Mining Association Rules between Values across Attributes in Data Streams

Shankar B. Naik
Dept. of Computer Science
S.S.A. Govt. College
Pernem, Goa, India
+919420897135
xekhar@rediffmail.com

Jyoti D. Pawar
Dept. of Computer Science and Technology
Goa University
Goa, India
+919422059112
jyotidpawar@gmail.com

***Abstract*- In this paper we propose a framework and approach to model events as elements of data stream and perform analysis to group values of attribute similar to each other within an attribute and find associations between clusters of values across two attributes. Experiments have been performed on both synthetic and real data sets**

Keywords: Data streams, Data mining, Sliding window, Clustering

I. INTRODUCTION

In recent times, data mining over data streams have gained a significant attention of researchers. A significant amount of work has been done in mining patterns from data streams. Analysis such as market basket analysis to mine frequent itemsets from transactional data stream, prominent patterns within a data stream, and clustering of streams have been the main focus of mining from data streams. In this paper, we perform analysis to find clusters of values of an attribute in a data stream and association rules between two clusters belong to two different attributes. An element of the data stream considered is a tuple of values pertaining to the attributes of the data stream (Fig. 1.).

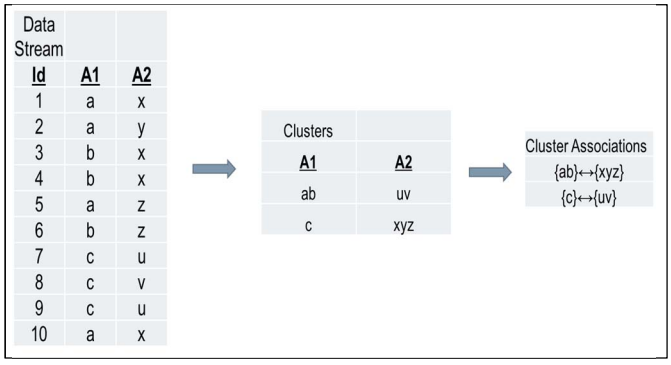

Fig 1. Data stream, clusters and cluster association

For example, users create their profiles on social media websites. These profiles contain a set of attributes pertaining to the personal, educational, or professional data. There can be a relation between the values of two different attributes in a user profile. This relation can be exploited to mine valuable patterns of information. For example, studying the relation between values of two different attributes 'educational institute' and 'workplace' of a user can help educational organizations to find the prospective companies offering jobs related to their area of studies, and the companies to identify the educational organizations producing students who can be employed in them.

The number of users on social websites is large and so are the number of updates done to their profiles. If a change by a user is called as an event, the frequency of these events is very high due to the large number of users online at a moment. In this scenario, these events can be modelled as a stream of attribute value sets (pairs) in which the element of a data stream is a set (pair) of two values, each belonging to a different attribute.

In this paper we propose a framework and approach to model these events as a data stream and perform analysis to group attribute values similar to each other within an attribute and find associations between clusters of values across two attributes.

The approach uses frameworks of market basket analysis as the focus is on finding associations between attribute values, and clustering analysis as these association patterns are then used to group these values of attributes. Methods like text mining, semantic analysis when induced in this approach make the approach more intelligent.

The remainder of the paper is as follows. Section 2 contains the related work. The problem is defined in Section 3. The algorithm is proposed in Section 4. Section 5 contains the experiments done, Section 6 concludes the paper.

II. RELATED WORK

A. Data stream

Data steam is a large collection of elements in which the arrival rate of elements is very high [3]. Data mining on a data stream is a challenging task due to the unknown size of the data stream, no possibility of storing all the elements of the entire data stream at once for analysis, and the approximations and errors in the results generated. There are three approaches that are used to mine patterns from a data stream: landmark windows, damped windows, and sliding windows. In the landmark window model, only the transactions between a timestamp called landmark and the latest transaction are considered for analysis. In damped

978-1-5090-5595-1/17/$31.00 ©2017 IEEE

window model, the recent transactions are considered more significant than the previous ones. In sliding window model the latest n transactions are considered for analysis, where n is the size of the sliding window. In this paper, we have focused on the sliding window model.

