American Journal of Software Engineering, 2014, Vol. 2, No. 2, 16-21
Available online at http://pubs.sciepub.com/ajse/2/2/1

DOI:10.12691/ajse-2-2-1

Development of Sequential ID3: "An advance Sequential mining Algorithm"

Swati Singh Lodhi*

Sanghvi Innovative Academy, Indore (MP) India
*Corresponding author: swati.singh0710@gmail.com

Received July 10, 2014; Revised July 17, 2014; Accepted July 21, 2014

Abstract Sequential pattern mining is an important data mining technique which discovers closed frequent sub sequence from a sequence database. Sequential pattern mining was used in a great spectrum of areas. Some of the applications of sequential pattern mining are namely bio-informatics, web access traces, system utilization logs etc. The data is naturally in the form of sequences. However it is very difficult as it generates explosive number of sub sequence in candidate generator and test approach. Previous sequential pattern mining algorithm like Clospan, Sequence generator, closed sequence-sequence generator mining (CSGM). In sequential pattern mining and web log mining a traditional algorithm Apriori is always reminded but due to some performance issues they were replaced with other algorithms and techniques. Many different techniques for mining frequent sequential patterns from the log data have been proposed in the recent past but still mining data from weblog files an effective and efficient algorithm is required that works with high performance. Moreover; it is required to authenticate the algorithm for that purposes we have used a traditional algorithm for mining sequential pattern from web log data. Thus the aim of the present work is to bridge these gaps by developing and proposing a new algorithm "Sequential ID3" for sequential pattern mining and their experimental validation on web log data.

Keywords: *Sequential ID3, web log data, CSSGM, algorithm*

Cite This Article: Swati Singh Lodhi, "Development of Sequential ID3: "An advance Sequential mining Algorithm"." *American Journal of Software Engineering*, vol. 2, no. 2 (2014): 16-21. doi: 10.12691/ajse-2-2-1.

1. Introduction

Sequential pattern is a sequence of item sets that frequently occurred in a specific order, all items in the same item sets are supposed to have the same transaction time value or within a time gap. Sequence data can be found at every place. For example, if a customer buys a car, he/she will eventually buy car insurance. This is potentially useful in designing personalized marketing strategy. Sequential pattern mining is used in a great spectrum of areas. In computational biology, sequential pattern mining is used to analyze the mutation patterns of different amino acids. Business organizations use sequential pattern mining to study customer behaviors. Sequential pattern mining is also used in system performance analysis and telecommunication network analysis. Sequential pattern mining involves discovering frequent sequences from a database where data to be mined is in some sequential order. The goal of sequential pattern mining is to discover all frequent sequences of item sets in a dataset. Sequential pattern mining identifies sequential patterns appearing with enough support. It has potential application in many areas such as analysis of market data, purchase histories, web logs, etc. Sequential rules express temporal relationships among patterns. It can be considered as a natural extension to many spurious patterns by introducing the notion of confidence to the set of patterns. Only rules satisfying both support and confidence thresholds are mined. Sequential rules extend the usability of patterns beyond the understanding of sequential data. Usually all the transactions of a customer are together viewed as a sequence, usually called customer-sequence, where each transaction is represented as an item sets in that sequence, all the transactions are list in a certain order with regard to the transaction-time. Contain, a sequence $<a_1, a_2,\ldots..,a_n>$ is contained in another sequence $<b_1, b_2,\ldots.,b_m>$, if $i_1< i_2< \ldots.. < i_n$ such that $a_1\subseteq b_{i1}, a_2\subseteq b_{i2},\ldots.., a_n\subseteq b_{in}$. For example, the sequence <(3)(6, 7, 9)(7, 9)> is contained in <(2)(3)(6,7, 8, 9)(7)(7, 9)>, since $(3)\subseteq(3),(6,7,9)\subseteq(6,7,8,9)$, $(7,9)\subseteq(7,9)$. However, sequence < (2) (3)> is not contained in sequence <(2,3)> since the former sequence means 3 is bought after 2 being bought, while the latter represents item 2 and 3 being bought together. A sequence is maximal if it is not contained in any other sequences.

