_{(s,a,r,s')\in B} (y - Q(s, a; \theta_k))^2 \tag{8}$$

where y is the target estimate of the Q function, the expression of y is:

$$y = r + (1 - t)\gamma \max_{a'} Q(s', a'; \theta_{k-1}) \tag{9}$$

where s' is the next state of s, a' is the action performed by agent in state s', t=1 if $terminal$=True; otherwise t=0.

The derivative of loss function (8) with respect to θ is:

$$\frac{\nabla L(\theta_k)}{\nabla \theta_k} = -2 \sum_{(s,a,r,s')\in B} (y - Q(s, a; \theta_k)) \frac{\nabla Q(s, a; \theta_k)}{\nabla \theta_k} \tag{10}$$

Now we can obtain the optimal Q^* function by minimizing the loss function (8), and the greedy policy under the optimal Q^* function will get the maximum cumulative rewards. So the optimal classification policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ for ICMDP is achieved.

3.3.2 Influence of reward function

In imbalanced data, the trained Q-network will be biased toward the majority class. However, due to the aforementioned reward function (1), it assigns different rewards for different classes and ultimately makes the samples from different classes having the same impact on Q-network.

Suppose the positive and negative samples are denoted as s^+ and s^-, their target Q values are represented as y^+ and y^-. According to (1) and (9), the target Q value of positive and negative samples is expressed as:

$$y^+ = (-1)^{1-I(a=l)} + (1 - t)\gamma \max_{a'} Q(s', a') \tag{11}$$

$$y^- = (-1)^{1-I(a=l)}\lambda + (1 - t)\gamma \max_{a'} Q(s', a') \tag{12}$$

where $I(x)$ is an indicator function.

Let P be the total number of positive samples, N be the total number of negative samples. Substituting (11) and (12) into (10), we get the following:

$$\begin{aligned}\frac{\nabla L(\theta_k)}{\nabla \theta_k} = & -2 \sum_{m=1}^{P+N} ((1 - t_m)\gamma \max_{a'_m} Q(s'_m, a'_m; \theta_{k-1}) \\ & - Q(s_m, a_m; \theta_k)) \frac{\nabla Q(s_m, a_m; \theta_k)}{\nabla \theta_k} \\ & -2 \sum_{i=1}^{P} (-1)^{1-I(a_i=l_i)} \frac{\nabla Q(s_i, a_i; \theta_k)}{\nabla \theta_k} \\ & -2\lambda \sum_{j=1}^{N} (-1)^{1-I(a_j=l_j)} \frac{\nabla Q(s_j, a_j; \theta_k)}{\nabla \theta_k}\end{aligned} \tag{13}$$

In (13), the second item relates to the minority class and the third item relates to the majority class. For imbalanced data set ($N > P$), if $\lambda = 1$, the immediate rewards of the two classes are identical, the value of the third item is larger than that of the second item because the amount of samples in the majority class are much more than that in the minority class. So the model is biased to the majority class. If $\lambda < 1$, λ can reduce the immediate rewards of negative samples and weaken their impact on the loss function of Q-network. What's more, the second item has the same value as the third item when λ is equal to the imbalanced ratio ρ.

3.3.3 Training details

We construct the simulation environment according to the definition of ICMDP. The architecture of the Q network depends on the complexity and amount of the training data set. The input of the Q network is consistent with the structure of training samples, and the number of outputs is equal to the number of sample categories. In fact, the Q network is a neural network classifier without the final softmax layer. The training process of Q network is described in Algorithm 2. In an episode, the agent uses the ϵ-greedy policy to pick the action, and then obtains the reward from the environment through the REWARD function in Algorithm 1. The deep Q-learning algorithm will be running about 120000 iterations (updates of network parameters θ).The converged Q network which then adds a softmax layer can be regarded as a neural network classifier trained by imbalanced data.

Algorithm 1 Environment simulation.

D_P represents the minority class sample set.
Function *REWARD*($a_t \in \mathcal{A}, l_t \in L$)
  Initialize $terminal_t$=False
  if $s_t \in D_P$ **then**
    if $a_t = l_t$ **then**
      Set r_t=1
    else
      Set r_t=-1
      $terminal_t$=True
  else
    if $a_t = l_t$ **then**
      Set $r_t = \lambda$
    else
      Set $r_t = -\lambda$
  return $r_t, terminal_t$

Algorithm 2 Training.

