ANADOLU ÜNİVERSİTESİ BİLİM VE TEKNOLOJİ DERGİSİ –A
Uygulamalı Bilimler ve Mühendislik

ANADOLU UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY –A
Applied Sciences and Engineering
Cilt/Vol.: 14-Sayı/No: 1 : 55-65 (2013)

ARAŞTIRMA MAKALESİ /**RESEARCH ARTICLE**

EFFECTS OF BINARY SIMILARITY MEASURES ON TOP-*N* RECOMMENDATIONS

Edip ŞENYÜREK [1], Hüseyin POLAT[2],

ABSTRACT

Shopping over the Internet through several e-commerce sites is receiving increasing attention. Customers want to purchase those products that they might like without wasting time and/or money. To help their customers, many online companies provide top-*N* recommendations by means of recommender systems. Similarity measures used to find out the most similar entities might affect the overall performance of top-*N* predictions. Although there are various binary ratings-based similarity metrics, their effects on accuracy and online efficiency of top-*N* recommendations have not been deeply studied.

In this study, we investigate seven well-known binary ratings-based similarity metrics in terms of both preciseness and efficiency while providing top-N recommendations. To compare them with respect to accuracy and competence, we perform several experiments based on two well-known real data sets. We modify top-N recommendation algorithm in such a way so that the most similar users' data are involved in recommendation process. We also study how varying controlling parameters affect overall performance with different similarity metrics. We analyze our empirical results and provide some suggestions.

Keywords: Similarity metric, Top-*N* recommendation, Accuracy, Online efficiency.

İKİLİ BENZERLİK METRİKLERİNİN ÜST-N ÖNERİLERİNE ETKİSİ

ÖZ

İnternet üzerinden sanal firmalar aracılığıyla alışveriş yapmak artan ilgi görmektedir. Müşteriler beğenebilecekleri ürünleri zaman ve/veya paralarını boşa harcamadan satın almak isterler. Müşterilerine bu süreçte yardımcı olmak için birçok sanal şirket öneri sistemlerinden yararlanıp müşterilerine en-iyi-*N* önerileri sunmaktadır. En benzer varlıkları belirlemede kullanılan benzerlik ölçütleri en-iyi-*N* önerileri hizmetinin genel performansını etkileyebilir. İkili değerler üzerinde işlem yapan birçok benzerlik ölçütü bulunmasına rağmen bunların en-iyi-*N* önerilerinin doğruluğu ve çevrimiçi performansı üzerindeki etkisi detaylı biçimde çalışılmamıştır.

Bu çalışmada iyi bilinen yedi adet ikili oy-tabanlı benzerlik ölçütü en-iyi-*N* önerileri için hem doğruluk hem de çevrimiçi performans kriterleri bakımından irdelendi. Bu ölçütleri doğruluk ve verimlilik açısından karşılaştırabilmek için iyi bilinen iki gerçek veri seti üzerinde birçok deneyler

1, Turgut Özal Üniversitesi, Ankara Meslek Yüksekokulu, Ankara, Türkiye.
E-mail: esenyurek@turgutozal.edu.tr
2, Anadolu Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskisehir, Turkey
E-mail: polath@anadolu.edu.tr

Received: 12 December 2012 ; **Revised:** 30 May 2013 ; **Accepted:** 31 May 2013

yapıldı. Ayrıca en-iyi-*N* öneri algoritması en benzer kullanıcıların verisi öneri üretilirken kullanılacak şekilde değiştirildi. Değişen kontrol parametrelerinin performansa olan etkisi araştırıldı. Deneysel sonuçlar doğruluk ve performans açısından analiz edilerek bazı öneriler sunuldu.

Anahtar Kelimeler: Benzerlik ölçütü, En-iyi-*N* önerisi, Doğruluk, Çevrimiçi performans

1. INTRODUCTION

Shopping over the Internet is increasingly turning out to be popular. Due to the widespread use of the Internet and increasing popularity of e-commerce, amount of data collected from many users becomes vast. The availability of massive quantity of data is called information overload (Yang et al., 2003). Since online vendors collect data about their customers, mining such data is vital for business purposes. Such companies utilize recommender systems to help their customers select appropriate products.

Collaborative filtering (CF) techniques are widely used by recommender systems. CF is a filtering and recommendation technique, which is widely used by many e-commerce sites. It helps people make correct choices according to the other people's selections (Resnick et al., 1994). Many users choose what to eat, read, listen, sight to see, and so on with the help of CF systems. When a user, referred to as an active user (*a*), intends to surf on a site to purchase a DVD, movie, book, etc., the online vendor recommends the product list that could be liked by her while considering the similarity of other users' ratings and her previous votes for the products in the database.

