International Journal of u- and e- Service, Science and Technology
Vol.8, No.1 (2015), pp.333-346
http://dx.doi.org/10.14257/ijunesst.2015.8.1.30

Dynamic Cost-sensitive Ensemble Classification based on Extreme Learning Machine for Mining Imbalanced Massive Data Streams

Yuwen Huang [1,2]

[1]Department of Computer and Information Engineering, Heze University, Heze 274015, Shandong, China
[2] Key Laboratory of computer Information Processing, Heze University, Heze 274015, Shandong, China
hzxy_hyw@163.com

Abstract

In order to lower the classification cost and improve the performance of the classifier, this paper proposes the approach of the dynamic cost-sensitive ensemble classification based on extreme learning machine for imbalanced massive data streams (DCECIMDS). Firstly, this paper gives the method of concept drifts detection by extracting the attributive characters of imbalanced massive data streams. If the change of attributive characters exceeds threshold value, the concept drift occurs. Secondly, we give Cost-sensitive extreme learning machine algorithm, and the optimal cost function is defined by the dynamic cost matrix. Build the cost-sensitive classifiers model for imbalanced massive data streams under MapReduce, and the data streams are processed in parallel. At last, the weighted cost-sensitive ensemble classifier is constructed, and the dynamic cost-sensitive ensemble classification based on extreme learning machine classification is given. The experiments demonstrate that the proposed ensemble classifier under the MapReduce framework can reduce the average misclassification cost and can make the classification results more reliable. DCECIMDS has high performance by comparing to the other classification algorithms for imbalanced data streams and can effectively deal with the concept drift.

Keywords: *Cost-sensitive; Extreme Learning Machine; Imbalanced Data Streams; Dynamic Matrix*

1. Introduction

With the development of industry technology, there are more and more massive data stream in the wireless sensors, the data communication, the pervasive computing calculation, etc. The current data mining methods are suit for the static data, and the massive data streams are the characteristics of dynamic change, so the traditional data mining technology can't effectively manage the data streams. It is a research hotspot how to design method and model of data mining for the massive data streams. Ireneusz proposed a new approach to use the ensemble classifiers for mining data stream with concept-drift [1]. Muhammad introduced a new clustering for evolving data stream by handling data of varying density [2]. The paper [3] proposed a new method to detect outliers by mining minimal infrequent patterns from data streams. Lee introduced a novel algorithm mining WMFPs with only one scan over sliding window-based data stream environment [4]. But for most real-word applications, the distribution of massive data streams is unbalanced, and some classification numbers are significantly less than the others. More and more industries and enterprises need valuable information from the imbalanced massive data streams, so the researches on imbalance data stream mining have many practical applications. Adel proposed a new approach for handling with non-stationary and imbalanced data streams by two separate cost-sensitive strategies [5]. The paper presented a method which makes use of k nearest neighbors and oversampling technique for imbalanced data stream [6]. The paper [7] proposed one possible data processing scenario for Imbalanced and Partially-Lapelled data streams. The paper [8] proposed a new method for imbalanced data streams to use naive Bayes as the base classifier.

In order to increase the classification performance for imbalance data streams, many approaches are proposed by improving traditional classification algorithms, for example, the cost-sensitive learning, the resample, the improved SVM, etc. The cost-sensitive learning takes a full consideration for the performance of the minority classification, and can resolve effectively the imbalanced classification in real life. Adel proposed an online ensemble of neural network classifiers for non-stationary and imbalanced data streams [9]. The paper [10] proposed an effective cost-sensitive ensemble decision trees for imbalanced data classification. The paper [11] introduced cost-sensitive learning techniques for imbalance by the MapReduce framework to distribute the big data. The paper [12] introduced the objective function of cost sensitive SVM to improve the performance of classification for imbalanced data. The cost-sensitive learning can improve the performance of the imbalanced data streams. Aiming at the limitation of static cost matrix for the imbalanced massive data streams, this paper proposes the approach of the dynamic cost-sensitive ensemble classification based on extreme learning machine for mining imbalanced massive data streams.

2. Concept Drifts Detection with Scenario Characteristics

The data of the massive data streams have the data scenarios, and the data change in data streams is reflected by the change of the data attribute value. For example, in the data streams of the credit card transaction, the data characteristics can be defined as the personal monthly income, the average monthly consumption scale, the education level, the work unit, the consumption area and so on.

The massive data streams $S=\{S_1,S_2,S_3,...,S_n\}$, S_i is the data block. The attribute set $I=\{A_1,A_2,...,A_n,C\}$, the class set $C=\{C_1,C_2,...,C_m\}$. The expected information of data block S_k is as follows.

$$Info(S_k)=-\sum_{i=1}^{m}p_i\log_2(p_i)$$

$p_i=\frac{|C_{i,S_k}|}{|S_k|}$ is the probability of the samples with class C_i. $|C_{i,S_k}|$ is the samples sums with class C_i, and $|S_k|$ is the samples sums of data block S_k.