Input: Data $D = \{(x_1, l_1), (x_2, l_2), ..., (x_T, l_T)\}$.
  Episode number K.
Initialize experience replay memory M
Randomly initialize parameters θ and ϕ
Initialize simulation environments ε
for $episode\ k = 1$ *to* K **do**
  Shuffle the training data D
  Initialize state $s_1 = x_1$
  for $t = 1$ *to* T **do**
    Choose an action based ϵ-greedy policy:
    $a_t = \pi_\theta(s_t)$
    $r_t, terminal_t = STEP(a_t, l_t)$
    Set $s_{t+1} = x_{t+1}$
    Store $(s_t, a_t, r_t, s_{t+1}, terminal_t)$ to M
    Randomly sample
    $(s_j, a_j, r_j, s_{j+1}, terminal_j)$ from M
    Set $y_j =$
    $\begin{cases} r_j, & terminal_j\text{=True} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \phi), & terminal_j\text{=False} \end{cases}$
    Perform a gradient descent step on $L(\theta)$ w.r.t.
    θ: $L(\theta) = (y_j - Q(s_j, a_j; \theta))^2$
    if $terminal_t$=*True* **then**
      break
    Update parameters $\phi := (1 - \eta)\phi + \eta\theta$

4 Experiment

We compared our DQNimb model with the other imbalanced learning methods and conducted an empirical study to evaluate the DQNimb model with different level of imbalance ratio. In order to make a fair and comprehensive comparison, Friedman test was conducted to show that the DQNimb model significantly outperforms the other methods. We also made an exploration on reward function to study the effect of different values of reward on the classification performances.

4.1 Comparison methods

Seven imbalanced learning methods was compared from the data level and the algorithmic level, including sampling techniques, and cost-sensitive learning methods, decision threshold adjustment method and deep imbalanced learning methods. A deep neural network trained with cross entropy loss function will be used as baseline in the experiments. The comparison methods are shown as follows:

- **DNN**: A method which trains the deep neural network using cross entropy loss function without any improvement strategy in imbalanced data set.
- **ROS**: A re-sampling method to build a more balanced data set through over-sampling the minority class by random replication [10].
- **RUS**: A re-sampling method to build a more balanced data set through under-sampling the majority class by random sample removal [10].
- **MFE**: A method to improve the classification performance of deep neural network in imbalanced data sets by using mean false error loss function [27]
- **CSM**: A cost sensitive method which assigns greater misclassification cost to the minority class and smaller cost to the majority class in loss function [21]
- **DTA**: A method to train the deep neural network in imbalanced data and to adjust the model decision threshold in test time by incorporating the class prior probability [23]
- **FL**: A method to reshape the standard cross entropy loss (called Focal Loss) by down-weighting the loss assigned to well-classified examples. [38]
- **CRL**: A method using batch-wise mining of hard sample on the minority class and formulate a Class Rectification Loss for the minority class incremental rectification [31]

The DNN was used as the base model for the imbalance learning approaches. For example, ROS and RUS used DNN as classification model by re-sampling the data through over-sampling the minority class or under-sampling the majority class. CSM trained the DNN model by assigning varying misclassification costs to different classes where the cost representation used the inverse class frequency. DTA trained the DNN model and adjusted the model decision threshold in test time. MFE, FL and CRL are several outstanding imbalanced learning approaches incorporate with deep learning in recent years. MFE and FL reshaped the cross entropy loss of DNN model with different way and CRL performs hard sample mining for the minority

Table 1 Dataset of experiments

Dataset	Dimension of sample	Imbalance ratio ρ	Training data		Test data	
			Positive samples	Negative samples	Positive samples	Negative samples
IMDB	1*500	10%	1250	12000	12500	12500
		5%	625			
		2%	250			
Cifar-10(1)	32*32*3	4%	400	10000	1000	2000
		2%	200			
		1%	100			
		0.5%	50			
Cifar-10(2)		4%	800	20000	1000	4000
		2%	400			
		1%	200			
		0.5%	100			
Fashion-Mnist(1)	28*28*1	4%	480	12000	2000	2000
		2%	240			
		1%	120			
		0.5%	60			
Fashion-Mnist(2)		4%	720	18000	3000	3000
		2%	360			
		1%	180			
		0.5%	90			
Mnist	28*28*1	1%	540	54042	1032	8968
		0.2%	108			
		0.1%	54			
		0.05%	27			

class during the training of the DNN model. The network architecture and parameters of DNN in above methods are the same for fair and conclusive comparisons.

4.2 Evaluation metrics

In our experiment, in order to evaluate the classification performance in imbalanced data sets more reasonably, G-mean and F-measure metrics [39] are adopted. G-mean is the geometric mean of recall and specificity and F-measure represents the geometric mean of recall and precision. The higher the G-mean score and F-measure score are, the better the algorithm performs. The formulae of G-mean and F-measure are shown as follows:

$$G-mean = \sqrt{Recall * Specificity} \tag{14}$$

$$F-measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{15}$$

where $Recall = \frac{TP}{TP+FN}$, $Specificity = \frac{TN}{TN+FP}$ and $Precision = \frac{TP}{TP+FP}$. TP is the number of true positive samples, TN means the number of true negative samples, FP denotes the number of false positive samples, and FN indicates the number of false negative samples.

