ATLANTIS PRESS

Advances in Engineering Research, volume 118

2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2017)

A Tree-based Concept Drift Detection Method by Three-way Decisions

Baohe Su[1, a]

[1] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts & Telecommunications, Chongqing 400065, China;

[a]subaohe@qq.com

Keywords: concept drift; three-way decisions; concept tree; data stream; uncertain factors.

Abstract. Concept drift detection is an active research area in data stream mining. The existing concept drift detection methods usually determine a concept is drifted or not drifted. In other words, the concepts are assigned into two classes such as drifted and un-drifted, which is typically based on two-way decisions. Most likely, these methods incorrectly determine that concept drift occurs due to some uncertain factors such as noise, and some real concept drifts are not detected. Inspired by the three-way decision theory, we propose a tree-based concept drift detection method by three-way decisions in this paper, in order to improve the accuracy of detection. The basic idea of the method is to assign the concepts into three classes such as drifted, nondeterministic drifted and un-drifted. Furthermore, concepts in the class of nondeterministic drifted are determined further according to the deviation between the classification error rates. The results of comparison experiments show the effectiveness and efficiency of the proposed methods.

1. Introduction

Massive data is produced at unprecedented rate in the form of a data stream in many applications. Examples of data streams include computer network traffic, phone conversations, ATM transactions, web searches, business data and sensor data. Accommodating large volumes of streaming data in the machine's main memory is impractical and often infeasible. Hence, the online processing is suitable. In this case, predictive models can be trained either incrementally by continuous update or by retraining using recent batches of data[1]. During the process of learning, changes in the hidden context can induce more or less radical changes in the target concept, which is generally known as concept drift[2]. After the occurrence of concept drift, the model obtained in prophase cannot handle the current data correctly. Hence, it is inevitable for classifiers to adapt to such changes.

There are several methods in machine learning to deal with changing concepts. In general, approaches to cope with concept drift can be classified into two categories: i) approaches that adapt a learner at regular intervals without considering whether changes have really occurred; ii) approaches that first detect concept changes, and next, the learner is adapted to these changes[3]. In the former approach, one method is learning new examples continuously using fixed size sliding window, forgetting old examples and reconstructing the classification model based on data included in the sliding window. Another method is using an incremental learning or online learning algorithm, making adjustment of classifiers according to the effect of classification on the new examples. Models with the former strategy forget old data in a fixed speed, which is too slow when concept drift occurs; however, the strategy is easy to apply. For the latter approach, by detecting concept drift periodically, the models can adjust classifiers with the latest examples when concept drift happens, forgetting old data faster and adapting to concept drift better. However, the performance would decline if concept drift detection makes misjudgments; this kind of models could not adapt to concept drift timely if the detection makes omissions. Thus, the objective of this paper is trying to find a way to improve the performance of detection regarding accuracy and delay times.

In the existing researches, some algorithms such as DDM[3] and EDTC[4] algorithm do not make further processing when it is hard to judge whether concept drift occurs in the data stream. In other words, the concepts are viewed as two classes such as drifted and un-drifted, which is typically based on two-way decisions. Most likely, these methods incorrectly determine that concept drift occurs due

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

to some uncertain factors such as noise, and some real concept drifts are not detected. Obviously, it is a typical uncertain problem. On the other hand, a theory of three-way decisions is proposed as an extension of the commonly used binary-decision model with an added third option[6]. In the last few years, we have witnessed a fast growing development and applications of three-way approaches in areas of decision making[7], email spam filtering[8], three-way investment decisions[9], three-way cluster analysis[10, 11] and many others[5]. Their preliminary results provide us with a tool for studying the problem of detecting concept drift. In this paper, we propose a tree-based concept drift detection method by three-way decisions, shorted by TCDDM-TWD, reducing the delay time and the omission rate, improving the performance of detection.

2. Related Work

We present in this section a short review of existing approaches on detecting concept drift in a data stream.

