International Journal of Hybrid Information Technology
Vol.8, No.1 (2015), pp.163-172
http://dx.doi.org/10.14257/ijhit.2015.8.1.14

Research on the Matthews Correlation Coefficients Metrics of Personalized Recommendation Algorithm Evaluation

Yingbo Liu[1,2], Jiujun Cheng [1,2,*], Chendan Yan[1,2], Xiao Wu[1,2] and Fuzhen Chen[1,2]

[1] Key Laboratory of Embedded System and Service Computing of Ministry of Education, Tongji University, Shanghai 201804, China
[2] Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
yingpooryu@gmail.com; chengjj@tongji.edu.cn; yanchendan@163.com; smfwuxiao@163.com; chenfuzhen16@outlook.com

Abstract

The personalized recommendation systems could better improve the personalized service for network user and alleviate the problem of information overload in the Internet. As we all know, the key point of being a successful recommendation system is the performance of recommendation algorithm. When scholars put forward some new recommendation algorithms, they claim that the new algorithms have been improved in some respects, better than previous algorithm. So we need some evaluation metrics to evaluate the algorithm performance. Due to the scholar didn't fully understand the evaluation mechanism of recommendation algorithms. They mainly emphasized some specific evaluation metrics like Accuracy, Diversity. What's more, the academia did not establish a complete and unified assessment of recommendation algorithms evaluation system which is credibility to do the work of recommendation evaluation. So how to do this work objective and reasonable is still a challengeable task. In this article, we discussed the present evaluation metrics with its respective advantages and disadvantages. Then, we put forward to use the Matthews Correlation Coefficient to evaluate the recommendation algorithm's performance. All this based on an open source projects called mahout which provides a rich set of components to construct the classic recommendation algorithm. The results of the experiments show that the applicability of Matthews correlation coefficient in the relative evaluation work of recommendation algorithm.

Keywords: *recommendation systems; metrics; accuracy; Matthews correlation coefficient*

1. Introduction

With the continuous improvement of the recommendation algorithm, it provides much more personalized services on the ecommerce websites and the social network sites. The recommendation system improves the user experience on different sites and generates enormous economic benefits for the sites.

The emergence and development of recommendation algorithm is experiencing an evolving progress, from the early algorithm of collaborative filtering algorithm [1- 4], hybrid recommendation algorithm [5], to the heat conduction recommendation algorithm [6] and material diffusion recommendation algorithm [7-8] with physics background, also some improved like Matrix Factorization techniques for recommendation algorithm [9] and new theory and technology included. All these algorithms above mentioned have obviously

[*] Corresponding author. E-mail: chengjj@tongji.edu.cn

ISSN: 1738-9968 IJHIT

different performances in different application environments, but all these new proposed or improved are supposed to much better than before. The paper [10] has made a comprehensive summary and analysis of existing recommendation algorithms. As relevant researches go into, some improved algorithms also proposed, its author mostly says the algorithm is much better in some respects than the similar algorithm before. So all this involved how to evaluate the algorithm objective and reasonable. For the already proposed evaluation metrics, for example, Accuracy [11-13], Recall [14], Diversity [15-16], Novelty [17-18] and so on. Many scholars mainly emphasized some certain kinds of evaluation metrics to evaluate the performance of their own proposed new recommendation algorithm. But the fact may be that when one algorithm has better accuracy, but worse performance in the respects of recall, the diversity of recommendation list and so on, All this leads to the recommendation algorithm evaluation chaos and the lack of a unified evaluation system. So, it's still having a long way to go to establish a comprehensive objective recommendation algorithm measures. A unified evaluation mechanism will be very beneficial for the recommendation algorithm research.

