

A New Classification Algorithm for Data Stream

Li Su

Xi'an University of Technology, Xi'an, China
Email: sunan971@163.com

Hong-yan Liu

Changqing Oilfield Company, Xi'an, China
Email: liuhongxia_xa@163.com

Zhen-Hui Song

ShiJiaZhuang Vocational Technology Institute
ShiJiaZhuang, China
E-mail: songzhenhui_sjz@163.com

Abstract—Associative classification (AC) which is based on association rules has shown great promise over many other classification techniques on static dataset. Meanwhile, a new challenge have been proposed in that the increasing prominence of data streams arising in a wide range of advanced application. This paper describes and evaluates a new associative classification algorithm for data streams AC-DS, which is based on the estimation mechanism of the Lossy Counting (LC) and landmark window model. And AC-DS was applied to mining several datasets obtained from the UCI Machine Learning Repository and the result show that the algorithm is effective and efficient.

Index Terms--data streams; associative classification; frequent itemsets

I. INTRODUCTION

In many fields, such as statistics, artificial intelligence, machine learning, and other disciplines cross discipline, data mining in recent years is becoming a hotspot. Various data mining techniques have been proposed and widely used in order to find useful information from a large number of complex data. With the large number of data produced, the data need to be addressed in one day is to millions or even no limit to the rate of growth. How to mine useful information from these continuous data streams, is becoming the new challenges we had to face [1-9].

At the early stage the data streams originated in the financial markets. Today, the data streams widespread in the Internet, monitoring systems, geology, meteorology, sensor networks and other domain. The data stream is very different with the traditional static data. The data stream is infinite amount of data, data continuous arrived and can only be read for one or a few times. So the faster method of data stream mining need to be updated.

Data-stream mining is a technique which can find valuable information or knowledge from a great deal of primitive data. Unlike mining static databases, mining data streams poses many new challenges [10].

Data stream has different characteristics of data collection to the traditional database model. Such as the date of data stream continuous generation with time progresses, and the data stream is dynamic, and the arrival of the data stream can not be controlled by the order. The data of Data stream can be read and process based on the order of arrival. The order of data can not be changed to improve the results of treatment.

Therefore, the processing of the data stream requires: First, each data element should be examined at most one time, because it is unrealistic to keep the entire stream in the main memory. Second, each data element in data streams should be processed as fast as possible. Third, the memory usage for mining data streams should be bounded even though new data elements are continuously generated. Finally, the results generated by the online algorithms should be instantly available when user requested.

A. Data Stream

Compared with traditional data collection, the data stream is a real-time, continuous, orderly, time-varying, infinite tuple. A data stream has the following distinctive features: a) orderly, b) cannot reproduce, c) high-speed, d) infinite, e) high dimensional, f) dynamic.

Let us now describe the concept of data stream:

Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of literals, called items. An itemset is a subset of I . An itemset consisting of m items is called a m -itemset. Let us assume that the items in an itemset are in lexicographical order. A transaction is a tuple, $(tid, itemset)$, where tid is the ID of the transaction.

A transaction data stream $DS = \{B_1, B_2, \dots, B_N, \dots\}$ be an infinite sequence of blocks, where each block is associated with a block identifier n , and N is the identifier of the latest block B_N . Each block B_i consists of a set of transactions, that is, $B_i = \{T_1, T_2, \dots, T_k\}$, where $k > 0$. Hence, the current length of the data stream is defined as $CL = |B_1| + |B_2| + \dots + |B_N|$.

The frequency of an itemset, X , denoted as $freq(X)$, is the number of transactions in B that support X . The $support$ of X is defined as $freq(X)/N$, where N is the total number of transactions received. X is a Frequent Itemset (FI) in B , if $sup(X) \geq minSupport$, where $minSupport$ ($0 \leq minSupport \leq 1$) is a user defined *minimum support threshold*.

There are many algorithms for mining data stream have been proposed. According to the processing model on the data stream, we can classify the research work into three fields: landmark windows, sliding windows and damped windows. Manku and Motwani[11] proposed a single-pass algorithm, Lossy Counting, to mining frequent itemsets, and the algorithm is based on a well known Apriori-property. Yu et al. [12] proposed an algorithm, FDPM, which is derived from the Chernoff bound, to approximate a set of FIs over a landmark window. Li et al.[13] propose a single-pass algorithm, DSM-FI, to mine all frequent itemsets over the entire history of data streams. Pedro Domingos and Geoff Hulten [14] described and evaluated an algorithm, VFDT, to predict the labels of the records it received.