Sequential pattern mining was first introduced by Agarwal et al. [1]. It is the process of extracting certain sequential patterns whose support exceeds a predefined minimal support threshold. Since the number of sequences can be very large, and users have different interests and requirements, to get the most interesting sequential patterns usually a minimum support is predefined by the users. By using the minimum support we can prune out those sequential patterns of no interest, consequently making the mining process more efficient. Obviously a higher support of a sequential pattern is desired for more

useful and interesting sequential patterns. However some sequential patterns that do not satisfy the support threshold are still interesting. Gaulet al. [2] introduced another metric called surprise to measure the interestingness of sequences. A sequence s is a surprising pattern if its occurrence differs greatly from the expected occurrence, when all items are treated equally. In the surprise metric the information gain was proposed to measure the overall degree of surprise, as detailed by [14]. Most of the basic and earlier algorithms for sequential pattern mining are based on the Apriori property proposed by Agarwal et al. [1]. The property states that any sub-pattern of a frequent pattern must be frequent. Based on this heuristic, a series of Apriori-like algorithms have been proposed: AprioriAll, AprioriSome, DynamicSome, GSP and SPADE Srikant et al. [3]. Yanet al. [4] a closed sequential pattern is a sequential pattern included in no other sequential pattern having exactly the same support. The first algorithm designed to extract closed sequential patterns is CloSpan with a detection of non-closed sequential patterns avoiding a large number of recursive calls.CloSpan is based on the detection of frequent sequences of length 2 such that "*A* always occurs Before after *B*". First, it adopts a novel sequence extension, called Bi-Directional Extension, which is used both to grow the prefix pattern and to check the closure property. Second, in order to prune the search space more deeply than previous approaches, it proposes a BackScan pruning method.Haoet al. [5] had worked on developing CSGM algorithmand uses a similar prefix-search-lattice data structure and the "projecteddatabase" concept as for CloSpan. The CSGM algorithm first scans the sequential database once, and finds all frequent length-1 sequences. These length-1 sequences are those patterns containing only one item. Since the generators of length-1 sequences are themselves, we put these sequences and a set of their corresponding generators together as sequence-generator pairs, and we also find the corresponding project databases for these sequences.

From the broad literature it was observed that the previous sequential pattern mining algorithm like Clospan, Sequence generator, closed sequence-sequence generator mining (CSGM). In sequential pattern mining and web log mining a traditional algorithm. Apriori is always reminded but due to some performance issues they were replaced with other algorithms and techniques. Many different techniques for mining frequent sequential patterns from the log data have been proposed in the recent past but still mining data from weblog files an effective and efficient algorithm is required that works with high performance. Moreover; it is required to authenticate the algorithm for that purposes we use a traditional algorithm for mining sequential pattern from web log data. Thus the aim of the present work is to bridge these gaps by developing and proposing a new algorithm "Sequential ID3" for sequential pattern mining and their experimental validation on web log data.

2. Methodology of Proposed Algorithm

Our project is designed with the main aim to mine log files and extract knowledge from the experimental web log and after training rules are generated these rules are helpful to find out different information related to log file. For that purpose we propose architecture to generate the rules from the experimental data set. This is done in these phases

