PLOS ONE

RESEARCH ARTICLE

Integrating Triangle and Jaccard similarities for recommendation

Shuang-Bo Sun[1], Zhi-Heng Zhang[2], Xin-Ling Dong[1], Heng-Ru Zhang[1], Tong-Jun Li[3], Lin Zhang[1], Fan Min[1]*

1 School of Computer Science, Southwest Petroleum University, Chengdu 610500, China, **2** School of Science, Southwest Petroleum University, Chengdu 610500, China, **3** School of Mathematics, Physics and Information Science, Zhejiang Ocean University, Zhoushan 316022, China

* minfanphd@163.com

Citation: Sun S-B, Zhang Z-H, Dong X-L, Zhang H-R, Li T-J, Zhang L, et al. (2017) Integrating Triangle and Jaccard similarities for recommendation. PLoS ONE 12(8): e0183570. https://doi.org/10.1371/journal.pone.0183570

Editor: Quan Zou, Tianjin University, CHINA

Received: June 5, 2017

Accepted: August 7, 2017

Published: August 17, 2017

Data Availability Statement: All code files and datasets are available from the Github database (https://github.com/FanSmale/TMJSimilarity.git).

Funding: This work was supported by National Natural Science Foundation of China (Grant 61379089 and 41604114), http://www.nsfc.gov.cn/, decision to publish and preparation of the manuscript; Natural Science Foundation of the Department of Education of Sichuan Province (Grant 16ZA0060), http://www.scsjyt.gov.cn/, study design; Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province (grant No. OBDMA201601), http://obdm.zjou.edu.

Abstract

This paper proposes a new measure for recommendation through integrating Triangle and Jaccard similarities. The Triangle similarity considers both the length and the angle of rating vectors between them, while the Jaccard similarity considers non co-rating users. We compare the new similarity measure with eight state-of-the-art ones on four popular datasets under the leave-one-out scenario. Results show that the new measure outperforms all the counterparts in terms of the mean absolute error and the root mean square error.

Introduction

The distance measure is essential in machine learning tasks such as clustering [1, 2], classification [3, 4], image processing [5], and collaborative filtering [6–9]. Collaborative filtering (CF) through k-nearest neighbors (kNN) is a popular memory-based recommendation [10–12] schema. The key issue of CF scheme is how to calculate the similarity between users [6, 13] or items [14, 15]. Various types of similarity measures [16, 17] have been adopted or designed for this issue. State-of-the-art ones include Cosine [18], Pearson Correlation Coefficient (PCC) [6, 19], Jaccard [20], Proximity Impact Popularity (PIP) [21], New Heuristic Similarity Model (NHSM) [22] and so on. Naturally, new similarity measures providing better prediction ability are always desired.

This paper proposes the Triangle multiplying Jaccard (TMJ) similarity. Only the item-based CF [14, 15, 23] will be considered since it performs better than the user-based [13, 24] one. As illustrated in Fig 1, the rating vectors of two items form a triangle in the space. The Triangle similarity is one minus the third divided by the sum of two edges corresponding to the vectors. Since it only considers the co-rating users, it is not good enough when used alone. Fortunately, the Jaccard similarity complements with it in that non co-rating users are considered. Therefore TMJ can take advantage of both Triangle and Jaccard similarities.

We compare TMJ with eight existing measures on four popular datasets under the leave-one-out scenario. These datasets include Movielens 100k, 1M, FilmTrust and EachMovie. The leave-one-out scenario is chosen because the result is not influenced by the division of the training/testing sets. Results show that the recommender system using TMJ outperforms all

cn/, data collection and analysis; and Innovation and Entrepreneurship Foundation of Southwest Petroleum University (Grant SWPUSC16-003), http://www.swpu.edu.cn/, data collection and analysis.

Competing interests: The authors have declared that no competing interests exist.

the counterparts in terms of the mean absolute error (MAE) and the root mean square error (RSME). Specifically, the MAE obtained on four datasets are 0.707, 0.671, 0.614 and 0.179, respectively.

In subsequent sections, we firstly review the basic concept of memory-based recommender system and eight popular similarity measures. Secondly we present the Triangle and TMJ similarities with a running example. Complexity analysis is also presented. Subsequently, we analyze the experimental results. Finally, we make our concluding remarks and indicate further

