Similarity Criterion Computation for User-based Collaborative Filtering Systems

Pitaya Poompuang
Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi
39 Moo 1, Rangsit-Nakhonnayok Road, Thanyaburi, Pathum Thani 12110, Thailand
Email: pitaya_p@.rmutt.ac.th

Abstract

Neighborhood selection scheme is one of the crucial steps of providing recommendation in the collaborative filtering system. Almost similarity computation algorithms operate on the information of rating pattern on a set of co-rated items between two users using only one similarity computation algorithm. Since the number of co-rated items of each user may be different, the quality of providing opinion between each user pair for a particular item may be also different. In this work we propose two criteria similarity computation method in order to make a refinement of neighborhood selection. The experimental results in this work show that our proposed algorithm can improve the performance of the collaborative filtering system in the perspective of f-measure.

Keywords: Recommender Systems, Similarity, Criteria, Collaborative Filtering, Neighborhood

1. Introduction

The user-based collaborative Filtering (CF) systems' performance in online shopping environment relies heavily on opinions of others similar neighbor of the active users in the system. Good opinions are usually gained by the user who have high similar in pattern of product preferences. Therefore neighborhood selection scheme become a crucial step in the process of providing recommendation of the CF system. Almost recommender system compares users using only one similarity computation algorithm. Basically most algorithms compute similarity between two users by considering rating pattern on the co-rated items of them. However each pair of users may have different in the number of co-rated items. Two users tend to be low quality neighborhood to provide an opinion for each other when similarity between them was calculated based on a small number of co-rated items. In this work we propose two criteria similarity computation method in order to make a refinement of neighborhood selection. Applying this method with the standard similarity computation will guarantee that the neighbors who have high quality in providing opinion for an active user are selected. With this concept, the performance of collaborative filtering system can be improved.

2. Background and Knowledge

Recommender system is usually implemented as a service program running in an online shopping web site on the Internet. The aim of the recommender system is to provide a user the convenience for decision making in choosing his relevant unseen products from a large pool of product items. In order to achieve this goal, the recommender system must collects data of users and products that users have experienced with and then identify the relationship between users and products. A common type of the relationship between a user and a product is the user' preferences which can be represented by a non-negative value in a certain range such as 1(disliked) to 5(liked), called the rating value. Most of recommender systems represent users' preferences data as a two dimension matrix, where each row of matrix represents information about a user and each column represents information of a specific item. The value that appeared at the intersection between a row and a column is the rating value that the user has assigned to the item. Note that the user-item rating matrix is very sparse, as most users do not rate most items, expressed as blank space in the matrix.

Generally the main task of recommender systems is to predict the rating of a user for a new item. Note that items can be anything, such as movie, books, journal articles, or vacation destinations in a system domain, for which a human user can express his preferences or rating. The user under current consideration for recommendations is called the *active* or the *target user*. The recommendation problem can be formulated as the following function *UF*: *U* x *I* → *R*, where *U* be the set of all users and *I* be the set of all possible items in the system domain such as books or movies that can be recommended to a user. The *UF* be a utility or preference function that measures usefulness of the item *i* to user *u.* The *R* is a totally ordered set of ratings.

2.1 Basic Recommendation Approaches

There are two fundamental approaches to produce a list of recommendation for a user [13] in recommender systems. The first one is called the content-based (CB)—the list of items is recommended based on characteristic of items that they have acquired and liked in the past [12], [17], [18] the second one is called collaborative filtering (CF)—the list of items is recommended to individual users based on the similar users' preferences. Among various recommendation approaches, collaborative filtering methods appear to be the most rapidly advancing research area. The first of the automated CF methods was introduced in the GroupLens Usenet article recommender system [21]. Collaborative filtering is also known as k-NN (k-Nearest Neighborhoods) collaborative filtering.

2.2 Collaborative Filtering

Usually CF execution begin from finding users whose past rating behavior is similar to that of the active user, and then it uses their rating on other items to predict what the active user will like. This method is called user-based CF. The User-based CF, while effective, suffers from scalability problems as the number of user base grows [7]. Searching for the neighbors must directly compute against all other users in the system. More scalable algorithm, Item-based collaborative filtering [11], [14], [23] takes a major step in this direction and is one of the most widely deployed collaborative filtering techniques today. Rather than using similarities between users' rating behavior to predict preferences item-based uses similarities between the rating patterns of items. If two items tend to have the same users like and dislike them, then they are similar and users are expected to have similar preferences for similar items. Item-based CF generates predictions using the user's own ratings for other items combined with those items' similarities to the target item, rather than other users' ratings and user similarities as in user –based CF.

