

Statistical Implicative Similarity Measures for User-based Collaborative Filtering Recommender System

Nghia Quoc Phan
Testing Office,
Travinh University,
Travinh City, Vietnam

Phuong Hoai Dang
Information Technology Faculty,
Danang University of Science and
Technology, Danang City, Vietnam

Hiep Xuan Huynh
College of Information &
Communications Technology, Cantho
University,
Cantho City, Vietnam

Abstract—This paper proposes a new similarity measures for User-based collaborative filtering recommender system. The similarity measures for two users are based on the Implication intensity measures. It is called statistical implicative similarity measures (SIS). This similarity measures is applied to build the experimental framework for User-based collaborative filtering recommender model. The experiments on MovieLense dataset show that the model using our similarity measures has fairly accurate results compared with User-based collaborative filtering model using traditional similarity measures as Pearson correlation, Cosine similarity, and Jaccard.

Keywords—Similarity measures; Implication intensity; User-based collaborative filtering recommender system; statistical implicative similarity measures

I. INTRODUCTION

Currently, recommender system is considered as a useful tool for solving partial information overload of the Internet [3][12]. Its development is always associated with the development of web technologies and machine learning algorithms. Based on the method of collecting and processing the data, the recommender systems can divide into three generations. The first generation of recommender systems uses traditional websites to gather information from three sources: (1) content-based data from the purchase or the use of products and services; (2) demographic data selected from the customer profile; (3) memory-based data collected from the user's preferences. In this generation, the quality of recommendation results is improved based on data classification algorithms and the integration of the data classification algorithms [3][15][27]. The second generation of recommender systems is the increasing use of Web 2.0 by collecting information through social network like Facebook, Zalo and other social networking sites. To satisfy explosive information issue from social networking sites, this generation continues to develop and improve the existing integrated methods and enhance solutions to exploit information from social networks more efficiently such as trust-aware algorithms [20], social adaptive approaches [4], social networks analysis [12][28] and other methods. The third generation of recommender systems is developed in parallel with the web 3.0 with information collected from integrated devices on the Internet such as cameras, sensors [22]. This generation uses approaches to integrate location information into the available recommendation algorithms in order to broaden its application in various fields such as health, weather, environment, and universe [1].

User-based collaborative filtering recommender system is the first version of the recommender systems based on collaborative filtering. It was first introduced in the article "GroupLens: an open architecture for Collaborative filtering of Netnews" in 1994 for GroupLens Usenet recommender system [21]. Subsequently, there are two other recommender systems also use this recommendation method: one for users to listen to music Ringo [25] and the other for users to watch movies Bellcore [26]. User-based collaborative filtering recommender system is a simple algorithm to clarify the core premise of collaborative filtering methods. That is to find out users in the past who had the same behavior as current users. Then, the value rating of users for the items is used to predict the preferences of current users. Thus, in order to obtain a list of items to introduce to new users, User-based collaborative filtering recommender system requires a function to compute the similarity of two users and a method to calculate the average deviation of rating values of similar users based on a rating matrix of users for items [14][15][17].

From the first appearance with the name "The information Lense system" in 1987 [13], recommender system has been developed greatly in technology and its application in the fields of life. In particular, recommender systems are used by many managers as an effective tool in order to support business activities in various fields such as Amazon, Netflix, and Pandora [2]. However, the present generation of recommender systems has not fully met the requirements of users yet. Therefore, research on recommender systems continues to be concerned such as research to improve methods and algorithms to increase accuracy of the existing recommender model [11][18][24], research to improve recommender systems to adapt to the information explosion and research to propose a new recommender model [8]. In addition, some new research directions are also set out, such as research on proper combination of existing recommendation methods that use different types of available information; research on using the maximum capabilities of the sensors and devices on the Internet; research on collecting and integrating information on trends related to habits, consumption and individual tastes of users in the recommendation process; research on ensuring the security conditions and privacy in the entire process of recommendation system; research on proposing the measures for evaluating recommender systems and develop a standard for assessment measures and research on developing a framework for automated analysis on heterogeneous data.

In this paper, a new similarity measures between two users based on Implication intensity measures is proposed for User-based collaborative filtering recommender system. We describe how to build measures and their application in User-based collaborative filtering recommender model. After building the model, the experiments of model was conducted on MovieLense dataset [5] and compared the results with the User-based collaborative filtering recommender model that uses the traditional similarity measures such as Pearson, Cosine, and Jaccard.