Classification of data stream mining is a challenging area of research. There are many problems to be solved, such as Handling continuous attributes, Concept drift, Sample taken question, Classification accuracy problem, Data stream management and Pretreatment of the data stream.

1) Handling continuous attributes

When classification of Data Stream face the real-time and memory limit, the reach of how to compute the evaluation function more quickly, how to more effectively compressed storage properties worth further study and how to more effectively compressed storage properties deserves further study.

2) Concept Drifts

Mining concept drifts from data streams is one of the most important fields in data mining. The reach of how more rapidly and accurately judge concept drift, how to effectively use concept drift acquisition, save and heavy use concept, and the trend of concept drift needs to seriously study.

3) Sample sampling

Although there is Hoeffding inequality for sampling methods, how to get better with less precision of the sample, remains a problem worthy to study.

4) Classification accuracy

High classification accuracy is the goal of all classification algorithms. How to improve the classification accuracy is very important research.

5) Data Stream management

Traditional database technology has greatly promoted the development of information technology, but traditional technology look powerless to Data Stream.

6) Pretreatment of the Data Stream

Pretreatment of the Data Stream also need to consider. The reach of how to design a lightweight preprocessing algorithm to guarantee the quality of mining results is very important. The pretreatment of the Data Stream occupies most if the running time, and how to Decrease running time is also important.

7) Re-use of traditional classification methods

Traditional classification is Decision Rules, Bayesian classification, Back-propagation method, Related classifications, K nearest neighbor classifier, Example based reasoning, Evolutionary algorithm, Rough set method, Fuzzy set the legitimate and so on. The current study was applied some of these methods to data stream. How to use the characteristics of the data stream for the application of these methods will be very valuable.

B. Associative Classification

Classification has been studied for many years, it got extensive research of different subjects, including statistics, pattern recognition, machine learning, data mining, etc. It is one of the most essential tasks in data mining and machine learning area. Classification mainly find meaningful information to meet the needs of the user association rules from data. Many effective models and algorithms have been proposed to solve the problem in different aspects, such as support vector machine, decision tree, rule-based classifier, etc.[15].

Different from some traditional rule-based algorithms, associative classification tries to mine the complete set of frequent patterns from the input dataset, given the user-specified minimum support threshold and/or discriminative measurements like minimum confidence threshold. For example, Apriori[16] and FP-growth[17] are now widely used

Association rules proposed by Agrawal etc. The mining of Association rules is a very important research of data mining, which used to find relationship between Itemsets in the database. Simply, Association rules is used to describe the extent of interaction between attributes. With the large amount of data continuously collected and stored, for many people in the industry from their database are increasingly interested in mining association rules. Business transaction records from a large number of interesting relationship was found can help many business decision making. Such as the classification of design, analysis of cross-shopping and the fire sale.

Let $I = \{i_1, i_2, \dots, i_m\}$ be a collection of items.

Task-related data set D is a collection of database transactions, Where each transaction T is a collection of items, and $T \subseteq I$. Each transaction has an identifier, called TID.

Let A is a Itemset, and T contain A , and only if $A \subseteq T$

Association rule is the implication of the form $A \Rightarrow B$, where $A \in I$, $B \in I$, and $A \cap B = \Phi$. For the association rule $A \Rightarrow B$.

$$\text{Support}(A \Rightarrow B) = P(A \cup B)$$

$$\text{Confidence}(A \Rightarrow B) = P(B | A) = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

$$\text{Lift}(A \Rightarrow B) = \frac{\text{support}(A \cup B)}{\text{support}(A) \times \text{support}(B)}$$

Support is percentage of D which contain A or B, is a measure of the importance of association rules, and explain this rule in all things have much representative. If support larger, association rules are more important. If itemsets meet the minimum support (min_sup), then it called frequent itemsets.

Confidence is percentage of D which contain A or B, is a measure of the accuracy of association rules. When predicting, Confidence is a natural choice. It reflects that the premise of A given the incidence of B.

Lift sometimes referred to as Interest, it is the rate between probability of A and B happen at the same time and probability of on the assumption that A and B independent premise A and B also happened. Lift used to measure the association between A and B with A and B are independent degree of deviation. If Lift close to 1, A and B are independent. If Lift is less than 1, this rule is not very meaningful. Lift the greater the practical meaning of the rules the better.

In recent years, Classification of association rules applied to obtain good results. These methods mainly excavate to the training set and get some high quality rules, then build classification according to these rules, and forecasting the new example of label.

