Improving novelty and diversity of nearest-neighbors recommendation by exploiting dissimilarities

Pablo Sánchez[1,3][0000−0003−1792−1706], Javier Sanz-Cruzado[2][0000−0002−7829−51740], and Alejandro Bellogín[3][0000−0001−6368−2510]

[1] Instituto de Investigación Tecnológica (IIT), Universidad Pontificia Comillas. Madrid. Spain
psperez@icai.comillas.edu
[2] School of Computing Science, University of Glasgow, Glasgow, United Kingdom
javier.sanz-cruzadopuig@glasgow.ac.uk
[3] Universidad Autónoma de Madrid, Madrid, Spain
{pablo.sanchezp,alejandro.bellogin}@uam.es

Abstract. Neighborhood-based approaches remain widely used techniques in collaborative filtering recommender systems due to their versatility, simplicity, and efficiency. Traditionally, these algorithms consider similarity functions to measure how close user or item interactions are. However, their focus on capturing similar tastes often overlooks divergent preferences that could enhance recommendations. In this paper, we explore alternative methods to incorporate such information to improve beyond-accuracy performance in this type of recommenders. We define three mechanisms based on various modeling assumptions to integrate differing preferences into traditional nearest neighbors algorithms. Our comparison on four well-known and different datasets shows that our proposed approach can enhance the novelty and diversity of the recommendations while maintaining ranking accuracy. Our implementation is available at https://github.com/pablosanchezp/kNNDissimilarities.

Keywords: Nearest neighbors · Beyond-accuracy evaluation · Dissimilarity.

1 Introduction

Motivation. Collaborative Filtering (CF) techniques are some of the most well-known approaches in recommender systems. These models analyze the user-item interactions in a system by establishing patterns between them in order to make personalized recommendations. Traditionally, CF algorithms are categorized in two different families: memory-based (or k-nearest neighbors, kNN), that generate recommendations by computing similarities between users and/or items [14] and model-based, that create a predictive model by approximating the interactions between the users and the items available in the system [8].

Model-based methods are currently the most widely used CF techniques due to the advances produced in both neural networks [3,11] and matrix factorization (MF) [5,9] approaches. Although these models perform well in top-n recommendation, they have notable drawbacks, including a lack of explainability and the need for many parameters to train the algorithms. In contrast, kNN approaches provide recommendations based on similarities between users/items. These methods are highly explainable, especially when compared to latent factors models [3,4], and flexible, being able to incorporate contextual information such as time or geographical location [25,27]. Although kNN algorithms are not always the most effective models, they are still considered strong baselines, being able to outperform recent neural network models [2]. Additionally, designing recommenders that perform well simultaneously in accuracy, novelty, and diversity is challenging, leading to the "accuracy-diversity" dilemma or tradeoff [6,12].

Open issues. The traditional collaborative filtering kNN schema focuses on exploiting the ratings of those users (or items) with the most similar behavior to the target user (or candidate item). Although previous works have shown that being restricted to this information is effective, the ratings of other users or items might also provide valuable insights to enhance the recommendations. Improving diversity and novelty in these methods, while maintaining accuracy, is challenging. This aligns with the accuracy-diversity dilemma acknowledged by the community, as previously discussed.

Our contributions. We explore the role that distant or dissimilar neighbors might have on the recommendations. Hence, we expand the traditional nearest neighbors models, not only to exploit the interactions of the users that are most similar to the target one, but also use the interactions of those other users that differ from her. First, we define how "dissimilar neighbors" can be incorporated into the recommendation process and combine them with classical neighborhood approaches to perform the final recommendations. We also propose metrics to capture the concept of "dissimilarity" between users, which will give higher values to those users who have rated the same items with disparate ratings, as this is interpreted as an evidence of differing preferences.

Therefore, the main contributions of this work are threefold: (i) three proposals for user similarities that capture when two users exhibit differing preferences (dissimilarity metrics); (ii) three schemes to integrate this information into a nearest neighbor formulation (neighbor models based on dissimilarities); (iii) and experimental evidence with four real-world datasets where the proposed models improve beyond-accuracy metrics without compromising their overall accuracy.