1. Data selection
2. Data processing using selected model
3. Model building and model evaluation
4. Performance study

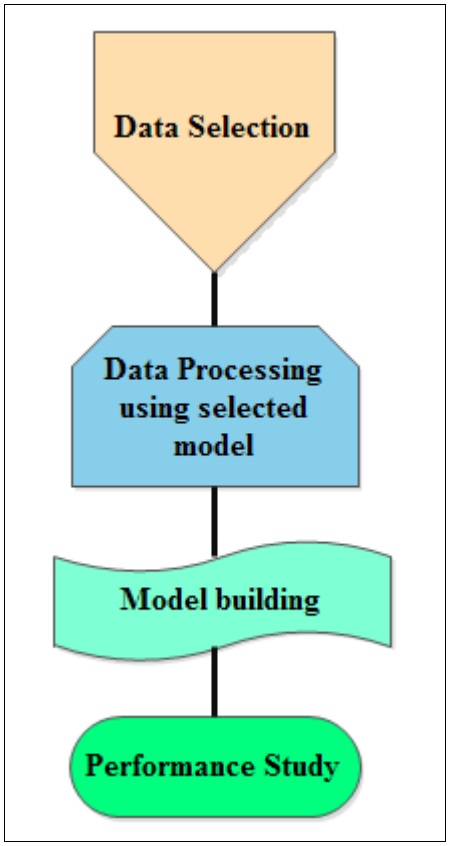

Figure 1. Basic structure of our proposed model

1. Experimental data selection: In this phase required to input log files in to the system for analysis the input log files are in w3c format
2. Data processing: In this phase system clean the data and separate them and arrange them.
3. Model building and evaluation: In this phase of system processing using the supplied data is converted in to data model using the selection of algorithm in other words selected data model is used to prepare a navigational model for queries of user.
4. Performance study: In this phase we calculate the performance parameters for results analysis.

Required Software and Hardware Specification

Tools-User Interface Design (UI Design) -Net Beans IDE 6.7.1

Technology/Framework-Framework-JDK 1.6

Hardware Specifications- 3 GB storage disk, 512 MB RAM (Min), Intel P4 Processor or higher

Software Specifications- Windows XP or higher

3. System Architecture

Figure 2 shows the system architecture of desired system. In this diagram we show the different sub systems of the complete system. These sub systems are work together and

form the complete system. To describe complete systems working we describe each stage of processing one by one.