Let D be the dataset. Let I be the set of all items in D and C be the set of class labels. We say that a data case $d_i \in D$ contains $X \subseteq I$, a subset of items, if $X \subseteq d_i$. A class association rule (CAR) is an implication of the form $X \rightarrow c$, where $X \subseteq I$, and $c \in C$.

A test instance is classified later using classifier trained based on the mined patterns. Bing Liu et al. [18] first proposed the AC approach, named classification based on association algorithm (CBA), for building a classifier based on the set of discovered class association rules. CBA is one of the most classical associative classification algorithms. Empirical results show that associative classification algorithm could provide better classification accuracy than other algorithms on categorical datasets. However, this approach takes a great amount of running time in both pattern mining and feature selection, since most of the mined frequent patterns are not the most discriminative ones and will be dropped later.

The difference between rule discovery in AC and conventional frequent itemsets mining is that the former task may carry out multiple frequent itemsets mining processes for mining rules of different classes simultaneously. Data mining in associative classification (AC) framework usually consists of two steps:

- (1) Generating all the class association rules (CARs) which has the form of $\text{iset} \Rightarrow c$, where iset is an itemset and c is a class.

- (2) Building a classifier based on the generated CARs. Generally, a subset of the association rules was selected to form a classifier and AC approaches are based on the confidence measure to select rules [19].

Most of the algorithms shown above were used for finding frequent itemsets. Since the foundation of associative classification is frequent pattern mining, we also introduce the definitions and notations related to frequent pattern mining. Some of them were used to classify the data streams with a decision tree.

We present an approach to mine class association rules and then to make a classifier in this paper. Classifying a data stream with an association classifier is a newly explored area of research, which may be viewed as a further extension to the earlier work on mining frequent itemsets over data stream.

II. ASSOCIATIVE CLASSIFICATION ON DATA STREAMS

A. Problem Definitions

A data stream is a large number of unconstrained data elements. Due to the unique characteristics of streaming data, most one-pass algorithms have to sacrifice the correctness of their analysis results by allowing some errors. Hence, the True support of an itemset X, denoted by $Tsup(X)$, is the number of transactions seen so far in which that itemsets occurs as a subset. The estimated support of the itemset X, denoted as $Esup(X)$, is the estimated support of X stored in the summary data structure constructed by the one-scan approaches, where $Esup(X) \leq Tsup(X)$. An itemset X is called a frequent itemset if $Tsup(X) \geq \text{MinSupport} * CL$.

Therefore, given a user-defined minimum support threshold minSupport ($0 \leq \text{minSupport} \leq 1$) and a data stream DS, our goal is to develop a single-pass algorithm to classify the streaming data in the landmark windows model using as little main memory as possible.

```

Input: DS---a data stream in which each record has N items.
      Swindow---the window size, Swindow=|Bi|.
      minSupport--- Support threshold.
Output: M---a classifier with a lot of association rules whose confidence are
        great than 50% and support value great than minSupport.
Method:
  Initial the rule memory M=Φ
  Do
    Read in a data block Bi={T1, T2, ..., Tk}
    m=0; Am=Φ //Clear the set of candidate itemsets A
    Am+1=Gen(Bi, Am) // Generate n candidate frequent itemsets,
                      // Itemset1, Itemset2,..., Itemsetm
                      //each Itemseti in A has 1 items.
  While A ≠ Φ
    For i=1 to n
      S=Supp(Itemseti) //Calculate the support of Itemseti
      If S>= minSupport then
        M←M+ Itemseti //Put the itemset into memroy
      Endif
    Endfor
    m=m+1
    Am+1=Gen(Bi, Am) //Generate the (m+1) generation A
  Endwhile
  M=Rank(M) //Rank rules by their confidence values
  M=Decay(M) //Decay the rules in memory
  While

```

Figure 1. The proposed associative classification algorithm.

B. Associative Classification Algorithm for Data Stream

The figure 1 is our the whole algorithm. The algorithm accepts two user-specified parameters: one is the support threshold minSupport and another is the window size $\text{Swindow} = |B_i|$. Let N denote the current length of the stream, i.e., the number of records seen so far. Every time received a record, our algorithm can forecast its class label based on the association rules extracted from the records before. Each of the rules has a estimated support $\text{Esup}(X)$, whose value is great than minSupport .

