Improving Jaccard Index for Measuring Similarity in Collaborative Filtering

Soojung Lee(✉)

Gyeongin National University of Education,
155 Sammak-ro, Anyang, Republic of Korea
sjlee@gin.ac.kr

Abstract. In collaborative filtering-based recommender systems, items are recommended by consulting ratings of similar users. However, if the number of ratings to compute similarity is not sufficient, the system may produce unreliable recommendations. Since this data sparsity problem is critical in collaborative filtering, many researchers have made efforts to develop new similarity metrics taking care of this problem. Jaccard index has also been a useful tool when combined with existing similarity measures to handle data sparsity problem. This paper proposes a novel improvement of Jaccard index that reflects the frequency of ratings assigned by users as well as the number of items co-rated by users. Performance of the proposed index is evaluated through extensive experiments to find that the proposed significantly outperforms Jaccard index especially in a dense dataset and that its combination with a previous similarity measure is superior to existing measures in terms of both prediction and recommendation qualities.

Keywords: Recommender system · Similarity measure · Collaborative filtering · Memory-based collaborative filtering · Jaccard index

1 Introduction

Internet users are very often overwhelmed by the amount of information provided by the web. A popular method to solve this problem is the recommender system. This system is usually utilized in commerce to recommend products that might be preferred by customers. Collaborative filtering (CF) is a well-known type of implementation of a recommender system. It refers to other likeminded users and recommends items which have been highly rated by them. This filtering method has been successful in many systems such as GroupLens, Ringo, and Amazon.com [1].

As CF basically performs by incorporating ratings of similar users, determination of similar users is a critical aspect of the CF system. Several approaches have been developed to calculate similarity. They are usually classified into correlation-based and vector cosine-based [1,2]. However, similarity calculation is based on the history of ratings of users, thus insufficient amount of rating data

K. Kim and N. Joukov (eds.), *Information Science and Applications 2017*,
Lecture Notes in Electrical Engineering 424, DOI 10.1007/978-981-10-4154-9_93

in the system often producing unreliable similar users. This problem, known as *data sparsity*, is fundamental due to the principle of CF. Detailed analysis of drawbacks resulting from the data sparsity problem of traditional similarity measures can be found in [3,4].

Various techniques in literature have addressed the data sparsity problem of CF, while the simplest technique to compute similarity may be the incorporation of Jaccard index into the previous similarity measure [4–8]. Jaccard index reflects the number of common items rated by two users. As Jaccard index becomes an important component of measuring similarity and reportedly improves performance of CF, this paper focuses on this index and proposes a novel improvement. To verify its novelty, we conducted extensive experiments using two datasets with vey different characteristics. The results state that the proposed index outperforms Jaccard index on both datasets. Especially, the degree of improvement is higher on a denser dataset. Furthermore, the proposed index is incorporated into a previous measure to produce a new similarity measure which proves to perform the best among or comparably to existing measures experimented.

2 Related Work

Jaccard index measures the proportion of the number of items commonly rated by two users out of the total number of items rated by them [7]. Let I_u be the set of items rated by user u and $|I_u|$ be its cardinality. Then Jaccard index between users u and v is calculated as follows.

$$Jaccard(u,v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|}.$$

As seen from the above formula, Jaccard index does not take the ratings into account, but only considers relative number of common items. Hence, it seems improper to use the index to estimate similarity between two users.

Several researchers propose new similarity measures that incorporate Jaccard index into traditional measures [4,5,9]. These approaches surely compensate the defects of previous similarity measures which are mainly caused by data sparsity or cold-start users. Bobadilla et al. proposed a new similarity measure that combines mean squared differences with Jaccard index [5]. Saranya et al. calculate similarity by incorporating Pearson correlation and Jaccard index which is reported to achieve a little improvement in recommendation quality [4]. In the meantime, a measure named UOD (Uniform Operator Distance) is suggested by Sun et al. [9]. This measure is combined with Jaccard index to better estimate similarity between two users. The authors report that their combined similarity measure called *JacUOD* leads to better prediction quality than when Jaccard index is combined with Pearson correlation. We examined performance of JacUOD through several experiments, whose results are presented in Sect. 4. The formula of JacUOD is defined as follows. Let $r_{u,i}$ be the rating of item i given by user u and $r_{u,max}$ be the maximum rating given by u. Also let $m = |I_u \cap I_v|$.

