PMD: An Optimal Transportation-Based User Distance for Recommender Systems

Yitong Meng[1](✉), Xinyan Dai[1], Xiao Yan[1], James Cheng[1], Weiwen Liu[1],
Jun Guo[3], Benben Liao[2], and Guangyong Chen[2]

[1] The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{ytmeng,xydai,xyan,jcheng,wwliu}@cse.cuhk.edu.hk
[2] Tencent Quantum Lab, Shenzhen, Guangdong, China
{bliao,gycchen}@tencent.com
[3] Tsinghua University, Shenzhen, Guangdong, China
eeguojun@outlook.com

Abstract. Collaborative filtering predicts a user's preferences by aggregating ratings from similar users and thus the user similarity (or distance) measure is key to good performance. Existing similarity measures either consider only the co-rated items for a pair of users (but co-rated items are rare in real-world sparse datasets), or try to utilize the non-co-rated items via some heuristics. We propose a novel user distance measure, called Preference Mover's Distance (PMD), based on the optimal transportation theory. PMD exploits all ratings made by each user and works even if users do not share co-rated items at all. In addition, PMD is a metric and has favorable properties such as triangle inequality and zero self-distance. Experimental results show that PMD achieves superior recommendation accuracy compared with the state-of-the-art similarity measures, especially on highly sparse datasets.

Keywords: Recommendation · User similarity · Optimal transport

1 Introduction

Collaborative filtering (CF) is one of the most widely used recommendation techniques [14,47]. Given a user, CF recommends items by aggregating the preferences of similar users. Among CF recommendation approaches, methods based on nearest-neighbors (NN) are widely used, thanks to their simplicity, efficiency and ability to produce accurate and personalized recommendations [13,35,44]. Although deep learning (DL) methods [16,19,43] have attracted much attention in the recommendation community over the past few years, a very recent study [12] shows that NN-based CF is still a strong baseline and outperforms many DL methods. For NN-based methods, the user similarity measure plays an important role. It serves as the criterion to select a group of similar users whose ratings form the basis of recommendations, and is used to weigh users so that more similar users have greater impact on recommendations. Besides

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12036, pp. 272–280, 2020.
https://doi.org/10.1007/978-3-030-45442-5_34

CF, user similarity is also important for applications such as link prediction [4], community detection [34] and so on.

Related Work. Traditional similarity measures, such as cosine distance (COS) [9], Pearson's Correlation Coefficient (PCC) [9] and their variants [18, 29,38,39], have been widely used in CF [13,44]. However, such measures only consider *co-rated* items and ignore ratings on other items, and thus may only coarsely capture users' preferences as ratings are sparse and co-rated items are rare for many real-world datasets [35,40,44]. Some other similarity measures, such as Jaccard [22], MSD [39], JMSD [8], URP [27], NHSM [27], PIP [5] and BS [14] do not utilize all the rating information [6]. For example, Jaccard only uses the number of rated items and omits the specific rating values, while URP only uses the mean and the variance of the ratings. Critically, all these measures give zero similarity value when there are no co-rated items, which would harm recommendation performance. Recently, BCF [35] and HUSM [44] were proposed to alleviate the co-rating issue by modeling user similarity as a weighted sum of item similarities, where the weights are obtained using heuristics. As the weights are not derived in a principled manner, they do not satisfy important properties such as triangle inequality and zero self-distance, which are important for a high quality similarity measure.

The Earth Mover's Distance (EMD) is a distance metric on probabilistic space that originates from the optimal transportation theory [25,37]. EMD has been applied to many applications, such as computer vision [7], natural language processing [17,23] and signal processing [41]. EMD has also been applied to CF [48] but is used as a regularizer to force the latent variable to fit a Gaussian prior in auto-encoder training rather than a user similarity measure.