For a given data block B_i , the first pass of the algorithm counts item occurrences to determine the frequent 1-itemsets. As long as the set of 1-itemsets was not empty, the algorithm subsequently carry out the next pass to find frequent $(m+1)$ -itemset. When the algorithm obtains all of the frequent itemsets, it will calculate the confidence of the rules and sort them in the memory. Then, if there is a request of classification, the classifier will predict the class label of a record. At the bound of block B_i , the memory rules will be pruned and the rules with low support value will be deleted.

C. Funtions and Data Structure

1) Data struture M

The data structure M is a set of entries of the form (itemset, $f_{\text{class}(1)}, f_{\text{class}(2)}, \dots, f_{\text{class}(i)}, t$), where itemset is a subset of conditional attributes, $f_{\text{class}(i)}$ is an integer representing the approximate frequency of class attributes i , and t is the number of the data blocks in which the itemset appeared firstly. Initially, M is empty. Whenever a new association rule (itemset, $\text{class}(i)$) arrives, we examine M to see whether an entry m_j already exists or not. If exists, we update the entry by incrementing its corresponding frequency $f_{\text{class}(i)}$ by one. Otherwise, we create a new entry of the form (set, $f_{\text{class}(1)}, f_{\text{class}(2)}, \dots, f_{\text{class}(i)}, t$). The parameter t is the number of the data block and $f_{\text{class}(x)}$ is the frequency of class x .

$$f_{\text{class}(x)} = 0 \quad (\text{class}(x) \neq \text{class}(i))$$

$$f_{\text{class}(x)} = 1 \quad (\text{class}(x) = \text{class}(i))$$

(1)

2) Function Gen(B_x, A_m)

The $\text{Gen}(B_x, A_m)$ function takes as argument A_m , the set of all frequent m -itemsets. It returns a superset of the set of all $(m+1)$ -itemsets. The function works as follows [20]:

a) Step 1, we join A_m with A_m :

Insert into A_{m+1} :
 Select $p.\text{item}_1, p.\text{item}_2, \dots, p.\text{item}_m, q.\text{item}_m$
 From A_m, p, A_m, q
 Where $p.\text{item}_1 = q.\text{item}_1, \dots, p.\text{item}_{m-1} = q.\text{item}_{m-1},$
 $p.\text{item}_m < q.\text{item}_m$;

b) Step 2, we delete all itemsets $a \in A_{m+1}$ such that some m -subset of a is not in A_m :

Forall itemsets $a \in A_{m+1}$ do
 Forall m -subsets s of a do
 If NOT ($s \in A_m$) then
 Delete a from A_{m+1} ;

3) Function Supp(Itemset $_x$)

The function $\text{Supp}(\text{Itemset}_x)$ was used to calculate the support value of Itemset_x . The value of the function was calculated like this:

$$\text{Supp}(\text{Itemset}_x) = \text{MAX}(f_{\text{class}(1)}, f_{\text{class}(2)}, \dots, f_{\text{class}(i)})/N$$

(2)

Where N is the number of current data block.

4) Function Rank(M)

The function $\text{Rank}(M)$ was used to sort the rules in M with their confidence values. The confidence value of a entry in M was calculated like this:

$$\text{Confidence}(mi) = \text{MAX}(f_{\text{class}(1)}, f_{\text{class}(2)}, \dots, f_{\text{class}(i)}) / (f_{\text{class}(1)} + f_{\text{class}(2)} + \dots + f_{\text{class}(i)})$$

(3)

5) Function Decay(M)

The function $\text{Decay}(M)$ was used to delete some entries at the boundary of a data block. If the expression below is true, the entry will be deleted from the memory M . N is the number of current data block.

$$\text{MAX}(f_{\text{class}(1)}, f_{\text{class}(2)}, \dots, f_{\text{class}(i)}) + \text{minSupport} * (t-1) * CL < \text{MINSUPPORT} * N * CL$$

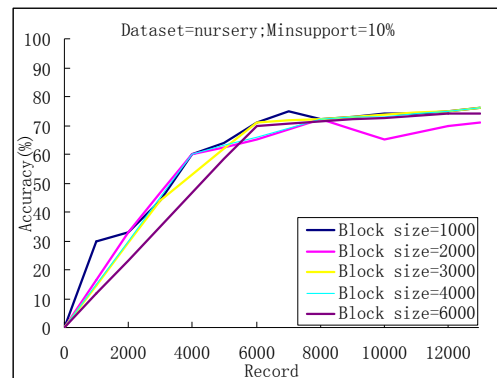
(4)