4.3 Dataset

In this paper, we mainly study the binary imbalanced classification problem with deep reinforcement learning. We perform experiments on IMDB, Cifar-10, Mnist and Fashion-Minist. Our approach is evaluated on the deliberately imbalanced splits. Our data sets used for the experiments are shown in Table 1.

IMDB is a text dataset, which contains 50000 movie reviews labeled by sentiment (positive/negative). Reviews have been preprocessed, and each review is encoded as a sequence of word indexes. The standard train/test split for each class is 12500/12500. The positive reviews are regarded as the positive class in our experiment.

Mnist is a simple image dataset. It consists of 28 × 28 grayscale images. There are 10 classes corresponding to

Retrieve the data sets from https://keras.io/datasets

Table 2 Network architecture used for text dataset

Layer	Input	Output
Embedding	500	(500,128)
LSTM	(500,128)	128
FullyConnected	128	64
ReLU	64	64
FullyConnected	64	2
Softmax	2	2

digits from 0 to 9. The number of train/test samples per class is almost 6000/1000. We let the images with label 2 as the positive class and the rest images as the negative class in our experiment.

Fashion-Mnist is a new dataset comprising of 28 × 28 grayscale images of 70000 fashion products with 10 categories. It is designed to serve as a direct drop-in replacement for the original Mnist dataset. The training dataset has 6000 images per class while the test dataset has 1000 images per class. To evaluate our algorithm on various scales of datasets, two simulated data sets of different sizes are extracted from this dataset. The first one chooses the images labeled by 0,2 (T-Shirt, Pullover) as the positive class and the images labeled by 1,3 (Trouser, Dress) as the negative class. The second one chooses the images labeled by 4,5,6 (Coat, Sandal, Shirt) as the positive class and the images labeled by 7,8,9 (Sneaker, Bag, Ankle boot) as the negative class.

Cifar-10 is a more complex image dataset than Fashion-Mnist. It contains 32x32 color images with 10 classes of natural objects. The standard train/test split for each class is 5000/1000. There are two simulated data sets of different sizes are extracted from this dataset. The first one chooses the images labeled by 1 (automobile) as the positive class and the images labeled by 3,4,5,6 (cat, deer, dog, frog) as the negative class. The other one takes the images labeled by 7 (horse) as the positive class and the images labeled by 8,9 (ship, truck) as the negative class.

The training dataset with different imbalance levels are obtained by reducing the number of positive class to $\rho \times N$ where N is the total number of negative class and ρ is imbalanced ratio of dataset. The detailed description of experiment dataset is shown in Table 1.

4.4 Network architecture

We use deep neural network to learn the feature representation from the imbalanced and high dimensional datasets. For the compared algorithms, the network architecture used for text (IMDB) dataset has a embedding layer, a long-short term memory (LSTM) layer, two fully connected layers and a softmax output layer. The detailed parameters are given in Table 2. The network architecture that is used for image (Mnist, Fashion-Mnist,Cifar-10) classification has two convolution layers (each of them is followed by a ReLu and max-pooling layer), two fully connected layers and a softmax output layer. Its detailed parameters are given in Table 3. For our model, the Q-network architecture is similar to the network structure of compared algorithms, but the final softmax output layer is removed because it does not need to scale the Q value of different actions between 0 and 1.

4.5 Parameter setting

In DQNimb model, ϵ-greedy policy is used for exploration, the probability of exploration ϵ is linearly attenuated from 1.0 to 0.01. The target network update factor η

Table 3 Network architecture used for image dataset

Layer	Width	Height	Depth	Kernel size	Stride
Input	28(32)	28(32)	1(3)	–	–
Convolution	28(32)	28(32)	32	5	1
ReLU	28(32)	28(32)	32	–	–
MaxPooling	14(16)	14(16)	32	2	2
Convolution	14(16)	14(16)	32	5	1
ReLU	14(16)	14(16)	32	–	–
MaxPooling	7(8)	7(8)	32	2	2
Flatten	1	1	1568(2048)	–	–
FullyConnected	1	1	256	–	–
ReLU	1	1	256	–	–
FullyConnected	1	1	2	–	–
Softmax	1	1	2	–	–

is 0.05. The size of experience replay memory is 50 000 and the interactions between agent and environment are approximately 120 000 steps. The discount factor of immediate reward γ is 0.1. Adam algorithm is used to optimize the parameters of Q-network and its learning rate is 0.00025. For other algorithms, DNN is used as a base classifier, the optimizer is Adam and its learning rate is 0.0005, the batch size is 64. We randomly select 10% samples of training data as the verification data and use early stopping technique [40] to monitor the validation loss during the training of deep neural network.