B. Social network analysis

Cheng et. al .[2] have presented a similar problem but not from data stream point of view. They have analyzed the information about workplace of users from the social network point of view. They collected the job-related information from various social media sources. Thereafter, the collected data were used to construct an inter-company job-hopping network. The vertices denote companies and the edges denote the movement of people between companies. They used graph mining techniques to generate clusters of related companies. Xu et. al.[4] have presented a similar problem from a social network model point of view. Both these papers have specifically focused on the work company attribute of the user and aim at finding relations between companies from employment point of view. The main aim is to cluster values of a single attribute i.e. workplace. The analysis is done offline. In this paper we provide an approach not pertaining specifically to the attribute work company alone. It is applicable to other attributes. The approach generates clusters of values for each attribute and then finds association between clusters belonging to different attributes. The approach enables the user to analyze data online.

B. Association Rules

The concept of association rule mining originates from market basket data analysis where rules like "A customer buying products p1 and p2 will also buy product p3 with probability p%" are found. They are applicable to a wide range of business problem and are not restricted to dependency analysis in retail applications.

An association rule is an expression of the form A→B, where A and B are sets of items. The meaning of association rules is quite intuitive. Given a transactional database D, where each T∈D is a set of items, A→B expresses that whenever a transaction T contains A then T probably contains B. The probability or confidence of rule is defined as the percentage of transactions containing elements of both A and B out of the number of transactions containing elements of A. The confidence of a rule can be considered as the conditional probability $p(B \subset T | A \subset T)$.

In this paper, we modify the concept of association rules as described in the next section. The associations here are between clusters of values of attributes, where both the clusters belong to different attributes.

III. PROBLEM DEFINITION

A. Preliminaries

Let A1 and A2 be the attributes of study. For example A1 is 'school' and A2 is 'workplace'. Let V1= {v11, v12, …, v1n} and V2= {v21, v22, …, v2n} be the sets of values for an attribute A1 and A2, respectively. For example V1={GMC, KLE,…} is a set of educational institutes and V2={Govt. Hospital, Vision, …} is a set of organizations employing people.

The pair T=(v11,v21) is an element of the data stream, where v11∈V1 for attribute A1 and v21∈V2 for attribute A2 for the same record. For example the element T=('GMC','Vision') means the person has studied in 'GMC' school and works for 'Vision' organization.

Two values v11∈T1 and v12∈T2 are similar if there exists Ti=(v11,v2i) and Tj=(v12,v2j) where v2i=v2j and i≠j. If v11 is similar to v12 and v12 is similar to v13 then v11, v12 and v13 are similar. For example, if ('GMC','Vision') and ('KLE','Vision') are elements of the data stream, then 'GMC' and 'KLE' are similar. The similarity between two values v11 and v12, denoted as sim(v11,v12), is the total count of elements containing both v11 and v12.

C={vji/ all vjis are similar} is a cluster of similar values of attribute Aj. For example, {'GMC','KLE'} is a cluster as 'GMC' and 'KLE' are similar.

A data stream D={T1,T2,…,Tn} is a stream of events. Two elements T1 and T2 are same if they represent same sets of values. Two values v11∈T1 and v12∈T2 are heavily similar if the number of Ti=(v11,v2i) and Tj=(v12,v2j) where v2i=v2j and i≠j., are is not less than a minimum threshold value s0 i.e. sim(v11,v12)>=s0. The value of s0 is decided by the user. The significance of s0 is that it allows to ignore the clustering together values that hardly have any transitions happening between them, which is quite often a situation in the real world. For example, a person changing joins a company not related to the type of study done at school. Such elements should be ignored so as to nullify their effect on the results. Such transitions seldom happen, which are less in number. A minimum support threshold will restrain such values from clustering together.

If sim(v11,v12)>= s0 and sim(v12,v13)>=s0 then v11 and v13 are heavily similar.