Our Solution. We propose the Preference Mover's Distance (PMD), which considers all ratings made by each user and is able to evaluate user similarity even in the absence of co-rated items. Similar to BCF and HUSM, PMD uses the item similarity as side information and assumes that if two users have similar opinions on similar items, then their tastes are similar. But the key difference is: PMD formulates the distance between a pair of users as an optimal transportation problem [26,36] such that the weights for item similarities can be derived in a principled manner. In fact, PMD can be viewed as a special case of EMD [33, 37,45], which is a metric that satisfies important properties such as triangle inequality and zero self-distance. We also make PMD practical for large datasets by employing the Sinkhorn algorithm [10] to speed up distance computation and using HNSW [30] to further accelerate the search for similar users. Experimental results show that PMD leads to superior recommendation accuracy over the state-of-the-art similarity measures, especially on sparse datasets.

2 Preference Mover's Distance

Problem Definition. Let $\mathcal{U}$ be a set of m users, and $\mathcal{I}$ a set of n items. The user-item interaction matrix is denoted by $\mathbf{R} \in \mathbb{R}^{m \times n}$, where $\mathbf{R}(u, i) \geq 0$ is the rating user u gives to item i. $\mathbf{R}$ is a partially observed matrix and usually highly sparse. For user $u \in \mathcal{U}$, her rated items are denoted by $\mathcal{I}_u \subset \mathcal{I}$. The item similarities are described by matrix $\mathbf{D}$ and $\mathbf{D}(i,j) \geq 0$ denotes the distance between items i and j. Item similarities can be derived from the ratings on items [35,44] or content information [46], such as item tags, comments, etc. In this paper, we assume $\mathbf{D}$ is given. We are interested in computing the distance between any pair (u,v) of users in $\mathcal{U}$ given $\mathbf{R}$ and $\mathbf{D}$. User similarity can be easily derived from the user distance as they are negatively correlated.

PMD. Let $\Sigma_k = \{\mathbf{p} \in [0,1]^k \mid \mathbf{p}^\top \mathbb{1} = 1\}$ denote a $(k-1)$-dimensional simplex and $\mathbb{1}$ is an all-1 column vector. We model a user's preferences as a probabilistic distribution $\mathbf{p}_u \in \Sigma_{|\mathcal{I}_u|}$ on $\mathcal{I}_u$, where $\mathbf{p}_u(i)$ indicates how much user u likes item i. In practice, the ground truth of $\mathbf{p}_u$ cannot be observed and we estimate it by normalizing user u's ratings on $\mathcal{I}_u$, i.e., $\mathbf{p}_u(i) \approx \frac{\mathbf{R}(u,i)}{\sum_{j \in \mathcal{I}_u} \mathbf{R}(u,j)}$ for $i \in \mathcal{I}_u$. We model the distance between users u and v, denoted by $d(\mathbf{p}_u, \mathbf{p}_v)$, as the weighted average of the distances among their rated items, i.e.,

$$\sum_{i \in \mathcal{I}_u} \sum_{j \in \mathcal{I}_v} \mathbf{W}_{u,v}(i,j) \mathbf{D}(i,j), \tag{1}$$

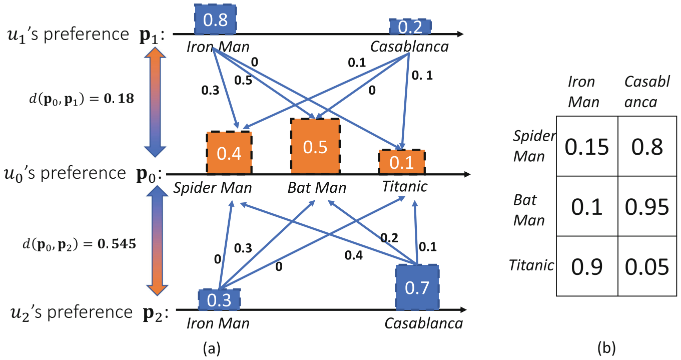
where $\mathbf{W}_{u,v}(i,j) \geq 0$ is the weight for an item pair (i,j) and we introduce the constraint $\sum_{i \in \mathcal{I}_u} \sum_{j \in \mathcal{I}_v} \mathbf{W}_{u,v}(i,j) = 1$ to control the scaling. $\sum_{j \in \mathcal{I}_v} \mathbf{W}_{u,v}(i,j)$ is the aggregate weight received by item i for user u and it should be large if $\mathbf{p}_u(i)$ is large such that $d(\mathbf{p}_u, \mathbf{p}_v)$ can focus on the items that user u likes. Similarly, $\sum_{i \in \mathcal{I}_u} \mathbf{W}_{u,v}(i,j)$ should also be large if $\mathbf{p}_v(j)$ is large. Thus, we constrain the marginal distributions of $\mathbf{W}_{u,v}$ follow $\mathbf{p}_u$ and $\mathbf{p}_v$, i.e., $\mathbf{W}_{u,v} \in U(\mathbf{p}_u, \mathbf{p}_v)$, where