4.6 Experiment results

Before the research of imbalanced data learning, we compare our DQNimb model to the DNN which is a supervised deep learning model in balanced data sets. The experiments were conducted on the six data sets (the imbalance ratio ρ is 1). The number of positive samples and negative samples are equal, so the reward function of the DQNimb model assigns the same reward or punishment to the positive and negative samples. For fairness and convincing comparisons, the network architecture of the DNN model is the same as the Q network architecture of the DQNimb model. The G-mean scores and F-measure scores of the experimental results are shown in Table 4. The DQNimb model obtains the optimal classification strategy by maximizing the cumulative rewards in the Markov process, while the DNN gets the optimal network parameters by minimizing the cross-entropy loss function, both models demonstrate good performance in experimental results. The G-mean scores and F-measure scores of the DQNimb model are slightly better than those of the DNN model.

Assume the number of the negative samples in the imbalanced data set is N, we randomly select $\rho \times N$ positive samples according to the imbalance ratio ρ, and conduct 23 experiments. We report the G-mean scores of our method and the other methods on the different imbalanced data sets in Table 5. Each training was repeated 5 times on the same data set. The results of data sampling methods, cost-sensitive learning methods, threshold adjustment method and deep imbalanced learning methods are much better than DNN model in imbalanced classification problems, however, our model DQNimb achieves an outstanding performance with an overwhelming superiority. In the IMDB text dataset, the G-mean scores of our method DQNimb are normally 5% higher than the second-ranked method CRL and significantly better than that of other methods.

We report the F-measure scores of different algorithms in Table 6. As is shown in Fig. 2, with the increase of data imbalance level, the F-measure scores of each algorithm demonstrate a significant decline. The DNN model suffers the most serious declination, that is, DNN can hardly identify any minority class sample when the data distribution is extremely skewed. Meanwhile, our DQNimb model has the smallest decrease because our algorithm possesses both the advantages of the data level models and the algorithmic level models. In the data level, our model DQNimb has an experience replay memory of storing interactive data during the learning process. When the model misclassifies a positive sample, the current episode will be terminated, which can alleviate the skewed distribution of the samples in the experience replay memory. In the algorithmic level, the DQNimb model gives a higher reward or penalty for positive samples, which raises the attention to the samples in the minority class and increases the probabilities of positive samples being correctly identified.

Table 4 Experiment results on balanced datasets

Dataset (balanced)	G-mean		F-measure	
	DNN	DQNimb	DNN	DQNimb
IMDB	0.873	**0.874**	0.872	**0.875**
Cifar-10(1)	0.962	**0.967**	0.941	**0.950**
Cifar-10(2)	0.959	**0.963**	0.946	**0.952**
Fashion-Mnist(1)	0.978	**0.984**	0.978	**0.984**
Fashion-Mnist(2)	0.990	**0.991**	0.990	**0.991**
Mnist	0.995	**0.997**	0.985	**0.992**

4.7 Friedman test

Friedman test was conducted according to the recommendation in [41]. We performed the pairwise post-hoc analysis recommended by Benavoli et al. [42] where the average rank comparison is replaced by a Wilcoxon signed-rank test with Holm's alpha (5%) correction [43, 44]. The results of the post-hoc tests can be visually represented with a critical difference diagram proposed in [41], which shows the average ranks of multiple classifiers over multiple data sets and summarise a significance test between the ranks. The horizontal black bars are cliques; if two classifiers are in the same clique, their ranks are not significantly different; if they are not in the same clique, they are significantly different. Figure 3 shows the results of the analysis of the data from Table 5. The DQNimb model significantly

Table 5 G-mean score of experiment results

Dataset	Imbalance ratio ρ	DQNimb (Ours)	Baseline (DNN)	MFE loss (MFE)	Over sampling (ROS)	Under sampling (RUS)	Cost sensitive (CSM)	Threshold-Adjustment (DTA)	Focal loss (FL)	CRL loss (CRL)
	10%	**0.832**	0.587	0.697	0.702	0.749	0.758	0.691	0.771	**0.782**
	5%	**0.791**	0.419	0.634	0.631	0.643	0.706	0.618	0.719	**0.731**
IMDB	2%	**0.695**	0.134	0.371	0.413	0.542	0.588	0.395	0.631	**0.643**
	4%	**0.956**	0.869	0.939	**0.947**	0.945	0.944	0.946	0.941	0.938
	2%	**0.941**	0.824	0.908	0.925	**0.929**	0.922	0.928	0.916	0.921
	1%	**0.917**	0.730	0.859	0.897	0.896	0.884	**0.912**	0.902	0.906
Cifar-10(1)	0.5%	**0.890**	0.579	0.759	0.838	0.866	0.853	**0.901**	0.869	0.876
	4%	**0.925**	0.815	0.882	0.904	0.906	0.911	0.915	0.908	**0.917**
	2%	**0.917**	0.758	0.852	0.894	0.887	0.886	**0.908**	0.898	0.903
	1%	**0.883**	0.677	0.769	0.854	0.859	0.850	0.873	0.864	**0.875**
Cifar-10(2)	0.5%	**0.829**	0.513	0.693	0.792	0.822	0.816	0.821	0.811	**0.824**
	4%	**0.971**	0.921	0.960	0.962	0.957	0.964	0.964	0.965	**0.967**
	2%	**0.966**	0.885	0.947	0.957	0.953	0.956	0.962	**0.963**	0.962
	1%	**0.959**	0.853	0.934	0.948	0.943	0.946	0.952	**0.957**	0.955
Fashion-Mnist(1)	0.5%	**0.950**	0.757	0.901	0.927	0.934	0.924	**0.944**	0.941	0.938
	4%	**0.985**	0.951	0.968	0.972	0.967	0.973	0.977	0.975	**0.978**
	2%	**0.982**	0.926	0.960	0.963	0.956	0.966	0.970	0.968	**0.973**
	1%	**0.979**	0.872	0.940	0.949	0.946	0.958	**0.962**	0.954	0.960
Fashion-Mnist(2)	0.5%	**0.972**	0.821	0.912	0.935	0.937	0.950	0.953	0.945	**0.956**
	1%	**0.991**	0.967	**0.982**	0.981	0.978	**0.982**	0.978	**0.982**	0.981
	0.2%	**0.983**	0.923	0.949	0.944	0.953	0.951	0.961	**0.971**	0.969
	0.1%	**0.968**	0.856	0.921	0.911	0.929	0.942	0.937	0.947	**0.949**
Mnist	0.05%	**0.941**	0.694	0.842	0.858	0.907	0.921	0.916	0.927	**0.932**