This paper has six sections. Section 1 introduces general recommender systems, User-based collaborative filtering recommender system, relevant studies, and addressing the research issue. Section 2 shows how to build a similarity measures between two users based on the statistical implication intensity measures. Section 3 describes the required steps to build User-based collaborative filtering recommender model based on statistical implicative similarity measures. Section 4 presents the evaluation methods of recommender systems. Section 5 presents the experimental results of the model and compares the results with other models. The final section summarizes some importantly achieved results of model using similarity measures between two users based on Implication intensity measures

II. SIMILARITY MEASURES BETWEEN TWO USERS BASED ON IMPLICATION INTENSITY MEASURES

A. Implication intensity measures

Statistical implicative analysis is the method of data analysis studying implicative relationships between variables or data attributes, allowing detecting the asymmetrical rules $A \rightarrow B$ in the form "if A then that almost B" or "consider to what extent that B will meet implication of A" [23].

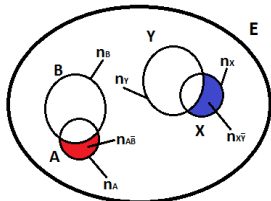


Fig. 1. The model represents a statistical implication rule $A \rightarrow B$

Every rule $A \rightarrow B$ is represented by a table based on the concept of probability called the probability distribution table 2x2 in order to store the counting frequency in the satisfaction of the established conditions. From this probability distribution table, the probability value is calculated based on the frequency of $n, n_A, n_B, n_{AB}, n_{A\bar{B}}$ respectively as follows: $P(A) = \frac{n_A}{n}, P(B) = \frac{n_B}{n}, P(A \cap B) = \frac{n_{AB}}{n}, P(A \cap \bar{B}) = \frac{n_{A\bar{B}}}{n}$.

TABLE I. PROBABILITY DISTRIBUTION TABLE 2X2 OF STATISTICAL IMPLICATION RULE $A \rightarrow B$

		A \rightarrow B		
		B	\bar{B}	
A	n_{AB}	$n_{A\bar{B}}$	n_A	
\bar{A}	$n_{\bar{A}B}$	$n_{\bar{A}\bar{B}}$	$n_{\bar{A}}$	
	n_B	$n_{\bar{B}}$	n	

Interestingness value of Implication intensity measures for rule $A \rightarrow B$ is determined by a formula based on the probability distribution table 2x2 under the following form [23]:

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

$$\text{Where } \lambda = \frac{n_A(n - n_B)}{n}$$

B. Building the similarity measures between two users

In order to calculate the similarity between two users based on Implication intensity measures, the algorithm is proposed consisting of the following steps:

Input: Rating data for items of two users u, v .

Output: Similarity values between two users u, v .

Begin

Step 1: Select the statistical implication rules for two users

- Select the Items that user u rated (I_u);
- Select the Items that user v did not rate (I_v);
- For rule set to be generated from rating matrix of users:

Begin

Select the rules of the form $\{X\} \rightarrow \{Y\}$
where $X \in I_u; Y \in I_v$ and $X \cap Y = \emptyset$;

End;

Step 2: Count the parameters $n, n_A, n_B, n_{A\bar{B}}$

- For each rule in the selected rule set:

Begin

Count the parameters $n, n_A, n_B, n_{A\bar{B}}$;

End;

Step 3: Calculate Implication intensity value for rule set

- For each rule in the selected rule set:

Begin

Calculate the value of Implication intensity measures: $\text{Implicationintensity}(n, n_A, n_B, n_{A\bar{B}})$;

End;

Step 4: Calculate the similarity between two users

$$\text{SIS}(u, v) = 1 - \left(\frac{\sum_{i=1}^k \text{Implicationintensity}_{I_{u,i} \rightarrow I_{v,i}}(n, n_A, n_B, n_{A\bar{B}})}{k} \right)$$

End;

Example: Let us a rating matrix of two users who rated for 4 items as follows:

	i_1	i_2	i_3	i_4
u_1	0	4	4	1
u_2	2	0	4	0

At the first step, selecting statistical implication rules between user u_1 and user u_2 including:

N^0	Statistical implication rules
1	$\{V2=4\} \Rightarrow \{V1=0\}$
2	$\{V4=1\} \Rightarrow \{V1=0\}$
3	$\{V3=4\} \Rightarrow \{V1=0\}$
4	$\{V3=4\} \Rightarrow \{V2=0\}$
5	$\{V3=4\} \Rightarrow \{V4=0\}$
6	$\{V2=4, V4=1\} \Rightarrow \{V1=0\}$
7	$\{V2=4, V3=4\} \Rightarrow \{V1=0\}$
8	$\{V3=4, V4=1\} \Rightarrow \{V1=0\}$
9	$\{V2=4, V3=4, V4=1\} \Rightarrow \{V1=0\}$