$$U(\mathbf{p}_u, \mathbf{p}_v) := \left\{ \mathbf{W}_{u,v} \in [0,1]^{|\mathcal{I}_u| \times |\mathcal{I}_v|} \mid \mathbf{W}_{u,v} \mathbb{1} = \mathbf{p}_u, \mathbf{W}_{u,v}^T \mathbb{1} = \mathbf{p}_v \right\}. \tag{2}$$

However, $U(\mathbf{p}_u, \mathbf{p}_v)$ contains many different configurations of $\mathbf{W}_{u,v}$, which means that the user distance is indeterminate. Therefore, we define the user distance as the smallest among all possibilities:

$$d(\mathbf{p}_u, \mathbf{p}_v) := \min_{\mathbf{W}_{u,v} \in U(\mathbf{p}_u, \mathbf{p}_v)} \sum_{i \in \mathcal{I}_u} \sum_{j \in \mathcal{I}_v} \mathbf{W}_{u,v}(i,j) \mathbf{D}(i,j). \tag{3}$$

Equation (3) is a special case of the earth mover's distance (EMD) [11], when the moment parameter $p = 1$ and the probability space is discrete. Moreover, PMD is a metric as long as $\mathbf{D}$ is a metric [37]. We call $d(\mathbf{p}_u, \mathbf{p}_v)$ the preference mover's distance (PMD) to highlight its connection to EMD. Being a metric has some nice properties that make the user distance meaningful. For example, the triangle inequality indicates that if both user A and user B are similar to a third user C, then user A and user B are also similar. Moreover, a user should be most similar to himself among all users if $\mathbf{D}(i,i) = 0$. In contrast, it is unclear whether BCF and HUSM also have these properties as they determine weights using heuristics.


u_1's preference $\mathbf{p}_1$:
0.8
0.2
Iron Man
Casablanca
$d(\mathbf{p}_0, \mathbf{p}_1) = 0.18$
0
0.1
0.3
0.5
0. 1
0
0.4
0.5
0.1
u_0's preference $\mathbf{p}_0$:
Spider Man
Bat Man
Titanic
$d(\mathbf{p}_0, \mathbf{p}_2) = 0.545$
0.3
0
0
0.4
0.2
0.1
0.3
0.7
u_2's preference $\mathbf{p}_2$:
Iron Man
Casablanca
(a)
(b)

	Iron Man	Casablanca
Spider Man	0.15	0.8
Bat Man	0.1	0.95
Titanic	0.9	0.05

Fig. 1. An example of PMD. (a) shows the preference distributions of u_0, u_1 and u_2 using histogram and the arrows depict the optimal transportation plan (i.e., $\mathbf{W}_{u,v}$) between the preference distributions. (b) is the distance matrix for the 5 movies, in which movies with the same genre have smaller distance, i.e., are more similar.