To be able to compare users' past preferences, a user-item database should be available. To create a database with the participation of the users, the preferences must be collected. Users can explicitly submit their ratings for given products. Such ratings can be given as scores on a rating scale from one to five. The user-item database, which is utilized by CF schemes, is an $n \times m$ matrix including ratings collected from n users for m products. Different recommendation algorithms are used for binary ratings. Miranda and Jorge (2009) mention four different algorithms for binary ratings. While in the user-based approach, recommendations for a new session are generated by analyzing the whole database, in the item-based approach, the authors need the similarities between each pair of items. Since typically the number of items is orders of magnitude smaller than the number of users, this results in an important memory and computational reduction (Sarwar et al., 2001).

CF systems provide two servicellbs. They either offer predictions for single items or provide a sorted list of items that might be liked by active users, called top-*N* recommendation (TN). One of the major steps in determining TN lists is to estimate the similarities between users and/or items in order to determine the best similar users and/or items. Therefore, utilizing the best similarity measure is imperative for the overall success of any recommendation system. Determining those entities very similar to the active users or the target items as neighbors helps CF schemes improve accuracy and online efficiency.

In this study, the effects of similarity measures on the quality of TN are scrutinized. The emphasis is given to binary similarity measures. The most popular seven binary similarity measures are investigated in terms of both accuracy and online efficiency. Since off-line costs like storage, computation, and communication (number of communications and amount of data to be transferred) costs are not that critical for the overall performance, the emphasis is given to online costs. Real data-based experiments are performed and the results are displayed.

2. RELATED WORK

Determining the best recommendation algorithm with respect to overall performance is imperative. Vozalis and Margaritis (2004) apply three existing filtering approaches, user-based, item-based, and hybrid, to evaluate the Unison-CF algorithm. Brozovsky (2006) describes a recommender system, where the author implements and performs a quantitative comparison of two CF and two global algorithms. The author implements a domain independent and freely available recommender system that is called ColFi system. The system architecture.has been designed to be flexible yet simple enough so that developers can focus on CF algorithms

Most recommendation systems employ variations of CF for formulating suggestions of items relevant to users' interests. However, CF requires expensive computations that grow polynomial with the number of users and/or items in the database. Papagelis et al. (2005) propose a method for addressing the scalability problem based on incremental updates of user-to-user similarities. Miranda and Jorge (2009) propose an incremental item-based CF algorithm, which works with binary ratings as it is typically the case in Web environment. Their method is capable of incorporating new information in parallel with performing recommendation.

Billus and Pazzani (1998) get their best performing algorithm, which is based on the singular value decomposition of an initial matrix of user ratings. Robu and Poutre (2009) propose a method for constructing the utility graphs of buyers automatically, based on previous negotiation data. That method is based on item-based CF and the experimental results have a high degree of accuracy. Miyahara and Pazzani (2000) discuss another approach to CF based on the simple Bayesian classifier, which is one of the most successful supervised machine-learning algorithms. Their proposed combined method, user- and item-based CF, performs better than single collaborative recommendation method (Miyahara and Pazzani, 2002)). Kaleli and Polat (2009) investigate how to improve Bayesian classifier-based CF systems' online efficiency. They divide users into clusters so that prediction can be generated on similar, dissimilar, or both similar and dissimilar users.

Cha et al. (2005) review, categorize, and evaluate various binary vector similarity and dissimilarity measures for character recognition. According to them, one of the most contentious disputes in the similarity measure selection problem is whether the measure includes or excludes negative matches. At last, the proposed similarity measure can be further boosted by applying weights and they demonstrate that it outperforms the weighted Hamming distance that is one of the similarity measures. Several dissimilarity measures for binary vectors are formulated and examined for their recognition capability in handwriting identification for which the binary micro-features are used to characterize handwritten character shapes. Zhang and Srihari (2003) study seven similarity measures, Jaccard-Needham, Correlation, Yule, Russell-Rao, Sokal-Michener, Rogers-Tanimoto and Kulzinsky, for binary feature vectors, which are summarized by Tubbs (1989). Choi et al. (2010) collect different similarity and distance measures and reveal their correlations through the hierarchical clustering technique. Veal (2011) investigates mathematical properties of specific binary similarity measures. The author also studies relationships among such measures.