$A_j=\{a_1,a_2,a_3,...,a_v\}$, a_i is the attribute value. $S_k=\{S_{A_ja_1},S_{A_ja_2},S_{A_ja_3},...,S_{A_ja_v},\}$. The expected information of A_j is as follows.

$$Info_{A_j}(S_k)=\sum_{i=1}^{v}\frac{|S_{a_i}|}{|S_k|}Info(S_{a_i})=\sum_{i=1}^{v}\frac{|S_{a_i}|}{|S_k|}\left(-\sum_{x=1}^{m}p_x\log_2(p_x)\right)$$

$\left|S_{a_i}\right|$ is the sample number with attribute value a_i . $p_z = \frac{\left|S_{C_z}\right|}{\left|S_i\right|}$, $\left|S_{C_z}\right|$ is the sample numbers of the class C_z of S_k. The information gain of attribute A_j is as follows.

$$Gain\left(A_j\right) = Info\left(S_k\right) - Info_{A_j}\left(S_k\right) = -\sum_{i=1}^{m} p_i \log_2\left(p_i\right) + \sum_{i=1}^{v} \frac{\left|S_{a_i}\right|}{\left|S_k\right|}\left(-\sum_{x=1}^{m} p_x \log_2\left(p_x\right)\right)$$

$\log_2\left(p_i\right)$ and $\left(p_i - 1\right)$ are the equivalent infinitesimal, so $\log_2\left(p_i\right)$ can been replaced by $\frac{\left(p_i - 1\right)}{\ln 2}$.

$$\begin{aligned} & Gain\left(A_j\right) = Info\left(S_k\right) - Info_{A_j}\left(S_k\right) \\ & = -\sum_{i=1}^{m} p_i \log_2\left(p_i\right) + \sum_{i=1}^{v} \frac{\left|S_{a_i}\right|}{\left|S_k\right|}\left(-\sum_{x=1}^{m} p_x \log_2\left(p_x\right)\right) \\ & = \frac{1}{\ln 2}\left[\left(1 - \sum_{i=1}^{m} p_i^2\right) - \sum_{i=1}^{v} \frac{\left|S_{a_i}\right|}{\left|S_k\right|}\left(1 - \sum_{x=1}^{m} p_x^2\right)\right] \end{aligned}$$

The time complexity of $Gain\left(A_j\right)$ is the linear level $O\left(n\right)$, and it is suitable for calculating the information gain in the massive data streams.

The information systems $S = \left(U, A, V, f\right)$, $A = \left\{A_1, A_2, A_3, \ldots.. A_n\right\}$. If $\forall P \subseteq A, \forall Q \subseteq A$, the dependency degree of P and Q is as follows.

$$POS_P\left(Q\right) = \left\{x \mid x \in R_{-}\left(x\right), x \in U/Q\right\}$$

$$k = \gamma_p\left(Q\right) = card\left(POS_P\left(Q\right)\right) / card\left(U\right)$$

Q depends on P with the degree k. $card\left(\ \right)$ is the function for the calculating the base number. $R_{-}\left(x\right)$ is the lower approximation. The attribute α is added to Q, and its important degree is as follows.

$$SGF\left(\alpha, P, Q\right) = \gamma_P\left(Q\right) - \gamma_{P - \{\alpha\}}\left(Q\right).$$

Calculate the important degree for all samples, and choose the best important attributes G^{t-1} and G^t at $t-1$ and t time. $G^{t-1} = \left(g_1^{t-1}, g_2^{t-1}, \ldots, g_n^{t-1}\right)$. $G^t = \left(g_1^t, g_2^t, \ldots, g_3^t\right)$

$$\begin{aligned} & g_i^{t-1} = Info\left(S_k^{t-1}\right) - Info_{A_j}\left(S_k^{t-1}\right) \\ & = \frac{1}{\ln 2}\left[\left(1 - \sum_{i=1}^{m}\left(p_i^{t-1}\right)^2\right) - \sum_{i=1}^{v} \frac{\left|S_{a_i}^{t-1}\right|}{\left|S_k^{t-1}\right|}\left(1 - \sum_{x=1}^{m}\left(p_x^{t-1}\right)^2\right)\right] \end{aligned}$$

.

$$\begin{aligned} & g_i^t = Info\left(S_k^t\right) - Info_{A_j}\left(S_k^t\right) \\ & = \frac{1}{\ln 2}\left[\left(1 - \sum_{i=1}^{m}\left(p_i^t\right)^2\right) - \sum_{i=1}^{v} \frac{\left|S_{a_i}^t\right|}{\left|S_k^t\right|}\left(1 - \sum_{x=1}^{m}\left(p_x^{t-1}\right)^2\right)\right]. \end{aligned}$$

The change of information gain between t and t-1 is as follows.

$$\theta=\sqrt{\left(g_1-d_1\right)^2+\left(g_2-d_2\right)^2+\ldots+\left(g_n-d_n\right)^2}$$

The paper extracts the attributive characters of the current data scenarios by the information gain method, and compares them with the previous time. If the change of attributive characters exceeds threshold value, the concept drift occurs.

3. Cost-sensitive Extreme Learning Machine

3.1. Extreme Learning Machine

In 2004, Huang professor proposed firstly the extreme learning machine for Single-hidden Layer feed forward Neural Network, and it is often abbreviated as SLFNs. The extreme learning machine assigns randomly to the weights of the neural network and the skewing of the hidden layer nodes. The extreme learning machine can calculate directly the output weights of the network, and the learning speed of the neural network is greatly improved.