Let C1i and C2j be clusters of the attributes A1 and A2, respectively. C1i and C2j are associated, represented as C1i↔C2j, if there exist v1a∈Cli and v2b∈C2j and v1a and v2b are heavily similar. In the above example {'GMC','KLE'}↔{'Govt. Hospital','Vision'} is an association since ('GMC',Vision') belongs to the data stream.

B. Problem Statement

Given two attributes A1 and A2, a set of values V1={v11,v12,…} and V2={v21,v22,…},a data stream D={T1,T2,…} consisting of elements element T=(v11,v21), where v11∈V1 for attribute A1 and v21∈V2 for attribute A2, generate a set of clusters of values for each attribute and the associations between clusters of values between attributes A1 and A2.

IV. THE PROPOSED FRAMEWORK

A. Intermediate Summary Structure

Intermediate summary data structure stores summary information about the elements of the data stream as they are processed. The data stored in the intermediate summary data structure is used to generate the results. The intermediate

summary data structure proposed in this paper has two parts, a matrix MT and a two lists of clusters CL1 and CL2 for attributes A1 and A2, respectively.

I) Adjacency Matrix MT: MT is the adjacency matrix whose rows and columns represent the values of V1 and V2 respectively. The value of MT(i,j) represents the number of elements in a data stream containing values v1i and v2j (Fig. 3.1).

II) Lists of Clusters CL1 and CL2: CL1 and CL2 are sets of clusters, of attributes A1 and A2, respectively, that are generated at various steps of the algorithm execution (Fig. 2.).

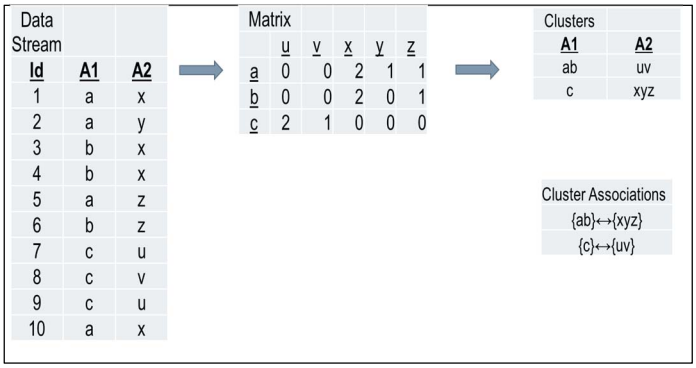

Fig. 2. Intermediate Summary Data Structure with Matrix MT and Cluster List

B. The Proposed Algorithm

This algorithm works in three steps, Update, Generate Cluster and Generate Association.

1) The Update step: This step updates the matrix MT when a new transitional element either enters (Add step) or leaves (Remove step) the sliding window SW.

a) The Add step: When a new element T={v1i,v2j} arrives at the sliding window, MT(i,j) is increased by one each.

b) *The Remove step:* When an element T={v1i,v2j} leaves the sliding window, MT(i,j) is decreased by one each.

2) The Generate Cluster step: This step generates clusters from the matrix MT.

Input: MT,s0
Output: CL1={C11,C12,…} and CL2={C21,C22,…}

1. For every row v1i of matrix MT, v1i∈V1, and v1i not already assigned to any cluster
   a. C2i={v2s/ MT(v1i,v2s)>=so}
2. Merge C2i and C2j for i,j=1,2,… if C2i∩C2j≠ϕ
3. Add the clusters to CL2
4. For every column v2i of matrix MT, v2i∈V2, and v2i not already assigned to any cluster
   a. C1i={v1s/ MT(v2i,v1s)>=so}
5. Merge C1i and C1j for i,j=1,2,… if C1i∩C1j≠ϕ
6. Add the clusters to CL1

Fig. 3. Pseudocode for Generate Step

A running example of the above algorithm is demonstrated in Fig. 4. Let the values of s0 be 1. For the first row 'a' C20={x,y,x} since MT[a,x], MT[a,y], and MT[a,x] are greater than or equal to s0=1. Similarly, for the second row 'b', C2={x,z} and for the third row C3={u,v}. Since C0∩C1={x,z}≠ϕ, C0 and C1 are merged. Hence, CL1={{x,y,z},{u,v}}.

