

User based Recommender Systems using Implicative Rating Measure

Lan Phuong Phan

College of Information and
Communication Technology
Can Tho University
Can Tho, Viet Nam

Hung Huu Huynh

University of Science and
Technology
University of Da Nang
Da Nang, Viet Nam

Hiep Xuan Huynh

College of Information and
Communication Technology
Can Tho University
Can Tho, Viet Nam

Abstract—This paper proposes the implicative rating measure developed on the typicality measure. The paper also proposes a new recommendation model presenting the top N items to the active users. The proposed model is based on the user-based collaborative filtering approach using the implicative intensity measure to find the nearest neighbors of the active users, and the proposed measure to predict users' ratings for items. The model is evaluated on two datasets MovieLens and CourseRegistration, and compared to some existing models such as: the item based collaborative filtering model using the Jaccard measure, the user based collaborative filtering model using Jaccard measure, the popular items based model, the latent factor based model, and the association rule based model using the confidence measure. The experimental results show that the performance of the proposed model is better when compared to other five models.

Keywords—Implicative rating measure; recommender system; user-based collaborative filtering

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 or 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 contextual information or the social information. Recommendation systems can be classified into different groups [2], [3], [5]: content based, collaborative, demographic based, knowledge based, hybrid, context based, social based, and group based. In the fields of research on recommendation systems, proposing the new recommendation models or improving the existing recommendation methods has still been the mainstream of research and received the most attention.

In recommendation techniques, collaborative filtering [1], [6]-[8] is the most important and widely used technique. A collaborative filtering system provides recommendations to an active user based on the assumption that similar users like similar items or a user prefers similar items. Therefore, the core component of collaborative filtering is the use of measures to find items or users with strong relationships, to predict ratings, and then to recommend the most relevant items to the active users.

Statistical implicative analysis is a method of data analysis initiated by Gras [9] to study the trends among data attributes [10]. Statistical implicative analysis is applied in many areas such as psychology, education, bioinformatics, knowledge management, art history, etc. [10], [11]. In this method, 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, statistical implicative measures can be used to develop the collaborative filtering systems.

This paper proposes a new recommendation model based on user-based collaborative filtering and the implicative rating measure to present to the active users the top N items. In the proposed model, the matrix of binary ratings is used as the main input; the implicative intensity is used to find the nearest neighbors of the active users; and the implicative rating measure developed on the typicality measure is used to predict users' ratings for items.

The remaining of the paper is organized into four sections. Section 2 describes the statistical implicative measures briefly. Section 3 not only proposes the implicative rating measure developed on the above mentioned measures, but also proposes the recommendation model using the user based collaborative filtering approach and the implicative rating measure. Section 4 presents the experimental results and the discussion of those results. Section 5 is the conclusions.

II. STATISTICAL IMPLICATIVE MEASURES

A. Implicative Intensity

Let E be a set of n objects described by a finite set of binary attributes I . 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 ; B_c 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 , $n_{b-} = n - n_b$ be the number of elements of B_- ; and $n_{ab-} = \text{card}(A \cap B_-)$ 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 with the element numbers to be n_a and n_b respectively.

The random variable $\text{card}(X \cap Y_-)$ follows the Poisson distribution with the parameter $\lambda = n_a n_b / n$. The probability of the $\text{card}(X \cap Y_-) \leq \text{card}(A \cap B_-)$ is defined by (1).

$$Pr(\text{card}(X \cap Y_-) \leq \text{card}(A \cap B_-)) = \sum_{s=0}^{\text{card}(A \cap B_-)} (\lambda^s / s!) e^{-\lambda} \quad (1)$$

For $n_b \neq 0$, the random variable $\text{card}(X \cap Y_-)$ is normalized to the random variable $Q(a, b_-)$ as (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.

$$Q(a, b_-) = (\text{card}(X \cap Y_-) - n_a n_b / n) / \text{sqr}t(n_a n_b / n) \quad (2)$$

$$q(a, b_-) = (n_{ab-} - n_a n_b / n) / \text{sqr}t(n_a n_b / n) \quad (3)$$

The implicative intensity [9] of the relationship $a \rightarrow b$ is defined by (4).