III. EXPERIMENTS

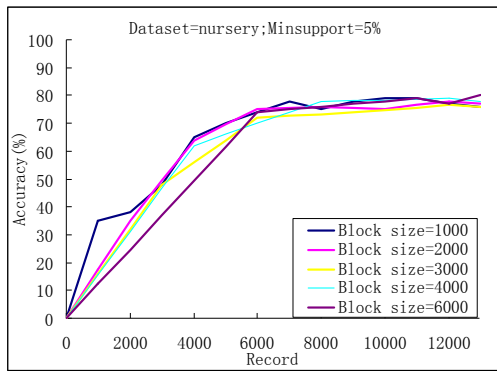
In this section, based on past experience, we provided a formal model of associative classification and further examine how such algorithm can be applied in data stream mining. We have conducted an experiment on a 2.93 GHz Pentium PC with 1GB of memory running with Microsoft Windows XP to measure the performance of the proposed approach. The datasets used in the experiment are obtained from the UCI Maching Learning Repository[21]. In order to compare our algorithm with other classification algorithms and make the evaluation more credible and reliable, we choose some large datasets from UCI. The Entropy method was used in the progress of discretization of continuous attributes[22].

A. Effects of Parameters

In this paper we investigated the effects of the different block sizes on the effectiveness and efficiency of the algorithm and found that the accuracy of the algorithm was not affected obviously by it. But the number of the rules found and the time cost were affected by it largely.



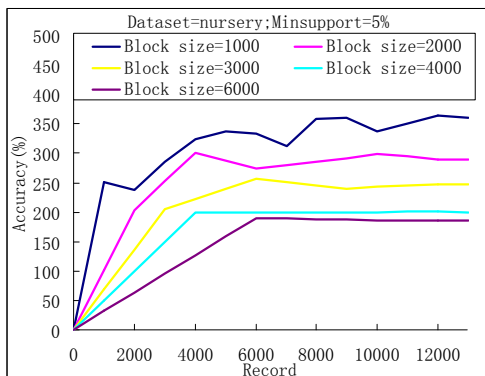
(a) minsupport 10%



(b) minsupport 5%



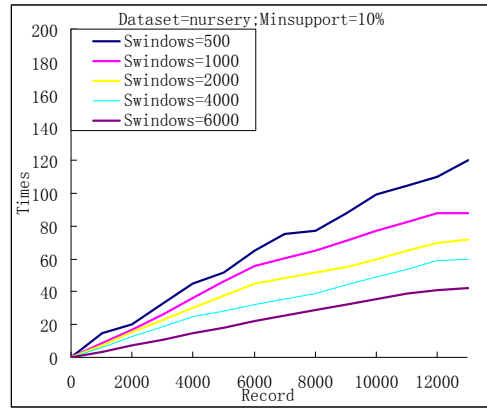
(c) minsupport 10%



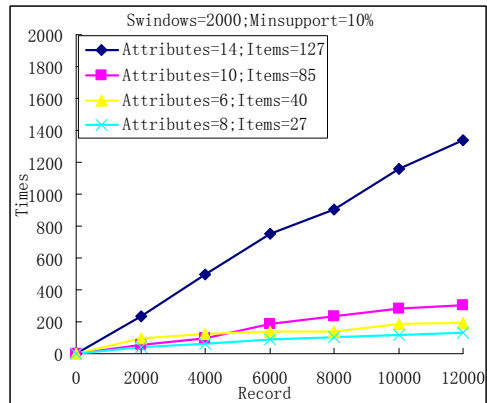
(d) minsupport 5%

Figure 2. The accuracy of the classification algorithm and the number of the rules found.

The figure 2 is the computational results, we can see that when the initial data block size was set to 1000, 2000, 3000, 4000 and 6000, and minimum support threshold was set to 10%, and when the initial data block size was set to 1000, 2000, 3000, 4000 and 6000, and minimum support threshold was set to 5%. So we got the similar forecast results. However, the numbers of the found rules are very different. When the block size getting larger, the number of the found rules became smaller. This is because that, although we delete some entries at the boundary of a data block, there still are some un-frequent rules in the memory and they will be deleted in the later decay step. The larger the data block size is, the less the number of un-frequent rules exists.



