Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

The Enhancement of Markov Chain Recommendation System (MCRS) Using Items' Popularity

[1]Ahmed Adam Ahmed, [2]Naomie Salim

[1]College of Computer Science and Information Technology, Sudan University of Science and Technology
[2]Faculty of Computing, University Technology Malaysia

Abstract: **Web applications use recommendation techniques, which are based on users' preferences for items, to recommend interesting items to the active user. Many researchers study ways of using Morkov model in recommendation systems; however, they do not consider the time factor in their techniques, while users' opinions and items popularities vary with time. In our previous paper, we propose a new recommendation system that is based on Morkov model, Markov Chain Recommendation System (MCRS), and it considers the time factor. In this paper, we enhance MCRS using the general weights of items, and extra enhancement is done using period weights of items. The new enhanced techniques are compared with MCRS for the evaluation. The experiments are conducted using MovieLens dataset for the evaluations that use the precision and recall, and the recommendation accuracy to compare between MCRS and its enhancements. The result illustrates that the enhancements techniques outperform MCRS.**

Keywords: **Markov chain, Collaborative filtering, Recommender Systems, Items popularity, Items weights**.

I. INTRODUCTION

Nowadays, the internet is widely used by many websites to provide their users with many services and items[1]. These websites are widely visited by many users to access millions of items e.g. when users open a website then they can view a movie, or mark a photo as like[2]. It has become an easy task to produce and upload new items to these websites, but it is very difficult to retrieve the actually needed items. Websites as well as their users face the problem of information overload [3]. For example, if the user U of the website W views a list of movies then what is the next subset of movies he will view out of the other millions? Recommendation systems (RSs) are software that can be used to address this problem; they have been used by many websites to recommend items to their users [4]. Those RSs are based on items' descriptions and contents, and users' opinions and their preferences for items. According to these information about users and items, RSs techniques are, therefore, divided into several categories e.g. Content-based CBRS, Collaborative filtering CFRS, Knowledge-base KBRS, Hybrid RSs, and etc. [5] [6].

Markov proposed that the outcome of a given experiment can affect on the outcome of the next experiment[7], [8]. This type of processes is called Markov chain. Markov chain contains states, the process function for moving from one state to another, and the starting state(s) [9]–[12]. If we have a set of states $S=\{s_1,s_2,s_3,...,s_n\}$ then the process starts from one of these states and moves successively to another, and each move is called a step. If the chain is currently in state s_i then it can move to state s_j in the next step with a probability that is denoted by p(i,j), and this probability does not depend on which states the chain was in before the current state. All probabilities are called transition probabilities, and the process can remain in the same state i with the probability p(i,i). Initial probabilities are given to the starting states; this is usually done by specifying particular states as starting states.

Markov chain model has been used in recommendations system. Shani et al. [13] proposed an MDP-Based Recommender System (Markov Decision Process). Their technique is based on frequents sequences of items that are followed by an item. They consider sequences of k items; however, many users can access less than k items and others can access more. The main drawback of MDP-based recommendation systems is the limitation of using k sequence of items while users can access more or less than k items. This limitation leads to an inaccurate result because they use small sequence of items. They exactly use k=5, and the system is become inapplicable if they take a big sequence of items. Rendle et al. [14] present a method that brings together both matrix factorization (MF) and Markov chains (MC). Their method is based on personalized transition graphs over underlying Markov chains. In their solution, any user has its own transition matrix, and all users have a transition cube to contain all transition matrixes that are generated for individual users. Empirically, the FPMC model outperforms both the matrix factorization and Markov chain MC model. However, this technique faces the problem of sparsity because any user accesses only tens of items out of millions, and users' transition matrixes contains only limited number of items.

The rest of the research introduces the basic Markov Chain Recommendation system in Section II. We discuss the limitation of MCRS Section III, and the motivation of its enhancement is given in Section IV. The general weights MCRS and the period weights MCRS are illustrated in Sections V and VI. The experimental design and result are given in Section VII. Then, we conclude the research in Section VIII.

