2A1-IOS-3b-4

A Community-based Service Recommendation System

Cheng Wei Wu[1], Shun-Chieh Lin[2], Huan-Wen Tsai[2], Kuang-Hung Cheng[2], Vincent S. Tseng[1*]

[1]Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, ROC
[2]Cloud Service Technology Center, Industrial Technology Research Institute (ITRI South), Taiwan, R.O.C
silvemoonfox@gmail.com, , {Jason.lin, hwtsai, chengkh}@itri.org.tw,
*Correspondence: tsengsm@mail.ncku.edu.tw

1. Introduction

Recommendation system is an important topic in many applications. The purpose of a recommendation system is to provide users with items/contents/services that meet their needs. Many studies have been devoted to devloping effective and efficient recommendation sysytem. They can be generally categorized into two types: *CF-based* (*Collaborative Filtering-based*) and *pattern-based reommendation systems*.

Given a set of user-to-item rating scores, CF-based approaches [4, 5, 6, 11, 16, 21] recommend items/contents/services/objects based on the historical user-to-item rating scores. However, in some applications, it is not easy to obtain user-to-item rating scores. For example, in a product marketing application, it is not easy to obtain user-to-item rating scores for all items that users ever purchased. Another limitation of CF-based approaches is that the performance of CF-based approaches is easy to be affected by the number of collected rating scores. In other words, insufficient number of rating scores may easily degrades the effectiveness of CF-based recommendation systems.

Another common recommendation system is pattern-based recommendation system [1, 2, 8, 9, 10, 15, 17, 19, 21, 22]. The main idea of pattern-based recommendation system is to use pattern mining skills, such as *frequent itemset* [1, 10] and *sequential pattern mining* [2, 9, 15, 17, 19, 22], to discover user behvior from their historical log data. The pattern-based approaches firstly mine rules or patterns from historical log data of users and then recommend items/contents/services/objects to users when user behaviors match discovered rules or patterns. However, one of the drawbacks of pattern-based recommendation is that it can not recommend users with items that users never purchased or used before.

In addition, *social network-based recommendation system* (*SNRS*) [3, 9, 12, 13, 18, 20, 23, 24, 25] has received lots of attentions in recent years. A social network consists of several entities. Entities between entities are connected by certain relationships. With the increasing popularity of Web 2.0, many social networking websites such as Facebook, Twitter, and Digg have emerged. Members in these social networking websites have their own personalized space. They can publish their interests, articles and biographies, and can send messages to other members. A social network can be transformed into a social graph. In a social graph, entities are represented as nodes and relationships between entities are represented as edges. Edges between nodes indicate the relationships (such as friend, follower, following relationships) between members. Social network-based recommendation system considers activities of users and relationships between users to recommend items/contents/services to users. Although edges between nodes can be used to represent certain relationships, they could not reflex similarities between members. For a recommendation system, it is common to consider similarities between members to enhance its performance. How to calculate similarities between members according to their behaviors is one of the important issues for recommendation systems.

Although many studies have been proposed for efficient recommendation sysytem, most of them recommend items/contents/services/objects to users according to the behavior of users in a single community without considering behavior of other users in different communities. Besides, the traditional recommendation systems may have the following problems: (1) *Cold-start problem*: when the number of user-to-item rating scores is few, the effectivenss of the recommendation system is limited. (2) *New user problem*: it is hard to predict the preference of a new user when the user does not rate scores for items. (3) *New item problem*: it is hard to predict the preference of a user to a new item since the new item is not rated by any user.

In this paper, we propose an effective community-based recommendation system to address the above issues. The proposed recommendation system utilizes behavior of users in different communities to reduce the influence of cold-start and new user/item problems. We investigated properties of different similarity measurements and proposed a pattern-based similarity measurement to calculate the similarities between users. Besides, we proposed several preference functions to calculate the user preference for items. The experimental results show that the proposed method outperforms traditional CF-based approaches.

The remainder of this paper is organized as follows. In Section 2, we introduce the scenario and define the problems. The proposed method is described in Section 3. Experimental results are shown in Section 4. Conclusion is given in Section 5.

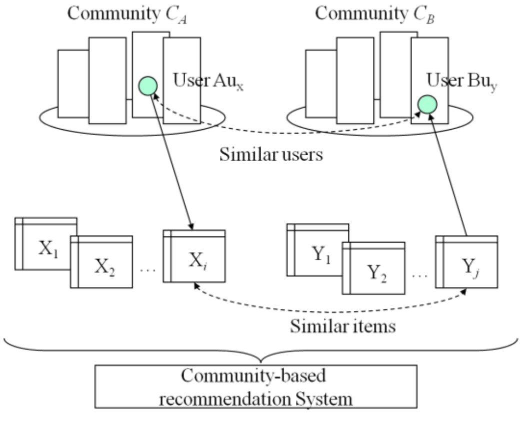


Figure 1. Scenario

2. The Proposed Framework

In this section, we introduce scenario, problem definitions and assumption of the proposed framework.