$$\varphi(a, b) = 1 - Pr(Q(a, b_-) \leq q(a, b_-)) \text{ if } n_b \neq n \quad (4)$$

$$\varphi(a, b) = 0 \text{ otherwise}$$

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

B. Typicality

To measure the typicality of an object i for the formation of the relationship $a \rightarrow b$, the typicality is proposed as (5) [9].

$$\gamma(i, a \rightarrow b) = 1 - d_I(i, a \rightarrow b) / \max_{j \in E} d_I(j, a \rightarrow b) \quad (5)$$

$d_I(j, a \rightarrow b)$ is the distance between the object i and the relationship $a \rightarrow b$. The value of $d_I(j, a \rightarrow b)$ is calculated by formula (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 $a \rightarrow b$.

$$d_I(i, a \rightarrow b) = ((\varphi(a, b) - \varphi_{i, a \rightarrow b})^2 / (1 - \varphi(a, b)))^{1/2} \quad (6)$$

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 .

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

$$\varphi_{i, a \rightarrow b} = 0 \text{ if } [a(i)=1 \text{ and } b(i)=0]$$

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

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

III. USER-BASED RECOMMENDATION MODEL USING IMPLICATIVE RATING MEASURE

Like the two-dimensional recommendation problem [7], the proposed recommendation model can be formalized by the follows:

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

- Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of items (e.g. products, movies, 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 $f: U \times I \rightarrow R$ 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' : $U \times I \rightarrow R'$ such that the performance (e.g. the precision and the recall) of the model is better when compared to some other models.

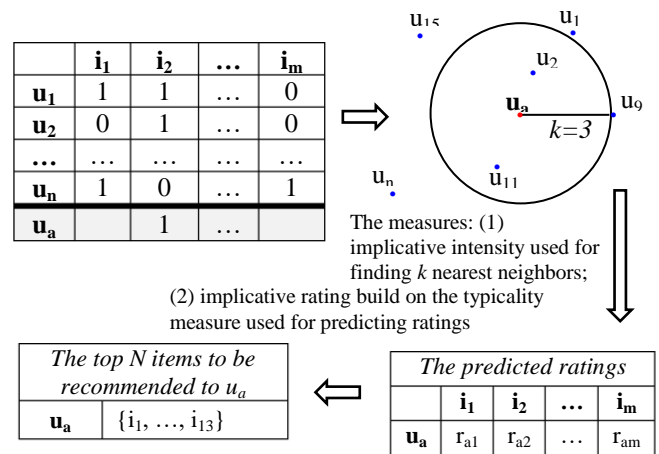


Fig. 1. The sketch of the proposed recommendation model.

The proposed recommendation model is sketched as Fig. 1. This model uses the user-based collaborative filtering approach and the implicative rating measure to recommend the top N items to the active user as described in Algorithm 1. During the recommendation process, the model will also use Algorithms 2 and Algorithm 3.

A. Implicative Rating Measure

We propose the implicative rating measure $KnnIR$ (K nearest neighbors based implicative rating) to predict the ratings that can be given by the active user u_a for each item $i \in I$. $KnnIR$ is built on the typicality measure γ and defined in formula (8).

$$KnnIR(u_a, i) = KIR(u_a, i) / \max_{l \in I} KIR(u_a, l) \quad (8)$$

Where

$$KIR(u_a, i) = \sum_{j=1}^k \gamma(i, u_a \rightarrow u_j) \text{ if } R(u_j, i) = 1 \quad (9)$$

k is the number of nearest neighbors; and $\gamma(i, u_a \rightarrow u_j)$ is the typicality of item i for the relationship formation $u_a \rightarrow u_j$.

B. Recommending the Top N Items to the Active User

The inputs of Algorithm 1 (recommending the top N items to the active user – UBCFImplicativeRS) are: the vector A of size m representing the active user u_a with given known ratings; the rating matrix R of U users and I items; and k nearest neighbors. Its output is the list of top N items to be

recommended to the active user u_a . The processing steps of Algorithm 1 are as follows:

- Calculating the implicative intensity of the relationship between the user u_a and a user $u_i \in U$ according to Algorithm 2.
- Identifying k nearest neighbors 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_i) where u_i is one of k nearest neighbors of u_a according to Algorithm 3.
- Predicting the rating value for each item $i_j \in I$ based on the implicative rating measure.
- Removing the given known items of u_a from the predicted list.
- Sorting the filtered list in descending order and recommending the top N items to the active user u_a .

