

A Categorized Item Recommender System Coping with User Interest Changes

Su Sande Ko Ko and Rachsuda Jiamthapthaksin

Abstract—One significant characteristic of data in specific domain like movie challenges research in recommender systems that user preferences naturally changes over time. Traditional collaborative filtering (CF) method does not take in consideration sequences of customer's rating, which reflects changes of customer's preference over a period of time. This paper proposes a novel recommender system that overcomes the limitation of CF by combining collaborative filtering and sequential pattern mining with time interval which reflects user's preference changes over a period of time. Sequential patterns of categories of items are generated which represents and summarizes interest changes of users varied over time, and are used for revising recommended items produced by traditional CF. Experimental results show that the proposed system show improvements over the traditional collaborative filtering method.

Index Terms—Collaborative filtering, interest drift, sequential pattern mining, recommender system.

I. INTRODUCTION

Recommender system helps users to process huge data and find the information or items of their interest. It has been used in communities where items like movies, music, books, news, etc. are suggested to users. A recommender system is a system to help customers to find what they want easily and help them to discover products they like but are not looking for. Main goal of our recommender system is to suggest items in diverse categories which reflect changes.

In particular to movies domain, we observe characteristics that items has categories, rating and then user's interest change over time. The changes results in poor quality of traditional collaborative filtering recommendation algorithm that commonly used in this domain [1], [2].

User preferences usually drift in two ways: change occurs when a user either develops new interests or changes preference for a particular item or genre. This means that interest ratings on the same category will go up or down and this change will affect quality of systems. Most of research on interest changes has been solved by using time decay function with higher weights applied to new items [3], [4]. These approaches might be used based on rating history only but these magnify the problem of data sparsity because of using

current preference of user for producing recommendations. Keunho *et al.* presents the system by integrating sequential mining and collaborative filtering to improve the performance of systems [5]. Nevertheless, it is not able to capture interest changes in high level like categories.

This paper proposes recommender systems for movies that use a sequential pattern mining method and a collaborative filtering (CF) method. By combining these methods, build systems with a higher accuracy of predictions. The proposed methods are based on sequential pattern mining algorithm. The traditional collaborative filtering approach assumes that all given taste of user are equally important. So it can collect irrelevant preferences of user. This assumption ignores the fact that a user's interests may be changed over time. Then it can recommend irrelevant items in inappropriate time. Therefore, we propose a new collaborative filtering approach to take user interest evolution into account.

Our main contribution is that a novel recommender system for movie domain based on frequent sequential pattern mining with time interval. The proposed recommender system generates patterns of categories of items as offline frequent sequential process which will be used in online process to revise the recall items produced by tradition Collaborative Filtering. One of the benefits of category based pattern generation is that it can also be used to further summarize the current behaviors of users in the community, which is valuable for marketers who creating marketing strategic plans. The input of the sequential pattern mining phrase is the categories of movies that are interested by users. The rating history log data is to be preprocessed and then sequential access patterns are mined from it. Frequent sequential access patterns are discovered from the mined sequential access patterns. Using a current user access sequence pattern, the proposed system can be given Top n genre list from derived frequent sequential patterns. The proposed system finds Top n movies from nearest neighbors of target users that movies match with Top n genre list. Finally, the proposed recommender system recommends Top n list to a target user.

The rests of the paper is organized as follows: Section II provides related work to the proposed recommender system. Section III illustrates the proposed recommender system. The experimental results and performance evaluation of the proposed system are explained in Section IV. Section V concludes the proposed work.

II. RELATED RESEARCH

Recommender systems are employed to help users to find their items based on their preferences. They are categorized into 3 types: content-based recommendation; collaborative

Manuscript received March 22, 2014; revised May 25, 2014.

The authors are with the Computer Science Department, Faculty of Science and Technology, Assumption University, Bangkok, Thailand (e-mail: susande86@gmail.com, rachsuda@scitech.au.edu; tel.: +66 2719-1515 ext. 3681, 3682).

filtering recommendation (CF) and Knowledge-based Recommendation [6]. Among all types, CF-based is mostly used in movie domain. According to our observation, CF-based is applied with other data mining algorithms to improve prediction performance.