2.1 Problem Definitions and Scenario

A *community C* consists of *n* distinct *users* $\{u_1, u_2,\ldots,u_n\}$. Let *I* be a finite set of distinct *items* $\{i_1, i_2, \ldots, i_m\}$. Each *user* u_i $(1 \le i \le n)$ of *C* is associated with a *transactional database D*. A transactional database $D = \{T_1, T_2, \ldots, T_s\}$ is a set of *transactions*, where each transaction $T_r \in D$, $(1 \le r \le s)$ is a subset of *I* and has an unique identifier *r*, called its *Tid*. An item in a transaction can be considered as a product that purchased by a customer or a service that accessed by a customer. Let C_A and C_B be two different communities. Users in the communities C_A and C_B are denoted as Au_i $(i \ge 1)$ and Bu_j $(j \ge 1)$ respectively. Items in a community can be categorized into two types: *global item* and *local item*. Every user can purchase/access global item, but a local item in a community *C* only can be purchased/accessed by the local users of *C*. Let $X = \{X_1, X_2, \ldots, X_p\}$ $(X_i \in I,\ 1\le i \le p)$ and $Y = \{Y_1, Y_2, \ldots, Y_q\}$ $(Y_j \in I,\ 1\le j \le q)$ be sets of local items in C_A and C_B, where $X \cap Y = \phi$. Consider the scenario shown in the Figure 1. Let $X_i \in I$ $(1\le i \le p)$ be a local item that purchased by the user $Au_x \in C_A$ $(x \ge 1)$ and $Y_i \in I$ $(1\le i \le q)$ be a local item that never be purchased by $Bu_y \in C_B$ $(y \ge 1)$. If buying behavior of user Au_x is similar to that of Bu_y $(y \ge 1)$ and local item X_i $(1\le i \le p)$ is similar to local item Y_i $(1\le j \le q)$, we trend to recommend item Y_i to user Bu_y.

2.2 Assumption

In the proposed framework, we assume that the similarities between items are already known and stored in an *item similarity matrix*. Table 1 shows an item similarity matrix. The value of item similarity ranges from 0 to 1, where 1 represents completely similar and 0 represents completely dissimilar. The similarities between items can be obtained by comparing their meat-data or other features such as price and category. To simply the problem, we assume similarities between items are already known and are stored in the item similarity matrix.

Table 1. Item similarity matrix

Item similarity matrix				
Item	i_1	i_2	…	i_m
i_1	1	0.1	…	0.2
i_2	…	1	…	0.3
…	…	…	1	0.5
i_m	…	…	…	1

Table 2. Transactional database for a user in the community C_A

Transactional database	
T_1	G_1, G_2, X_1, X_2
T_2	G_1, X_2
T_3	G_2, X_1
T_4	G_1, G_2, X_1

	Au_1	Au_2	…	Au_5
X_1	5	2	…	4
X_2	4	1	…	3
G_1	1	5	…	2
G_2	3	5	…	2

	Bu_1	Bu_2	…	Bu_5
Y_1	5	2	…	3
Y_2	3	1	…	2
G_1	3	3	…	2
G_2	3	4	…	2

Figure 2. Rating score tables for users in C_A and C_B

3. The Proposed Method

In this section, we present the proposed method. For each user, we calculate *supports of items*. The *support count* of an item is the total number of transactions which contain the item. The *support* of an item is defined as the ratio of the number of transactions which contain this item to the total number of transactions in the database. For example, Table 2 shows a transactional database for a user in the community C_A. The global item G_1 appears in the transactions T_1, T_2 and T_4. The support count of G_1 is 3. The support of G_1 is (3/4) = 75%. We transform the supports of items to user-to-item rating scores that range from 1 to 5 to express the user preference to the items. The rating scores are stored in a *rating score table*. Figure 2 shows two rating score tables for users in community C_A and C_B respectively. In the Figure 2, the rating score of user Au_1 for the local item X_1 and global item G_1 are 5 and 1 respectively.

For each user, we mine *frequent itemsets* from his/her transactional database. The concept of frequent itemset is described as follows. An itemset $X = \{i_1, i_2, \ldots, i_\ell\}$ is a set of ℓ distinct items, where $i_j \in I$, $1 \le j \le \ell$, and ℓ is the *length* of *X*. An itemset *X* is said to be contained in a transaction T_r if $X \subseteq T_r$. The *support count* of an itemset is the number of transactions which contain the itemset. The *support of an itemset* is defined as the ratio of the number of transactions which contain the itemset to the total number of transactions in the database. An itemset is called *frequent itemset* if its support is no less than a user-specified *minimum support threshold*. For example, the itemset $\{G_1, X_1\}$ is contained in transactions T_1 and T_4. The support count and support of $\{G_1, X_1\}$ are 2 and (2/4) = 50%, respectively. When the minimum support threshold is 50%, $\{G_1, X_1\}$ is a frequent itemset.

After discovering frequent itemsets from transactional database of each user, each user has a frequent pattern set (a set of frequent patterns that mined from his/her transactional database). The similarities between any two users can be measured by comparing their frequent pattern sets. For any two users, if their frequent pattern sets are similar, they may have similar buying behavior. Figure 3 shows a similarity function for calculating similarity between two frequent pattern sets.

– Association Pattern

➢ *Variation of Structure*

$$V_{str}\ (p_1, p_2) = \begin{cases} 0, & if\ \ p_1 \subseteq p_2 \\ 1 - \dfrac{|\, p_1 \cap p_2 \,|}{|\, p_1 \cup p_2 \,|} & ,else \end{cases}$$

➢ *Variation of Support*

$$V_{\sup}\ \ (s_1, s_2) = \frac{|\, s_1 - s_2 \,|}{\max\,(s_1,\, s_2)} \times Log_{\max\,(s_1,\, s_2)}\, |\, s_1 - s_2 \,|$$

➢ *Variation of two pattern (set)*

$$V_{ap}(p_1,\ p_2) = w \times V_{str}(p_1,\ \ p_2) + (1-w) \times V_{\sup}(s_1,\ \ s_2)$$

$$V_{AP} = \frac{\displaystyle\sum_{i=i}^{n} V_{ap}\,(p_i, couple(p_i)) \times \Big(\frac{s_i}{\sum_{j=1}^{n} s_j}\Big) + \sum_{i=i}^{m} V_{ap}\,(p_i', couple(p_i')) \times \Big(\frac{s_i'}{\sum_{j=1}^{m} s_j'}\Big)}{2}$$

Figure 3. Similarity function for two frequent pattern sets

Table 3. User similarity matrix

User similarity matrix				
User	u_1	u_2	…	u_n
u_1	1	0.8	…	0.7
u_2	…	1	…	0.6
…	…	…	1	0.4
u_n	…	…	…	1

