Implicative Rating-Based Hybrid Recommendation Systems

Lan Phuong Phan, Hung Huu Huynh, and Hiep Xuan Huynh

Abstract—This paper proposes the implicative rating measures built on the statistical implicative measures. The paper also proposes the hybrid recommendation model, the combination of the user-based collaborative filtering approach and the association rule based approach using the implicative rating measures to suggest a list of top N items to active users. The proposed hybrid model is compared to some existing models on two datasets MSWeb and CourseRegistration. The experimental results show that the performance of the proposed model is higher than that of the compared models.

Index Terms—Implicative rating measure, hybrid recommendation model, user-based collaborative filtering, association rule.

I. INTRODUCTION

Recommender systems/recommendation systems (RSs) [1] are techniques or software tools embedded in an application or website to predict the preferences of an individual or a group of users for a specific product or service; and/or to recommend the appropriate products or services to an individual or a group of users, thereby reducing the information overload. Currently, the recommendation systems are applied in many areas of life [2] such as e-commerce, e-learning, e-services, etc. The techniques (methods) of recommendation are based on the ones used in data mining and machine learning [3], [4] such as classification, clustering, association rule mining, regression models, or some of the supervised/unsupervised learning methods. Recommendation techniques are divided into two main classes: the class of basic techniques such as collaborative filtering, content filtering or hybrid; and the class of techniques developed on the basic techniques and the additional data such as the social information or the contextual information. RSs can be classified into various groups [2], [3], [5]: content based, collaborative, knowledge based, demographic based, hybrid, context based, social based, and group based RSs. In the fields of research on RSs, improving the existing recommendation methods or proposing the new recommendation models has still been the mainstream of research and received the most attention.

The hybrid recommendation system is a combination of two or more recommendation techniques together to achieve the higher performance. Hybrid RSs can be grouped into seven variants: weighted, switching, cascade, feature augmentation, feature combination, meta-level, and mixed ones [3], [6]; and are in one of three forms [5]: monolithic, parallel, and sequential design. In the parallel design, the various RSs function independently of one another, and predictions of individual systems are combined at the end. Therefore, this design is the least invasive one.

Statistical implicative analysis [7] is a method of data analysis initiated by Gras to study the trends among data attributes. Statistical implicative analysis is applied in many areas such as psychology, education, bioinformatics, art history, knowledge management, etc. In this method, the statistical implicative measures are used to detect the strong relationships, or to measure the typicality of an object for the formation of a relationship, or to measure the responsibility of an object for the existence of a relationship. Therefore, the statistical implicative measures can be used to develop RSs.

This paper proposes a new hybrid recommendation model based on the association rules based recommendation approach and the user-based collaborative filtering approach to present to the active users the top N items. The paper also proposes the implicative rating measures built on the statistical implicative measures are used to predict users’ ratings for items. In the hybrid proposed model, the matrix of binary ratings is used as the main input.

The remaining of the paper is organized into four sections. Section II describes the statistical implicative measures briefly. Section III not only proposes the implicative rating measures built on the above mentioned measures, but also proposes the hybrid recommendation model using the implicative rating measures. Section IV presents the experimental results and the discussion of those results. Section V is the conclusion.

II. STATISTICAL IMPLICATIVE MEASURES

A. Implicative Intensity

Let \( E \) be a set of \( n \) objects described by a finite set of binary attributes. Let \( A \subseteq E \) be a subset of objects with the attribute \( a \). Let \( B \subseteq E \) be a subset of objects with the attribute \( b \); \( \overline{B} \) be the complement of \( B \); \( n_a = \text{card}(A) \) be the number of elements of \( A \); \( n_b = \text{card}(B) \) be the number of elements of \( B \); \( n_{\overline{B}} = n - n_b \) be the number of elements of \( \overline{B} \); and \( n_{aB} = \text{card}(A \cap \overline{B}) \) be the number of objects that has the attribute \( a \) but does not have the attribute \( b \).

Let \( X \) and \( Y \) be two random sets whose the element numbers are \( n_x \) and \( n_y \) respectively. The random variable \( \text{card}(X \cap \overline{Y}) \) follows the Poisson distribution with the parameter \( \lambda = n_x n_{\overline{Y}} / n \).