The above algorithm can be executed by the user at any time by specifying the minimum threshold value s0. But, it does not incrementally update the clusters in CL1 and CL2. We propose algorithms IncAdd and IncDelete to incrementally maintain the clusters in CL1 and CL2. These algorithms can be executed only if the Generate algorithm in Fig. 3 at least once.

a) IncAdd: This algorithm is executed after an element has arrived in the sliding window SW (Fig. 5).

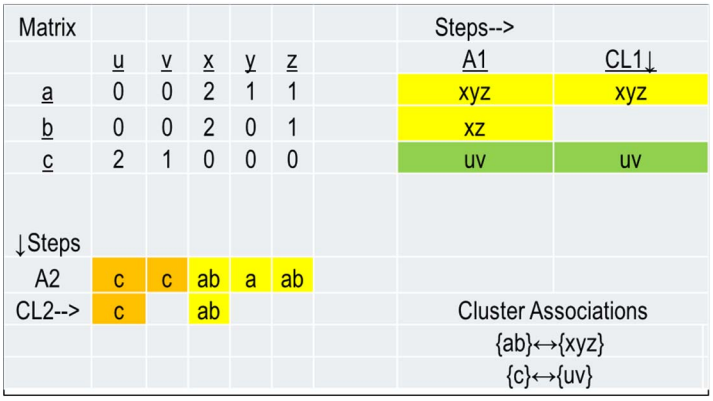

Fig. 4. Generate Cluster Example

Input: Matrix MT, CL1,CL2, s0
Output: Updated CL1, CL2

1. When element T=(v1i,v2j) enters SW
   a. MT[v1i,v2j]++;
   b. if MT[v1i,v2j]=s0;
      i. C1new=C1aUC1b, C1a,C1b∈CL1, v1i∈C1a, and sim(v1i,v1n)>=s0 for some v1n∈Clb
      ii. CL1=CL1UC1new-{C1a,C1b}
      iii. C2new=C2aUC2b, C2a,C2b∈CL2, v2j∈C2a, and sim(v2j,v2n)>=s0 for some v2n∈C2b
      iv. CL2=CL2UC2new-{C2a,C2b}

Fig. 5. Algorithm IncAdd

The working of the above algorithm with an example is demonstrated in Fig. 6. Let the value of s0 be 1. The element entering the sliding window is T=(x,c). Value of MT[c,x] is increased by 1. The clusters in CL1 are {x,y,z} and {u,v}. Since x ∈{x,y,z}, sim(x,u)>s0 and u∈{u,v}, the clusters {x,y,z} and {u,v}

are merged i.e. C1new={x,y,z}U{u,v}. Hence, CL1={{u,v,x,y,z}}. Similarly CL2={{a,b,c}}.

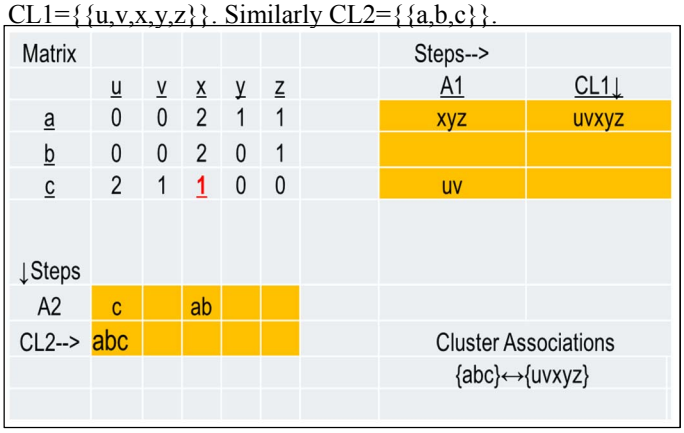

Fig 6. IncAdd Step Example

b) IncDelete: This algorithm is executed after an element has left in the sliding window SW (Fig.7).