UBCFImplicativeRS (vector A , ratingmatrix R , int k) {

$IIntensity = calculateImplicativeIntensity(A, R);$

$Neighbors = findKNearestNeighbors(IIntensity, k);$

$Typic = calculateTypicality(A, Neighbors, R, IIntensity);$

for each $i_j \in I$ do

$KIR[u_a, i_j] = \frac{sumColumn(Typic[Neighbors, i_j]}{*R[Neighbors, i_j];}$

$KnnIR[u_a] = KIR[u_a] / maxRow(KIR[u_a]);$

$Filteredlist = removeKnownRatings(A, KnnIR);$

$Reclist = getTopNItems(Filteredlist);$

return $Reclist$;

}

C. Calculating the Implicative Intensity of Relationship (U_a, U_i)

The inputs of Algorithm 2 (calculating the implicative intensity of relationship (u_a, u_i) – calculateImplicativeIntensity) are: the vector A of size m representing the active user u_a with given known ratings; and the rating matrix R of U users and I items. The output of this algorithm is the implicative intensity of relationship (u_a, u_i) with $u_i \in U$. Algorithm 2 conducts the following steps:

- Finding the values n , n_a , n_b and n_{ab} which represent the implicative relationship between two users u_a and u_i .
- Calculating the implicative intensity of (u_a, u_i) using formula (4).

calculateImplicativeIntensity(vector A , ratingmatrix R) {

$R = bindRow(R, A);$ $IIntensity = Null;$

for each $u_i \in U$ do {

$sum = 0;$

$n = countColumn(R);$ $n_a = 0;$

$n_b = 0;$ $n_{ab} = 0;$
for each $i_j \in I$ do {
 $n_a += R[u_a, i_j];$
 $n_b += R[u_i, i_j];$
 $n_{ab} += R[u_a, i_j] * R[u_i, i_j];$
}
 $n_{ab} = n_a - n_{ab};$
 $lambda = n_a * (n - n_b) / n;$
for $s = 0$ to n_{ab} do
 $sum += (lambda^s / fact(s)) * exp(-lambda);$
 $p = 1 - sum;$
 $IIntensity = bindRow(IIntensity, p);$
}
return $IIntensity$;

D. Calculating the Typicality of an Item for the Formation of a Relationship

Algorithm 3 (calculateTypicality) gets the following inputs: the vector A of size m representing the active user u_a with given known ratings; the vector $Neighbors$ consisting of k nearest neighbors of u_a ; the rating matrix R of U users and I items; and the vector $IIntensity$ containing the implicative intensity values of the relationship (u_a, u_i) with $u_i \in U$. This algorithm outputs the matrix $Typicality$ consisting of the typical values of item $i_j \in I$ for the relationship formation (u_a, u_i) with $u_i \in Neighbors$. The processing steps of Algorithm 3 are as follows:

- Filtering the rating matrix R that only contains the ratings of k nearest neighbors of u_a .
- Filtering the vector $IIntensity$ that only consists of the implicative intensity of k nearest neighbors of u_a .
- Calculating the contribution of item i_j for the existence of relationship (u_a, u_i) with $u_i \in Neighbors$ according to formula (7).
- Calculating the distance from i_j to the relationship (u_a, u_i) with $u_i \in Neighbors$ according to formula (6).
- Finding the typical value of item i_j for relationship formation (u_a, u_i) with $u_i \in Neighbors$ according to formula (5).

calculateTypicality($A, Neighbors, R, IIntensity$) {

$R = R[Neighbors,];$

$IIntensity = IIntensity[Neighbors];$

for each $u_i \in Neighbors$ do

for each item $i_j \in I$ do {

if $(R[u_i, i_j] = 1)$ Contribution $[u_i, i_j] = 1;$

else if $(A[i_j] = 1$ and $R[u_i, i_j] = 0)$

Contribution $[u_i, i_j] = 0;$

else if $(A[i_j] = 0$ and $R[u_i, i_j] = 0)$

Contribution $[u_i, i_j] = 0.5;$

Dist $[u_i, i_j] = sqrt((IIntensity[u_i] - Contribution[u_i, i_j])^2 / (1 - IIntensity[u_i]));$