2 Exploiting dissimilarities in neighbor-based recommendation

The underlying assumption of traditional user-based (UB) nearest-neighbor recommenders is that *preferences for a user could be predicted by taking a weighted mean among those users with higher (positive) similarities*, and whose weights depend on predefined similarity scores that account for correlations or trends between users in the community [14]. A similar rationale is applied for the item-based (IB) scenario, where similarities between items are computed instead.

Although previous works have shown that being restricted to this information is effective, the ratings of the rest of the users (or items) might also provide valuable insights which could be used to improve our recommendations. For instance, knowing the most dissimilar users may help us discover which items are not interesting for the target user. In fact, our proposal is related to the idea of "dissimilarity" defined in [18,19], where the authors obtained promising results by increasing the diversity of the recommendations at the expense of obtaining a lower accuracy using a k-furthest neighbor algorithm. At the same time, it resembles the neighborhood-diversification from [24,26], where Yang et al. and Zanitti et al. defined a function to balance the diversity of a set of neighbors.

Nonetheless, in the approach we present next, we do not limit our models to exploit either the most similar or dissimilar neighbors. Instead, we present combination strategies that consider both neighborhoods to maximize the trade-off between recommendation accuracy and other dimensions like novelty and diversity. As we shall see later, these methods yield positive results.

***Neighbor models based on dissimilarities*.** To improve novelty and diversity, we propose three neighbor models that exploit similar and dissimilar neighbors (users or items) according to different hypotheses. They share the following formulation, borrowed from the traditional UB model (the item-based variation would be equivalent) [14]:

$$\begin{aligned} \hat{s}(u,i;\lambda,\gamma,\theta) &= \mathrm{agg}_\theta(\hat{s}^{\pm}(u,i;\lambda,\gamma),\hat{s}^{-}(u,i)); \\ \hat{s}^{\pm}(u,i;\lambda,\gamma) &= \textstyle\sum_{v\in N_u^+}(\lambda w_{uv}^+ + \gamma w_{uv}^-)\cdot r_{vi}; \\ \hat{s}^{-}(u,i) &= \textstyle\sum_{v\in N_u^-} w_{uv}^-\cdot r_{vi} \end{aligned} \tag{1}$$

where w_{uv}^+ denotes classical similarity metrics between users u and v and w_{uv}^- represents the dissimilarity metrics we introduce in the next section. agg_θ is an aggregation function and N_u^+ or N_u^- denote the neighborhoods computed either with w_{uv}^+ or w_{uv}^-.

It is important to note that none of these proposals increase the computational complexity of the classic neighbor-based model.

- Our first proposal for neighbor model (*nndiv*) only uses the nearest neighbors of a classic neighborhood-based approach, but considering both the similarity and dissimilarity scores of every neighbor with respect to the target user or item. By doing this, we are not considering dissimilar neighbors, but include in the prediction formula the dissimilarity of close neighbors, so *their differing stances with respect to the target user/item are taken into account*. Hence, $\mathrm{agg}_\theta(a,b)=a, \lambda=1, \gamma=\pm 1$.
- Our second proposal for neighbor model (*inds*) fuses the scores of near and far (dissimilar) neighbors independently; thus, $\mathrm{agg}_\theta(a,b)=a+\theta b, \lambda=1, \gamma=0, \theta\in\mathbb{R}$. In this case, differing preferences are integrated by *computing two neighborhoods independently*, which may be completely different, *and combine the prediction at the score level*.
- Our last proposal (*indr*) is an extension of the previous neighbor model, but *integrating the preferences at the last step of the recommendation (ranking level)* instead of when computing the score. Thus, we generate two different recommendation rankings, one obtained with a classic neighborhood and another with dissimilar neighbors; these rankings are later combined using ranking fusion techniques [10]. This is equivalent to using a rank aggregation function as agg_θ.

***Dissimilarity metrics*.** In our proposals for neighbor models, we need to discover dissimilar users or items with respect to a target user or item. Now we propose three metrics that exploit how differing or dissimilar neighbors should be captured. We focus on users, but item dissimilarities could be defined in the same way.

Table 1. Final statistics of the datasets used in the experiments.