Then

$$UOD(u,v) = \begin{cases} \frac{\sqrt{m(r_{u,max}-r_{u,min})^2}}{0.9+\sqrt{\sum_{i \in I_u \cap I_v}(r_{u,i}-r_{v,i})^2}}, & \text{if } r_{u,i} = r_{v,i} \text{ for all } i \\ \frac{\sqrt{m(r_{u,max}-r_{u,min})^2}}{\sqrt{\sum_{i \in I_u \cap I_v}(r_{u,i}-r_{v,i})^2}}, & \text{otherwise.} \end{cases}$$

$$JacUOD(u,v) = Jaccard(u,v) \times UOD(u,v)$$

3 Proposed Index

3.1 Motivation

The idea of our study is based on the work by [5]. This work examines the mean and deviation of the ratings of MovieLens (http://www.movielens.org) and NetFlix datasets (http://www.netflixprize.com). Within the integer range of [1..5] allowed in these datasets, it is found that users tend to give ratings higher than the median and avoid the extreme values. The highest frequency of ratings is associated with the rating of four, followed by the rating of three. It is also found that the standard deviation of ratings is seldom larger than 1.2. This result implies that two users giving a same extreme rating can be treated as more similar than those giving a more common rating.

The above observation motivated our research to improve Jaccard index. This index calculates the number of commonly rated items by two users, regardless of the rating values. However, as discussed above, it is worth considering that the rating of a common item is normal or extreme.

3.2 Formulation of the Index

We are interested in how many items are commonly rated with normal or extreme values. Hence, in our index, the rating range allowed in the system is divided into three sub-intervals, within each of which Jaccard index is computed separately. Specifically, let L_{bd} and H_{bd} be boundaries of the sub-intervals, where $L_{bd} < H_{bd}$. That is, when $[r_L, r_H]$ represents the range, $r_L < L_{bd} < H_{bd} < r_H$. We divide the set of items rated by user u, I_u, into three sets as follows, based on the rating values assigned by u.

$$I_{L,u} = \{i \in I_u | r_{u,i} \leq L_{bd}\},\ I_{M,u} = \{i \in I_u | L_{bd} < r_{u,i} < H_{bd}\},\ I_{H,u} = \{i \in I_u | r_{u,i} \geq H_{bd}\}.$$

Then three types of Jaccard indexes between users u and v are defined as follows.

$$Jac_L(u,v) = \frac{|I_{L,u} \cap I_{L,v}|}{|I_{L,u} \cup I_{L,v}|},\ Jac_M(u,v) = \frac{|I_{M,u} \cap I_{M,v}|}{|I_{M,u} \cup I_{M,v}|},\ Jac_H(u,v) = \frac{|I_{H,u} \cap I_{H,v}|}{|I_{H,u} \cup I_{H,v}|}$$

Finally, our metric, named as *JacLMH*, is calculated as an arithmetic average of the three Jaccard indexes as follows.

$$JacLMH(u,v) = \frac{1}{3}(Jac_L(u,v) + Jac_M(u,v) + Jac_H(u,v))$$

Table 1. Characteristics of the datasets

	Matrix size (users × ratings)	Rating scale	Sparsity level
MovieLens	1000×3952	1∼5 (integer)	0.9607
Jester	998×100	$-10 \sim +10$ (real)	0.2936

4 Performance Experiments

4.1 Experiments Plan

We conducted extensive experiments using two popular datasets with very different characteristics, as presented in Table 1. Sparsity level represents how sparse the dataset is. It is defined by 1-(total number of ratings/matrix size).

The baseline similarity measures of our experiments are Jaccard index (Jaccard), Pearson correlation (PCC), JacUOD, UOD, the proposed JacLMH, and JacLMH×UOD (JLMHUOD). The last measure is experimented to compare the degree of improvement made by incorporating JacLMH instead of Jaccard into UOD. We adopted five-fold cross validation [10] to obtain more reliable results, where the ratio of training and testing data is set to 80:20 for each experiment.

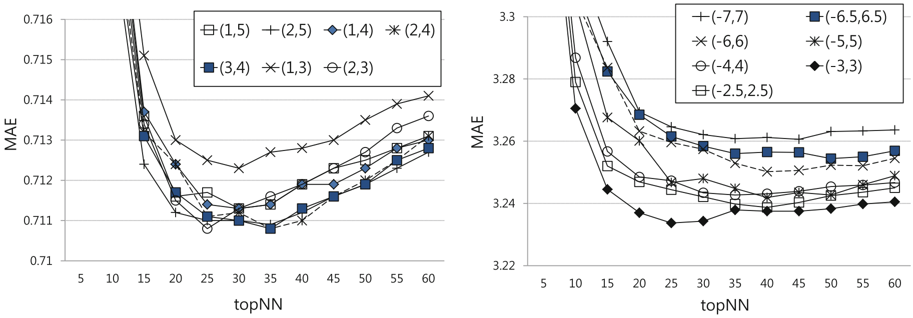
Performance is evaluated based on two well-known standards in related studies, prediction quality and recommendation quality. MAE (Mean Absolute Error) is usually used to measure prediction quality, which is the mean difference between the predicted rating of an unrated item and its corresponding real rating. The rating prediction is typically made by referring to ratings of users similar to the current user, while weights are imposed according to the degree of similarity. Recommendation quality is usually measured by precision and recall metrics or their harmonic mean F1 [11]. We employed only F1 due to the space constraint.