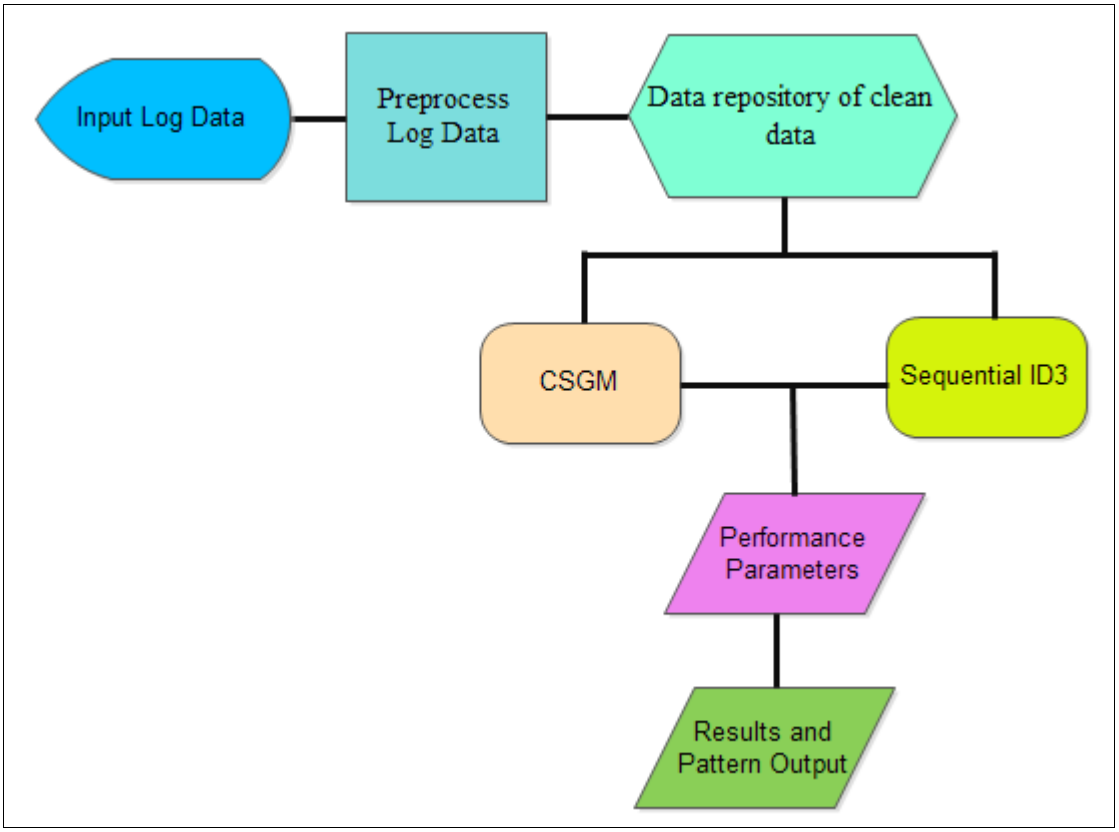

Figure 2. System architecture

3.1. Proposed Algorithm for Model Building

1. Import web log fie
2. Filter data in row column format
3. Find user sessions
4. User sessions defined as a class
5. Get all unique attribute values
6. Calculate the threshold according to class values using formula

n = no class in dataset.

$$\text{Entropy} = -\frac{\text{class (a)}}{\text{no. of row}}\log_n\frac{\text{class (a)}}{\text{no. of row}} - \frac{\text{class (b)}}{\text{no. of row}}\log_n\frac{\text{class (b)}}{\text{no. of row}}$$

7. Calculate info gain for all attributes using formula

n = no of attribute in a column.

Gain(S, A) is information gain of example set S on attribute A is defined as Gain(S, A) = Entropy(S) – Σ ((|Sv| / |S|) * Entropy (Sv))
Where:
Σ is each value v of all possible values of attribute A
Sv = subset of S for which attribute A has value v
|Sv| = number of elements in Sv
|S| = number of elements in S

8. Sort all attribute value accordingly to best attribute values.
9. Create Sub, Sets of all sorted data set.
10. Repeat till all attribute get a unique value.

Example

Table 1. Input data set

S.No	IP address	Method	URL	Agent
1	151.48.123.70	GET	http://www.smsync.com	Mozilla/4.0
2	151.48.123.70	GET	http://www.smsync.com	Mozilla/4.0
3	200.88.101.168	HEAD	http://www.123loganalyzer.com	Mozilla/5.0
4	200.88.101.168	GET	http://www.smsync.com	Mozilla/5.0
5	86.132.136.211	GET	http://www.123loganalyzer.com	Mozilla/4.0
6	151.48.123.70	HEAD	http://www.google.com/source	Mozilla/4.0

Unique values of IP address is =3
Unique values of Method is =2
Unique values of URL is =3
Unique values of IP address is =3
Unique values of Agent =2
If there is assume target value is agent.
Entropy of Input data set is

S = - (4/6) $\log_2$ (4/6)-(2/6) $\log_2$ (2/6)=0.39+.52=0.91

Notice entropy is 0 if all members of S belong to the same class (the data is perfectly classified). The range of entropy is 0 ("perfectly classified") to 1 ("totally random").

Suppose S is a set of 6 examples in which one of the attributes is Method. The values of Method can be GET and HEAD. The classifications of these 6 examples are 4

Mozilla/4.0 and 2 Mozilla/5.0. For attribute GET, suppose there are 4 occurrences of Method = GET and 2 occurrences of Method = Head. For Method = GET, 3 of the examples are Mozilla/4.0 and 1 are Mozilla/5.0. For Method = Head, 1 are Mozilla/4.0 and 1 are Mozilla/5.0.

