Research on cost-sensitive ensemble classification 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*

Abstract

The existing classifiers for massive data streams do not consider the imbalance distribution and cost factors, so this paper proposes the approach of the cost-sensitive ensemble classification for imbalanced massive data streams (CECIDS). Firstly, this paper gives the construction method for cost-sensitive ensemble SVM Classification, which is integrated by the classifiers with oversampling, sub-sampling and reconstituted sample space. Secondly, we propose a classifier method BL_KNNModel, which is based on KNNModel algorithm for imbalanced massive data streams. BL_KNNModel can detect the concept drift streams by using the variable windows size, which has lower time complexity. At last, the cost-sensitive ensemble classifier for imbalanced massive data streams is given, which has the virtue of high classification and lower time complexity. In addition, the cost-sensitive ensemble SVM algorithm is used to handle the confused instances. The experiments using both synthetic and real datasets show that compared to the other classification algorithms for imbalanced data streams, CECIDS has higher evaluating indicator and more excellence integrated learning curve.

Keywords: Imbalanced data streams, Ensemble classification, Cost-sensitive SVM

1 Introduction

In recent years, with the development of computer techniques, more and more massive data Streams are produced in wireless sensor networks, network traffic monitoring, credit card fraud detection, earthquake monitoring and weather forecast. The traditional data mining methods are designed on static data, and massive data streams are real-time and dynamic variability, so the traditional data mining techniques can not effectively handle massive data and the research for massive data streams is the hot spot in mining data. The current model for massive data streams can be classified into two categories, one is the single-model structure, and the other is multi-model. A model structure is used to classify the data flow in single-model structure, and the common classification models are as follows. Abdulsalam proposed a method to use incremental learning for data stream classification, which can accelerate self-renewal by the incremental processing model [1]. Hashemi designed a complete data stream processing classification method, which is called the flexible decision tree model [2]. Tang proposed SVM model for unbalanced data stream classification [3]. Kuncheva proposed a data stream classification variable model with sliding window, which can automatically enlarge and reduce the size of the window when the concept drifts occur [4]. Liang extended the very fast decision tree algorithm for handling no labels data [5]. The integrated classification models for massive data streams use multiple identical and different classification models with the hybrid architecture to build strong classifiers. Amin was inspired by thoughts of the integrated learning, and used multiple single-layer neural networks to build an integrated learning model [6]. Avci proposed a mixed method based on the genetic algorithm and support vector machine model [7]. Bu built the hybrid learning model to use Bayesian networks and probabilistic neural network combination by the context of human motional data stream [8]. Chen proposed a mixed element based on Meta-evolution rules respond classification model [9]. Chi proposed an integrated learning ideological classification model based the hyper spectral remote sensing data stream [10]. Dembczynski built integrated learning classification model for the multi-standard problem to use the monotonic constraint method [11]. Hashemi built an integrated multi-decision tree classification model by a hybrid approach [12]. Huang used the ant colony optimization algorithm and support vector machine model to build an integrated classification model [13]. Adler used the bootstrap method by extracting different training set to build an integrated learning model [14]. Baumgartner and Serpen proposed a heuristic classification hybrid model [15]. The distributions of massive data streams in practical applications in are unbalanced, and the numbers of certain classes are less than other classes, which are called imbalanced massive data stream. There are little researches on imbalanced data stream classification. Gao proposed an ensemble classification model by oversampling for imbalanced data streams [16]. Adel proposed cost-sensitive neural network for imbalanced

[*] ***Corresponding author*** e-mail: 57065873@qq.com

data streams [17]. Liu gave algorithm of reusing data for classifying skewed data stream [18]. GAO proposed an approach for classifying data with imbalanced class distributions and concept drifts [19]. By improving KNNModel algorithm and constructing cost-sensitive ensemble SVM Classification, this paper proposes the approach of the cost-sensitive ensemble classification for imbalanced massive data streams.