Illustration. Intuitively, $d(\mathbf{p}_u, \mathbf{p}_v)$ can be viewed as the minimum cost of transforming the ratings of user u to the ratings of user v, which we show in Fig. 1. $\mathbf{p}_u$ and $\mathbf{p}_v$ define two distributions of mass, while $\mathbf{D}(i, j)$ models the cost of moving one unit of mass from $\mathbf{p}_u(i)$ to $\mathbf{p}_v(j)$. Therefore, PMD can model the similarity between u and v even if they have no co-rated items. If two users like similar items, $\mathbf{W}_{u,v}(i, j)$ takes a large value for item pairs with small $\mathbf{D}(i, j)$, which results in a small distance. This is the case for u_0 and u_1 in Fig. 1 as they both like science fiction movies. In contrast, if two users like dissimilar items, $\mathbf{W}_{u,v}(i, j)$ is large for item pairs with large $\mathbf{D}(i, j)$, which produces a large distance. In Fig. 1, u_0 likes science fiction movies while u_2 likes romantic movies, and thus $d(\mathbf{p}_{u_0}, \mathbf{p}_{u_2})$ is large. Even if u_0 has no co-rated movies with u_1 and u_2, PMD still gives $d(\mathbf{p}_{u_0}, \mathbf{p}_{u_1}) < d(\mathbf{p}_{u_0}, \mathbf{p}_{u_2})$, which implies that u_0 is more similar to u_1 than to u_2.

Computation Speedup. An exact solution to the optimization problem in Eq. (3) takes a time complexity of $O(q^3 \log q)$ [36], where $q = |\mathcal{I}_u \cup \mathcal{I}_v|$. To reduce the complexity, we use the Sinkhorn algorithm [10], which produces a high-quality approximate solution with a complexity of $O(q^2)$. To speed up the lookup for similar users in large datasets, we employ HNSW [30], which is the state-of-the-art algorithm for similarity search. HNSW builds a multi-layer k-nearest neighbour (KNN) graph for the dataset and returns high quality nearest neighbours for a query with $O(\log N)$ distance computations, in which N is the number of users. With these two techniques, looking up for the top 100 neighbours takes only 0.02 s on average for a user and achieves a high recall of 99.2% for the Epinions dataset in our experiments. We conduct the experiment on a machine with two 2.0 GHz E5-2620 Intel(R) Xeon(R) CPU (12 physical cores in total), 48 GB RAM, a 450 GB SATA disk (6 Gb/s, 10k rpm, 64 MB cache), and 64-bit CentOS release 7.2.

Positive/Negative Feedback. We can split the user ratings into positive ratings $\mathbf{R}^p$, e.g., 3, 4 and 5 if a score of 1–5 is allowed, which indicates that the user likes the item, and negative ratings $\mathbf{R}^n$, e.g., 1 and 2, which indicates that the user dislikes the item. Based on $\mathbf{R}^p$ and $\mathbf{R}^n$, we define positive preference $\mathbf{p}_u^p$ and negative preference $\mathbf{p}_u^n$, i.e., $\mathbf{p}_u^p(i) = \frac{\mathbf{R}^p(u,i)}{\sum_{j \in \mathbf{R}^p} \mathbf{R}^p(u,j)}$ and $\mathbf{p}_u^n(i) = \frac{\frac{1}{\mathbf{R}^n(u,i)}}{\sum_{j \in \mathbf{R}^n} \frac{1}{\mathbf{R}^n(u,j)}}$. Then we can define more fine-grained user distances using Eq. (3), e.g., $d(\mathbf{p}_u^p, \mathbf{p}_v^p)$, $d(\mathbf{p}_u^n, \mathbf{p}_v^n)$, $d(\mathbf{p}_u^n, \mathbf{p}_v^p)$ and $d(\mathbf{p}_u^p, \mathbf{p}_v^n)$. A small $d(\mathbf{p}_u^n, \mathbf{p}_v^n)$ indicates that the two users dislike similar items and can be used to avoid making bad recommendations that may lose users. A small $d(\mathbf{p}_u^p, \mathbf{p}_v^n)$ or $d(\mathbf{p}_u^n, \mathbf{p}_v^p)$ means that the interests of the two users complement each other and may be used for friend recommendation in social networks. We may also construct composite PMD (CPMD) such as:

$$\tilde{d}(\mathbf{p}_u, \mathbf{p}_v) := \mu d(\mathbf{p}_u^p, \mathbf{p}_v^p) + (1 - \mu) d(\mathbf{p}_u^n, \mathbf{p}_v^n), \tag{4}$$

where $\mu \in [0, 1]$ is a tuning parameter weighting the importance of the distances of positive and negative preferences.