The probability of the \( \text{card}(X \cap \overline{Y}) \leq \text{card}(A \cap \overline{B}) \) is defined by (1).

\[
\Pr(\text{card}(X \cap \overline{Y}) \leq \text{card}(A \cap \overline{B})) = \sum_{s=0}^{\text{card}(A \cap \overline{B})} \frac{\lambda^s e^{-\lambda}}{s!}
\]

(1)

For \( n_{\overline{B}} \neq 0 \), the random variable \( \text{card}(X \cap \overline{Y}) \) is...
normalized to the random variable $Q(a, b)$ which is defined as in (2). In experiment, the observed value of $Q(a, b)$ is represented by $q(a, b)$ and is defined according to (3). $q(a, b)$ is called the implicative index.

$$Q(a, b) = \frac{\text{card}(X \cap Y) - n_0 n_2}{n}$$  \hspace{1cm} (2)

$$q(a, b) = \frac{n_0 n_2}{n}$$  \hspace{1cm} (3)

The implicative intensity [7] of the relationship $a \rightarrow b$ is defined by (4).

$$\varphi(a, b) = 1 - \Pr(Q(a, b) \leq q(a, b)) = \frac{1}{\sqrt{2\pi}} \int_{q(a, b)}^{\infty} e^{-t^2/2} dt$$  \hspace{1cm} (4)

if $n_b \neq n$ or $q(a, b) = 0$ otherwise.

The relationship $a \rightarrow b$ is admissible for a given threshold $\alpha$ if $q(a, b) \geq \alpha$.

B. Typicality Measure

The typicality measure is proposed as (5) [7] to calculate the typicality of an object $i$ for the formation of the relationship $a \rightarrow b$.

$$\gamma(i, a \rightarrow b) = 1 - \frac{d_1(i, a \rightarrow b)}{\max_{j=1}^{n} d_1(j, a \rightarrow b)}$$  \hspace{1cm} (5)

In (5), $d_1(j, a \rightarrow b)$ is the distance between the object $i$ and the relationship $a \rightarrow b$. The value of $d_1(j, a \rightarrow b)$ is calculated by (6) where $\varphi(a, b)$ is the implicative intensity of $a \rightarrow b$ and $\varphi_{i, a \rightarrow b}$ is the contribution (the responsibility) of the object $i$ for the existence of the relationship $a \rightarrow b$.

$$d_1(i, a \rightarrow b) = \sqrt{(\varphi(a, b) - \varphi_{i, a \rightarrow b})^2} / 1 - \varphi(a, b)$$  \hspace{1cm} (6)

The contribution $\varphi_{i, a \rightarrow b}$ is defined by (7) where $a(i)$ (resp. $b(i)$) is the binary value which characterizes the presence/absence of the attribute $a$ (resp.$b$) for object $i$.

$$\varphi_{i, a \rightarrow b} = 1 \text{ if } a(i) = 1 \text{ or } b(i) = 0$$
$$\varphi_{i, a \rightarrow b} = 0 \text{ if } a(i) = 1 \text{ and } b(i) = 0$$
$$\varphi_{i, a \rightarrow b} = p \in [0, 1] \text{ if } a(i) = b(i) = 0$$  \hspace{1cm} (7)

In practice, $p$ is set to the neutral value 0.5.

C. Cohesion Measure

The cohesion measure [7] is used for discovering the relationship with a good implicative quality. The cohesion value of the relationship $a \rightarrow b$, denoted by $c(a, b)$, is defined by (8).

$$c(a, b) = \sqrt{1 - (p \log_2 p + (1 - p) \log_2 (1 - p))}$$  \hspace{1cm} (8)

if $p \geq 0.5$ or $c(a, b) = 0$ otherwise.

In (8), $p$ is the implicative intensity of the relationship $a \rightarrow b$ (i.e. $p = \varphi(a, b)$).

III. HYBRID RECOMMENDATION BASED ON IMPLICATIVE RATING MEASURES

Like the two-dimensional recommendation problem [8], the proposed hybrid recommendation model can be formalized by the follows:

Let $U = \{u_1, u_2, \ldots, u_n \}$ be a set of users.