On one hand, in the approaches based on classifiers for detection of concept drifts, the decision tree is one of the most popular tools due to its advantage in inducing rules from empirical data. There are some achievements based on decision trees. For example, Hulten et al.[12] proposed the concept-adapting very fast decision tree learner(CVFDT) algorithm based on the single decision tree model in 2001; the algorithm stays current while making the most of old data by growing an alternative subtree whenever an old one becomes questionable, and replacing the old with the new when the new becomes more accurate. Street and Kim[13] proposed a streaming ensemble algorithm (SEA), which uses new classifiers instead of the unnecessary old ones according to some quality criterion while maintaining a certain number of classifiers. Wang et al.[14] proposed the AWE which is also an ensemble method, the algorithm weights the classifiers judiciously based on their expected classification accuracy on the test data under the time-evolving environment. Kolter and Maloof proposed the dynamic weighted majority algorithm[15] and the additive expert ensemble algorithm[16] for online learning in succession. Brzezinski and Stefanowski[17] proposed the online accuracy updated ensemble (OAUE) algorithm, which uses the proposed function to incrementally train and weight component classifiers.

On the other hand, it is a common way to detect the concept drift by tracking the performance of classifiers on new example set. Namely, the concept drift occurs in new example set when the performance has a larger decline. Thus, some detection methods are presented. Widmer and Kubat[18] proposed the FLORA series algorithm, which adjusts the sliding window size according to the coverage of positive examples and classification accuracy. Gama et al.[3] proposed the drift detection method (DDM), which is applied to problems where the information is available sequentially over time. Mouss et al.[19] use the Page-Hinkley test detection method to detect the concept drift, the method can effectively detect the mean change of sequence data in Gaussian distribution. ROSS et al.[20] proposed an approach based on an exponentially weighted moving average (EWMA) chart[21] to monitor the misclassification rate of a streaming classifier.

From the review of existing studies, we can identify that these methods do not pay enough focus on the inherent uncertain factors in the data. However, the uncertainty is inevitable in many applications due to various factors such as the limitations of measuring equipment and delays in data updates[22]. For example, the data collected from wireless sensors will be affected by many random factors such as the change of temperature, the change of air humidity and the communication barriers. In the radio frequency identification (RFID) streams, the accuracy of collected data is difficult to guarantee due to the limitation of energy, cost and other factors. Practically, the observed read rate in real-world RFID deployments is often in the 60% ~70% range; in other words, over 30% of the tag readings are routinely dropped[23]. Unfortunately, these factors are difficult to avoid, leading to generation of noisy data in data stream.

We can discuss the problem of detecting concept drift in a view of a problem of decision making. When making decisions, people usually use reliable data and information to judge. In this scenario, the results of detection are viewed as two classes such as drifted and un-drifted, which is a typical

two-way decisions. In fact, as we have discussed above, uncertain is common in data streams. It is not entirely reasonable to make binary-decisions on the uncertain information. Thus, Professor Yao[6] proposed the three-way decision theory to overcome some drawbacks of binary-decisions. The essential ideas of three-way decision theory is that to divide a universe into three disjoint parts and process the different part using different strategy[5]. Therefore, in this paper, we use the three-way decision theory to deal with the uncertainty of detecting concept drift and we are able to make new and unique contributions to concept drift detection and applications.

3. The TCDDM-TWD Concept Drift Detection Method

The algorithm framework of this paper is shown in figure1. The algorithm first generates the initial concept tree, which is the decision tree classifier proposed in this paper. And then run TWCDD algorithm to detect whether the concepts drift, and adjust the tree to adapt to the latest concepts. In fact, this paper proposes a method to generate an initial concept tree which is similar to HoeffdingTree[25] using examples in the data stream, which is called algorithm1.