The proposed similarity function is called *VPS* (*Variation of two Pattern Sets*), which is described as follows. For any two patterns p_1 and p_2, their similarity can be measured by *structure variation* and *support variation*. The structure variation of two patterns p_1 and p_2 is denoted as $V_{str}(p_1, p_2)$ and is defined as *Jaccard similarity* if p_1 is not a subset or superset of p_2. Otherwise, the structure variation of p_1 and p_2 is 0. The support variation of p_1 and p_2 is denoted as $V_{sup}(s_1, s_2)$. The definition of $V_{sup}(s_1, s_2)$ is shown in the Figure 3, where s_1 and s_2 are the supports of p_1 and p_2 respectively. The *variation of two patterns* is denoted as $V_{ap}(p_1, p_2)$ and defined as $w \times V_{str}(p_1, p_2) + (1 - w) \times V_{sup}(s_1, s_2)$, where w is a user-specified weight that used to balance the importance of $V_{str}(p_1, p_2)$ and $V_{sup}(s_1, s_2)$. For any two pattern sets P_A and P_B, their similarity is denoted as V_{AP} and calculated as follows. For each pattern p_i in the pattern set P_A, we search the most similar pattern *couple*(p_i) from P_B and calculate their similarity by $V_{ap}(p_i, couple(p_i))$. Let S_A be the summation of similarities of such pairs for all patterns in P_A. For each pattern p_i' in P_B, we search the most similar pattern *couple(*p_i*')* from P_A and calculate their similarity by $V_{ap}(p_i', couple(p_i'))$. Let S_B be the summation of similarities of such pairs for all patterns in P_B. The similarity of P_A and P_B is $V_{AP} = (S_A+ S_B)/2$.

The similarity between any two users can be calculated by the method mentioned above. After calculating similarities for all users, similarities between users are stored in a *user similarity matrix*. For example, Table 3 shows a user similarity matrix and the similarity between users u_1 and u_2 is 0.8.

Other similarity measurements can be replaced. The following equations are *Jaccard*, *Extended Jaccard* and *Cosine similarity* measurements.

(*a*) Jaccard

$$J(p_1, p_2) = \frac{|p_1 \cap p_2|}{|p_1 \cup p_2|} \quad \text{…Equation (1)}$$

(*b*) Extended Jaccard

$$EJ(p_1, p_2) = \frac{|p_1 \cap p_2|}{|p_1 \cup p_2| - |p_1 \cap p_2|} \quad \text{…Equation (2)}$$

(*c*) Cosine

$$COS(p_1, p_2) = \frac{|p_1 \cap p_2|}{\sqrt{|p_1||p_2|}} \quad \text{…Equation (3)}$$

Equations (1), (2) and (3) are suitable for categorical data. The following equations can be used for measuring the similarities of numerical data. Given two ordered sequences $V_1 = <V_1^1, V_1^2,…,V_1^N>$ and $V_2 = <V_2^1, V_2^2,…,V_2^N>$，we can calculate the similarity between V_1 and V_2 by the following equations (4)~(8).

(*d*) Euclidean

$$D_{EUC}(V_1, V_2) = \sqrt{\sum_{i=1}^{N} |V_1^i - V_2^i|^2} \quad \text{…Equation (4)}$$

(*e*) City Block

$$D_{CB}(V_1, V_2) = \sum_{i=1}^{N} |V_1^i - V_2^i| \quad \text{…Equation (5)}$$

(*f*) Minkowski

$$D_{MK}(V_1, V_2) = \sqrt[p]{\sum_{i=1}^{N} |V_1^i - V_2^i|^p} \quad \text{…Equation (6)}$$

(*i*) Chebyshev

$$D_{CHEB}(V_1, V_2) = \max\{v \mid v = |V_1^i - V_2^i|, i = 1 \sim N\} \quad \text{…Equation (7)}$$

(*j*) PCC(Pearson Correlation Coefficient)

$$D_{PCC}(V_1, V_2) = \frac{\sum_{i=1}^{N} (V_1^i - \overline{V_1})(V_2^i - \overline{V_2})}{\sqrt{\sum_{i=1}^{N} (V_1^i - \overline{V_1})^2}\sqrt{\sum_{i=1}^{N} (V_2^i - \overline{V_2})^2}} \quad \text{…Equation (8)}$$

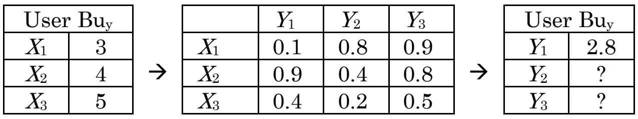
To recommend items in the category Y for a user Bu_y of the community C_B, we predict the rating score of Bu_y for every item in category Y and recommend k items with highest predicted rating scores (top-k items) to Bu_y. To predict the rating score of Bu_y for an item $Y_j \in Y= \{Y_1, Y_2, …, Y_q\}$, $1\leq j \leq q$, we first predict the rating score of Bu_y for every item $X_i \in X = \{X_1, X_2, …, X_p\}$, $1\leq i \leq p$. Let $p_1(Bu_y, X_i)$ be the predicted rating score of Bu_y for the item X_i, $\{Au_1, Au_2,…, Au_k\}$ be the k users in the community C_A with the highest similarities with Bu_y. The predicted rating score of Bu_y for the item X_i can be calculated by the Equation (9).

$$p_1(\mathrm{Bu_y}, \mathrm{X_i}) = \frac{1}{k}\sum_{v=1}^{k} sim(\mathrm{Bu_y}, \mathrm{A}u_\mathrm{v}) \times r(\mathrm{A}u_\mathrm{v}, \mathrm{X_i}) \quad \text{… Equation (9)}$$