2 Cost-sensitive ensemble SVM classification

2.1 COST-SENSITIVE SVM CLASSIFICATION

The goal of traditional SVM minimizes the objective function, and the linear inseparable is as follows.

$$R(w,\xi)=\frac{1}{2}\|w\|^2+C\left(\left(\sum_{i=1}^{n}\xi_i\right)\right), \tag{1}$$

$$s.t\ \ y_i(x_i\times w+b)\geq 1-\xi_i,\quad \xi_i\geq 0, i=1,...,n.$$

Misclassification is asymmetric, and cost-sensitive SVM is expressed as follows.

$$R(w,\xi)=\frac{1}{2}\|w\|^2+C\left(\left(\sum_{i=1}^{n}\cos t_i\xi_i\right)\right), \tag{2}$$

$$s.t\ \ y_i(x_i\times w+b)\geq 1-\xi_i,\quad \xi_i\geq 0, i=1,...,n.$$

$\cos t_i$ is misclassification of different sample, and $\sum_{i=1}^{n}\cos t_i\xi_i$ is experience cost. C is relaxation factor, and it controls the balance of the structure and experience cost.

The optimal function for cost-sensitive SVM classification in the case of the linear separable is as follows.

$$f(x)=\operatorname{sgn}\left\{\sum_{i=1}^{n}\alpha_i^* y_i(x_i\cdot x)+b^*\right\}. \tag{3}$$

The above formula is turned into dual form by the Lagrange equation, and the optimal function for cost-sensitive SVM classification with kernel function is as follows in the case of the linear inseparable.

$$f(x)=\operatorname{sgn}\left\{\sum_{i=1}^{n}\alpha_i^* y_i K(x_i\cdot x)+b^*\right\}. \tag{4}$$

2.2 OVERSAMPLING COST-SENSITIVE SVM CLASSIFICATION

$\cos t_i$ is misclassification cost of class i, and $\cos t_{ij}(i,j\in 1,2,...,C)$ is cost of classifying i to j. $\cos t_{ii}=0\ (i,j\in 1,2,...,C)$. $\cos t_i=\sum_{j=1}^{C}\cos t_{ij}$. Oversampling method can change the distribution for the training set, and copy the samples with high cost, so the distribution of each class is proportional to its cost.

The Oversampling samples number is calculated as follows.

$$\lambda=\arg\min_{i\in C}\frac{\cos t_i}{N_i},\ i=1,2,...,C, \tag{5}$$

$$N_k^*=\left\lfloor\frac{\cos t_k}{\cos t_\lambda}N_\lambda\right\rfloor.$$

N_k is the number of class k, and $N_k^*\text{-}N_k$ is the added samples number. The added samples are replaced randomly, and the description of oversampling Cost-sensitive SVM Classification is as follows.

Input: Training sample set $S{=}S_1\cup S_2\cup...\cup S_C$.

Output: Oversampling Cost-sensitive SVM Classification OCS_SVM

```
Step 1:   $S^*{=}S$ is new data set.
Step 2:  for (i=0;i<=$C$;i++)
{
    $\lambda=\arg\min_{i\in C}\frac{\cos t_i}{N_i},\ i=1,2,...,C.$
    $N_k^*=\left\lfloor\frac{\cos t_k}{\cos t_\lambda}N_\lambda\right\rfloor$;
    If ( $N_k^*\succ N_k$ ) $N_k^*\text{-}N_k$ samples are randomly placed into $s^*$ from $S_k$;
}
Step 3: Use data set $S^*$ to train the cost-sensitive SVM.
```

2.3 SUB-SAMPLING COST-SENSITIVE SVM CLASSIFICATION

Sub-sampling method changes the distribution of the train sets, and leads that the samples number is proportional to its cost. Sub-sampling method is different with oversampling, and it decreases samples with low cost.

The sub-sampling samples number is calculated as follows.

$$\lambda=\arg\max_{i\in C}\frac{\cos t_i}{N_i},\ i=1,2,...,C.$$

$$N_k^*=\left\lfloor\frac{\cos t_k}{\cos t_\lambda}N_\lambda\right\rfloor.$$

N_k is the number of class k, if $N_k^*\prec N_k$, sample numbers $N_k^*\text{-}N_k$ in class k are decreased. The description of sub-sampling Cost-sensitive SVM Classification is as follow.