The sample $\left(x_i,t_i\right)$, $x_i\in R^m,t_i\in R^n,i=1,2,\ldots N$. $x_i=\left(x_{i1},x_{i2},\ldots,x_{in}\right)^T$, $t_i=\left(t_{i1},t_{i2},\ldots,t_{im}\right)$.

SLFNs contains the hidden layer nodes with number N, and the output of it is as follows.

$$\sum_{i=1}^{N}\beta_i g\left(w_i\bullet x_j+b_i\right)=t_j \quad j=1,2,\ldots,N$$

$$w_i=\left(w_{i1},w_{i2},\ldots,w_{in}\right)^T$$

$$\beta_i=\left(\beta_{i1},\beta_{i2},\ldots,\beta_{im}\right)^T$$

w_i is the connection weight of the hidden layer, and b_i is the threshold value. β_i is the connection weight form the hidden layer to the output layer. $w_i\bullet x_j$ is the inner product. $g\left(x\right)$ is the activation function, this paper chooses the Sigmoid function.

$$\sum_{i=1}^{N}\beta_i g\left(w_i\bullet x_j+b_i\right)=t_j \ .$$

For short is $H\beta=T$.

$$H\left(w_1,w_2,\ldots,w_N,b_1,b_2,\ldots,b_N,x_1,x_2,\ldots,x_N\right)=$$

$$\begin{bmatrix} g\left(w_1\bullet x_1+b_1\right) & g\left(w_2\bullet x_1+b_2\right) & \ldots g\left(w_N\bullet x_1+b_N\right) \\ g\left(w_1\bullet x_2+b_1\right) & g\left(w_2\bullet x_2+b_2\right) & \ldots g\left(w_N\bullet x_2+b_N\right) \\ \ldots & \ldots \quad \ldots & \ldots \\ g\left(w_1\bullet x_N+b_1\right) & g\left(w_2\bullet x_N+b_2\right) & \ldots g\left(w_N\bullet x_N+b_N\right) \end{bmatrix}$$

$$\beta=\begin{bmatrix}\beta_1^T\\ \beta_2^T\\ \ldots\\ \beta_N^T\end{bmatrix}_{N\times m} \qquad T=\begin{bmatrix}t_1^T\\ t_2^T\\ \ldots\\ t_N^T\end{bmatrix}_{N\times m}$$

The parameters w_i and b_i of the hidden layer aren't adjusted in training process, and can been assigned randomly. Therefore, the training process for SLEFNs is the equal of solving the least square solution β in the linear system of equations

$$H\beta = T .$$

$$\beta = H^{+}T$$

H^{+} is the generalized inverse Moore of the output matrix H in the hidden layer. The extreme learning machine can randomly generate the parameters of the hidden layer, and it avoids the iterative adjustment for the neural network. At the same time, the extreme learning machine can save time and get a better generalization performance comparing to the traditional learning algorithm. Therefore, the extreme learning machine can be applied to the massive data streams for improving the prediction accuracy and computational efficiency.

3.2. Cost-sensitive Extreme Learning Machine

The classification costs are different, and the cost-sensitive machine learning uses classification model for achieving the minimum cost. The formula based on Bayesian risk is follows.

$$R(\alpha_i \mid x) = \min_{\alpha_i \in C} \arg \sum_{j=1}^{M} \lambda(\alpha_i \mid w_j) P(w_j \mid x)$$

$R(\alpha_i \mid x)$ is the least conditional risk, $\lambda(\alpha_i \mid w_j)$ is the loss if the classification w_j is chosen. $P(w_j \mid x)$ is the prior probability.

The risk minimization is converted to the classification cost. If the data sets have M different categories, the costs of misclassification and test cost in classification process are gotten, and the cost matrix is as follows.

$$C_{M\times M} = \begin{bmatrix} c_{1,1}, c_{1,2}, \ldots, c_{1,\mathrm{M}} \\ c_{2,1}, c_{2,2}, \ldots, c_{2,\mathrm{M}} \\ \ldots \ \ldots \ \ldots \ \ldots \\ c_{\mathrm{M},1}, c_{\mathrm{M},2}, \ldots, c_{\mathrm{M},\mathrm{M}} \end{bmatrix}$$

The row stand for a real classification and the column is a predicted class. $c(i,j)$ is the sum of the misclassification and test cost. If α_i and w_j represent classification, the cost that a sample X is classified is as follows.

$$C(i \mid x) = \sum_{j=1}^{M} C(i,j) P(j \mid x)$$

On the basis of the minimum cost of classification rule, the formula of the minimum cost is follows.

$$C(i \mid x) = \min_{i \in M} \arg \sum_{j=1}^{M} C(i,j) P(j \mid x)$$

In cost - sensitive classification, if $i \neq j$, $c(i,j) \neq c(j,i)$, and the sample classification isn't determined only by the maximum probability. The classification steps of the cost-sensitive extreme learning machine are as follows.

Input: test samples $X = \{x_1, x_2, \ldots, x_n\}$, the classification number N, unknown classification samples $T = \{t_1, t_2, \ldots, t_n\}$, the cost matrix $C_{N\times N}$, the hidden layer nodes with number N.

Output: classification results of $T = \{t_1, t_2, \ldots, t_n\}$.

