Recommendation using Rule based 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*—The paper presents a rule based implicative rating measure to calculate the ratings of users on items. The paper also presents a new model using the ruleset with the rule length of 2 and the proposed measure to suggest to users the list of items with the highest ratings. The new model is compared to the three existing models that use items (such as the popular items, the items with highest similarities, and the items with strong relationships) to make the suggestion. The experiments on the MSWeb dataset and the MovieLens dataset indicate that the proposed recommendation model has the higher performace (via the Precision - Recall and the ROC curves) than the compared models for most of the given.**

Keywords—Model evaluation; recommendation model; rule based implicative rating measure; ruleset

I. Introduction

Recommendation systems (or recommender systems - RSs) [1] are used to predict the ratings of users for products; and then suggest to users the products that can be preferred by those users. Therefore, RSs help to reduce the information overload; and used in many fields of life [2]. There are various kinds of RSs [2]-[4], such as content based RSs, collaborative filtering RSs, context based RSs, hybrid RSs, etc. RSs always use the measures such as finding the nearest neighbors, identifying the strong relationships among items, etc. to make the suggestion. In recommendation techniques, the association rule based approach can provide the deep explanation on recommendation to users [5]. This approach uses the support and confidence measures and the maximum rule length to find the ruleset and make the suggestion. The recommendation performance (via e.g. the precision, the recall) will be high if the maximum rule length is greater than 2. However, if the length is increased, the number of rules will increase; and consequently, the recommendation time will also increase or the computer may be overloaded. For that reason, in order to maintain the high performance, another measure applied on the ruleset where the length of each rule equals to 2 should be used.

The statistical implicative analysis [6], [7] is applied in knowledge management, bioinformatics, etc. to study the trends (the strong implicative relationships) among data attributes. This method proposed the measures such as the implicative intensity, the typicality to detect those trends. Therefore, the statistical implicative measures can be considered for building RSs. Author in [8] proposed an approach to make a RS using the implicative intensity measure, but did not compare the performance of the approach with that of others. In [9], we proposed a recommendation model based on the important statistical implicative measures and association rules, and then conducted the internal evaluation on the performance of model. Although that model shows the high performance when compared to some existing models, it is still inherited the drawback of the association rule-based approach mentioned above. Besides, there is the lack of consistency in comparing models because we used the random values for a few compared models instead of selecting the best suitable values.

This paper proposes a new measure called as the rule based implicative rating measure to predict the users' ratings and a new recommendation model to present to users the top *N* items (e.g. movies, songs, products, etc.). The new model is based on the rule mining approach and the proposed measure. The model uses the binary rating matrix of as the input; the measures including the confidence measure and the support measure for mining the rules with the rule length of 2, the statistical implicative intensity for calculating the implicative value of each rule, and the proposed measure for predicting the ratings of users on items. The new measure uses the confidence and the statistical implicative values of rules and the items that were rated by those users.

The remaining of paper is organized as follows. Section II proposes not only the rule based implicative rating measure, but also the recommendation model using the ruleset and that measure. Section III presents the experiments. Section IV is the conclusion.

II. Recommendation Model using Rule based Implicative Rating Measure

The proposed recommendation model consists of *n* users $u_1, u_2, \ldots, u_n$ (*U*); *m* items $i_1, i_2, \ldots, i_m$ (*I*); and the ratings of users on items stored in a binary matrix of size *n x m* (*R*). Each cell of matrix r_{jk} is set to 0 if the item i_k is not liked (rated or known) by the user u_j, and 1 otherwise.

Fig. 1 shows the sketch of proposed recommendation model. The model uses the rule mining approach and the proposed measure *ArIR1* (Rule based Implicative Rating). However, the model just uses the ruleset with the rule length of 2 for reducing the recommendation time or avoiding the computers' overload. The model generates the ruleset using the confidence measure and the support measure, calculates the implicative value of each rule using the statistical implicative intensity, creates an item matrix of *m* items x *m* items basing on the confidence and implicative intensity values of rules, predicts the ratings *R'* that can be given by a user who needs the recommendation, and then presents to that user the items whose predicted ratings are highest (the top *N* items). The general objective of proposed recommendation model is that the performance of model (via the Precision-Recall and the ROC curves) is higher when compared to those of some existing models.