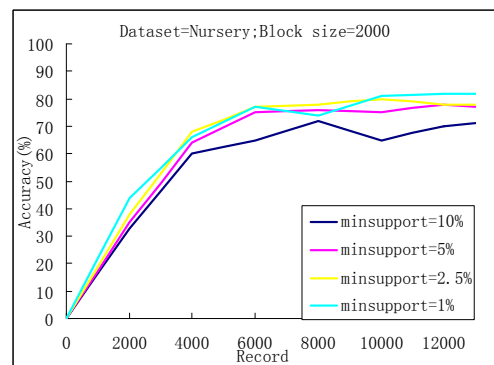
(a) Dataset=nursery ; minsupport 10%



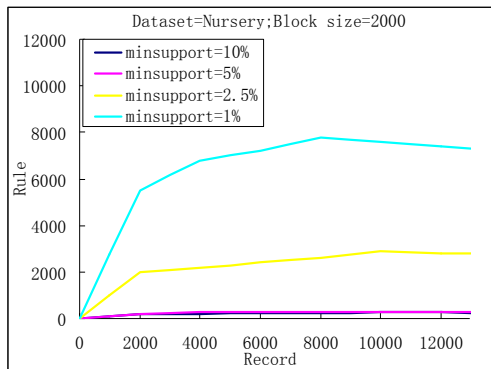
(b) $S_{windows}=2000$; minsupport 10%

Figure 3. The run time of the classification algorithm.

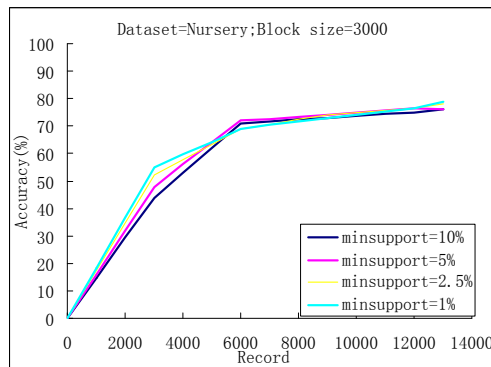
Through our previous research, we further investigated the effects of different block sizes on the time cost by the algorithm and found that with the same dataset, as the increasing of the block size, the total run time becomes more and more long. With the different dataset, the run time was decided mainly by the number of attributes and the number of the items of the dataset. We can see this from Fig.3, the numbers of the attributes of the four datasets, Poke and Adult, are 8, 6, 10 and 14. And the numbers of the items of them are 27, 40, 85 and 127. Although, the support thresholds and the block sizes of them are the same, the run time of them are very different and it increased obviously with the increasing of the item and attribute.



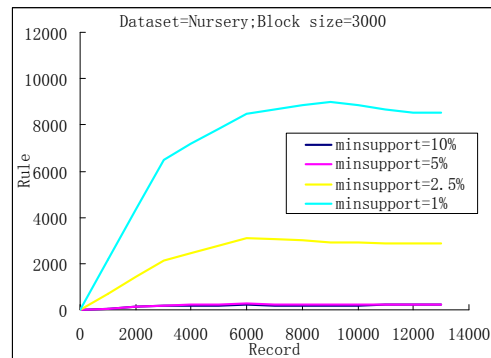
(a) Dataset=nursery; Accuracy Size 2000



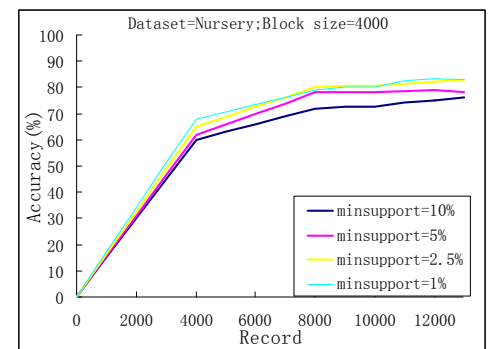
(b) Dataset=nursery; Accuracy Size 2000



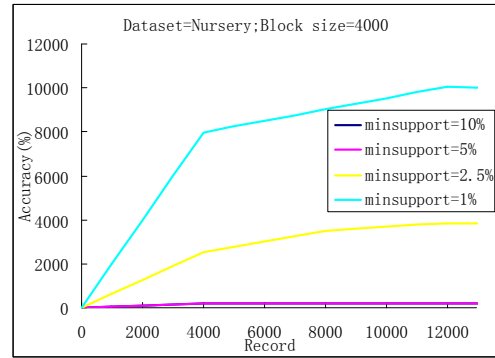
(c) Dataset=nursery; Accuracy Size 3000



(d) Dataset=nursery; Accuracy Size 3000



(e) Dataset=nursery; Accuracy Size 4000



(f) Dataset=nursery; Accuracy Size 4000

Figure 4. The effects of different support thresholds.