There are several data mining algorithms which are used together with CF-based filtering. These algorithms [7], such as Prefix Span, GSP, SPAM, consider only the item occurrence order but do not consider the item intervals between successive items. Keunho *et al.* implemented A hybrid online-product recommendation (HOPE) system, which integrates CF-based recommendation using implicit ratings and SPA-based recommendations [5]. This system first calculated the predicted preference of Sequential Pattern Analysis (SPA) and the predicted preference of collaborative filtering (CF) by using implicit ratings, and then computes a linear summation of these predicted preferences. The limitation of collaborative filtering can be avoided by integrating CF with SPA which reflects changes in user interest over time with sequential patterns. However this system cannot be carried out efficiently on large files and databases.

In addition, Wang and Li proposed the system that used the web access logs to sequential mining algorithm [8]. And then they stored the derived patterns by constructing of pattern-tree. By construction of pattern-tree, the system has to scan all sequential patterns at once. Pattern-tree can be used for matching the current user interest and generating recommendation according to a proposed generation recommendation rules algorithm. Since user access pattern changes with time, there will be a need to build pattern tree time by time in order to capture interest changes. However, neither of them [5], [8] is able to capture interest changes in high level like categories.

Beside utilizing sequential pattern mining to capture user interest change, a number of research works that addresses this issues by adding decay function in rating prediction in CF based. We can find user recent preference of the active user from a list of recently watched movies. At ranking RS, if the movies are not very new, we may have some "assumption" to reduce its interesting by some degree level (e.g. you may not be interested in watching 70's movies). User interest change on time partition that we have divided user's rating history into several periods, analyzed users interest distribution in these periods and quantizes every user's interest. At the same time, we can find user's recent interest by setting a time window. Qian-Qian and Ning proposed new collaborative filtering system that combines traditional collaborative filtering with time-based weighted on user interests and time-based weighted on items [3]. They presented new formula for $r_{ui}' = r_{ui} + f_u(t) + f_i(t)$ where $f_u(t)$ is an influence function that reflects user interests changing over time to the actual rating on item i by user u and $f_i(t)$ is an influence function that reflects the changes in item popularity over time.

Chen and Xu proposed the system that combined the User profile weight and time weight and applied to a personalized collaborative filtering recommendation system [4]. User profile weight is added to the computation of user similarity because users with same characteristics deal with same

interest. They considered age, gender and occupation as the factor. Time weight is added to the computation of prediction. They used a monotonic increasing time function. It neglected the early rating or interest.

III. PROPOSED RECOMMENDER SYSTEM

The sequential pattern mining-based recommendation method, which considers the process of customers' rating sequences, is proposed to overcome the restriction of traditional Collaborative Filtering methods. The changes of consumer's preferences reflect the sequence of rating in a period of time. However, we have been studied about mining the consumers' interest based on rating sequences and we proposed a new method that watches a consumer's rating sequence evolves over time for predicting consumers' interest changes. Sequential patterns are frequently rated items sequentially by users. The works of Keunho *et al.* [5] and Wang and Li [8] make use of sequential pattern mining for web personalization and online-product recommendation system. Unlike these approaches, our approach focuses on movie domain; specifically, suggesting movies in new genres that target users have not experienced before and thus it provides the diversity in movies recommended. Our proposed work also takes consideration on time factor that predicts current interest of the target user.

Our proposed framework comprises of 2 phases as shown in Fig. 1. In Offline Phase, derives frequent sequential patterns from history log (MovieLens Dataset). Firstly, Data Pre-processing Phase finds Top n movies genres per month per user. Secondly, the patterns are derived according to the time interval and then it summarizes patterns with higher support count stored in pattern table. In Online Phase, the system matches target user current access pattern with pattern in the pattern table. If the exact match is found, it extracts the list of genres and match with movies' genres those movies already rated by Top n users (Based on the similarity between Targets user and other users). Otherwise, it finds the most similar patterns consider overlapping with user's access pattern and then match with movies' genres that movies already rated by Top n users (Based on the similarity between Targets user and other users). Finally, Top n items are recommended to a target user. The important steps for generating recommendations to the user is as follows,

- Preprocess Data
- Finds frequent sequential patterns with time interval and Transform patterns with high support
- Constructs Pattern Table and Generates of Top n genres List with user current access sequences
- Generates recommendation movies list from nearest neighbors of Target users that movies' genres match with Top n genres list.

A. Preprocessing

We transform the original customer transaction database into a customer sequence table. Each sequence in the table consists of all the transactions of a user while each transaction consists of a set of movies' genres at time interval (Top n genres per user per month). This sequence table will be used as the data resource for our sequential pattern mining algorithms.