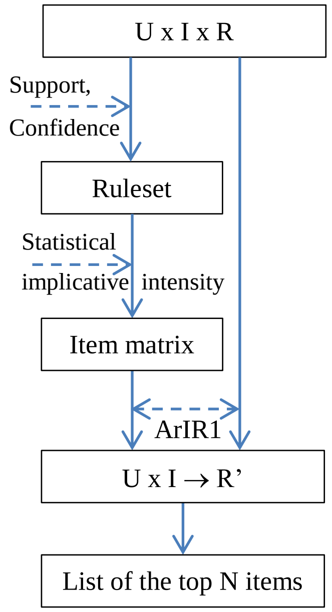

Fig. 1. The recommendation model using the rule based implicative rating measure.

A. Rule based Implicative Rating

The proposed measure *ArIR1* is built on the confidence value and the statistical implicative intensity value of rules, and the items that were rated by the user needing the recommendation. The purpose of *ArIR1* is to predict the rating of a user on an item.

1) Statistical Implicative Intensity

The statistical implicative intensity measure was proposed by Gras [6] to calculate the implicative value between the antecedent and consequent of a rule $a \rightarrow b$. The formula of this measure is shown in (1).

$$\varphi(a,b)=1-\sum_{s=0}^{nab\_}(\lambda^{s}/s!)e^{-\lambda} \text{ if } n_b \neq n; \text{ and } 0 \text{ otherwise} \quad (1)$$

Where n is the number of users; n_a (and n_b) is the number of users rating the item a (and the item b); $n_{ab\_}$ is the number of users who rate the item a but do not rate (or know) the item b; and $\lambda=n_a(1-n_b)/n$.

2) Rule based Implicative Rating

The measure *ArIR1* predicts the final ratings given by a user u_a for an item $i \in I$. The value of $ArIR1(u_a,i)$ is in range [0,1]; and is defined by (2) where $AIR1(u_a,i)$ (Atom Implicative Rating) is the original implicative rating value of u_a for i. The formula of *AIR1* is shown in (3).

$$ArIR1(u_a,i)=AIR1(u_a,i)/max_{l \in I}AIR1(u_a,l) \quad (2)$$

$$AIR1(u_a,i)=\sum_{j=1}^{k}IIntConf(i_j,i) \quad (3)$$

In (3), k is the number of items that were rated (liked) by u_a; i_j is one of k items rated by u_a; and $IIntConf(i_j,i)$ is the combination of two values as shown in (4). $\varphi(i_j,i)$ is the statistical implicative intensity value of the rule $i_j \rightarrow i$; and $conf(i_j,i)$ is the confidence value of the rule $i_j \rightarrow i$. The difference among rules will increase quickly if we use the *IIntConf* instead of φ or *conf*.

$$IIntConf(i_j,i)= \varphi(i_j,i)*conf(i_j,i) \quad (4)$$

The value of $AIR1(u_a,i)$ is calculated from the confidence value and the implicative intensity of rules where the right side of those rules are the same item (i) and the left side are the items rated by u_a.

B. Top N Recommendation

1) Creating a Matrix of Items

A matrix of items *IIntConf* is a m x m square matrix where each element $IIntConf[i_j,i]$ is the value of the rule $i_j \rightarrow i$ according to (4) and i_j as well as i is an item of the itemset I. The algorithm named *CreateItemMatrix* is used for building this matrix. The inputs of algorithm are: the rating matrix R; the thresholds of the support measure and the confidence measure to be s and c respectively. The processing steps of *CreateItemMatrix* are:

- Generating a ruleset by the algorithm *Apriorio* [10] using the thresholds s and c.
- Presenting each rule of ruleset by four values $\{n, n_a, n_b, n_{ab\_}\}$, then calculating the implicative intensity value of that rule according to the algorithm named as *ImplicativeIntensity*. We proposed this algorithm in [11].
- Building a matrix of items *IIntConf* according to (4).

```
CreateItemMatrix(ratingmatrix R; float s,c) {
    Ruleset = Apriorio(R,s,c);
    Ruleset = ImplicativeIntensity(Ruleset,R);
    for each rule $r_k$ of form $i_j \rightarrow i$ belongs to Ruleset {
        IInt[$i_j$,i] = GetImplicativeIntensity(Ruleset,$r_k$)
        Conf[$i_j$,i] = GetConfidence(Ruleset,$r_k$);
        IIntConf[$i_j$,i] = IInt[$i_j$,i] * Conf[$i_j$,i];
    }
    return IIntConf;
}
```