Input: Training samples set $S{=}S_1\cup S_2\cup...\cup S_C$.

Output: Sub-sampling Cost-sensitive SVM Classification UCS_SVM

```
Step 1: $S^*=S$ is new data set.
Step 2: for (i=0; i<=$C$;i++)
{
  $S^*-S_k$ Samples are placed into $s^*$
  $\lambda=\arg\max_{i\in C}\frac{\cos t_i}{N_i}$, $i=1,2,...,C.$
  $N_k^*=\left\lfloor\frac{\cos t_k}{\cos t_\lambda}N_\lambda\right\rfloor$;
  if ( $N_k^*\prec N_k$ ) $\left\lfloor\frac{N_k^*}{2}\right\rfloor$ samples are randomly placed into $s^*$ from $S_k$;
  else $S_k$ samples are randomly placed into $s^*$.
  If (the samples number with class $k$ is more than $N_k^*$)
    Delete randomly samples.
}
Step 3: Use data set $s^*$ to train the cost-sensitive SVM.
```

2.4 COST-SENSITIVE SVM CLASSIFICATION OF RECONSTITUTED SAMPLE SPACE

The sample set $X=\{(x_1,y_1),(x_2,y_2)...(x_i,y_i)...(x_n,y_n)\}$, $x_i\in R^\ell$, $y_i\in\{C_1,C_2,...C_m\}$. $p(j|x)$ is the probability that sample x belongs to class j, The minimum conditional risk of classifying sample x into class i is as follows by Bayes decision theory.

$$\mathrm{R}(\mathrm{i}\,|\,x)=\sum_j P(j\,|\,x)C(i,j)\,.$$

The reconstituted training samples $\bar{y}=\arg\min_{i\in m}\{R(i\,|\,x)\}$.

Input: cost-sensitive SVM, training sample set $X=\{(x_1,y_1),(x_2,y_2)...(x_i,y_i)...(x_n,y_n)\}$,. $y_i\in\{C_1,C_2,...C_m\}$.

Output: Cost-sensitive SVM Classification of reconstituted sample space RS_SVM

```
Step1: for( i=1;i<=n; i++)
         for(j=1;j<=m; j++)
  Calculate $P(C_j\,|\,x_i)$ by Cost-sensitive SVM Classification.
Step 2: for (k=1; k<=n;k++)
  {
    $\mathrm{R}(i\,|\,x)=\sum_{i,j\in m}P(j\,|\,x)C(i,j)$
    $\bar{y}=\arg\min_{i\in m}\{R(i\,|\,x_k)\}$
  }
Step 3: Reconstruct training set $X$ into $\bar{X}$.
Step 4: Reconstitute Cost-sensitive classifier $RS\_SVM$ by training set $\bar{X}$ and Cost-sensitive SVM Classification.
```

2.5 COST-SENSITIVE WEIGHT INTEGRATION OF CLASSIFIER

The traditional integrated learning algorithms for massive data streams prefer the classification accuracy and overall performance, and set the accuracy as weights of the base classifier. For imbalanced massive data, we must not only consider the overall performance, but also the classification accuracy of minority class, so this paper proposes the cost-sensitive weight integration. Cost matrix is as follows.

$$\cos t_{M*\mathrm{M}}=\begin{bmatrix}\cos t(a_1,\mathrm{a}_1) & \cos t(a_1,a_2) & \cos t(a_1,a_3) & ...... & \cos t(a_1,a_M)\\ \cos t(a_2,a_1) & \cos t(a_2,a_2) & \cos t(a_2,a_3) & ...... & \cos t(a_2,a_M)\\ \cos t(a_3,a_1) & \cos t(a_3,a_2) & \cos t(a_3,a_3) & ...... & \cos t(a_3,a_M)\\ ...... & ...... & ...... & ...... & ......\\ \cos t(a_M,a_1) & \cos t(a_M,a_2) & \cos t(a_M,a_3) & ...... & \cos t(a_M,a_M)\end{bmatrix}$$

M is class number, and $\cos t(a_\mathrm{i},a_\mathrm{j})$ is cost for forecasting a_i into a_j, $\cos t(a_\mathrm{i},a_i)=0$.