In Equation (9), $sim(Bu_y, Au_v)$ is the similarity between users Bu_y and Au_v, $r(Au_v, X_i)$ is the rating score of Au_v for the item X_i. The predicted rating score of Bu_y for the item X_i $p_1(Bu_y, X_i)$ is calculated by considering similarities and rating scores of k most similar users in community C_A. After calculating rating scores of Bu_y for every item in category X, we predict the rating score of Bu_y for every item $Y_j \in Y= \{Y_1, Y_2, …, Y_q\}$, $1 \leq j \leq q$, by considering k most similar items in category X and the rating scores of Bu_y for these items. Let $p_1(Bu_y, Y_j)$ be the predicted rating score of Bu_y for the item Y_j, $\{X_1, X_2,…, X_k\}$ be the k items in category X with the highest similarities with Y_j. The predicted rating score of Bu_y for the item Y_j can be calculated by the Equation (10).

$$p_1(\mathrm{Bu_y}, \mathrm{Y_j}) = \frac{1}{k}\sum_{v=1}^{k} r(\mathrm{Bu_y}, \mathrm{Y_v}) \times sim(\mathrm{X_v}, \mathrm{Y_j}) \quad \text{… Equation (10)}$$

In Equation (10), $sim(X_v, Y_j)$ is the similarity between X_v and Y_j, $r(Bu_y, Y_v)$ is the rating score of Bu_y for the item Y_v. We take Figure 4 as an example to explain how to predict rating score of Bu_y for the item Y_1 when $k = 2$. According to Equation (10), $p_1(Bu_y, Y_1) = (r(Bu_y, X_2)\times sim(X_2, Y_1)+r(Bu_y, X_3)\times sim(X_3, Y_1))/2 = (4\times 0.9+5\times 0.4)/2 = 2.8$



User	Bu_y
X_1	3
X_2	4
X_3	5

→

	Y_1	Y_2	Y_3
X_1	0.1	0.8	0.9
X_2	0.9	0.4	0.8
X_3	0.4	0.2	0.5

→

User	Bu_y
Y_1	2.8
Y_2	?
Y_3	?

Figure 4. Predict rating score of Bu_y for the item Y_1

In this study, we propose other preference functions for predicting rating scores of B_1 for the item X_i. The other preference functions are p_2、p_3、p_4 and p_5。Equation (11)~(14) shows the functions. In these equations, R is the maximum value of the rating score.

$$p_2(B_1, X_i) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(B_1, A_j) \times p(A_j, X_i)\right)}{\sum_{j=1}^{k} sim(B_1, A_j)} \qquad \cdots\text{Equation (11)}$$

$$p_3(B_1, X_i) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(B_1, A_j) \times p(A_j, X_i)\right)}{\frac{1}{R}\sum_{j=1}^{k} p(A_j, X_i)} \qquad \cdots\text{Equation (12)}$$

$$p_4(B_1, X_i) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(B_1, A_j) \times p(A_j, X_i)\right)}{\left(\sum_{j=1}^{k} sim(B_1, A_j) + \frac{1}{R}\sum_{j=1}^{k} p(A_j, X_i)\right)/2} \qquad \cdots\text{Equation (13)}$$

$$p_5(B_1, X_i) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(B_1, A_j) \times p(A_j, X_i)\right)}{\sqrt{\sum_{j=1}^{k} sim(B_1, A_j) + \frac{1}{R}\sum_{j=1}^{k} p(A_j, X_i)}} \qquad \cdots\text{Equation (14)}$$

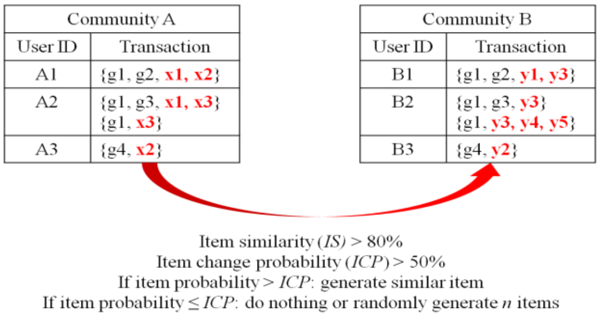
Based on Equation (11)~(14), there are four equations for calculating preference of B_1 for item Y_1, which are shown in the Equations (15)~(18).

$$p_2(B_1, Y_1) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(Y_1, X_j) \times p_2(B_1, X_j)\right)}{\sum_{j=1}^{k} sim(Y_1, X_j)} \qquad \cdots\text{Equation (15)}$$

$$p_3(B_1, Y_1) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(Y_1, X_j) \times p_3(B_1, X_j)\right)}{\frac{1}{R}\sum_{j=1}^{k} p(B_1, Xj)} \qquad \cdots\text{Equation (16)}$$

$$p_4(B_1, Y_1) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(Y_1, X_j) \times p_4(B_1, X_j)\right)}{\left(\sum_{j=1}^{k} sim(Y_1, X_j) + \frac{1}{R}\sum_{j=1}^{k} p(B_1, X_j)\right)/2} \qquad \cdots\text{Equation (17)}$$

$$p_5(B_1, Y_1) = \frac{\frac{1}{k}\left(\sum_{j=1}^{k} sim(Y_1, X_j) \times p_5(B_1, X_j)\right)}{\sqrt{\sum_{j=1}^{k} sim(Y_1, X_j) + \frac{1}{R}\sum_{j=1}^{k} p(B_1, X_j)}} \qquad \cdots\text{Equation (18)}$$



Community A	
User ID	Transaction
A1	{g1, g2, **x1, x2**}

A2	{g1, g3, **x1, x3**}
	{g1, **x3**}

A3	{g4, **x2**}

Community B	
User ID	Transaction
B1	{g1, g2, **y1, y3**}

B2	{g1, g3, **y3**}
	{g1, **y3, y4, y5**}

B3	{g4, **y2**}

Item similarity (*IS*) > 80%
Item change probability (*ICP*) > 50%
If item probability > *ICP*: generate similar item
If item probability ≤ *ICP*: do nothing or randomly generate *n* items