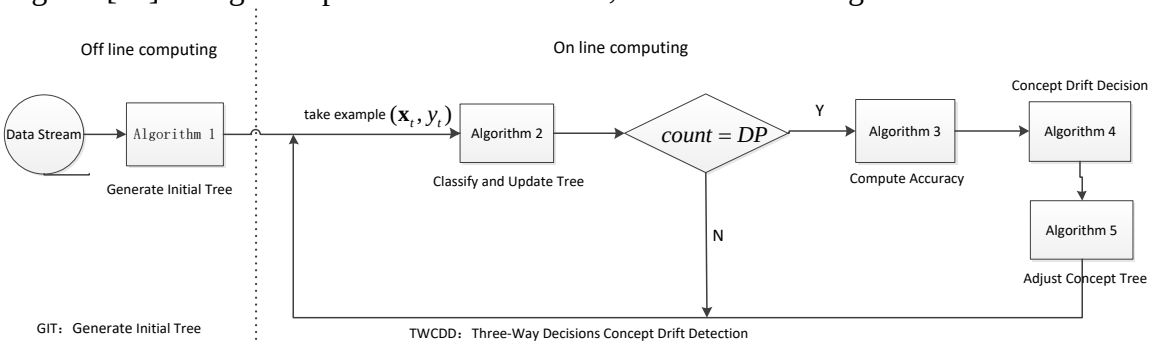

Fig. 1 Tree-based Concept Drift Detection Method by Three-way Decisions

3.1 Generate an Initial Concept Tree.

Assume a data stream S appears incrementally as a sequence of labeled examples $(\mathbf{x}_t, y_t)(t = 1, 2, ..., T)$, where $\mathbf{x}_t$ is a vector of attribute values and y_t is a class label. To learn the latest concepts of the data stream, this paper uses SW (the size of sliding window) examples to generate the initial concept tree.

First, a node is initialized as the root node of the concept tree, and then a tree is generated from the root node. Examples are obtained one by one from the data stream until the number of them reaches $n_{\min}$, and the root node records the statistics of them and stores in a form n_{ijk}, which represents the number of examples who has the j th value on the i th attribute and the k class label. Then choose splitting attribute for the root node according to the information gain[24] and Hoeffding inequality[25]. Assuming that the splitting attribute is a nominal attribute, for each value of the splitting attribute, grow an empty leaf node as child node.

Then get one example from data stream. Starting from the root node, test the example's value on the splitting attribute of the internal node, and pass it down to one child node along the branch corresponding to the value, continuously until the child node is a leaf node. All nodes traversed in the process record the statistic of the example. If there are sufficient statistics, namely, the number of examples seen at the leaf node reaches the threshold $n_{\min}$, choose splitting attribute for the leaf node according to information gain and Hoeffding inequality. Then replace the leaf node with an internal node that split on the attribute. For each value of the splitting attribute, grow an empty leaf node as child node. Repeat this process until $t = SW$ when an initial concept tree is generated.

3.2 Three-way Decisions Concept Drift Detection Method

This section describes the TWCDD algorithm. In the last part we use examples in the data stream to get the initial concept tree. Next, the TWCDD algorithm is continuously running to decide whether concept drift occurs for the concept tree and adjust the concept tree. TWCDD can be divided into 3 steps:

Step1: TWCDD takes examples from the data stream. In algorithm 2, the concept tree is used to classify the examples and update the concept tree. The number of processed examples is represented by a counter *count* . When *count* is less than a numeric value *DP* , repeat the procedure. In algorithm2, $(\mathbf{x}_t, y_t)$ is classified firstly, and the number of misclassification at the leaf node is updated by comparing whether the prediction class and y_t are consistent. The example is then added to the sliding window sequence.

Step2: Directly decide whether the concept of the whole concept tree drifted is not accurate enough, therefore, in this paper, decide whether the concept of each subtree drifted. The classification accuracy of each subtree is calculated in algorithm3. Then in the algorithm4 decide whether the concept of each subtree drifted. According to the classification accuracy of the subtrees, the root nodes of the subtrees will be divided into L domain, R domain, M domain, respectively subtrees with concept not drift, subtrees with concept drifted, subtrees with concept may drift.

Step3: After concept drift decision, process the subtrees whose concept drifted. Replace the subtrees with empty leaf nodes. After the replacement, new subtrees are learned with examples of the sliding window. And then set *count* to 0.

3.2.1 Classify and Update Concept Tree

The following is the process of the concept tree classifying example $(\mathbf{x}_t, y_t)$. Starting from the root node test the example's value on the splitting attribute of the internal node, and pass it down to one child node along the branch corresponding to the value, continuously until the child node is a leaf node. All nodes traversed in the process update n_{sum} , the total number of seen examples; n_c , the number of examples seen in the detection; n_{ijk} , the statistics of examples. Then compare the prediction class of the leaf node with y_t . If they are not the same, the misclassification number in the detection of the leaf node is updated.

The next step is to test whether the leaf node satisfies the splitting condition and split the leaf node according to Hoeffding inequality and the information gain if it satisfies.