As presented above, in order to provide TN recommendation lists efficiently, various approaches have been proposed in the literature. Moreover, various schemes have been proposed to overcome several problems of CF methods. In addition, different binary similarity measures have been investigated for better character recognition and handwriting. However, comparison of binary similarity measures for providing TN recommendation lists has not been studied before. In this study, various binary similarity measures are determined and investigated with respect to both accuracy and online efficiency while generating TN recommendations. Such measures are evaluated by performing some real data-based experiments.

3. BACKGROUND

Generally speaking, recommender systems perform two basic filtering services, as briefly presented before. They offer predictions for single items. They also generate TN recommendation lists. In order to offer TN lists, predictions are first estimated for all unrated items, they are then sorted, and finally the first N items are returned as the TN list to a (Polat and Du, 2008).

As explained previously, Tubbs (1989) summarizes various binary similarity measures, while Zhang and Srihari (2003) study several similarity measurements in the context of handwriting. Although there are normally various similarity measurements, we investigate the most well-known seven measures.

According to StataCorp (1996), similarity measures can be classified as continuous measures, binary measures, and mixed measures. Similarity measures for continuous data are called continuous measures, for binary data, they are called binary measures; and for a mix of continuous and binary data, they are called mixed measures. There are different examples for each group of measures. In this study, the binary similarity measurements, shown in Table 1, are investigated.

Similarity measures for binary data are based on four values. First one is the number of ones from two vectors (S_{11}), second one is the number of ones from the first vector and zeros from the second vector (S_{10}), third one is thenumber of zeros from the first vector and ones from the second vector (S_{01}), and the last one is the number of zeros from two vectors (S_{00}). To clarify the calculations of similarity measures, we can give a simple example, as follows: Suppose that X and Y represent two vectors, where X = (100101010110) and Y = (101101010011). Given X and Y, $S_{11} = 5$, $S_{10} = 1$, $S_{01} = 2$, and $S_{00} = 4$.

Table 1. Binary similarity measurements

Similarity Metric	**Definition**
Anderberg	$\frac{\frac{S_{11}}{S_{11}+S_{10}} + \frac{S_{11}}{S_{11}+S_{01}} + \frac{S_{00}}{S_{01}+S_{00}} + \frac{S_{00}}{S_{10}+S_{00}}}{4}$
Gower2	$\frac{S_{11}S_{00}}{\sqrt{(S_{11}+S_{10})(S_{11}+S_{01})(S_{10}+S_{00})(S_{01}+S_{00})}}$
Jaccard	$\frac{S_{11}}{S_{11}+S_{10}+S_{01}}$
Kulczynski	$\frac{\frac{S_{11}}{S_{11}+S_{10}} + \frac{S_{11}}{S_{11}+S_{01}}}{2}$
Ochiai	$\frac{S_{11}}{\sqrt{(S_{11}+S_{10})(S_{11}+S_{01})}}$
Pearson's Correlation	$\frac{S_{11}S_{00} - S_{10}S_{01}}{\sqrt{(S_{11}+S_{10})(S_{11}+S_{01})(S_{10}+S_{00})(S_{01}+S_{00})}}$
Yule	$\frac{S_{11}S_{00} - S_{10}S_{01}}{S_{11}S_{00} + S_{10}S_{01}}$

4. EFFECTS OF SIMILARITY MEASURES ON THE QUALITY OF TN

To determine the best similarity measures or to compare different similarity measures in terms of both accuracy and online efficiency, we conducted several experiments using two well-known real data sets.

4.1 Data Sets

In this study, we utilized the well-known two data sets; MovieLens (ML) and Jester. ML data set includes ratings for several movies. It was collected by the GroupLens research team (www.cs.umn.edu/research/GroupLens) at the University of Minnesota. It contains ratings for 3,900 movies by 6,041 users. The ratings were numeric and discrete, ranging from one to five. In ML, each user has rated at least 20 movies. Jester is a web-based joke recommendation system (eigentaste.berkeley.edu/user/index.php). The data set contains ratings for 100 jokes by 17,998 users. The ratings were numeric and continuous ranging from -10 to 10. We chose ML to represent a sparse data set while we selected Jester to represent a dense data set. Table 2 describes both data sets.

