International Journal on Emerging Technologies **11**(2): 81-91(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Performance Analysis of Sequential Rule Mining Techniques for Web Page Recommendation System

Satyaveer Singh[1] and Mahendra Singh Aswal[2]
[1]Research Scholar, Department of Computer Science,
Gurukul Kangri Vishwavidyalaya,Haridwar (Uttarakhand), India.
[1]Assistant Professor, Department of Computer Science and Engineering,
Maharishi Markandeshwar (Deem to be University), Mullana (Haryana), India.
[2]Assistant Professor, Department of Computer Science,
Gurukul Kangri Vishwavidyalaya, Haridwar (Uttarakhand), India.

(Corresponding author: Satyaveer Singh)
(Received 19 November 2019, Revised 10 January 2020, Accepted 18 January 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Recommendation systems aim to make valuable suggestions to users, by taking into consideration their profile, preferences and/or actions throughout interaction with an application or website. A number of studies have presented the use of sequence mining from the web usage dataset that plays an important role for generating web page recommendation to web users. However, it is a big challenge to select an effective sequential mining algorithms for discovering the web usage knowledge. The paper presents taxonomy of the existing sequential rule mining algorithms and compares them in tabular form based on the different key features. This paper makes an attempt to give a comparative performance analysis of two excellent algorithms, i.e., RuleGrowth and RuleGen. A web page recommendation system based on sequential rule mining discovered from web usage data has also been presented along with a detailed comparative evaluation. For a given user's web navigation sequence, the web page recommendation system provides recommendations on the basis of the generated sequential rules. These results are used to pick a suitable sequence mining algorithms for web page recommendation system developers.

Keywords: Sequential pattern mining, Sequential rule mining, Recommendation System, Web usage mining.

I. INTRODUCTION

With the rapid increase of web pages on World Wide Web, it's become a warehouse of the vast amount of data and information. It became a real challenge to provide the relevant web pages to the users with diverse needs. Therefore, exploring the web usage data is used to understand the web user navigation behaviours and discover valuable usage patterns. Web usage mining can appropriately study the user's navigational behaviours and determines the usage patterns based on their current browsing history [1]. In the context of web page recommendation system, these extracted usage patterns are assisting users to automatically recommending web pages. To recommend accurate information to online users, the selection of the efficient mining algorithm plays a noteworthy job.

Web usage mining is primarily interested in knowledge extracted from web server files, which is used to discover user behavioural patterns on the web. Generally, the discovered patterns are a set of sequences which are frequently used by groups of users with a common interest. The generated pattern retains the sequence of information in the navigation. So, unlike earlier approaches, sequential rule mining techniques are preferred. They are suitable for this purpose since they take input in the form of web access sequences and produces sequential rules as output which is in a sequential form that let us identify which pages are visited and in what order. An extensive literature on the survey of sequential pattern mining algorithms is available [2]. Sequential rule mining is a variation of the sequential pattern mining. It has been found from the literature that sequential rule mining algorithms can be categorized as the apriori based, pattern growth based and early pruning approaches [3].

Firstly, the main sequential rule mining algorithms are reviewed and compared in a tabular form based on the different key features. Secondly, the performance of the two algorithms RuleGrowth and RuleGen are compared using the dataset of three real websites based on execution time, a number of sequential rules and maximum memory usages [4, 5].

Finally, these two algorithms are applied on a web page recommendation system to observe the accuracy of the recommendation rules.

The organization of the paper is as follows: Section II describes the taxonomy of sequential pattern mining algorithms and also presents the tabular comparison of mining algorithms based on different key features. Section III presents a web page recommendation system framework which generates the recommendation set for users based on the current user's navigation history and sequential rules. These sequential rules are discovered with the help of sequential rule mining techniques. In section IV, comparison of two mining algorithms and the performance evaluation of web page recommendation system are presented. The paper is concluded in section V.

II. WEB USAGE MINING AS SEQUENTIAL RULE MINING APPLICATION

Web usage mining is considered as the main application of sequential rule mining techniques on user navigational patterns to extract knowledge about the preferences and behaviour from web log files, where, usage knowledge is successive in nature, i.e., each bit of knowledge is associate ordered list/sequence of visited web pages [6]. In the literature, sequential rule mining techniques have been introduced as main web usage mining techniques [6]. This section reviews the major existing sequential rule mining techniques. But, our main focus is on two pattern growth based algorithms namely RuleGrowth and RuleGen [4, 5].

A. Web Usage Mining

Server log files are the primary information resources for web usage mining in which navigational activities of web users are recorded. Web usage mining consist of three phases, i.e., pre-processing, pattern discovery and pattern analysis [7]. In the pre-processing phase, a web server log file is organized into user session file so that the useful pattern discovery and analysis is achieved. Once the pre-processing phase is successfully completed, user session file is converted into web access sequences and stored into a dataset [8]. During the pattern discovery phase, the important web usage knowledge is extracted from web access sequences using some data mining techniques. In the last phase, the discovered knowledge will be utilized in any particular application, e.g. web page recommendation system, within which recommendation rules can be generated using this knowledge to support web page recommendation system. In this research work, sequential rule mining techniques have been used for the pattern analysis. It is a key usage of the web usage mining process and performs an important job in a web page recommendation so that users can be supported to form good decisions on their current web navigation history.

B. Sequential Rule Mining Algorithms

Sequential rule mining has discovered sequential rules of the form A → B [4, 9, 10] where some items B will be followed to some items A which appear in sequence with a certain confidence. It is an alteration of sequential pattern mining problem. The sequential rules and association rules have similar concept except that A must occur before B according to the sequential ordering. These sequential rules are mined in sequences instead of the transaction. The main drawback of sequential pattern mining is that it considers some sequential patterns which frequently occur in sequence database and have a very low confidence. These patterns have been useless for decision-making or prediction. This problem is addressed by sequential rule mining. For example, Table 1 has shown the sequence database. We assumed here, if minimum support of sequential pattern is two then it is frequent. The sequential pattern {(u) (v)} is treated frequently because its minimum support is two. It appears in two sequences. Therefore, it can be tempting to think that {v} can be followed in other sequences after {u}. However, this is not the case. Table 1 shows that {u} is really followed by {v} in only two of the four sequences where {u} occurs. It is shown in this example that sequential patterns can be misleading.

Table 1: A sequence database.

SID	Sequence
1	< {z, y}, {x}, {u, t},{t} {v}>
2	<{z, w},{x},{y},{z, y, v, u}>
3	<{z},{y},{u, t},{v}>
4	<{y},{u, t}>

Sequential rules have considered their support and confidence to tackle this problem. For example, it has shown in the sequence database, the sequential rule {u} → {v} is frequent but not a strong rule because it has a support of 2 sequences and a confidence of 50%. Formally, the confidence of a sequential rule A → B is defined as the number of sequences containing the items A before the items B divided by the number of sequences containing the items A [10]. Sequential rules have been described as more beneficial than sequential patterns for some of the tasks involved in predictions [10]. Sequential rule mining has diverse applications such as e-learning, web page pre-fetching, anti-pattern detection, stock market analysis, weather observation, drought management, restaurant recommendation and alarm sequence analysis [4, 10].

