An Improved Similarity Metric for Recommender Systems

SAMIYAH AL-ANAZI
Department of Software Engineering
King Saud University
Riyadh
SAUDI ARABIA
samiahal3nazi@gmail.com

PANDIAN VASANT
Department of Fundamental and Applied Sciences
Universiti Teknologi PETRONAS
Perak Darul Ridzuan
Malaysia
pandian_m@petronas.com.my

M. ABDULLAH-AL-WADUD*
Department of Software Engineering
King Saud University
Riyadh
SAUDI ARABIA
mwadud@ksu.edu.sa

Abstract: - Due to pervasive technologies in various applications, which are used in our everyday lives, recommender systems have become widely used in most of these applications to estimate the users' needs depending on his/her preferences. The development of recommendation methods typically focuses on maximizing the prediction accuracy of the users' interests. Currently, collaborative filtering (CF) is a widely used approach for recommender systems. The similarity measures play a major role in such recommender systems. In spite of the availability of many different similarity measures, user similarity is yet to be calculated perfectly in recommender systems. We propose a similarity metric that helps to increase the accuracy of recommended items.

Key-Words: - Recommender system, Collaborative filtering, Content-based filtering, Similarity, Pearson correlation

* Corresponding author

1 Introduction

Day by day, the amount of information that is available on the Internet has been increasing, leading to information overload. This has become a major problem for users; they must investigate what they are looking for or what they are interested in. Recommender systems can help resolve this issue. Recommender systems are software tools and techniques that suggest useful items in academic and commercial fields [1]. To recommend items to a given user, the system needs to collect user preference information. Depending on the type of information sought, a variety of approaches are used to generate recommendations [6, 8]. These include collaborative filtering, which bases its predictions and recommendations on the ratings or behaviours of other users in the system [7]. There are two types of collaborative filtering: 1) user-based CF [12] and 2) item-based CF [13]. Both types identify the nearest neighbourhood algorithm (NN) of the active user [5]. In contrast, content-based filtering; recommends items based on the content of items versus how other users rate them [2, 7]. It relies on two type of data 1) set of users and 2) set of categories that have been assigned to the items [1]. Finally, hybrid filtering [10] is a combination of two or more recommendation techniques that work together to achieve better system optimization.

One of the most important parts of a recommender system is the user similarity. There are various similarity metrics can be used in different implementations to calculate the similarity between two pair of users [9]. The common metrics used in recommender systems are: Pearson correlation-based similarity, Euclidean distance-based similarity, cosine measure similarity and log-

likelihood test. Therefore, the randomness of a system's ranking leads to inaccurate recommendations and subsequent reductions in system quality.

In [4], the authors proposed the novel Bayesian similarity measure for recommender systems based on the Dirichlet distribution, taking into account both the direction and length of rating vectors. In addition, correlations due to chance and user bias were removed to accurately measure users' correlation. The efficient and the performance of recommender systems changes depending on the user similarity metric that used on build the system [3].

In this paper, we focus on collaborative filtering to calculate the user similarities (user-based and item-based) based on common rating items between users based on Pearson correlation similarity in an improved method.

2 Proposed Method

In this section, we first present a general architecture of user similarity-based the recommender systems, followed by the Pearson correlation, a well-accepted similarity measure. Finally, we point out some shortcomings of this measure and present our proposed similarity metric in detail.

Algorithm 1. General outline of user-based collaborative filtering

Input User ratings of items, user *u*
Output A list of *r* items recommended for *u*
Procedure
for all users *w* other than *u* **do**
Compute user similarity (*s*) between *u* and *w*
end

Retain the top *n* users, ranked by similarity(*s*)

for all items *i* that has a preference by any user in *n*, but u has no preference **do**
for every other user *v* in *n* having a preference for *i* **do**
Compute a similarity *s* between *u* and *v*
Combine *v*'s preference for (i), weighted by s, into a running average
end
end
Recommend top *r* item based on the running average

2.1 General process of collaborative filtering

The process of recommending items using collaborative filtering (user-based) in recommender system is based on Algorithm 1 [14].

2.2 Pearson correlation similarity

Pearson correlation is one of the most widely used measure in data mining and recommendation engines. It computes the statistical correlation between two users' common ratings to determine their similarity [11] as follows:

$$sim(u,v) = \frac{\sum_{i\in S(u,v)}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in S(u,v)}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in S(u,v)}(r_{vi}-\bar{r}_v)^2}} \quad (1)$$

where *u* and *v* are a pair of users, r_{ui}, r_{vi} are the rating of item I rated by u and v, respectively. $S_{(u,v)}$ is the item-set that users u and v have both rated in common and $\bar{r}_u$ and $\bar{r}_v$ are the average of non-zero ratings of the two users.

2.3 The Proposed Metric

The Pearson correlation focuses only on the ratings of the items that are rated by both the users. It does not take into account the other items rated by them. Moreover, the similarity may not always reflect the true correlation as it may give the same similarity score when different numbers of item are rated by two users.

Suppose that users *u* and *v* have rated *k* items in common, based on which we get the Pearson similarity *sim*(*u*, *v*) = *s*. Again, suppose that users *x* and *y* have rated *j* items, where $j < k$, in common, based on which we also get the Pearson similarity *sim*(*x*, *y*) = *s*. In these two cases, even though the Pearson correlation is giving the same score, general intuition is that u and v are proven to be more stable correlated than x and y since u and v has rated more items and maintained the correlation. Hence, the number of common items rated by the two users should be incorporated to the similarity score to reflect a better similarity measure.