4.2 Effect of Bounds

To examine how L_{bd} and H_{bd} parameters used in our index affect performance, we measured MAE with various combinations of these parameter values. Figure 1 shows the results with two datasets. With MovieLens, it seems that (1,3) is definitely worst, while others are almost competitive. In particular, MAE results using (2,4), (2,5), and (3,4) are virtually no different with one another. Hence, we chose (2,5) for ensuing experiments of our metric. With Jester, the results using different parameter values are more distinguishable than with MovieLens. We used (−3,3) yielding the obviously lowest MAE in our further experiments.

4.3 Performance Results

Figure 2 shows performance results with MovieLens with varying number of nearest neighbors (topNN). To view any performance improvement of our index



(a) (b)

Fig. 1. MAE results with varying bounds of (L_{bd}, H_{bd}) pairs: (a) MovieLens (b) Jester

over Jaccard index more clearly, two metrics, Jaccard and JacLMH, are provided together, separately from the others. It is observed that JacLMH yields better MAEs consistently, implying that our idea of separate application of Jaccard index to sub-ranges of ratings proves successful with respect to prediction accuracy.

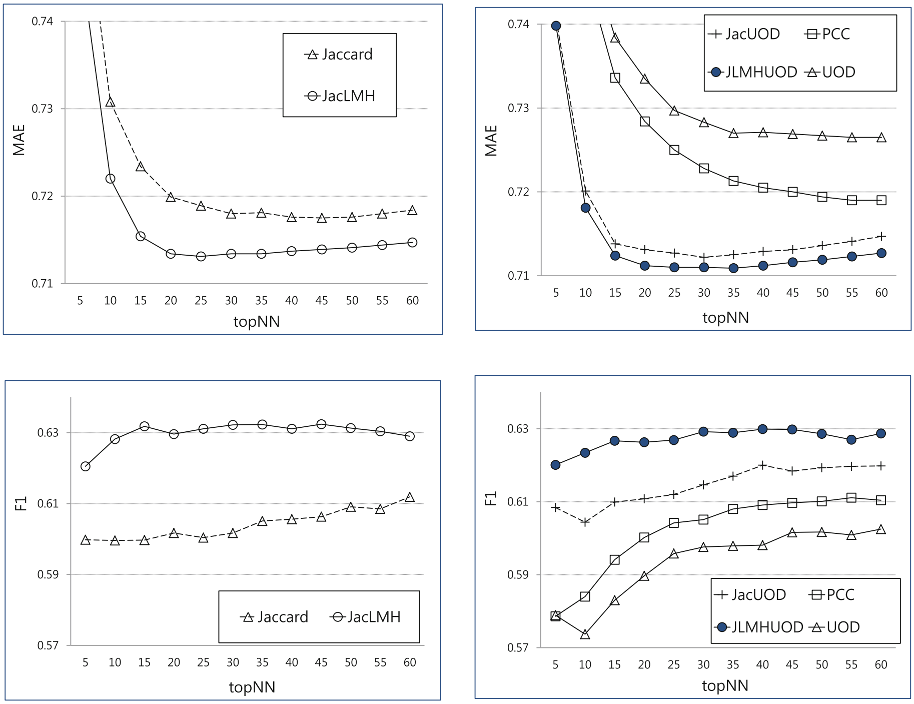


Fig. 2. MAE and F1 results with MovieLens dataset

Note that PCC performs very poor compared to JacUOD and JLMHUOD, although it consults specific ratings of neighbors, which is not the case for the latter two metrics. The reason for this poor performance of PCC comes mostly from sparseness of the dataset, as many researchers discussed the resulting drawbacks of traditional measures [3,4]. JLMHUOD, somewhat against our expectation, outperforms JacUOD only slightly, compared to the performance difference between Jaccard and JacLMH shown in the left figure. Nevertheless, it is notable that JLMHUOD performs best among all the metrics experimented, even though it divides the rating range and so may be disadvantageous with a sparse dataset such as MovieLens. F1 results of the proposed metric are vastly superior to the others, better achievements than MAE results. One thing to note is that JacLMH is slightly better than JLMHUOD, although it does not reflect any ratings but only the number of ratings.

Performance of metrics with Jester dataset is presented in Fig. 3. As seen, Jaccard and JacLMH differ greatly in MAE performance, where its difference is much bigger than with MovieLens. This is because Jester dataset is much denser, thus providing much more meaningful Jaccard indexes in sub-intervals of JacLMH. JacLMH improves about 4.35 to 7.1% of Jaccard results with Jester.