Fig. 1 shows the taxonomy of sequential rule mining algorithms. These algorithms are compared in tabular form based on different key features as given in Table II. Sequential rule mining algorithms can be broadly categorized into three major categories such as apriori based, pattern growth and early pruning approaches [2, 3]. The algorithms are different mostly in two manners [3]. The first manner in which candidate sequences are produced and stored. The key objective here is to reduce I/O cost by reducing the number of candidate sequences generated. The second manner in which support is calculated and how to test candidate sequences for frequency. For the support counting purposes, at all the times a database has to be maintained. The key plan here is to eliminate the database.

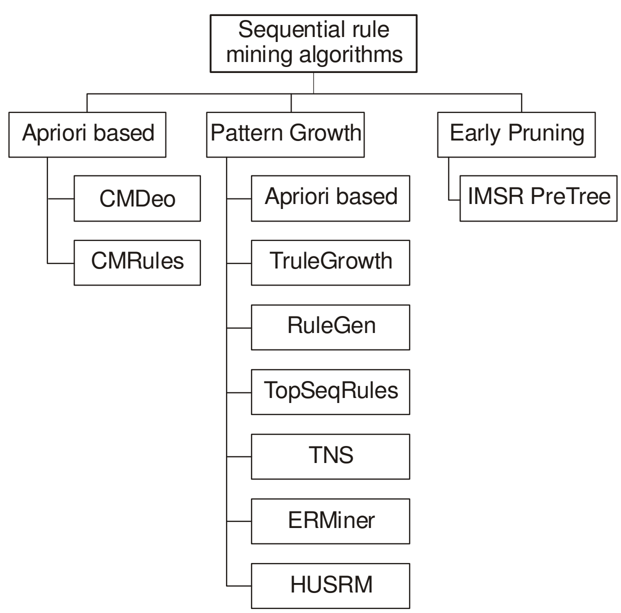


Fig. 1. Taxonomy of Sequential Rule Mining Algorithms.

(a) **Apriori Based:** At the each k^{th} iteration, for finding frequent itemsets of size k, the database is scanned many times by apriori based algorithms and then carries out an extensive join operation to produce a huge set of candidate sequences. Candidate sequences that do not satisfy the apriori property are deducted and until there are no more candidate sequences. As a result, the numbers of sequences in the database get enlarged and mining process takes more time. The significant features of apriori based algorithms such as Breadth first search (BFS) generate and test, and multiple scans of the database that cause serious problems and hinder the effectiveness [3].

The typical apriori based algorithms are CMDeo and CMRules [10, 11] which use generate and test method in which the rules are formed and then looked into the database to find out their support and confidence. These algorithms can use breadth first search technique to find the search space of rules. A large quantity of candidate rules is usually generated by the CMDeo [10] which is the main drawback of it. A big proportion of these rules is worthless or does not show in the database. Therefore, enough time is spent by the algorithm to inform distant valid rules from invalid ones. CMRules [11] is more capable for low support thresholds and has a better scalability. Its performance decreases as the number of rule increases and becomes inefficient when the dataset is large.

(b) **Pattern Growth:** Pattern based methods came into existence as a solution to the problems of apriori based algorithms. The most important consideration of this approach is to avoid the problem of the candidate sequence generation and to see only the limited segment of the primary database. The construction of a database representative is the goal of pattern growth algorithms and then it divides search space into a proper manner and the previously extracted frequent sequences increase for as possible as to generate a few candidate sequences. After that, a frequent sequence is searched using apriori property. RuleGrowth [4] algorithm relies on the pattern growth approach and avoids the candidate sequence generation problem by keeping the track of the first and last events of each item. An alternative of the RuleGrowth [4] algorithm is TRuleGrowth [10] which use the sliding window size constraint. A number of significant benefits are revealed by discovering the rules using this constraint. It can reduce the execution time by pruning the search space, reducing disk space requirement by generating the smaller set of rules and also increasing the prediction accuracy. This constraint could also be added with CMDeo and CMRules algorithms [10, 11]. However, it performs better with RuleGrowth [4] algorithm because window constraint is checked by it when sequences are scanned to search items. Whereas, CMDeo and CMRules algorithms can only confirm that rules admire the window size constraint after rules have been generated [10, 11]. However, it is very difficult to choose a suitable window size because it is depends on dataset. When window size is set to large value, the execution of TRuleGrowth becomes slower than RuleGrowth because TRuleGrowth has to perform extra calculations for verifying the window size constraint.

For mining sequential rules from the dataset, it is necessary for the user to state the minimum support and confidence parameters that are hard to set. TopSeqRules algorithm resolves this issue by letting the users directly the number of rules to be discovered which are specified by the value of k [12]. It is based on the most valuable sequential rules mining algorithm which is used in the RuleGrowth [4]. However, it should be noted from the literature that the computation cost of the problem of top k sequential rule mining is more than the problem of sequential rule mining. Thus, depending on the dataset for the values of k up to 1000 or 2000, it is suggested to utilize TopSeqRules [12]. If the number of rules is found more then, it would be better to employ RuleGrowth or TRuleGrowth [4, 10] algorithm for more efficiency. The main difference between the performance of TopSeqRules and RuleGrowth is in the memory usage. TopSeqRules uses more memory because it keeps the set R of rules to expand onto memory.

There is a variant of the TopSeqRules [12] which is named TNS (Top-K Non-Redundant Sequential Rules) [13]. The enhancement in TNS [13] is that it always generates the top k non-redundant sequential rules. TNS [13] algorithm is more expensive than TopSeqRules [12] but it takes the advantage of removing some redundancy in the result. ERMiner [9] algorithm is the variation of the RuleGrowth [4] that uses the equivalence classes concept to find out the sequential rules. It resolves the frequently carrying out of datasets holding dense or long sequences issue that is generated by the RuleGrowth [4] for mining the sequential rules. It can be faster than RuleGrowth but usually consumes more memory [4].

RuleGen algorithm was proposed by Zaki that discovers sequential rules appearing in sequence databases [5]. Firstly, sequential pattern mining algorithm applies to the sequence database and then the pairs of sequential patterns join between two sequential patterns to generate the rules. Apart from this, it is significant to remain noted that the generated rules always have the form X → Y where X is a subsequence of Y. Whereas, in the case of other sequential rule mining algorithms such as CMDeo [10], CMRules [11], RuleGrowth [4], TRuleGrowth [10], TopSeqRules [12] and TNS [13] where X and Y are unordered item sets and X is not a subset of Y. It is found that as compared to RuleGen [5], more general rules are generated by these algorithms. Moreover, it has been found in the literature that higher prediction accuracy is achieved by using the rules generated by RuleGrowth [4] and CMRules [11] as compared to using rules generated by RuleGen [5].