The $1^{st}/2^{nd}$ best results are indicated in red/blue

Table 6 F-measure score of experiment results

Dataset	Imbalance ratio ρ	DQNimb (Ours)	Baseline (DNN)	MFE loss (MFE)	Over sampling (ROS)	Under sampling (RUS)	Cost sensitive (CSM)	Threshold-Adjustment (DTA)	Focal loss (FL)	CRL loss (CRL)
	10%	**0.826**	0.474	0.656	0.647	0.740	0.741	0.640	0.747	**0.762**
	5%	**0.773**	0.194	0.558	0.569	0.611	0.632	0.534	0.677	**0.706**
IMDB	2%	**0.664**	0.012	0.175	0.231	0.476	0.534	0.254	0.591	**0.617**
	4%	**0.915**	0.847	0.905	**0.909**	0.879	0.910	0.904	0.901	0.894
	2%	**0.899**	0.811	0.883	**0.891**	0.854	0.889	0.887	0.882	0.875
	1%	**0.869**	0.689	0.830	**0.857**	0.818	0.851	0.852	0.841	0.849
Cifar-10(1)	0.5%	**0.833**	0.567	0.736	0.817	0.713	0.825	**0.838**	0.819	0.828
	4%	**0.906**	0.797	0.877	0.890	0.866	0.892	**0.895**	0.884	0.893
	2%	**0.886**	0.727	0.834	0.871	0.840	**0.872**	0.867	0.869	**0.872**
	1%	**0.857**	0.601	0.808	**0.841**	0.806	0.833	0.839	0.825	0.834
Cifar-10(2)	0.5%	**0.812**	0.383	0.743	0.774	0.753	0.701	0.798	0.791	**0.803**
	4%	**0.973**	0.916	0.958	0.961	0.956	0.962	0.962	**0.965**	**0.965**
	2%	**0.969**	0.882	0.939	0.955	0.950	0.956	0.955	0.958	**0.961**
	1%	**0.963**	0.843	0.928	0.945	0.941	0.933	0.949	0.947	**0.951**
Fashion-Mnist(1)	0.5%	**0.956**	0.749	0.899	0.927	0.932	0.920	**0.940**	0.935	0.937
	4%	**0.985**	0.951	0.968	0.972	0.967	0.973	**0.977**	0.975	**0.977**
	2%	**0.982**	0.930	0.959	0.962	0.956	0.970	0.971	**0.972**	**0.972**
	1%	**0.979**	0.863	0.938	0.949	0.945	0.957	**0.961**	**0.961**	0.959
Fashion-Mnist(2)	0.5%	**0.971**	0.809	0.909	0.935	0.931	0.949	0.952	0.954	**0.955**
	1%	**0.984**	0.915	0.970	0.973	0.958	**0.972**	0.970	0.971	0.971
	0.2%	**0.970**	0.832	0.948	0.941	0.919	0.951	0.955	**0.957**	0.952
	0.1%	**0.961**	0.856	0.917	0.904	0.796	0.931	0.926	**0.933**	0.930
Mnist	0.05%	**0.947**	0.678	0.826	0.851	0.698	**0.912**	0.904	0.908	0.906