At the next step, count the parameters n, n_A, n_B, n_{AB} for each statistical implication rule and calculate implication intensity values based on the parameters:

N^0	Statistical implication rules	n	n_A	n_B	n_{AB}	ImplicationIntensity
1	$\{V2=4\} \Rightarrow \{V1=0\}$	2	1	1	0	0.486582881
2	$\{V4=1\} \Rightarrow \{V1=0\}$	2	1	1	0	0.486582881
3	$\{V3=4\} \Rightarrow \{V1=0\}$	2	2	1	1	0.384940011
4	$\{V3=4\} \Rightarrow \{V2=0\}$	2	2	1	1	0.384940011
5	$\{V3=4\} \Rightarrow \{V4=0\}$	2	2	1	1	0.384940011
6	$\{V2=4, V4=1\} \Rightarrow \{V1=0\}$	2	1	1	0	0.486582881
7	$\{V2=4, V3=4\} \Rightarrow \{V1=0\}$	2	1	1	0	0.486582881
8	$\{V3=4, V4=1\} \Rightarrow \{V1=0\}$	2	1	1	0	0.486582881
9	$\{V2=4, V3=4, V4=1\} \Rightarrow \{V1=0\}$	2	1	1	0	0.486582881

At the final step, the similarity between user u_1 and user u_2 is determined as follows:

$$SIS(u_1, u_2) = 1 - 0.452701924 = 0.54729808.$$

III. USER-BASED COLLABORATIVE FILTERING RECOMMENDER SYSTEM BASED ON SIS MEASURES

User-based collaborative filtering recommender model based on statistical implicative similarity measures is defined as follows:

Suppose that $U = \{u_1, u_2, \dots, u_m\}$ is a set of m users, $I = \{i_1, i_2, \dots, i_n\}$ is a set of n items, $R = \{r_{j,k}\}$ is a rating matrix of m users for n items with each row representing a user u_j ($1 \leq j \leq m$), each column represents an item i_k ($1 \leq k \leq n$), $r_{j,k}$ is the rating value of user u_j for item i_k and $u_a \in U$ is user who needs recommendation.

According to initial data, the model implemented through the following steps:

Step 1: Measure the similarity between users u_a and other users in the system by using function: $SIS(u_a, u_i)$.

Step 2: Determine the list of k similarity users who are similar with u_a : $N(a) \in U$.

Step 3: Identify the item categories that user u_a has not rated yet, determined by: $I_a = I \setminus \{i_k \in I | r_{a,k} \geq 1\}$ and calculate predicted rating values for this item categories by the following formula: $\hat{r}_{a,k} = \frac{1}{\sum_{i \in N(a)} s_{a,i}} \sum_{i \in N(a)} s_{a,i} r_{i,k}$, where $s_{a,i}$ is the similarity value between user u_a and user u_i .

Step 4: Recommend N -items which obtained the highest predicted rating value to user u_a : $T_N \in I_a$.

The model is presented by the following diagram:

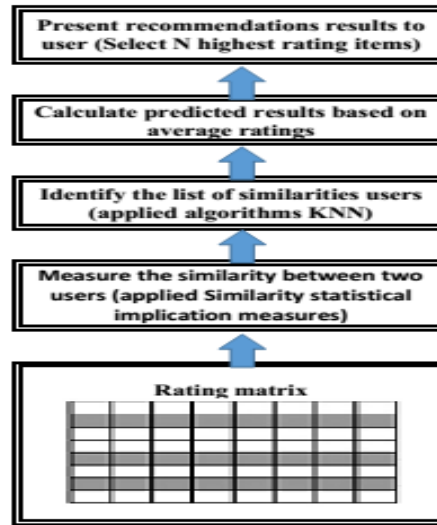


Fig. 2. User-based collaborative filtering recommender model based on statistical implicative similarity measures

IV. EVALUATING THE RECOMMENDER MODEL

The evaluation of the accuracy of the recommender model is an important step in the recommender system design process [6][7][9]. It helps designers choose models, check the accuracy of the model before applying the model into practice. To evaluate User-based collaborative filtering recommender model, the recommender system designers can be conducted through two steps:

A. Preparing the data to evaluate the models

In order to evaluate the quality of a predictive model, experimental datasets have divided into two parts: one for modeling and the rest for testing. Therefore, the first step is to prepare the data; in this step the experimental dataset is divided into two subsets: training dataset and testing dataset [17]. Currently, many methods are being used to split datasets for evaluating recommender models such as:

Splitting: is the initial method to build a training set and test set by cutting experimental dataset into 2 parts [17]. For this method, the model designer should decide the percentage for the training set and test set. For example, the training set accounts for 80 percent and the test set account for the remaining 20 percent.