Therefore

Gain(S,Method) = entropy(s) – 4/6*$Entropy_{Get}$ - 2/6*$Entropy_{Method}$

$Entropy_{Get}$ = - (3/4) $\log_2$ (3/4)-(1/4) log2(1/4)=0.311+0.5 =0.811

$Entropy_{Head}$=-(1/2)$\log_2$(1/2)-(1/2)$\log_2$(1/2)=0.15+0.15=0.3

Gain(s,Method)= 0.91-0.540-0.1 =0.27

For each attribute, the gain is calculated and the highest gain is used in the decision node.

Output:

Relation Name: Server Log File
Number of Instances: 24
Attributes:
Method
Requested_Value
Requested_Value = /images/download.gif
| Method = GET: http://www.123loganalyzer.com/
| Method = HEAD: http://www.123loganalyzer.com/
| Method = POST: null
Requested_Value = /images/samle.gif
| Method = GET: http://www.123loganalyzer.com/
| Method = HEAD: http://www.123loganalyzer.com/
| Method = POST: null
Requested_Value = /images/contact.gif
| Method = GET: http://www.123loganalyzer.com/
| Method = HEAD: null
| Method = POST: http://www.123loganalyzer.com/

4. Results and Discussion

To study the significance of the developed algorithm accuracy based testing is performed, for proving the utilization of new developed and improved sequential ID3 algorithm the results of the implementations were compared with the CSSGM algorithm.

Accuracy of the system is defined by the actually predicted values verses wrong values predicted. The accuracy of system is calculated using the cross validation in this method we calculate the values using given formula

$$Accuracy = \frac{totalvalues - wrongvalues}{totalvalues} X100$$

Accuracy of the system is derived using above formula, Table 2 demonstrates the results obtained by the system in six experiments conducted using the same parameters on CSSGM and developed Sequential ID3 algorithm.

Table 2. Comparative study of Accuracy of both CSSGM and Sequential ID3

Exp. no	CSSGM	Sequential ID3	No. of attributes
1	83.42 (support=2)	87.85%(No.of fold=2)	4
2	83.45%(support=3)	98.77%(No.of fold=3)	4
3	71.24%(support=4)	86.81%(No.of fold=4)	4
4	71.26%(support=5)	99.25%(No.of fold=5)	4
5	71.26%(support=5)	99.91%(No.of fold=5)	4
6	71.26%(support=6)	95.93%(No.of fold=6)	4

Figure 3 depicts the accuracy of the system using CSSGM and Sequential ID3 algorithm, and it can be seen from the figure that when as we minimize the support and increase the parameters accuracy of system decreases.

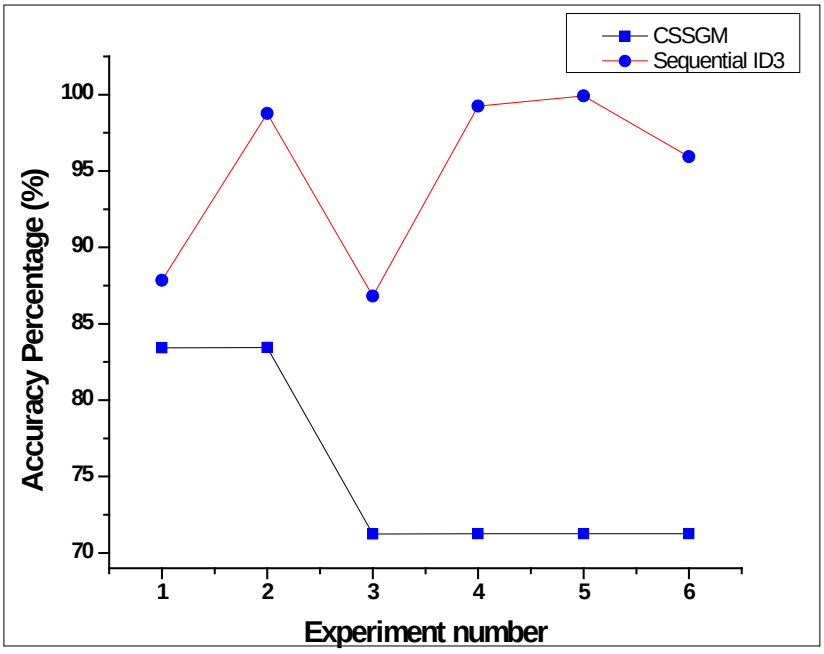

Figure 3. Graphical representation of Accuracy

Moreover it proposed method include all parameters and thus simulate better results for the evaluation of such kind of data.