II. THE BASIC MARKOV CHAIN RECOMMENDATION SYSTEM (MCRS)

In our previous paper, we propose a new technique which is entitled Markov Chain Recommendation System (MCRS). The main components of MCRS are the initial vector and the transition matrix. To generate the initial vector, we need to understand the active user's vector.

The active's user:

The active user's vector is the target of the recommendation system. It represents all items that have been accessed by all users, which can be divided into two subsets. The first subset contains items that are accessed by the active user which can be used to recommend items from the other subset that are not accessed by him. Consider, the first subset A contains s items.

$$A=\{i_j=1 : j=1,2,3 \ldots s \text{ for } 1<=s<=n\} \qquad (1)$$

Here s is the number of accessed items by the active user and n is the number of all items. Then, the second subset B contains (n-s) items $B=\{i_z = 0: z=1,2,3, ..., (n-s)\}$. The active user vector is $V=(A)\cup(B)$. Normally, items are distributed and items that are accessed by the active user are not put together.

Example:

$$V=\{1,1,0,0,1,0,0,0,0,0,1,1,0,0,1,0,1,0,0,0\} \qquad (2)$$

In the above example, equation 2, n=20 and s=7.

The subset A contains seven items that are accessed by the active user, and the positions of accessed item in V are signed by 1. They can be used to recommend items from the subset B that contains thirteen items which do not accessed by the active user, and their positions in V are signed by 0. The active user's vector can be represented as in the equation 3:

$$V= \{i_e: e=1,2,3,\ldots,n \} \qquad (3)$$

Here, n is the number of all items, and i_e =1 if the active user has accessed $item_{e,}$ otherwise i_e =0.

In general, users have accessed n items in m session or period of time. Any user has accessed a limited subset of items from all items, and if he has accessed an item then that item is signed to one otherwise it signed to zero as represented in Table 1.

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

Table 1: the value of a(j,i) =1 if a user has accessed item$_i$ in session$_j$ otherwise a(j,i)=0, i=1,2,3,..,n and j=1,2,3,..,m.

	item$_1$	item$_2$	item$_3$	. . .	item$_n$
session$_1$	$a_{(1,1)}$	$a_{(1,2)}$	$a_{(1,3)}$	. . .	$a_{(1,n)}$
session$_2$	$a_{(2,1)}$	$a_{(2,2)}$	$a_{(2,3)}$	. . .	$a_{(2,n)}$
session$_3$	$a_{(3,1)}$	$a_{(3,2)}$	$a_{(3,3)}$	. . .	$a_{(3,n)}$
...	...	...	...	. . .	...
session$_m$	$a_{(m1)}$	$a_{(m,2)}$	$a_{(m,3)}$	. . .	$a_{(m,n)}$

The initial vector:

The initial vector 'I', of Markov chain, equals to the active user's vector V divided by the number of times of accessing all items by the active user i.e. the active user's vector V divided by the summation of elements of the active user's vector sum(V).

$$I=\frac{V}{\text{(The number of times of accessing all items by the active user)}}=\frac{V}{\text{sum}(V)} \qquad (4).$$

'I' is the initial vector that represents the probabilities of accessing items by the active user at the moment of the recommendation process.

The transition matrix of MCRS

Markov Chain transition matrix $T_{(n,n)}$ contains n rows and n columns; any row represents an item, and it has n columns correspinding to the n items. Here, n is the number of all items as represented in question 5 and Table 2.

$$T_{(i,j)}=p_{(i,j)}=\frac{\sum_{i=1}^{n}(\text{row}_{(i,j)} \text{ where the column of item}_{(i)}=1)}{\sum_{j=1}^{n}\sum_{i=1}^{n}(\text{row}_{(i,j)} \text{ where the column of item}_{(i)}=1)} \qquad (5)$$