2) Predicting Ratings

The algorithm named as *PredictRatings* is used for predicting the ratings of a user for each item in the itemset I. The inputs of *PredictRatings* are: the vector A of size m, $A[i_j]=0$ if u_a did not rate (or know) the item i_j and 1 otherwise,

u_a is a user needing the recommendation; the square matrix of items *IIntConf*. The steps of *PredictRatings* are as follows:

```
    PredictRatings(vector A; matrix IIntConf) {

    AIR1 = InitializeMatrix(0,1,m);
    ArIR1 = InitializeMatrix(0,1,m);
    for each item $i_j \in I$
        if ($A[i_j]$==1)
            for each item $i \in I$
                AIR1[$u_a$,i] += IIntConf[$i_j$,i];
    ArIR1[$u_a$, ] = AIR1[$u_a$, ]/max(AIR1[$u_a$, ])
    return ArIR1;
}
```

3) Recommendating the Top N Items

The proposed model uses the algorithm named as *IRRS* to present the top *N* items to a user u_a who needs the recommendation. Its inputs are: the vector *A* as mentioned above; the rating matrix *R*; the thresholds of two measures (support, confidence) to be *s* and *c* respectively. *IRRS* conducts the following steps:

- Building a matrix of items *IIntConf* by the algorithm named as *CreateItemMatrix*.
- Predicting the ratings of the user u_a for each item $i \in I$ using the ruleset and the measure *ArIR1* according to the algorithm named as *PredictRatings*.
- Filtering the predicted result by removing *given* items that were rated by u_a.
- Suggesting *N* items (of the filtered list) with the highest predicted implicative ratings to u_a.

```
IRRS(vector A; ratingmatrix R; float s,c){
    IIntConf = CreateItemMatrix(R,s,c);
    ArIR1 = PredictRatings(A,IIntConf);
    Ratings = RemoveKnownGiven(A, ArIR1);
    Reclist = TopNItems(Ratings);
    return Reclist;
}
```

III. Experiments

A. Experimental Setup

1) Experimental Data

MovieLens[1] and MSWeb[2] are the datasets to be used in our experiments. The former dataset stores the ratings of users on movies while the latter dataset stores all Vroots (the areas of the website) visited by users. The number of users, the number of movies or Vroots, and the number of ratings of these two dataset are (943, 1,664, 99,392) and (32,710, 285, 98,653) respectively. For the MovieLens dataset, the value of rating is from 1 to 5. Therefore, the dataset has to be binarized in order to use the proposed recommendation model. A rating is set to 1 if the user rated the movie and 0 otherwise.

To avoid bias because of lack of data, the MovieLens dataset and the MSWeb dataset have to be preprocessed by removing the items rated only a few times as well as the users rating only a few items. The minimum number of items per user and the minimum number of users per item to be selected for extracting these datasets are (50, 100) for MovieLens and (10, 50) for MSWeb. Table I shows the information of these two datasets after preprocessing.

TABLE I. Information of two Datasets after Preprocessing

Dataset	The number of users	The number of items	The number of ratings	The maximum number of given[*]
MovieLens	565	202	41,245	10
MSWeb	875	135	10,487	7

[*]: For evaluating the models, if a user needing the recommendation were rated *t* items (i.e. *t* ratings), then *given* ratings (also called as *given* known ratings) is used for making the recommendations and the remaining ratings (*t-given* ratings) is used for evaluating the recommended result. The maximum number of given is identified by analyzing the percentiles of datasets.

2) Experimental Tool

We developed the recommendation model using rule based implicative measure in the R language, and integrated it in the Interestingnesslab tool [12]. Besides, the three existing models of recommenderlab package [13] are also used for the comparison purpose. Those models are named as IBCF - the item based collaborative filtering model, POPULAR - the popular model, and AR - the association rule based model. The reason for selecting three models is that they all are based on the items to suggest the top *N* items to users. The first one uses the items with highest similarities. The last two ones use the popular items and the items with strong relationships (rules), respectively.