With the broad meanings of the word "similarity", for user-based CF, users can be compared on various aspects, such as demographic information and product consuming/purchasing behavior. For item-based approach, similarity between items must be compared based on the rating patterns of items. From this perspective, user-based CF is more flexible than the item-based approaches. User-based CF can be developed using various methods and can be applied to various types of items.

2.3 Neighborhood Based Method

There are several methods can be used in collaborative filtering such as Bayesian networks [4], singular value decomposition with neural net classification [5] , inductive rule learning [4], a graph–theoretic approach [2], a Bayesian mixed weighed majority weighting [6], clustering in reduced dimensions using principle component analysis [8] and latent class models [10], however neighborhood-based methods or nearest neighborhoods methods are the most prevalent algorithms used in collaborative filtering [9].

In the neighborhood-based methods a subset of appropriate users called neighbors of an active user are chosen based on their similarity to the active user. In Neighborhood-based methods can be separated into three steps. The system start by first weighing all user with respect to similarity with the active user, second, selecting a subset of users to use as a set of predictors, and Third, calculating a prediction from a weighted combination of selected neighbors' rating.

2.4 Similarity and Prediction

Similarity measurement, in the user-based collaborative filtering, is about comparison between users. This comparison is determined by analyzing their past user-item interaction [3]. When the system represent a user in term of vector space, the past rating pattern of the same items —called co-rated items—of two users can be compared using an appropriate similarity computation algorithm.

There are a number of algorithms for calculating the similarity. The most common method is the Pearson Correlation Coefficient (PCC) which measures the degree to which a linear relationship exist between two users or two items [22], [21]. Other well-known similarity computation method is the vector similarity called the Cosine Vector Similarity [1]; [15]. While the Pearson Correlation Coefficient can work well, it requires more effort in correlation computation and application because the results of PCC can be either positive or negative real number ranged from -1 to 1. Also the cosine method does not work well when applied to small sized vectors [19]; [20]. In this work we apply the Tanimoto correlation method to compute user-user similarity as shown below in Eq. (1). The Tanimoto algorithm calculate correlation base on rating's pattern between users like the Pearson Correlation Coefficient but its correlation value is positive real number. Once the similarity measure between users (for user-base CF) or items (for item-based CF) is acquired, the similarity value can be used as the weights for a user's rating in prediction phase of the recommender system. Although there are various methods for predicting the rating of yet unknown items for a user, the most simple and popular one is the weighted sum and the adjusted weighted sum as shown in Eq. (2) and Eq. (3), respectively. Once the ratings of the yet unknown items for a user are predicted, we can recommend to the user the item(s) with the highest estimated rating(s).

$$sim(x,y) = \frac{\sum_{i\in I_{x,y}} (r_{x,i}) . (r_{y,i})}{\sqrt{\sum_{i\in I_{x,y}} (r_{x,i})^2 + (r_{y,i})^2} - \sqrt{\sum_{i\in I_{x,y}} (r_{x,i})(r_{y,i})}} \qquad (1)$$

$$r_{x,i} = \bar{r}_x + z.\sum_{y\in U_i} (r_{y,i} - \bar{r}_y).sim(x,y) \qquad (2)$$

$$r_{x,i} \;=\; z.\sum_{y\in U_i} r_{y,i}.sim(x,y) \qquad (3)$$

The $I_{x,y}$ represents a set of items that both users x and y have rated. The $r_{x,i}$ and $r_{y,i}$ represent ratings on item i given by user x and y. The Z is the normalization factor defined as

$1 / \sum_{y \in U_i} | sim(x, \; y)$.

2.5 Significance-Weighting Factor of Similarity

One important issue (i.e., trust in a correlation with a neighbor) addressed in literature of [9] is that it was common for the active user to have highly similar neighbors that were based on a very small number of co-rated items. It was frequently proved that two users tend to be low quality predictors for each other when similarity between them was calculated based on a small number of co-rated item (three to five). The more data points to compare, the more we can trust that the computed similarity is representative of the true correlation between the two users. Therefore the accuracy of prediction algorithms would be improved if we adjust similarity measure that were computed based on a small number of co-rated items with correlation significance-weighting factor. So the linear drop-off was applied in the work of in literature of [9].

