Metadata of the chapter that will be visualized in SpringerLink

Book Title	Analysis of Images, Social Networks and Texts	
Series Title		
Chapter Title	Similarity Aggregation for Collaborative Filtering	
Copyright Year	2015	
Copyright HolderName	Springer International Publishing Switzerland	
Author	Family Name	**Sarwar**
	Particle	
	Given Name	**Sheikh Muhammad**
	Prefix	
	Suffix	
	Division	Institute of Information Technology
	Organization	University of Dhaka
	Address	Dhaka, Bangladesh
	Email	smsarwar@du.ac.bd
Author	Family Name	**Hasan**
	Particle	
	Given Name	**Mahamudul**
	Prefix	
	Suffix	
	Division	Department of Computer Science and Engineering
	Organization	University of Dhaka
	Address	Dhaka, Bangladesh
	Email	munna@gmail.com
Author	Family Name	**Billal**
	Particle	
	Given Name	**Masum**
	Prefix	
	Suffix	
	Division	Department of Computer Science and Engineering
	Organization	University of Dhaka
	Address	Dhaka, Bangladesh
	Email	billalmasum93@gmail.com
Corresponding Author	Family Name	**Ignatov**
	Particle	
	Given Name	**Dmitry I.**
	Prefix	
	Suffix	
	Division	
	Organization	National Research University Higher School of Economics
	Address	Moscow, Russia

Email	dignatov@hse.ru

Abstract	In this paper we show how several similarity measures can be combined for finding similarity between a pair of users for performing Collaborative Filtering in Recommender Systems. Through aggregation of several measures we find super similar and super dissimilar user pairs and assign a different similarity value for these types of pairs. We also introduce another type of similarity relationship which we call medium similar user pairs and use traditional JMSD for assigning similarity values for them. By experimentation with real data we show that our method for finding similarity by aggregation performs better than each of the similarity metrics. Moreover, as we apply all the traditional metrics in the same setting, we can assess their relative performance.
Keywords (separated by '-')	Recommender Systems - Collaborative Filtering - Similarity measures - Similarity fusion

Similarity Aggregation for Collaborative Filtering

Sheikh Muhammad Sarwar[1], Mahamudul Hasan[2], Masum Billal[2], and Dmitry I. Ignatov[3](✉)

[1] Institute of Information Technology, University of Dhaka, Dhaka, Bangladesh
smsarwar@du.ac.bd
[2] Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh
{munna,billalmasum93}@gmail.com
[3] National Research University Higher School of Economics, Moscow, Russia
dignatov@hse.ru

AQ1

Abstract. In this paper we show how several similarity measures can be combined for finding similarity between a pair of users for performing Collaborative Filtering in Recommender Systems. Through aggregation of several measures we find super similar and super dissimilar user pairs and assign a different similarity value for these types of pairs. We also introduce another type of similarity relationship which we call medium similar user pairs and use traditional JMSD for assigning similarity values for them. By experimentation with real data we show that our method for finding similarity by aggregation performs better than each of the similarity metrics. Moreover, as we apply all the traditional metrics in the same setting, we can assess their relative performance.

Keywords: Recommender Systems · Collaborative Filtering · Similarity measures · Similarity fusion

1 Introduction

Recommendation is a social process through which people close to a target user suggest her movies, songs, food etc. However, this social process has become a prevalent component in the virtual world as well because of the tremendous growth of information in the World Wide Web. Unlike social recommendation process, recommendation in the virtual world is rather implicit. It means that people do not directly get suggestion from their peers, rather a computational process helps to generate recommendations for them by automatically identifying a cluster of people who behave similarly. Naturally, a person takes recommendations or suggestions from another person if they both have similar choices or preferences. But, in the virtual world we have access to the preference of millions of users and hence it is possible to get recommendation as a service by assessing similarity of a specific user and a group of users computationally. In the literature this process is referred to as collaborative filtering.