Test sample $X=\{x_{1,}x_2,x_3,......,x_n\}$, $x_i=\{x_{i1},x_{i2},x_{i3},......,x_{id}\}$, the basic classifier f_i consumes the misclassification cost as follows.

$$\text{mis\_}\cos t_{f_i}=\sum_{i=1}^{n}\cos t(a_i,a_j)\,. \quad (6)$$

The test for sample is also paid, and test cost of sample x_i is as follows.

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

This paper integrates OCS_SVM, RS_SVM and UCS_SVM. The weight of three cost-sensitive SVM is as follows.

$$w_{f_i} = \sum_{j=1}^{n} accurcy(x_i) \Big/ \left(\text{mis\_cost}_{f_i} + test\_\text{cost}_{f_i}\right), i = 1,2,3. \qquad (7)$$

Use OCS_SVM, RS_SVM and UCS_SVM in current training samples, and structure the base classifiers $\left(f_1^1, f_2^1, ..., f_k^1\right)$, $\left(f_{K+1}^2, f_{k+2}^2, ..., f_{2*k}^2\right)$, $\left(f_{2k+1}^3, f_{3k+2}^3, ..., f_{3*k}^3\right)$. Select k classifiers with higher weight for all classifiers to form $\left(f_1^t, f_2^t, ..., f_k^t\right), t \in 1,2,3$.

$$f(x) = \sum_{i=1}^{k} \left(w_{fi} * f_i^t(x)\right). \qquad (8)$$

$x \in R^n$, $f_i^t(x)$ is predicted value of sample x. $f(x)$ is classification result for test sample.

3 Improved KNNModel

Guo proposed KNNModel algorithm, which extends outward by each training centre, and build the multiple models cluster for training set, so that the models cover maximum similar instances without any heterogeneous instances. KNNModel algorithm has higher classification accuracy when the samples are covered by the model clusters, but if the classified samples are boundaries, the classification accuracy is affected. The time complexity of KNNModel is high, and is not suitable for rapid classification of data stream. This paper improves KNNModeld algorithm for the concept drift detection of imbalanced massive data streams.

The training set $X = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i), ..., (x_n, y_n)\}$, $x_i = \langle x_{i1}, x_{i2}, ..., x_{id} \rangle$, $y_i \in \{1, 2, ..., K\}$. Define the class cluster $unba\_class_i^k = \left(type_i^k, center_i^k, class_i^k, Tolerance_i^k, Radius_i^k, Num_i^k\right)$, $unba\_class_i^k$ is the model cluster i with class k. If class k is the minority class, $type_i^k = 1$. Otherwise $type_i = 0$. $Center_i^k = \dfrac{1}{Num_i^k} \sum_{y_i = i} x_i$. $class_i^k = k$ is the class of model cluster, and $Tolerance_i^k$ is tolerability. If the class is minority, $Tolerance_i^k = 0$. In other word, the model clusters of minority class are not different samples. $Radius_i^k$ is radius, and Num_i^k is sample number of model cluster.

Algorithm: Training process of BL_KNNModel

```
Input: Training data set $X$, the majority class $mc \in \{1, 2, ..., K_1\}$, the minority class $lc \in \{1, 2, ..., K_2\}$, the tolerability $\alpha$ for the majority class.
Output: Model cluster of each class.
Begin
For (k=1; k<=K1;k++)
{
Use k-means algorithm to structure three cluster centre $center_1^k$, $center_2^k$, $center_3^k$ for class $k$.
$unba\_class_i^k = \left(type_i^k, center_i^k, class_i^k, Tolerance_i^k, Radius_i^k, Num_i^k\right), i = 1,2,3$.
}
For (k=1;k<=K2;k++)
{
Label all samples into uncovered in class $k$.
Each uncovered sample is centre, and expand it the area, which covers the same class point without the other point. Each model cluster is $unba\_class_i^k = \left(type_i^k, center_i^k, class_i^k, Tolerance_i^k, Radius_i^k, Num_i^k\right), i = 1,2...$
Order degradedly the model clusters according to the samples number.
If the model cluster $unba\_class_i^k$ is covered by $\left\{unba\_class_0^k, unba\_class_1^k, ..., unba\_class_{i-1}^k\right\}$, delete it.
If the number that the model cluster covers is not more than three, delete it.
}
Structure cost-sensitive ensemble Classification.
End.
```