Several sequential rule mining algorithms have been proposed in the last decade. However, the number of items in sequences and their unit profit is not considered by these algorithms. Although, for product recommendation and market basket analysis based applications, it is very important and useful. To consider the profit and quantities of items in sequences while mining sequential rules by the algorithm is thus a significant research problem. HUSRM (High-Utility Sequential Rule Miner) [14] has resolved this problem by formalizing the high-utility sequential rule mining and its properties.

(c) **Early Pruning:** Early pruning algorithms reduce the candidate sequences very quickly in the mining process by using some sort of position induction and prevent the support counting as much as possible. A table is used by these algorithms to follow the previous position of each item in the sequence and which is helpful in early candidate sequence pruning. Because, based on the previous position of an item, it is decided if an item can be added in the order of a given prefix or not. Therefore, support counting and candidate non-frequent sequences generation can be avoided by these algorithms.

IMSR_PreTree [15] is an enhanced version of MSR_PreTree [16] applied on sequence databases to discover sequential rules. The reduction in the cost of sequential rule mining process is the purpose of this enhancement. In the early stage of mining, non-critical rules are discovered which are reduced by pruning subtrees. During the mining process, the search space can be decreased by this algorithm which is extremely helpful in big databases mining process.

Table 2: Tabular Comparison of Sequential Rule Mining Algorithms.

Algorithm Key Features	Apriori Based		Pattern Growth							Early Pruning
	CMDeo [10]	CMRules [10, 11]	RuleGrowth [4]	TRuleGrowth [10]	RuleGen [5]	TopSeqRules [12]	TNS [13]	ERMiner [9]	HUSRM [14]	IMSR_PreTree [15]
Generate and Test	Yes	Yes	No	No	No	No	No	No	No	No
Database Multiple Scan	Yes	Yes	No	No	No	No	No	No	No	No
BFS Traversal	Yes	Yes	No	No	No	No	No	No	No	No
Bottom Up Search	Yes	Yes	No	No	No	No	No	No	No	No
DFS Traversal	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Database Vertical Projection	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Top Down Search	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Support Counting Avoidance	No	No	No	No	No	No	No	No	No	Yes
Position Coded	No	No	No	No	No	No	No	No	No	Yes

C. Comparison

The above mentioned sequential rule mining are compared based on the different features as shown in Table 1 and as follows:

– The main drawback of the apriori based algorithms is that expensive candidate generate and test, and multiple database scan, when applied to mine long sequential patterns and also hinder the performance of the mining process.

– The pattern based algorithms do not use generate candidate and test and multiple database scan approach. Therefore, these algorithms can be much more efficient and scalable.

– It can be seen that except the RuleGen algorithm, all other pattern based algorithms are the variation of the RuleGrowth algorithm. But, they consumes more memory than RuleGrowth algorithm.

III. WEB PAGE RECOMMENDATION SYSTEM

Web page recommendation system is a process that automatically recommends web pages to users based on their current navigation behavior. It is shown from the literature that model based collaborative filtering technique has been mostly used by web page recommendation system framework [17]. In the current research, we used this filtering technique for web page recommendation system framework. Fig. 2 illustrates a framework for web page recommendation system. The framework consists of two parts: offline phase and online phase. The overall working of the framework is based on the classical web usage mining systems.

In the offline phase, web server log file pre-processing, web access sequence generation and rule extraction are done. The basic web log cleaning technique is used for pre-processing [18].

The pre-processing step removes the non-responded web requests and requests made by software agents such as web crawlers. After that, user identification and session identification must be done for the extraction of navigation history of each session i.e. web access sequence from the log file.

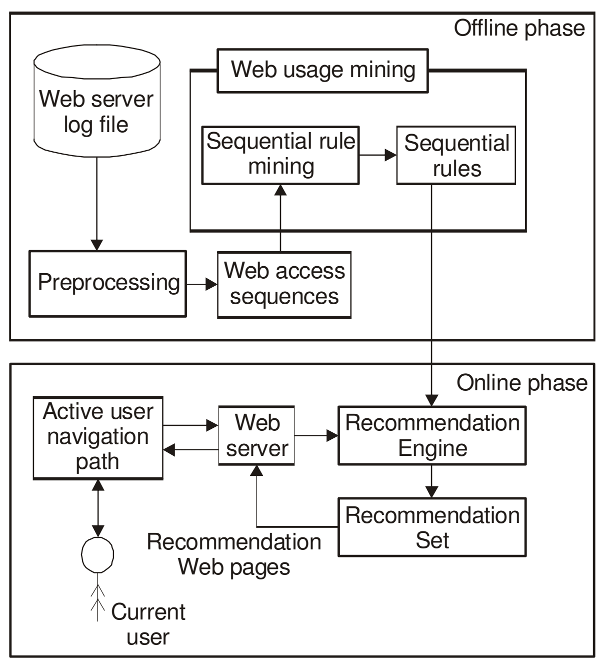


Fig. 2. Framework for Web Page Recommendation System.

Sequential rule mining technique takes the web access sequences as the input and generates the sequential rules. In this framework, we used the RuleGrowth [4]

and RuleGen [5] algorithms for extracting the sequential rules. These algorithms are implemented using the SPMF [19] open source data mining framework.
In the online phase, active user navigational path is fed into recommendation engine module. The recommendation engine compares the active user navigation path with the sequential rules and generates the recommended pages for the user's next move in his or her navigation.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experiments were conducted on three web log files that belong to real world websites which have been chosen as the datasets. The Clark School Administrative Portal (ClarkNet dataset) log file and the University of Saskatchewan (Sask dataset) log file can be openly downloaded from http://ita.ee.lbl.gov/html/traces.html [20]. These log files have been usually helpful in various research works for experimental activities and were created in 1995.Microsoft anonymous web data (Microsoft dataset) log file can be openly downloaded from http://kdd.ics.uci.edu/databases/msweb/msweb.html [21] which was created in 1998. The dataset records of these websites has shown in Table 3.

Table 3: Dataset Records.