Then a tuple contain the example and the max *id* , the sequence number of nodes, is added into the sliding window. If the number of examples in the window is greater than *SW* , take the oldest example, delete the statistics of the example from the concept tree along the path of classification. Finally remove it from the sliding window.

3.2.2 Compute Accuracy.

The last part describes the process of classifying examples and updating the concept tree. When $count = DP$, compute the classification error rate of each subtree. Then decide whether the concept of each subtree drifted. The following describes the process of calculating the classification error rate. Assume N_r is the root node of a subtree, the classification error rate $p_t = \frac{\sum n_{error}}{n_c}$. $\sum n_{error}$ is the sum of n_{error} of all leaf nodes in the subtree. n_c is the number of recorded example information in the detection of N_r . The classification error rate and its standard deviation are calculated in algorithm3. In the process, $\sum n_{error}$ of a parent node is the sum of $\sum n_{error}$ of all child nodes, thus the classification error rate of the subtrees is calculated in order of postorder traversal.

3.2.3 Concept Drift Decision

Concept drift decision is divided into two steps: preliminary division and further division. The preliminary division is that decide whether the concept of each subtree drifted according to a determine threshold and a warning threshold, then the root nodes of the subtrees whose concept not drift are divided into the L domain and will be added to the set *FD*; the root nodes with concept drifted are divided into the R domain and will be added to the set *TD*; the root nodes with concept may drift are divided into the M Domain.

Further division is that for some nodes are divided into M domain, which are also in M domain before the detection, they may belong to gradual concept drift. Therefore, in this paper, the deviation

of the classification error rate of the nodes in M domain is cumulated in every detection. When a node's cumulative sum reach a threshold, it will be transferred to R domain, join TD; or else, continue to divide into M domain, added to the set MD.

Before the operation of TWCDD algorithm, FD, MD, TD are initialized to the empty set. After each detection, FD, TD are set to the empty set.

The Setting of Thresholds In Decision. The following is the setting process of thresholds in decision. For each node on the concept tree, thresholds are set independently. The setting process of thresholds in a node N_r is to be introduced. Assume the concept drift decision begin at t.

The setting process of thresholds in preliminary division. Assume the concept of data stream does not drift, the classification error rate of the classifier is approximately by a Normal distribution. If $p_t + \sigma_t$ increasing reaches a certain level of the original distribution, the classifier's performance is decreased and decide the concept drifted. Set a determine threshold β. If $p_t + \sigma_t \geq \beta$, then decide the concept of the subtree drifted. Set a warning threshold α. If $\alpha \leq p_t + \sigma_t < \beta$, then decide the concept of the subtree may drift; if $p_t + \sigma_t < \alpha$, then decide the concept of the subtree not drift. α and β is respectively set to $p_{\min} + 1.5 * \sigma_{\min}$ and $p_{\min} + 3 * \sigma_{\min}$. $p_{\min}$ and $\sigma_{\min}$ is set to the values when $p_t + \sigma_t$ reaches its minimum value during the process.

The setting process of threshold in further division. For each nodes in the M domain, accumulate the deviation between $p_t + \sigma_t$ and α, $\eta = \eta + (p_t + \sigma_t - \alpha)$. Set a deviation threshold γ. If a node belongs to PM and $\eta > \gamma$, decide the concept drifted. In this paper, γ is set to $2 * \sigma_{\min}$.

The Process of Concept Drift Decision. After compute the classification error rate, decide the concept of each subtree in concept drift decision. Before concept drift decision, if MD is not empty, nodes in M domain in the last detection will be stored with an empty set PM. Then the root nodes of subtrees are divided into three domains according to classification error rate and thresholds. After the division, check whether to update thresholds. If $p_t + \sigma_t$ is lesser than $p_{\min} + \sigma_{\min}$, update $p_{\min}$, $\sigma_{\min}$ and the three thresholds; if the node is divided into R domain, that is, the concept drifted, set $p_{\min}$, $\sigma_{\min}$ and the three thresholds too.