Let $I = \{i_1, i_2, \ldots, i_m \}$ be a set of items (e.g. products, movies, songs, etc.).

Let $R = (r_{jk})$ where $j = 1..n$ and $k = 1..m$ be a rating matrix storing the feedbacks of users on items. $r_{jk} = 1$ if the user $u_j$ likes the item $k$; $r_{jk} = 0$ if the user $u_j$ does not like (or know) the item $k$.

Let $f: U \times I \rightarrow \mathbb{R}$ be a function that maps the user-item combinations to the ratings $r$.

The objective of the proposed model is to find a function $f''': U \times I \rightarrow \mathbb{R}'''$ such that the performance (e.g. the precision and the recall) of the model is higher when compared to some other models.

Fig. 1 shows the proposed hybrid recommendation model combining two sub-models: association rule based collaborative filtering recommendation (or association rule based recommendation) and user based collaborative filtering recommendation (or user-based recommendation). In this model, the implicative rating measures (ArIR3, KnnIR and IR) are built on the statistical implicative measure such as the cohesion, the implicative intensity, and the typicality. The proposed hybrid model uses: (1) - the association rules based recommendation approach to predict the ratings of active user according to the measure ArIR3; (2) - the user-based approach to predict the ratings of active user according to the measure KnnIR; (3) - the implicative rating measure IR to predict the final ratings; and (4) the final ratings to filter, sort, and then recommend the top $N$ items to active user.

A. Implicative Rating Measures

We propose the measure ArIR3 (Association rule based Implicative Rating) to be used by the sub-recommendation model based on the set of association rules. ArIR3 predicts the ratings that can be given by the active user $u_i$ for each item $i \in I$. This measure is defined in (9) where AIR3 (Atom Implicative Rating) is the combination of the cohesion value and the confidence value as in (10).

$$\text{ArIR3}(u_i, i) = \frac{\text{AIR3}(u_i, i)}{\max_{j=1}^{n} \text{AIR3}(u_j, i)}$$  \hspace{1cm} (9)
In (10), \( n \) is the number of items that were rated by \( u_a \); \( r \) is the number of rules; \( \text{Conf}(\text{rule}_k) \) is the confidence value of the rule \( \text{rule}_k \); \( c(\text{rule}_k) \) is the cohesion value of \( \text{rule}_k \); and \( \text{rule}_k \) is in the form \( l_j \rightarrow i \) where \( l_j \) is one of \( n \) items rated by \( u_a \).

\[
\text{AIR}_3(u_a,i) = \sum_{j=1}^{m} \text{Cohe}_\text{Conf}(i,j) = \sum_{j=1}^{m} \text{Conf}(\text{rule}_j) \times c(\text{rule}_j)
\]  

(10)

We also propose the measure \( \text{KnnIR} \) (K nearest neighbors based Implicative Rating) to predict the ratings that can be given by the active user \( u_a \) for each item \( i \). This measure is used by the sub-recommendation model based on \( k \) nearest neighbors. \( \text{KnnIR} \) is built on KIR (Kernel Implicative Rating) as in (11).

\[
\text{KnnIR}(u_a,i) = \frac{\text{KIR}(u_a,i)}{\max_{i \neq i'} \text{KIR}(u_a,i')}
\]  

(11)

\( \text{KIR}(u_a,i) \) is calculated from the typicality values of the nearest neighbors of \( u_a \) who liked the item \( i \). KIR is defined by (12) in which \( k \) is the number of nearest neighbors; \( u_j \) is one of \( k \) nearest neighbors of \( u_a \); \( \gamma(i,u_a \rightarrow u_j) \) is the typicality value of item \( i \) for the relationship formation \( u_a \rightarrow u_j \); and \( r(u_j,i) \) is the rating value given by the user \( u_j \) for the item \( i \).

\[
\text{KIR}(u_a,i) = \sum_{j=1}^{k} \gamma(i,u_a \rightarrow u_j) \times r(u_j,i)
\]  

(12)

The hybrid model uses the implicative rating measure \( \text{IR} \) (Implicative Rating) to predict the final ratings and then to present the top \( N \) items to the active user. This measure is defined by (13). The value of \( w \) is identified by the experiment.