Step 1: for(i=1; i<=K; i++)

{

1. Give randomly the weight and deviation $\left(w_j^i, b_j^i\right)$ of the input layer in the ith ELM, $j=\left(1,2,\ldots L\right)$, and L is the number of the hidden node.

2. Calculate the output matrix $H_{L\times M}^i$ of the hidden layer.

$$H_{L\times M}^i = \begin{bmatrix} f\left(w_1x_1+b_1\right) & \ldots & f\left(w_1x_M+b_1\right) \\ \ldots & \ldots & \ldots \\ f\left(w_Lx_1+b_L\right) & \ldots & f\left(w_Lx_M+b_L\right) \end{bmatrix}.$$

3. Calculate the kth output weight β^i, $\beta^i=\left(H^i\right)^+T$, and T is the output matrix of the ultimate goal.

4. Save the ith extreme learning machine.

}

Step 2: for(i=1; i<=K; i++)

For each test sample, use the ith classifier to predict its classification.

Step 3：Each test sample X has K predict outcomes, and calculate the probability $P\left(\mathrm{i}|\mathrm{x}\right)$ that test sets belong to each classification. Calculate $P\left(\mathrm{i}|x\right)=\dfrac{NUM_i}{K},\mathrm{x}=1,2,\ldots,\mathrm{N}$

Step 4: Use $C\left(i|x\right)=\min\arg_{i\in M}\sum_{j=1}^{M}C\left(i,j\right)P\left(j|x\right)$, and calculate the classifications for $T=\left\{t_1,t_2,\ldots,t_i,\ldots\right\}$.

4. Dynamic Cost-sensitive Ensemble Classification under the MapReduce framework for Mining Imbalanced Massive Data Streams

4.1. Framework of MapReduce

MapReduce is very suit for processing the massive data streams. A massive task is divided into several subtasks by MapReduce, and the subtasks are scheduled to virtual compute nodes. Map task divides the massive task into M parts that are executed concurrently by workers, and outputs the disposed intermediate results. Reduce analyzes and gathers the results, and outputs the final results. In MapReduce programming model, Map tasks numbers are more than Reduces tasks, so each worker executes more than one task. The process of Map and Reduce is as follows.

$$Map\left(k_1,v_1\right)\rightarrow\left[\left(k_2,v_2\right)\right]$$
$$\mathrm{Reduce}\left(k_2,\left[v_2\right]\right)\rightarrow\left[\left(k_3,v_3\right)\right]$$

4.2. Dynamic Cost Matrix

In cost-sensitive learning, Turney proposed nine sort costs, and this paper concerns mainly the misclassification and test cost. The misclassification cost is the expenditure that a real classification is wrongly predicated as the other. The test costs are the expenditure for acquiring the attribute value by the test method, and the test costs are relative. There are static and dynamic cost in cost-sensitive learning, and the current researches focus on the static cost. The stationary cost leads the instability of the algorithms, and weakens the generalization performance of classifiers, so it is only applied to the experiments and prospective study. The static cost mechanism has deficiencies, especially in the face of the serious imbalanced distribution, and the performance of the classifiers structured by the static cost is inefficient. Dynamic cost mechanism can search the optimal cost in feasible cost space, and improve the correct classification rate as far as possible. In the meantime, the total expenditure is low. Combined with the experience and knowledge of the experts, this paper uses the multiple feasible costs in test cost space, and the construction method for dynamic test cost is as follows.

Step 1: Give the question of the application area. For example, the patient needs the type-B ultrasonic examination.

Step 2: Choose M difference ways for processing data. For example, choose M hospitals for the type-B ultrasonic examination.

Step3: Combine the expertise in the application area, and find the cost of M difference ways for processing data. For example, the medical professionals of ultrasonic examination have different levels, so the costs are different.

Step 4: Sort M costs and get the ascending sequence, and the closed interval of the cost space is constituted by the minimum and maximum in sequence. For example, in M different hospital, the minimum cost of type-B ultrasonic examination is \$10, and the maximum is \$30, so the test cost space is closed interval [10, 30].

If the test cost spaces of all classification are known, the test cost and misclassification matrix in data sets are as follows.

$$test\_cost_{M\times M}=\begin{bmatrix} tc_{1,1},tc_{1,2},\ldots,tc_{1,M} \\ tc_{2,1},tc_{2,2},\ldots,tc_{2,M} \\ \ldots\ \ldots\ \ldots\ \ldots \\ tc_{M,1},tc_{M,2},\ldots,tc_{M,M} \end{bmatrix},\ mis\_cost_{M\times M}=\begin{bmatrix} mc_{1,1},mc_{1,2},\ldots,mc_{1,M} \\ mc_{2,1},mc_{2,2},\ldots,mc_{2,M} \\ \ldots\ \ldots\ \ldots\ \ldots \\ mc_{M,1},mc_{M,2},\ldots,mc_{M,M} \end{bmatrix}$$

The row is the real classification, and the column is the prediction. The variable cost $tc_{i,j}\in\left[a_{ij},b_{ij}\right]$ is the test cost space, and $mc_{i,j}$ is the static misclassification cost. $c\left(i,j\right)=test\_\cos t+mis\_\cos t$.