Algorithm4: Concept Drift Decision
Input: HT, the concept tree; set FD, MD, TD
Output: HT, the concept tree; set FD, MD, TD
Assume that is the d th concept drift decision and begin at $t = u$.
Add all the internal nodes in HT to a set W, $PM = \varnothing$;
if $d > 1$ and $MD \neq \varnothing$ **then**
{
$PM = PM \cup MD$;
$MD = \varnothing$;
}
for each node N_r in W **do**
{
if $p_u(N_r) + \sigma_u(N_r) < \alpha(N_r)$ **then**
{
$FD = FD \cup \{N_r\}$;
$\eta(N_r) = 0$;
}**else if** $p_u(N_r) + \sigma_u(N_r) \geq \beta(N_r)$ **then**
{
$TD = TD \cup \{N_r\}$;
$\eta(N_r) = 0$;
}**else**
{
// nodes in M domain
$\eta(N_r) = \eta(N_r) + (p_u(N_r) + \sigma_u(N_r) - \alpha(N_r))$;
if $PM \neq \varnothing$ **then**
{

if $N_r \in PM$ and $\eta(N_r) > \gamma(N_r)$ **then**
{
$TD = TD \bigcup \{N_r\}$;
$\eta(N_r) = 0$;
}
else $MD = MD \bigcup \{N_r\}$;
}
else $MD = MD \bigcup \{N_r\}$;
}
if $p_u(N_r) + \sigma_u(N_r) < p_{\min}(N_r) + \sigma_{\min}(N_r)$ or $N_r \in TD$ **then**
{
$p_{\min}(N_r) = p_u(N_r)$;
$\sigma_{\min}(N_r) = \sigma_u(N_r)$;
$\alpha(N_r) = p_{\min}(N_r) + 1.5 * \sigma_{\min}(N_r)$;
$\beta(N_r) = p_{\min}(N_r) + 3 * \sigma_{\min}(N_r)$;
$\gamma(N_r) = 2 * \sigma_{\min}(N_r)$;
}
}
return HT, FD, MD, TD;

3.2.4 Adjust the Concept Tree.

After the completion of concept drift decision, run the algorithm5 to deal with subtrees whose concept drifted. Replace the subtrees with empty leaf nodes. After the replacement, the examples in the sliding window are passed down along the path of classification to the replacement leaf nodes, generating new subtrees.

4. Experiment and Result Analysis

In this paper, we do experiments on artificial datasets and real datasets. Artificial datasets are LED dataset and SEA dataset, which are generated by the MOA[26]platform. Real datasets are Poker-Hand dataset and Electricity dataset[27], which are common in contrast experiment in concept drift researches.

In the LED dataset, concept drift occurs at 50000th example. As Table1 presents, in the twice experiment, TCDDM-TWD is one of the two algorithms which have the highest classification accuracy and lowest range of decline. TCDDM-TWD has lower decrease than AUE and OAUE, and can restore to normal quickly when concept drift occurred.

Table 1 The results of the comparison experiments of LED datasets

Algorithm	D1			D2		
	least accuracy	average accuracy	decline	least accuracy	average accuracy	decline
TCDDM-TWD	**43.40%**	68.27%	**24.87%**	**40.20%**	67.72%	**27.52%**
EWMA	**69.60%**	69.12%	**0**	**58.40%**	68.94%	**10.54%**
AUE	19.90%	72.84%	52.94%	31.00%	72.89%	41.89%
OAUE	39.10%	73.20%	34.10%	44.30%	73.17%	28.87%

In the SEA dataset, concept drift occurs at 25000th, 50000th, 75000th example. As Table2 presents, on the SEA dataset, TCDDM-TWD performs better than AUE and OAUE on three concept drift of D3 dataset and the third concept drift of D4. And on the first and second concept drift of D4 dataset, TCDDM-TWD is inferior to OAUE slightly.

Table 2 The results of the comparison experiments on least classification accuracy of SEA datasets

Algorithm	D3			D4		
	1	2	3	1	2	3
TCDDM-TWD	**79.00%**	**80.30%**	**82.30%**	80.50%	79.20%	**80.80%**
EWMA	**82.40%**	**81.30%**	**82.40%**	**82.40%**	**81.60%**	**81.20%**
AUE	78.40%	78.10%	78.50%	80.70%	79.90%	80.60%
OAUE	78.40%	79.00%	78.60%	**80.90%**	**80.00%**	80.40%

As Fig2 and Tabel3 presents, on the Poker-Hand dataset, TCDDM-TWD algorithm proposed in this paper is well than other algorithms on classification accuracy. Meanwhile, TCDDM-TWD can detect the change in data streams earlier than OAUE and EWMA. For example, at 320000th-365000th, 485000th-530000th, 665000th-7100000th examples, when the classification accuracy of other classifiers declined, TCDDM-TWD keep the accuracy even have better performance, adapting to new example distribution faster, implying its capacity to detect concept drift earlier.