The classification Accuracy measures the ability of recommendation system can correctly predict the preferences of the user to the recommended items. The Accuracy metric suits for the recommendation system with user binary classification preferences. If the use preference is not binary classification, we can determine the preferences threshold to convert it. At present, the commonly used indicators are Accuracy, Recall, F1 index [14] and AUC index [16, 19-20]. But the Accuracy metric cannot very well reflect the recommendation system performance; because the Accuracy of recommendation system will be affected by the sparsity of item rating score data in dataset. The data sparsity means only a few items have been rated by user. The problem of Recall index is that when in practical application there is no rated item data in the early stage; the recommendation cannot accurately know whether the user like it or not. Then scholars have put forward the Precision and Recall rate compared with the consequent of random recommended. But the most important is that the Precision and Recall cannot be used alone, they often are negatively correlated and dependent on the length of the recommended list. So we got the F1 index which is the result of combined with Precision and Recall, and these indicators have also been used in the evaluation of information retrieval results. But all these three indicators are not suitable for evaluating the recommendation system without user binary preference classification, so for better evaluation of the recommendation algorithm performance, we can use the AUC index to evaluate the accuracy of recommended result. AUC index, that is the area under the ROC curve [20], and the ROC drew is based on a series of different binary classification. The advantage of AUC indicator is that it cannot be effected by the length of the recommended list and the users preference threshold, use a numerical value can reflect the performance of the recommendation algorithm, so the ROC curve are widely used in the recommendation system and the evaluation of information retrieval. But the main drawback of the ROC curve is the complicated drawing step; it needs to analyze all of items each user may interest. Due to only consider the area under the curve, when the recommended result of a certain period of recommended list is correct, the impact to the curve area that the result item location of recommended list is none, and it also did not consider the specific items effects of ranking position, finally in order to guarantee the correctness of the result of evaluation, it often requires a large magnitude of data to better identify the area among different curves.

In this article, we proposed the new thoughts that using the Matthews correlation coefficient [21] to evaluate the performance of recommendation algorithm. We call this metric as Matthews correlation coefficient precision (MCCP) evaluation metric. We also put forward a way how to set the user preference threshold value to better classify the no-binary classification of user preference in the recommendation system. Then we use the open source project called Mahout to construct the classic collaborative filtering algorithm to do the relative experiments. The data set we used is called MoiveLens dataset. At last, we validated the applicability of the MCCP evaluation metric by relative experiments and discussed it

2. Related Work

When the evaluation work of the personalized recommendation system mostly focused on the item recommended by recommendation algorithm whether the user like or not. The classification accuracy evaluation metric can better measure it. The most common classification accuracy evaluation metrics are Precision, Recall and F1 index.

The Precision metric is defined as the number of recommended item user like in the accounts for the proportion of all recommended items:

$$Precision = \frac{\left|\{N_{rs}\} \cap \{N_s\}\right|}{\left|\{N_s\}\right|} \tag{1}$$

Where, *Nrs* is the number of recommended item that user like. *Ns* represent the total number of item that the recommendation algorithm recommended them to user, which value equals to the length of recommended list. If the user likes all the items in the recommended list, then the value of precision metric will be 1.0.

The Recall metric is defined as the proportion of the number of recommended items which user like in the recommended list and the number of items user like in the item set of recommendation system:

$$Recall = \frac{\left|\{N_{rs}\} \cap \{N_s\}\right|}{\left|\{N\}\right|} \tag{2}$$

Where, the *N* represents the number of user like items in the item set of recommendation system. The other two symbols are the same with Precision metric. The Recall metric means the probability of user favorite item can be recommended to user.

Due to the Precision and Recall are negatively correlated and influenced by the length of recommended list. If we only use one of them to do the evaluation work, the evaluation of recommendation algorithm is not so meaningful. We should consider the precision at a certain level of recall, so we get the F1 index which combined with Precision and Recall to evaluate the performance of recommendation algorithm. The equation of F1 index is defined as:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2\left|\{N_{rs}\} \cap \{N_s\}\right|}{\left|\{N_s\}\right| \cup \left|\{N\}\right|} \tag{3}$$

The symbols in the equation are the same meaning with them in the equation of Precision metric and Recall metric. But the F1 index gives the composite evaluation result of Precision metric and Recall metric while evaluating the recommendation performance. The bigger value of F1 index gets, the better performance of personalized recommendation algorithm it is.

3. MCCP Evaluation Metric

3.1. The Definition of MCCP

When the user preference of recommendation system is no binary classification, the user preference still can be converted to binary classification by setting the preference threshold.

For example, if the users give the number of starts to indicate the degree of preference while rating the movie, we can set the threshold number of stars to convert user's preference to user like and dislike two cases.

For the five star scoring systems [22], we can generally regard the 3-5 stars as the user like it, while the 1-2 stars means the user do not like it. But such a conversion exits drawback. Due to the user rating score are subjective and also affect by the context environment, if for a certain user, the 4 stars is the base line that means user like it, so there is a certain degree of conversion errors. In order to eliminate the user's subjectivity and better convert user preference to binary classification preference. We can use the Gaussian distribution to count the history rating score of user.