Table 2 Markov Chain Transition Matrix

	item$_1$	item$_2$	item$_3$	. . .	item$_n$
item$_1$	$p_{(1,1)}$	$p_{(1,2)}$	$p_{(1,3)}$	. . .	$p_{(1,n)}$
item$_2$	$p_{(2,1)}$	$p_{(2,2)}$	$p_{(2,3)}$	. . .	$p_{(2,n)}$
item$_3$	$p_{(3,1)}$	$p_{(3,2)}$	$p_{(3,3)}$	. . .	$p_{(3,n)}$
...	...	...	...	. . .	...
item$_n$	$p_{(n,1)}$	$p_{(n,2)}$	$p_{(n,3)}$	. . .	$p_{(n,n)}$

Any row in $T_{(n,n)}$ represents an item and items that has been accessed with it by users in the same period of time. The row $p_{(i,j)}$ in $T_{(n,n)}$ is the row of item$_i$ in which i = 1,2,3,...,n; j=1,2,3,..., n; and n is the number of items that have neen accessed with item$_i$ i.e. any item has row and column. The value $p_{(i,j)}$ is the probability of accessing item$_j$ with item$_i$ in the same period of time. It can be calculated by retrieving rows that have the value 1 in the column of the item in Table 1. The probability vector of that item is the summation of the retrieved rows divided by the summation of these rows cells, see equation 5 above. This vector gives the row of item$_i$ in the transition matrix.

The basic MCRS is the vector product of (the initial vector) I and (the transition matrix) $T_{(i,j)}$.

$$R=I*T_{(i,j)} \qquad (6)$$

The result is given in the equation (6); the vector R contains the probabilities of accessing items by the active user. We sort these probabilities descending then items of the highest probabilities are recommended to the active user.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

III. THE LIMITATION OF THE BASIC MCRS

The basic MCRS technique outperforms the conventional Collaborative Filtering recommendation systems. However, more enhancements can be done using the time factor because items popularities vary with time. Items normally can be divided into three classes with respect to their popularities:

- First class: items that their popularities increase with time.
- Second class: items that their popularities decrease with time.
- Last class: items that are not affected either positively or negatively time.

According to these three classes, the basic MCRS has two limitations:

- The technique can recommend items that are not popular at the moment of the recommendation because some items are popular in the long term, but they are become not popular at the last period of time. These items may be popular in general, but they are not popular in the last period of time
- The technique can excludes some items from the recommendation list because some items are not popular in the long term, but they are popular in the short term, and users are interesting of them i.e. these items may not be popular in general, but they are popular in the last period of time.

These two limitations can violate the accuracy of the recommendation. To recommend the actually needed items to the active user, the basic MCRS can be enhanced using the time factor.

IV. THE MOTIVATION OF MCRS ENHANCEMENT USING THE TIME FACTOR

Markov Chain Recommendation System (MCRS) is based on users' preferences for items. From users' preferences for items, we can generate items popularities. As more users access an item its popularity increase and vice versa. There are two factors that can increase items popularities:

Items live time:

Items live time is the time interval in that users have been accessing these items. Some items are submitted in the web earlier; thus, their live times are long. The recent submitted items have short live time. Items that have long live time might be more accessed by users.

The interesting items to users:

Users rate for the interesting items, even if they have short live time.

MCRS can be used to recommend items to users, and the recommendation result can be one of these situations, according to the mentioned factors:

- The Basic MCRS can be used to generate a list of interesting items by users in general, using the active user's accessed items. However, many items can be interesting to users in general, but they are not recommended to the active user. Because, the recommendations are based on the initial vector. In this case some interesting items by users are not recommended by the system. Therefore, this limitation can be solved if the result is weighted using the general weights of items before the generation of the recommendation list. More details are given in Section V.
- The basic MCRS can be used to generate a list of the most rated items by users, but some of these items are not interesting in the last period of time. They are popular because they are rated in the long term. We can solve this problem if the result is weighted using the period weights of items before the generation of the recommendation list. More details are given in Section VI.