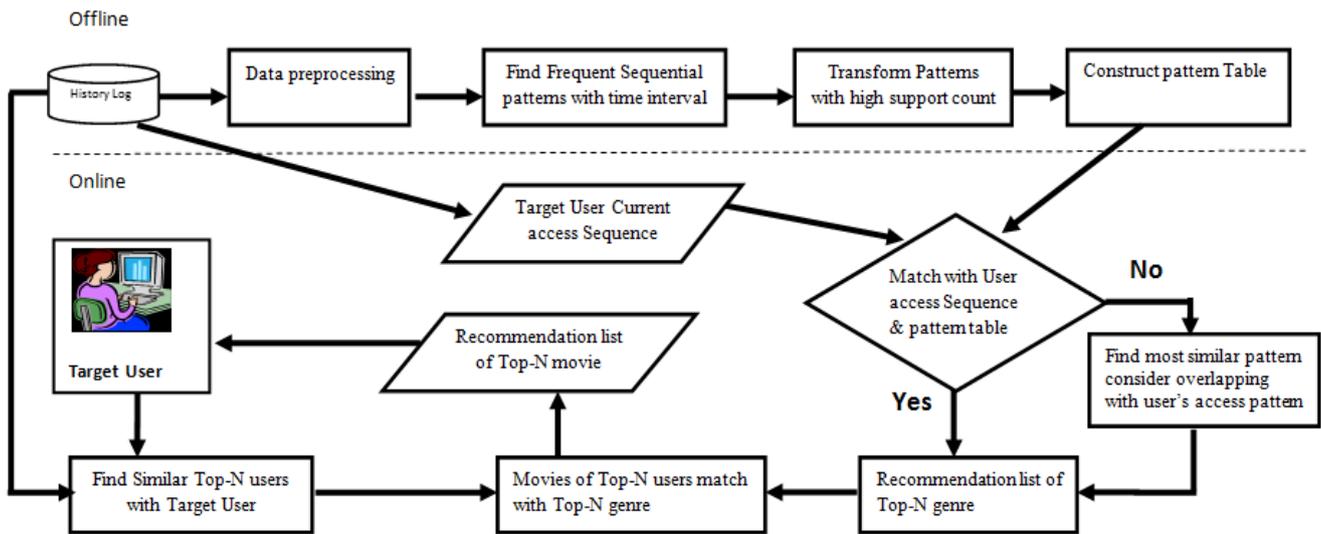


Fig. 1. Framework of proposed recommender system.

B. Finds Frequent Sequential Patterns with Time Interval and Transform Patterns with High Support

The proposed system applied sequential pattern mining with time interval algorithms [9], for mining the complete set of sequential access pattern from the database. The Hirate-Yamana presents an algorithm for discovering frequent sequential patterns respecting some time-constraints to filter uninteresting patterns [9]. Use of time constraints is very interesting because it can greatly reduce the number of patterns found and it is also faster. Further more it uses less memory than if all patterns are discovered. In fact, this algorithm is an extension of the Prefix Span algorithm. A time-extended sequence database is a set of time-extended sequences. A time-extended sequence is a list of item sets (groups of Top n genres per user per month). Each item set is annotated with a timestamp that is an integer value. Note that it is assumed that a genre should not appear more than once in an item set and that items in an item set are lexically ordered.

It specified four kinds of interval constraints are as follow:

C1 is a minimum item interval between any two adjacent items, C1 is defined as $\geq \text{min interval for all } \{i|1 < i < m - 1\}$.

C2 is a maximal item interval between any two adjacent items, C2 is defined as $\leq \text{max interval for all } \{i|1 < i < m - 1\}$.

C3 is a minimum item interval between the head and tail of the sequence, C3 is defined as $\geq \text{min whole interval}$.

C4 is the maximal item interval between the head and tail of the sequence, C4 is defined as $\leq \text{max whole interval}$.

Constraints C1, C2, C3, and C4 are optional input parameters. C1, C2 and C4 are Anti-monotone constraint and C3 is a monotone constraint. C1, C2 and C4 satisfy when a sequence a sequence does not satisfy the given constraints; any superset of sequence also does not satisfy the constraint. C2 satisfies when a sequence satisfies the constraint; any superset of A also satisfies the constraint. Algorithm finds all frequent time-extended sequences in a database ISDB that satisfies with minsup, C1, C2, C3 and C4. The notation ISDB| (t, i) represents the time-extended database with a pair (timestamp, item).