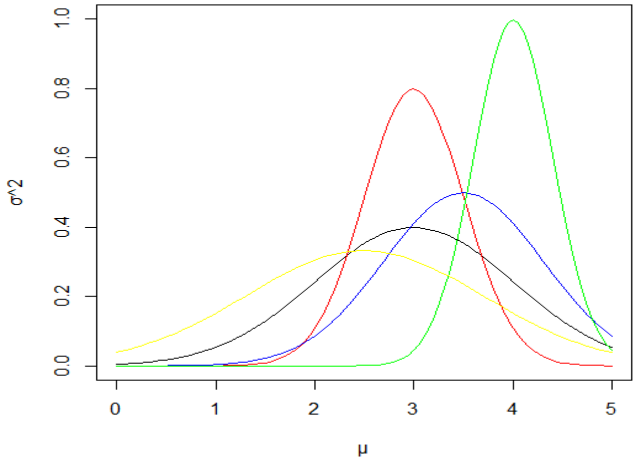


Figure 1. The Distribution of User Rating Score in MovieLens

In the Gaussian distribution *N(μ, σ2)* rating score, the symbol describes the central tendency of user rating scores and the symbol σ describes the dispersion degree of user rating scores:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(- \frac{(x-\mu)^2}{2\sigma^2} \right) \tag{4}$$

We can set the preference threshold value be the value of μ. We can better convert the user preference to the binary classification of user preference by the dynamic value μ, which the value is computed from the user historical rating score date. With the better result of binary classification, we can get better recommendation result.

While evaluating the performance of personalized recommendation system, the ROC curve only considers the situation of true-positive and false-positive. The exact meaning of true-positive in recommendation system is that the recommended item is actually user like it. If the recommendation system recommends item to user, but the user does not like it. We call this situation as the false-positive. But the ROC curve leaves out the other two situations true-negative and false-negative. The true-negative represents that the recommendation system did not recommend the item to user, and user does not like it too. The false-negative means that the recommendation system did not recommend the item but user like it. The F1 index only considers three situations of them, but only leaves out the true-negative one. It should be mentioned at here is that the mistaken of false-positive will worsen the user experience, and lose user in the real application environment.

So we will take all the four situations of the recommendation result into account and use the Matthews correlation coefficient to evaluate the classification accuracy of recommended result. It considered the true, false positive and negative four situations, so it can be regarded as a balanced evaluation index. The Matthews correlation coefficient abbreviated as MCC is essentially the correlation coefficient between the observed and the predicted in the binary classification; it will return a value range in -1 to 1. And the value of correlation coefficient 1 represents completely correct predictions in the recommend list that user likes all the recommended items; the value of -1 represents a completely opposite prediction. In the process of Matthews correlation coefficient computation, we can use the confusion matrix to get the relative value of recommended result classification. The confusion matrix of recommendation system will be showed below:

Table 1. Confusion Matrix of Recommendation Results

preference	recommended	Not recommended
Like	True-Positive TP	True-Negative TN
Not like	False-Positive FP	False-Negative FN

The formula of Matthews correlation coefficient defines as:

$$MCCP = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

The Precision metric, Recall metric and F1 index will be given again according to the confusion matrix.

The equitation of Precision metric is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The Recall metric is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

The definition of F1 index is:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{8}$$

In the equation, the *TP* is the number of true position, namely the quantity of items in recommended list which user likes. The *TN* is the number of true negatives, which means the quantity of items user likes but the recommendation system did not recommend. The *FP* is the number of false positives equals to the quantity of items user doesn't like but be recommended by recommendation system. The *FN* is the number of false negatives, namely the quantity of items both user doesn't like and also not be recommended by recommendation system. The value of *TP+FP* in the denominator is the length of recommended list.

3.2. The Algorithm Analysis

In this section, the algorithm of Matthews Correlation Coefficient Precision (MCCP) evaluation metric will be analyzed here. The mainly process of this algorithm is to get the relative statistic data of those four situations in the confusion matrix. The pseudo code of MCCP algorithm shows below:

```
Algorithm: The MCCP evaluation metric algorithm
Input: DataModel: data source; UserID: user id; Length: the length of recommended list
       length; EvaluationPercentage: percent of test data set; RelevanceThreshold: user
       preference threshold; RecommenderBuilder: recommendation algorithm constructor
Output: value of MCCP
Step1:  Let $TP = 0, FP = 0, FN = 0, TN = 0$;
Step2:  Generate recommendation model by function recommender (RecommenderBuilder,
        DataModel, RelevanceThreshold, EvaluationPercentage);
Step3:  Get recommendedItemsList by fuction recommender.recommend( UserID, Length);
Step4:  Treverse the recommendedItemsList, if the user liked item rating score value >
        RelevanceThreshold, then TP++;
Step5:  FP = recommendedItemsList.size() - TP;
Step6:  Traverse all the items in the DataModel, if rating score < RelevanceThreshold, then
        FN++;
Step7:  Get all the items of UserID in the DataModel, traverse the recommended list to
        filter out the recommended item in the recommendItemsList, then get the set of
        userSelectItems, get the TN by using userSelectItems.size()-FN;
Step8:  TNPlusFP = TN+FP;
Step9:  TNPlusFN = TN+FN;
Step10: TPPlusFP = TP+FP;
Step11: TPPlusFN = TP+FN;
Step12: Get the MCCP value of UserID according to the equation (5);
```