Table 2. Data sets with their density

	ML	*Jester*
Number of users	6,041	17,998
Number of items	3,900	100
Number of ratings	788,063	906,474
Density (%)	3.34	50.37

4.2 Top-N Recommendation Method

We proposed to utilize the following algorithm to offer top-*N* recommendations: Traditional algorithms are based on frequencies and the most frequently bought items by similar users are returned as TN lists. Our approach, on contrast, does not use frequencies. Our method includes the following steps:

i. Compute similarity weights between *a* and each user *u* in the database (w_{au})

ii. Choose the most similar *k* users as neighbors based on similarity weights

iii. For each unrated item *j* of *a*, do the followings:

a. Determine those neighbors who rated item *j* as 1; and sum their similarity values ($\sum_{sj}$)

b. Determine those neighbors who rated item *j* as 0; and sum their similarity values ($\sum_{dj}$)

c. Compute $\sum_j = \sum_{sj} - \sum_{dj}$ value

iv. After calculating $\sum_j$ values for all unrated items, sort them in descending order

v. Return the first *N* items as TN list to *a*

The quality of TN, thus, depends on similarity metric that is used to form neighborhoods. In order to show the effects of similarity metrics on the overall performance of TN, we conducted several experiments. The details of them are given in the following.

4.3 Our Methodology

The chosen data sets, ML and Jester, have numeric ratings. First, the numeric ratings must be converted to binary ones, as proposed by Miyahara and Pazzani (2000). For ML data set, the ratings were transformed into one (*like*) if they are bigger than three; or zero (dislike) otherwise. Similarly, for Jester data set, the ratings were converted into one (*like*) if they are bigger than two; or zero (*dislike*) otherwise. Thus, in our data sets, zero (0) represents the disliked items and one (1) represents the liked items.

After data transformation, we uniformly randomly selected 3,000 users who rated at least 30 and 40 items from ML and Jester, respectively. We then uniformly randomly selected train and test sets. To do so, we uniformly randomly divided these users into two sub sets. One of the sets, referred to as train set, contains 2,000 users. The other set, called test set, includes the remaining 1,000 users. For test sets, we selected those users who rated at least 60 items. Notice that the train and test sets are disjoint. In each set of trials conducted in the followings, two thirds of total numbers of users were used for training and one third of total numbers of users were used for testing. For example, when we used 1,000 uniformly randomly chosen users from train set for training, then we utilized 500 uniformly randomly selected users from test set for testing. In Table 3, we show the number of users used for training and testing.

Table 3. Number of train and test users

Total number of users	3,000	1,500	750	375	186
Number of train users	2,000	1,000	500	250	124
Number of test users	1,000	500	250	125	62

For each user in the test set, we determined their rated items. After utilizing our method using different similarity metrics, we estimated $\sum_j$ values for all rated items. We sorted such items according to $\sum_j$ values in descending order. We finally returned the first five, 10 or 20 items as top-5, top-10 or top-20 recommendation lists, respectively. We assumed that if an item is in TN list, then its rating is one (like) because it does not make sense to include disliked products in the TN lists. We compared their predicted values (1s) with their true votes. After computing hit ratios as percent (*number of liked items listed in TN lists*/*N*), we displayed them. We also calculated total amount of online times (*T* values) for different metrics and showed them, too. We used both data sets with varying controlling parameters that might affect the overall performance. Number of users (*n*), number of items (*m*), number of neighbors (*k*), density, and similarity measurements are among such parameters.

4.4 Experiments

We first performed experiments using Jester data set, where we set *n* at 2,000. We varied *k* from 2,000 to 25. We also changed *N* from five to 20. The results for *N* being five, 10, and 20 are very similar to each other. Therefore, we displayed the results for *N* = 10 only. Figure 1 shows hit ratios with varying *k* values for all similarity metrics.

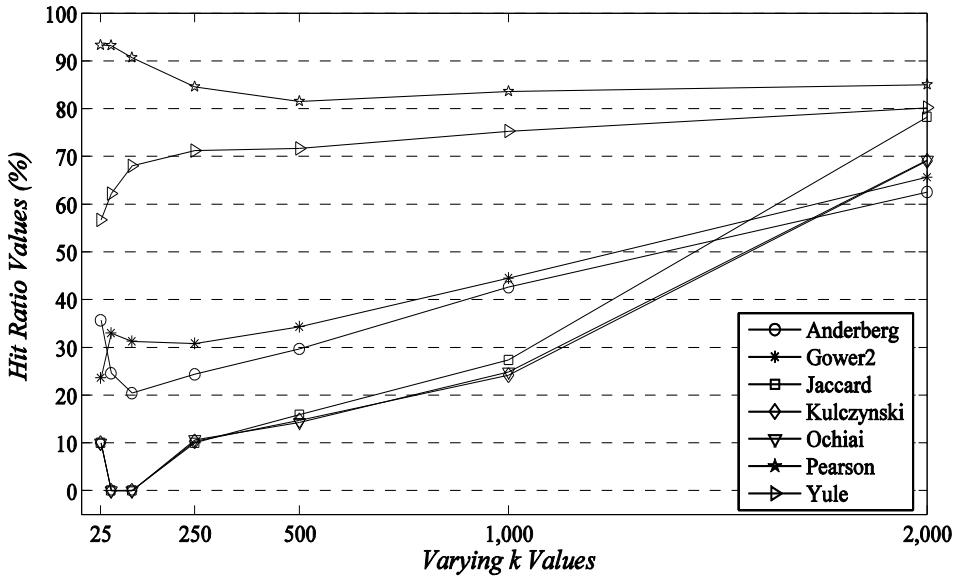

Figure 1. Hit ratio values with varying *k* values (Jester & *n* = 2,000)