Execution Time: To find the execution time we calculate the time required to build model results evaluation time included and we found that below given results.

Table 3.

Exp. no	CSSGM	Sequential ID3	No. of attribute
1	0.77 (support=2)	0.521 (No. of fold=2)	4
2	1.53 (support=3)	1.063 (No. of fold=3)	4
3	1.36 (support=4)	0.575 (No. of fold=4)	4
4	1.03 (support=5)	1.113 (No. of fold=5)	4
5	2.17 (support=5)	1.94 (No. of fold=5)	4
6	2.17(support=6)	0.873(No. of fold=6)	4

From Figure 4 its can be seen that execution time simulated by sequential ID3 algorithm is better than CSSGM. Because the CSSGM time consumption graph is more uneven than proposed algorithm. And it is also considered that most of the time our model is much efficient then CSSGM.

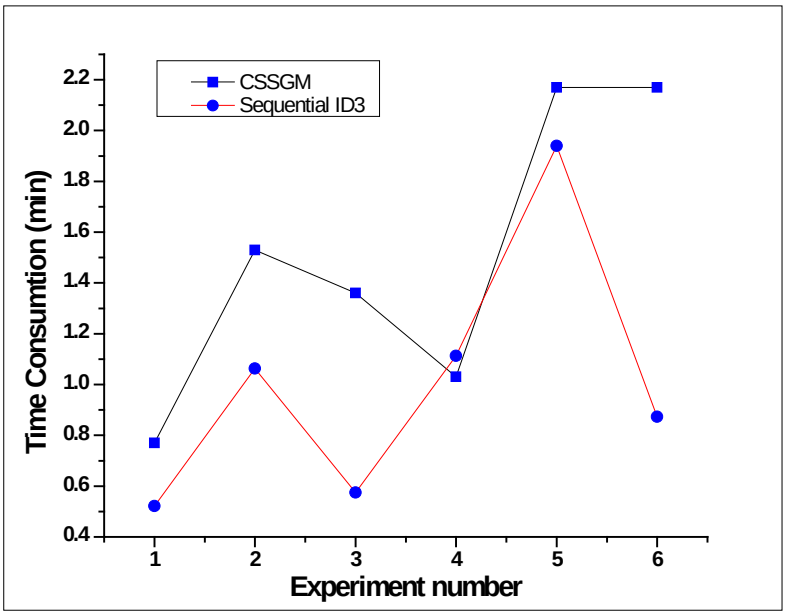

Figure 4. Graphical representation of Execution time

Memory uses: In any software system the major concern for developer is to reduce the use of the memory, thus main memory testing is performed to find the memory used by the sequential ID3 algorithm in comparison to the present CSSGM algorithm. The results simulate the memory used in terms of MB.

Table 4. Comparison of Memory Consumption of both CSSGM and Sequential ID3

Exp. no	CSSGM	Sequential ID3	No. of attribute
1	20.051(support=2)	81.49(No. of fold=2)	4
2	85.74(support=3)	104.79(No. of fold=3)	4
3	55.41(support=4)	51.49(No. of fold=4)	4
4	16.82(support=5)	47.18(No. of fold=5)	4
5	98.52(support=5)	57.50(No. of fold=5)	4
6	78.60(support=6)	119.64(No. of fold=6)	4