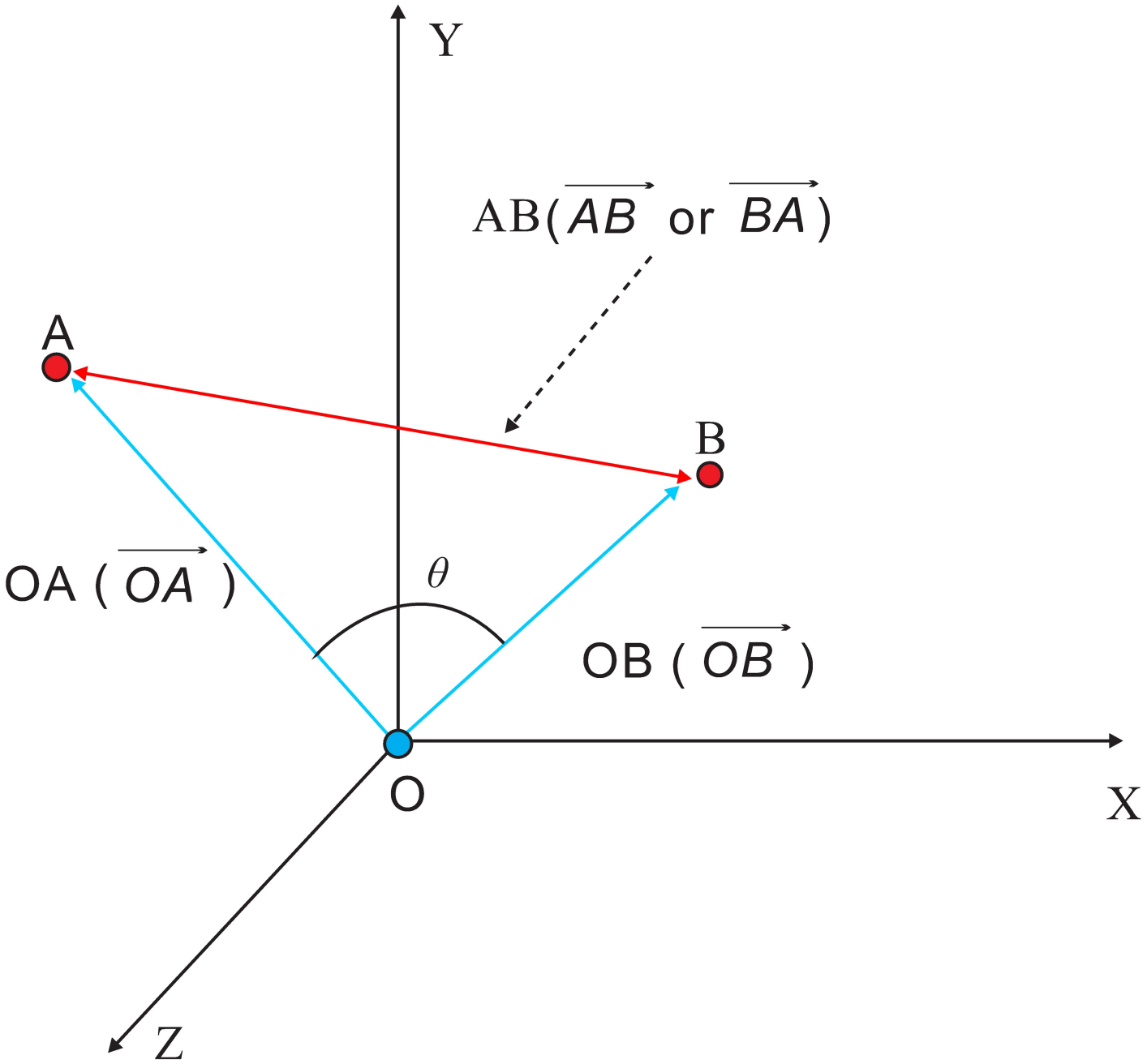

Fig 1. The Triangle in 3D space.

https://doi.org/10.1371/journal.pone.0183570.g001

work. All code files and data sets are available from the Github database (https://github.com/FanSmale/TMJSimilarity.git).

Related work

In this section, we review eight similarity measures including the Cosine [18], PCC [6, 19], Constrained Pearson Correlation Coefficient (CPCC) [13], Jaccard [20], Bhattacharyya Coefficient (BC) [25, 26], Euclidean similarity (ES) [27, 28], PIP [21] and NHSM [22].

Rating system

The user-item relationship is often expressed by a rating system. Let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users of a recommender system and $I = \{i_1, i_2, \ldots, i_n\}$ be the set of all possible items that can be recommended to users. Then the rating function is often defined as [29]

$$r : U \times I \rightarrow R, \tag{1}$$

where R is the rating domain used by the users to evaluate items.

For convenience, we let $r_{u,i}$ be the rating of item $i \in I$ evaluated by user $u \in U$, $r_i = (r_{u_1,i}, r_{u_2,i} \cdots, r_{u_m,i})$ be the rating vector of item i, and $\forall 1 \leq j \neq q \leq n$, C_{i_j,i_q} be the set of co-rating users who have rated i_j and i_q. Here we have the following example.

Example 1 Table 1 *lists an example of rating system.* $R = \{1, 2, 3, 4, 5\}$, *where the numbers 1 through 5 represent the five rating levels; 0 indicates that the user has not rated the item. Given* u_4 *and* i_2, $r_{u_4,i_2} = 1$ *means that the rating of* u_4 *to* i_2 *is 1.* $r_{i_1} = \{4, 5, 4, 2, 4\}$ *is the rating vector of item* i_1; $C_{i_1,i_3} = \{u_1, u_3, u_5\}$ *is the set of co-rating users who have rated* i_1 *and* i_3.

The leave-one-out scenario

Leave-one-out cross validation is a general training/testing scenario for evaluating the performance of a recommender system as well as a classifier. Each time only one rating is used as the test set, and the remaining ratings are used as the training set. Different from split-in-two or 10-fold cross validation, the result is not influenced by the division of the training/testing sets.

An example of the leave-one-out scenario is listed as follows.

Example 2 *Based on* Table 1, *we first leave* r_{u_1,i_1} *out and replace it with "?". The purpose is to predict the value of "?". After we obtain the prediction value called* p_{u_1,i_1}, *the error of prediction is hence computed by* $|r_{u_1,i_1} - p_{u_1,i_1}|$. *Then, we restore the value of* r_{u_1,i_1} *and leave the next rating out. This process terminates until all ratings are left out and predicted.*

Table 1. Rating system.

UID/IID	i_1	i_2	i_3	i_4	i_5
u_1	4	3	5	4	2
u_2	5	3	0	0	4
u_3	4	3	3	2	1
u_4	2	1	0	1	2
u_5	4	2	3	0	2

https://doi.org/10.1371/journal.pone.0183570.t001

MAE and RSME

Given a rating system, the MAE [30] of the predictors is computed by

$$\mathrm{MAE}(r,p)=\frac{\sum_{u\in U,i\in I,r_{u,i}\neq 0}|r_{u,i}-p_{u,i}|}{|\{\langle u,i\rangle|u\in U,i\in I,r_{u,i}\neq 0\}|}, \tag{2}$$

where $p_{u,i}$ is the prediction rating of user u for item i, and the RSME [30] is computed by

$$\mathrm{RSME}(r,p)=\sqrt{\frac{\sum_{u\in U,i\in I,r_{u,i}\neq 0}|r_{u,i}-p_{u,i}|^2}{|\{\langle u,i\rangle|u\in U,i\in I,r_{u,i}\neq 0\}|}}. \tag{3}$$

They are widely used to evaluate the performance of recommender systems. Naturally, the lower the value of MAE and RSME, the better the performance of the recommender system.

Popular similarities

Various popular similarities are employed in recommender systems.

PIP. PIP, consisting of three factors (i.e., Proximity, Impact, and Popularity), is defined as [21]

$$\mathrm{PIP}(i_j,i_q)=\sum_{u\in C_{i_j,i_q}}\mathrm{Pro}(r_{u,j},r_{u,q})\times\mathrm{Imp}(r_{u,j},r_{u,q}))\times\mathrm{Pop}(r_{u,j},r_{u,q}), \tag{4}$$

where the detail calculation can be found in [21].

NHSM. NHSM, consisting of two factors (i.e., JPSS and URP), is defined as [22]

$$\mathrm{NHSM}(i_j,i_q)=\mathrm{JPSS}(i_j,i_q)\times\mathrm{URP}(i_j,i_q), \tag{5}$$

where the detail calculation can be found in [22].

Cosine. Cosine which focuses on the angle between two vectors of items is defined as [18]

$$\mathrm{Cosine}(i_j,i_q)=\frac{\vec{r_j}\cdot\vec{r_q}}{|\vec{r_j}|\times|\vec{r_q}|}, \tag{6}$$

where $\vec{r_j}=(r_{u_1,j},r_{u_2,j},\cdots,r_{u_m,j})^{\mathrm{T}}$ is the rating vector of item i_j.

PCC. PCC which considers the linear correlation between two ratings vectors is defined as [6, 19]

$$\mathrm{PCC}(i_j,i_q)=\frac{\sum_{u\in C_{i_j,i_q}}(r_{u,j}-\bar{r}_j)(r_{u,q}-\bar{r}_q)}{\sqrt{\sum_{u\in C_{i_j,i_q}}(r_{u,j}-\bar{r}_j)^2}\times\sqrt{\sum_{u\in C_{i_j,i_q}}(r_{u,q}-\bar{r}_q)^2}}. \tag{7}$$

CPCC. CPCC based on PCC, which considers the impact of positive and negative ratings, is defined as [13]

$$\mathrm{CPCC}(i_j,i_q)=\frac{\sum_{u\in C_{i_j,i_q}}(r_{u,j}-r_{med})(r_{u,q}-r_{med}))}{\sqrt{\sum_{u\in C_{i_j,i_q}}(r_{u,j}-r_{med})^2}\times\sqrt{\sum_{u\in C_{i_j,i_q}}(r_{u,q}-r_{med})^2}}, \tag{8}$$

where r_{med} is the median of R. If the $R=\{1,2,3,4,5\}$, we have $r_{med}=3$.