As seen from Figure 2, the best hit ratio values are provided by Pearson Correlation similarity measurement for all *k* values except 1,000. With increasing *k* values from 250 to 1,000, the results become worse for Pearson Correlation metric. When *k* = 2,000, Pearson Correlation achieves the best outcome. Yule metric is the second best metric for smaller *k* values. Gower2 measure performs the worst for smaller *k* values. The outcomes for smaller *k* values do not display a stable trend. This phenomenon can be explained the sparsity of ML data set.

After displaying hit ratio values, we also estimated online duration times. In Figure 3, we showed *T* values for all similarity metrics for Jester data set. As seen from Figure 3, the best durations are observed for Yule similarity measurement. In terms of online efficiency, Pearson Correlation metric follows Yule measure. Although Anderberg performs the worst, other metrics except Pearson Correlation and Yule metrics behave very similar to it in terms of online efficiency. Since there are limited number of items (100 jokes), online amount of times spent for generating TN lists are smaller.

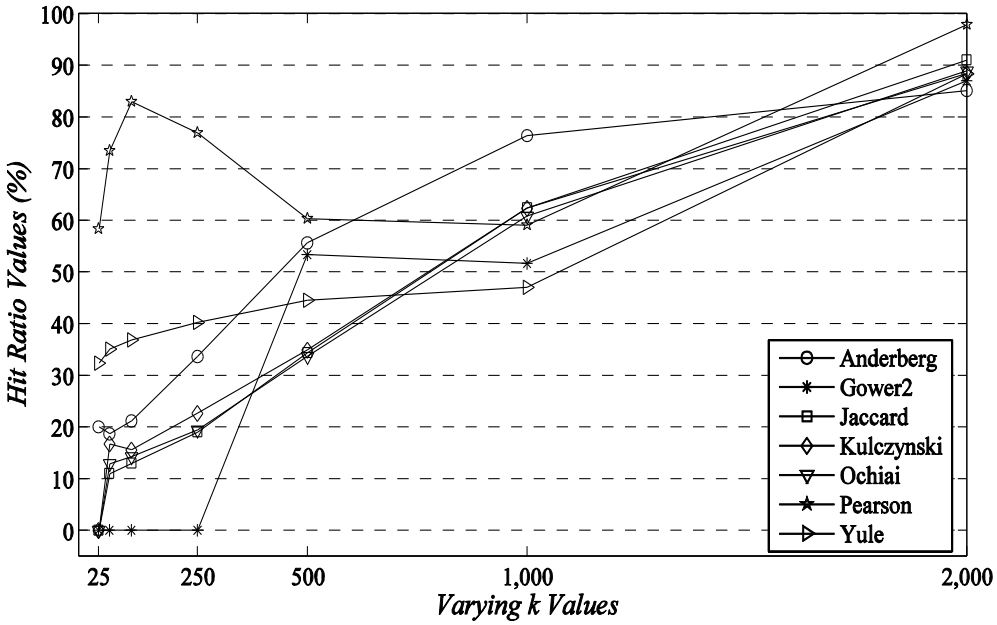

Figure 2. Hit ratio values with varying *k* values (ML & *n* = 2,000)