\[
\text{IR}(u_a,i) = w \times \text{KnnIR}(u_a,i) + (1-w) \times \text{AIR}_3(u_a,i)
\]  

(13)

B. Matrix \( \text{Cohe}_\text{Conf} \)

The matrix \( \text{Cohe}_\text{Conf} \) of size \( m \) items \( \times m \) items is used by the association rule based sub-model to reduce the recommendation time. The matrix is built on the association rules and the value of that rules according to the cohesion measure and the confidence measure. Each cell of the matrix is assigned a value calculated by (14) where the meaning of \( r \), \( \text{Conf}(\text{rule}_k) \), \( \text{rule}_k \), and \( c(\text{rule}_k) \) is explained in the above section.

\[
\text{Cohe}_\text{Conf}(i,j) = \sum_{k=1}^{m} \text{Conf}(\text{rule}_k) \times c(\text{rule}_k)
\]  

(14)

The matrix \( \text{Cohe}_\text{Conf} \) is created by Algorithm 1. Its inputs consist of: the rating matrix \( R \) of users \( U \) and items \( I \); the thresholds of measures (support, confidence) to be \( s \), \( c \) respectively; and the number of subsets \( k \) to be used for splitting the rating matrix. The main steps of Algorithm 1 are:

1. Splitting the rating matrix \( R \) into \( k \) folds of equal size.
2. Selecting \( (k-1) \) subsets randomly to conduct the following sub-steps: Generating the set of association rules by the algorithm Apriorio based on \( (k-1) \) randomly selected subsets and the thresholds \( s \) and \( c \); Creating the matrix \( \text{Cohe}_\text{Conf} \) according to (14) in which the cohesion value of a rule is calculated according to the algorithm that we presented in [9].

Repeating the second step in \( k \) times.

Calculating the average of matrix \( \text{Cohe}_\text{Conf} \), and writing it to the file in the .csv format.

C. Hybrid Recommendation Based on Implicative Rating Measures

The hybrid recommendation model uses Algorithm 2 (named as IRHybridRS) to present the top \( N \) items to active users. The inputs of this algorithm are: the vector \( A \) of size \( m \) representing the active user \( u_a \) with \( \text{given} \) known ratings; the rating matrix \( R \) of users \( U \) and items \( I \); the \( k \) nearest neighbors \( knn \); the threshold of the support measure \( s \); the threshold of the confidence measure \( c \); and the value of \( w \) to be used in (13). The output of Algorithm 2 is the list of top \( N \) items to be recommended to the active user \( u_a \).

After importing the matrix \( \text{Cohe}_\text{Conf} \) from the .csv file, Algorithm 2 conducts the following steps.

1. Predicting the ratings of the active user \( u_a \) for each item \( i \) based on the matrix \( \text{Cohe}_\text{Conf} \) and the measure \( \text{ArIR3} \) according to Algorithm 3.

2. Predicting the ratings of the active user \( u_a \) for each item \( i \) based on the \( k \) nearest neighbors \( knn \) of \( u_a \) and the measure \( \text{KnnIR} \) according to Algorithm 4.

3. Calculating the final rating value for each item \( i \) based on the predicted ratings and the implicative rating measure \( \text{IR} \) defined in (13).

4. Removing the given known items of \( u_a \) from the list of predicted ratings.

5. Sorting the filtered list in descending order and recommending the top \( N \) items to the active user \( u_a \).

IRHybridRS(vector A; ratingmatrix R; int knn; float s,c,w) {
    Cohe Conf = ReadFile("Cohe Conf.csv");
    ArIR3 = ArIRRS(A,Cohe Conf);
    KnnIR = KnnIRs(A,R,knn);
    IR = w*KnnIR + (1-w)*ArIR3;
    Ratings = RemoveKnownGiven(A, IR);
    PresentTopNItems(Ratings);
    }

D. Association Rule Based Recommendation

In the association rule based recommendation sub-model, Algorithm 3 (named as ArIRRS) is used for predicting the ratings of the active user. This algorithm is based on the association rules and the measure \( \text{ArIR3} \). The inputs of Algorithm 3 are: the vector \( A \) of size \( m \) representing the active user \( u_a \) with \( \text{given} \) known ratings; the square matrix \( \text{Cohe}_\text{Conf} \) of size \( m \) items \( \times m \) items consisting of the values calculated by Algorithm 1. The output of Algorithm 3 is the predicted ratings of the active user \( u_a \). This algorithm is presented as the following.