The $1^{st}/2^{nd}$ best results are indicated in red/blue

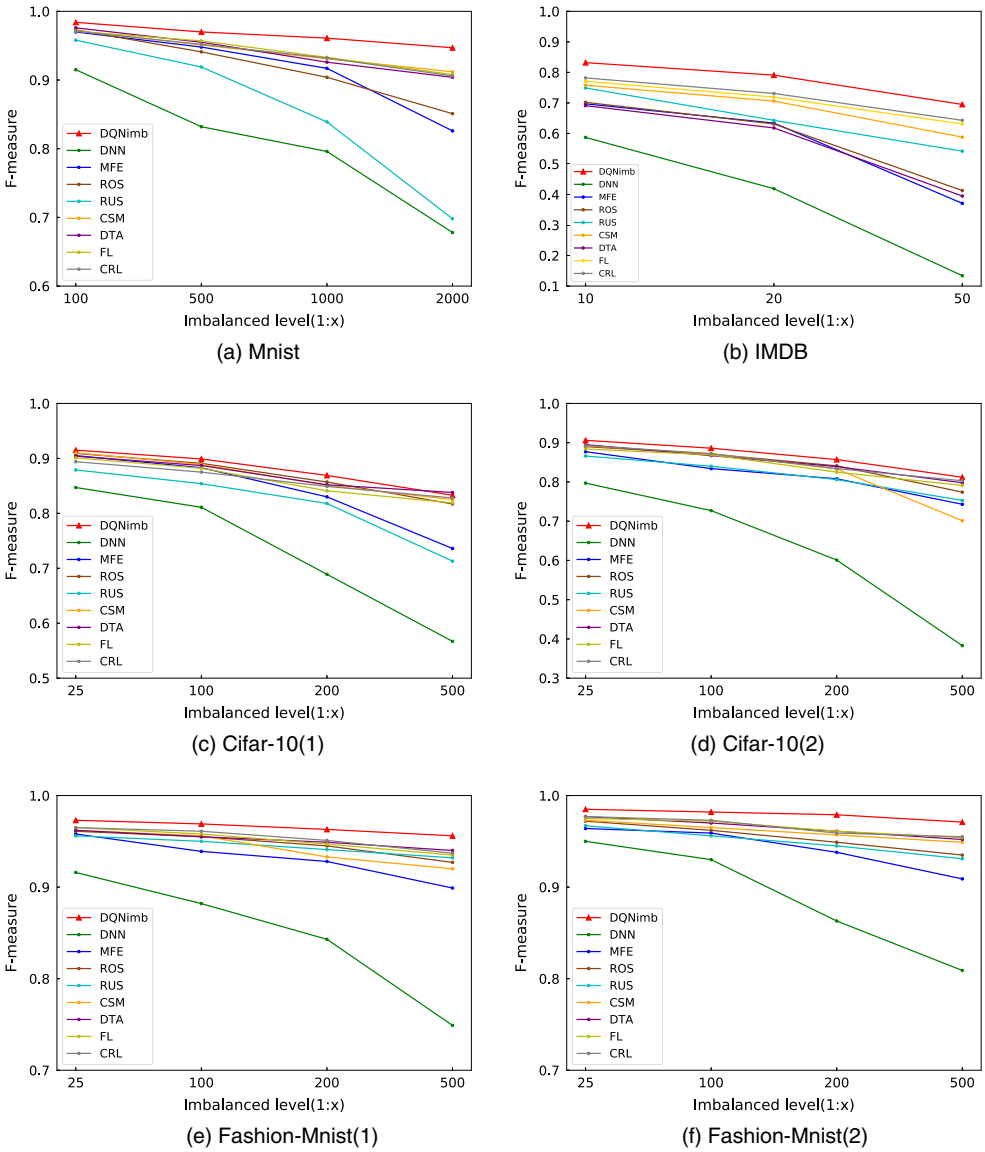

Fig. 2 Comparison of methods with respect to F-measure score on different datasets

outperforms the other approaches with an average rank of almost 1. The other imbalanced learning methods are not significantly different, except of MFE.

4.8 Exploration on reward function

Reward function is used to evaluate the value of actions performed by the agent and inspires it to work toward to the goal. In DQNimb model, the reward of the minority class is 1 and the reward of the majority class is λ. In above experiments, we let $\lambda = \rho$. To study the effect of different values of λ on the classification performance, we test values of $\lambda \in \{0.05\rho, 0.1\rho, 0.5\rho, \rho, 5\rho, 10\rho, 20\rho\}$. The experimental results are shown in Fig. 4.