3 Experiments

We evaluate PMD by comparing its performance for NN-based recommendation with various user similarity measures. Two well-known datasets, i.e., MovieLens-1M [2] and Epinions [1], are used and their statistics are reported in Table 1. The rating user u gives to item i is predicted as a weighted sum of its top-K neighbours in the training set, i.e., $\hat{\mathbf{R}}(u,i) = \bar{u} + \sum_{v \in \mathcal{N}_u} \frac{s(u,v) \times (\mathbf{R}(v,i) - \bar{v})}{\sum_{v \in \mathcal{N}_u} s(u,v)}$ [13], in which $\bar{u}$ is the average of the ratings given by user u, $\mathcal{N}_u$ contains the top-K neighbours of u and $s(u,v)$ is the similarity between a user pair u and v. We convert PMD into a similarity measure using $s(u,v) = 2 - d(\mathbf{p}_u, \mathbf{p}_v)$ and divide all ratings into train/validation/test sets, with an 8:1:1 ratio. Hyper-parameters are tuned to be optimal on the validation set for all methods. The mean absolute error (MAE) and the root mean square error (RMSE) [15,31,32] of the predicted ratings on the test set are used to evaluate the recommendation performance.

Table 1. Data statistics.

	MovieLens	Epinions
#user	6,040	116,260
#item	3,959	41,269
#rating	1,000,000	181,394
sparsity	4.14%	0.0038%
#rating/user	166	1.56
#rating/item	250	4.40

Table 2. CPMD under different K and μ.

	MovieLens ($K = 200$)		Epinions ($K = 50$)			MovieLens ($\mu = 0.6$)		Epinions ($\mu = 0.6$)	
μ	MAE	RMSE	MAE	RMSE	K	MAE	RMSE	MAE	RMSE
0.2	0.7126	0.9019	0.8542	1.1340	30	0.7148	0.9064	0.8518	1.1294
0.4	0.6970	0.8851	0.8506	1.1302	50	0.7084	0.9052	**0.8458**	**1.1260**
0.6	**0.6918**	**0.8817**	**0.8458**	**1.1260**	100	0.6972	0.8898	0.8550	1.1456
0.8	0.6955	0.8875	0.8550	1.1456	200	**0.6918**	**0.8817**	0.8592	1.1435
0.95	0.6989	0.8915	0.8596	1.1520	300	0.6938	0.8846	0.8667	1.1506

Table 3. Comparison with other user similarity measures.

Dataset	Metric	COS	PCC	MSD	Jaccard	JMSD	NHSM	BCF	HUSM	PMD	CPMD
Movie lens	MAE	0.7477	0.7234	0.7387	0.7109	0.7024	0.7079	0.7044	0.7034	0.7019	**0.6918**
	RMSE	0.9394	0.9182	0.9293	0.9125	0.8982	0.9080	0.9089	0.9067	0.8935	**0.8817**
Epin ions	MAE	1.0476	1.0468	1.0449	1.0340	1.0392	1.0213	0.9846	0.9734	0.8757	**0.8458**
	RMSE	1.4412	1.4384	1.4380	1.4226	1.4291	1.3969	1.3014	1.2846	1.1701	**1.1260**

Table 4. Comparison with latent factor models.