Dataset	Number of Sessions	Source
ClarkNet	2552	The Clark School Administrative Portal
Sask	808	University of Saskatchewan
Microsoft	1509	Microsoft anonymous web data

The experiments were carried out on a system having Intel Core i3 processor with a CPU clock rate of 2.40 GHz, 3GB of main memory and running on Windows 7 platform. The dataset is partitioned into two parts. The 75% of the dataset is treated as the training dataset and the remaining 25% of the dataset is treated as a testing dataset. By applying the sequential rule mining algorithm on training data set, sequential rules are generated. Here, the execution time, memory usage and the number of generated sequential rules of the two selected mining algorithms are calculated. Also, the precision, coverage and m-metric of the three datasets with different minimum supports are calculated. Finally, the accuracy of the web page recommendation system is compared based on two selected algorithms for the performance evaluation of algorithms.
The execution times of the two mining algorithms are illustrated in Fig. 3 with distinct minimum support threshold values along the three datasets. The runtime of the RuleGen algorithm is more in comparison with RuleGrowth algorithm against the three datasets that can be shown in Fig. 3. Once the support threshold value becomes small, it shows that there is a diverse variance between RuleGen algorithm and RuleGrowth algorithm. The number of sequential rules acquired from of the two mining algorithms is illustrated in Fig. 4 with distinct minimum support threshold values along the three datasets. In most test cases, the sets of sequential rules produced by RuleGrowth are subsets of the rules produced by RuleGen which have been shown in Fig. 4. That's why, it can be described that the execution time of RuleGrowth is always much less than that of RuleGen, as shown in Fig. 3. The memory usages of the two mining algorithms are illustrated in Fig. 5 with distinct minimum support threshold values along the three datasets. The memory usages of RuleGrowth are slightly more as compared to RuleGen against the three datasets which have been shown in Fig. 5. It can be noticed from Fig. 5 that RuleGen is more memory efficient than RuleGrowth.
The performance of the web page recommendation system is evaluated based on the metrics such as precision, coverage and m-metric defined in [22]. The efficiency of the system is measured based on these metrics using RuleGrowth [4] and RuleGen [5] algorithms.Our evaluation methodology is as follows. Each transaction t in the testing dataset is divided into two parts. The first n web pages in t are used as the input of the recommendation engine for generating the recommendations. The remaining part of t is used to evaluate the generated recommendation which is denoted as $Eval\_list_t$. Once the recommendation engine generates a set of web pages, which is denoted as Rec_list_t, the set is compared with $Eval\_list_t$ web pages.
The precision of a transaction t is specified as the ratio of a number of web pages exactly predicted to the total number of web pages predicted.

$$\text{Precision}_t = \frac{|\text{Rec\_list}_t \cap \text{Eval\_list}_t|}{|\text{Rec\_list}_t|} \quad (1)$$

The coverage of a transaction is given as the ratio of a number of web pages exactly predicted to the total number of web pages visited by the user.

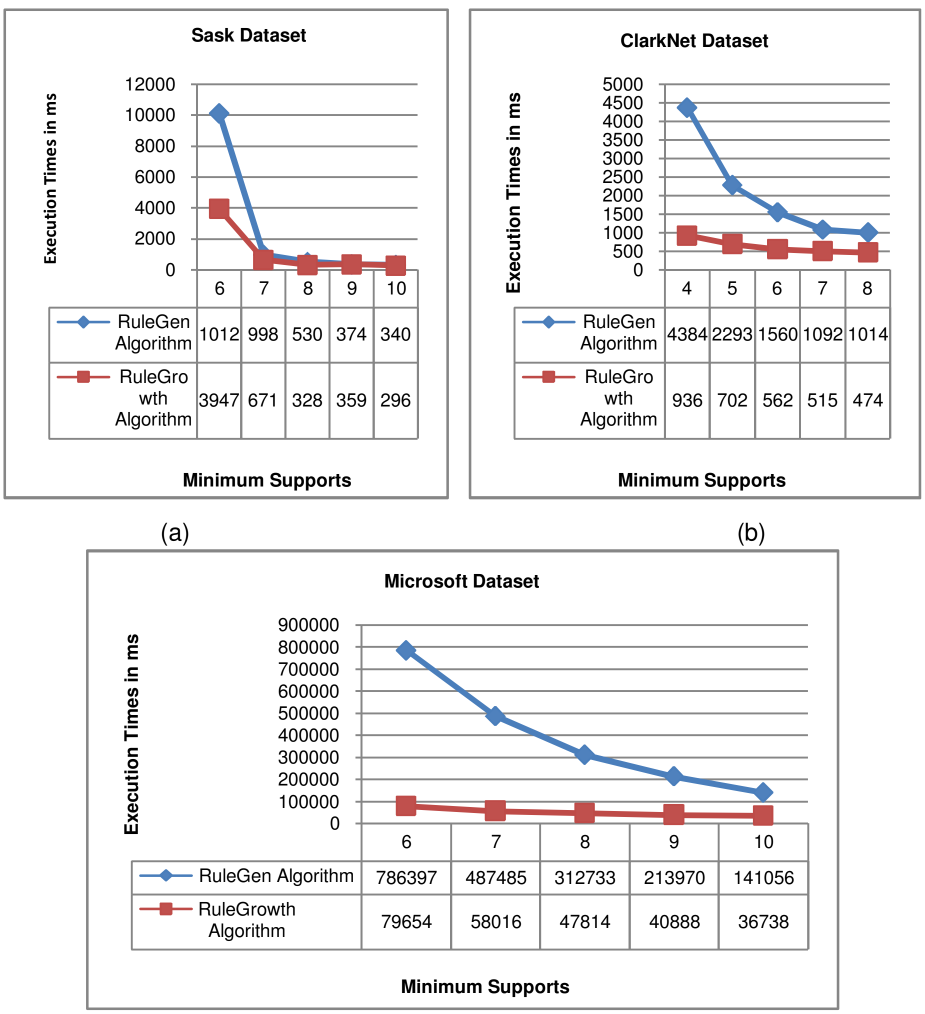
$$\text{Coverage}_t = \frac{|\text{Rec}_{\text{list}_t} \cap \text{Eval}_{\text{list}_t}|}{|\text{Eval}_{\text{list}_t}|} \quad (2)$$

The m-metric can be interpreted as a weighted average of the precision and coverage. It is evaluated to achieve its maximum value when both $coverage_t$ and $precision_t$ are maximized.

$$\text{M} = \frac{2 * \text{Coverage}_t * \text{Precision}_t}{\text{Coverage}_t + \text{Precision}_t} \quad (3)$$