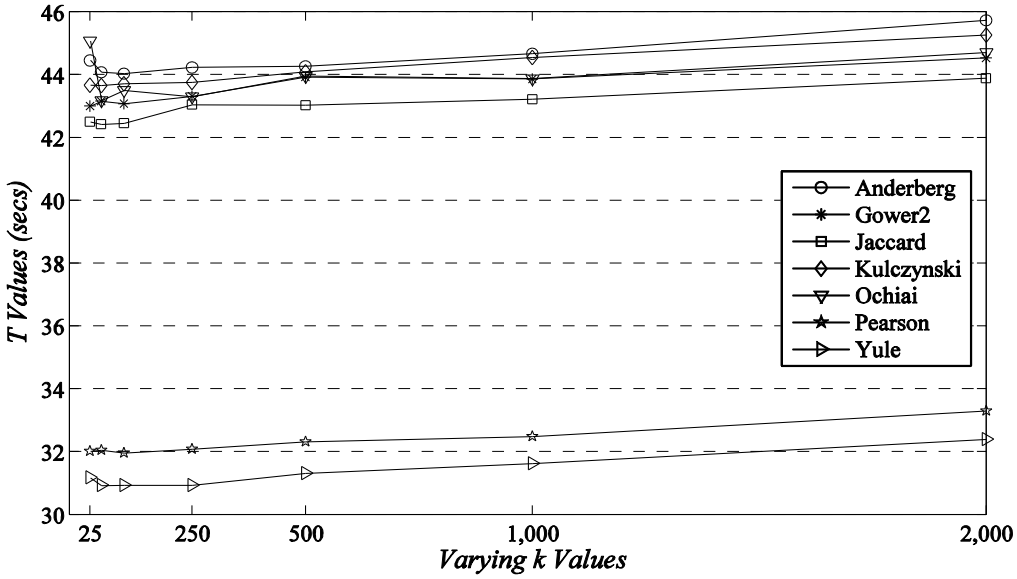

Figure 3. *T* values with varying *k* values (Jester & *n* = 2,000)

We also computed online duration times for ML data set. Figure 4 shows *T* values for all similarity metrics. As seen from Figure 4, like we observed for Jester, Yule again achieves the best performance for ML. Similarly, Pearson Correlation metric follows Yule measure. The worst duration values are observed for Anderberg measure. Other metrics behave very similar in terms of online time, as seen from the figure. Due to the larger number of items, *T* values are bigger for ML than Jester.

We also conducted similar sets of experiments using both data sets while varying *n* from 2,000 to 124. Since we observed very similar results, we did not display them. Also note that we used a dense (Jester) and a sparse (ML) data set in our experiments to show how density affects overall performance.

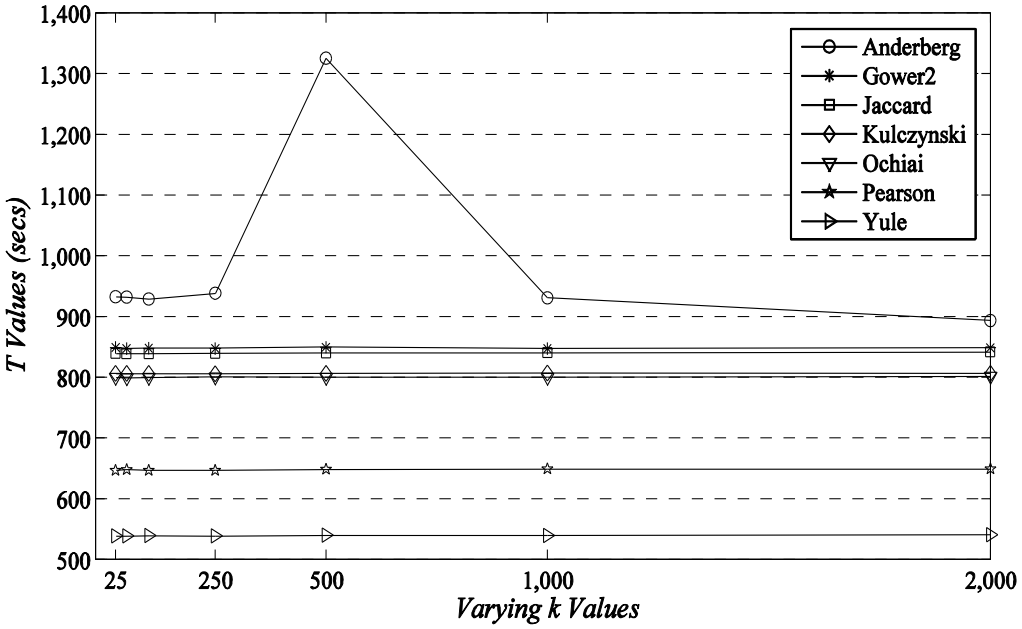

Figure 4. *T* values with varying *k* values (ML & *n* = 2,000)

In addition to varying *n* and *k* values, different *m* values might also affect overall performance of TN recommendation scheme. Therefore, after performing experiments to demonstrate the effects of varying *n* and *k* values, we also conducted trials to show the effects of varying *m* values on TN using ML data set only because there is limited number of items in Jester. We varied *m* from 3,900 to 500. We estimated TN recommendation lists for each active or test user while varying *m* (*m* = 3,900, 2,000, 1,000, or 500) and using different similarity metrics, where we also set *N* at 20, 10, or five. We used 900 and 450 train and test users, respectively in which we set *k* at 100. In the following, since we obtained the similar outcomes, we demonstrated the hit ratios and *T* values for *N* = 10 only in Figure 5 and Figure 6, respectively.