Figure 5. Generating community C_B from C_A

Table 4. Parameters and Descriptions

Parameters	Notations	Default
Number of items	*N*	1,500
Number of users per community	*NU*	1,000
Number of transaction in a community	*D*	100,000
Number of transactions per users	*NT*	100
Average size of transactions	*T*	10
Average size of potential frequent itemset	*I*	6
Item similarity threshold	*IS*	0.5
Item change probability	*ICP*	0.9
Minimum support threshold	*MST*	5%

4. Experimental Evaluation

4.1 Simulation Data and Parameter Settings

We use IBM data generator [1] to produce the transactions in community A. The parameters are described in Table 4. We use Figure 5 to explain how to use transactions in community *A* to generate transactions in community *B*. During the generation, user need to set a threshold named *item change probability* (*ICP*) threshold and an *item similarity probability* (*IS*) threshold. For each transaction T_A in the community *A*, we generate a similar transaction T_B in the community *B*. For each item *x* in the transaction T_A, we generate a positive value named *IP* that ranges from 0 to 1. If *IP* is no less than the user-specified *ICP* threshold, the system generates an item *y* for T_B, where the similarity between *x* and *y* is no less than *IS*. If *IP* is smaller than the *ICP*, the system does not generate any item or randomly generate *RN* items for T_B. In our experiments, the parameter *RN* is set to 1.

After generating the transactions in communities, we transform transactions into rating table. To validate the effectiveness of our approaches, we randomly delete some rating scores in the rating table. Then we use the proposed method to predict the rating scores that have been deleted. The scores that are predicted by our methods are called predicted rating scores. We use MAE, RMSE and NDCG to evaluate the effectiveness of the proposed methods.

(a) MAE (Mean Square Error)

MAE is calculated by the Equation (19). In the Equation (19), *N* is the number of predict rating scores, r_i is the *i*-th rating score, pr_i is the *i-th* predicted rating score. Let's take Table 5 as a running example. MAE is (|3-4|+|4-4|+|5-3|+|2-1|+|1-1|)/5 = 0.8.

$$MAE = \frac{1}{N}\sum_{i=1}^{k} |r_i - pr_i| \qquad \cdots\text{Equation (19)}$$

Table 5. Original rating scores and predicted rating scores

Item	A	B	C	D	E
Rating score	3	4	5	2	1
Predict rating score	4	4	3	1	1

(b) RMSE (Root Mean Square Error)

RMSE is calculated by the Equation (20). Let's take Table 5 as a running example. RMSE is $[(|3-4|^2 + |4-4|^2 + |5-3|^2 + |2-1|^2 + |1-1|^2)/5]^{1/2}$= 1.0954.

$$RMAE = \sqrt{\frac{1}{N}\sum_{i=1}^{k} |r_i - pr_i|^2} \qquad \cdots\text{Equation (20)}$$

(c) NDCG (Normalize Discount Cumulative Gain)

If we rank items in their rating score and predict rating score, we can obtain two ranking orders. The purpose of NDCG measurement is to compare the difference between the two orders. The related equation is shown as Equation (21) and (22). In Equation (21) and (22), the variable *p* is the number of items that are recommended, the variable *i* is the rank of the item, rel_i is the original rating score of *i*-th item in the rating table.

$$DCG_p = rel_i + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \qquad \cdots\text{Equation (21)}$$

$$NDCG@p = \frac{DCG_p}{IDCG_p} \qquad \cdots\text{Equation (22)}$$

Table 6. Items are ranked according to their rating scores

Rank	*i*	rel_i	$\log_i$	$rel_i/\log_i$
A	1	15	0	N/A
C	2	10	1	10
B	3	3	1.59	1.89
D	4	0	2.0	0
E	5	0	2.32	0

Table 7. Items are ranked according to their predicted rating scores

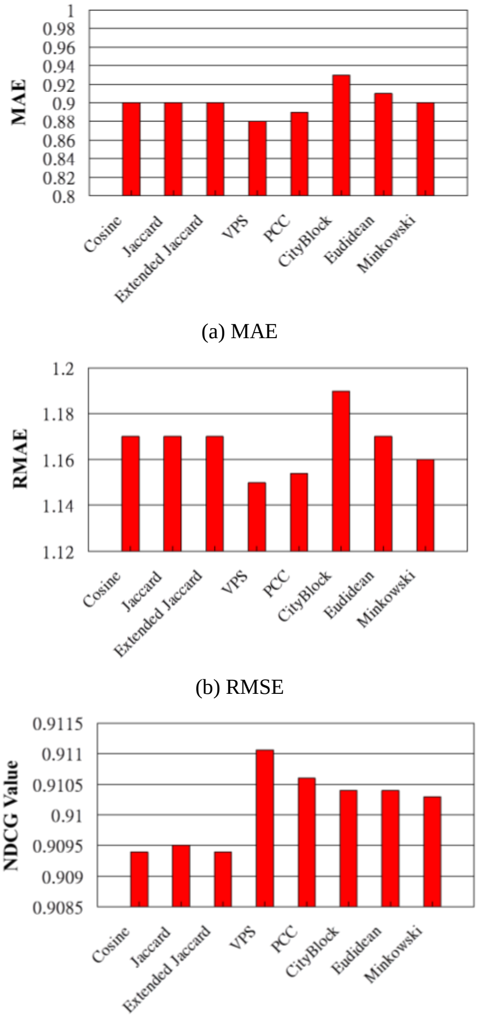
Rank	*i*	rel_i	$\log_i$	$rel_i/\log_i$
A	1	15	0	N/A
D	2	0	1	0
B	3	3	1.59	1.89
C	4	10	2.0	5
E	5	0	2.32	0

Let's take the Table 6 and Table 7 as running examples. In Table 6, the rating scores of A, B, C, D and E are 15, 3, 10, 0 and 0. After arranging items in descending order of their rating scores, we obtain the list A, C, B, D, E. Therefore, the value *i* for A, C, B, D, E are 1, 2, 3, 4, 5. The corresponding $\log_2 i$ for A, C, B, D, E are 0, 1, 1.59, 2, 2.32. The corresponding ($rel_i/\log_2 i$) for A, C, B, D, E are N/A,10, 1.89, 0 and 0. The idea DCG value ($IDCG_5$) is (15+10+1.89+0+0) = 26.89.