ArIRRS (vector A; matrix Cohe Conf) {
for each item $i \in I$
for each item $j \in I$
if $(A[I][j] == 1)$
$AIR3[u, j] += \text{Cohe}_{\text{Conf}}[i, j]$
$ARIR3[u, ] = AIR3[u, ] / \max(\theta_{\text{AIR3}}(u, ))$
return $ARIR3$

E. User Based Recommendation

In the user-based collaborative filtering recommendation sub-model, Algorithm 4 (named as KnIRR S) is implemented to predict the ratings of the active user. This algorithm is based on the k nearest users $k_{\text{nn}}$ and the measure KnIR. Its inputs are: the vector $A$ of size $m$ representing the active user $u_a$ with given known ratings; the rating matrix $R$ of users $U$ and items $I$; and the k nearest neighbors $k_{\text{nn}}$. The output of Algorithm 4 is the predicted ratings of the active user $u_a$. The processing steps of this algorithm are the follows.

Calculating the implicative intensity of the relationship between the user $u_a$ and a user $u_t \in U$ according to the algorithm that we presented in [9].

Identifying the k nearest neighbors $k_{\text{nn}}$ of $u_a$ that have the highest implicative intensity values.

Calculating the typicality value of each item $i_j \in I$ for the formation of relationship $(u_a, u_t)$ where $u_t$ is one of k nearest neighbors of $u_a$ according to the algorithm that we presented in [10].

Calculating and returning the rating value for each item $i_j \in I$ based on the measure KnIR defined in (11).

$\text{KnIRR S}(\text{vector A; rating matrix R; int knn})$
$\text{Intensity} = \text{CalculatelImplicavtiveIntensity}(A, R)$
$\text{Neighbors} = K\text{Nearest} \text{Neighbors}(\text{Intensity, knn})$
$\text{Typic} = \text{Typicality}(\text{A, Neighbors, R, Intensity})$
for each $i \in I$
do$\text{KIR}[u_a, i] = \text{ColSum}(\text{Typic} \times R[\text{Neighbors}])$
$\text{KnIR}[u_a, i] = \text{KIR}[u_a, i] / \max(\text{KIR}[u_a, i])$
return $\text{KnIR}$

IV. EXPERIMENTS

A. Experimental Data

Two datasets used in the experiment are MSWeb, and CourseRegistration. The dataset MSWeb was generated by sampling and processing the logs of www.microsoft.com in one week timeframe. For each user of anonymous and randomly selected users, this dataset lists all the areas of the website (Vroots) visited by that user. MSWeb consists of: 32710 users, 285 Vroot, and 98653 ratings with value of TRUE. The dataset CourseRegistration is collected through the Can Tho University’s admissions website https://htqu.edu.vn. This dataset saves the course registration of students who will take part in the third semester (of a program of study) at Faculty of Information and Communication Technology. The dataset consists of 1172 students, 81 courses and 5705 ratings (registrations) with values of TRUE.

These experimental datasets need to be preprocessed to increase the accuracy of recommendations. The evaluations may be biased if we keep items that are only rated a few times and users who only rate a few items. Besides, we should also pay attention to the constraints in the registration regulations when conducting the recommendation on the dataset CourseRegistration. For example, the maximum number of

credits a student can register for a semester is limited to 20 credits; in most cases, each course is worth 3 credits; one of eligibility criteria for receiving the scholarship is that a student has to register at least 15 credits per semester; etc. Therefore, (1) - for the dataset MSWeb, the number of users viewing at least 10 Vroots and the number of Vroots viewed by at least 50 users are selected to extract data; (2) - for the dataset CourseRegistration, the number of students registering at least 5 courses and the number of courses registered by at least 20 students are chosen to extract data. The general information of these datasets after filtering is shown in Table I.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of users</th>
<th>Number of items</th>
<th>Number of ratings</th>
<th>Maximum number of given items</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSWeb</td>
<td>875</td>
<td>135</td>
<td>10487</td>
<td>7</td>
</tr>
<tr>
<td>CourseRegistration</td>
<td>779</td>
<td>36</td>
<td>4095</td>
<td>3</td>
</tr>
</tbody>
</table>