4.3. Definition of the Optimal Cost Function

In order to measure whether the searched value is the optimal in cost space, this paper extends the geometric mean function, and the cost function of any point in cost space is defined as follow.

$$f\left(\cos t\_usu\right)=\sqrt{RE_r\left(\cos t\_any\right)\times PR_r\left(\cos t\_any\right)\times RE_m\left(\cos t\_any\right)\times PR_m\left(\cos t\_any\right)}\ .$$

$$RE=\frac{TP}{TP+FN}\ .$$

$$PR=\frac{TP}{TP+FP}\ .$$

The response rate RE is the number ratio of the correct predicted positive samples and the sum of the real positive sample, and it is the recall ratio of the classifier. The precision ratio PR is the number ratio of the correct predicted positive samples and the sum of the predicted positive samples. $\cos t\_usu$ is a point of the cost space. $RE_r\left(\cos t\_any\right)$ and

$RE_m(cost\_any)$ are the response rate any point of cost space by the cross validation in the minority and major classes, and $PR_r(cost\_any)$ and $PR_m(cost\_any)$ are its precision ratio. The goal of the above cost function is to search the optimal equilibrium point in between the minority and major classes. In other words, the response and precision of the minority and major classes are improved at the same time. The optimal cost function in cost space is defined as follow.

$$f(cost\_opt) = \arg\max_{cost \in COST} \sqrt{RE_r(cost\_opt) \times PR_r(cost\_opt) \times RE_m(cost\_opt) \times PR_m(cost\_opt)}$$

$cost\_opt$ is the optimal point in the cost space, and it is the optimal equilibrium point of the response and precision in between the minority and major classes.

4.3. Dynamic Cost Optimization Algorithm

The enumerative iteration is the best way for searching for the optimal in cost space, and the search results can be the best approximation. However, the enumerative iteration consumes too much time, and it isn't suitable for the massive data stream. In most real-word applications, all test costs increase in the same pace, so this paper proposes the cost searching method based on step length. In other words, the cost space is divided into N equal step length, and the searching time is save, so the method is suitable for the massive data streams.

Input: Data sets X, dynamic test cost matrix $test\_cost_{M \times M}$, $c_{ij} \in [a_{ij}, b_{ij}]$, static misclassification cost matrix $mis\_cost_{M \times M}$, step length N.

Step 1. Initialize steps length d=1, the test cost matrix $test\_cost_{M \times M}^{\text{optimal}}$, misclassification cost matrix $mis\_cost_{M \times M}$. $c_{ij} = a_{ij}$, $mid\_cost_{M \times M}^{d} = test\_cost_{M \times M}^{\text{optimal}}$.

Step 2. Use $mid\_cost_{M \times M}^{d}$ and $mis\_cost_{M \times M}$ to structure the cost-sensitive basic classifier M_d by the cost-sensitive extreme learning machine algorithm in data set X.

Step 3. Classifier M_d run the 10 - fold cross validation in data set X, and calculate the cost function $f(cost)_d$. If $d=0$, $f(cost)_{optimal} = f(cost)_0$. Otherwise, if $f(cost)_{optimal} \prec f(cost)_d$, $f(cost)_{optimal} = f(cost)_d$.

Step 4. $d = d+1$, $c_{ij} \in mid\_cost_{M \times M}^{d}$, $c_{ij} = c_{ij} + \frac{b_{ij} - a_{ij}}{N}, i = 1,2,...,M, j = 1,2,...,M$. If $d \succ N$, turn to step 5. Otherwise, turn to step 2.

Step 5. Algorithm ends.

4.5. Weight of Cost-sensitive Ensemble Classifiers

The goal of the traditional ensemble learning for the large-scale classification is the accuracy rate and total performance, and the weight of the basic classifier is the predicting accuracy. For the imbalance mass unbalance data stream, this paper considers the response and precision rate, and combines the cost-sensitive thought to determine the weight of the base classifier.

Test samples $X = \{x_1, x_2, x_3, ..., x_n\}$, $x_i = \{x_{i1}, x_{i2}, x_{i3}, ..., x_{id}\}$. The consumption of the basic classifier f_i is constituted by the misclassification and test cost.

$$\mathrm{cost}_{f_i} = test\_\mathrm{cost}_{f_i} + mis\_\mathrm{cost}_{f_i} ,$$

$$\mathrm{mis\_cost}_{f_i} = \sum_{a=1}^{M}\left(p_a \times \sum_{b=1}^{M} \mathrm{mis\_cost}(b,a) \right),$$

$$test\_\mathrm{cost}_{f_i} = \sum_{i=1}^{n}\sum_{j=1}^{d} test\_\mathrm{cost}(x_{ij})$$

$\mathrm{mis\_cost}_{f_i}$ is the misclassification cost of the basic classifier f_i, and $test\_\mathrm{cost}_{f_i}$ is its test cost. $\mathrm{mis\_cost}(b,a)$ is the consumption that category A is wrongly classified as B. $test\_\mathrm{cost}(a_i,a_i)=0$. $test\_\mathrm{cost}(x_{ij})$ is the cost for test the attribute x_{ij}. p_a is the ratio of the category I and the total number of samples. The weight of the basic classifier f_i is as follows.

$$w_{f_i} = \left(\alpha \times RE_{f_i} \times PR_{f_i}\right) \Big/ \left(\beta \times \left(error\_num_{f_i} \times \sum_{a=1}^{M}\left(p_a \times \sum_{b=1}^{M} \mathrm{mis\_cost}(b,a) \right) + \sum_{k=1}^{n} test\_\mathrm{cost}_{f_i}(x_k) \right) \right)$$

α and β is the regulatory factor. RE is the response rate of classifier f_i, and PR is the precision rate. $error\_num_{f_i}$ is the misclassification numbers of classifier f_i.