}

Typicality = Null;

for each $u_i \in Neighbors$ do {

Rowmax = maxRow(Dist $[u_i,]$);

```
Typic = 1 - Dist[ui] / Rowmax;  
Typicality = bindRow(Typicality,Typic);  
}  
return Typicality;  
}
```

IV. EXPERIMENTS

A. Experimental Setup

1) Experimental Data

Two datasets used in the experiment are MovieLens and CourseRegistration. The MovieLens dataset is collected through website movielens.umn.edu during the seven-month timeframe. The dataset consists of 943 users, 1664 movies, and 99392 ratings (with values of 1-5) storing the feedbacks of users to movies. To be used in the proposed model, the MovieLens dataset is binarized with the threshold of 3. The ratings are set to 1 if they are greater than or equal to this threshold and 0 otherwise. The CourseRegistration dataset is collected through the Cantho University's admissions website <https://htql.ctu.edu.vn>. The 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 1.

To increase the accuracy of recommendations, the experimental datasets need to be preprocessed. If we keep items that are only rated a few times and users who only rate a few items, the evaluations may be biased. Besides, we should also pay attention to the constraints in the registration regulations when conducting the recommendation on the CourseRegistration dataset. For example, the maximum number of credits a student can register for a semester (except in the last semester of study) 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; and in order to open a course, there must be at least 25 students registered.

Therefore, with the MovieLens dataset, the number of users viewing at least 50 movies and the number of movies viewed by at least 100 users are selected to extract data. With the CourseRegistration dataset, the number of students registering at least 5 courses and the number of courses registered by at least 25 students are chosen to extract the data. The general information of these datasets after filtering is shown in Table 1.

TABLE I. GENERAL INFORMATION OF MOVIELENS AND COURSEREGISTRAION AFTER FILTERING

Dataset	The number of users	The number of items	The number of ratings	The maximum number of given items*
MovieLens	565	336	48,698	14
CourseRegistration	779	36	4,095	3

*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) Experimental Tool

The proposed recommendation model is developed in the R language and uses the functions that we built in the Interestingnesslab tool [12]. Besides, we also use some recommendation models of the recommenderlab package¹, to compare with the proposed model. Those models are: the item based collaborative filtering model using the Jaccard measure (IBCF), the user based collaborative filtering model using the Jaccard measure (UBCF), the popular model recommending the most common items to the active users (POPULAR), the latent factor model (ALS_implicit), and the association rule based model using the confidence measure (AR).

3) Evaluation

To evaluate the recommendation models, the rating matrix (dataset) 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. The k-fold cross validation method is used to split the dataset into *k* folds (subsets) of equal size, evaluates *k* times, and then calculates the average. 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 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). In addition, we also use the accuracy measure (the fraction of correct recommendations to total possible recommendations) and the F1 measure (the harmonic mean of the precision and the recall) to evaluate the performance of recommendation models. These six measures are built on the values of the confusion matrix shown in Table 2, and defined in formula (10), (11), (12), (13), and (14). In Table 2, TN is the number of items not recommended by the system and also not preferred by users; FP is the number of items recommended by the system but not preferred by users; FN is the number of items not recommended by the system but preferred by users; and TP is the number of items recommended by the system and also preferred by users.

¹<https://cran.r-project.org/web/packages/recommenderlab>

TABLE II. CONFUSION MATRIX

Actual/ Predicted	Not recommended	Recommended
Not preferred	True-Negative (TN)	False-positive (FP)
Preferred	False-Negative (FN)	True-Positive (TP)

$$Precision = TP/(TP+FP) \quad (10)$$

$$Recall \text{ or } TPR=TP/(TP+FN) \quad (11)$$

$$FPR=1-specificity=1-TN/(TN+FP)=FP/(TN+FP) \quad (12)$$

$$Accuracy = (TN+TP)/(TN+FP+FN+TP) \quad (13)$$

$$F1=2*precision*recall/(precision+recall) \quad (14)$$

B. Results

To compare the proposed model and some existing recommendation models of the recommenderlab package, six RSs are developed. They are named as: IBCFJaccard RS, UBCFJaccard RS, Popular RS, LatentFactor RS, ARConfidence RS, and UBCFImplicative RS. IBCFJaccard RS is created by using item based collaborating filtering model whereas UBCFJaccard RS is created by using user based collaborating filtering model. Both systems use the Jaccard measure to find the nearest neighbors. LatentFactor RS is based on the latent factor model. Popular RS uses the popular model to recommend the most common items to the active users. ARConfidence RS is developed by using the association rule based model and the confidence measure for finding the relevant items. UBCFImplicative RS is developed by using the proposed model and the implicative rating measure.