Dataset	Metric	NMF	SVD	SVD++	PMD	CPMD
Movie lens	MAE	0.7252	0.6864	**0.6739**	0.7019	0.6918
	RMSE	0.9177	0.8741	**0.8629**	0.8935	0.8817
Epin ions	MAE	0.9444	0.9482	0.9439	0.8757	**0.8458**
	RMSE	1.2096	1.2154	1.2091	1.1701	**1.1260**

Item Similarity. Both MovieLens and Epinions come with side information for computing item similarities. For MovieLens, we compute movie similarity using Tag-genomes [3,42]. For Epinions, we evaluate item similarity by applying Doc2Vec [24] on the comments. Since both Tag-genome and doc2vec derive item similarity by cosine, we convert item similarity into distance using $\mathbf{D}(i,j) = \arccos(s(i,j))$, which is a metric on the item space. For fair comparison, the same item similarity matrix is used for PMD, BCF and HUSM[1].

Comparison Methods. COS, PCC and MSD are three classical user similarity measures. Jaccard, JMSD, NHSM, BCF, HUSM are five state-of-the-art measures. NMF [28], SVD [21] and SVD++ [20] are latent factor models for CF.

We report the performance of various similarity measures in Table 3, where PMD is based on Eq. (3) and CPMD is based on Eq. (4). The results show that PMD and CPMD consistently outperform other similarity measures and the improvement is more significant on the Epinions dataset which is much more sparse. We believe that our methods achieve good performance on sparse datasets mainly because it utilizes all rating information and derives the weights of the items using the optimal transportation theory, which works well when there are only few or no co-rated items. This is favorable as ratings are sparse in many real-world datasets [40]. CPMD achieves better performance than PMD, which suggests that it is beneficial to distinguish positive and negative feed-backs.

We also compare our methods with the latent factor models in Table 4. On the sparse Epinions dataset, both PMD and CPMD outperform the latent factor models. We report the performance of CPMD-based NN CF under different configurations of K and μ in Table 2. CPMD performs best when μ is around

[1] BCF and HUSM originally compute item similarity using the Bhattacharyya coefficient or the KL-divergence of ratings but we found that using the tag-genomes and doc2vec provides better performance.

0.6 on both datasets possibly because positive ratings can better represent the taste of a user than the negative ratings. In contrast, the optimal value of K is dataset dependent.

4 Conclusions

We proposed PMD, a novel user distance measure based on optimal transportation, which addresses the limitation of existing methods in dealing with datasets with few co-rated items. PMD also has the favorable properties of a metric. Experimental results show that PMD leads to better recommendation accuracy for NN-based CF than the state-of-the-art user similarity measures, especially when the ratings are highly sparse.

Acknowledgement. The authors thank Prof. Julian McAuley for his valuable suggestions on this paper, and Prof. Shengyu Zhang for his support. This work was supported by ITF 6904945, and GRF 14208318 & 14222816, and the National Natural Science Foundation of China (NSFC) (Grant No. 61672552).

References

1. https://cseweb.ucsd.edu/jmcauley/datasets.html
2. https://grouplens.org/datasets/movielens/
3. https://grouplens.org/datasets/movielens/tag-genome/
4. Farshad Aghabozorgi and Mohammad Reza Khayyambashi: A new similarity measure for link prediction based on local structures in social networks. Phys. A: Stat. Mech. Appl. **501**, 12–23 (2018)
5. Hyung Jun Ahn: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf. Sci. **178**(1), 37–51 (2008)
6. Al-bashiri, H., Abdulgabber, M.A., Romli, A., Hujainah, F.: Collaborative filtering similarity measures: revisiting. In: Proceedings of the International Conference on Advances in Image Processing, pp. 195–200. ACM (2017)
7. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
8. 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)
9. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
10. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems, vol. 26, pp. 2292–2300 (2013)
11. Cuturi, M., Solomon, J.M.: A primer on optimal transport. In: Tutorial of 31st Conference on Neural Information Processing Systems (2017)
12. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 101–109. ACM (2019)

13. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer, Boston (2011). https://doi.org/10.1007/978-0-387-85820-3_4
14. Guo, G., Zhang, J., Yorke-Smith, N.: A novel Bayesian similarity measure for recommender systems. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)
15. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
16. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. International World Wide Web Conferences Steering Committee (2017)
17. Huang, G., et al.: Supervised word mover's distance. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS 2016, pp. 4869–4877 (2016)
18. Jamali, M., Ester, M.: TrustWalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 397–406. ACM (2009)
19. Karamanolakis, G., Cherian, K.R., Narayan, A.R., Yuan, J., Tang, D., Jebara, T.: Item recommendation with variational autoencoders and heterogeneous priors. In: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, pp. 10–14. ACM (2018)
20. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
21. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **8**, 30–37 (2009)
22. Koutrika, G., Bercovitz, B., Garcia-Molina, H.: FlexRecs: expressing and combining flexible recommendations. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp. 745–758. ACM (2009)
23. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings of The 32nd International Conference on Machine Learning, pp. 957–966 (2015)
24. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
25. Levina, E., Bickel, P.J.: The earth mover's distance is the mallows distance: some insights from statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, pp. 251–256 (2001)
26. Ling, H., Okada, K.: An efficient earth mover's distance algorithm for robust histogram comparison. IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 840–853 (2007)
27. Liu, H., Zheng, H., 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)
28. Luo, X., Zhou, M., Xia, Y., Zhu, Q.: An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Trans. Ind. Inform. **10**(2), 1273–1284 (2014)

29. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–46. ACM (2007)
30. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Trans. Pattern Anal. Mach. Intell. **42**, 824–836 (2018)
31. Meng, Y., Chen, G., Li, J., Zhang, S.: Psrec: social recommendation with pseudo ratings. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 397–401. ACM (2018)
32. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
33. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie royale des sciences de Paris (1781)
34. Pan, Y., Li, D.-H., Liu, J.-G., Liang, J.-Z.: Detecting community structure in complex networks via node similarity. Phys. A: Stat. Mech. Appl. **389**(14), 2849–2857 (2010)
35. Patra, B.K., Launonen, R., Ollikainen, V., Nandi, S.: A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowl.-Based Syst. **82**, 163–177 (2015)
36. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467. IEEE (2009)
37. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), pp. 59–66 (1998)
38. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J., et al.: Item-based collaborative filtering recommendation algorithms. In: Www, vol. 1, pp. 285–295 (2001)
39. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating "word of mouth". In: CHI, vol. 95, pp. 210–217. Citeseer (1995)
40. Symeonidis, P., Nanopoulos, A., Papadopoulos, A.N., Manolopoulos, Y.: Collaborative filtering: fallacies and insights in measuring similarity. Universitaet Kassel (2006)
41. Thorpe, M., Park, S., Kolouri, S., Rohde, G.K., Slepčev, D.: A transportation LP distance for signal analysis. J. Math. Imaging Vis. **59**(2), 187–210 (2017)
42. Vig, J., Sen, S., Riedl, J.: The tag genome: encoding community knowledge to support novel interaction. ACM Trans. Interact. Intell. Syst. (TiiS) **2**(3), 13 (2012)
43. Wang, H., Wang, N., Yeung, D.-Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1235–1244. ACM (2015)
44. Wang, Y., Deng, J., Gao, J., Zhang, P.: A hybrid user similarity model for collaborative filtering. Inf. Sci. **418**, 102–118 (2017)
45. Wolsey, L.A., Nemhauser, G.L.: Integer and combinatorial optimization. Wiley, Hoboken (2014)
46. Yao, Y., Harper, F.M.: Judging similarity: a user-centric study of related item recommendations. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 288–296. ACM (2018)
47. Zheng, V.W., Cao, B., Zheng, Y., Xie, X., Yang, Q.: Collaborative filtering meets mobile recommendation: a user-centered approach. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
48. Zhong, J., Zhang, X.: Wasserstein autoencoders for collaborative filtering. arXiv preprint arXiv:1809.05662 (2018)