AlgoHirate

INPUT ISDB, min sup, C1, C2, C3, C4

OUTPUT R (=frequent sequences satisfying constraints)

R: = Scan ISDB and find all frequent items with support higher than minsup

For each frequent item I,

Add $(0, i)$ to R.

AlgoProjection (ISDB| $(0, i)$, R, minsup, C1, C2, C3, C4)

Return R;

AlgoProjection (ISDB|prefix, R, minsup, C1, C2, C3, C4)

INPUT (ISDB|prefix, R, minsup, C1, C2, C3, C4)

OUTPUT R

Scan ISDB|prefix to find all pairs of item and timestamp, denoted (t, i) satisfying minsup, c1 and c2.

For each pair (t, i) found

NewPrefix: = Concatenate (prefix, (t, i)).

IF newprefix satisfies C3 and C4

Add newPrefix to R.

If (size of ISDB|newprefix) \geq minsup

AlgoProjetion (ISDB| newprefix, R, minsup, C1, C2, C3, C4).

Return R.

TABLE I: SAMPLE SEQUENCE DATASET

SID	SEQUENCES
S1	$(0, 1), (1, 1\ 2\ 3), (2, 1\ 3)$
S2	$(0, 1) (1, 1\ 2), (2, 1\ 2\ 3), (3, 1\ 2\ 3)$
S3	$(0, 1\ 2), (1, 1\ 2)$
S4	$(0, 2), (1, 1\ 2\ 3)$

Dataset (Table I) is a time-extended sequence database. The dataset contains 4 sequences. Each sequence contains itemsets that are annotated with a timestamp. For example, consider the sequence S1. This sequence indicates that itemset $\{1\}$ appeared at time 0. It was followed by the itemset $\{1\ 2\ 3\}$ at time 1. Finally, itemset $\{1\ 2\}$ appeared at time 2.

From the dataset (Table I), if we run the algorithm with minsup= 55 %, min_time_interval = 0, max_time_interval = 2, min_whole_interval = 0, max_whole_interval = 2, we obtain the following results:

According to AlgoHirate algorithm, frequent actions {3}, {2 3}, {2}, {1 2 3}, {1 2}, {1 3} and {1} are found. (0, 3), (0, 2 3), (0, 2), (0, 1 2 3), (0, 1 2) and (0, 1) are added to R, the set of sequences found. Then the projected database ISDB|(0, 3), ISDB|(0, 2 3), ISDB|(0, 2), ISDB|(0, 1 2 3), ISDB|(0, 1 2), ISDB|(0, 1 3) and ISDB|(0, 1) are created. For each of these databases, the algorithm AlgoProjection is executed. AlgoProjection finds all frequent pairs (timestamp, item) that satisfy the constraints C1, C2 and minsup. For example, for ISDB|(0, 2), the frequent pair (1, 3), (1, 1 2) and so on are found. (1, 1 2 3) is as infrequent pair because it appears in S2 and S4 only that is not satisfying minsup. So these frequent pairs are concatenated to (0, 2) to obtain sequences <(0, 2), (1, 3)> and <(1, 3), (1, 1 2)> and so on. Because these sequences respect C3 and C4, they are added to the set R of sequences found. Then ISDB|<(0, 2), (1, 3)> and ISDB|<(1, 3), (1, 1 2)> are calculated. Because these are more than minsup sequences, AlgoProjection is executed again. The pattern S9 indicates that the item 2 was followed by item 1 and 2 one time unit after. This pattern has a support of 75 % because it appears at S2, S3 and S4 of four sequences in dataset. After completing of execution, there are 16 sequences patterns as shown in Table II.

TABLE II: SEQUENTIAL PATTERNS

ID	SEQUENTIAL PATTERN	SUPPORT	SID
S1	(0, 3)	75%	S1,S2,S4
S2	(0, 2 3)	75%	S1,S2,S4
S3	(0, 2)	100%	S1,S2,S3,S4
S4	(0, 1 2 3)	75%	S1,S2,S4
S5	(0, 1 2)	100%	S1,S2,S3,S4
S6	(0, 1 3)	75%	S1,S2,S4
S7	(0, 1)	100%	S1,S2,S3,S4
S8	(0, 2), (1, 3)	75%	S1,S2,S4
S9	(0, 2), (1, 1 2)	75%	S2,S3,S4
S10	(0, 2), (1, 1 3)	75%	S1,S2,S4
S11	(0, 2), (1, 1)	100%	S1,S2,S3,S4
S12	(0, 2), (1, 2)	75%	S2,S3,S4
S13	(0, 1), (1, 1 2)	75%	S1,S2,S3
S14	(0, 1), (1, 1)	75%	S1,S2,S3
S15	(0, 1), (1, 2)	75%	S1,S2,S3
S16	(0, 1 2), (1, 1)	75%	S1,S2,S3