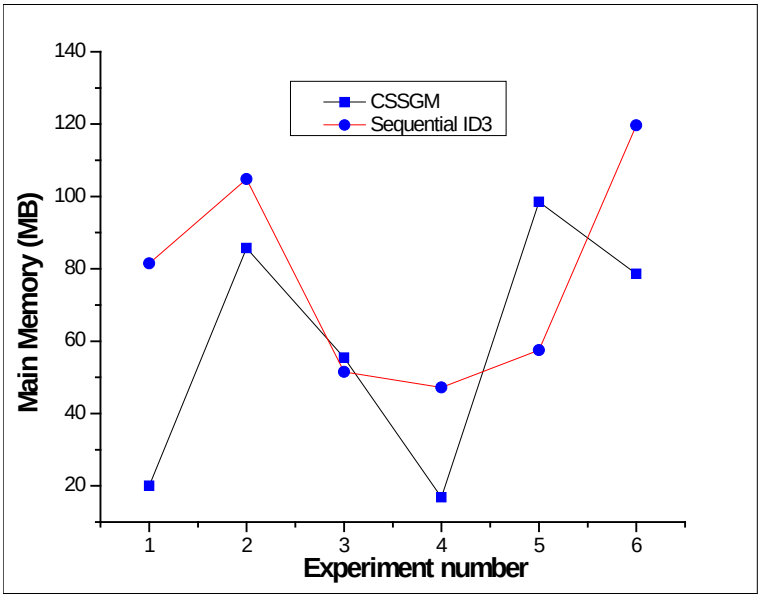

Figure 5. Graphical representation of Memory Consumption

Figure 5 depicts the memory consumptions using above results we can clearly see that CSGM algorithm consumes less memory then our proposed algorithm.

5. Conclusion

The aim of the present research work is to develop an algorithm to overcome the limitation of the old CSSGM algorithm and the results obtained in terms of accuracy, time consumption, and memory use clearly support that the new developed sequential ID# algorithm has the potential as an alternate algorithm. The obtained results can summaries as follows:

1. Accuracy of proposed algorithm 75%-85% is better than CSGM algorithm.
2. Memory uses of proposed algorithm found higher than Apriori.
3. Time required to execute model is 85%-95% less than CSGM algorithm
4. Proposed algorithm is good algorithm but when where required less resource it is fail to work with low configuration system.
5. Memory Uses of proposed algorithm is 80%-85% is higher than CSGM.
6. Thus the new developed algorithm sequential ID3 is useful and far better than the existing algorithms in terms of time consumption and accuracy but it lacks in memory consumption only.

References

[1] Agarwal, R., and Srikant, R. Mining sequential patterns. Proceedings of the Eleventh International Conference on Data Engineering, 1995.

[2] Gaul, W., and Schmidt-Thieme, L. Mining Generalized Association Rules for Sequential and Path Data. Proceedings of the 2001 IEEE International Conference on Data Mining, 2001.

[3] Srikant, R., and Agarwal, R. Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, 1996.

[4] Yan, X., Han, J., and Afshar, R. CloSpanMining Closed Sequential Patterns in Large Datasets. Proceedings of the SIAM International Conference on Data Mining (SDM'03)2003.

[5] Hao zang, and yue xu. Non redundant Sequential association rule mining and application in recommender System.IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010.

[6] Xu, Y., & Li, Y. Concise representations for approximate association rules. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, SMC, 2008.

[7] Li, J., Li, H., Wong, L., Pei, J., & Dong, G. Minimum description length principle: generators are preferable to closed patterns. Proceedings of the21st National Conference on Artificial Intelligence, 2006.

[8] Desikan, P., Pathak, N., Srivastava, J., and Kumar, V. Incremental page rank computation on evolving graphs. Paper presented at the Special interest tracks and posters of the 14th International Conference on World Wide Web, 2005.

[9] Cooley, R. Web Mining: Information and Pattern Discovery on the World Wide Web. Proceedings of the 9th International Conference on Tools with Artificial Intelligence.1997.