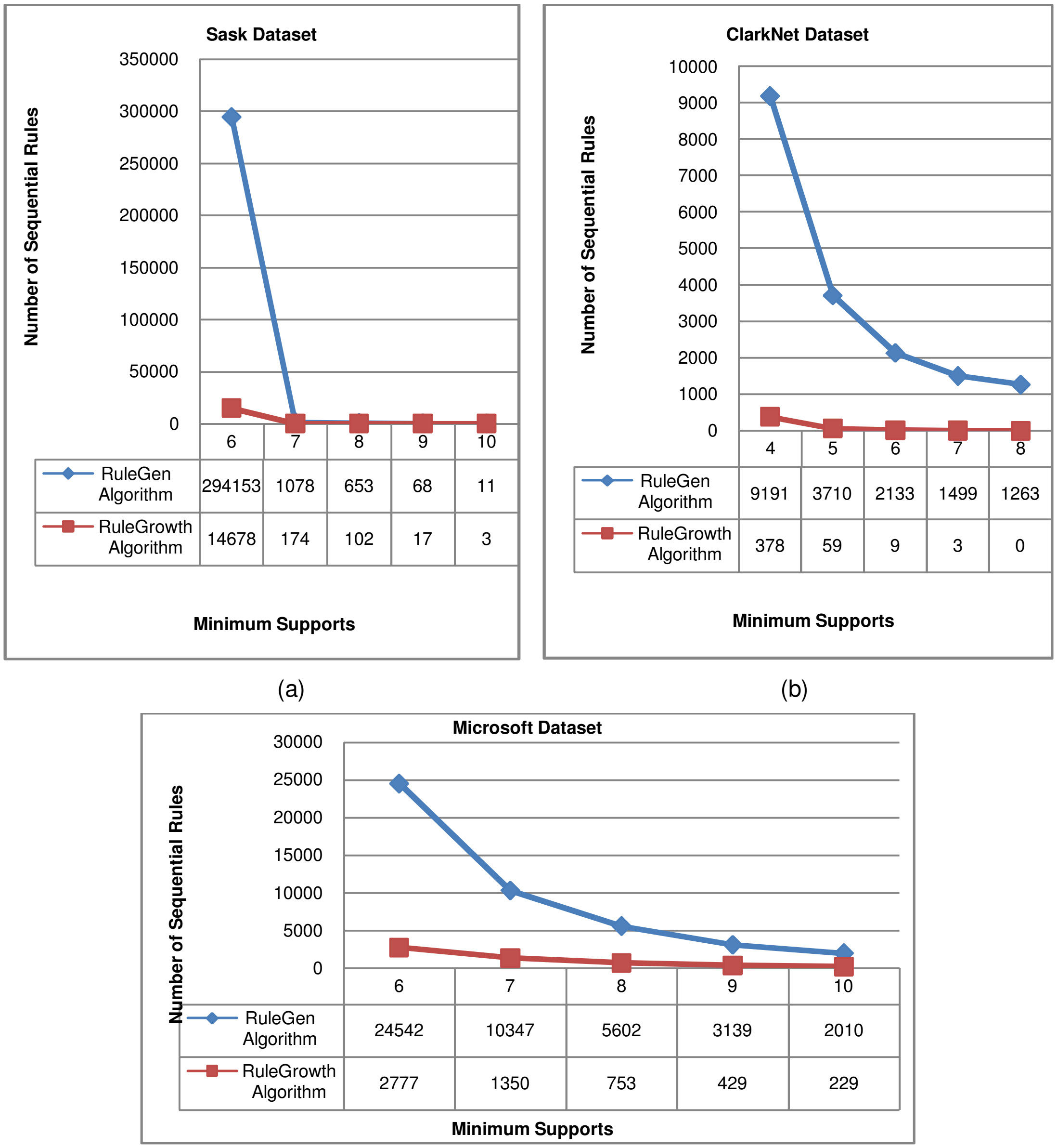
In the experiment, the precision, coverage and m-metric have been evaluated for all the transactions in the testing dataset and their averages have been calculated. The average precision and average coverage values assist to evaluate the system.
In our experiment, Fig. 6 shows the precision of the recommendation system against the three datasets. It is expected that when the minimum support increases, the precision decreases. This is due to the fact that when minimum support increases, the number of sequential rules tends to decrease as shown in Fig. 4 and the ratio of the number of the accurate sequential rules to the number of all rules also decreases. Therefore, more accurate recommendations need to be generated. In case of the RuleGrowth algorithm, as shown in Fig. 6(a) and (c) initially, the precision of the recommendation system increases when the minimum support increases (sequential rules also decreases). This can be attributed to the fact that irrelevant sequential rules are eliminated and more accurate recommendation can be generated. However, after a breaking point (8% minimum support threshold), the precision of recommendation system starts to decrease because the number of sequential rules and recommendations decrease.

In case of the RuleGen algorithm, as shown in Fig. 6(a), initially, the precision of the recommendation system decreases when the minimum support increases. However, after a breaking point (8% minimum support threshold), the precision of recommendation starts to increase when the number of sequential rules and recommendations decrease, the precision of recommendation system is expected to decrease here. In case of the RuleGrowth algorithm, as shown in Fig. 6(b), as expected, the precision of recommendation system decreases when the minimum support increases. However, in case of the RuleGen algorithm in Fig. 6(b) and (c), the precision of the recommendation system increases with the increase in the minimum support values (decrease in a number of sequential rules). Therefore, Fig. 6 illustrates that RuleGrowth algorithm generates more precise results for the precision of the recommendation system as compared to RuleGen algorithm against the three datasets.
Fig. 7 shows the coverage of the recommendation system for the two algorithms. It decreases constantly when the minimum support values increase except for the RuleGen algorithm against the Sask dataset in which coverage increases after the breaking point. This is expected since the coverage is the ratio of the exactly predicated pages in the evaluation part to the number of all pages in that part and the number of pages in the evaluation part is always constant. It is already clear that with the increase in minimum support values, the number of sequential rules decreases. This will further cause the number of the exactly predicated pages to decrease. Therefore, the coverage value should decrease always. But in case of the RuleGen algorithm against the Sask dataset, coverage increases after the breaking point as shown in Fig. 7(a). Therefore, Fig. 7 illustrates that RuleGrowth algorithm provides more justifiable results for the coverage of the recommendation system as compared to RuleGen algorithm against the three datasets.
Fig. 8 demonstrates the evaluation results in terms of m-metric of the two mining algorithms with different minimum support threshold values in the three datasets. It shows that results of m-metric of the recommendation system are accurate in case of RuleGrowth as compared to RuleGen against the three datasets. This is expected because RuleGrowth generates more accurate results for precision and coverage as compared to RuleGen as shown in Fig. 6 and 7 respectively.