According to the above pseudo code , we can find that the main factors influent the time complexity of MCCP evaluation algorithm are the size of item set and the length of recommended list in the recommendation system. Let the length of recommended list is $|L|$, the number of user in recommendation system is $|M|$ and the size of item set in recommendation system is $|N|$ .the length of recommended list is also equal to $|TP+FP|$. So the time complexity of step4 is $O(|TP|)$, the time complexity of step6 is $O(|FN|)$ and the time complexity of step7 is $O(|N|\times|FP|)$. We can get the total time complexity of the MCCP evaluation metric algorithm is $O(|M|\times (|TP|+|FN|+|N|\times|FP|)) = O(|M|\times |N|\times|FP|)$

4. Experimental Evaluation

4.1. Experimental Environment

The relative experiments of the Matthews correlation coefficient evaluation metric algorithm are based on the open source project called Mahout to construct the classic collaborative filtering algorithm and also an ideal simple lightweight SlopOne included. The MovieLens data set is used to complete the relative work.

4.1.1. The Recommendation Algorithm: The Apache Mahout Project is an open source project of Apache Software Foundation (ASF), mainly to build scalable classic machine learning algorithms: Clustering, Collaborative Filtering, Classification and Frequent Pattern Mining. With the sup-port of Apache Hadoop library, Mahout can spread to cloud computing environment effectively. This provides quiet a good solution to process large data.

Mahout provides many functions, especially in clustering and collaborative filtering. The Taste contains conventional user based and item based collaborative filtering recommenders. It also includes SlopOne, a new and efficient approach, some experimental, preliminary implementations based on the singular value decomposition (SVD) and more. All this help us to construct our personal recommender engine.

4.1.2. The Data Set: The dataset of this experiment used mainly is the MovieLens dataset. MovieLens has collected and made available rating data sets from users' film preferences. When new user registration, users need to complete 15 movies rated task. The principle of rating is that the score ranging from 1 to 5 rated 0.5. The MovieLens contains three size of dataset: MovieLens 100k-consists of 100,000 ratings from 943 users on 1682 movies, MovieLens 1M-consists of 1 million ratings from 6040 users on 3900 movies and MovieLens 10M- consists of 10 million ratings from 71567 users on 10681 movies.

4.2. Experiment Results

For comparing the pros and cons of the User-based collaborative filtering algorithm, and Item-based collaborative filtering algorithm and the lightweight SlopOne algorithm by the MCCP evaluation metric, we will use the MovieLens dataset which 6040 users' approximately 100 million rating records included and the method of 10-fold cross validation [23] to conduct this experiment. The test dataset accounted for 10% of the whole MovieLens dataset. By changing the length of recommended list, we can get the following figure:

The Figure 2 shows that User-based collaborative filtering algorithm, SlopOne algorithm and Item-based collaborative filtering algorithm exists quite a big difference. The first two kinds of algorithm are better than the last Item-based collaborative filtering algorithm. In order to validate MCC and F1 index to recommend system algorithm evaluation results have unity, Now based on MovieLens data set with 943 users, 100,000 rating record dataset, recommendation system preferences threshold set to 3.0 points, the test set of the whole data set 10%, the optimal value of nearest neighborhood value is 85 based on the user based collaborative filtering algorithm.