M.Y. Khachay et al. (Eds.): AIST 2015, CCIS 542, pp. 1–7, 2015.
DOI: 10.1007/978-3-319-26123-2_23

One of the major tasks of a Recommender System (RS) is a prediction, i.e. a process through which a RS predicts the rating of a specific item for a user. Rating scale can vary in different ways for different systems. Usually, a rating scale takes integer values from 1 to 5 or from 1 to 10. So, two entities are associated with a rating; one is the user and the other is an item. When a system is being used by several users and consists of several items, a user-item matrix holding the rating data for all the items can be formed. This matrix is the major source for finding similarity between different users in the system. So, the basic philosophy is to analyze the previous ratings of two users and based on these values try to asses similarity of the users' preferences and use that to predict ratings for items which have not yet been rated. The most notable part of CF algorithms refers to the group of metrics used to determine the similarity between each pair of users, among which the Pearson Correlation Coefficient (PCC) is one of the most popular similarity measures [1].

Apart from PCC, there are several similarity measures having inherent advantages and drawbacks. Popular methods include cosine similarity, constrained Pearson correlation coefficient (CPCC), sigmoid function based Pearson correlation coefficient (SPCC), adjusted cosine measure (ACOS), Jaccard similarity and mean squared differences (MSD) [2]. Furthermore, Jaccard and MSD can be combined by multiplication to form a new measure, which is referred to as JMSD [2]. In this paper we hypothesize that to get the most out of the measures we need to combine them in some way as all do not perform well in different situations. Specifically, we state the importance of using different measures for computing similarity of different user pairs. Practically it is rather hard to develop a working heuristic to select a proper similarity measure for a specific user pair. In order to achieve this goal to some extent, we introduce the notion of support; it is defined as the number of measures endorsing the similarity relation between two users. We specifically handle the cases where the relation between a couple of users have high support, low support or average support. As a result, we do not specifically develop a new measure, rather we show how to reap the benefits of existing measures to design an approach which performs better than each of them.

2 Proposed Method

In our experimentation we have used 8 different similarity measures; PCC, SPCC, CPCC, ACOS, COS, JMSD, MSD and Jaccard. All of them are described in section. There are many papers on these measures reporting their individual performances in various tasks [3], but in this paper we implement all the metrics individually under the same experimental setup and report their MAE (Mean Absolute Error). MAE determines the accuracy of recommendations by defining the average absolute deviation between the system's predicted rating against the actual rating assigned by the user [4]. A lower MAE value corresponds to a higher recommendation accuracy. Given the set of actual/predicted pairs $(r_{u,i}, p_{u,i})$ for all the movies (M_u) rated by user u, the MAE for user u is computed as:

$$MAE = \frac{\sum_{i \in M_u} |r_{u,i} - p_{u,i}|}{|M_u|}. \tag{1}$$

2.1 Computing Support Matrix

Using 8 similarity measures in total, we calculate a support value performing the following steps:

1. For a single measure, we calculate the similarity between every pair of users.
2. Then we calculate the median from this similarity measure among all user pairs.
3. Using the median as a threshold, we classify the whole similarity space into two binary classes 0 and 1. Values higher than the median fall into class 1, while the rest fall to class 0.
4. Now, we introduce the notion of support. We assert that if the similarity class of two users is 1, then their similarity relation is supported by the measure we have used to compute similarity. Hence, we increment the support count for that pair of users by 1.
5. We continue this process for the all eight matrices and increment the support value of those two users who satisfy the rule above.
6. Finally, as an outcome of this process, we retrieve a user by user support matrix $S \in \mathbb{R}^{n \times n}$, where n is the number of users in the system, and $S_{uv} \in \{0, 1, \ldots, 8\}$.

2.2 Finding Super Similar, Average Similar and Super Dissimilar User Pairs

Now, we introduce the notion of super similar, medium similar, and super dissimilar users using our support matrix S.

Definition 1. *Super Similar Users: If $S_{uv} \geq 5$, for a pair of users u and v, then we denote them as super similar users.*

Definition 2. *Super Dissimilar Users: If $S_{uv} \leq 2$, for a pair of users u and v, then we denote them as super dissimilar users.*

Definition 3. *Medium Similar Users: Relationship classes that neither belong to super similar or super similar falls into the classes for medium similar, that is $S_{uv} \in \{3, 4\}$.*