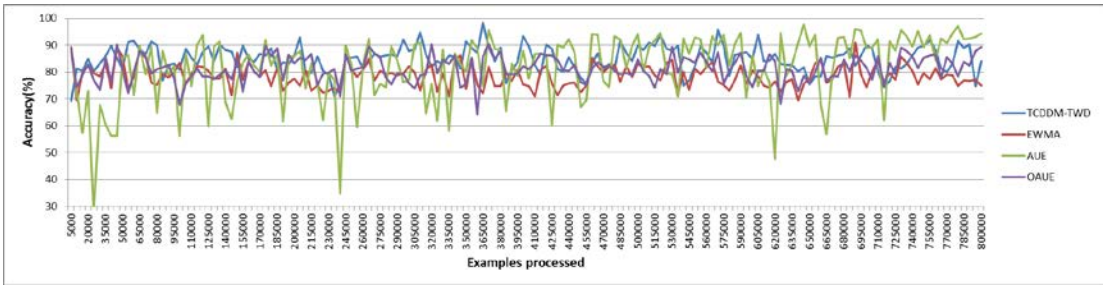

Fig. 2 Poker-Hand dataset, 800000 examples

Table 3 The experiment results on average classification accuracy of real datasets

Algorithm	Poker-Hand	Electricity
TCDDM-TWD	**84.68%**	**86.91%**
EWMA	78.64%	86.45%
AUE	81.85%	77.34%
OAUE	81.33%	**87.71%**

As Fig3 describes, in Electricity dataset, TCDDM-TWD algorithm is well than EWMA and AUE on classification accuracy. At 6000th, 13000th examples, when classification accuracy decline, TCDDM-TWD can adjust the classifier with lesser delay than EWMA, making classification accuracy pick up to some degree. As to OAUE algorithm, when the numerical attributes is much enough, ensemble classifiers can obtain well performance than single classifier.

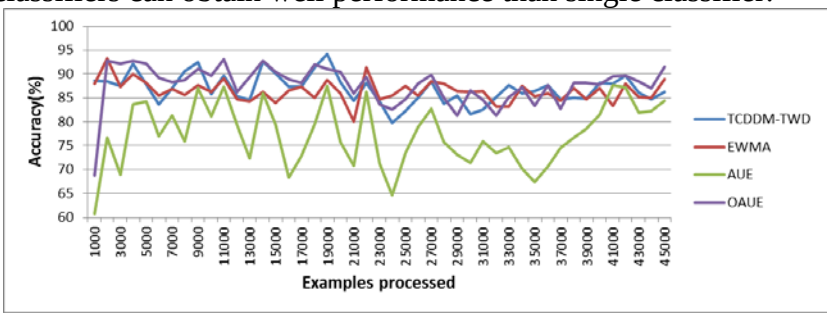

Fig. 3 Electricity dataset, 45312 examples

5. Conclusions

Through conducting contrast experiments on artificial and real datasets, it shows that the proposed algorithm can detect concept drift earlier, improving the classification accuracy of classifiers. Especially on real datasets which are susceptible to uncertain factors, the algorithm is superior to most contrast algorithms. In this paper, three-way decisions concept drift detection method is proposed, extending the application of the three-way decision theory, which has reference value for its application in other fields. The deviation threshold is set according to the empirical value in this paper. In our future work, we will make further research on the automatical setting of the deviation threshold.

6. References

[1] Gama J, Žliobaitė I, Bifet A, et al. A survey on concept drift adaptation[J]. ACM Computing Surveys (CSUR), 2014, 46(4): 44.

[2] Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts[J]. Machine learning, 1996, 23(1): 69-101.

[3] Gama J, Medas P, Castillo G, et al. Learning with drift detection[C]//Brazilian Symposium on Artificial Intelligence. Springer Berlin Heidelberg, 2004: 286-295.