Dataset	Users	Items	Ratings	Sparsity (%)	Rating scale
Mov20M	138,493	26,744	20,000,263	0.53998	[0.5-5]
GReads	18,892	25,475	1,378,033	0.28633	[0-5]
Vinyls (5-core)	75,258	64,443	1,097,592	0.02263	[1-5]
Lastfm	1,892	17,632	92,834	0.27828	[1-5]

- First, we propose to compute the average difference in ratings between the items consumed by both users (*rat-diff*). Here, a large value of this metric represents high discrepancies between the two users, understanding that *two users have differing views if their ratings/scores are as different as possible*. However, as we want this function bounded in the $[0, 1]$ interval, we average the inverse of the rating differences and return 1 minus the inverse of the previous score:

$$\text{sim}_{\text{rat-diff}}(u, v) = \left(1 - \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} \frac{1}{|r_{ui} - r_{vi}| + 1}\right) \tag{2}$$

  where I_{uv} represents the items consumed by both users u and v, and r_{ui} is the rating of user u for item i.
- As noted in the field [13], some users rate few items, while others have many interactions. To emphasize users who rate a larger set of common items differently, we apply a penalty factor using the Jaccard index on top of the previous metric (rat-diff). By doing this, we assume that *two users have differing views if their ratings are different and the support of these differences is large* (*rdsupp*):

$$\text{sim}_{\text{rdsupp}}(u, v) = \text{sim}_{\text{rat-diff}}(u, v) \cdot \frac{|I_{uv}|}{|I_u \cup I_v|} \tag{3}$$

- Our third approach (*bin-sets*) separately considers items rated positively (I_u^+, I_v^+) and negatively (I_u^-, I_v^-) by each pair of users u and v. Here, *two users have differing views if one likes what the other dislikes, and viceversa.* Hence, we take a binary perspective of preferences. A user scores an item positively if it exceeds a positive threshold δ^+ ($I_u^+ = \{i \in I_u | r_{ui} > \delta^+\}$) and negatively if it is below a negative threshold δ^- ($I_u^- = \{i \in I_u | r_{ui} < \delta^-\}$). The final formulation of this metric is as follows:

$$\text{sim}_{\text{bin-sets}}(u, v) = \frac{|I_u^+ \cap I_v^-| + |I_u^- \cap I_v^+|}{|I_u^+ \cup I_v^-| + |I_u^- \cup I_v^+|} \tag{4}$$

3 Experimental settings

Datasets and split partitions. We report experiments on the following four datasets: Movielens20M (Mov20M)[4], Goodreads spoilers (GReads)[5], CDs and Vinyls (Vinyls, from Amazon)[6], and HetRec Lastfm (Lastfm)[7]. We select these datasets because they

[4] Movielens20M dataset: https://grouplens.org/datasets/movielens/20m/
[5] GoodReads spoilers dataset: https://cseweb.ucsd.edu/∼jmcauley/datasets.html#spoilers
[6] Amazon CDs and Vinyls dataset: https://nijianmo.github.io/amazon/index.html
[7] HetRec 2011 Lastfm dataset: https://grouplens.org/datasets/hetrec-2011/

belong to different platforms and domains, from movies, books, and music. Their statistics are shown in Table 1 where we observe that there is a great variety in the number of users, items, interactions, and sparsity. We should note that on the CDs and Vinyls dataset we apply a 5-core forcing each user/item to have at least 5 interactions. For the Lastfm dataset, as it contains just the number of listenings per artist, we transform that implicit information to explicit ratings as follows: $r_{ui} = round\left(4\frac{l_{ui}}{L_u}\right) + 1$, where l_{ui} is the number of listenings from a specific user u for an artist i, and L_u is the maximum number of listenings from that user.

For all datasets, we perform the experiments using a global partition, in which we randomly split 80% of all interactions in each dataset as the training set and the rest as the test set. We follow the TrainItems methodology [17], in which we consider as candidate to be ranked every item the target user has not rated previously in the training set. In all cases, we classify as relevant any item the user rated in the test set with at least a value of 4, as the maximum rating is 5 in all datasets.

Evaluation metrics. Recommendation quality is measured in terms of ranking accuracy using nDCG, novelty (i.e., recommending less popular items to users) in terms of Expected Popularity Complement (EPC) [21], and diversity measured using Gini and Item Coverage (IC, number of different items recommended to the user) [22]. While we acknowledge other dimensions (such as serendipity) or definitions of these metrics are available (see [1,21]), we leave an analysis of those alternatives as future work.