Algorithm 2: Classification process of BL_KNNMode

```
Input: Data streams $DS$, Cost-sensitive ensemble Classification, the model clusters of minority and majority class.
Output: Classification of data streams $DS$.
Begin
Calculate the distance form each test sample to each model centre.
If (The test sample is covered by the model clusters with the same classifier)
Label the test sample as the covered model cluster.
else if (The test sample is covered by different classifiers)
Label the classifier, which is the nearest form the test sample with the model centre.
else use cost-sensitive ensemble Classification.
```

BL_KNNModel runs K times k-Means cluster for structuring the model, and time complexity of k-Means cluster is $O(n_k * Num)$, and Num is cluster number, so the total complexity is $O\left(\max(k_1, k_2) * n_k * Num\right)$. $\max(k_1, k_2)$ and Num are free-running with n_k, $Num \ll n_k$, $\max(k_1, k_2) \ll n_k$, so the time complexity of BL_KNNModel is $O(n)$, and it is suitable for massive data streams classification.

4 Modules for concept drift detection

Concept drift is the characteristics of the data streams, and the overall data distribution changes over time, so that the accuracy of classification models is lower. According to concept drift problem, how to design to data streams classification model has become a hot research. At present, there are ways of instance selection, weighted instance and integration learning. This paper improves BL_KNNModel algorithm, if the density of minority class and majority class change, the concept drift occurs and the classification models should been updated. The existing concept drift detection ways use the fixed length window, but how to set the window length is difficult. The window is too long, and it is not sensitive to concept drift. On the other hand, the window is too short, and it cannot contain the enough new instances that update the model, so this paper proposes the up-and-down window. BL_KNNModel detects the massive data stream and finds the instances that are covered by any model cluster, if the uncovered instance number adds up to the fixed value $window\_length$, all current instances constitute the up-and-down windows. Detect the probability density of each model cluster, and if the differential density with previous windows exceeds the fixed threshold, the concept drift occurs.

Algorithm 3: Concept drift detection algorithm.

Input: Massive data streams DS, the probability density for each previous model cluster, the fixed $window\_length$ for the uncovered instance number, the probability density threshold τ_1 of minority class, the probability density threshold τ_2 of majority class.

Output: Occurrence of concept drift

Begin

Step 1: Run repeatedly BL_KNNModel for data streams DS, until the uncovered instance number adds up to the fixed value $window\_length$.

Step 2: Calculate the probability density of each current model cluster.

Step 3: Compare the probability density of the same model cluster in current and previous windows, if the difference exceeds the fixed threshold, the concept drift occurs.

End.

5 Cost-sensitive ensemble classifier for imbalanced data streams

The descriptions of Cost-sensitive Ensemble Classification for Imbalanced Massive Data Streams are as follows.

Input: Imbalanced Massive Data Streams $S=\{S_1,S_2,...,S_j,...\}$, the BL_KNNModel model clusters and cost-sensitive weight integration of classifier, the previous windows, the uncovered instance number $window\_length$

Output: Classification of $S_1,S_2,...,S_j,...$

```
Begin
i=0; l=0;
while (i<= $window\_length$)
{
Use algorithm BL_KNNModel to classify an sample $S_i$.
l++;
If ( $S_i$ is uncovered by any model)
i++;
}
Use algorithm 3 to test whether $\{S'_1,S'_2,...,S'_l\}$ occurs concept drift.
If (Concept drift occurs)
Use BL_KNNModel to reconstruct the model that occurs concept drift, and the cost-sensitive weight integration of classifiers are also rebuilt.
Use rebuilt BL_KNNModel to classify $\{S'_1,S'_2,...,S'_l\}$.
```