Jaccard. Jaccard is defined as the size of the intersection divided by the size of the union of the rating users [20]

$$\text{Jaccard}(i_j, i_q) = \frac{|I_j \cap I_q|}{|I_j \cup I_q|}, \tag{9}$$

where $I_j = \{u \in U | r_{u,j} > 0\}$ and $I_q = \{u \in U | r_{u,q} > 0\}$.

BC. BC, which measures similarity by means of two probability distributions, is defined as [25, 26]

$$\text{BC}(i_j, i_q) = \sum_{x \in R} \sqrt{P_{j,x} \times P_{q,x}}, \tag{10}$$

where $P_{j,x}$ is the probability distribution of the rating x in item j.

ES. Euclidean distance (ED) which is the real distance between two points in Euclidean space is defined as [27, 28]

$$\text{ED}(i_j, i_q) = \sqrt{\sum_{u \in C_{i_j, i_q}} (r_{u,j} - r_{u,q})^2}. \tag{11}$$

In Fig 1, $|AB|$ is ED(A, B).

Therefore, ES can be computed by

$$\text{ES}(i_j, i_q) = 1 - \frac{\text{ED}(i_j, i_q)}{\text{ED}_{max}}, \tag{12}$$

where ED_{max} is defined as

$$\text{ED}_{max} = (R_{max} - R_{min}) \sqrt{|C_{i_j, i_q}|}, \tag{13}$$

where R_{max} is the maximum value (e.g., 5) of rating set R, and R_{min} is the minimum one (e.g., 1).

kNN-based CF approach

The type of CF schema includes memory-based and model-based [31, 32] methods. The kNN [33, 34] algorithm is one of the most fundamental CF recommendation techniques. Here we adopt the kNN-based CF approach to predict the ratings. One key to kNN algorithms is the definition of the similarity measures. Popular measures have been presented. The prediction value of $r_{u,j}$ is computed as follows.

$$p(u, j) = \bar{r}_{i_j} + \frac{\sum_{q \in h} Sim(i_j, i_q)(r_{u,i_q} - \bar{r}_{i_q})}{\sum_{q \in h} |Sim(i_j, i_q)|}, \tag{14}$$

where h is set of neighbors, and $Sim(i_j, i_q)$ is similarity of items i_j and i_q.

Integrating Triangle and Jaccard similarities

In this section, we first propose the definition of Triangle similarity. Then we define the TMJ, and presented complexity analysis. Finally, we present a running example of TMJ.

Triangle

The Triangle similarity is defined by

$$\text{Triangle}(i_j, i_q) = 1 - \frac{\sqrt{\sum_{u \in C_{i_j,i_q}} (r_{u,j} - r_{u,q})^2}}{\sqrt{\sum_{u \in C_{i_j,i_q}} r_{u,j}^2} + \sqrt{\sum_{u \in C_{i_j,i_q}} r_{u,q}^2}}, \tag{15}$$

whose value range is [0, 1], where 0 indicates $C_{i_j,i_q} = \emptyset$. The bigger value of Triangle, the more similar they are.

With the perspective of geometry, Eq (15) also can be defined as follows.

$$\text{Triangle}(i_j, i_q) \quad = \text{Triangle}(\overrightarrow{OA}, \overrightarrow{OB}) = 1 - \frac{|AB|}{|OA| + |OB|}, \tag{16}$$

where $\overrightarrow{OA}$ is the rating vector of i_j, $\overrightarrow{OB}$ is the rating vector of i_q.

Triangle considers both the length of vectors and the angle between them, so it is more reasonable than the angle based Cosine measure. For example, given the two vectors A = (5, 5, 5) and B = (1, 1, 1), the Cosine similarity is 1, which is contrary to common sense. In contrast, the Triangle similarity between them is 0.33, more in line with expectations.

TMJ

However, Triangle only considers the co-rating users. To provide more information about non co-rating users, we further combine Jaccard measure to improve Triangle, hence obtain a new hybrid measure as follows.

$$\text{TMJ}(i_j, i_q) = \text{Triangle}(i_j, i_q) \times \text{Jaccard}(i_j, i_q), \tag{17}$$

which is the multiplication of Triangle and Jaccard similarity.

Complexity analysis

Let the number of users and items be m and n, respectively. According to Eqs (9), (15) and (17), the time complexity of item similarity computation of Jaccard, Triangle, and TMJ is $O(m)$.

kNN is employed to find the nearest k neighbors for each item. Therefore, for one item, the time complexity of finding all neighbors is $O(mn)$.

In the leave-one-out cross validation scenario, all ratings should be predicted and validated. Since the maximal number of ratings is mn, the time complexity of testing the whole dataset is $O(m^2 n^2)$.

A running example

Given a rating system by Table 1. First, the co-rating users is obtained as $C_{i_1,i_3} = I_1 \cap I_3 = \{u_1, u_3, u_5\}$. Second, the Triangle similarity between i_1 and i_3 is computed by

$$\begin{aligned} \text{Triangle}(i_1, i_3) \quad &= 1 - \frac{\sqrt{(4-5)^2 + (4-3)^2 + (4-3)^2}}{\sqrt{4^2 + 4^2 + 4^2} + \sqrt{5^2 + 3^2 + 3^2}} \\ &\approx 0.872. \end{aligned}$$

Table 2. Summaries of datasets.

Dataset	$\|U\|$	$\|I\|$	Ratings	Scale
MovieLens 100K	943	1,682	{1, 2, 3, 4, 5}	10^5
MovieLens 1M	6,040	3,952	{1, 2, 3, 4, 5}	10^6
FilmTrust	1,508	2,071	{0.5, 1, 1.5, . . ., 4}	10^5
EachMovie	72,916	1,628	{0, 0.2, 0.4, 0.6, 0.8, 1}	10^6

https://doi.org/10.1371/journal.pone.0183570.t002

Table 3. The MAE comparison.