All metrics are reported using a cutoff of 5. We also measure User Coverage (UC, percentage of users to whom we are able to perform a recommendation). Higher values imply a better performance of the recommender. This includes the Gini coefficient, for which we adopt the complement of the standard definition, as in [22]. For nDCG and EPC, we check whether the differences between systems are statistically significant using a one-tailed paired t-test with $p < 0.05$. As they provide a single global value for every system, statistical significance could not be computed for Gini, UC, and IC.

Recommendation methods. We report the following algorithms covering several recommendation families, to obtain a representative picture of the state-of-the-art: (i) Pop, recommends the items that have been consumed by the largest number of users. (ii) UB, pure non-normalized user-based neighborhood that recommends items that other similar users rated before [14]. (iii) IB, pure item-based neighborhood with a similar formulation to the UB model [14]. (iv) HKV: matrix factorization algorithm that uses Alternate Least Squares for optimization (from [5]). (v) BPRMF, Bayesian Personalized Ranking (a pairwise personalized ranking loss optimization algorithm) using a matrix factorization approach (from [16]). (vi) EASEr, Embarrassingly Shallow Autoencoders for Sparse Data from [20]. (vii) RP$^3\beta$, graph-based method from [15].

Against the UB and IB algorithms, we shall compare our proposed variations defined in Section 2 (kNN$_{\mathrm{nndiv}}$, kNN$_{\mathrm{inds}}$, and kNN$_{\mathrm{indr}}$, where kNN is either UB or IB). These approaches obtain the recommendations according to different neighbor models, either using UB or IB as base formulation, together with user/item dissimilarity metrics,

Table 2. Performance results in terms of nDCG@5 for all the recommendation algorithms in the four analyzed datasets. The best result is shown in bold, underlined the best kNN. Statistical improvements (one-tailed t-test p < 0.05) w.r.t. Pop, UB, IB, HKV, BPRMF, EASEr, RP$^3\beta$ are superscripted with a, b, c, d, e, f, and g respectively.

Datasets	**Pop**$^{(a)}$	**UB**$^{(b)}$	**IB**$^{(c)}$	**HKV**$^{(d)}$	**BPRMF**$^{(e)}$	**EASEr**$^{(f)}$	**RP**$^3\beta^{(g)}$
Mov20M	0.126	$\underline{0.300}^{aceg}$	0.205^{ae}	0.312^{abceg}	0.182^{a}	$\mathbf{0.319}^{abcdeg}$	0.260^{ace}
GReads	0.024	0.099^{aef}	$\underline{\mathbf{0.114}}^{abdefg}$	0.104^{abefg}	0.050^{a}	0.081^{ae}	0.097^{aef}
Vinyls	0.003	$\underline{\mathbf{0.068}}^{acdefg}$	0.056^{adefg}	0.054^{aefg}	0.029^{a}	0.052^{aeg}	0.037^{ae}
Lastfm	0.096	$\underline{0.278}^{aceg}$	0.222^{a}	0.306^{aceg}	0.224^{a}	$\mathbf{0.308}^{aceg}$	0.201^{a}

Table 3. Performance of our kNN variations based on dissimilarities. Best results in bold, and * indicates significant gains with respect to the kNN baseline (one-tailed t-test $p < 0.05$).

Dataset	**Rec**	**nDCG**	**EPC**	**Gini**	**IC**	**UC**
Mov20M	kNN/UB	0.300	0.772	0.004	4.8%	100%
	kNN$_{\text{nndiv}}$	0.301	0.772	0.004	4.7%	100%
	kNN$_{\text{inds}}$	**0.304***	**0.783***	**0.005**	**5.4%**	100%
	kNN$_{\text{indr}}$	0.282	0.763	0.004	4.3%	100%
GReads	kNN/IB	0.114	0.974	0.081	36.2%	99.38%
	kNN$_{\text{nndiv}}$	**0.119***	**0.976***	**0.086**	37.1%	99.38%
	kNN$_{\text{inds}}$	0.115	0.974	0.080	36.0%	99.38%
	kNN$_{\text{indr}}$	0.102	0.970	0.081	**37.5%**	99.38%