V. THE GENERAL WEIGHTS MCRS

Users of web applications are faced by the challenge of retrieving the actually interesting items. The challenge comes from the information overload problem. The number of available items (e.g. movies) that can be accessed (e.g. viewed) is very big. Users cannot retrieve all items to identify the interesting ones. Recommendation systems work on behalf of users to generate suggestions for items that might be of interest to users. MCRS recommends items to users according to the list items that have been accessed by the active user. However, items popularities vary with time. As more users access an item its popularity increases and vice versa.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

Consider a web site provides movies (A,B,C) to users. For new users, the question is which movie is suitable to be viewed first? The popularities of items identify the best choice of items to be viewed first. The most viewed items by users are the most popular. Our target is how to calculate items popularly?

For example:

If ten users have accessed three movies A,B and C as represented in Figure 1**Error! Reference source not found.**; then, what is the most popular one?

The answer:

We have ten users. This mean any item has ten chances to be viewed. The most accessed items by the ten users are the most popular.

- Movie A viewed 7 times out of 10.
- Movie B viewed 3 times out of 10.
- Movie C viewed 6 times out of 10.

The most popular movie are A and C respectively. The general weights of items is the vector W, that contains weights of itemj (j=1,2,3,...,n). The vector W can be calculated using Table 1 and the following equation:

$$W=\{W_j : W_j = \frac{(\text{Count of all users}'\text{sessions that contain item}_j)}{(\text{A count of all users}'\text{sesions })} \text{ and } j = 1,2,3,\ldots,n\} \qquad (7)$$

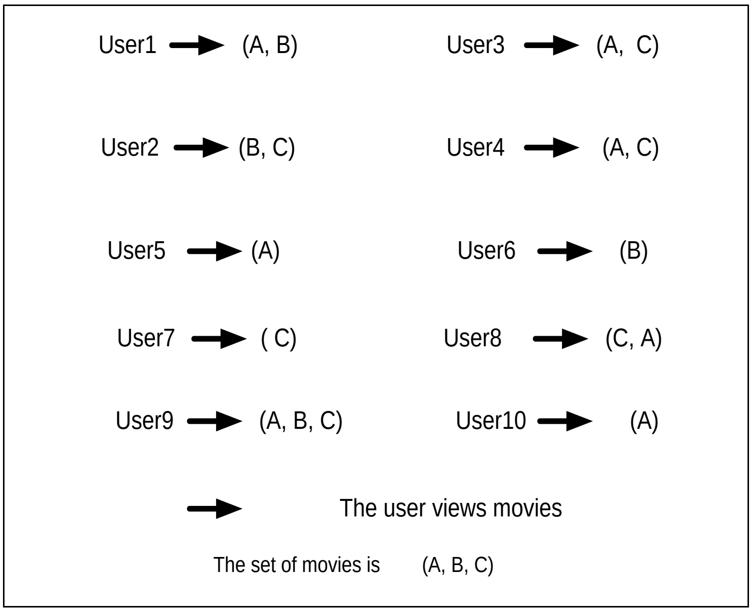

Figure 1 the most accessed items by all users

The general weights are the summation of (items' columns) divided by the number of all records. The general weights do not consider the time factor which will be discussed later in weights of items in a period of time, Section VI. The weight of item depends on users' preferences on it. When more users access an item its weight increases and vice versa. The general weights of items can be used to enhance the basic MCRS. Before recommending a list of items to the active users, the basic MCRS can be weighted by the general weight of items. The weighted MCRS is compared with the basic MCRS for the evaluation.

The general weights MCRS is given in equation 8.

$$\text{G-MCRS}= I^*T_{(i,j)} .W \qquad (8)$$

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

VI. THE PERIOD WEIGHTS MCRS

Users view any movie at specific point of time. For simplicity, the time interval, in which users have viewed movies, is divided into periods. If ten users have viewed three movies A,B and C (Figure 2); then, what is the most popular one in the last period of time?