3) Recommendation System Evaluation

We use the evaluation method presented in [14]. In that way, the k-folds cross validation method [15], the Precision - Recall curves and the Receiver Operating Characteristic (ROC) curves [16] are used for evaluating the performance of RSs. The ROC curve is plotted from the values of True Positive Rate (TPR or recall) and False Positive Rate (FPR) while the Precision - Recall curve is plotted from the values of precision and recall. In the experiments of this research, the value of k-folds is set to 4.

B. Experimental Results

We develop four RSs named as: Popular RS, IBCFJaccard RS, ArIR1 RS, and ARConfidence RS.

- ArIR1 RS uses the proposed recommendation model and the rule based implicative rating measure.
- Popular RS suggests to users the top *N* popular items without known items. The POPULAR model is used by this RS.
- IBCFJaccard RS uses the IBCF model and the Jaccard measure. The Jaccard measure is used because it only focuses on matching ones and thus prevents the problem with zeroes. IBCFJaccard RS recommends the top *N* items based on the items with highest similarities.
- ARConfidence RS finds the strong rules (the strong relationships among items) and then suggest to users the

[1] http://www.grouplens.org/node/73
[2] http://www.ics.uci.edu/~mlearn/MLRepository.html

top *N* items. This RS uses the AR model and the confidence measure.

All of these RSs use the following parameters:

- The number of items to be presented to the users (*recs*) is: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 for the MovieLens dataset; and 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12 for the MSWeb dataset.
- The number of known ratings of each user used for making the recommendation (*given*) is: from 1 to 10 for the MovieLens dataset; and from 1 to 7 for the MSWeb dataset.
- The number of times of evaluation (*times*) is varied. For each times of evaluation, the k-folds cross validation method is used.

1) Selected Values of Important Parameters

The important parameters of IBCFJaccard RS and ARConfidence RS are: *k* items to be used for finding the items with highest similarities; and the support value *s*, the confidence value *c*, and the maximum length of rule *l*, respectively. To find the suitable values of these parameters, we vary their values as well as the values of *times*, *given*; then observe the ROC curves and the Precision - Recall curves. *k* is varied from 15 to 202 for the MovieLens dataset, from 20 to 135 for the MSWeb dataset with the difference between two adjacent values to be about 10. To do not omit the association rules with high quality, *s* and *c* should be assigned to small values: from 0.01 and 0.2 respectively for the MovieLens dataset; and from 0.01 and 0.02 respectively for the MSWeb dataset. As a result, Table II shows the selected values of important parameters.

TABLE II. SELECTED VALUES OF IMPORTANT PARAMETERS

Dataset	**k**	**s, c, l**

MovieLens	202 for all givens	s=0.01,c=0.3, and l=3 for given from 1 to 5.
		s=0.04,c=0.5, and l=3 for given from 6 to 10**

MSWeb	135 for all givens	s=0.01, c=0.1, l=7 for all given

**: The larger the value of *l* is, the bigger the size of ruleset will be. The computer used in our experiments is overloaded if the size of ruleset is too big. For example, if s=0.01, c=0.3 and l=3, the size of ruleset is more than 3.5 million rules. Therefore, on the MovieLens dataset, the AR model used the above listed values to compare the models.

The important parameters of ArIR1 RS are the support value *s*, the confidence value *c*. They are used for generating a ruleset where each rule has length of 2. In this paper, *s* and *c* to be selected are equal to ones of ARConfidence RS.

2) Comparison of Recommendation Systems

On the MSWeb dataset, the evaluation of four RSs (Popular RS, IBCFJaccard RS, ArIR1 RS, and ARConfidence RS) using the parameters as mentioned above is conducted.

Fig. 2 displays the ROC curves of four RSs with *times*=10, *given*=4, and the selected values (*recs* from 1 to 12, *k*=135, *s*=0.01, *c*=0.1 and *l*=7) as shown in Table II. The result shows that the ROC curve of RS using the proposed recommendation model is above the other curves; the probability of false alarm (i.e. FPR) of that RS is lowest when compared to that of others for each value of *recs*. When varying the value of *times* and using the selected values of important parameters, we obtain the result similar to Fig. 2 for *given* from 3 to 7.