1) Evaluating the recommendation models on the MovieLens dataset

We conduct 30 times of evaluation (times = 30) where the number of known ratings (i.e. items) of each user in the query set is 14 (given = 14); the number of nearest neighbors is 50 (knn = 50); and the numbers of recommendations to be presented to the active user are: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Each time of evaluation uses 4-fold cross validation method described above. Fig. 2 and 3 show the average F1 values and the average accuracy values of six recommendation models. The results indicate that the F1 values and the accuracy values of the proposed model are higher when compared to other models. However, the difference between the proposed model and the user based collaborating filtering model using the Jaccard measure is small.

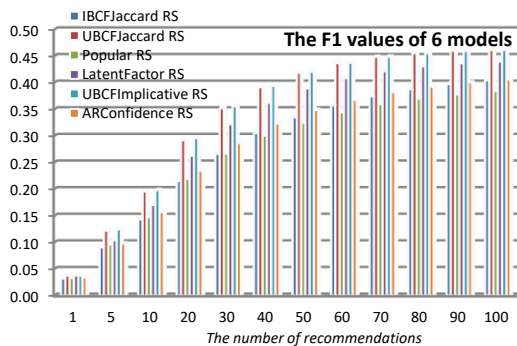


Fig. 2. The F1 values of six recommendation models on the MovieLens dataset.

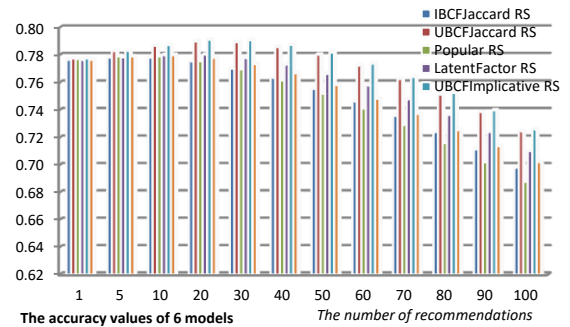


Fig. 3. The accuracy values of six recommendation models on the MovieLens dataset.

Fig. 4 and 5 display the ROC curves and the Precision – Recall curves of the six recommendation models with times = 30, given = 8, and knn = 50. 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; as well as the precision and the recall of the proposed model are higher than those of the other models.

When changing the parameters: the number of times of evaluation (times), number of ratings of each user in the query set (given) from 3 to 14, and the number of nearest neighbors (knn), we get the results similar to the above figures. Therefore, the performance of the proposed recommendation model is better than that of other five models.

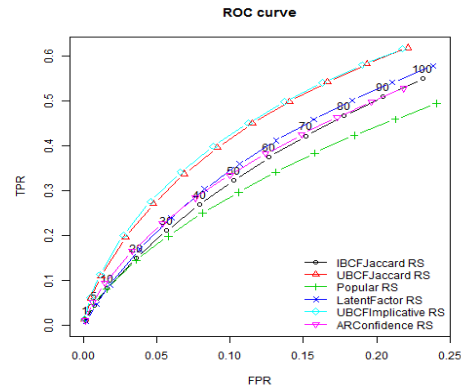


Fig. 4. The ROC curves of six recommender systems on the MovieLens dataset.

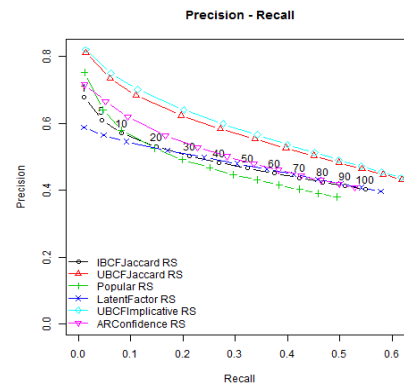


Fig. 5. The Precision – Recall curves of six recommender systems on the MovieLens dataset.