The popularities of movies in the last period of time are different from the general popularities of movies.

- In general, as represented in Figure 1:
  - Movie A viewed 7 times out of 10.
  - Movie B viewed 3 times out of 10.
  - Movie C viewed 6 times out of 10.
  - The most popular movie are A and C respectively.
- In the last period of time p3, as represented in Figure 1:
  - Movie A viewed 3 times out of 4.
  - Movie B viewed 1 times out of 4.
  - Movie C viewed 3 times out of 4.

The most popular movies are A and C; all have the same popularities.

In general, A is most popular than C. In the last period of time the popularity of C is increased, and it become equal to the popularity of A.

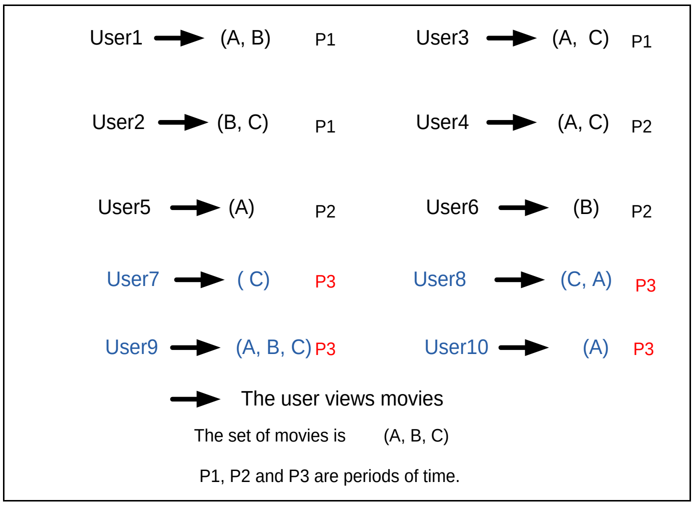

Figure 2 the most viewed movies by all users per periods of time

MCRS can use users' preferences on items in specific time interval P which can be divided into equal periods of time p (e.g. days, weeks or months).

The weights of items in period p_t is the vector w_{p_t} where:

$$w_{p_t}=\{w_{jt} : w_{jt} = \frac{(\text{count of all users' sessions that contains item}_j \text{ in period } p_t)}{(\text{count of all users' sessions in period } p_t)} \text{ and } j = 1,2,3,\ldots,n\} \qquad (9)$$

The weights of items vary with time. More users can access an item in the old periods of time; the same item can be accessed by less number of users in the recent periods and vice versa. For more enhancements, weight of items in the last period of time can be used to weight the result of the basic MCRS. Procedure one (Figure 3) can be used to calculate the period weights of items.

Procedure one:

 # Users-items table (Table 1) is used to generate the period weights of items).

"Items" :the list of all items in the given dataset.

"Periods" :the number of all periods of the time.

Period-eights = null;

for all records of all users.

If the period is (the last period of time in the data)

period-weights = period-weights + (fields of Items in the current record);

end if

end for

The period Weights of items= period-weights /(the number of all records)

Figure 3 the creation of the period weights of items.

The period weights of items give the popularities of items in the last period of time. We need to identify the probability of accessing all items in the last period. Thus, the vector of period weights of items can be considered as the period initial vector in the last period.

There are three cases:

- The basic MCRS result.
- The probability of accessing items in general.
- The probability of accessing items in the last period of time.

The period weights MCRS is: (The basic MCRS result) and (the probability of accessing items in general) and (the probability of accessing items in the last period of time)

$$\text{P-MCRS}= (I*T_{(i,j)}).W.(w_{p_t} * T_{(i,j)})$$

The period weights MCRS is compared with the basic MCRS and The general weights MCRS for the evaluation.