Input: Matrix MT, CL1,CL2, s0
Output: Updated CL1, CL2
1. When element T=(v1i,v2j) leaves SW
   a. MT[v1i,v2j]--;
   b. if MT[v1i,v2j]=s0-1;
      i. Split C1∈CL1 into C1a and C1b such that sim(v1i,v1a)>=s0 and sim(v1i,v1b)<s0 for all v1a∈C1a, v1b∈C1b and v1i∈C1
      ii. CL1=CL1UC1aUC1b-C1
      iii. Split C2∈CL2 into C2a and C2b such that sim(v2j,v2a)>=s0 and sim(v2j,v2b)<s0 for all v2a∈C2a, v2b∈C2b and v2j∈C2
      iv. CL2=CL2UC2aUC2b-C2

Fig.7 Algorithm IncDelete

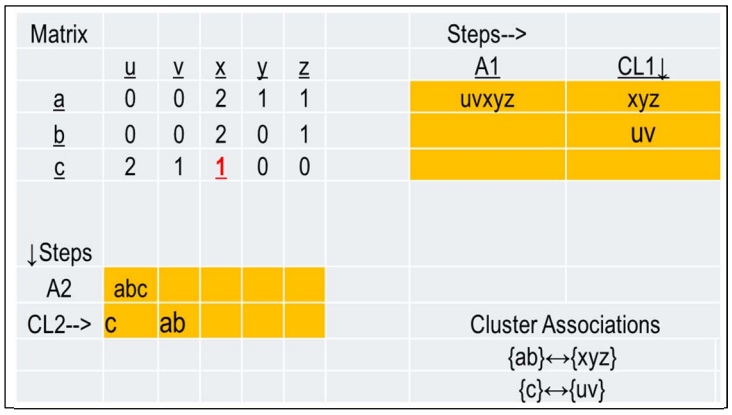

Fig 8 IncDelete Step Example

The working of the above algorithm with an example is demonstrated in Fig. 8. Let the value of s0 be 1. The element leaving the sliding window is T=(x,c). Value of MT[c,x] is decreased by 1. The clusters in CL1 is {u,v,x,y,z}. Since x ∈{u,v,x,y,z} the cluster {u,v,x,y,z} is split into the clusters {x,y,z}(sim(x,u) and sim(x,v)<s0) and {u,v} (as sim(x,,y) and sim(x,z)>=s0) Hence, CL1={{u,v},{x,y,z}}. Similarly CL2= {{a,b},{c}}.

3) Generate Association Step: This step generates associations between clusters of different attributes. Two clusters C1∈CL1 and C2∈CL2 are associated if sim(v1,v2)>=s0, v1∈C1 and v2∈C2. The association between two clusters C1 and C2 is denoted as C1↔C2 (Fig 9).

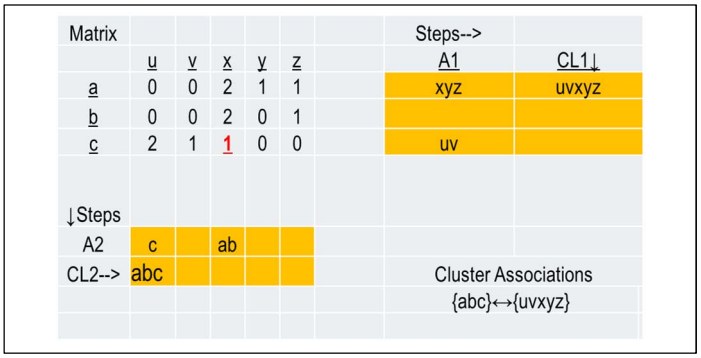

Fig 9 Generate Association Step Example

V. EXPERIMENTS

Experiments were performed to check the efficiency of the proposed algorithm on two data sets, synthetic and a real data set. All experiments were performed on a system with 2.26GHz Intel® Core™ i3 processor, 3 GB memory and Windows 7 operating system. The algorithms were implemented in C++ language and was compiled with GNU GCC compiler.