Moreover, in the association rules based recommendation approach, the support threshold and the confidence threshold should be assigned to suitable and small values to do not miss the association rules with high quality. We presented how to select these values in [10]. For both MSWeb and CourseRegistration, thresholds of confidence and support are assigned to 0.1 and 0.01 respectively ($c=0.1$, $s=0.01$).

B. Experimental Tool

The proposed hybrid recommendation model is developed in the R language and uses the functions that we built in the Interestingsnesslab tool [11]. Besides, we also use some recommendation models of the recommenderlab package\(^2\) to compare with the proposed model. Those models are: (1) - the popular model recommending the most common items to the active users (POPULAR); (2) - the item based collaborative filtering model using the Jaccard measure (IBCF); (3) - the user based collaborative filtering model using the Jaccard measure (UBCF); and (4) - the association rule based model using the confidence measure (AR).

C. Evaluation Method

To evaluate the recommendation models, the k-fold cross validation method [12] is used to split the dataset into k folds (subsets) of equal size, evaluates $k$ times, and then calculates the average. The dataset (rating matrix) is splitted into the training set and the test set. The test set is then divided into the query set and the target set which have the same size. However, for each user, the query set only has the randomly selected given known ratings; the target set consists of the remaining ratings. The query set together with the training set are used to predict recommendations whereas the target set is used to evaluate the recommended results. For each time of evaluation, $(k-1)$ folds are used as the training set and the remaining fold is used as the test set. In this experiment, $k$ is chosen as 4. When the number of items (the length of recommendation list) that need to be presented to the user is not predetermined, evaluating the algorithm over a range of lengths of recommendation list is preferable rather than using the maximum number of given items for each user of the query set (the maximum number of known items of an active user) is used for building recommendations and evaluating the recommendation models. This number is based on the percentiles of ratings.

\(^2\)https://cran.r-project.org/web/packages/recommenderlab
a fixed length. Therefore, the Precision-Recall and ROC (Receiver Operating Characteristic) curves are often used [13]. In addition, the ROC curve is often used to compare the performance of multiple algorithms. An algorithm is more efficient if its ROC curve completely dominates the ROC curves of other algorithms [13]. The Precision - Recall curve is built on the precision and the recall. The ROC curve is based on the sensitivity and the complement of specificity. The sensitivity (also called True Positive Rate - TPR) is equivalent to the recall. The complement of specificity is also called False Positive Rate (FPR) which is the probability of false alarm. Moreover, the measure named as the area under the ROC curve is also used to evaluate the performance of recommendation models. This measure summarizes the precision and the recall of ROC curve which is useful for comparing algorithms independently of application [13].

D. Experimental Results

Scenario 1: Identifying the value of \( w \)

To identify the suitable value of \( w \) in (13) on the specific dataset, nine RSs are developed by using the proposed hybrid model. Nine RSs are: IRHybrid RS01, IRHybrid RS02, IRHybrid RS03, IRHybrid RS04, IRHybrid RS05, IRHybrid RS06, IRHybrid RS07, IRHybrid RS08, and IRHybrid RS09 using the value of \( w \) from 0.1 to 0.9 respectively.

On the dataset CourseRegistration, we conduct the evaluation where the number of known ratings of each user (i.e., the number of courses registered by each student) in the query set is from 2 to 3 (given = 2 or 3); the number of nearest neighbors is from 20 to 80 (\( knn = 20 \) or 30 or \ldots or 70 or 80); and the numbers of recommendations (courses) to be presented to the active user are: 1, 2, 3, 4, and 5. Fig. 2 displays the area under ROC curves of nine RSs when given=2 or 3, and \( knn=20 \). The experimental result shows that the area of ROC curve of IRHybrid RS 08 is greater than that of the other RSs. When changing the value of \( knn \), we obtain the same result. Therefore, the value of weight \( w \) to be used in (13) for the dataset CourseRegistration is 0.8.