6 Simulation experiment

6.1 DATA SET

This paper selects the synthetic and real datasets. In view of rotating hyper plane, the synthetic datasets with concept drift are composed by the modified MOA Task Launcher. The sample $X=\{x_1,x_2,...,x_d,\}$. $\sum_{i=1}^{d} a_i x_i = a_0$, $a_0 = r\sum_{i=1}^{d} a_i$. a_i is weight number, $r\left(r \neq \frac{1}{2}\right)$ is gradient of slope of imbalanced datasets. This paper sets a series of parameters for concept drift in data streams. The parameter t is the change size of weight number a_i, The parameter $s_i \in \{-1,1\}(1 \le i \le k)$ is the change direction. When a new sample is produced, a_i is adjusted by $s \bullet t$. The dimensionality for data streams is 50, the attribute varies with time, and the imbalance ratio is 0.05.

Use KDDCUP'99 as the real datasets. This paper chooses randomly Normal, Neptune and smurf as the majority, and they are negative and share 99.95%. The connection type of Land, Rootkit, Warezmaster are selected as the minority, and they are positive and share 0.5%. Sample randomly KDDCUP as the second experiment data streams which is named as Stream_KDD.

6.2 EVALUATION METRICS FOR IMBALANCED DATA STREAMS

In general, the overall accuracy is often used to measure the performance for classification model. However, the classification accuracy of the negative samples in unbalanced data streams is greater influence than the positive. Therefore, the classification accuracy is not suit for unbalanced data streams. TP is the positive samples number that are correctly classified and TN is negative. FP is the positive samples number that are improperly classified and FN is negative. This paper gives the performance evaluation as follows.

$$recall = \frac{TP}{TP+FN}, \qquad (9)$$

$$Precision(\Pr) = \frac{TP}{TP+FP}, \qquad (10)$$

$$G-Mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}, \qquad (11)$$

$$F-measure(Fm) = \frac{(1+\beta)^2 \times recall \times precision}{\beta^2 \times recall + precision}, \qquad (12)$$

$$TPR = \frac{TP}{TP+FN}, \qquad (13)$$

$$TNR = \frac{TN}{TN+FP}, \qquad (14)$$

$$Mean\ Squared\ Error(MSE) = \frac{1}{|T|}\sum_{x_i \in T}\left(f(x_i) - p(+|x_i)\right)^2 . \qquad (15)$$

T is test sample datasets, and $f(x_i)$ is classification result of sample x_i. $p(+|x_i)$ is real posterior probability of sample x_i.

6.3 RESULT OF EXPERIMENTS

For test the efficiency of CECIDS, the test environment is Pentium D3.5GHz CPU, 8GM RAM windows 7+64bit, and select Weka+JDK6.0 as the development environment. The parameters of all algorithms refer to the papers [17, 19], and the results of synthetic datasets with concept drift are as follows. CECIDS that this paper proposes compares with RDFCSDS [16], SS [18], and OACLNIS [17], which are the other classification algorithms for imbalanced data streams.

TABLE 1 Results of synthetic datasets with concept drift

Algorithm	TPR	TNR	G-mean	Pr	Fm	MSE
SS	0.0067	0.9987	0.0923	0.9032	0.0107	0.0215
RDFCSDS	0.0246	0.9966	0.1235	0.9223	0.0345	0.0427
OACLNIS	0.0642	0.9952	0.2445	0.9465	0.1854	0.0564
CECIDS	0.0723	0.9941	0.3384	0.9668	0.2365	0.0736

The results of Stream_KDD datasets are as follows.

TABLE 2 Results of Stream_KDD datasets

Algorithm	TPR	TNR	G-mean	Pr	Fm	MSE
SS	0.0008	0.9967	0.0848	0.9254	0.0144	0.3564
RDFCSDS	0.0045	0.9943	0.1437	0.9365	0.0383	0.5324
OACLNIS	0.0093	0.9925	0.1865	0.9523	0.1624	0.6384
CECIDS	0.0102	0.9906	0.2873	0.9786	0.2413	0.8414

Form the table 1 and 2, for the evaluating indicator of TPR, G-mean, Precision, F-measure and MSE, CECIDS is higher than the other algorithms for imbalanced massive data streams, and the evaluating indicator TNR in CECIDS is lower, so CECIDS algorithm is more excellent than the other.