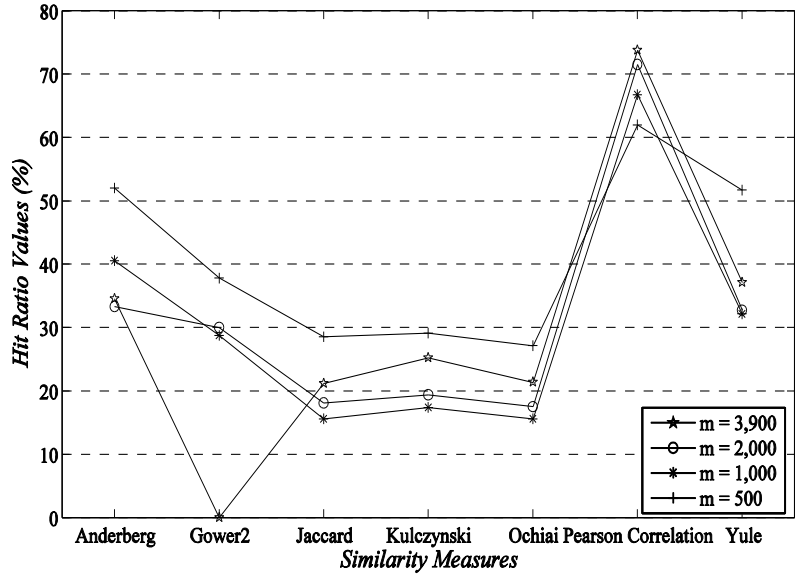

Figure 5. Hit ratio values with varying *m* values

Figure 5 displays the hit ratio values with varying *m* values for ML, where *N* is 10. As seen from Figure 5, Pearson Correlation measure provides the most accurate TN lists for all *m* values. The results for different *m* values are close to each other for Pearson Correlation metric. The quality of the TN recommendations is the worst when we utilized Ochiai metric. The only exception is *m* being 3,900 for which Gower2 is not able to provide any true TN lists.

Figure 6 represent the *T* values with varying *m* values for ML. Remember that we fixed *k* at 100. As expected, while the number of item decreases, online duration time decreases, as well. The less number of items involves in recommendation process, the less time spent on online computations. The best results are observed when Yule measure is used. Pearson Correlation metric achieves the second best results. For other similarity measures, the outcomes are very close to each other. Ochiai metric slightly performs worse than the remaining measures do.

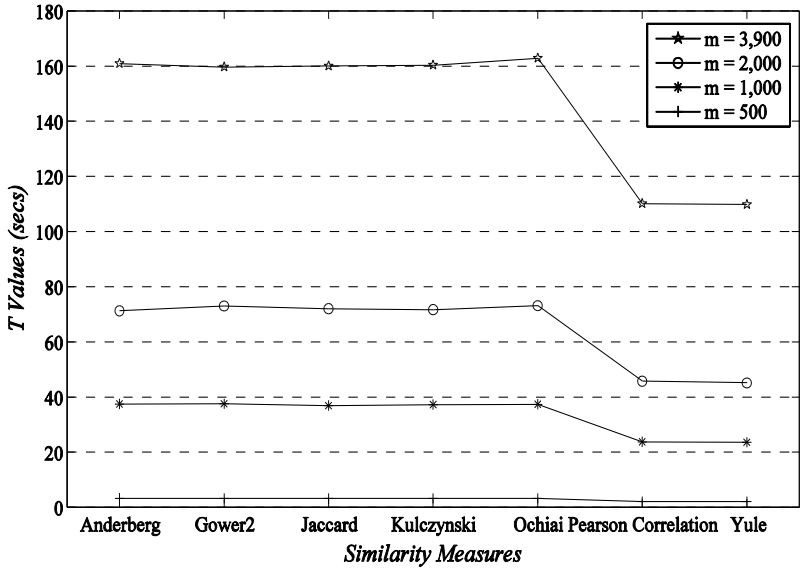

Figure 6. *T* values with varying *m* values

5. CONCLUSION AND FUTURE WORK

We compared seven binary similarity metrics in terms of accuracy and online efficiency in top-*N* recommendation algorithm. We performed real data-based experiments using two data sets in which we varied different controlling parameters like number of users, number of items, number of neighbors, density, and so on. Consequently, we can say that in order to get the best results with respect to accuracy for dense and sparse data sets, like Jester and ML, respectively, Pearson Correlation or Yule metric is the best choice if the number of train users is 2,000. For dense data sets, Jaccard, Ochiai, or Kulczynski measures are not good choices, because they provide the worst TN lists. For sparse data sets, Gower2 measure is not the right choice. Also note that since we observed the similar outcomes for smaller *n* values, Pearson Correlation and Yule metrics can be chosen for better accuracy.