Measure/Dataset	MovieLens 100K	MovieLens 1M	FilmTrust	EachMovie
ES	0.764	0.808	0.852	0.234
BC	0.735	0.704	0.643	0.191
PCC	0.735	0.695	0.656	0.185
CPCC	0.731	0.694	0.657	0.186
Cosine	0.732	0.696	0.625	0.187
PIP	0.729	0.704	0.625	0.185
NHSM	0.718	–	0.617	–
Jaccard	0.711	0.674	0.617	0.180
Triangle	0.724	0.688	0.621	0.183
TMJ	**0.707**	**0.671**	**0.614**	**0.179**

https://doi.org/10.1371/journal.pone.0183570.t003

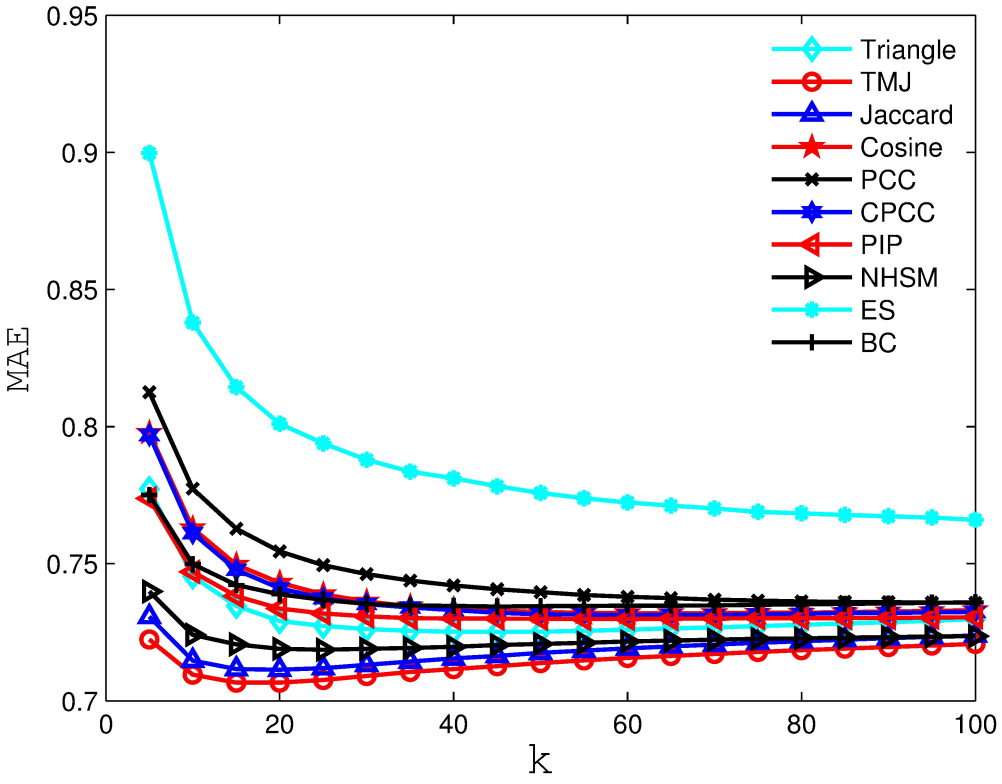

Fig 2. The MAE obtained by the recommender system using different similarity measures on MovieLens 100K.

https://doi.org/10.1371/journal.pone.0183570.g002

The Jaccard similarity between i_1 and i_3 is computed by

$$\text{Jaccard}(i_1, i_3) = \frac{I_1 \cap I_3}{I_1 \cup I_3} = \frac{3}{5} = 0.6.$$

Finally, the TMJ similarity between i_1 and i_3 is computed by

$$\text{TMJ}(i_1, i_3) = \text{Triangle}(i_1, i_3) \times \text{Jaccard}(i_1, i_3) = 0.872 \times 0.6 = 0.523.$$

Experiments

In this section, quality measures like the MAE, the RSME are applied to evaluate the above 10 similarity measures. Experiments are undertaken on four real world datasets such as MovieLens 100K, MovieLens 1M, FilmTrust and EachMovie.

Datasets

In the experiments we used four real world datasets such as MovieLens 100K, MovieLens 1M, FilmTrust and Each Movie. The dataset schema is as follows.

- User (userID, age, gender, occupation)
- Movie (movieID, release-year, genre)
- Rating (userID, movieID)

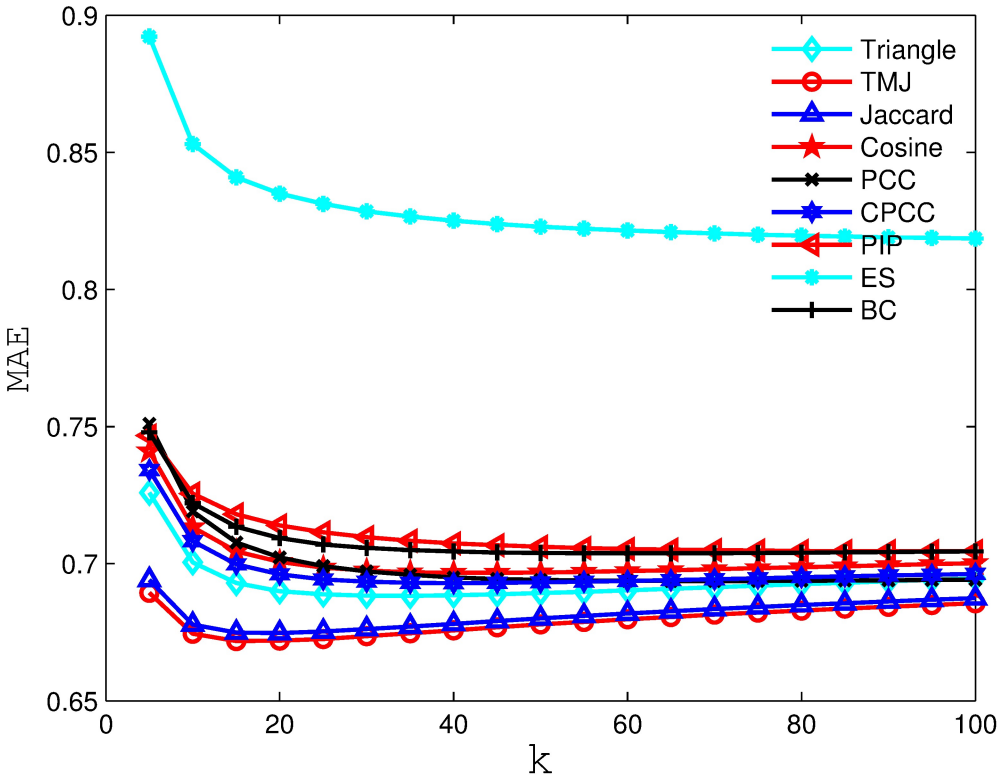

Fig 3. The MAE obtained by the recommender system using different similarity measures on MovieLens 1M.

https://doi.org/10.1371/journal.pone.0183570.g003

We used the MovieLens 100K (943 users× 1,682 movies), MovieLens 1M (6,040 users × 3,952 movies), FilmTrust (1,508 users × 2,071 movies), and EachMovie (72,916 users × 1,628 movies). The detail of these datasets are shown in Table 2. However, 0 is a rating level in EachMovie dataset.

Comparison of the MAE

Table 3 compares the MAE obtained by recommender systems using 10 similarity measures. Symbol "–" indicates that the algorithm cannot be completed within an acceptable period of time when the measure is used. The recommender system using the TMJ measure achieves the best/minimal MAE. In these four datasets, it is lower by 0.4%- 5.7%, 0.3%- 13.7%, 0.3%- 23.8%, and 0.1%- 5.5%, respectively, than the values obtained by other methods. The MAE of Triangle is also acceptable. It ranked fourth in the first dataset and third in the other three.