The classification accuracy is not suitable for imbalanced massive data streams, and this paper selects G-mean and F-measure as integrated learning curve for imbalanced massive data streams. The integrated learning curve for imbalanced massive data streams in the synthetic datasets is as follows in figure 1 and figure 2.

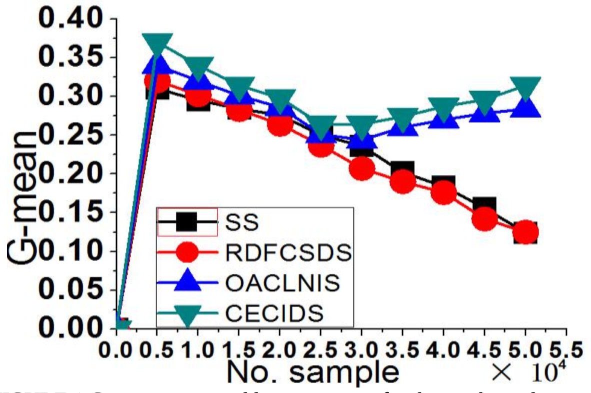

FIGURE 1 G-mean integrated learning curve for the synthetic datasets

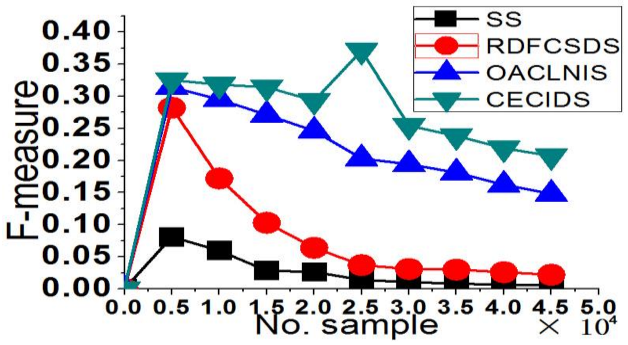

FIGURE 2 F-measure integrated learning curve for the synthetic datasets

The integrated learning curve for imbalanced massive data streams in the Stream_KDD datasets is as follows in figure 3 and figure 4.

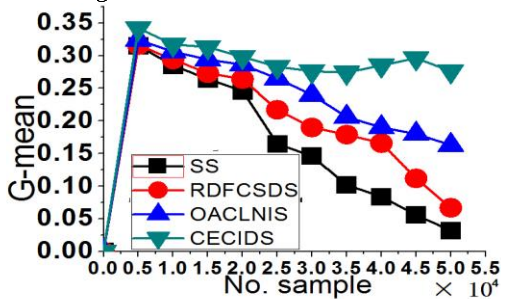

FIGURE 3 G-mean integrated learning curve for Stream_KDD datasets

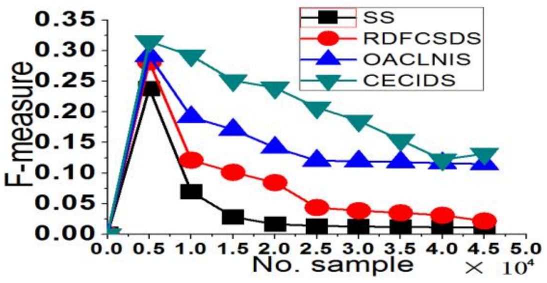

FIGURE 4 F-measure integrated learning curve for Stream_KDD datasets

From figure 1 to 4, when the data streams increase gradually, G-mean and F-measure integrated learning curve of CECIDS for synthetic and Stream_KDD datasets are more effective than the other classifiers for imbalanced data streams. Almost all the experiments validate that CECIDS that this paper proposes are more excellence than the other classification algorithms for imbalanced data streams.

7 Conclusions

The classification of concept drifting data streams has been a hot research, and this paper proposes the approach of the cost-sensitive ensemble classification for imbalanced massive data streams. A new classifier method BL_KNNModel based on KNNModel algorithm is given, which can detect the concept drift streams by using the variable windows size. By constructing the different model cluster for every class, if the concepts drift occurs on the model cluster, it only needs to rebuild the corresponding classification model. The experiments on the synthetic and real datasets prove the effectiveness of the algorithms that this paper proposes. The next work focuses on trying to adopt a new measure to calculate the distance between two samples, and distinguishes effectively the noise and concept drift.