VII. EXPERIMENTAL DESIGN AND DESIGN

A. Experimental design:

We use MovieLens dataset to conduct the experiments for the evaluation of the general weights MCRS and the period weights MCRS models. To do that, we divide the time interval of the data into months, and we consider any two months as one period. The starting month is p out of 137. The initial value of p= 70. In the first experiment the training data starts from at the first month and end at the 70th month. The next two months are used for testing. The last two months in the training data are used to generate the period weights of items. All the training data is used to generate the general weights of items. Then, for the next experiment we increment p by 3. The number of all experiments is twelve.

The basis MCRS technique and the enhanced techniques using the general weights and the period weights of items can be used to recommend items to an active user. The active user accesses only small subset (A) of items. The subset (A) can be used in the recommendation processes. The rest of items that not accessed by the active user can be considered as B.

To evaluate the general weight MCRS and the period weights MCRS, we can use the set A that accessed by the active user to recommend items from the set B, which is hidden, to the active user. The training data can be used to recommend items to the active user as follows:

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

- The set A can be used by the basic MCRS to generate "MCRS result".
- It can be used by the general weight MCRS to generate "G-MCRS result".
- It can be used by the period weight MCRS to generate "P-MCRS result".

The testing data can be used to retrieve the actually accessed items as follows:

- The set A is used to retrieve "The actually accessed items" from the testing data i.e. we retrieve all records that contains any item accessed by the active user. Then, we can calculate the accessibility of items by the summation of all retrieved records. Then, we can normalize such that the summation of the accessibility of all items equal to one.

The evaluation can be done using the accuracy. It can be done using the mean average precision. In these cases, the k highest probability items can be taken from the actually accessed items and the different results. The best results have the highest accuracy and mean average precision.

B. Experimental results:

The basic MCRS technique is enhanced twice. First, the general weights MCRS. In this case the general weights of items are calculated from the training data in the twelve tests. These weights are used to enhance the result of MCRS. The enhancement result is the scalar product of the basic MCRS result and the general weights of items. Second, the enhancement is done using the period weights of items. The period weights of items are calculated from the last two periods in the training data in all of the twelve tests. The periods weights MCRS is the scalar product of the periods weights of items and the result of the basic MCRS.

The evaluation of the enhanced MCRS's is done using the accuracy and the mean average precision. Twelve user's accessed items are used in the twelve tests. The same user, the same training data, and the same testing data are used by the basic MCRS and the enhanced techniques. The general weights of items are calculated in the twelve tests using the training data; and the periods weights of items are calculated using the last two periods of the training data in any test.

The mean average precision:

The mean average precision of the general weights MCRS is 0.868223287; its better than the basic MCRS by 0.015476038. This means the general weights MCRS outperforms the basic MCRS. The mean average precision of the period weights MCRS is 0.873066328 it is better than the basic MCRS by 0.004843041 and better than the general weights MCRS by 0.020319079. This means the period weights MCRS outperforms the basic MCRS and the general weights MCRS using the mean average precision (Figure 4).