The synthetic data were generated using IBM Synthetic Data Generator [1]. The synthetic dataset parameters are mentioned in Table I.

TABLE 1.

Parameter	Value
Number of transactions	300K
Average items per transaction	20
Number of items	200

The real data set [8] has the following characteristics. The data set contains traffic accident data. This data set is obtained from the National Institute of Statistics (NIS) for the region of Flanders (Belgium) for each traffic accident that occurs with injured or deadly wounded casualties on a public road in Belgium. In total, 340,184 traffic accident records are included in the data set.

The traffic accident data contain information on the different circumstances in which the accidents have occurred: course of the accident, traffic conditions, environmental conditions, road conditions, human conditions and geographical conditions. On average, 45 attributes are filled out for each accident in the data set.

These experiment were performed on the above data sets using the sliding window model approach to determine the number of clusters by changing the value of minimum threshold. The size of the sliding window was kept to 10K in both the cases

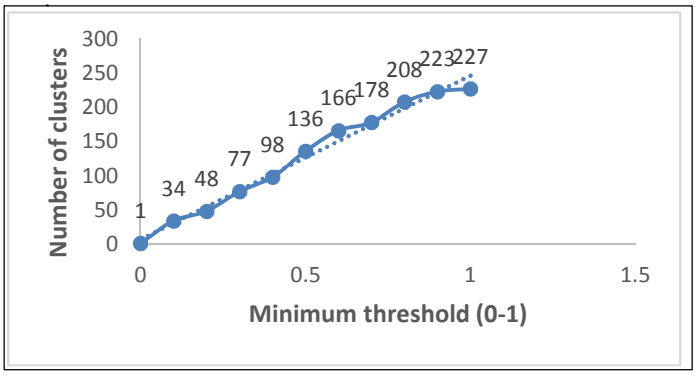

Fig.10 Results on Synthetic Data Set

The experiment in Fig. 10. was performed on synthetic data set by varying the value of minimum support threshold. The clusters shown contain values of only one attribute. The number of clusters increases as the value of minimum support threshold increase. For lower values of minimum support threshold, the number of clusters is les because values that even have low similarity between them, but higher than the minimum support value, are grouped together. As the value of minimum support threshold increases the more and more values become dissimilar as the similiraty between the values begins to falll below the minimum support threshold.

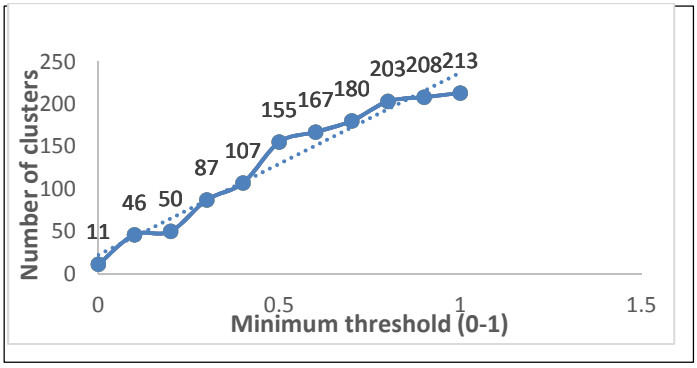

Fig. 11 Results on Real Data Set

Similarly, the experiment in Fig. 11. was performed on the real data set by varying the value of minimum support threshold. The clusters shown contain values of only one attribute i.e. ‘road condition’. Likewise in the case of synthetic data set, a similar observation was made here.

The main objective of performing experiments on real data set was to check the accuracy, precision and recall of the algorithm. This was done by varying the value of minimum support. The attributes considered were ‘Road Condition’ and ‘Place’. For every value of minimum support the clusters were generated for values of each attribute. Thereafter, associations between these two clusters each of different attribute were generated. The generated clusters and associations between them were compared with results obtained by applying apriori algorithm to the real data set to find the accuracy, precision and recall separately for the attributes and the associations. The results are as below.