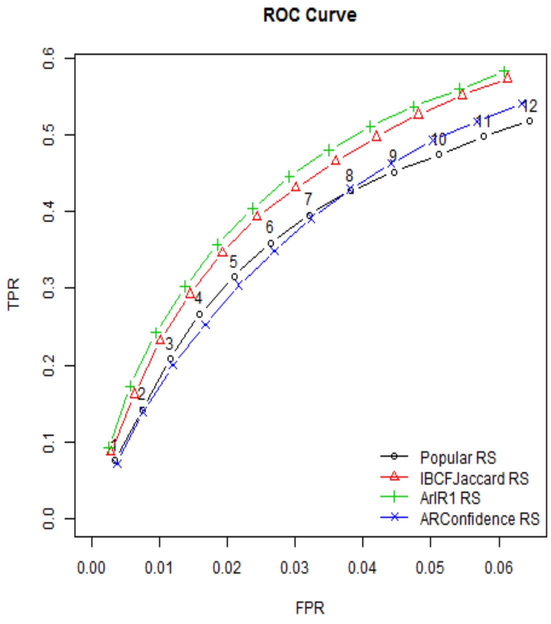

Fig. 2. The ROC curves of four RSs on the MSWeb dataset with times=10, given=4, recs from 1 to 12, k=135, s=0.01, c=0.1, and l=7.

Fig. 3 shows the Precision - Recall curves of four RSs with *times*=2, *given*=2, *recs* from 1 to 12, *k*=135, *s*=0.01, *c*=0.1 and *l*=7. The result indicates that the precision value and the recall value of ArIR1 RS are highest if *recs* is from 1 item to 8 items; and higher than those of Popular RS and IBCFJaccard RS if *recs* is from 9 item to 12 items. For *given*=1, the precision and the recall of ArIR1 RS are just higher than those of IBCFJaccard RS if *recs* is from 1 item to 12 items. When varying the value of *times* and using the selected values, we also achieve the same results as described for *given* less than 3.

On the MovieLens dataset, the performances of four RSs are also compared one another.

Fig. 4 displays the Precision - Recall curves of four RSs with *times*=4, *given*=3, and the selected values shown in Table II (*recs* from 1 to 100, *k*=202, *s*=0.01, *c*=0.3, and *l*=3). The precision value and the recall value of system using the proposed recommendation model are: highest when the number of items to be presented to users (*recs*) is less than 70; and lower than those of IBCFJaccard RS, yet still higher than those of remaining RSs when *recs* is greater than 70. Fig. 4 also shows that the difference between corresponding values of ArIR1 RS and IBCFJaccard RS is quite small for *recs* greater than 30 (the curve portion from 30 items to 100 items). When we vary the value of *times*, use the selected values as shown in Table II, and change *given* from 4 to 10, the achieved result is similar to that obtained for *given*=3. Besides, the length of curve portion with small difference (of ArIR1 RS and IBCFJaccard RS) is increased while *given* is increased.

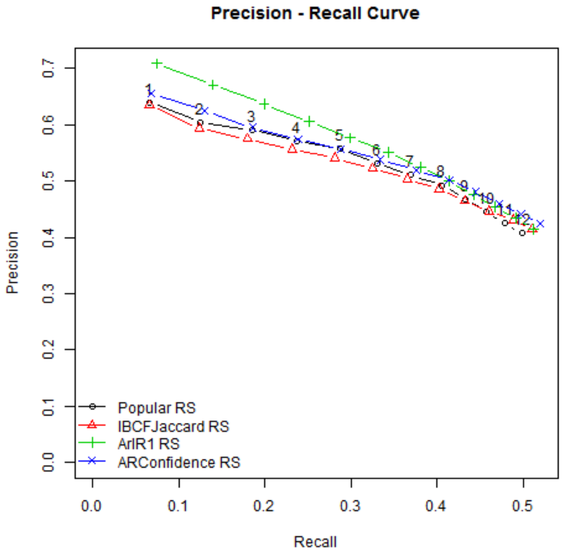

Fig. 3. The Precision - Recall curves of four RSs using the MSWeb dataset, times=2, given=2, recs from 1 to 12, k=135, s=0.01, c=0.1, and l=7.