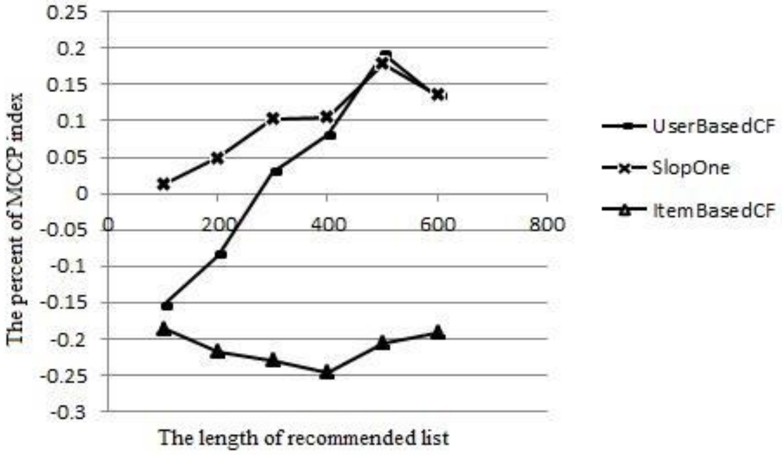
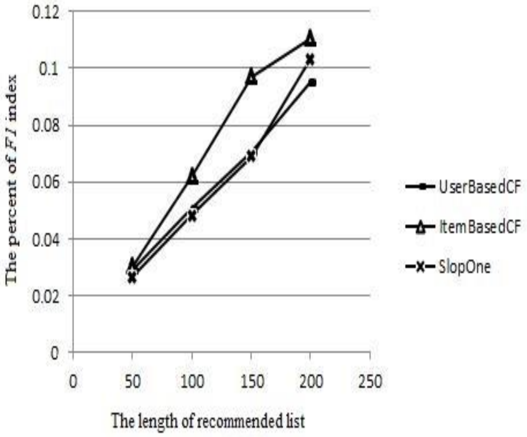


Figure 2. The Accuracy Metric of Recommendation Algorithm

In keeping the above parameters same with the F1 index experiment, we also use the 10-fold cross validation method to process the data set, and then get Figure 3 and figure 4 by changing the length of the recommended list, the difference between the Figures 2 is the size of used dataset is different. This leads to each figure of the abscissa that the change scope of recommended list length is different.



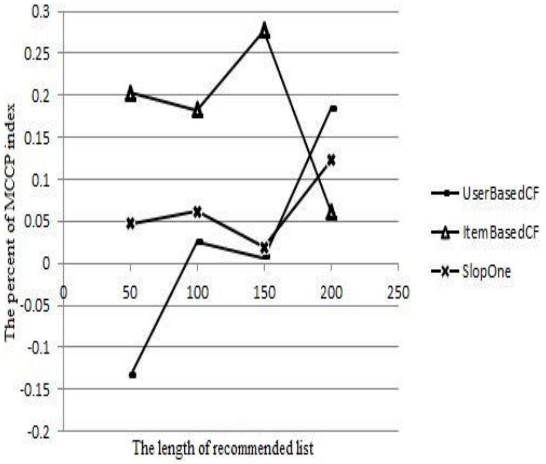


Figure 3. The F1 Index Metric Result

Figure 4. The MCCP Index Metric Result

From the Figure 3, we can get Item-based collaborative filtering recommendation algorithm is better than the other two kinds of algorithm. The performance of User-based collaborative filtering algorithm and SlopOne is about the same. Taking into account the precision and recall for F1 indicators makes it have a wide range of applications in the retrieval results of the evaluation of information retrieval.

From the Figure 4, we can get that Item-based collaborative filtering recommendation algorithm is better than the other two kinds of algorithm. The deviation of the data points is caused by the amount of data limitations of the dataset, the specific reason remains to study. But we can see the conclusion of the MCCP indicator and conclusion obtained through F1 indicators is the same. This proves that the MCCP metric is applicable to user preference recommendation system.

Matthews correlation coefficient obvious draw-back is its applicability, It is mainly used to evaluate a recommendation system has binary preferences classification; to no significant preferences division rating recommended system is not very adequate. What's more, the range of correlation coefficient is between -1 to 1, which is different from the usually evaluation index range in 0 to 1. So in the assessment of recommendation system, to select the most suitable evaluation metric, fully understand the suit-ability of different evaluation metric according to the application context of recommendation algorithm and construct the rational evaluation metric system, all these will help to better complete recommendation system evaluation work

5. Conclusion

In this article, we discussed the advantages and dis-advantages of existing recommendation algorithm evaluation metrics, put forward and discussed a new classification accuracy evaluation metric which called Matthews correlation coefficient Evaluation metric. The experimental results show that the new accuracy evaluation metric has the same experimental result with original and widely used F1 indicators based on the same dataset. It proves the applicability of Matthews correlation coefficient in the relative evaluation work of recommendation algorithm. All the relative work helps the researcher have more options to choose the best metric to reflect the superiority of their proposed recommendation algorithms.