Figs 2, 3, 4 and 5 compare the MAE obtained by the recommender system using different similarity measures and setting different k values (i.e., number of the nearest neighbors). As we can see from the figure, the recommender system always obtains the best MAE when using TMJ, regardless of the k value. However, it obtains the best MAE, when k on the four datasets are 15, 15, 10, and 15, respectively.

Comparison of the RMSE

Table 4 compares the RSME obtained by recommender systems using 10 similarity measures. Symbol "–" indicates that the algorithm cannot be completed within an acceptable period of

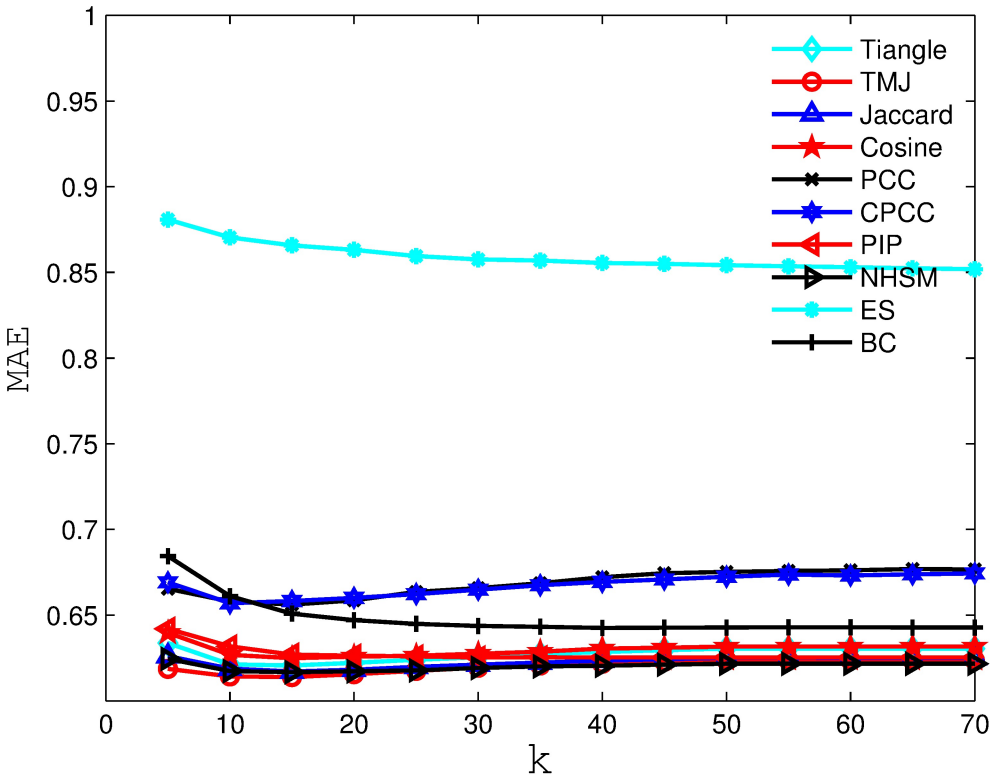

Fig 4. The MAE obtained by the recommender system using different similarity measures on FilmTrust.

https://doi.org/10.1371/journal.pone.0183570.g004

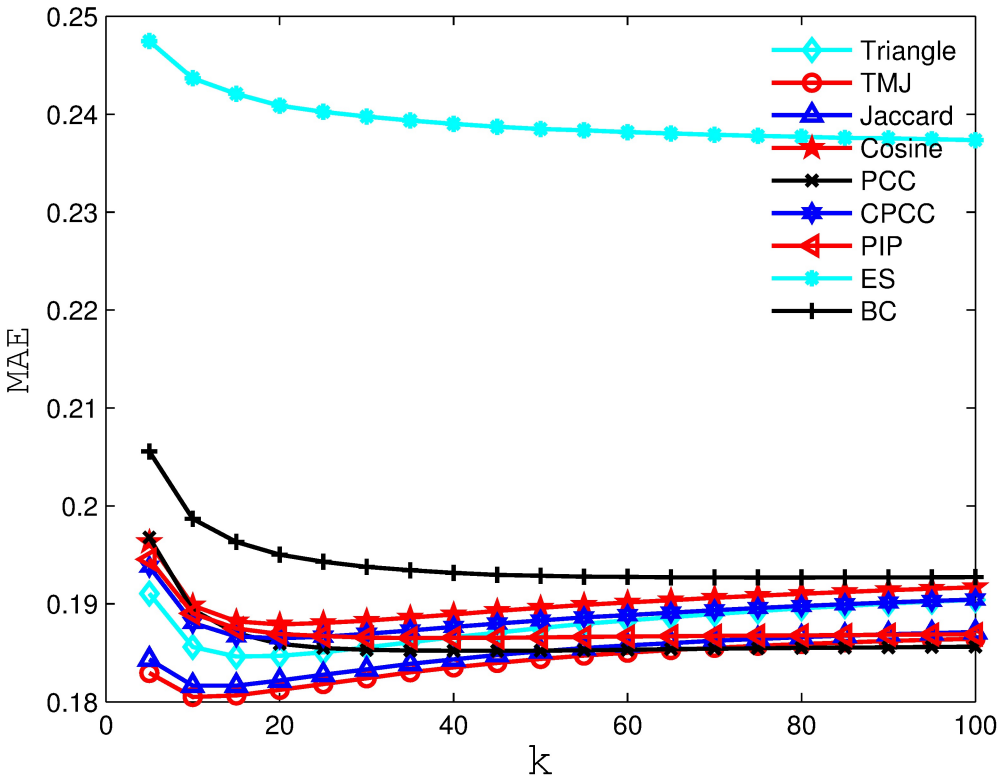

Fig 5. The MAE obtained by the recommender system using different similarity measures on EachMoive.

https://doi.org/10.1371/journal.pone.0183570.g005

time when the measure is used. The recommender system using the TMJ measure achieves the best/minimal RSME. In these four datasets, it is lower by 0.5%- 6.6%, 0.3%- 18%, 0.1%- 22.7%, and 0.1%- 6.1%, respectively, than the values obtained by other methods. The RSME of Triangle is also acceptable. It ranked fourth in the first dataset and third in the other three.

Figs 6, 7, 8 and 9 compare the RSME obtained by the recommender system using different similarity measures and setting different k values (i.e., number of nearest neighbors). As we can see from the figure, the recommender system always obtains the best RSME when using

Table 4. RSME comparison.