The threshold for choosing super similar user pairs comes from an empirical analysis, which is shown in Table 1. In the table, we show MAE values for different support settings. In order to find out a proper value of support for super similar users, we enumerate the values of support from greater than or equal to 0 to greater than or equal to 8 and check the MAE. We set user-user similarity value as 1 (*i.e.* we make them super similar) for a specific set of support values (for example, greater than or equal to 5), and we set 0 for all the support values below that specific set of support values. As a result, only super similar users are having a full influence on each other, while other users who are not super similar do not have any effect. It indicates that a user-user pair for which we have support value less than a specific threshold value are totally unrelated. We can see that for support value being greater than or equal to 5 the respective MAE value is comparatively lower. The plot, which is based on the table and shown in Fig. 1, makes more sense since it explains the reasoning for our definition of super similar users in Sect. 2.1. However, when super similar users have support value greater than or equal to 5, super dissimilar users should have support value equal or less than 4. But, here we make a finer distinction and define medium similar and super dissimilar users for better performance of the system.

Table 1. MAE values for different support thresholds of super similar users

Minimal support	0	1	2	3	4	5	6	7	8
MAE	0.6883	0.6875	0.6870	0.6822	0.6758	**0.6731**	0.6744	0.6937	0.7679

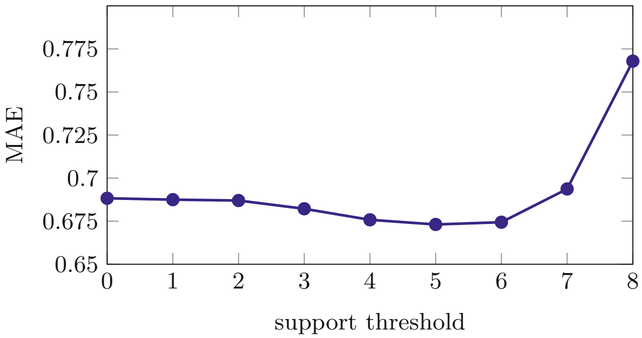

Fig. 1. A graph for finding the optimal similarity values for super similar and super dissimilar users

2.3 Prediction Function

Our prediction function is typical for collaborative filtering; however, it is based on similarity defined in our own way. To calculate the predicted rating p_u^i for user u of an item i, the following Deviation From Mean (DFM) as aggregation approach is used [4]:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in N_u} sim(u,v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in N_u} sim(u,v)} \tag{2}$$

In Eq. 2, N_u is a set of k most similar users to a given user u,$\bar{r}_u$ represents the average of ratings made by the given user u and $\bar{r_v}$, $r_{v,i}$ are the average of ratings and rating of item i made by the neighbor v, respectively. In Eq. 2 we set $sim(u,v) = 0.9$ for $S_{uv} \geq 5$ and $sim(u,v) = -0.3$ for $S_{uv} \leq 2$. Finally, we set

$sim(u, v) = JMSD(u, v)$ if $S_{uv} = 3$ or $S_{uv} = 4$. Now, we describe the reasoning behind the usage of the aforementioned values.

In Fig. 2 we show the MAE values we obtain for setting different values for super dissimilar users keeping the similarity value for super similar users constant. If we observe the graphs closely, we can see that MAE comes down to the lowest value and then rises. Moreover, we can see that if we take –0.3 as the similarity value for all the super dissimilar users and 0.9 as similarity value for all super similar users, it results in a good MAE.

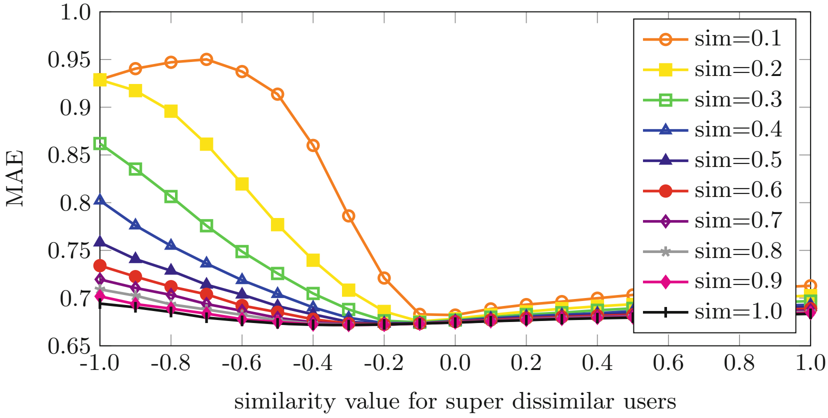

Fig. 2. MAE curves for different similarity values of super dissimilar users parametrized by super similarity values (see the legend)

3 Experimental Result

We have tested our hypothesis using MovieLens dataset. We used the training data with 80 % of the available ratings and 20 % of the rating data was set as the test set. Details of the dataset and testing procedure can be found in [5].