2) Evaluating The Recommendation Models On The CourseRegistration Dataset

Fig. 6 and 7 show the ROC curves and the Precision – Recall curves of the six recommendation models where the number of times of evaluation is 50 (times = 50); the number of known ratings of each user in the query set is 3 (given = 3); the number of nearest neighbors is 30 (knn = 30); and the numbers of courses to be presented to the active student is: 1, 2, 3, 4, and 5. The results indicate that the ROC curve of the proposed model dominates the other ROC curves; the probability of false alarm of the proposed model is lower than that of the other models; and the precision and the recall of the proposed model are higher than those of the other models.

Fig. 8 and 9 display the average F1 values and the average accuracy values of six recommendation models with times = 50, given = 2, and knn = 30. The results show that the F1 values and the accuracy values of the proposed model are higher when compared to other models.

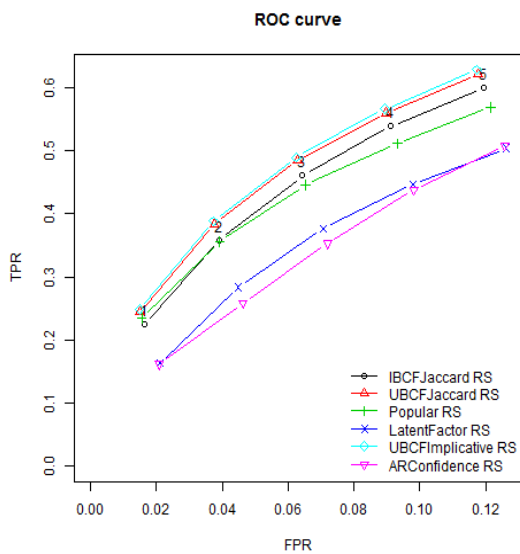


Fig. 6. The ROC curves of six recommender systems on the CourseRegistration dataset.

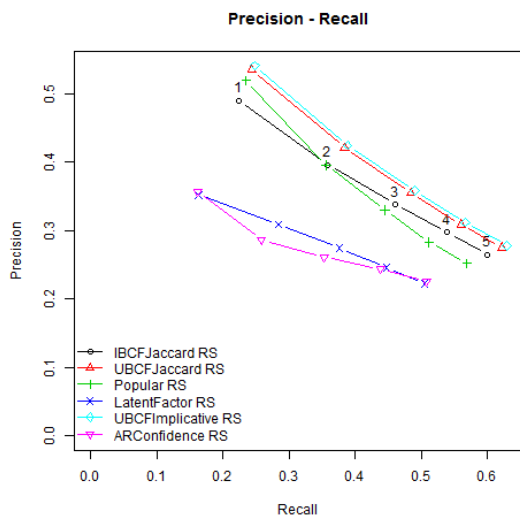


Fig. 7. The Precision – Recall curves of six recommender systems on the CourseRegistration dataset.

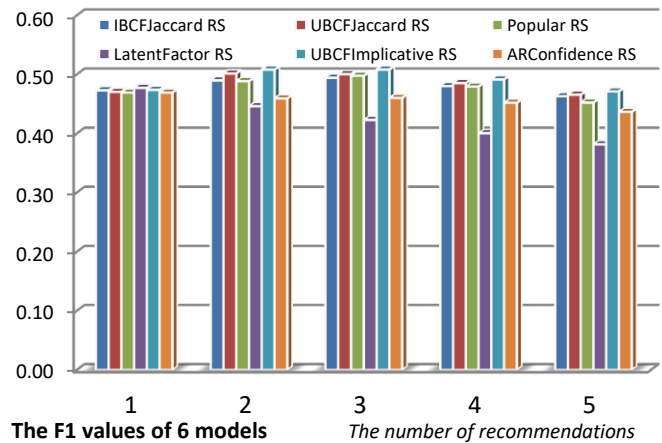


Fig. 8. The F1 values of six recommendation models on the CourseRegistration dataset.