Bootstrap sampling: is a method used to build a training set and test set by cutting the experimental dataset into 2 parts. However, this approach is done randomly and repeatedly in order that a user may be a member of the training set in this cutting time but is a member of test set in the next cutting time. This can overcome the disadvantages of heterogeneity of the experimental dataset and increase optimization for small-sized dataset. [17].

K-fold cross-validation: is a method used to build a training set and test set by cutting the experimental dataset into

k subsets with the same size (called k-fold). After that, the model is evaluated k times. Every evaluation uses one subset for the test set and the k-1 subsets are used as the training set. The evaluation results of this method are average value of k evaluations. This approach ensures that all users have appeared at least one time in the test set [17]. Therefore, it is the most accurate of the three methods. However, it is costly for the calculation compared with the remaining two methods.

B. Evaluate recommender model

There are two methods for evaluating recommender model: evaluation based on the ratings and evaluation based on the recommendations. The first method evaluates the ratings generated by the model. The remaining method evaluates directly on the recommendations of the model.

Evaluation based on the ratings: a method evaluates the accuracy of the model by comparing the predicted rating value with the real value. More precisely, this method is to find out the average error value based on three indicators RMSE, MSE and MAE. A model is evaluated good if these indicators show low value [7][9].

Root mean square error (RMSE): This is the standard deviation between the real and predicted ratings.

$$RMSE = \sqrt{\frac{\sum_{(i,j) \in \kappa} (r_{ij} - \hat{r}_{ij})^2}{|\kappa|}}$$

Mean squared error (MSE): This is the mean of the squared difference between the real and predicted ratings. It's the square of RMSE, so it contains the same information.

$$MSE = \frac{\sum_{(i,j) \in \kappa} (r_{ij} - \hat{r}_{ij})^2}{|\kappa|}$$

Mean absolute error (MAE): This is the mean of the absolute difference between the real and predicted ratings.

$$MAE = \frac{1}{|\kappa|} \sum_{(i,j) \in \kappa} |r_{ij} - \hat{r}_{ij}|$$

where κ is the set of all user ratings for items; r_{ij} real rating value of user i for item j; \hat{r}_{ij} is predicted rating value of user i for item j.

Evaluation based on the recommendations: a method evaluates the accuracy of the model by comparing the model's recommendations to purchase choice of the users. This approach uses confusion matrix to calculate the value of three indicators: Precision, Recall and F-measure. The model is evaluated good if three indices gain high value [7][9].

TABLE II. CONFUSION MATRIX

User Choices	Recommendations of the model	
	Recommend	Not recommend
Purchase	TP	FN
Not purchase	FP	TN

Let's explain the confusion matrix:

True Positives (TP): These are recommended items that have been purchased.

False Positives (FP): These are recommended items that haven't been purchased.

False Negatives (FN): These are not recommended items that have been purchased.

True Negatives (TN): These are not recommended items that haven't been purchased.

The formula of three indicators is used to evaluate:

$$\text{Precision} = \frac{\text{Correctly recommended items}}{\text{Total recommended items}} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{\text{Correctly recommended items}}{\text{Total useful recommendations}} = \frac{TP}{TP + FN}$$

$$F - \text{measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

V. EXPERIMENT

A. Data description

The experimental dataset is MovieLense [5] of GroupLens research project at the University of Minnesota in 1997. This dataset is collected from the rating results of 943 users for 1.664 movies (99.392 rating results from 0 to 5) through the MovieLense website (movielens.umn.edu) during 7 months (from 09/19/1997 to 22/04/1998). This dataset is organized in a matrix format consisting of 943 rows, 1.664 columns and 1.569.152 cells containing rated values. However, each user is able to watch her/his favourite movies. Thus, the rating matrix has only 99.392 rating values of users for movie categories.

B. Implementation tools

In order to conduct experiment, we use ARQAT tool which is developed on language R by our team. This is a tool package to be developed from engine platform ARQAT on language Java [10]. This tool includes the following functions: processing data, generating statistical implication rules, counting parameters n, n_A, n_B, n_{AB} , calculating value of objective interestingness measures based on 4 statistical implication parameters, calculating similarity of two users based on statistical implicative similarity measures, and designing and evaluation recommender models [16].