When it comes to online efficiency, Pearson Correlation or Yule can be selected as appropriate metrics. On the other hand, Anderberg similarity measurement's online efficiency is the worst for both kinds of data sets, sparse and dense sets, when n = 2,000. Since we observed the similar results for the other *n* values such as 1000, 500, 250, or 124, Pearson Correlation or Yule can be selected for better performance.

When we changed number of items involving in recommendation process, Pearson Correlation or Yule measure achieves the most accurate results for almost all *m* values. On the other hand, Gower2 and Ochiai give worst results.

Like top-*N* recommendations, recommender systems can perform prediction services. Hence, we are planning to investigate the same metrics while performing prediction services. Although we studied seven metrics, other binary similarity metrics should also be investigated.

REFERENCES

Billus, D. and Pazzani, M.J. (1998). Learning Collaborative Information Filters. *Proceedings of the 15th International Conference on Machine Learning*, Adison, WI, USA, 46-54.

Brožovský, L. (2006). Recommender System for A Dating Service. *Master's thesis*, Prague, Czech Republic: Charles University in Prague.

Cha, S.-H., Yoon, S., and Tappert, C.C. (2005). On Binary Similarity Measures for Handwritten Character Recognition. *Proceedings of the 2005 8th International Conference on Document Analysis and Recognition*, Seoul, Korea, vol. 1, 4-8.

Choi, S.-S., Cha, S.-H., and Tappert, C.C. (2010). A Survey of Binary Similarity and Distance Measures. *Journal of Systemics, Cybernetics and Informatics* 8 (1), 43-48.

GroupLens Research, Data Sets, 2006. http://www.grouplens.org/node/12. (May 15, 2012).

Kaleli, C. and Polat, H. (2009). Similar or Dissimilar Users? Or both? *Proceedings of the 2009 2nd International Symposium on Electronic Commerce and Security*, Nanchang City, China, vol. 2, 184-189.

Miranda, C. and Jorge, A.M. (2009). Item-based and user-based incremental collaborative filtering for Web recommendations. *Lecture Notes in Computer Science* 5816, 673-684.

Miyahara, K. and Pazzani, M.J. (2000). Collaborative filtering with the simple Bayesian classifier. *Lecture Notes in Computer Science* 1886, 679-689.

Miyahara, K. and Pazzani, M.J. (2002). Improvement collaborative filtering with the simple Bayesian classifier. *Transactions of Information Processing Society of Japan* 43 (11), 3429-3437.

Papagelis, M., Rousidis, I., Plexousakis, D., and Theoharopoulos, E. (2005). Incremental Collaborative Filtering for Highly-Scalable Recommendation Algorithms. *Lecture Notes in Computer Science* 3488, 553-561.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.T. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, USA, 175-186.

Robu, V. and Poutré, H.L. (2009). Learning The Structure of Utility Graphs Used In Multi-Issue Negotiation Through Collaborative Filtering. *Lecture Notes in Computer Science* 4078, 192-209.

Sarwar, B.M., Karypis, G., Konstan, J.A., and Riedl, J.T. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, 285-295.

Stata Corp. LP, Stata 11 help for measure option, 1996. http://www.stata.com/help.cgi?measure+option (May 15 2012).

Tubbs, J.D. (1989). A Note on Binary Template Matching. *Pattern Recognition* 22 (4), 359-365.

Veal, B.W.G. (2011). Binary Similarity Measures And Their Applications In Machine Learning. PhD thesis, London, United Kingdom: London School of Economics.

Vozalis, M. and Margaritis, K.G. (2004). Unison-CF: A Multiple-Component, Adaptive Collaborative Filtering System. *Lecture Notes in Computer Science* 3137, 255-264.

Yang, C.C., Chen, H., and Hong, K. (2003). Visualization of Large Category Map For Internet Browsing. *Decision Support Systems* 35 (1), 89-102.

Zhang, B. and Srihari, S.N. (2003). Binary Vector Dissimilarity Measures for Handwriting Identification. *Document Recognition and Retrieval X* vol. 5010 (1), 28-38.