The final result of K classifiers is as follows.

$$class(x) = \sum_{i=1}^{k} w_{f_i} * f_i(x)$$

$x \in X$, w_{f_i} is the weight of the basic classifier f_i, and $f_i(x)$ is the classification results of the classifier f_i.

4.6. Model of the Cost-sensitive Classifiers for Massive Data Streams under MapReduce

Firstly, dynamic data streams are transformed into the static data flow. In the Map phase, the main tasks are the concept drift detection and the construction for basic classifier. If the data characteristics changes of the previous data block exceed a threshold value, the concept drift occurs. In reduce phase, the basic classifiers are integrated. The model of the cost-sensitive classifier for massive data streams is as follows.

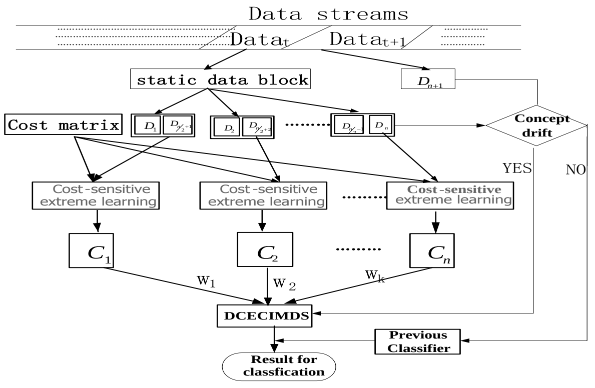

Figure 1. Cost-sensitive Ensemble Classification for Massive Data Streams under Map_reduce

4.7. Cost-sensitive Classification Algorithm Under MapReduce for Imbalance Massive Data Streams

Description of cost-sensitive classification algorithm Under MapReduce for imbalance massive data streams is as follows.

Input: Imbalanced data streams $X=\{x_1,x_2,...,x_n,x_{n+1},....\}$, the classification number N. Dynamic test cost matrix $test\_cost_{M\times M}$, static misclassification cost matrix $mis\_cost_{M\times M}$, the hidden layer nodes with number N.

Output: Classification result of $X=\{x_1,x_2,...,x_n\}$.

Step 1. Initialization.

Turn the dynamic data streams into the static data block by the sliding window technology. Get the static data block $D_{1\sim n}$ and D_{n+1} in chronological order, and the data block D_{n+1} would be classified. The size of $D_{1\sim n}$ is N times more than D_{n+1}.

Step 2. Map tasks.

Firstly, divide $D_{1\sim n}$ into $\{D_1,D_2,...,D_n\}$, and use the information gain method for the concept drift detection. Compare the data characteristics of each D_i and $D_{n/2+i}$ in the data block. If the change of attributive characters doesn't exceeds threshold value, the concept drift doesn't occur, and turn to step 4. Secondly, each D_i is assigned to Workers, and all Workers execute concurrently the cost optimization algorithm based on the cost-sensitive extreme learning machine. The cost-sensitive basic classifiers sets $\{C_1,C_2,...,C_n\}$ with the optimal costs are produced.

Step 3. Reduce task.

Calculate the weight for each classifier, and integrate the basic classifiers to structure the cost-sensitive integrated classifier.

Step 4. Use the current classifier to classify the data block D_{n+1}, and the output classification results.

Step 5. The algorithm ends.

5. Simulation Experiment

5.1. Data sets

5.1.1. Synthetic Datasets: In order to verify the concept drift, choose to STAGGER as testing datasets $Stream\_1$, and STAGGER contains three categorical features. In order to simulate the skew distribution, experiments need to choose two features as majority class, and the rest is minority class. The imbalance ratio is set as 0.01. The Hyperplane is widely applied in data stream classification, On the basis of the Hyperplane, modify MOA Task Launcher for synthesizing concept drift data set $Stream\_2$. The data samples of D-dimension are structured as follows.

$$\sum_{i=0}^{d} a_i x_i = a_0$$

x_i is I-dimension coordinate of sample x, $a_i \in [0,1]$ is weight value. The concept drifts are generated by setting the different parameter. The dataset was generated rotating the Hyperplane from 0 to 360, incrementing 5 degrees in each step. At each step, 1,0000 instances were generated. The imbalance ratio of the whole dataset is set to 0.05.

5.1.2. Real Datasets: Choose the classic data set KDD99 as the experimental data. KDD99 data set contains 494021 data samples, and each sample contains 42 attributes. This paper disposes KDD99 as imbalanced data sets. One normal classification is majority, and the remaining network attacks are reduced into 2 classifications as minority classes. The ratio of the majority is 99.8%, and each minority is 0.1%. PAKDD 2009 Credit Card dataset comes from the private label credit card operation of a major Brazilian retail chain (2003-2008), and it contains 50,000 data sample with 27 data characters. There are two classifications that are bad and good for client's status, and the imbalance ratio is 0.02.