The significance threshold was defined to adjust similarity measure. If two users had fewer than 50 commonly rated items, their correlation would be adjusted by a significance weight of n/50, where n is the number of co-rated items. If these were more than 50 co-rated items, then no adjustment was applied. In this manner, correlation with small numbers of co-rated items are devalued, but the correlation with 50 or more commonly co-rated items are not dependent on the number of co-rated items. However in the small system that has a small number of items even in the larger system but the majority of users rated only a few items, the threshold 50 would not appropriate.

3. Methodology

In this work, we apply the three steps of neighborhood-based methods mentioned in the section of Neighborhood-Based Method as a framework to implement collaborative filtering engine. In the first step: we define two different choices of applying similarity computation methods, by applying Tanimoto similarity computation method and by adjusting traditional similarity measure with two criteria similarity computation method. In the second step: we select a subset of users who have high similar to a target user in several level percentage of similarity ranges of value i.e. 90-100, 80-100, 70-100, 60-100, and 50-100. Then we apply the simplest tradition prediction algorithm known as weighted sum average, see equation (2).

The experimental results of our method are compared against to the results of traditional approach which identify user's neighbor without considering any significant weight of similarity. The evaluation of this work are performed based on classification technique in term of the f-measure using the threshold value = 4 for Top-N recommendations, i.e. top-3, top-5 and top-7 recommendations. The higher f-measure value is the better lower f-measure. Note that, the f-measure serves as the harmonic mean of precision and recall to provide a simplified description of the results' evaluation and discussion.

The challenging part of our work is about the quality of similarity measure. Typically, similarity measure between users can be calculated based on the commonly rated items between users and their preference values. It is probably sometimes that different pairs of the compared users may have differences in terms of the number of common rated items. This indicates that only similarity algorithm cannot provide enough quality of similarity measure between users.

3.1 Two Criteria Similarity

To optimize user-user similarity, we introduce a concept of two-criterion weighted–similarity for user-base Collaborative Filtering. we define overall similarity between users as a combination between two criteria of similarity. First criterion relies on preferences pattern of users based on expressed common rated items (sp). Second criterion relies on the number of common rated items (sc). In this work we assign weight for each criteria in several values. However the combining weight of two criteria should be 1.0. For example, for the first criterion, we apply Tanimoto similarity computation and weight the results with 0.5, as shown in equation. (4). For the second criterion, we compute similarity based on the number of common rated item between users (ncri(x, y)), normalized by normalization factor (nf), as shown in equation (5). The normalization factor is defined as equation (6), where Ui is a set of User who rated an item i. Finally we combine these two criteria of similarity to provide an overall similarity between users as equation (7).

$$sp(x,y) = sim(x,y) * (0.5) \qquad (4)$$

$$sc(x,y) = \frac{ncri(x,y)}{nf} * (0.5) \qquad (5)$$

$$nf = \underset{y \in U_i}{MAX}(ncri(x,y)) \qquad (6)$$

$$sim(x, y) = sp(x, y) + sc(x, y) \qquad (7)$$

4. Experiments

Our experiments are conducted based on the use of MovieLens dataset (Miller et al. 2003), provided by the GroupLens Research Project at the University of Minnesota. The dataset contains 100,000 ratings, given by 943 users on 1682 movies. Each user has rated at least 20 movies using an integer in the range 1 to 5

The practice of confirming an experimental finding by repeating the experiment using an independent assay technique based on 5 series of different dataset to provide more reliability of experimental results. In order to provide 5-fold dataset for cross validation environment, the different series of training and testing dataset are prepared in two steps. First, the entire user-item dataset is randomly divided into two disjoint subsets, 80% for training, and 20% for testing. Second, the random division process is performed repeatedly five times to get five different pairs of 80% training and 20% testing dataset. Each training dataset then can be presented as the User-Item rating matrix.

5. Experimental Result and Discussion

Before performing the experiments, we developed a traditional user-based collaborative filtering recommender system. This system predicts rating of an item for a user based on opinions of all possible neighbors using the weighted sum average algorithm. The overall performance of the recommendation engine is evaluated in the top-3, top-5 and top-7 recommendations situation in term of f-measure value. The results of this experiment shown in table 1 and then are used as the baseline for comparing with others results in other experiments.

Table 1 Baseline results, the F-measure of traditional user-based CF.