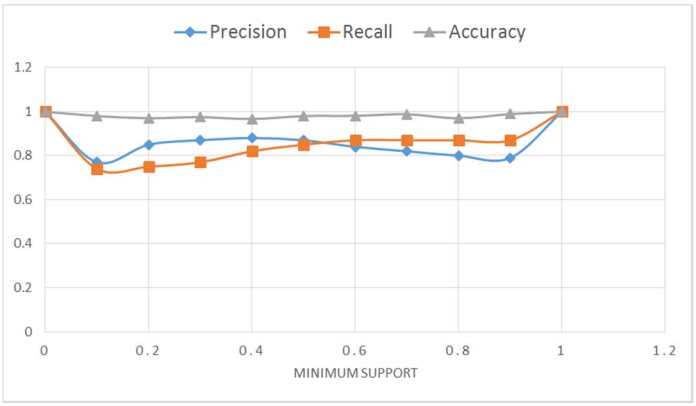

Fig. 11 Precision, Recall and Accuracy of association of cluster on Real Data Set

The experiment (fig. 12) was performed by varying the value of minimum support from 0 to 1. The precision value for the associations varied with upper bound as 0.88 and lower bound as 0.77. The value recall value varied between 0.74 and 0.87. The recall value increased consistently with the value of minimum support and stayed stable after 0.6 in 87%, while the accuracy was always above 96%.

IV. CONCLUSION AND FUTURE WORK

We proposed algorithms and an approach for analysis data streams whose every element is as pair of values of two different attributes. These values within an attribute were clustered to generate groups of attribute values related to each other and then find association between clusters of values across different attributes. These values can be objects like company names, educational institutes, places, etc.

We have limited our study by considering only the attribute values. Data about the values themselves can be incorporated into the algorithms to generate better patterns.

The study is limited to a data stream pertaining to two attributes only. It can be applied to data streams with multiple attributes.

As there is no similar work done in this area a comparative study was not possible and experiments were performed to check the precision, recall and accuracy of the algorithm.

Subsequently, our future work shall address these issues.

REFERENCE

[1] R. Agrawal and R. Shrikant. Fast algorithms for mining association rules.Proceedings of the 20th international conference on very large databases, 487-499, 1994.

[2] Y. Cheng, Y. Xie, Z.Chen, A. Agrawal. Jobminer: A realtime system for mining job-related patterns from social media, 2013

[4] H. Xu, J. Yang, H.Xiong, H.Zhu. Talent Circle Detection in job transition networks. KDD, 2016

[5] S. B. Naik, J.D. Pawar. An efficient incremental algorithm to mine closed frequent itemsets over data streams. Proceedings of the 19th COMAD 2013

[6] Naik, S.B. and Pawar, J.D. 2015 A quick algorithm for incremental mining closed frequent itemsets over data streams. Proceeding of the 2nd IKDD CODS, (2015)

[7] Naik, S. B., and Pawar J.D. . "Finding frequent item sets from data streams with supports estimated using trends." Journal of Information and Operations Management 3.1 (2012): 153.

[8] Geurts, K., Wets, G., T. Brijs and K. Vanhoof (2003). Profiling High Frequency Accident Locations Using Association Rules. Electronic Proceedings of the 82th Annual Meeting of the Transportation Research Board, Washington, January 12-16, USA, 18pp.

[9] Y. Cheng, Y. Xie, K. Zhang, A. Agrawal, and A. Choudhary. Cluchunk: clustering large scale user-generated content incorporating chunklet information. In Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, ProgrammingModels and Applications, BigMine '12, pages 12–19, 2012.

[10] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10, pages 199–208, New York, NY, USA, 2010. ACM.

[11] M. E. Newman. Modularity and community structure in networks. Proc Natl Acad Sci U S A, 103(23):8577–8582, June 2006.

[12] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. Physical Review E, 76(3):036106+, Sept. 2007.

[13] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. Detecting communities and their evolutions in dynamic social networks–a bayesian approach. Mach. Learn., 82(2):157–189, Feb. 2011.