Acknowledgments

This work is supported by the science and technology projects of the Shandong province universities (No. J14LN65, No. J13LN53). This paper comes from the research fund of Heze University (No. XY12KJ03).

References

[1] Aboalsamh H L 2009 A novel incremental approach for stream data mining *AEJ-Alexandria Engineering Journal* **48**(4) 419-26
[2] Hashemi S, Yang Y 2009 Flexible decision tree for data stream classification in the presence of concept change, noise and missing values *Data Mining and Knowledge Discovery* **19**(1) 95-131
[3] Tang Y, Zhang Y Q, Chawla N V 2009 SVMS modeling for highly imbalanced classification *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **39**(1) 281-8
[4] Kuncheva L I, Zliobaite I 2009 On the window size for classification in changing environments *Intelligent Data Analysis* **13**(6) 861-72
[5] Liang C, Zhang Y, Shi P, et al. 2012 Learning very fast decision tree from uncertain data streams with positive and unlabeled samples *Information Sciences* **213**(5) 50-67
[6] Amin M F, Islam M M, Murase K 2009 Ensemble of single-layered complex-valued neural networks for classification tasks Neurocomputing **72**(10-12) 2227-34
[7] Avci E 2009 Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithm-support vector machines: HGASVM *Expert Systems with Applications* **36**(2) 1391-402
[8] Bu N, Okamoto M, Tsuji T 2009 A Hybrid Motion Classification Approach for EMG-Based Human-Robot Interfaces Using Bayesian and Neural Networks *IEEE Transactions on Robotics* **25**(3) 502-11
[9] Chen T C, Tsao H L 2009 Using a hybrid meta-evolutionary rule mining approach as a classification response model *Expert Systems with Applications* **36**(2) 1999-2007
[10] Chi M M, Kun Q, Benediktsson J A, et al. 2009 Ensemble Classification Algorithm for Hyperspectral Remote Sensing Data *IEEE Geoscience and Remote Sensing Letters* **6**(4) 762-6
[11] Dembczynski K., Kotlowski W, Slowinski R 2009 Learning Rule Ensembles for Ordinal Classification with Monotonicity Constraints *Fundamenta Informalicae* **94**(2) 163-78
[12] Hashemi S, Yang Y, Mirzamomen Z, et al. 2009 Adapted One-versus-all decision trees for data stream classification *IEEE Transactions on Knowledge and Data Engineering* **21**(5) 624-37
[13] Huang C L 2009 ACO based hybrid classification system with feature subset selection and model parameters optimization *Neurocomputing* **73**(1-3) 438-48
[14] Adler W, Brenning A, Potapov S, et al. 2011 Ensemble classification of paired data *Computational Statistics & Data Analysis* **55**(5) 1933-41
[15] Baumgartner D, Serpen G 20122 A design heuristic for hybrid classification ensembles in machine learning *Intelligent Data Analysis* **16**(2) 233-46
[16] Gao J, Ding B L, Fan W, Han J W, Philip S Y 2008 Classifying data streams with skewed class distributions and concept drifts *IEEE Internet Computing* **12**(6) 37-49
[17] Adel G, Reza M, Hadi S Y 2013 Ensemble of online neural networks for non-stationary and imbalanced data streams *Neurocomputing* **122** 535-44
[18] Liu P, Wang Y, Cai L J, Zhang L B 2020 Classifying skewed data streams based on reusing data *In Computer Application and System Modeling (ICCASM) of 2010 International Conference* **4** V4-90-V4-93
[19] Guo G D, Li N, Chen L F 2014 Concept Drift Detection for Data Streams Based on Mixture Model *Journal of Computer Research and Development* **51**(4) 731-42

Author

Yuwen Huang, born in 1978 at Shanxian

Current position, grades: lecturer at the Department of Computer and Information Engineering, Heze University.
University studies: Master of Engineering in Computer Science from the "Guangxi Normal university" in 2009.
Scientific interest: the data-mining, intelligence Calculation.