C. Constructs Pattern Table and Generates Genre List with User Current Access Sequences

For each item in Time0, we maximize the support count of items that occur in Time1. For Example, The derived patterns S8, S9, S10, S11, S12 contain {2}_{time0} and followed by {1,2,3}_{time1}. We maximize the support count of each as follow: Sup (1) = Max {Sup (1), Sup (1, 2), Sup (1, 3)} = 100%, Sup (2) = Max {Sup (2), Sup (1, 2)} = 75%, Sup (3) = Max {Sup (3) Sup (1, 3)} =75%. Then, we construct pattern table according to the result of maximize patterns and order by maximum support of each *item_{time1}*. At This Step, we omitted the same patterns ({2}_{time1}, {2}_{time1}) because we want the different patterns between time0 and time1. After that, we matched target user current access pattern with the derived patterns *item_{time0}*. If the exact match is found, it extracts the

Top *n* (*item_{time1}*) (Top *n* genres List) for target user. Otherwise, it finds the most similar patterns consider overlapping with user's access pattern and order by percentage of overlapping patterns. Then we selected Top *n* genres.

D. Generates Recommendation Movies List

In this paper, we used traditional Pearson's correlation to calculate the similarity between target user and other users [6].

$$PC_{(u,v)} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (1)$$

where r_{ui} and r_{vi} denote the actual rating of the user *v* and user *v* on the item *i*, \bar{r}_u and \bar{r}_v denote the average rating of user *u* and user *v* for all the co-rated items and I_{uv} denote a set of items rated by both users. As the neighbors of target user, the group of Top *n* users is selected by the degree of similarity in descending order. The Top *n* neighbors are then used to predict the preference of user *u* on movie *i* that match its genres and Top *n* genre List for user *u* [6].

$$pred_{(u,i)} = r_u + \frac{\sum_{v \in N} PC_{(u,v)} * (r_{v,i} - \bar{r}_v)}{\sum_{v \in N} |PC_{(u,v)}|} \quad (2)$$

where \bar{r}_u and \bar{r}_v denote the average all ratings of the user *u* and user *v*, $r_{v,i}$ denote the actual rating of the user *v* on the item *i* and $PC_{(u,v)}$ denote the similarity of the user *u* and the user *v* and *N* is the set of the neighbors. After calculating prediction values of movies watched by neighbors of the target user, the Top *n* movies are recommended.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Design and Characteristic of Dataset

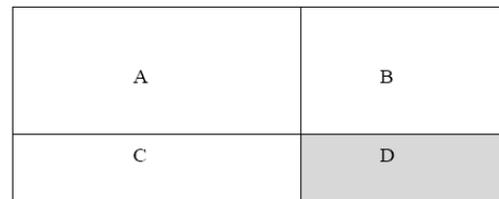


Fig. 2. Partition of dataset.

The recommendations were performed on the MovieLen dataset. This dataset currently contains 1,000,209 ratings (on 1-5 rating scale) from 6,040 users on 3,883 movies and 18 genres. The maximum number of genres per movie is 6 and minimum number is 1. Training dataset is used for designing the recommender system based on mined sequential patterns. Test dataset is used to test the proposed recommender system. We follow the experiment benchmark as specified in [5]. We

partitioned our dataset as shown in Fig. 2. We partitioned data into four parts: A, B, C, D. Part A and Part C are partitioned by random users. Part A is used for training dataset, and Part C is used for test dataset. Part A and Part C are used for calculating similarity between target user and other users and recommendation items in part D.

At first, we defined the characteristics of dataset for changing over time. We derived frequent sequential pattern with 0.2 minimum supports. We can classify the characteristics of derived patterns with overlapping values over first time interval and next time interval. We calculated the shifting interest percentages of categories from current time interval to next time interval according to (3), where a score reaches its best value at 0% that user totally changes her interest and worst score at 100%. We examined that only 9% of users aren't change their interest from time to time in period M1, and 0.2% of users totally shift their interest and 49% of users highly shift their interests. Most of users shift their interests at all time intervals, as shown in Fig. 3. In a nutshell, Movie Len dataset presents the users' preference for particular categories of movies changes over time so that it is suitable to be used for evaluating our proposed system.