C. Select and process data

The MovieLense dataset is stored under a real rating matrix. It consists of 943 rows, 1.664 columns and 1.569.152 cells containing rated value. In particular, more than 93 percent cells have rating values equal 0 and nearly 7 percent remaining cells have rating values from 1 to 5 (value 0 is 1.469.760; value 1 is 6.059 ; value 2 is 11.307; value 3 is 27.002; value 4 is 33.947; value 5 is 21.077). Therefore, the entire MovieLense dataset has only truly 99.392 rating value from users for movies. In particular, the majority of rating values range from 3 to 5 and 4 is rating value with the highest amount. In order to find out the number of users rating for each movie and the number of movies that each user rated, statistical calculations are performed on each movie and each user and illustrate the results in Figure 3.

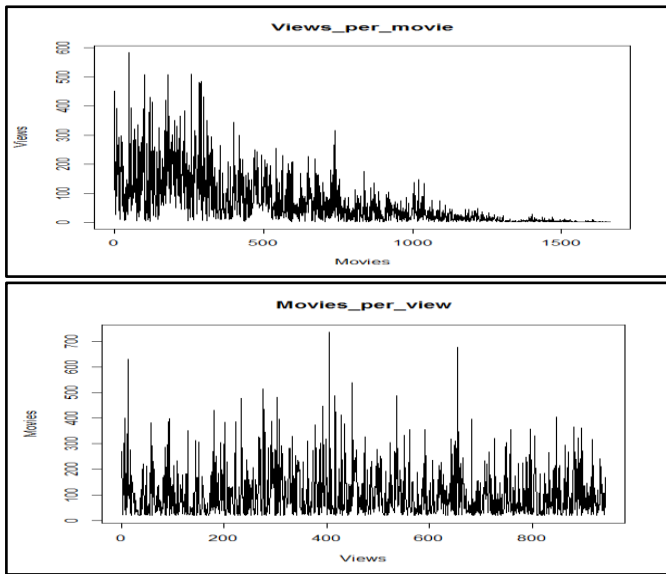


Fig. 3. The chart presents statistical results for each movie and each user on MovieLens dataset

Figure 3 reflects that some movies have only been rated by a few users and some users have only been rated for a few movies. If this case is used for training model, it is likely to lead to bias due to lack of data. Thus, users rate at least for 50 movies and movies rated by at least 100 users are selected to build experimental datasets for model. From there, rating matrix has only 560 rows, 332 columns and 55.298 rating value. In particular, the dataset is split into two subsets: Training set accounting for 80 percent and Test set does the remaining 20 percent.

D. The result of the model

From the result of data processing steps, the model trains on training set with 445 users and tests on test set with 115 users. The result of the model is exported in matrix format with structure 6 x 115 (each column is a user; each cell is a selected movie to recommend for the user in the corresponding column). Figure 4 presents the results of recommender model to the first 4 users; each of them selects the 6 highest rated movies.

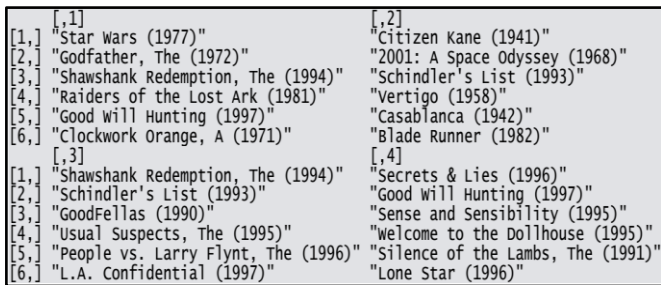


Fig. 4. Presenting recommendation results of the first 4 users

Based on the recommendation result matrix, we calculate the number of times that each movie is recommended and build a histogram for the distribution of movies in Figure 5. The chart shows that the number of movies is recommended from 5 times or less accounting for relatively large numbers. In particular, up to 38 movies are only recommended 1 time and 24 movies are recommended twice. In contrast, the number of movies is recommended from 5 to 40 times accounting for a very small number. Most of them have the number from 1 to 2 movies.

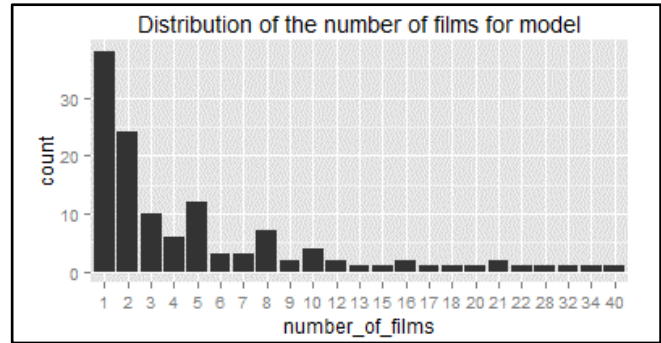


Fig. 5. Presentation distribution of the number of movies for model

E. Evaluation the model

1) Evaluation based on the ratings

In this section, the error parameters (RMSE, MSE, MAE) are calculated for each user and for the model based on the data which is built by k-fold method (with k = 4). For the error parameters of each user, the distribution of each error parameter is performed by a chart and compared them with the error parameters of the model using similarity Pearson measures (Figure 6). The chart shows that the number of users distributed on the error parameters of the model using SIS measures has a higher value than that of the model using similarity Pearson measures. For the error parameters of the model, the value of error parameters is compared with the error parameters of the model using Pearson similarity measures in table 3. The results of comparison found that the values of error parameters of our model are lower than the model using similarity Pearson measures on MovieLens dataset.