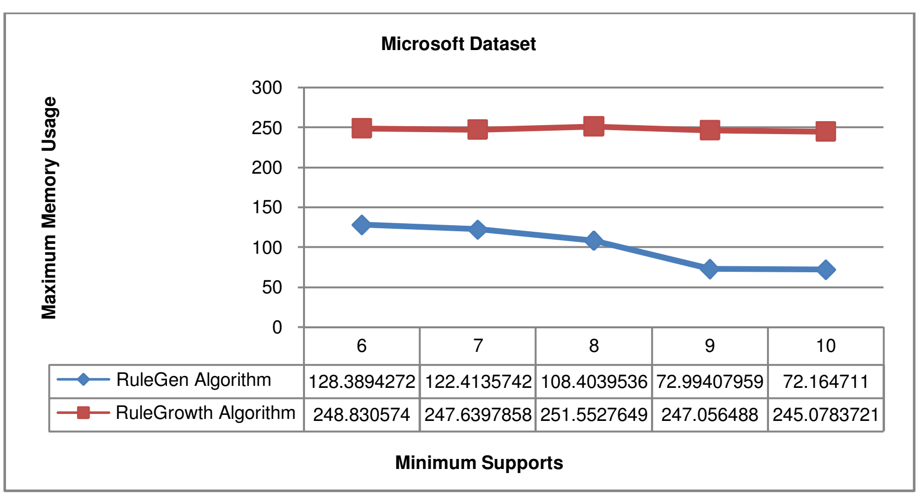
(c)

Fig. 3. Execution Times of the two mining algorithms with different support threshold values.

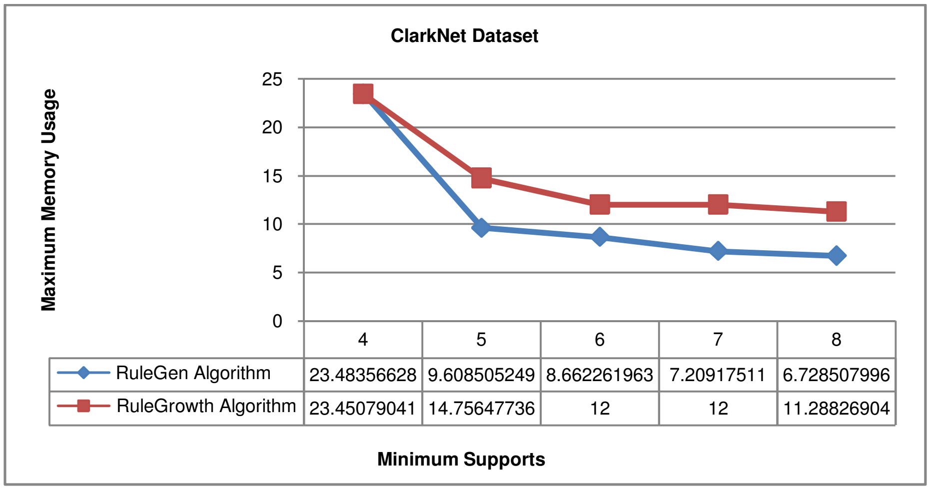


(c)

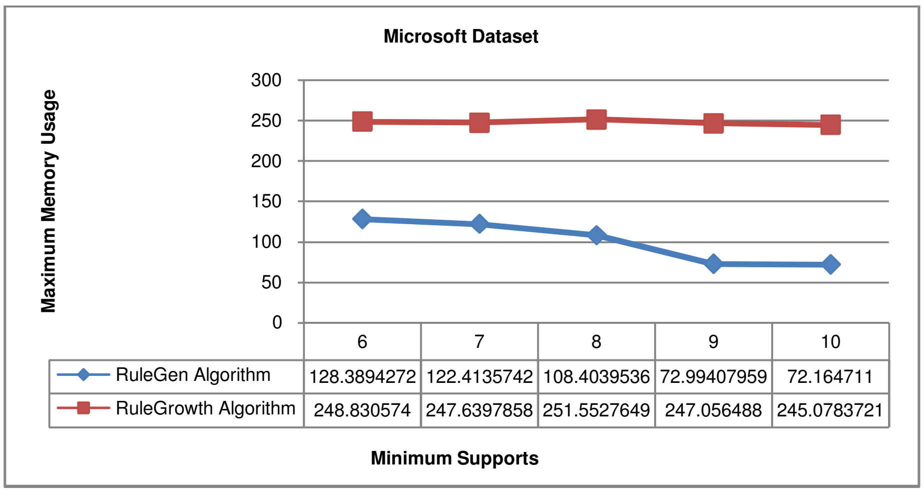
Fig. 4. Number of Sequential Rules acquired from the two mining algorithms with different support values.



(a)

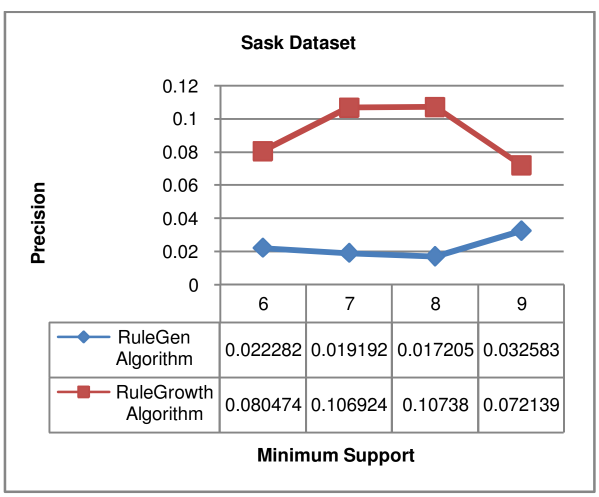


(b)

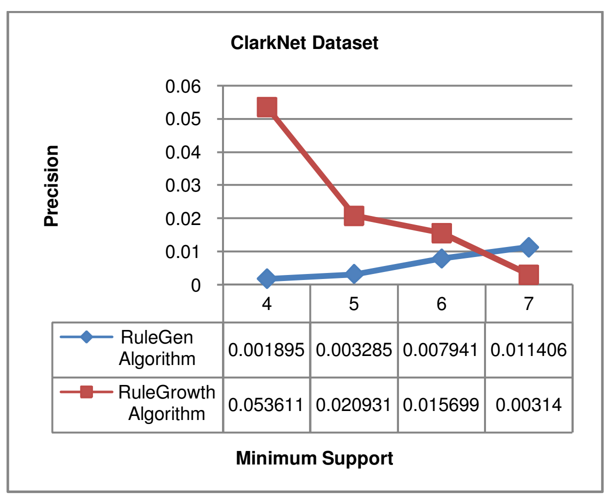


(c)

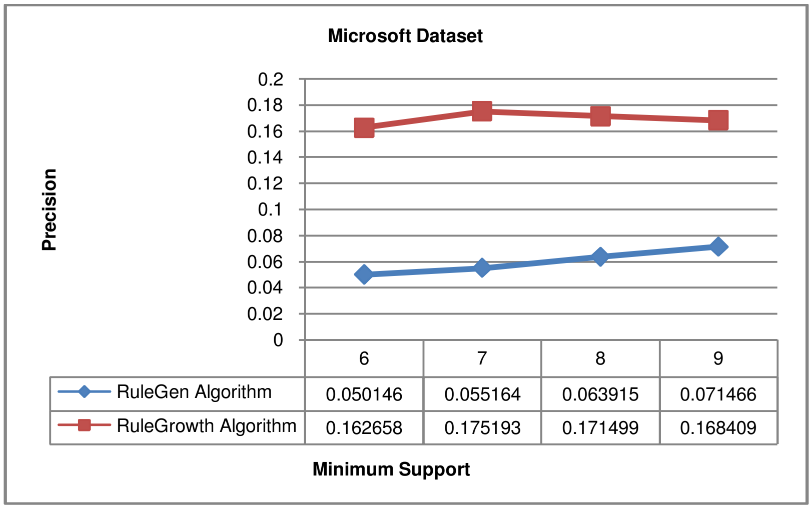
Fig. 5. Maximum Memory Usages of the two mining algorithms with different support values.