Top-3(%)	Top-5(%)	Top-7(%)
65.01	62.56	60.64

Table 2 The f-measure of traditional user-based CF with similarity ranges of neighbors.

	Similarity Ranges(%)				
	50-100	60-100	70-100	80-100	90-100
Top-3	64.06	65.02	65.17	65.78	67.36
Top-5	62.13	63.09	63.83	67.89	65.58
Top-7	61.31	61.70	63.05	64.12	64.19

Table 2 shows that the last one correlation thresholds (90-100) for top-3, the last three highest correlation threshold(70-100, 80-100, and 90-100) for top-5 and top-7, the f-measure values are increased more than 1% from the base line, otherwise the f-measure values are closed to the baseline. The three most percentage of f-measure improvement from the baseline are 3.6%, 8.5% and 5.8% belong to top-3 with similarity range 90-100, top-5 with similarity range 80-100 and top-7 with similarity range 80-100 respectively. These results also shows that the best (f-measure) performance of recommendation engine can be derived from using correlation threshold 90% up for the result of top-3 and top-7 a, whereas 80% up for the result of top-5. This information confirms the concept that the quality of recommendations is affected by high quality of neighbors. The quality neighbors is the users who have somewhat high similar to an active user. In this case, users with similarity equal to or higher than 80% are identified as the high quality neighbor.

Table 3 F-measure result in top-3 when apply 2-criteria similarity and select group of neighbor based on several ranges of similarity.

	Similarity Ranges(%)	
sc	80-100	90-100
0.1	66.48	69.70
0.2	69.79	68.62
0.3	**72.45**	67.77
0.4	**73.15**	68.53
0.5	**73.30**	69.19
0.6	**73.28**	69.68
0.7	**72.72**	70.23
0.8	**72.43**	70.28
0.9	**72.97**	70.13

The next experiment we try to make a refinement in neighborhood selection by applying a two criteria similarity computation before applying the correlation threshold technique. We only concentrate on the last two correlation threshold (80-100 and 90-100 % of similarity values). The experimental results (table 3, 4 and 5) show that applying the two criteria similarity computation yields the positive effects to the performance of recommendations in the f-measure perspective. More than 1% f-measures are improved from the base line results in all cases. The f-measure values written in bold latter format in table 3, 4 and 5 show the cases that the f-measure are improved more than 10% from the base line, almost these case belong to the sc criteria higher than 0.2. The best f-measure performance (shown by the number with underline-bold format) belong to the criteria sc(x,y) = 0.5 at the range of 80-100. It is improved from the base line to 12.75%. For top-5 and top-10 recommendation, the best f-measure performance (shown by the number with underline-bold format) still fall into the similarity range of 80-100 when the criteria sc(x,y) = 0.6, f-measure values are improved 14.07% and 15.89% for top-5 (see Table 3), and top-7 (see Table 4) respectively.

Table 4 F-measure result in top-5 when apply 2-criteria similarity and select group of neighbor based on several ranges of similarity.

	Similarity Ranges (%)	
sc	80-100	90-100
0.1	63.95	68.24
0.2	67.30	67.24
0.3	**70.05**	66.40
0.4	**71.05**	67.27
0.5	**71.28**	68.12
0.6	**71.37**	68.56
0.7	**71.04**	69.00
0.8	**70.77**	69.09
0.9	**71.21**	68.82

Table 5 F-measure result in top-7 when apply 2-criteria similarity and select group of neighbor based on several ranges of similarity.

	Similarity Ranges (%)	
sc	80-100	90-100
0.1	61.93	67.18
0.2	65.53	66.55
0.3	68.42	65.79
0.4	69.76	66.65
0.5	70.03	67.49
0.6	70.28	68.10
0.7	70.22	68.53
0.8	69.98	68.46
0.9	70.23	68.47

6. Conclusion

As the performance of user-based CF rely on other users' opinion, choosing neighborhood and calculating similarity between users become crucial steps of the prediction. In order to improve performance of prediction, we proposed the two criteria similarity computation to make a refinement of neighborhood selection, we also apply correlation threshold for neighborhood selection by define several similarity ranges (i.e., 50-100, 60-100, 70-100, 80-100, 90-100), make experiments and evaluate the performance of the proposed method base on top-3, top-5, and top-7 recommendation method in the perspective of f-measure. The experimental results show that the performance of recommender systems is improved when compare to the base line results in all case. However the best case tend to be occurred when assigning criteria of sc(x,y) to 0.5-0.6%. The criteria threshold for neighborhood selection is about 80-100 percentage of similarity value between users. Although our work only focusing on two criteria of similarity computation, the concept of its can be enhanced by other criteria to get better performance, however weighting for each criteria must be done appropriately.