TABLE III. PRESENT COMPARISON ERROR PARAMETERS OF TWO MODELS

	RMSE	MSE	MAE
Model using SIS measures	0.9146675	0.8366166	0.7132866
Model using similarity Pearson measures	0.9796664	0.9597462	0.7704055

Model using SIS measures

Model using similarity Pearson measures

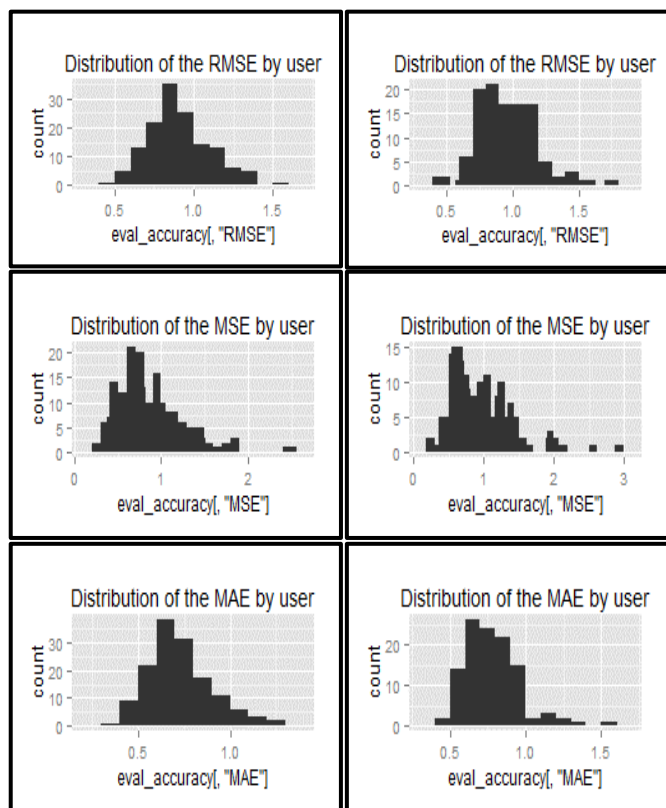


Fig. 6. Comparison of error parameters of each user on two models

2) Evaluation based on the recommendations

In this evaluation method, we also calculate the indicators: TP, FP, FN, TN, Precision, Recall and F-measure based on k-fold data that has been built above. In order to examine the accuracy of the model, the model is tested with the number of movies to be recommended to users which gradually increase (1 to 15). The average rating result of 4 k-fold on models that uses statistical implicative similarity measures and the model using similarity Pearson measures is presented in Figure 7. This figure shows that the indicators Precision, Recall and F-measure on both models are of relatively equal value. This shows that the model that uses statistical implicative similarity measures has the same accuracy as the model that uses similarity Pearson measures on MovieLens dataset.

Model using SIS measures

	TP	FP	FN	TN	precision	recall	f-measure
1	0.7053571	0.2946429	72.43571	243.5643	0.7053571	0.01226171	0.02410440
2	1.3250000	0.6750000	71.81607	243.1839	0.6625000	0.02248717	0.04349790
3	1.9160714	1.0839286	71.22500	242.7750	0.6386905	0.03234240	0.06156713
4	2.4607143	1.5392857	70.68036	242.3196	0.6151786	0.04137372	0.07753297
5	3.0125000	1.9875000	70.12857	241.8714	0.6025000	0.05031998	0.09288254
6	3.5250000	2.4750000	69.61607	241.3839	0.5875000	0.05844200	0.10630885
7	4.0285714	2.9714286	69.11250	240.8875	0.5755102	0.06632461	0.11894179
8	4.4821429	3.5178571	68.65893	240.3411	0.5602679	0.07333510	0.12964115
9	4.9642857	4.0357143	68.17679	239.8232	0.5515873	0.08061004	0.14066328
10	5.4017857	4.5982143	67.73929	239.2607	0.5401786	0.08729768	0.15030477
11	5.8607143	5.1392857	67.28036	238.7196	0.5327922	0.09398769	0.15978786
12	6.3125000	5.6875000	66.82857	238.1714	0.5260417	0.10077316	0.16914367
13	6.7357143	6.2642857	66.40536	237.5946	0.5181319	0.10666154	0.17690566
14	7.1607143	6.8392857	65.98036	237.0196	0.5114796	0.11400974	0.18645771
15	7.5482143	7.4517857	65.59286	236.4071	0.5032143	0.11933871	0.19292476