Dataset	**Rec**	**nDCG**	**EPC**	**Gini**	**IC**	**UC**
Vinyls	kNN/UB	0.068	0.998	0.078	37.9%	99.99%
	kNN$_{\text{nndiv}}$	0.068	0.998	0.076	37.1%	99.99%
	kNN$_{\text{inds}}$	**0.069**	0.998	0.075	37.1%	99.99%
	kNN$_{\text{indr}}$	0.062	0.998	**0.085**	**41.6%**	99.99%
Lastfm	kNN/UB	0.278	0.890	**0.007**	**4.3%**	99.89%
	kNN$_{\text{nndiv}}$	0.279	**0.891**	**0.007**	4.2%	99.89%
	kNN$_{\text{inds}}$	**0.281**	**0.891**	**0.007**	**4.3%**	99.89%
	kNN$_{\text{indr}}$	0.268	0.887	0.006	4.0%	99.89%

as defined in Equations 2, 3, and 4. We used grid search for hyperparameter selection[8], optimizing nDCG@5.

4 Results

Performance comparison of baseline models. Table 2 shows how the recommendation algorithms compare against each other in terms of nDCG@5. We observe that, despite their simplicity, kNN approaches obtain competitive results in every considered dataset. In fact, in GReads and Vinyls they achieve the best performance, and these improvements tend to be statistically significant. It should be noted that UB outperforms IB in three out of the four datasets (all except GReads). Because of this, in the rest of the experiments we shall use in each dataset the best kNN approach and its corresponding variations.

Beyond-accuracy performance of dissimilarity-based neighbor models. Table 3 compares how our dissimilarity-based neighbor proposals allow to improve novelty and diversity metrics with respect to the baseline methods (either classic UB or IB), while keeping the same accuracy levels or slightly better. We observe that among our three proposals, kNN$_{\text{inds}}$ is the one that in general obtains a better result both in terms of accuracy and in terms of novelty and diversity, in some cases far surpassing the pure UB algorithm, as in Mov20M (where the difference is actually statistically significant). Our kNN$_{\text{nndiv}}$ proposal also achieves improvements, especially in GReads where diversity, measured with Gini, is improved by a 6.2%, and novelty, measured with EPC, rises by

[8] For the sake of space, tested and optimal parameters are included in the code repository, available at https://github.com/pablosanchezp/kNNDissimilarities.

Table 4. Performance results for the different dissimilarities. Best results in bold.

	Mov20M			GReads			Vinyls			Lastfm		
Rec	**nDCG**	**EPC**	**Gini**	**nDCG**	**EPC**	**Gini**	**nDCG**	**EPC**	**Gini**	**nDCG**	**EPC**	**Gini**
kNN	0.3003	0.7721	0.0045	0.1140	0.9741	0.0810	0.0683	0.9982	0.0780	0.2778	0.8901	0.0070
$k\text{NN}_{\text{inds}}$ (rat-diff)	0.3019	0.7818	**0.0053**	0.1134	0.9741	0.0806	0.0665	**0.9986**	**0.1163**	0.2799	**0.9088**	**0.0117**
$k\text{NN}_{\text{inds}}$ (rdsupp)	0.3025	0.7792	0.0051	0.1145	**0.9745**	**0.0821**	**0.0686**	0.9982	0.0750	**0.2807**	0.8912	0.0072
$k\text{NN}_{\text{inds}}$ (bin-sets)	**0.3039**	**0.7832**	**0.0053**	**0.1148**	0.9743	0.0799	0.0684	0.9982	0.0767	0.2802	0.8907	0.0071

a 0.2%, while nDCG accuracy increases (significantly) by 4.4%. This shows that combining dissimilarity models with classical neighborhood approaches may improve the recommendations in multiple dimensions.

Dissimilarity definition comparison. We now study in Table 4 the effectiveness of the three dissimilarity metrics proposed in Section 2. We limit this analysis to the $k\text{NN}_{\text{inds}}$ model, as it achieves the best results in our previous analysis. A first observation highlights that all dissimilarities are able to improve the baseline in terms of nDCG, EPC, and Gini. Vinyls and GReads are the only datasets where this does not occur in all the cases. When comparing these metrics, there is not a clear winner on accuracy for this neighbor model. However, rat-diff stands out in both novelty and diversity, achieving the best EPC and Gini results in 3/4 datasets. Considering its nDCG results, this similarity combined with the $k\text{NN}_{\text{inds}}$ model provides the best combination between accuracy and beyond-accuracy effectiveness of all the neighbor models.