Measure/Dataset	MovieLens 100K	MovieLens 1M	FilmTrust	EachMovie
ES	0.969	1.039	1.043	0.296
BC	0.934	0.895	0.838	0.248
PCC	0.937	0.889	0.903	0.240
CPCC	0.932	0.885	0.900	0.243
Cosine	0.931	0.886	0.829	0.243
PIP	0.926	0.893	0.823	0.240
NHSM	0.915	–	0.817	–
Jaccard	0.908	0.862	0.822	0.236
Triangle	0.923	0.877	0.821	0.240
TMJ	**0.903**	**0.859**	**0.816**	**0.235**

https://doi.org/10.1371/journal.pone.0183570.t004

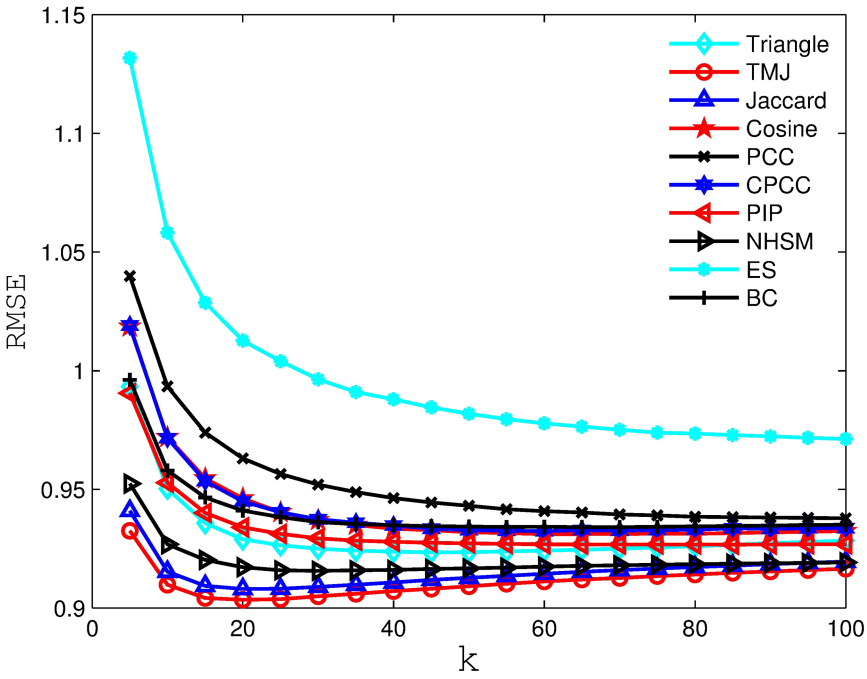

Fig 6. The RSME obtained by the recommender system using different similarity measures on MovieLens 100K.

https://doi.org/10.1371/journal.pone.0183570.g006

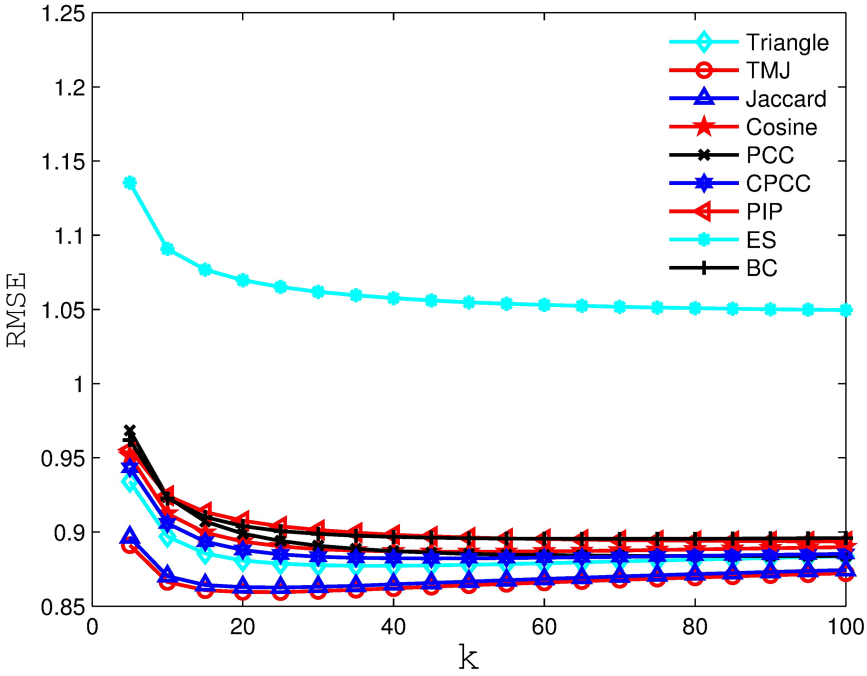

Fig 7. The RSME obtained by the recommender system using different similarity measures on MovieLens 1M.

https://doi.org/10.1371/journal.pone.0183570.g007

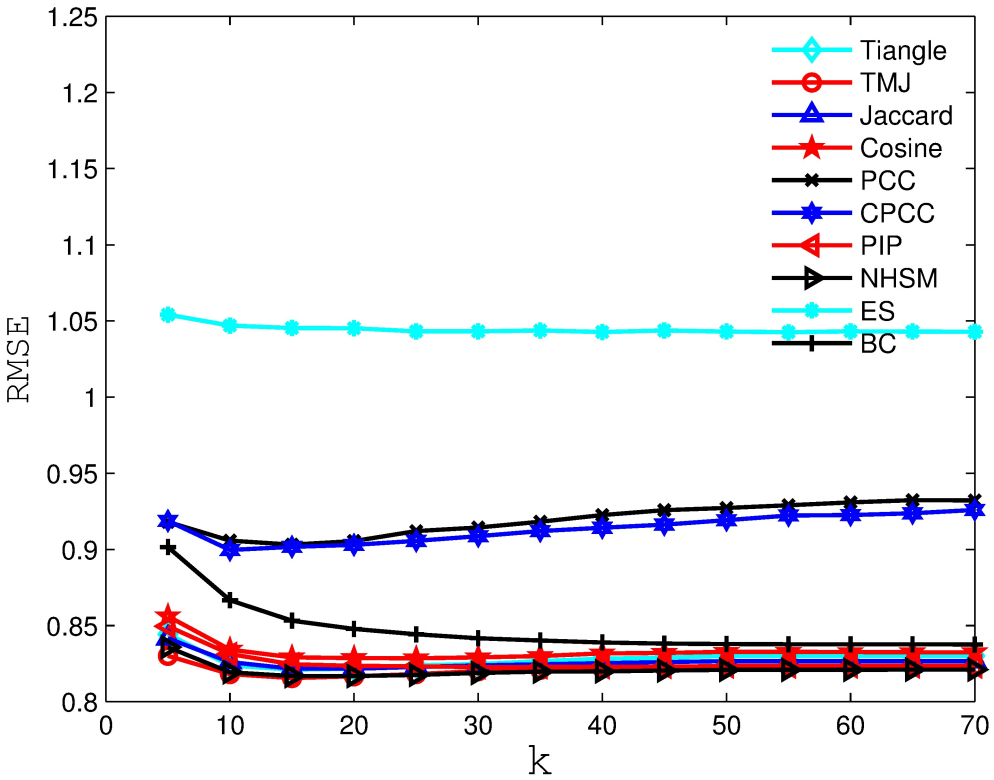

Fig 8. The RSME obtained by the recommender system using different similarity measures on FilmTrust.

https://doi.org/10.1371/journal.pone.0183570.g008

TMJ, regardless of the *k* value. However, it obtains the best RSME, when *k* on the four datasets are 15, 15, 10, and 15, respectively.