Acknowledgements

This work was supported in part by National 863 Programs (under grant 2012AA062800), by ISTCP (under grant 2013FM10100), by Natural Science Foundation Programs of Shanghai (under grant 13ZR1443100), by the National Science and Technology Support Plan (under grant 2012BAH15F03), and by NSFC (under grant 51034003, 51174210).

References

[1] R. R. Liu , C. X. Jia and T. Zhou, Personal recommendation via modified collaborative filtering, J. Physica A, vol. 388, **(2009)**, pp. 462-468.

[2] J. A. Konstan, B. N. Miller and D. Maltz, "GroupLens: applying collaborative filtering to usenet news", J. Comm ACM, vol. 40, **(1997)**, pp. 77-87.

[3] G. Linden, B. Smith and J. York, Amazon.com recommendations: item-to-item collaborative filtering, J. IEEE Internet Computing, vol. 7, **(2003)**, pp. 76-80.

[4] M. Balabanovic and Y. Shoham, Fab: content-based, collaborative recommendation, J. Comm ACM, vol. 40, **(1997)**, pp. 66-72.

[5] K. Yoshii, M. Goto and K. Komatani, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model", J. IEEE Transactions on Audio speech and Language Processing, vol. 16, **(2008)**, pp. 435-447.

[6] Y. C. Zhang, M. Blattner and Y. K. Yu, Heat conduction process on community networks as a recommendation model ,J. Phys. Rev. Lett., vol. 99, **(2007)**, pp. 154301 -1543054.

[7] Y. C. Zhang, M. Medo and J. Ren, "Recommendation model based on opinion diffusion", J. Europhys. Letter, vol. 80, **(2007)**, pp. 68003-68007.

[8] Z. K. Zhang, T. Zhou and Y. C. Zhang, "Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs", J. Physica A: Statistical Mechanics and its Applications, vol. 389, **(2010)**, pp. 179-186.

[9] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems", Computer, vol. 42, **(2009)**, pp. 30-37.

[10] J. G. Liu , T. Zhou and B. H. Wang, "Progress of the personalized recommendation systems", J. Progress of Nature and Science, vol. 19, **(2009)**, pp. 1-15.

[11] D. Billsus and M. J. Pazzani, "Learning collaborative information filters", Proceedings of the 15th National Conference on Artifical Intelligence, **(1998),** Menlo Park, California.

[12] C. Basu, H. Hirsh and W. W. Cohen, "Recommendation as classification: using social and content-based information in recommendation", Proceedings of the 15th National Conference on Artificial Intelligence, **(1998),** Menlo Park, California.

[13] B. M. Sarwar, G. Karypis and J. A. Konstan, "Analysis of recommendations for e-commerce", Proceedings of the 2nd ACM Conference on Electronic Commerce, **(2000),** New York.

[14] M. Pazzani and D. Billsus, "Learning and revising user profiles the identification of interesting web sites", J. Machine Learning, vol. 27, **(1997)**, pp. 313-331.

[15] T. Zhou, L. Jiang and R. Q. Su, "Effect of initial configuration on network-based recommendation", J. Europhysics. Letters, vol. 81, **(2008)**, pp. 58004-58007.

[16] J. L. Herlocker, J. A. Konstan and L. G. Terveen, "Evaluating collaborative filtering recommender systems", J. ACM Transactions on Information Systems, vol. 22, **(2004)**, pp. 5-53.

[17] K. Swearingen and R. Sinha, "Beyond algorithms: an HCI perspective on recommender systems", J. ACM SIGIR 2001 Workshop on Recommender Systems, vol. 13, **(2001)**, pp. 393-408.

[18] B. M. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms", Proceedings of the 10th international conference on World Wide Web, ACM, **(2001)**, New York.

[19] J. A. Swets, "Effectiveness of information retrieval methods", j. AmerDoc, vol. 20, **(1969)**, pp. 72-89.

[20] J. A Hanley and B. J. Mcneil, "The meaning and use of the area under a receiver operation characteristic (ROC) curve", J. Radiology, vol. 143, **(1982)**, pp. 29-36.

[21] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme", J. Biochim Biophys Acta, vol. 405, **(1975)**, pp. 442-451.

[22] B. J. Dahlen, J. A. Konstan, J. L. Herlocker, N. Good, A. Borchers and J. Riedl, "Jump-starting movielens: User benefits of starting a collaborative filtering system with "dead data", J. (**1998**), pp. 98:017.

[23] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection IJCAI", **v**ol. 14, **(1995)**, pp. 1137-1145.