(a)

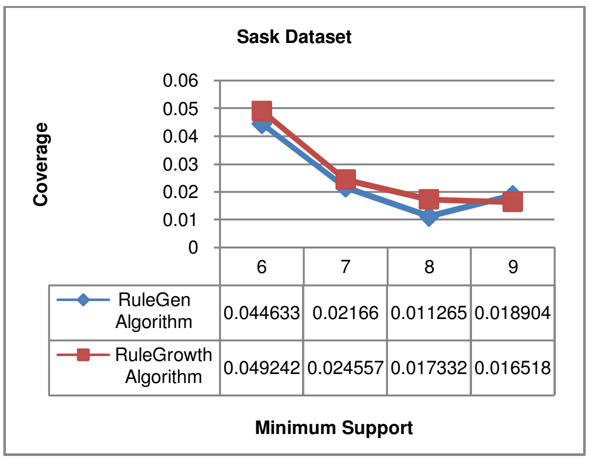


(b)

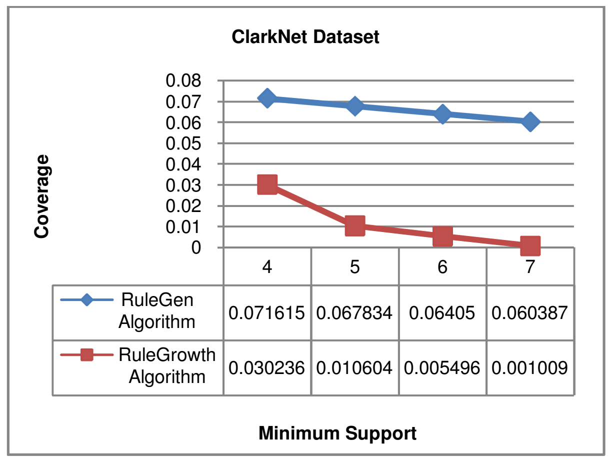


(c)

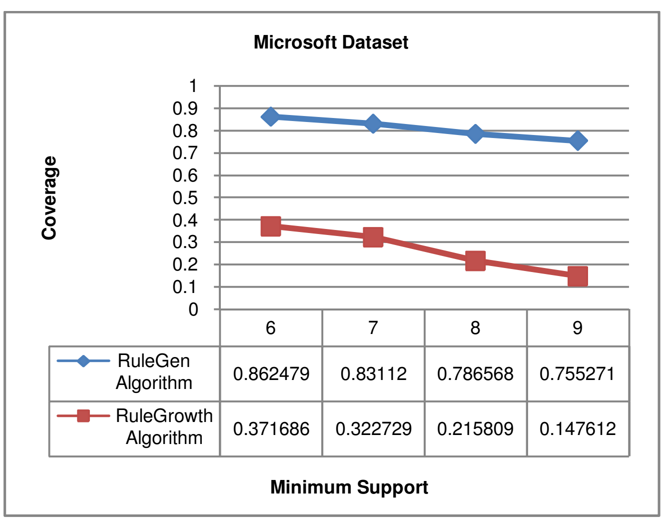
Fig. 6. Precision of the Recommendation System.



(a)



(b)



(c)

Fig. 7. Coverage of Recommendation System.