References

[1] Adomavicius, G., Manouselis, N., Kwon, Y., "Multi-Criteria Recommender Systems", Recommender Systems Handbook, Springer US, 2011.

[2] Aggarwal CC, Wolf JL, Wu K-L and Yu PS, "Horting hatches an Egg: A new graph-theoretic approach to collaborative filtering", In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1999

[3] Anand, D. and K. K. Bharadwaj, "Adaptive user similarity measures for recommender systems: a genetic programming approach", in Proceedings of Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference, vol.8, 2010.

[4] Basu C, Hirsh H and Cohen WW, "Recommendation as classification: Using social and content-based information in

recommendation". In: RichCand MostowJ, eds., Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98). AAAI Press, Menlo Park, CA, pp. 714–720, 1998.

[5] Billsus D and Pazzani MJ, "Learning collaborative information filters" In: Rich C and Mostow J, eds., Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98). AAAI Press, MenloPark, CA, pp. 46–53, 1998.

[6] Delgado J and Ishii N, "Memory-based weighted majority prediction for recommender systems" In SIGIR Workshop on Recommender Systems, 1999.

[7] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan, "Collaborative Filtering Recommender Systems, Now Publishers Inc, 2011.

[8] Goldberg, K., T. Roeder, D. Gupta, and C. Perkins. Eigentaste, "A constant-time collaborative filtering algorithm", Information Retrieval, vol.4, no. 2, pp.133–151,2001.

[9] Herlocker, Jonathan, Joseph A. Konstan, and John T. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms." Information retrieval vol.5, no. 4, pp. 287-310, 2002.

[10] Hofmann T and Puzicha J, "Latent class models for collaborative filtering", In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 688–693, 1999.

[11] Karypis, George. "Evaluation of item-based top-n recommendation algorithms", In Proceedings of the tenth international converence on information and knowledge management, ACM, pp.247-245, 2001.

[12] Lang, K. "Newsweeder: Learning to filter netnews", In Proceedings of the 12t International Conference on Machine Learning, Tahoe City, Calif., USA, 1995

[13] Lee, Joonseok, Mingxuan Sun, and Guy Lebanon, "A comparative study of collaborative filtering algorithms", arXiv preprint arXiv Report, 1205.3193, 2012.

[14] Linden, G., B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering", IEEE Internet Computing, vol.7, no. 1, pp.76-80, 2003.

[15] Manouselis N, Costopoulou, C. Nikos, "Experimental Analysis of Design Choices in Multiattribute Utility Collaborative Filtering", International Journal of Pattern Recognition and Artificial Intelligence, vol.21, no. 2, pp.311-331, 2007.

[16] Miller, B. N., I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl., "MovieLens unplugged: experiences with an occasionally connected recommender system", In Proceedings of the 8th international conference on Intelligent user interfaces, pp.263-266, 2003.

[17] Mooney, R. J. and L. Roy, "Content-based book recommending using learning for text categorization", In Proceedings of ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, pp.195-204, 1999.

[18] Pazzani, M. and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites", Machine Learning, vol.27, no. 3, pp.313-331, 1997.

[19] Poompuang P. and Wichian P., "User and Item Pattern Matching in Multi-criteria Recommender Systems", In proceedings of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD), pp.20-25, 2010.

[20] Premchaiswadi, W., and Pitaya P. "Hybrid Profiling for Hybrid Multicriteria Recommendation Based on Implicit Multicriteria Information", Applied Artificial Intelligence , Taylor & Francis , vol.27, no. 3, pp.213-234, 2013.

[21] Resnick, P, N Iacovou, M Suchak, P Bergstrom, and J Riedl. "GroupLens: an open architecture for collaborative filtering of netnews", In Proceedings of the ACM conference on Computer supported cooperative work, pp:175-186, 1994.

[22] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth", In Proceedings of the ACM Conference on Human Factors in Computing Systems, 1995.

[23] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J., "Analysis of recommendation algorithms for e-commerce", In proceedings of the 2nd ACM conference on Electronic commerc, 2000.