Table 2. MAE values for 8 different similarity measures

PCC	SPCC	CPCC	ACOS	COS	JMSD	MSD	JACCARD
0.688	0.687	0.685	0.687	0.687	**0.680**	0.688	0.682

In Table 2, we show the MAE values for all the measures implemented by us and we can see that JMSD performs better than all the other metrics. However, in Table 3 we show that the proposed approach – super similar (with similarity 0.9) combined with average user (with the same similarity as JMSD value) and super dissimilar (with similarity –0.3) performs better than JMSD. We also show the performance JMSD combined with super similar and super dissimilar users respectively. Note that for all the metrics, including ours, we multiply a confidence value with similarity value as multiplying confidence produces better

Table 3. MAE values for different combinations of similarity ranges

Super similar ($\geq$5) + super dissimilar (<5) (no medium similarity)	Super similar + JMSD	Super dissimilar + JMSD	JMSD	Super similar + medium similar + super dissimilar
0.673	0.675	0.735	0.680	**0.668**

result for all the metrics. More details on confidence value can be found in [6], but we provide its Formula 3 below:

$$conf(u, v) = \frac{|I_u \cap I_v|}{|I_v|}. \tag{3}$$

Here, $|I_u \cap I_v|$ is the number of common ratings between user u and user v, and $|I_v|$ is the number of assigned ratings by user v.

4 Conclusion

This paper is our initial footstep of proving the fact that a specific metric or similarity value might be suitable for a specific set of users. Here we performed our experimentation using three groups of user-user pairs: super similar, medium similar and super dissimilar. We show through the experimentation that among the existing metrics JMSD outperforms others in terms of MAE. However, our hybrid approach by aggregation outperforms JMSD using the the same measure.

Since we had a look only at user-based measures, the important venue of our future work could be similarity fusion with the item-based measures. In fact, our heuristic approach is performed better in terms of MAE than similarity fusion based approach of that type reported in [7]. We hope that to this end we can use similarity measures from Formal Concept Analysis to exploit interplay between objects (users) and items (attributes) of the proposed support matrix [8].

Acknowledgment. The first three authors were partially supported by their university. The last author was partially supported by the Russian Foundation for Basic Research grants no. 13-07-00504 and 14-01-93960 and made a contribution within the project "Data mining based on applied ontologies and lattices of closed descriptions" supported by the Basic Research Program of the National Research University Higher School of Economics. We also deeply thank the reviewers and Konstantin Vorontsov for their comments and remarks that helped.

References

1. Ortega, F., Sánchez, J.L., Bobadilla, J., Gutiérrez, A.: Improving Collaborative Filtering-based recommender systems results using Pareto dominance. Inf. Sci. **239**, 50–61 (2013)

2. Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X.: A new user similarity model to improve the accuracy of Collaborative Filtering. Knowl. Based Syst. **56**, 156–166 (2014)
3. Ahn, H.J.: A new similarity measure for Collaborative Filtering to alleviate the new user cold-starting problem. Inf. Sci. **178**(1), 37–51 (2008)
4. Bobadilla, J., Ortega, F., Hernando, A., Alcalá, J.: Improving Collaborative Filtering recommender system results and performance using genetic algorithms. Knowl. Based Syst. **24**(8), 1310–1316 (2011)
5. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce, EC 2000, New York, USA, pp. 158–167, ACM (2000)
6. Kaleroun, A.: Hybrid bee colony trust mechanism in recommender system. Ph.D. thesis, Thapar University (2014)
7. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 501–508 (2006)
8. Eklund, P.W., Ducrou, J., Dau, F.: Concept similarity and related categories in information retrieval using formal concept analysis. Int. J. Gen. Syst. **41**(8), 826–846 (2012)

Author Queries

Chapter 23

Query Refs.	Details Required	Author's response
AQ1	Kindly note that "compiled.tex" has been followed. Please check and confirm.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	— under matter to be changed	
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	~ under matter to be changed	~
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	| between characters or words affected	