The formula to calculate the interest change from time to time is:

$$S (\%) = 100 * (N_c * 2) / (N_f + N_n) \quad (3)$$

where N_c number of common Items , N_f total number of items in first time interval and N_n total number of items in next time interval.

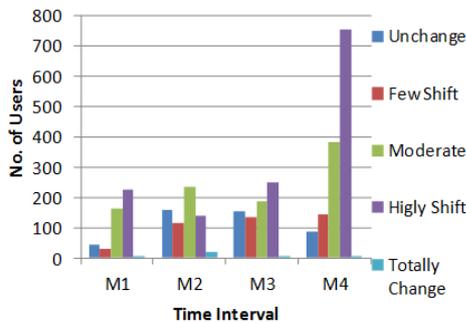


Fig. 3. The shifting interest of categories from current time interval to next time interval.

B. Evaluation Measures

We want to find out whether the user would be interested in rating or purchasing some or all the items included in that Top n list. For this purpose, we used Precision, Recall and F1 measures [10]. We can now define recall, precision and F1 for Top n recommender systems in the following way:

Recall, when referring to Recommender Systems, can be defined as the ratio of the number of relevant records retrieved over the total number of relevant records:

$$\text{Recall} = \frac{tp}{tp + fn} \quad (4)$$

Precision, when referring to Recommender Systems, can be

defined as the ratio of the number of relevant records retrieved over the total number of irrelevant and relevant records retrieved.

$$\text{Precision} = \frac{tp}{tp + fp} \quad (5)$$

where tp is the number of relevant records retrieved, fn is the number of relevant records not retrieved and fp is the number of irrelevant records retrieved.

F1 is the harmonic mean of precision and recall. This score combines Precision and Recall into one measure that can be used to evaluate the overall performance of the classifier.

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

C. Experimental Results and Implications

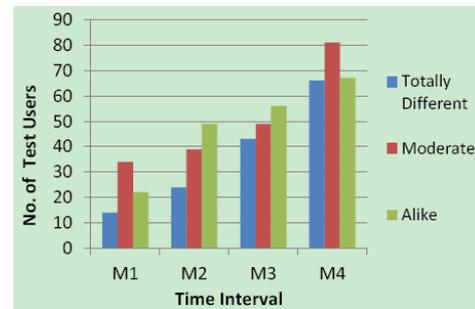


Fig. 4. Similarity Between our recommended categories to actual categories (ground truth).

Our proposed system and traditional collaborative filtering system were executed with the number of neighbors set to 1, 2, 3, 4, or 5 and with the number of recommended items set to 10, 20, 30, 40 or 50. We propose an approach towards categorizing user interests in a recommender system by using sequential pattern mining. The algorithm requires very little resource in terms of both space and time than other sequential pattern mining recommendation system [5]. To evaluate the recommendation accuracy of categories for our proposed system, we did another calculation of the percentages of similarity between our recommended categories to actual categories of user by using (3). In this measure, where a similarity score reaches its best value at 100% that made the best predictions for user and worst score at 0%. The recommendation of categories generated by our system is treated as related to the corresponding categories by the user. In Fig. 4 show that only 14% of user did not match with actual data and in their recommendations of categories and 62% of user almost match with their recommendation of categories at M1 interval. The above result shows that the our recommendation system adapt to the user changing interest in a very effective manner.

The best accuracy of our system was obtained when the number of recommended items is 10 and when the number of neighbors is 5. It was performed by calculating similarity between target user and other users whose have rated the common movies were greater than 20 in the MovieLens dataset. The results as shown in Fig. 5, Fig. 6 and Fig. 7 depict the efficiency of precision, recall and f1 parameters. Our

proposed system outperformed of CF-Based recommender system in all measures of precision, recall, and F1, as shown as in Fig. 7.

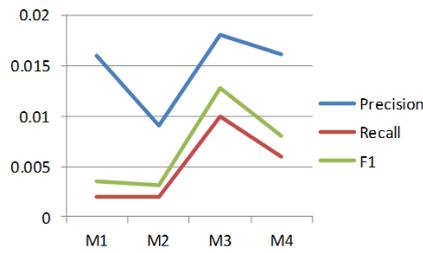


Fig. 5. Precision, Recall, and F1 of CF-Based recommender system.