5.2. Evaluation Index and Cost Matrix

In general, the overall accuracy is often used to measure the performance for classification models. However, the influence of minority classification is grater than the majority for imbalanced data streams, therefore, so the evaluation index is defective. In this paper, the evaluation criterions for classifier are as follows.

The overall accuracy (OA): $OA=\frac{TP+TN}{TP+FN+TN+FP}$

$$G-Mean=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}},$$

$$Recall(\mathrm{Re})=\frac{TP}{TP+FN},$$

$$\mathrm{P}recision\left(\mathrm{PR}\right)=\frac{TP}{TP+FP}$$

$$F-measure(Fm)=\frac{2\times recall\times precision}{Recall+precision}=\frac{2\times TP}{2\times TP+FP+FN},$$

$$TPR=\frac{TP}{TP+FN}\ ,\quad TNR=\frac{TN}{TN+FP}.$$

The test cost matrix $test\_cost_{M\times M}$ is dynamic, $c_{ij}\in test\_cost_{M\times M}$, $c_{ij}=\left[a_{ij},b_{ij}\right]$. The lower bound a_{ij} is 1.0, and the upper bound b_{ij} is 100 in this paper. The misclassification cost matrix $mis\_cost_{M\times M}$ is fixed. If the misclassified cost that the negative sample is classified as the positive sample is 1 in this paper, the cost is 100 when the positive is classified as the negative.

5.3. Result of Experiments

Mapreduce framework is composed of 11 PC. One PC is NameNode, and 10 PC is DataNode. Among them, each PC is configured with Intel Xeon (R) (R) E5620 2.40 GHz, dual-core CPU, 4 G RAM. Use Weka as experiment platform, and choose Weka + JDK6.0 as the development environment. In the process of structuring the cost-sensitive classification, choose a manageable number of samples as the training set, and the rest is test set. Each dataset is repeated 30 times in the experiments with the same parameters, and the mean is the final result.

Firstly, verify the validity of the dynamic Cost-sensitive Ensemble Classification this paper proposes for imbalanced massive data streams (DCECIMDS). RDFCSDS [13], OACLNIS [14] are the other algorithms for mining imbalanced data streams. The classification results in the common evaluation criterion for imbalanced data are as follows.

Table 1. Classification Results for KDD99

Algorithm	TPR	TNR(RE)	G-mean	PR	F-measure
RDFCSDS	0.0234	0.9932	0.1524	0.9123	0.0285
OACLNIS	0.0673	0.9914	0.2583	0.9326	0.1365
DCECIMDS	0.0685	0.9906	0.2605	0.9744	0.3241

Table 2. Classification Results for PAKDD 2009 Credit Card Dataset

Algorithm	TPR	TNR(RE)	G-mean	Pr	F-measure
RDFCSDS	0.0281	0.9935	0.1670	0.9347	0.0361
OACLNIS	0.0712	0.9927	0.2654	0.9512	0.1725
DCECIMDS	0.0734	0.9904	0.2696	0.9735	0.2453

Form the Table 1 and 2, TPR, PR, G-mean, F-measure of DCECIMDS are more excellent than the other classifiers for imbalanced massive data streams.

Secondly, validate the dynamic cost is superior to static cost. Alter the dynamic test matrix as the static cost, and structure the algorithm of static cost-sensitive ensemble Classification for imbalanced massive data streams (SCECIMDS). The G-mean curves in KDD99 and PAKDD 2009 Credit Card dataset are as follows in Figure 2 and 3.

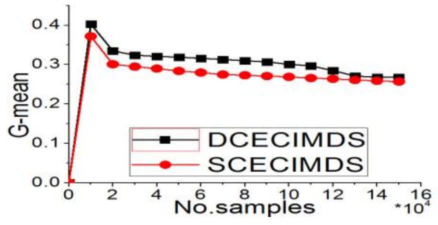

Figure 2. Geometric Mean Learning Curve for KDD99

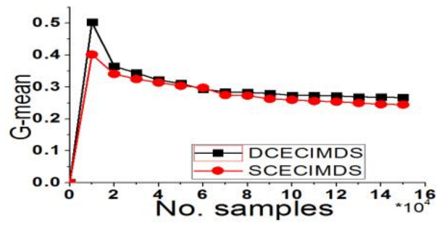

Figure 3. G-mean Learning Curve for PAKDD 2009 Credit Card Dataset

As can be seen from Figure 2 and 3, G-mean performance of the dynamic test matrix is higher than the static matrix, so the dynamic cost is feasible.

At last, validate DCECIMDS can efficiency deal with the concept drift for imbalanced massive data streams. In order to detect the concept drift, divide synthetic datasets *Stream_One* and *Stream_Two* into 15 blocks. The F-measure curves that DCECIMDS compares with RDFCSDS and OACLNIS in *Stream_One* and *Stream_Two* are as follows in Figure 4 and 5.