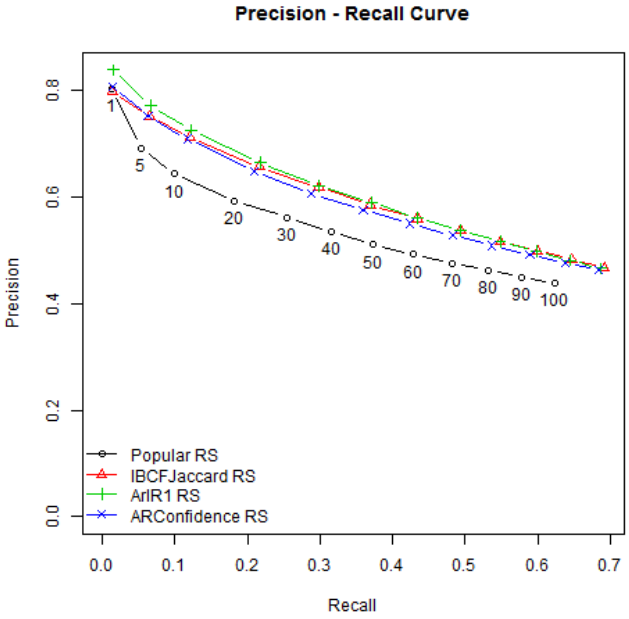

Fig. 4. The Precision - Recall curves of four RSs using the MovieLens dataset, times=4, given=3, recs from 1 to 100, k=202, s=0.01, c=0.3, and l=3.

Fig. 5 shows the ROC curves of four RSs with *times*=2, *given*=1, and the selected values. The ROC curve of ARConfidence RS dominates that of others. The curve of ArIR1 RS is the next one almost dominating the remaining curves. The difference between curves (of ARConfidence RS and ArIR1 RS) is quite small on the portion from 1 item to 30 items. Besides, when observing the values of FPR (the probability of false alarm), we see that the FPR values of ArIR1 RS are almost lowest at all *recs*, especially for the curve portion from 1 to 70 items. We also obtain the result as shown in Fig. 5 when varying the values of *times*.

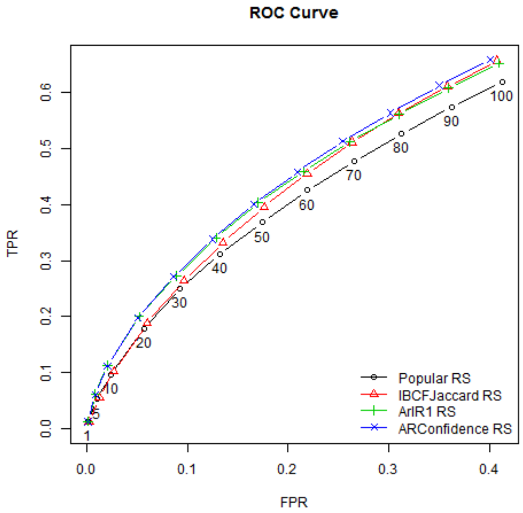

Fig. 5. The ROC curves of four RSs using the MovieLens dataset, times=2, given=1, recs from 1 to 100, k=202, s=0.01, c=0.3, and l=3.

Fig. 6 shows the Precision-Recall curves of four RSs using *times*=5, *given*=2, and the selected values. The figure shows that the precision and recall values of ArIR1 RS are highest on the curve portion from 1 to 70 items; and nearly equal to those of IBCFJaccard RS as well as higher than those of remaining RSs on the portion greater than 70. When varying the values of *times*, the achieved result is similar to that shown in Fig. 6.