Model using similarity Pearson measures

	TP	FP	FN	TN	precision	recall	f-measure
1	0.6696429	0.3303571	72.45179	243.5482	0.6696429	0.01107190	0.02178363
2	1.3017857	0.6982143	71.81964	243.1804	0.6508929	0.02151953	0.04166166
3	1.8875000	1.1125000	71.23393	242.7661	0.6291667	0.03130096	0.05963508
4	2.4482143	1.5517857	70.67321	242.3268	0.6120536	0.04036936	0.07574293
5	3.0035714	1.9964286	70.11786	241.8821	0.6007143	0.04897361	0.09056394
6	3.5392857	2.4607143	69.58214	241.4179	0.5898810	0.05732153	0.10448934
7	4.0232143	2.9767857	69.09821	240.9018	0.5747449	0.06465853	0.11624010
8	4.5232143	3.4767857	68.59821	240.4018	0.5654018	0.07248945	0.12850361
9	4.9821429	4.0178571	68.13929	239.8607	0.5535714	0.07909891	0.13841931
10	5.4571429	4.5428571	67.66429	239.3357	0.5457143	0.08603460	0.14863599
11	5.8803571	5.1196429	67.24107	238.7889	0.5345779	0.09229024	0.15740574
12	6.3033571	5.6946429	66.81607	238.1839	0.5254464	0.09832497	0.16565204
13	6.7446429	6.2553571	66.37679	237.6232	0.5188187	0.10567796	0.17559006
14	7.1589286	6.8410714	65.96250	237.0375	0.5113520	0.11138375	0.18292286
15	7.5767857	7.4232143	65.54464	236.4554	0.5051190	0.11765170	0.19085069

Fig. 7. Comparison of indicators based on the recommendations of two models

VI. CONCLUSION

In this paper, we built User-based collaborative filtering recommender model by suggesting a new similarity measures based on Implications intensity measures in order to determine the similarity of two users. Like other User-based collaborative filtering recommender models, our model follows the main steps such as process data, build the rating matrix, compute the similarity between two users, identify the item list that the similarity users rated highly in order to the recommendation results and evaluate accuracy of the model. However, the new point of this model is to identify the similarity user list, using statistical implicative similarity measures instead of using the familiar measures such as Pearson correlation, Cosine similarity, Jaccard to determine the similarity between two users. The experiments show that our model results are relatively accurate on MovieLens dataset. In particular, the error parameters (RMSE, MSE, MAE) have a lower value than the model using the similarity Pearson measure; Indicators of Precision, Recall and F-measure have the equivalent values compared to the model using Pearson similarity measures. This result shows that the User-based collaborative filtering recommender model using the statistical implicative similarity measures is capable to practice.

REFERENCES

- [1] Ali Elkahky, Yang Song and Xiaodong He, "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems," International World Wide Web Conference Committee (IW3C2), WWW 2015, May 18–22, 2015, Florence, Italy, ACM 978-1-4503-3469-3/15/05, 2015.
- [2] Ben Schafer, Joseph Konstan and John Ried, "Recommender Systems in E-Commerce," EC '99 Proceedings of the 1st ACM conference on Electronic commerce, ISBN:1-58113-176-3, 1999, pp.158-166.
- [3] Bobadilla, Ortega, Hernando and Gutiérrez, "Recommender systems survey," Knowledge-Based Systems 46 (2013), 2013, pp.109-132.
- [4] F. Liu and H. J. Lee, "Use of social network information to enhance collaborative filtering performance," Expert Systems with Applications 37(7), 2010, pp.4772-4778.
- [5] F. Maxwell Harper and Joseph A. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems (TiIS) 5, 4, Article 19, 2015, pp.1-19.
- [6] Feng Zhang, TiGong, Victor E. Lee, Gansen Zhao, Chunming Rong and Guangzhi Qu, "Fast algorithms to evaluate collaborative filtering recommender systems," Knowledge-Based Systems 96 (2016), 2016, pp.96-103.
- [7] Gunawardana A, Shani G, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," Journal of Machine Learning Research, 10, 2009, p.2935-2962.