Discussion

From the viewpoint of multiple kernel learning, the similarity measures such as Jaccard and Triangle meet the requirements of kernel function. TMJ is a product of Jaccard and Triangle. According to the property proved in [35] (pages 75–76), TMJ is also a kernel function.

There are various types of recommendation algorithms, such as kNN, NMF, LMF, etc. NMF algorithms address the recommendation task as the matrix completion problem with high sparsity. They intrinsically work in batch mode to predict all missing values. Since they do not need any similarity measure, we cannot incorporate our new measure into them. In fact, our new measure only serves as the basis of some similarity-based prediction models such as kNN. It can replace the existing measures anywhere, such as Manhattan, cosine, etc. In this sense it is general enough. However, support for batch mode is provided by the prediction model, rather than through the similarity measure. Hence we do not discuss this issue in more detail. To the best of our knowledge, kNN-based approaches usually predict rating one-by-one even for the split-in-two scenario.

Conclusions

This paper defined the TMJ measure by integrating Triangle and Jaccard similarities. The new measure outperforms all the counterparts in terms of the MAE and the RMSE. In the

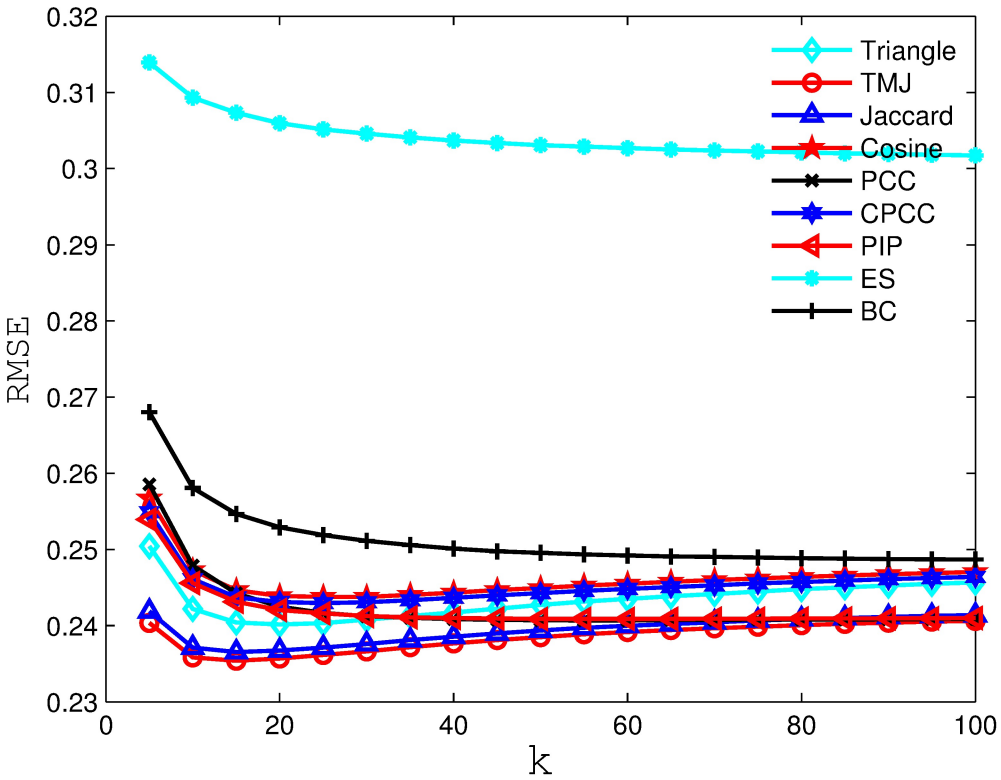

Fig 9. The RSME obtained by the recommender system using different similarity measures on EachMoive.

https://doi.org/10.1371/journal.pone.0183570.g009

future, we will apply the new measure to other tasks, such as the three-way recommendation [7, 36–42], clustering [2, 43], and image processing [5, 44, 45]. We will also develop other similarity measures in the light of multi-kernel learning [44, 46].

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61379089, 41604114 and the Natural Science Foundation of the Department of Education of Sichuan Province under Grant 16ZA0060 and Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province (grant No.OBDMA201601) and the Innovation and Entrepreneurship Foundation of Southwest Petroleum University (Grant SWPUSC16-003).

Author Contributions

Conceptualization: Shuang-Bo Sun, Xin-Ling Dong.

Data curation: Shuang-Bo Sun, Zhi-Heng Zhang, Fan Min.

Formal analysis: Shuang-Bo Sun, Zhi-Heng Zhang, Heng-Ru Zhang, Fan Min.

Funding acquisition: Tong-Jun Li, Fan Min.

Investigation: Shuang-Bo Sun, Zhi-Heng Zhang, Xin-Ling Dong, Fan Min.

Methodology: Shuang-Bo Sun, Fan Min.

Project administration: Tong-Jun Li, Lin Zhang, Fan Min.

Resources: Zhi-Heng Zhang, Heng-Ru Zhang.

Supervision: Xin-Ling Dong, Heng-Ru Zhang, Lin Zhang, Fan Min.

Validation: Shuang-Bo Sun, Zhi-Heng Zhang, Heng-Ru Zhang, Fan Min.

Writing – original draft: Shuang-Bo Sun, Zhi-Heng Zhang.

Writing – review & editing: Shuang-Bo Sun, Fan Min.