MAE of JLMHUOD is also the lowest and even better than that of PCC overall, especially with lower topNNs. This result is surprising, since one of most

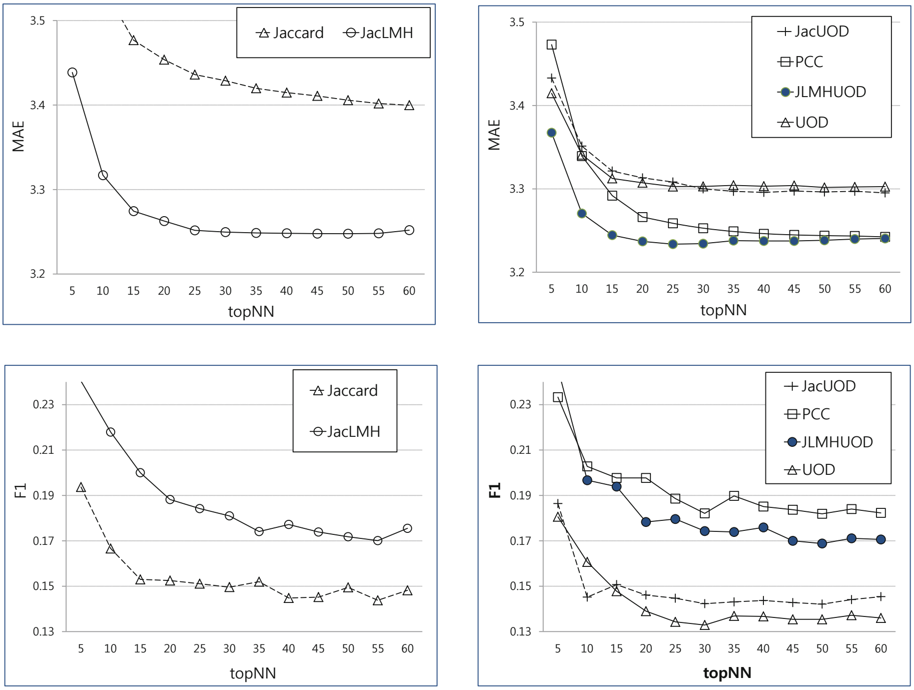


Fig. 3. MAE and F1 results with Jester dataset

popularly used similarity measures such as PCC is thought to perform better. The other two metrics, JacUOD and UOD are very competitive throughout topNNs. This means that considering the number of common ratings as in Jaccard index is no longer effective with sufficient number of ratings data provided. Comparison of F1 performance between metrics is analogous to MAE, as observed in the figure. Roughly, the metrics are grouped into two, in terms of performance. Note that different from MAE results, PCC, followed by JLMHUOD, yields the best F1 results. In conclusion, the proposed metric and its combination with UOD are proved to yield the best overall prediction and recommendation qualities regardless of the rating data density.

5 Conclusion

This study proposed a novel improvement of Jaccard index. The proposed idea takes the frequency of rating values assigned by users as well as the number of common items into consideration. We investigated the performance of the proposed index when used for collaborative filtering and found that it outperformed Jaccard index especially on a dense dataset. Furthermore, the combination of the proposed index with a previous measure is used as a similarity measure for collaborative filtering, whose experimentation results are found superior to those of previous measures, regardless of the data sparsity of datasets. One possiblve limitation of the proposed index is determination of the boundaries of sub-intervals, on which extensive experiments are conducted with various boundaries and their performance results are provided in the text.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence 2009 (2009)
3. Ahn, H.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf. Sci. **178**(1), 37–51 (2008)
4. Saranya, K.G., Sadasivam, G.S., Chandralekha, M.: Performance comparison of different similarity measures for collaborative filtering technique. Indian J. Sci. Technol. **9**(29), 1–8 (2016)
5. Bobadilla, J., Serradilla, F., Bernal, J.: A new collaborative filtering metric that improves the behavior of recommender systems. Knowl. Based Syst. **23**(6), 520–528 (2010)
6. Bobadilla, J., Ortega, F., Hernando, A., Bernal, J.: A collaborative filtering approach to mitigate the new user cold start problem. Knowl. Based Syst. **26**, 225–238 (2012)
7. Koutrica, G., Bercovitz, B., Garcia-Molina, H.: FlexRecs: expressing and combining flexible recommendations. In: The 2009 ACM SIGMOD International Conference on Management of Data, pp. 745–758. ACM (2009)

8. 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)
9. Sun, H.-F., et al.: JacUOD: a new similarity measurement for collaborative filtering. J. Comput. Sci. Technol. **27**(6), 1252–1260 (2012)
10. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of K-fold cross-validation. J. Mach. Learn. Res. **5**, 1089–1105 (2004)
11. Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A.: Recommender systems survey. Knowl. Based Syst. **46**, 109–132 (2013)