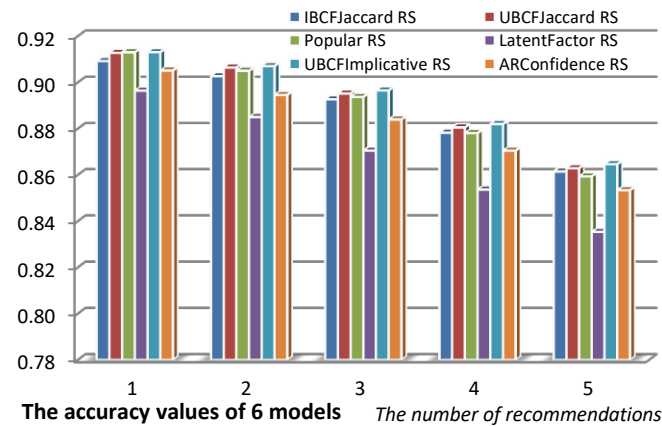


Fig. 9. The accuracy values of six recommendation models on the CourseRegistration dataset.

When the parameters (times, given and knn) are changed, we also get the results similar to the above figures. Therefore, the performance of the proposed recommendation model is better than that of other five models.

V. CONCLUSION

This paper proposes: the implicative rating measure developed on the statistical implicative measures, and a new recommendation model based on the user based collaborative filtering and the proposed measure. The input of the proposed model is the binary rating matrix. In order to filter out, rank, and recommend the top N items to the active users, the proposed model uses the implicative intensity measure to find the nearest neighbors of the active users; the proposed implicative measure to predict users’ ratings for items. The performance of the proposed model is compared to the performance of five existing models of the recommenderlab package: the item based collaborative filtering model using the Jaccard measure, the user based collaborative filtering model using the Jaccard measure, the popular model recommending the most common items to the active users, the latent factor model, and the association rule based model using the confidence measure. The experimental results on the MovieLens dataset and the CourseRegistration dataset show

that the performance of proposed model (through the ROC curves, the Precision – Recall curves, the F1 measure, and the accuracy measure) is better than that of the compared five models.

REFERENCES

- [1] F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, *Recommender Systems Handbook*, Springer US, 2011.
- [2] J. Lu, and et.al, “Recommender system application developments: a survey”, *Decision Support Systems*, 74, pp. 12-32, 2015.
- [3] C. Aggarwal, *Recommender Systems: The Textbook*, Springer International Publishing Switzerland, 2016, ISBN 978-3-319-29657-9.
- [4] X. Amatriain, A. Jaimes, N. Oliver, J. Pujol (2011), “Data mining methods for recommender systems”, *Recommender Systems Handbook*, Springer US, pp. 39-71.
- [5] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *An introduction to recommender systems*, Cambridge University Press, 2011.
- [6] A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, S. Reiterer, and M. Stettinger, “Basic Approaches in Recommendation Systems”, *Recommendation Systems in Software Engineering*, pp. 15-38, Springer-Verlag Berlin Heidelberg, 2014.
- [7] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”, *IEEE Transactions on Knowledge and Data Engineering*, 17, pp. 734-749, 2005.
- [8] J.B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems”, *The Adaptive Web*, LNCS 4321, pp. 291-324, Springer-Verlag Berlin Heidelberg, 2007. 13 – FAIR 2016
- [9] R. Gras and P. Kuntz, “An overview of the Statistical Implicative Analysis (SIA) development”, *Statistical Implicative Analysis, Studies in Computational Intelligence*, Springer-Verlag, 127, 2008, pp. 11-40.
- [10] R. Gras, E. Suzuki, F. Guillet and F. Spagnolo, *Statistical Implicative Analysis*, Springer-Verlag, 2008.
- [11] J. Régnier, Y. Slimani, and R. Gras, *Proceedings of Conference: Analyse Statistique Implicative, des sciences dures aux sciences humaines et sociales*, ASI 8, Tunisie, 2015, ISBN : 978-9973-9819-0-5.
- [12] L.P. Phan, N.Q. Phan, K.M. Nguyen, H.H. Huynh, H.X. Huynh, F. Guillet, “Interestingnesslab: A Framework for Developing and Using Objective Interestingness Measures”. In: Akagi M., Nguyen TT., Vu DT., Phung TN., Huynh VN. (eds) *Advances in Information and Communication Technology. ICTA 2016. Advances in Intelligent Systems and Computing*, vol 538. Springer, Cham, pp. 302-311, 2017
- [13] A. Gunawardana and G. Shani, “A Survey of Accuracy Evaluation Metrics of Recommendation Tasks”, *Journal of Machine Learning Research*, 10, 2009, pp. 2935–2962.