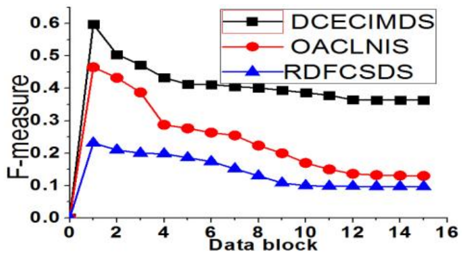

Figure 4. F-measure Learning Curve for *Stream_One*

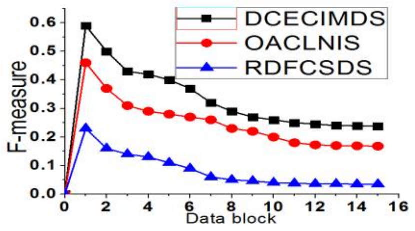

Figure 5. F-measure Learning Curve for *Stream_Two*

From Figure 4 and 5, F-measure Learning curves of DCECIMDS are more excellence than RDFCSDS and OACLNIS for unbalanced data stream with concept drift in *Stream_One* and *Stream_Two* .

6. Conclusion

There are more and more imbalanced massive data stream in the real life, however, due to the massive, dynamic nature of data stream, the traditional classification algorithm has been unable to reach the data stream processing requirements, effective data stream processing and mining valuable information has become a hotspot study at home and abroad. This paper proposes the approach of the dynamic cost-sensitive ensemble classification based on extreme learning machine for mining imbalanced massive data streams. We give a new method which sets the threshold of the scenario characteristic to predict the occurrence of concept drift. The cost-sensitive learning is incorporated into classifiers based on extreme learning machine. The weighted cost-sensitive ensemble classifier is constructed, and the model of dynamic cost-sensitive ensemble classification under Mapreduce is given. Different Map and Reduce tasks can be executed highly parallel for mining imbalanced massive data streams, and the dynamic cost-sensitive ensemble algorithm is developed by the cloud platform. Experiments results show the proposed algorithm has high performance by comparing to the other classification algorithms for imbalanced data streams and can effectively deal with the concept drift.

Acknowledgements

This work is supported by the science and technology projects of the Shandong province universities (No. J14LN65, No. J13LN53). This work has been partially supported by grants from the research fund of heze university (No. XY14KJ06).

References

[1] Czarnowski and P. Jędrzejowicz, "Ensemble Classifier for Mining Data Streams", Procedia Computer Science, vol. 35, (**2014**), pp. 397-406.
[2] M. Z. Rehman, T. Li, Y. Yang and H. Wang, "Hyper-ellipsoidal clustering technique for evolving data stream", Knowledge-Based Systems, vol. 70, (**2014**) November, pp. 3-14.
[3] C. S. Hemalatha, V. Vaidehi and R. Lakshmi, "Minimal infrequent pattern based approach for mining outliers in data streams", Expert Systems with Applications, vol. 42, no. 4, (**2015**) March, pp. 1998-2012.
[4] G. Lee, U. Yun and K. H. Ryu, "Sliding window based weighted maximal frequent pattern mining over data streams", Expert Systems with Applications, vol. 41, no. 2, (**2014**) February 1, pp. 694-708.
[5] A. Ghazikhani, R. Monsefi and Yazdi, "Online cost-sensitive neural network classifiers for non-stationary and imbalanced data streams", Neural Comput. & Applic., vol. 23, (**2013**), pp.283–1295.
[6] A. Godase and V. Attar, "Classifier Ensemble for Imbalanced Data Stream Classification", Proceedings of the CUBE International Information Technology Conference, ACM New York, NY, USA, (**2012**), pp. 284-289.
[7] R. J. Lyon, J. M. Brooke, J. D. Knowles and B. W. Stappers, "A Study on Classification in Imbalanced and Partially-Labelled Data Streams", Systems, Man, and Cybernetics (SMC), IEEE International Conference on, (**2013**), pp. 1506-1511.
[8] H. M. Nguyen, E. W. Cooper and K. Kamei, "Online learning from imbalanced data streams", Soft Computing and Pattern Recognition (SoCPaR), 2011 International Conference of 14-16 October 2011, Da Lian, (**2011**), pp. 347 - 352.
[9] A. Ghazikhani, R. Monsefi and H. S. Yazdi, "Ensemble of online neural networks for non-stationary and imbalanced data streams", Neurocomputing, vol, 122, (**2013**) December 25, pp. 535-544.
[10] B. Krawczyk, M. Woźniak and Gerald Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification", Applied Soft Computing, vol. 14, Part C, (**2014**) January, pp. 554-562.
[11] V. López, S. D. Río, J. M. Benítez and F. Herrera, "Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data", Fuzzy Sets and Systems, vol. 258, (**2015**) January 1, pp. 5-38.
[12] P. Cao, D. Zhao and O. Zaiane, "An Optimized Cost-Sensitive SVM for Imbalanced Data Learning", Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science, vol. 7819, (**2013**), pp. 280-292.
[13] J. Gao, B. L. Ding, W. Fan, J. W. Han and S. Y. Philip, "Classifying data streams with skewed class distributions and concept drifts", IEEE Internet Computing, vol. 12, no. 6, (**2008**), pp. 37-49.
[14] G. Adel, M. Reza and S. Y. Hadi, "Ensemble of online neural networks for non-stationary and imbalanced data streams Neurocomputing", (**2013**) December 25, pp. 535-544.

Author

Yuwen Huang, he was born in 1978 at Shanxian, and received the Master of Engineering in Computer Science from the "Guangxi Normal university" in 2009. She is now a lecturer at the Department of Computer and Information Engineering, Heze University. His research interests include the data-mining, intelligence Calculation.