- [8] Hao Wang, Naiyan Wang, and Dit-Yan Yeung, "Collaborative Deep Learning for Recommender Systems," KDD'15, August 10-13, 2015, Sydney, NSW, Australia, 2015 ACM, ISBN 978-1-4503-3664-2, DOI: <http://dx.doi.org/10.1145/2783258.2783273>, 2015, pp.1235-1244.
- [9] Herlocker JL, Konstan JA, Terveen LG, and Riedl JT, "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems, 22(1), ISSN 1046-8188, 2004, pp.5-53.
- [10] Hiep Xuan Huynh, Fabrice Guillet and Henri Briand, "ARQAT: An Exploratory Analysis Tool For Interestingness Measures", in International symposium on Applied Stochastic Models and Data Analysis, 2005, pp.334-344.
- [11] Huizhi Liang and Timothy Baldwin, "A Probabilistic Rating Auto-encoder for Personalized Recommender Systems," CIKM'15, October 19-23, 2015, Melbourne, Australia, 2015 ACM, ISBN 978-1-4503-3794-6, DOI: <http://dx.doi.org/10.1145/2806416.2806633>, 2015, pp.1863-1866.
- [12] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang and Huan Liu, "Recommendation with Social Dimensions," Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2016, pp.251-257.
- [13] Malone TW, Grant KR, Turbak FA, and Brobst SA, "Cohen MD Intelligent information sharing systems," Communications of the ACM, 30(5), ISSN 0001-0782, 1987, pp.390-402.
- [14] Martin P. Robillard, Walid Maalej, Robert J. Walker and Thomas Zimmermann, "Recommendation Systems in Software Engineering," Springer Heidelberg New York Dordrecht London, ISBN 978-3-642-45135-5, 2014.
- [15] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, "Collaborative Filtering Recommender Systems," Foundations and Trends in Human-Computer Interaction Vol. 4, No. 2 (2010), 2010, pp.81-173.
- [16] Michael Hahsler, "Lab for Developing and Testing Recommender Algorithms," Copyright (C) Michael Hahsler (PCA and SVD implementation) (C) Saurabh Bathnagar), <http://R-Forge.R-project.org/projects/recommenderlab/>, 2015.
- [17] Michael Hahsler, "recommenderlab: A Framework for Developing and Testing Recommendation Algorithms," the Intelligent Data Analysis Lab at SMU, <http://lyle.smu.edu/IDA/recommenderlab/>, 2011.
- [18] Mingjie Qian, Liangjie Hong, Yue Shi and Suju Rajan, "Structured Sparse Regression for Recommender Systems," CIKM'15, October 19-23, 2015, Melbourne, VIC, Australia, 2015 ACM, ISBN 978-1-4503-3794-6, DOI: <http://dx.doi.org/10.1145/2806416.2806641>, 2015, pp.1895-1898.
- [19] Nghia Quoc Phan, Hiep Xuan Huynh, Fabrice Guillet and Régis Gras, "Classifying objective interestingness measures based on the tendency of value variation," VIII Colloque International -VIII International Conference, A.S.I. Analyse Statistique Implicative — Statistical Implicative Analysis Radès (Tunisie) - Novembre 2015, Bibliotheeefque Nationale de Tunisie, ISBN: 978-9973-9819-0-5, 2015, pp.143-172.
- [20] P. Bedi, H. Kaur, and S. Marwaha, "Trust based recommender system for semantic web," IJCAI'07 - Proceedings of the 2007 International Joint Conferences on Artificial Intelligence, 2007, pp.2677-2682.
- [21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in ACM CSCW '94, 1994, pp. 175-186.
- [22] Quanjun Chen, Xuan Song, Harutoshi Yamada and Ryosuke Shibasaki, "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference," Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2016, pp.338-344.
- [23] R. Gras and P. Kuntz, "An overview of the Statistical Implicative Analysis (SIA) development," Statistical Implicative Analysis – Studies in Computational Intelligence (Volume 127), Springer-Verlag, 2008, pp.11-40.
- [24] Ting Yuan, Jian Cheng, Xi Zhang, Shuang Qiu, and Hanqing Lu, "Recommendation by Mining Multiple User Behaviors with Group Sparsity," AAAI Publications, Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp.222-228.
- [25] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating "word of mouth," ACM Press/Addison-Wesley Publishing Co., 1995, pp. 210-217.
- [26] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194-201.
- [27] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Hindawi Publishing Corporation, Advances in Artificial Intelligence, Volume 2009, Article ID 421425, doi:10.1155/2009/421425, 2009, pp.1-9.
- [28] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu, "On top-k recommendation using social networks," ACM RecSys'12 - Proceedings of the sixth ACM conference on Recommender systems, 2012, pp.67-74.