Table 7 shows the ranking of A, B, C, D, E according to the descending order of their predicted rating scores, that is, A, D, B, C, E. The value *i* for A, D, B, C, E are 1, 2, 3, 4, 5. The corresponding $\log_2 i$ for A, D, B, C, E are 15, 0, 3, 10, 0. The corresponding ($rel_i/\log_2 i$) for A, D, B, C, E are N/A, 0, 1.89, 5 and 0. The DCG_5 for predict rating score is (15+0+1.89+5+0) =21.89. NDCG@5 = $DCG_5/NDCG_5$ = 21.89/26.89 = 0.814.

4.2 Performance Evaluation for Different Similarity Measurements and Preference Functions

Figure 6 shows the effectiveness of our methods by using different similarity measurements. The preference functions used in this experiment are Equations (11) and (15). The parameter *k* for finding top-k items/users in our method is set to 10. Figure 6 (a) shows that MAE of VPS is better than that of Cosine, Jaccard and Extended Jaccard. The best MAE among different similarity measurements is VPS. The MAE of CityBlock is the worst. Figure 6(b) shows RMSE for different similarity measurements. Results show that RMSE f VPS is the best. The worst is CityBlock. Figure 6(c) shows NDCG@10 for different similarity measurements. The best one is VPS and the worst one is PCC.

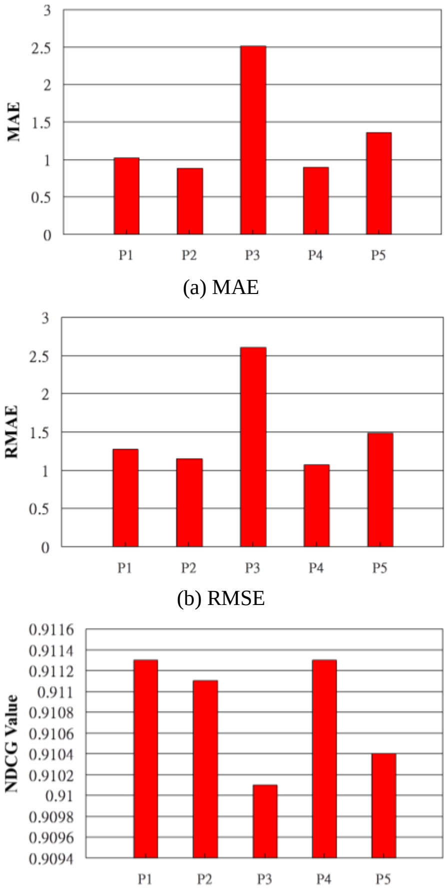


(a) MAE

(b) RMSE

(c) NDCG@10

Figure 6. The effectiveness of different similarity measurements



(a) MAE

(b) RMSE

(c) NDCG@10

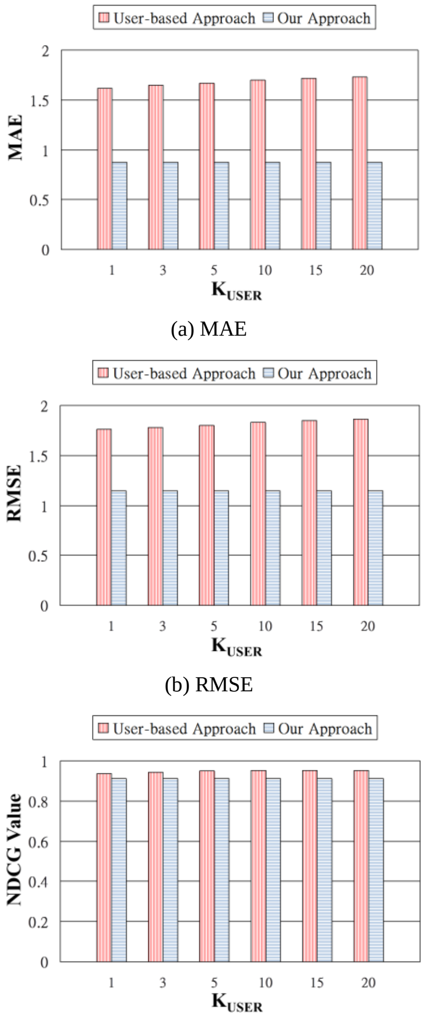
Figure 7. The effectiveness of different preference functions

4.3 Comparison with CF-based Approaches

Figure 7 shows the effectiveness of the different preference functions. The similarity measurement used in this experiment is VPS. The parameter k for finding top-k users/items in our method is set to 10. Figure 7(a) shows that the MAE of P3 is the worst and P2 is the best. Figure 7(b) shows that the RMSE of P4 is the best and P3 is the worst. The RMSE of P2 is close to P4. Figure 7(c) shows that the NDCG@10 of P4 is the best and the worst is P3. From the result, we can observe that the performance of P2 and P4 are better than the other preference functions.

Figure 8 shows the MAE, RMSE and NDCG@10 for User-based approach [5] and our approach. The preference function used in the experiment is P2. The parameter *k* for finding top-*k* users/items in our method is set to 10. During the calculation, User-based approach needs to find top-*k* users. The parameter for User-based approach is denoted as K_{USER}. We vary K_{USER} for User-based approach and compare the performance of User-based approach with our approach. Figure 8(a) shows that MAE of our approach is much smaller than that of User-based approach. MAE of the User-based approach is about twice higher that of our approach. Figure 8(c) shows that the NDCG@10 of User-based approach is a little higher than that of our approach. But NDCG@10 for both methods are higher than 0.95.