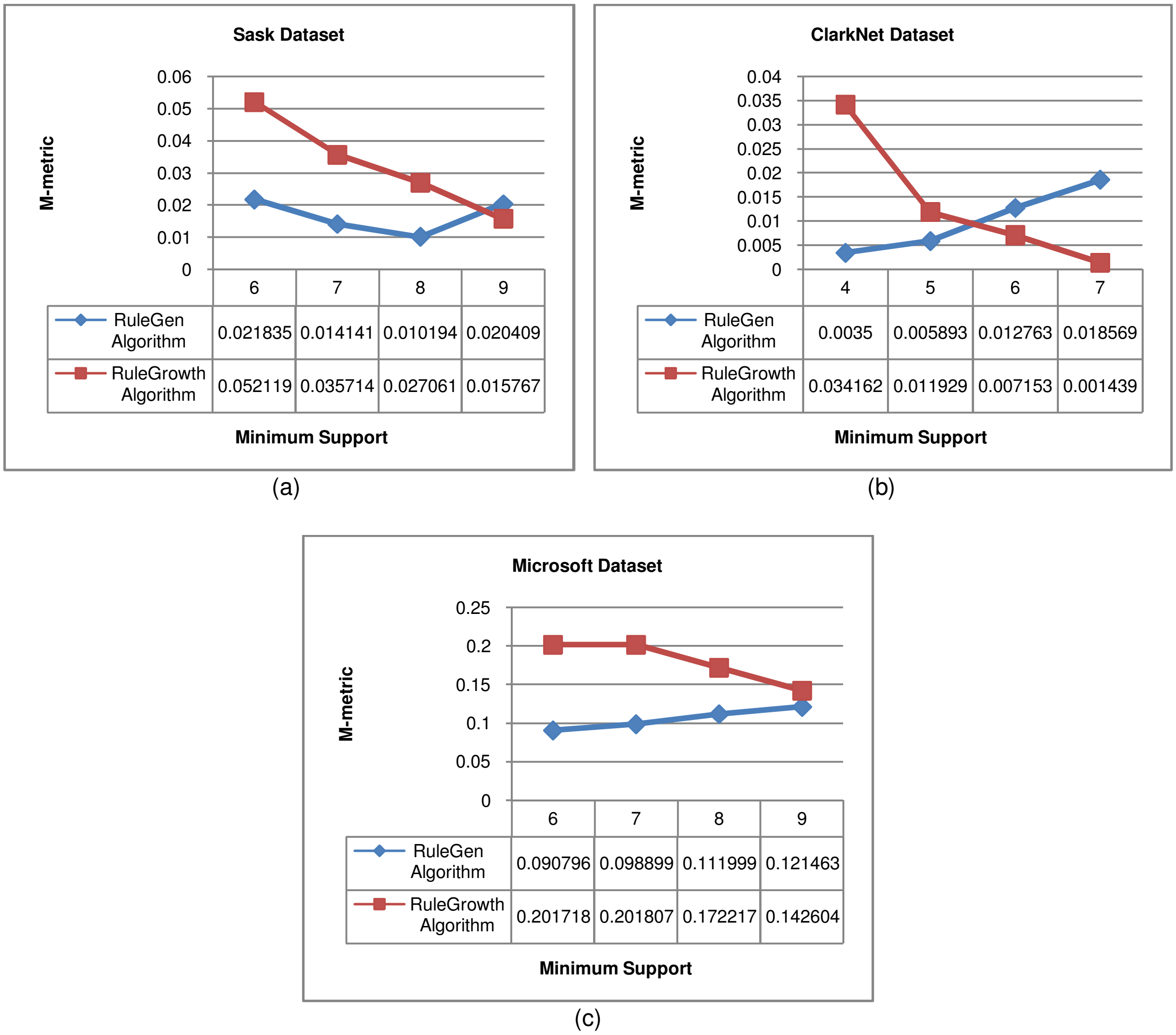


Fig. 8. M-metric of the Recommendation System.

V. CONCLUSION AND FUTURE WORK

A number of studies are being carried out on the web usage dataset using sequential rule mining resulting in web page recommendation systems for the web users. Firstly, the paper has been presented the taxonomy of sequential rule mining algorithms which classify the current algorithms into three major categories, namely, apriori based, pattern growth and early pruning. A comparison among different sequential rule mining algorithms belonging to three categories has also been presented in tabular form. It is observed that due to the removal of expensive candidate generation and test and the reduction of the number times of database scans, the performance of the pattern based mining algorithms is better than the apriori based algorithms.

Secondly, a framework for web page recommendation System based on sequential rule mining discovered from web usage data has also been presented. At last, the two pattern based algorithms, RuleGrowth and RuleGen are compared by applying them in the web page recommendation system using web access sequence datasets of three real world websites. The effectiveness of the web page recommendation system framework was evaluated in terms of three important factors, namely precision, coverage and m-metric of the generated recommendations. The performance comparison results show that RuleGrowth algorithm can be used to recommend web pages more effectively than RuleGen algorithm but with extra memory usage. This work can be enhanced further by integrating the semantic knowledge of the real website in the terms of domain ontology into all the phases of the presented web page recommendation system.

REFERENCES

[1]. Mobasher, B. (2007). Data Mining for Web Personalization. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) The Adaptive Web. *Lecture Notes in Computer Science*, 90-135. Springer, Berlin, Heidelberg.

[2]. Fournier-Viger, P., Lin, J. C. W., Kiran, R. U., Koh, Y. S., & Thomas, R. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition Ubiquitous Internationa*, *1*(1), 54-77.

[3]. Mabroukeh, N. R., & Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, *43*(1), 1-41.

[4]. Fournier-Viger, P., Nkambou, R. & Tseng, V. S. (2011). RuleGrowth: Mining Sequential Rules Common to Several Sequences by Pattern-Growth, *SAC'11,* TaiChung, Taiwan, 1-7.
[5]. Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine learning*, *42*(1-2), 31-60.
[6]. Nguyen, T. T. S., Lu, H. Y., Tran, T. P., & Lu, J. (2012). Investigation of sequential pattern mining techniques for web recommendation. *International Journal of Information and Decision Sciences*, *4*(4), 293-312.
[7]. Woon, Y. K., Ng, W. K. & Lim, E. P. (2005). Web Usage Mining: Algorithms and Results, *in A. Scime (Ed.): Web mining: Applications and Techniques*, 373–392, IGI Global, Brockport, USA.
[8]. Ezeife, C. I., & Liu, Y. (2009). Fast incremental mining of web sequential patterns with PLWAP tree. *Data mining and knowledge discovery*, *19*(3), 376-416.
[9]. Fournier-Viger, P., Gueniche, T., Zida, S., & Tseng, V. S. (2014). ERMiner: sequential rule mining using equivalence classes. In *International Symposium on Intelligent Data Analysis*, 108-119.
[10]. Fournier-Viger, P., Wu, C. W., Tseng, V. S., Cao, L., & Nkambou, R. (2015). Mining partially-ordered sequential rules common to multiple sequences. *IEEE Transactions on Knowledge and Data Engineering*, *27*(8), 2203-2216.
[11]. Fournier-Viger, P., Faghihi, U., Nkambou, R., & Nguifo, E. M. (2012). CMRules: Mining sequential rules common to several sequences. *Knowledge-Based Systems*, *25*(1), 63-76.
[12]. Fournier-Viger, P., & Tseng, V. S. (2011). Mining top-k sequential rules. In *International Conference on Advanced Data Mining and Applications*, 180-194.
[13]. Fournier-Viger, P., & Tseng, V. S. (2013). TNS: mining top-k non-redundant sequential rules. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 164-166.
[14]. Zida, S., Fournier-Viger, P., Wu, C. W., Lin, J. C. W., & Tseng, V. S. (2015). Efficient mining of high-utility sequential rules. In *International workshop on machine learning and data mining in pattern recognition*, 157-171. Springer, Cham.
[15]. Van, T. T., Vo, B., & Le, B. (2014). IMSR_PreTree: An improved algorithm for mining sequential rules based on the prefix-tree. *Vietnam Journal of Computer Science*, *1*(2), 97-105.
[16]. Van, T. T., Vo, B., & Le, B. (2011). Mining sequential rules based on prefix-tree. In *New Challenges for Intelligent Information and Database Systems*, 147-156. Springer, Berlin, Heidelberg.
[17]. Singh, S., & Aswal, M. S. (2016). Towards a framework for web page recommendation system based on semantic web usage mining: A case study. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, 329-334.
[18]. Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Knowledge and information systems*, *1*(1), 5-32.
[19]. SPMF: An Open-Source Data Mining Library. (2019) http://www.philippe-fournier-viger.com/spmf/index.php.
[20]. Traces Available in the Internet Traffic Archive. (2019) http://ita.ee.lbl.gov/html/traces.html.
[21]. Microsoft Anonymous web. (2019) http://kdd.ics.uci.edu/databases/msweb/msweb.html.
[22]. Nakagawa, M., & Mobasher, B. (2003). Impact of site characteristics on recommendation models based on association rules and sequential patterns *In: IJCAI 2003 Workshop on Intelligent Techniques for Web Personalization*, *3*, 1-10.

How to cite this article: Singh, Satyaveer and Aswal, Mahendra Singh (2020). Performance Analysis of Sequential Rule Mining Techniques for Web Page Recommendation System. *International Journal on Emerging Technologies*, *11*(2): 81–91.