On the dataset MSWeb, we calculate the area under the ROC curves of nine RSs in the same way as we do for the dataset CourseRegistration. The parameters used for this case are: the number of known ratings of each user in the query set to be from 2 to 7 (given = 2 or 3 or 4 \ldots or 7); the number of nearest neighbors to be from 10 to 100 (\( knn = 10 \) or 20 or 30 or \ldots or 100); the numbers of recommendations presented to the active user to be: 1, 2, 3, 4, 5, 6, \ldots, 11, and 12. As the result, the value of \( w \) to be used in (13) is 0.2 if the given is greater than 2 or 0.8 otherwise.

Scenario 2: Evaluating the recommendation models

To compare the proposed model and some existing models of the recommenderlab package, five RSs are developed. They are named as: Popular RS, IBCFJaccard RS, UBCFJaccard RS, IRHybrid RS, and ARConfidence RS. Popular RS uses the popular model to recommend the most common items to the active users. IBCFJaccard RS uses the item based collaborating filtering model whereas UBCF Jaccard RS uses the user based collaborating filtering model. Both systems use the Jaccard measure to find the nearest neighbors. ARConfidence RS uses the association rule based model and the confidence measure for finding the relevant items. IRHybrid RS uses the proposed hybrid model and the value of \( w \) identified as in scenario 1. The thresholds of confidence and support used by ARConfidence RS and IRHybrid RS are \( c=0.1, s=0.01 \) respectively.

On the dataset MSWeb, we conduct the evaluation where the number of known ratings of each user (i.e., the number of Vroots visited by each user) in the query set is from 2 to 7 (given = 2 or 3 or 4 \ldots or 7); the number of nearest neighbors is from 10 to 100 (\( knn = 10 \) or 20 or 30 or \ldots or 100); the numbers of recommendations (Vroots) to be presented to the active user are: 1, 2, 3, 4, 5, 6, \ldots, 11, and 12.

Fig. 3 displays the ROC curves of five RSs on the dataset MSWeb. Fig. 4 displays the precision – recall curves of those models with given=6 and \( knn=60 \). It shows that the precision and the recall of the proposed model are higher than those of the other models. When changing the value of given and \( knn \), we get the results similar to these figures.

On the dataset CourseRegistration, we conduct the evaluation where given = 2 or 3; \( knn = 20 \) or 30 or \ldots or 70 or 80; and the numbers of recommendations to be presented to
the active user: 1, 2, 3, 4, and 5. Fig. 5 displays the ROC curves of the five recommendation models with given=2 and knn=30. Fig. 6 displays the precision – recall curves of those models with given=3 and knn=70. The results show that the ROC curve of the proposed model dominates the other ROC curves; the probability of false alarm (FPR) of the proposed model is lower than that of the other models; the precision and the recall of the proposed model are higher than those of other models. When changing the values of given and knn, we get the results similar to these figures.

Therefore, the performance of the propose hybrid model is higher than that of other compared models.

Fig. 6. The Precision-Recall curves of five RSs on the dataset CourseRegistration.

V. CONCLUSIONS

The paper proposes: (1) - a hybrid recommendation model developed on two sub-recommendation models (user-based collaborative filtering model and association rule based model) to present the top N items to the active user; and (2) - the implicating rating measures ArIR3, KnnIR, and IR to predict the ratings used by sub-models and the hybrid model respectively. The hybrid model is compared to some existing recommendation models integrated in the recommenderlab package on two datasets MSWeb and CourseRegistration. The compared models are the popular model, the item based collaborative filtering model using the Jaccard measure, the user based collaborative filtering model using the Jaccard measure, the association rule based model using the confidence measure, and the proposed hybrid model. The experimental results show that the performance of the hybrid model is higher than that of remainders. However, the proposed hybrid model is based on the collaborating filtering (association rule based and user based approaches), we will improve the performance of the hybrid recommendation model when the number of given known ratings is less than 2 in the future work. Besides, we will use the hybrid model based on implicating rating measures to develop a recommendation system in the reality.

REFERENCES