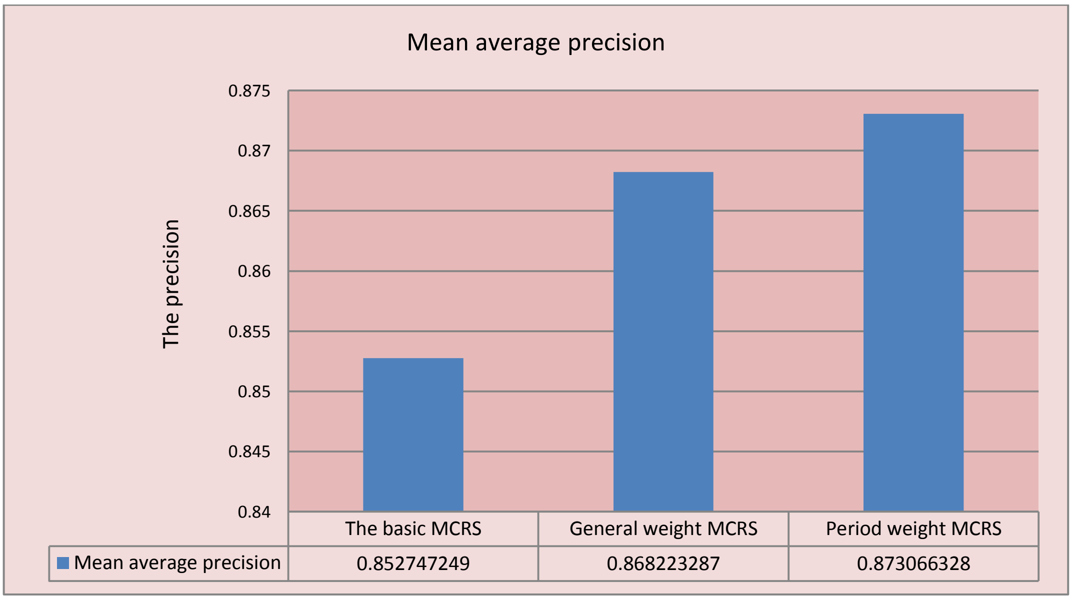

Figure 4 Mean average precision of the basic MCRS VS the general weights MCRS and the period weights MCRS.

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: www.noveltyjournals.com

The average accuracy:

The average accuracy of the general weights MCRS is 0.834666667; it's greater than the basic MCRS accuracy by 0.017. This means the general weights MCRS outperforms the basic MCRS. The accuracy of the period weights MCRS is 0.839666667 it is greater than the basic MCRS by 0.005 and greater than the general weights MCRS by 0.022. This means the period weights MCRS outperforms the basic MCRS and the general weights MCRS using the accuracy as showen in Figure 5 .

Recommendation systems have been used by many websites to ease the selection of the next actually needed items to their users. The basic MCRS can be used to recommend items to users. It is enhanced twice. First, it is enhanced using the general weights of items which calculated using the training data. Second, MCRS is enhanced using the period weights of items that calculated from the last two periods in the training data. The evaluation of MCRS is done using the average precision, and the accuracy. Firstly, we prove that the general weight MCRS outperforms the basic MCRS. Secondly, we found that the period weights MCRS is better than the basic MCRS, and the general weights MCRS.