References

1. Xing EP, Ng AY, Jordan MI, Russell S. Distance Metric Learning, With Application To Clustering With Side-Information. Advances in Neural Information Processing Systems. 2002; 15:505–512.
2. Wang M, Min F, Zhang ZH, Wu YX. Active learning through density clustering. Expert Systems with Applications. 2017; 85. https://doi.org/10.1016/j.eswa.2017.05.046
3. Ling H, Jacobs DW. Shape classification using the inner-distance. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007; 29(2):286–299. https://doi.org/10.1109/TPAMI.2007.41 PMID: 17170481
4. Weinberger KQ, Saul LK. Distance Metric Learning for Large Margin Nearest Neighbor Classification. Journal of Machine Learning Research. 2009; 10(1):207–244.
5. Li C, Xu C, Gui C, Fox MD. Distance Regularized Level Set Evolution and Its Application to Image Segmentation. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society. 2010; 19(12):3243–3254. https://doi.org/10.1109/TIP.2010.2069690 PMID: 20801742
6. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM; 1994. p. 175–186.
7. Zhang HR, Min F, Shi B. Regression-based three-way recommendation. Information Sciences. 2017; 378:444–461. https://doi.org/10.1016/j.ins.2016.03.019
8. Yu H, Li JH. Algorithm to solve the cold-start problem in new item recommendations. Journal of Software. 2015; 26(6):1395–1408.
9. Katarya R, Verma OP. Effectual recommendations using artificial algae algorithm and fuzzy c-mean. Swarm and Evolutionary Computation. 2017;. https://doi.org/10.1016/j.swevo.2017.04.004
10. Jeong B, Lee J, Cho H. Improving memory-based collaborative filtering via similarity updating and prediction modulation. Information Sciences. 2010; 180(5):602–612. https://doi.org/10.1016/j.ins.2009.10.016
11. Ghazarian S, Nematbakhsh MA. Enhancing memory-based collaborative filtering for group recommender systems. Expert Systems with Applications. 2015; 42(7):3801–3812. https://doi.org/10.1016/j.eswa.2014.11.042
12. Yu H, Zhou B, Deng MY, Hu F. Tag recommendation method in folksonomy based on user tagging status. Journal of Intelligent Information Systems;.
13. Shardanand U. Social information filtering: algorithms for automating word of mouth. In: Sigchi Conference on Human Factors in Computing Systems; 1995. p. 210–217.
14. Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. ACM; 2001. p. 285–295.
15. Barragns-Martnez AB, Costa-Montenegro E, Burguillo JC, Rey-Lpez M, Mikic-Fonte FA, Peleteiro A. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences. 2010; 180(22):4290–4311. https://doi.org/10.1016/j.ins.2010.07.024
16. Su XY, Khoshgoftaar TM. A survey of collaborative filtering techniques. Advances in Artificial Intelligence. 2009; 2009(12):4.
17. Zou Q, Li JJ, Song L, Zeng XX, Wang GH. Similarity computation strategies in the microRNA-disease network: a survey. Knowledge-Based Systems. 2016; 105:190–205.
18. Salton G, Mcgill MJ. Introduction to modern information retrieval. McGraw-Hill; 1983.
19. Pearson K, Stouffer SA, David FN. On the distribution of the correlation coefficient in small samples. Biometrika. 1932; 24(3-4):382–403. https://doi.org/10.1093/biomet/24.3-4.382
20. Jaccard P. Nouvelles researches sur la distribution florale. Bull Soc Vaud Sci Nat. 1908; 44:223–270.

21. Ahn HJ. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences. 2008; 178(1):37–51. https://doi.org/10.1016/j.ins.2007.07.024
22. Liu HF, Hu Z, Mian A, Tian H, Zhu XZ. A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems. 2014; 56:156–166. https://doi.org/10.1016/j.knosys.2013.11.006
23. Deshpande M, Karypis G. Item-based top- N recommendation algorithms. Acm Transactions on Information Systems. 2004; 22(1):143–177. https://doi.org/10.1145/963770.963776
24. Schafer JB, Konstan J, Riedl J. Recommender systems in e-commerce. In: Proceedings of the 1st ACM conference on Electronic commerce; 1999. p. 158–166.
25. Bhattacharyya A. On a measure of divergence between two statistical populations defined by their probability distributions. Bull Calcutta Math Soc. 1942; 35:99–109.
26. Patra BK, Launonen R, Ollikainen V, Nandi S. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowledge-Based Systems. 2015; 82:163–177. https://doi.org/10.1016/j.knosys.2015.03.001
27. Elmore KL, Richman MB. Euclidean Distance as a Similarity Metric for Principal Component Analysis. Monthly Weather Review. 2001; 129(3):540–549. https://doi.org/10.1175/1520-0493(2001)129%3C0540:EDAASM%3E2.0.CO;2
28. Qiang G, Sural S, Gu YL, Pramanik S. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In: Proceedings of Acm Symposium on Applied Computing. ACM; 2004. p. 1232–1237.
29. Zhang HR, Min F. Three-way recommender systems based on random forests. Knowledge-Based Systems. 2016; 91:275–286. https://doi.org/10.1016/j.knosys.2015.06.019
30. Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research. 2005; 30:79–82. https://doi.org/10.3354/cr030079
31. Billsus D, Pazzani MJ. Learning Collaborative Information Filters. In: Machine Learning. In: Proceedings of the Fifteenth International Conference; 1998.
32. Hofmann T. Collaborative filtering via gaussian probabilistic latent semantic analysis. In: SIGIR 2003: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28—August 1, 2003, Toronto, Canada; 2003. p. 259–266.
33. Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition. 2007; 40(7):2038–2048. https://doi.org/10.1016/j.patcog.2006.12.019
34. Song Y, Liang J, Lu J, Zhao X. An efficient instance selection algorithm for k nearest neighbor regression. Neurocomputing. 2017; 251:26–34. https://doi.org/10.1016/j.neucom.2017.04.018
35. Shawe-Taylor J, Cristianini N. Kernel Methods for Pattern Analysis. Cambridge university press; 2004.
36. Huang J, Wang J, Yao YY, Zhong N. Cost-sensitive three-way recommendations by learning pair-wise preferences. International Journal of Approximate Reasoning. 2017; 86:28–40. https://doi.org/10.1016/j.ijar.2017.03.005
37. Li XN, Yi HJ, She YH, Sun BZ. Generalized three-way decision models based on subset evaluation. International Journal of Approximate Reasoning. 2017; 83:142–159. https://doi.org/10.1016/j.ijar.2017.01.005
38. Li HX, Zhang LB, Huang B, Zhou XZ. Sequential three-way decision and granulation for cost-sensitive face recognition. Knowledge-Based Systems. 2016; 91:241–251. https://doi.org/10.1016/j.knosys.2015.07.040
39. Huang CC, Li JH, Mei CL, Wu WZ. Three-way concept learning based on cognitive operators: An information fusion viewpoint. International Journal of Approximate Reasoning. 2017; p. 1–20.
40. Xu W, Guo Y. Generalized multigranulation double-quantitative decision-theoretic rough set. Knowledge-Based Systems. 2016; 105:190–205. https://doi.org/10.1016/j.knosys.2016.05.021
41. Li Y, Zhang ZH, Chen WB, Min F. TDUP: an approach to incremental mining of frequent itemsets with three-way-decision pattern updating. International Journal of Machine Learning and Cybernetics. 2015; p. 1–13.
42. Gao C, Yao YY. Actionable Strategies in Three-way Decisions. Knowledge-Based Systems. 2017;. https://doi.org/10.1016/j.knosys.2017.07.001
43. Xue ZA, Cen F, Wei LP. A weighting fuzzy clustering algorithm based on euclidean distance. In: Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery. vol. 1; 2008. p. 172–175.
44. Molina-Giraldo S, Alvarez-Meza AM, Peluffo-Ordonez DH, Castellanos-Domnguez G. Image Segmentation Based on Multi-Kernel Learning and Feature Relevance Analysis; 2012.

45. Zhao H, Zhu PF, Wang P, Hu QH. Hierarchical feature selection with recursive regularization. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence; 2017.

46. Zhang JF. Chaotic time series prediction based on multi-kernel learning support vector regression. Acta Physica Sinica. 2008; 57(5):2708–2713.