[4] Li P, Wu X, Hu X, et al. Learning concept-drifting data streams with random ensemble decision trees[J]. Neurocomputing, 2015, 166: 68-83.

[5] Yu H, Wang G, Hu B, et al. Methods and Practices of Three-Way Decisions for Complex Problem Solving (in Chinese). Beijing: Science Press; 2015.

[6] Yao Y. An outline of a theory of three-way decisions[C]//International Conference on Rough Sets and Current Trends in Computing. Springer Berlin Heidelberg, 2012: 1-17.

[7] Yao J T, Azam N. Web-based medical decision support systems for three-way medical decision making with game-theoretic rough sets[J]. IEEE Transactions on Fuzzy Systems, 2015, 23(1): 3-15.

[8] Zhou B, Yao Y, Luo J. Cost-sensitive three-way email spam filtering[J]. Journal of Intelligent Information Systems, 2014, 42(1): 19-45.

[9] Liang D, Liu D. Systematic studies on three-way decisions with interval-valued decision-theoretic rough sets[J]. Information Sciences, 2014, 276: 186-203.

[10] Yu H, Liu Z, Wang G. An automatic method to determine the number of clusters using decision-theoretic rough set[J]. International Journal of Approximate Reasoning, 2014, 55(1): 101-115.

[11] Yu H, Zhang C, Wang G. A tree-based incremental overlapping clustering method using the three-way decision theory[J]. Knowledge-Based Systems, 2016, 91: 189-203.

[12] Hulten G, Spencer L, Domingos P. Mining time-changing data streams[C]//Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2001: 97-106.

[13] Street W N, Kim Y S. A streaming ensemble algorithm (SEA) for large-scale classification[C]//Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2001: 377-382.

[14] Wang H, Fan W, Yu P S, et al. Mining concept-drifting data streams using ensemble classifiers[C]//Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003: 226-235.

[15] Kolter J Z, Maloof M A. Dynamic weighted majority: A new ensemble method for tracking concept drift[C]//Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. IEEE, 2003: 123-130.

[16] Kolter J Z, Maloof M A. Using additive expert ensembles to cope with concept drift[C]//Proceedings of the 22nd international conference on Machine learning. ACM, 2005: 449-456.

[17] Brzezinski D, Stefanowski J. Combining block-based and online methods in learning ensembles from concept drifting data streams[J]. Information Sciences, 2014, 265: 50-67.

[18] Widmer G, Kubat M. Effective learning in dynamic environments by explicit context tracking[C]//European Conference on Machine Learning. Springer Berlin Heidelberg, 1993: 227-243.

[19] Mouss H, Mouss D, Mouss N, et al. Test of page-hinckley, an approach for fault detection in an agro-alimentary production system[C]//Control Conference, 2004. 5th Asian. IEEE, 2004, 2: 815-818.

[20] Ross G J, Adams N M, Tasoulis D K, et al. Exponentially weighted moving average charts for detecting concept drift[J]. Pattern Recognition Letters, 2012, 33(2): 191-198.

[21] Roberts S W. Control chart tests based on geometric moving averages[J]. Technometrics, 2000, 42(1): 97-101.

[22] Ding X, Lian X, Chen L, et al. Continuous monitoring of skylines over uncertain data streams[J]. Information Sciences, 2012, 184(1): 196-214.

[23] Jeffery S R, Garofalakis M, Franklin M J. Adaptive cleaning for RFID data streams[C]//Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006: 163-174.

[24] Quinlan J R. Induction of decision trees[J]. Machine learning, 1986, 1(1): 81-106.

[25] Domingos P, Hulten G. Mining high-speed data streams[C]//Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2000: 71-80.

[26] Bifet A, Read J, Pfahringer B, et al. Cd-moa: Change detection framework for massive online analysis[C]//International Symposium on Intelligent Data Analysis. Springer Berlin Heidelberg, 2013: 92-103.

[27] http://moa.cms.waikato.ac.nz/

[28] Brzeziński D, Stefanowski J. Accuracy updated ensemble for data streams with concept drift[C]//International Conference on Hybrid Artificial Intelligence Systems. Springer Berlin Heidelberg, 2011: 155-163