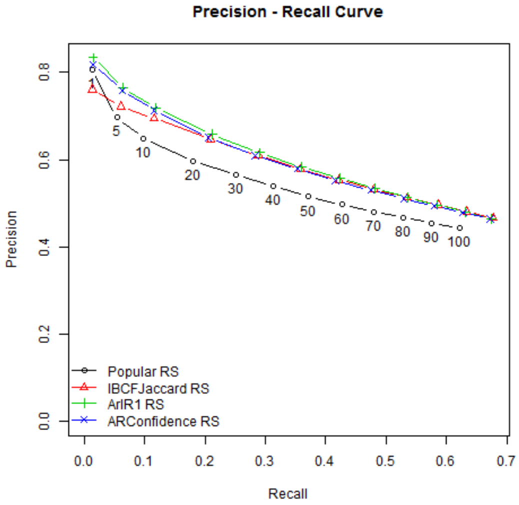

Fig. 6. The ROC curves of four RSs using the MovieLens dataset, times=5, given=2, recs from 1 to 100, k=202, s=0.01, c=0.3, and l=3.

From the experimental results, it is found that: 1) the AR model should only be used for developing RS if *given* equals to 1, its performance is decreased while *given* is increased and the number of generated rules is big if the maximum length of rule is increased; 2) the proposed model should be used for most of *given* (such as: *given* greater than 2 for the MSWeb dataset, and *given* greater than 1 and the number of items to be presented to users less than and equal to 70 for the MovieLens dataset). Besides, either IBCF model or the proposed model can be used for most *given* greater than 1 and the number of items to be presented to users greater than 70 for the MovieLens dataset because the difference between corresponding values of these two models is quite small.

IV. Conclusion

RSs help to reduce the information overload because they can suggest the valuable products to users from the rating matrices. The paper proposes the measure named as *ArIR1* for predicting the users' ratings on items. *ArIR1* is built on the implicative intensity and the confidence value of rules, and the items rated by users needing the recommendation. Besides, the paper also presents the new model based on the ruleset with the rule length of 2 and the proposed measure. This model with the binary rating matrix will suggest to users the top *N* items. The performance of the proposed recommendation model (via the Precision - Recall and ROC curves) is compared with that of some existing models integrated in the recommenderlab package (IBCF, POPULAR, and AR) on two datasets (MovieLens and MSWeb). The results indicate that the performance of proposed recommendation model is higher than that of others for most of *given*.

However, the proposed model has not yet been effective when the number of items that were rated by a user needing the recommendation is small (given=1 or 2 for the MSWeb dataset, and given=1 for the MovieLens dataset). Therefore, in the future direction, we will improve the performance of proposed model by using the hybrid approach such as using the AR model for the small *given* but the creation of a ruleset and a matrix of items will be performed offline; and switching to use the proposed model for the remaining *given*. Moreover, we will also observe the performance of proposed model by varying the threshold value of *IIntConf* to identify the suitable value.

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] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, An introduction to recommender systems, Cambridge University Press, 2011.

[5] 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.

[6] R. Gras, E. Suzuki, F. Guillet and F. Spagnolo, Statistical Implicative Analysis, Springer-Verlag, 2008.

[7] 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.

[8] R. Couturier, "Un système de recommandation basé sur l'A.S.I.", L'Analyse Statistique Implicative. Des Sciences dures aux Sciences Humaines et Sociales, dir. R.Gras, Cépaduès Ed. Toulouse, 2017, pp. 447-452.

[9] L.P. Phan, H.H. Huynh, H.X. Huynh, and R. Gras, "Recommender System based on Important Implicative Measures", In Proceedings of the 9th International Conference A.S.I. Analyse Statistique Implicative, Belfort, France, pp.508-531, 2017.

[10] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207–216, 1993.

[11] L.P. Phan, K.M. Nguyen, H.H. Huynh, and H.X. Huynh, "Association-based recommender system using statistical implicative cohesion measure", In Proceedings of the 8th International Conference on Knowledge and Systems Engineering - KSE, Hanoi, pp. 144-149, 2016.

[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 Advances in Intelligent Systems and Computing, vol 538. Springer, Cham, pp. 302-311, 2017

[13] M. Hahsler, "recommenderlab: A Framework for Developing and Testing Recommendation Algorithms", SMU, 2011.

[14] L.P. Phan, H.H. Huynh, and H.X. Huynh, "User based Recommender Systems using Implicative Rating Measure", International Journal of Advanced Computer Science and Applications, pp. 37-42, 2017.

[15] R.Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection.", In Proceedings of the14th International Joint Conference on Artificial Intelligence, pp.1137-1143, 1995.

[16] A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks", Journal of Machine Learning Research, 10, 2009, pp. 2935–2962.