In the IMDB datasets with different imbalanced ratio, DQNimb model performs best when the reward of the majority class λ is equal to the imbalanced ratio ρ. In other data sets, the results show similar phenomenons. The closer the reward of the majority class λ is to ρ, the better the classification performance of the model is. The value of λ can adjust the impact of majority samples on DQNimb model. When $\lambda = \rho$, the second item and the third item in (13) are equal, the majority samples and minority samples have the same impact on the gradient of Q-network. When

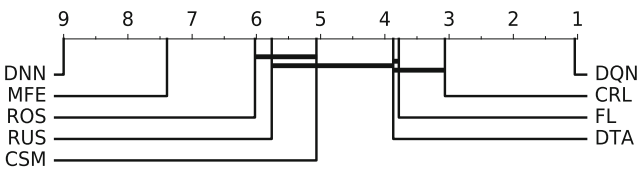

Fig. 3 Critical difference diagram showing pairwise statistical difference comparison of the nine deep learning classifiers on the 23 imbalanced data sets

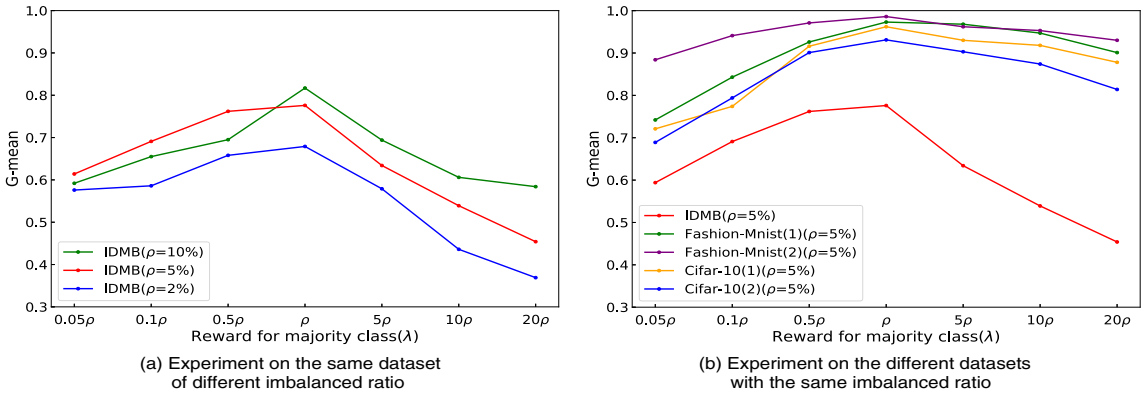

Fig. 4 Different values of rewards (λ) for the majority class to find the optimal reward function

$\lambda > \rho$, the majority samples have an increased influence and the classification performance is declined. If the value of λ is too small, the classification performance is also degraded.

5 Conclusion

This paper introduces a novel model for imbalanced classification using deep reinforcement learning, which formulates the classification problem as a sequential decision-making process (ICMDP), in which the environment returns a higher reward for minority class sample and a lower reward for majority class sample, and the episode will be terminated when the agent misclassifies a minority class sample. We use deep Q-learning algorithm to find the optimal classification policy for ICMDP, and theoretically analyze the impact of the specific reward function on the loss function of Q-network during training. The effect from the two types of the samples on the loss function can be balanced by reducing the reward value the agent receives from the majority samples. Experiments demonstrate that the classification performance of this model in imbalanced data sets is better than other imbalanced classification methods, especially in text data sets and extremely imbalanced data sets. In future work, we will apply improved deep reinforcement learning algorithms to the model, and explore the design of reward function and the establishment of learning environment for classification in imbalanced multi-class data sets.

References

1. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal 6(5):429–449
2. Weiss GM (2004) Mining with rarity: a unifying framework. ACM Sigkdd Explorations Newsletter 6(1):7–19
3. He H, Garcia EA (2008) Learning from imbalanced data. IEEE Trans Knowl Data Eng 9:1263–1284
4. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: Review of methods and applications. Expert Syst Appl 73:220–239
5. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. arXiv:1312.5602
6. Gu S, Holly E, Lillicrap T, Levine S (2017) Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 3389–3396
7. Zhao X, Zhang L, Ding Z, Yin D, Zhao Y, Tang J (2017) Deep reinforcement learning for list-wise recommendations. arXiv:1801.00209
8. Feng J, Huang M, Zhao L, Yang Y, Zhu X (2018) Reinforcement learning for relation classification from noisy data. In: Proceedings of AAAI
9. Martinez C, Perrin G, Ramasso E, Rombaut M (2018) A deep reinforcement learning approach for early classification of time series. In: EUSIPCO, 2018
10. Drummond C, Holte RC et al (2003) C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced datasets II, vol 11, Citeseer, pp 1–8
11. Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: International Conference on Intelligent Computing, Springer, pp 878–887
12. Mani I (2003) I Zhang, knn approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets, vol 126
13. Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6(1):20–29
14. Akkasi A, Varoğlu E, Dimililer N (2017) Balanced undersampling: a novel sentence-based undersampling method to improve recognition of named entities in chemical and biomedical text. Appl Intell, pp 1–14
15. Gupta D, Richhariya B (2018) Entropy based fuzzy least squares twin support vector machine for class imbalance learning. Appl Intell 48(11):4212–4231
16. Wu G, Chang EY (2005) Kba: Kernel boundary alignment considering imbalanced data distribution. IEEE Trans Knowl Data Eng 17(6):786–795
17. Tang Y, Zhang Y-Q, Chawla NV, Krasser S (2009) Svms modeling for highly imbalanced classification. IEEE Transactions