5 Conclusions and future work

This paper presents various approaches to exploit dissimilarity metrics in neighborhood-based recommendations, proposing hypotheses on identifying and integrating dissimilar users or items. In the reported results, we always obtain an improvement in novelty and diversity with respect to the corresponding nearest neighbor approach, and in some cases better (or at least, equal) performance in terms of accuracy.

Nonetheless, we believe there is still room for improvement. One of our main limitations is that our approaches have only been evaluated in rating-based datasets. As future work, we would like to adapt these approaches to an implicit information environment to analyze their potential. Besides, other definitions of dissimilarities might be defined by incorporating contextual information such as sequentiality and time, as well as item characteristics like genres, sentiments, tags, etc. [23]. Additionally, we aim to enhance the results with other novelty and diversity metrics that utilize this item content information [1,21], and through re-ranking approaches, that have been used in the past to diversify user recommendations [7].

Acknowledgments. This work was supported by Grant PID2022-139131NB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF, a way of making Europe."

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Clarke, C.L.A., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: Myaeng, S., Oard, D.W., Sebastiani, F., Chua, T., Leong, M. (eds.) Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008. pp. 659–666. ACM (2008). https://doi.org/10.1145/1390334.1390446, http://doi.acm.org/10.1145/1390334.1390446
2. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Bogers, T., Said, A., Brusilovsky, P., Tikk, D. (eds.) Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019. pp. 101–109. ACM (2019). https://doi.org/10.1145/3298689.3347058, https://doi.org/10.1145/3298689.3347058
3. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Huang, J., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., Liu, Y. (eds.) Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. pp. 639–648. ACM (2020). https://doi.org/10.1145/3397271.3401063, https://doi.org/10.1145/3397271.3401063
4. Hu, R., Pu, P.: A comparative user study on rating vs. personality quiz based preference elicitation methods. In: Conati, C., Bauer, M., Oliver, N., Weld, D.S. (eds.) Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI 2009, Sanibel Island, Florida, USA, February 8-11, 2009. pp. 367–372. ACM (2009). https://doi.org/10.1145/1502650.1502702, https://doi.org/10.1145/1502650.1502702
5. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy. pp. 263–272. IEEE Computer Society (2008). https://doi.org/10.1109/ICDM.2008.22, https://doi.org/10.1109/ICDM.2008.22
6. Isufi, E., Pocchiari, M., Hanjalic, A.: Accuracy-diversity trade-off in recommender systems via graph convolutions. Inf. Process. Manag. **58**(2), 102459 (2021). https://doi.org/10.1016/J.IPM.2020.102459, https://doi.org/10.1016/j.ipm.2020.102459
7. Kaminskas, M., Bridge, D.: Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. TiiS **7**(1), 2:1–2:42 (2017). https://doi.org/10.1145/2926720, https://doi.org/10.1145/2926720
8. Koren, Y., Bell, R.M.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 77–118. Springer (2015). https://doi.org/10.1007/978-1-4899-7637-6_3
9. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer **42**(8), 30–37 (2009). https://doi.org/10.1109/MC.2009.263, https://doi.org/10.1109/MC.2009.263
10. Lee, J.: Analyses of multiple evidence combination. In: Belkin, N.J., Narasimhalu, A.D., Willett, P., Hersh, W.R., Can, F., Voorhees, E.M. (eds.) SIGIR '97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 27-31, 1997, Philadelphia, PA, USA. pp. 267–276. ACM (1997). https://doi.org/10.1145/258525.258587, https://doi.org/10.1145/258525.258587

11. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: Champin, P., Gandon, F., Lalmas, M., Ipeirotis, P.G. (eds.) Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018. pp. 689–698. ACM (2018). https://doi.org/10.1145/3178876.3186150, https://doi.org/10.1145/3178876.3186150
12. Liu, J.G., Shi, K., Guo, Q.: Solving the accuracy-diversity dilemma via directed random walks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **85**(1), 016118 (2012)
13. McLaughlin, M.R., Herlocker, J.L.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Sanderson, M., Järvelin, K., Allan, J., Bruza, P. (eds.) SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004. pp. 329–336. ACM (2004). https://doi.org/10.1145/1008992.1009050, https://doi.org/10.1145/1008992.1009050
14. Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 37–76. Springer (2015). https://doi.org/10.1007/978-1-4899-7637-6_2
15. Paudel, B., Christoffel, F., Newell, C., Bernstein, A.: Updatable, accurate, diverse, and scalable recommendations for interactive applications. ACM Trans. Interact. Intell. Syst. **7**(1), 1:1–1:34 (2017). https://doi.org/10.1145/2955101, https://doi.org/10.1145/2955101
16. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: Bilmes, J.A., Ng, A.Y. (eds.) UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009. pp. 452–461. AUAI Press (2009)
17. Said, A., Bellogín, A.: Comparative recommender system evaluation: benchmarking recommendation frameworks. In: Kobsa, A., Zhou, M.X., Ester, M., Koren, Y. (eds.) Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014. pp. 129–136. ACM (2014). https://doi.org/10.1145/2645710.2645746, http://doi.acm.org/10.1145/2645710.2645746
18. Said, A., Fields, B., Jain, B.J., Albayrak, S.: User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In: Bruckman, A.S., Counts, S., Lampe, C., Terveen, L.G. (eds.) Computer Supported Cooperative Work, CSCW 2013, San Antonio, TX, USA, February 23-27, 2013. pp. 1399–1408. ACM (2013). https://doi.org/10.1145/2441776.2441933, https://doi.org/10.1145/2441776.2441933
19. Said, A., Jain, B.J., Kille, B., Albayrak, S.: Increasing diversity through furthest neighbor-based recommendation. In: Proceedings of the WSDM'12 Workshop on Diversity in Document Retrieval (DDR'12) (2012)
20. Steck, H.: Embarrassingly shallow autoencoders for sparse data. In: Liu, L., White, R.W., Mantrach, A., Silvestri, F., McAuley, J.J., Baeza-Yates, R., Zia, L. (eds.) The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019. pp. 3251–3257. ACM (2019). https://doi.org/10.1145/3308558.3313710, https://doi.org/10.1145/3308558.3313710
21. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: Mobasher, B., Burke, R.D., Jannach, D., Adomavicius, G. (eds.) Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011. pp. 109–116. ACM (2011). https://doi.org/10.1145/2043932.2043955, http://doi.acm.org/10.1145/2043932.2043955

22. Vargas, S., Castells, P.: Improving sales diversity by recommending users to items. In: Kobsa, A., Zhou, M.X., Ester, M., Koren, Y. (eds.) Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014. pp. 145–152. ACM (2014). https://doi.org/10.1145/2645710.2645744, https://doi.org/10.1145/2645710.2645744
23. Wang, J., Bellogín, A., Cantador, I.: Integrating sentiment features in factorization machines: Experiments on music recommender systems. In: Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2024, Cagliari, Italy, July 1-4, 2024. pp. 183–188. ACM (2024). https://doi.org/10.1145/3627043.3659561, https://doi.org/10.1145/3627043.3659561
24. Yang, C., Ai, C.C., Li, R.: Neighbor diversification-based collaborative filtering for improving recommendation lists. In: 10th IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, HPCC/EUC 2013, Zhangjiajie, China, November 13-15, 2013. pp. 1658–1664. IEEE (2013). https://doi.org/10.1109/HPCC.and.EUC.2013.234, https://doi.org/10.1109/HPCC.and.EUC.2013.234
25. Yuan, Q., Cong, G., Ma, Z., Sun, A., Magnenat-Thalmann, N.: Time-aware point-of-interest recommendation. In: Jones, G.J.F., Sheridan, P., Kelly, D., de Rijke, M., Sakai, T. (eds.) The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013. pp. 363–372. ACM (2013). https://doi.org/10.1145/2484028.2484030, https://doi.org/10.1145/2484028.2484030
26. Zanitti, M., Kosta, S., Sørensen, J.K.: A user-centric diversity by design recommender system for the movie application domain. In: Champin, P., Gandon, F., Lalmas, M., Ipeirotis, P.G. (eds.) Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018. pp. 1381–1389. ACM (2018). https://doi.org/10.1145/3184558.3191580, https://doi.org/10.1145/3184558.3191580
27. Zhang, J., Chow, C.: Core: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations. Inf. Sci. **293**, 163–181 (2015). https://doi.org/10.1016/j.ins.2014.09.014, https://doi.org/10.1016/j.ins.2014.09.014