When the block size was set to 2000, we tested the performance of the algorithm with the different support thresholds, and when the block size was set to 3000 and 4000, we also tested the performance of the algorithm with the different support thresholds. As it can be seen from Fig.4, the block size of figure (a) and (b) was set to 2000, figure (c) and (d) was set to 3000, and figure (e) and (f) was set to 4000, with the decreasing of the support threshold value, the more accurate classification result we can get. But, at the same time, more classification rules will appear in our classifier, it means more run time.

B. Comparison with different algorithms

Here, we compared the CBA algorithm [8] and AC-DS algorithm. The minimum confidence was both set to 50%, and the data block size was set to 1000. The support thresholds of the two algorithms were shown in the Table.1.

Column 1: It lists the names of the 6 datasets.

Column 2: It lists the number of attributes in the dataset.

Column 3: It lists the number of classes in the dataset

Column 4: It shows the support threshold used in the algorithm.

Column 5: It shows the classification accuracy of the algorithm CBA.

Column 6: It shows the number of the records which had to been read in the memory by CBA.

Column 7: It shows the classification accuracy of the algorithm AC-DS.

Column 8: It shows the number of records in a data block which was read in the memory by AC-DS.

We test the adult, digit, Letter, Mushroom, Nursery, Krkopt, 6 data sets which come from UCI Machine Learning Repository. As can be seen from table.1, the mean accuracy of the two algorithms is very similar, but the memory used by CBA is obviously greater than AC-DS. Since CBA is a classification algorithm for static dataset and AC-DS is a classification algorithm for mining data streams, the more total time cost by AC-DS than CBA should be think acceptable as long as the two

accuracy rates of them are similar and the memory cost by AC-DS is in a special limit.

TABLE I. COMPARISON WITH DIFFERENT ALGORITHM

Dataset	#attr	#class	#supp	CBA		AC-DS	
				#accu	#mem	#accu	#mem
Adult	14	2	10%	83.7%	48842	82.1%	1000
Adult	14	2	5%	83.9%	48842	81.9%	1000
Adult	14	2	1%	85.3%	48842	83.7%	1000
Digit	16	10	10%	78.7%	10992	75.6%	1000
Digit	16	10	5%	86.0%	10992	83.4%	1000
Digit	16	10	1%	87.2%	10992	86.0%	1000
Letter	16	26	10%	59.2%	20000	62.8%	1000
Letter	16	26	5%	69.0%	20000	67.4%	1000
Letter	16	26	1%	70.1%	20000	68.6%	1000
Mushroom	22	2	20%	94.0%	8124	91.4%	1000
Mushroom	22	2	10%	96.4%	8124	93.2%	1000
Mushroom	22	2	2%	96.6%	8124	94.6%	1000
Nursery	8	5	10%	89.9%	12960	90.4%	1000
Nursery	8	5	5%	91.5%	12960	89.5%	1000
Nursery	8	5	1%	96.1%	12960	93.8%	1000
Krkopt	6	18	10%	89.2%	28056	87.4%	1000
Krkopt	6	18	5%	94.2%	28056	91.8%	1000
Krkopt	6	18	1%	97.9%	28056	97.1%	1000

IV. DISCUSSIONS AND CONCLUSION

With the development of information technology, more and more applications produce or receive a steady stream of data streams. Data streams analysis and mining have become a research hotspot.

As each transaction was not processed, we used to read in the available main memory as many transactions as possible .So, we always select a large data block size to process. When the support threshold is fixed, the more the data block size is, the less combinatorial explosion of itemsets takes place.

This paper introduced a kind of mining association of data stream classification. Our algorithm was designed to deal with dataset in which the all data was generated by a single concept. If the concept function is not a stationary one, in other words, a concept drift takes place in it, our algorithm will not output an accurate result. Empirical studies show its effectiveness in taking advantage of massive numbers of examples. AC-DS’s application to a high-speed stream is under way.