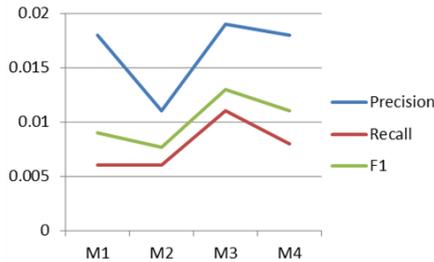


Fig. 6. Precision, recall, and F1 of our proposed recommender system.

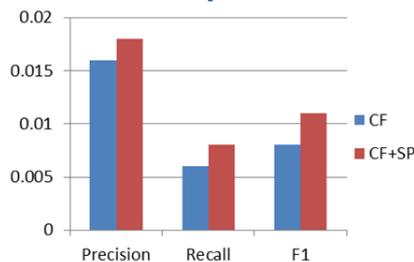


Fig. 7. Comparison of precision, recall, and F1 among our system and CF-based recommender system.

V. CONCLUSION

This paper proposed a novel movie recommender system that (i) considers the changes of user interest by using sequential pattern mining with time interval and integrate with traditional collaborative filtering and (ii) uses categories of movie, which is able to detect a user's interest change and utilizes to improve quality of recommendations. By product, the algorithm generates category based patterns, which can be used by marketers to analyze current behaviors of users in the community, which is valuable for marketers who creating marketing strategic plans. Moreover, this work presents an analysis of the Movie Len dataset to demonstrate a significant characteristic that the interest changes of users occurred over period of times. Additionally, our experimental results show improvement of proposed system in term of Precision, Recall and F1. The current system applies this system in the domain of movie recommender system which could further be applied

to in other domains, such as products and books and news, where interest of users change over time.

REFERENCES

- [1] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," *Recommender Systems Handbook*, Springer, Berlin, pp. 107-114, 2011.
- [2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation," in *Proc. the 10th international conference on World Wide Web*, ACM, 2001.
- [3] Q.-Q. Zhang and N. Ye, "Collaborative filtering algorithm adapting to changes over dynamic time," *International Proceedings of Computer Science & Information Technology*, Feb. 2012.
- [4] D. Chen and D. Xu, "A collaborative filtering recommendation based on user profile weight and time weight," in *Proc. International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, 2009.
- [5] K. Choi, D. Yoo, G. Kim, and Y. Suh, "A hybrid online-product recommendation system: combining implicit rating-based collaborative filtering and sequential pattern analysis," *Electronic Commerce Research and Applications*, Feb. 2012.
- [6] *Recommender Systems Handbook*, Springer, Berlin, 2011.
- [7] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: A survey," technical report, CAIS, Nanyang Technological University, Singapore, 2003.
- [8] X.-G. Wang and Y. Li, "Web mining based on user access patterns for web personalization," in *Proc. ISECS International Colloquium on Computing, Communication, Control, and Management*, vol. 1, pp. 194-197, 2009.
- [9] G. Y. Hirate and H. Yamana, "Generalized sequential pattern mining with item intervals," *Journal of Computers* vol. 1, no. 3, pp 51-60, June 2006.
- [10] G. Shani and A. Gunawardana, "Evaluating recommendation systems," *Recommender Systems handbook*, Springer, Berlin, pp 735-767, 2011.



Su Sande Ko Ko received her bachelor degree in computer science from University of Computer Studied, Yangon, Myanmar, in 2005. She is currently pursuing her master degree in Computer Science Department, Faculty of Science and Technology, Assumption University, Bangkok, Thailand. Her research interests are artificial intelligent, data mining and database management

and web applications.



Rachsuda Jiamthapthaksin received her Ph.D. in 2009 from the University of Houston, Texas, United States. She obtained a bachelor degree in computer science in 1997 and graduated with Honors in master degree in computer science, Dean's Prize for Outstanding Performance, in 1999 from Assumption University, Bangkok, Thailand. She is currently a lecturer in the

Department of Computer Science and also a researcher of the Intelligent Systems Laboratory at the Assumption University. Her research interests include intelligent agents, fuzzy systems, cluster analysis, data mining and knowledge discovery. She has served as a reviewer for many conferences and served as a volunteer staff in an organization of the 2005 IEEE ICDM Conference, November 2005, Houston, Texas. She also served as a session chair for the 2009 Symposium on Computational Intelligence and Data Mining.