Figure 9 shows the MAE, RMSE and NDCG@10 for Item-based approach [5, 16] and our approach. The preference function used in the experiment is P2. The parameter k for top-k in our method is set to 10. During the calculation, Item-based approach needs to find top-k items, the parameter for Item-based approach is denoted as K_{USER}. We vary K_{USER} for User-based approach and compare the performance of User-based approach with our approach. Figure 9 shows that MAE of our approach is much smaller than that of User-based approach. MAE of the User-based approach is about twice higher that of our approach. Figure 9(c) shows that the NDCG@10 of User-based approach is a little higher than that of our approach. But NDCG@10 for both methods are higher than 0.95.

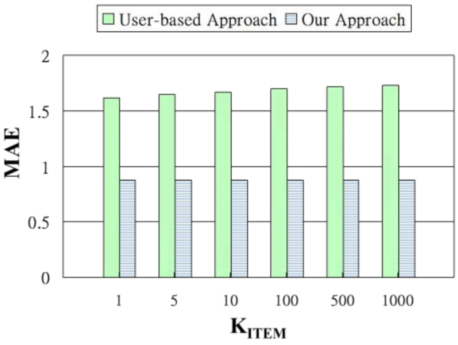


(a) MAE

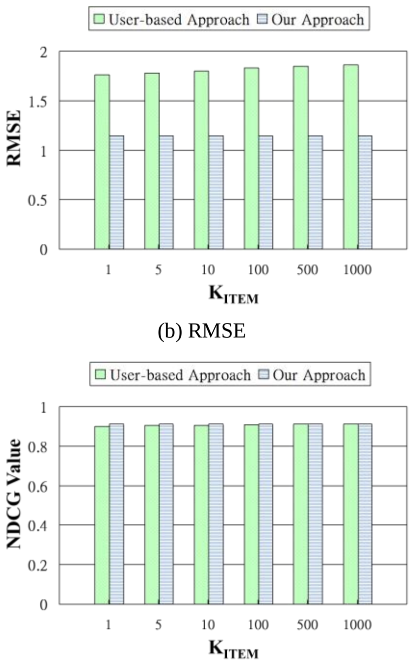
(b) RMSE

(c) NDCG@10

Figure 8. Comparing our method with User-based approach



(a) MAE



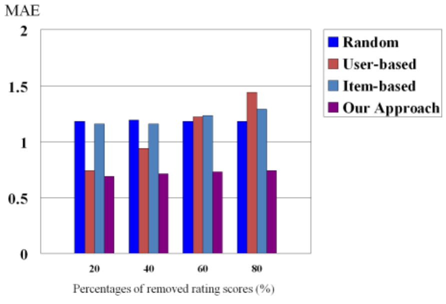
(b) RMSE

(c) NDCG@10

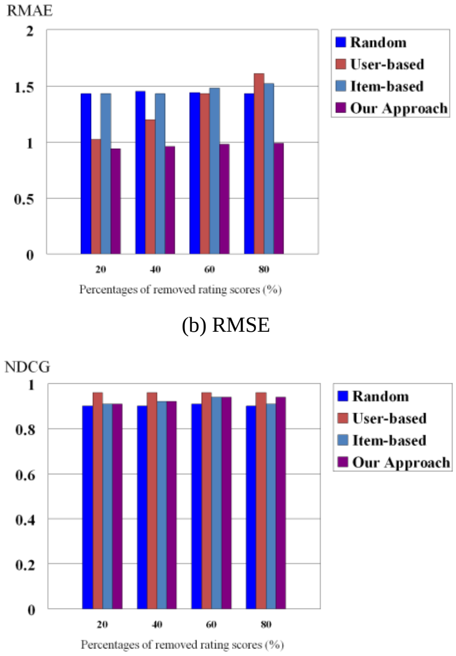
Figure 9. Comparing our method with Item-based approach

4.4 Robustness on Sparse Data

In this subsection, we test the robustness of the proposed method for the sparse data. The preference function used in the experiment is P2. In this experiment, we remove different percentages of rating scores of users in the community *B*. Figure 10 shows the performance of the proposed method and the CF-based approaches on T10I2N1KD10K dataset. In the Figure 10, values at *x*-axis represent different percentages of rating scores that are removed from the community *B*. Results show that our approach is better than User-based, Item-based approaches and random method (randomly recommend items / contents / services / objects to users). With increasing sparsity, MAE of User-based approach increases. In addition, MAE and RMSE of Item-based approach are higher than 1. MAE and RMSE of the random method ranges from 1.3 to 1.4 and do not change too much with increasing sparsity. MAE and RMSE of the proposed method is lower 1, which shows that the proposed method is robust even with sparse data. For the NDCG@10, our approach is slightly lower than the User-based and Item-based approaches.