on Systems, Man, and Cybernetics, Part B (Cybernetics) 39(1):281–288
18. Su C, Cao J (2018) Improving lazy decision tree for imbalanced classification by using skew-insensitive criteria. Applied Intelligence
19. Zadrozny B, Elkan C (2001) Learning and making decisions when costs and probabilities are both unknown. In: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 204–213
20. Zadrozny B, Langford J, Abe N (2003) Cost-sensitive learning by cost-proportionate example weighting. In: ICDM, 2003, Third IEEE International Conference on Data Mining, 2003, IEEE, pp 435–442
21. Zhou Z-H, Liu X-Y (2006) Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans Knowl Data Eng 18(1):63–77
22. Krawczyk B, Woźniak M (2015) Cost-sensitive neural network with roc-based moving threshold for imbalanced classification. In: International Conference on Intelligent Data Engineering and Automated Learning, Springer, pp 45–52
23. Chen J, Tsai C-A, Moon H, Ahn H, Young J, Chen C-H (2006) Decision threshold adjustment in class prediction. SAR QSAR Environ Res 17(3):337–352
24. Yu H, Sun C, Yang X, Yang W, Shen J, Qi Y (2016) Odoc-elm: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data. Knowl-Based Syst 92:55–70
25. Ting KM (2000) A comparative study of cost-sensitive boosting algorithms. In: Proceedings of the 17th International Conference on Machine Learning Citeseer
26. Janisch J, Pevnỳ T, Lisỳ V (2017) Classification with costly features using deep reinforcement learning. arXiv:1711.07364
27. Wang S, Liu W, Wu J, Cao L, Meng Q, Kennedy PJ (2016) Training deep neural networks on imbalanced data sets, in Neural Networks (IJCNN). In: 2016 International Joint Conference on. IEEE, pp 4368–4374
28. Huang C, Li Y, Change Loy C, Tang X (2016) Learning deep representation for imbalanced classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5375–5384
29. Yan Y, Chen M, Shyu M-L, Chen S-C (2015) Deep learning for imbalanced multimedia data classification. In: Multimedia (ISM). In: 2015 IEEE International Symposium on. IEEE, pp 483–488
30. Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Netw Learn Syst 29(8):3573–3587
31. Dong Q, Gong S, Zhu X (2018) Imbalanced deep learning by minority class incremental rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence
32. Wiering MA, van Hasselt H, Pietersma A-D, Schomaker L (2011) Reinforcement learning algorithms for solving classification problems. In: 2011 IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), IEEE, pp 91–96
33. Zhang T, Huang M, Zhao L (2018) Learning structured representation for text classification via reinforcement learning. AAAI
34. Liu D, Jiang T (2018) Deep reinforcement learning for surgical gesture segmentation and classification. arXiv:1806.08089
35. Zhao D, Chen Y, Lv L (2017) Deep reinforcement learning with visual attention for vehicle classification. IEEE Trans Cogn Develop Syst 9(4):356–367
36. Abdi L, Hashemi S (2014) An ensemble pruning approach based on reinforcement learning in presence of multi-class imbalanced data. In: Proceedings of the Third International Conference on Soft Computing for Problem Solving, Springer, pp 589–600
37. Dixit AK, Sherrerd JJ et al (1990) Optimization in economic theory. Oxford University Press on Demand
38. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2980–2988
39. Gu Q, Zhu L, Cai Z (2009) Evaluation measures of the classification performance of imbalanced data sets. In: International Symposium on Intelligence Computation and Applications, Springer, pp 461–471
40. Bengio Y (2012) Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade, Springer, pp 437–478
41. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(Jan):1–30
42. Benavoli A, Corani G, Mangili F (2016) Should we really use post-hoc tests based on mean-ranks? J Mach Learn Res 17(1):152–161
43. Wilcoxon F (1992) Individual comparisons by ranking methods. In: Breakthroughs in Statistics, Springer, pp 196–202
44. Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian journal of statistics, pp 65–70

Enlu Lin is currently a M.S. candidate in School of Computer Science and Technology, South China University of Technology (SCUT). He received B.S degree from the department of software engineering in South China Agricultural University (SCAU) in 2017.His research interests include machine learning, deep learning, reinforcement learning and imbalanced classification.

Qiong Chen is currently an Associate Professor in School of Computer Science and Engineering, South China University of Technology. She received her B.S degree from Beijing Institute of Technology in 1987, M.S degree from Harbin Engineering University in 1990, and Ph.D degree from South China University of Technology in 2001. Her research interests include machine learning, imbalanced classification and reinforcement learning.

Xiaoming Qi received B.S degree from the department of computer software in South China University of Technology in 2015, M.S degree from the School of Computer Science and Engineering, South China University of Technology in 2018. He is currently a software development engineer at E Fund Management Co. Ltd. His research interests include machine learning, deep reinforcement learning and quantitative analysis.