Again, suppose the number of total items rated by a user also gives a clue to the similarity value. When we find the same number of common rated items are found with fewer number of total items rated, it shows a higher possibility of similarity. On the other hand, if the number of total items rated is higher, it means that the similarity is not that high (because to have k common items many items were needed to be rated).

To reflect both the issues described above, we propose to scale up the Pearson similarity score by the total number of common items and discount the score by the total number of item rated by a user. Finally, our proposed Modified SIMilarity metric (MSIM) is given by

$$msim(u,v) = \frac{\frac{k}{t_u} sim(u,v) + \frac{k}{t_v} sim(u,v)}{2} \quad (2)$$

which simplifies to

$$msim(u,v) = \left(\frac{1}{t_u} + \frac{1}{t_v}\right)\frac{k}{2} sim(u,v) \quad (3)$$

where t_u and t_v are the total number of items rated by users *u* and *v*, respectively.

3 Experimental Evaluation

In this section, we evaluate our proposed MSIM and compare it with the traditional Pearson correlation matric respectively using real world dataset.

3.1 Dataset

To evaluate our method, we used experimental data from MovieLens dataset. This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. We randomly picked 70% data for training and the rest for testing.

3.2 Experimental Results

Recommender systems researchers have used several types of measures for evaluating the quality of a recommender system. In our paper, we use the Average Absolute Difference (AAD) evaluator, which calculates the average difference between the actual and estimated preferences. A low AAD value means that the estimated preferences do not differ much from the actual (ground truth) preferences. AAD = 0 indicates perfect recommendations.

Table 1 The Average Absolute Difference (AAD) values obtained by the recommender system when different similarity metrics are employed

Number of recommended items	Pearson Correlation	MSIM
2	0.9877049	0.8122535
4	0.9621144	0.8447912
8	0.9340911	0.8361041
17	0.9231780	0.8187648
20	0.9175603	0.8145739
24	0.9063748	0.8116835
37	0.8872837	0.8011758

From Table 1, we can conclude that our method is more accurate than the traditional Pearson correlation matric.

4 Conclusion

In this paper, we first studied similarity metric between users based on ratings to improve recommendation quality. Actually, the traditional Pearson correlation is the popularity and efficiently used in recommender systems. On the other hand, our proposed matric provided a high quality recommendation than Pearson correlation.

References:

[1] F. Ricci, L. Rokach and B. Shapira, Introduction to recommender systems handbook, in *Recommender Systems Handbook*, 1st ed., F. Ricci, L. Rokach, B. Shapira and P. Kantor, Ed. New York: Springer-Verlag New York, 2010, pp. 1-35.

[2] S. Jain, A. Grover, P. Thakur and S. Choudhary, Trends, Problems And Solutions of Recommender System, in *International Conference on Computing, Communication and Automation (ICCCA2015)*, 2015.

[3] Bellogin A, de Vries AP, Understanding similarity metrics in neighbour-based recommender systems, In *Proceedings of the 2013 conference on the theory of information retrieval (ICTIR '13)*, ACM, New York, USA, 2013, pp 48–55

[4] G. Guo, J. Zhang, and N. Yorke-Smith, A novel bayesian similarity measure for recommender systems, in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[5] M. Robillard and R. Walker, An Introduction to Recommendation Systems in Software Engineering', in *Recommendation Systems in Software Engineering*, 1st ed., M. Robillard, W. Maalej, R. Walker and T. Zimmermann, Ed. Software, IEEE (Volume:27 , Issue: 4): IEEE, 2010, pp. pp: 1-11.

[6] R. Prasad, A Categorical Review of Recommender Systems, *International Journal of Distributed and Parallel systems*, vol. 3, no. 5, 2012, pp. 73-83.

[7] J. Schafer, D. Frankowski, J. Herlocker and S. Sen, Collaborative Filtering Recommender Systems, in *The Adaptive Web: Methods and Strategies of Web Personalization*, 1st ed., P. Brusilovsky, A. Kobsa and W. Nejdl, Ed.

Springer Berlin Heidelberg, 2007, pp. pp: 291-324.

[8] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, 2005, pp. 734-749.

[9] L. Zhongduo, Indoor Location-Based Recommender System, *Master's thesis*, University of Toronto, Department of Electrical and Computer Engineering, August 2013.

[10] S. Spiegel, *A Hybrid Approach to Recommender Systems based on Matrix Factorization*, Technical University Berlin, Department for Agent Technologies and Telecommunications, 2009.

[11] S. Bouhali, A. H and M. Aïcha, Handling preferences under uncertainty in recommender systems, in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, 2014, pp. 2262 - 2269.

[12] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl, GroupLens: applying collaborative filtering to Usenet news, *Communications of the ACM*, vol. 40, no. 3,1997, pp. 77-87.

[13] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Item-based collaborative filtering recommendation algorithms, in *Proceedings of the International Conference on the World Wide Web*, 2001, pp. 285–295.

[14] S. Owen, R. Anil, T. Dunning, and E. Friedman. *Mahout in action*, 2010.