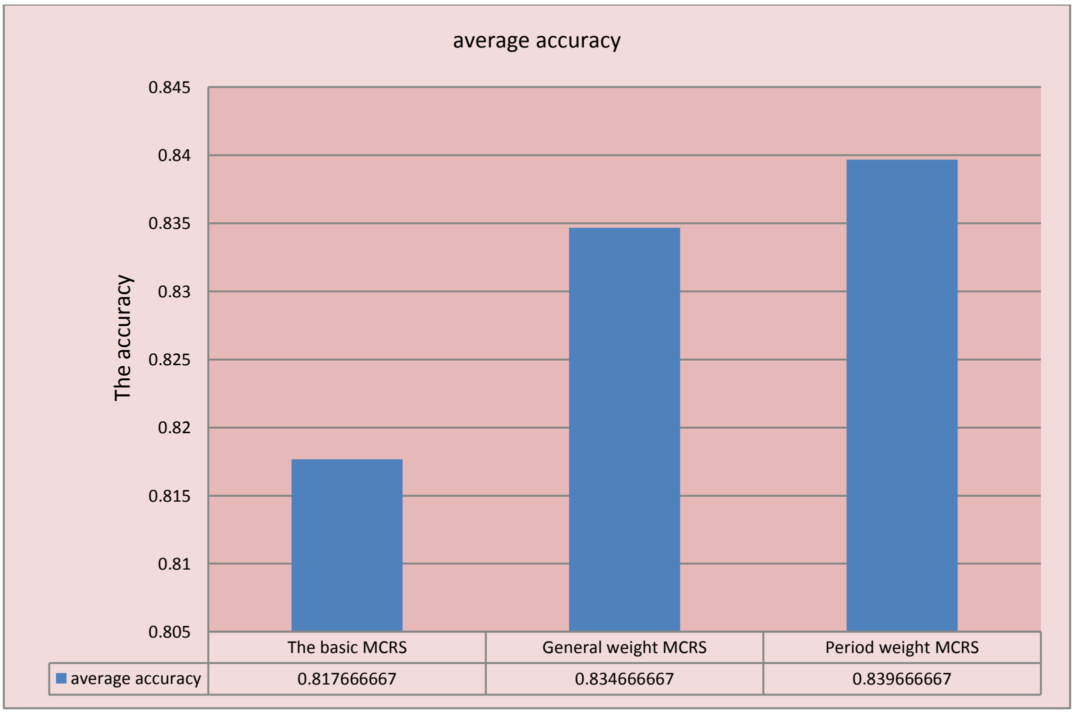

Figure 5 Accuracy of the basic MCRS VS the general weights MCRS and the period weights MCRS

VIII. CONCLUSION

The basic MCRS is based on users' preferences for items to recommend items to users. We find some limitations violate the accuracy of the recommendation using MCRS. These limitations are caused by the variation of users' opinions and items popularities with time. The time factor is used to enhance the basic MCRS. We use the time of users' preferences for items to calculate items popularities in general. Also, we can find items' popularities in the last period of time.

In this paper, we enhance the basic MCRS as follows:

- Firstly, the basic MCRS is enhanced using items' popularity in general.
- Extra enhancement is done using items' popularity in the last period of time.

The experimental results show that the time factor affects positively or negatively in the recommendation. The enhancements of MCRS using the time factor outperform the basic MCRS.

The future researches can be done to enhance MCRS using time series.

REFERENCES

[1] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Comput. vol. 7, no. 1, pp. 76–80, 2003.

[2] A. Das, "Google News Personalization : Scalable Online," WWW 2007 / Track Ind. Pract. Exp., pp. 271–280, 2007.

[3] A. Edmunds and A. Morris, "The problem of information overload in business organisations: a review of the literature," Int. J. Inf. Manage., vol. 20, no. 1, pp. 17–28, Feb. 2000.

[4] P. Lops, M. De Gemmis, and G. Semeraro, Recommender Systems Handbook. Boston, MA: Springer US, 2011.

[5] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," Phys. Rep., vol. 519, no. 1, pp. 1–49, 2012.

[6] A. Hossein, N. Rafsanjani, N. Salim, and A. R. Aghdam, "Recommendation Systems : a review," Int. J. Coputational Engneering Res., vol. 03, no. 5, pp. 47–52, 2013.

[7] G. P. Basharin, A. N. Langville, and V. a. Naumov, "The life and work of A.A. Markov," Linear Algebra Appl., vol. 386, pp. 3–26, Jul. 2004.

[8] D. P. Foster and L. H. Ungar, "Spectral dimensionality reduction for HMMs," Representations, no. 1989, pp. 1–26, Mar. 2012.

[9] D. Stirzaker, "Markov Chains," pp. 396–477, Oct. 2003.

[10] M. Jacobsen, "Markov Chains and Markov Processes," no. 1, pp. 1–3, Sep. 2006.

[11] T. Konstantopoulos, "MARKOV CHAINS AND RANDOM WALKS," 2009.

[12] T. Markov, "FINITE-STATE MARKOV CHAINS," vol. 103.

[13] G. Shani, D. Heckerman, and R. I. Brafman, "An MDP-Based Recommender System ∗," vol. 6, pp. 1265–1295, 2005.

[14] S. Rendle, C. Freudenthaler, and L. Schmidt-thieme, "Factorizing Personalized Markov Chains for Next-Basket Recommendation," 2010.