REFERENCES

[1] B Babcock, S Babu, M Datar, et al-Models and issues in datastreams systems [C]-The 21st ACM SIGACT-SIGMOD-SIGART Symp on Principles of Database Systems, Madison,2002

[2] P Domingos, G Hulten-Mining high-speed data streams [C]-The Association for Computing Machinery 6th Int’l Conf on Knowledge Discovery and Data Minings, Boston, 2000

[3] R Jin, G Agrawal-Efficient decision tree construction on streaming data [C]-The ACM SIGKDD 9th Int’l Conf on Knowledge Discovery and Data Mining, Washington, 2003

[4] S Muthukrishnan-Data streams: Algorithms and applications[C]-The 14th Annual ACM-SIAM Symp on Discrete Algorithms, Baltimore, MD, USA, 2003

[5] [H Wang, W Fan, P Yu, et al-Mining concept-drifting datastreams using ensemble classifiers [C]-The 9th ACM Int’l Conf on Knowledge Discovery and Data Mining (SIGKDD), Washington, 2003

[6] Q H Xie-An efficient approach for mining concept-drifting datastreams: [Master dissertation][D]-Tainan, China: National University of Tainan, 2004

[7] M Guetova, Holldobter, H P Storr-Incremental fuzzy decision trees [C]-The 25th German Conf on Artificial Intelligence(KI2002), Aachen, Germany, 2002

[8] Yang Yidong, Sun Zhihui, Zhang Jing-Finding outliers in distributed data streams based on kernel density estimation [J]-Journal of Computer Research and Development, 2005, 42(9):1498-1504 (in Chinese)

[9] Qian Jiangbo, Xu Hongbing, Dong Yisheng, et al-A windowjoin optimization algorithm based on minimum spanning tree[J]-Journal of Computer Research and Development, 2007, 44(6): 1000-1007 (in Chinese)

[10] R. Agrawal, T. Imielinski and A. Swami. “Mining association rules between sets of items in large databases”. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C, May 1993.

[11] Gurmeet Singh Manku and Rajeev Movtwni, “Approximate Frequency Counts over Data Streams”. Proceedings of the 28th VLDB conference, Hong Kong, China, 2002.

[12] Yu J, Chong Z, Lu H et al. “False positive or false negative: mining frequent itemsets from high speed transactional data streams. In: Nascimento et al.(eds) Proceedings of the thirtieth international conference on very large data bases”, Toronto, Canada, September 3-August 31, 2004, pp 204-215.

[13] Li H, Lee S, Shan M. “An efficient algorithm for mining frequent itemsets over the entire history of data streams”. Proceedings of the first international workshop on knowledge discovery in data streams, Pisa, Italy, 2004.

[14] Pedro Domingos and Geoff Hulten. “Mining high-speed data streams”, In Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, page 71-80, Boston, MA, 2000. ACM Press.

[15] [1] Chuancong Gao, Jianyong Wang. Direct Mining of Discriminative Patterns for Classifying Uncertain Data

[16] [1]Agrawal R, Imilinski T, Swami A. Mining Association Rules Between Sets of Items in Large Database[C] –Proceedings of the ACM SIGMOD Conference on Management of Data. Washington DC, 1993: 207-216

[17] [2]Han J, Pei J , Yin Y. Mining frequent pat terns with out candidate generation[C] –Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, TX, 2000:1-12

[18] B. Liu, W. Hsu, and Y. Ma. “Integrating classification and association rule mining”. In KDD 98, New York, NY, Aug. 1998.

[19] B. Liu, Y. Ma, and C.-K. Wong, “Improving an association rule based classifier,” in Proc.4th Eur. Conf. Principles Practice Knowledge Discovery Databases(PKDD-2000),2000.

[20] R. Agrawal and R.Srikant, “Fast algorithms for mining association rules”. In Proc. 20th Int. Conf. Very Large Data Bases(VLDB), 1994, pp.1-12

[21] D. J. Newman, S. Hettich, C. Blake, and C. Merz, “UCI Repository of Machine Learning Databases”. Berleley, CA: Dept. Information Comput. Sci., University of California, 1998.

- [22] R. Kohavi, D. Sommerfield, and J. Dougherty, "MLC++: A machine learning library in C++," in Proc.6th Int. Conf. Tools Artificial Intelligence, New Orleans, LA, 1994, pp.740–743.

Li Su was born in Tai yuan city, Shanxi province, China, in 1982. He received the B.Sc. degrees from Taiyuan University of technology, in 2005.

Hongyan Liu was born in Zi zhou city, Shanxi province, China, in 1974. She received the B.Sc. degrees from Lanzhou University of technology, in 1997.

He is currently an engineer with the Company of Changqing Oilfield, China. His research interest includes evolutionary management and data mining, et al.

Zhen-hui Song was born in Handan city, Hebei province, China, in 1965. He received the B.Sc. degrees from Hebei University of technology, in 1987. He is currently a professor with the ShiJiaZhuang Vocational Technology Institute, ShiJiaZhuang, China. His research interest includes evolutionary computation and data mining, et al.