(a) MAE

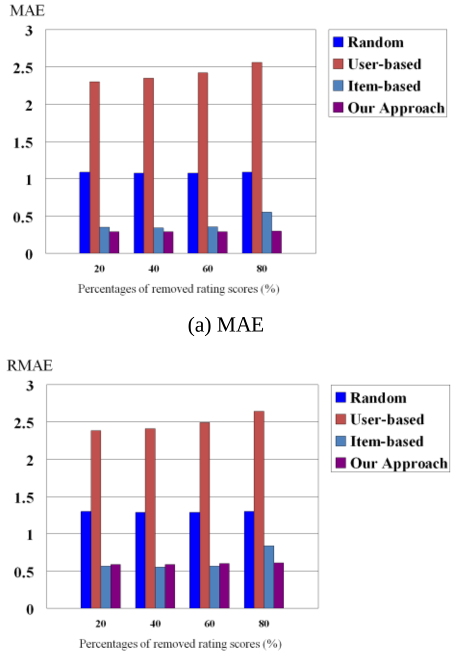


(b) RMSE

(c) NDCG@10

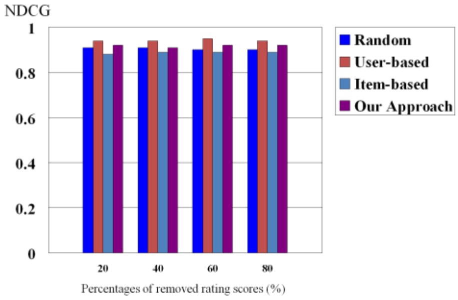
Figure 10. Performance of different methods on T10I2N1KD10K dataset

Figure 11 shows the performance of the proposed method and the CF-based approaches on T10I2N3KD10K dataset. Results show that our approach better than User-based, Item-based approaches and random method. When the sparsity increases, MAE of the User-based approach increases. In addition, MAE of the User-based approach are higher than 2. MAE and RMSE of the random method ranges from 1.2 to 1.3 and do not change too much with increasing sparsity. MAE and RMSE of the proposed method are lower than the Item-based approach. The NDCG@10 of our approach is very close to that of the User-based approach.



(a) MAE

(b) RMSE



(c) NDCG@10

Figure 11. Performance of different methods on T10I2N3KD10K dataset

5. Conclusion

In this paper, we propose a community-based recommendation system with a novel solution to reduce the influence of cold-start and new item/user problems. We propose *VPS* (*Variation of two Pattern Sets*) method to calculate the similarity between users. We also propose several preference functions to predict the preference of users. We utilize the information in other community to enhance the effectiveness of the recommendation system. The results show that the MAE and RMSE of our approach are much lower than that of the User-based and Item-based approaches. In addition, the NDCG of our method is very close to that of the User-based approach and Item-based approaches.

References

[1] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, In Proc. 20th Very Large Databases (VLDB) Conference, pp. 487-499, 1994.

[2] R. Agrawal, R. Srikant, Mining Sequential Patterns, in: Proceedings of the 11th International Conference on Data Engineering, 1995, pp. 3-14.

[3] C.C Aggarwal, J.L. Wolf, K-L. Wu and Yu P.S. Horting.(1999). Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In KDD'99, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 201-212. ACM Press.

[4] Kamal Ali and Wijnand van Stam , ”TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture” , in KDD 2004

[5] D. Billsus and M. Pazzani (1998). Learning Collaborative Information Filters, ICML 1998: 46-54.

[6] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. Of UAI*, 1998.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. Classification and Regression Trees. Wadsworth and Brooks, Pacific Grove, California.

[8] C. Budak, D. Agrawal, and A. E. Abbadi, “Structural Trend Analysis for Online Social Networks,” in VLDB 2011.

[9] Y.L. Chen, M.C. Chiang and M.T. Ko. Discovering time-interval sequential patterns in sequence databases. Expert Systems with Applications, vol. 25, 2003, pp. 343-354.

[10] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proc. of the 2000 ACM-SIGMOD Int'l Conf. on Management of Data, Dallas, Texas, USA, May 2000.

[11] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In Proc. of IJCAI, 1999.

[12] M. Jamali and M. Ester, “TrustWalker: a random walk model for combining trust-based and item-based recommendation”, In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 397-406, 2009.

[13] I. Konstas, V. Stathopoulos and J. M. Jose, “On social networks and collaborative recommendation” In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 195-202, 2009.

[14] G. Kossinets, J. Kleinberg, and D. Watts, “The Structure of Information Pathways in a Social Communication Network,” in SIGKDD 2008.

[15] Hye-Chung(Monica) Kum, Joong Kyuk Chang, and Wei Wang “Sequential Pattern Mining in Multi-Databases via Multiple Alignment.” Data Min. Knowl. Discov. 12(2-3):151-180, 2006.

[16] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1):76–80, 2003.

[17] Matthieu Capelle, Cyrille Masson, and Jean-Francois Boulicaut “Mining Frequent Sequential Patterns under a Similarity Constraint.” IDEAL 2002: 1-6.

[18] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network Motifs: Simple Building Blocks of Complex Networks,” Science, 298(25), 2002.

[19] P. Moen “Attribute, Event Sequence, and Event Type Simarity Notions for Data Mining.” PhD thesis, Dept. of Computer Science, University of Helsinki, Finland, February 2000.

[20] J.-P. Onnela, J. Sarama, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A.-L. Barabasi, ”Structure and Tie Strengths in Mobile Communication Networks,” PNAS, 104(18), 2007.

[21] Pazzani, M., and D. Billsus. Content-based Recommendation Systems, in The Adaptive Web: Methods and Strategies of Web Personalization, P. Brusilovsky, A. Kobsa, and W. Nejdl, Editors. 2006, Springer-Verlag:London.

[22] R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the 5th International Conference on Extending Database Technology (EDBT), pages 3-17, Avignon, France, March 1996.

[23] P. Symeonidis, A. Nanopoulos and Y. Manolopoulos, “A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis”, . IEEE Trans. Knowl. Data Eng. 22(2): pp. 179-192, 2010.

[24] Q. Yuan, S. Zhao, L. Chen, S. Ding, X. Zhang and W. Zheng, “Augmenting Collaborative Recommender by Fusing Explicit Social Relationships”, In ACM Conference on Recommender Systems, workshop on Recommender Systems and the Social Web, New York City, NY, USA, October 22-25, 2009.

[25] Q. Zhao, Y. Tian, Q. He, N. Oliver, R. Jin, and W.-C. Lee, “Communication Motifs: A Tool to Characterize Social Communications,” in CIKM 2010.