Utilising Crowdsourcing to Assess the Effectiveness of Item-based Explanations of Merchant Recommendations

Denis Krasilnikov[1,3], Oleg Lashinin[1], Maksim Tsygankov[1], Marina Ananyeva[1,2] and Sergey Kolesnikov[1]

[1]Tinkoff, 2-Ya Khutorskaya Ulitsa, 38A, bld. 26, Moscow, 117198, Russian Federation
[2]National Research University Higher School of Economics, Myasnitskaya Ulitsa, 20, Moscow, 101000, Russian Federation
[3]Lomonosov Moscow State University, Ulitsa Kolmogorova, 1, bld. 52, Moscow, 119234, Russian Federation

Abstract
The explainability of recommendations is a common research topic among researchers and providers of recommender systems. Numerous approaches and inference types were developed in order to find explanations for recommendations. For example, we can send users the following recommendation with an explanation: ”Since you recently made a purchase from merchant X, we suggest you merchant Y”. A variety of methods can be used to produce the (X, Y) item pairs with this explanation logic. Despite this, some users might not understand the logical connection between the recommendation Y and explanation X. In this study, we validate 23,000 recommendation explanations with the help of 400 crowdworkers. Additionally, we suggest a novel method for evaluating the quality of the (X, Y) item pair explanations based on crowdworkers' responses. Finally, we evaluate 9 different approaches and produce interesting findings. We hope that, in future research, our method will be expanded upon and further studied for additional types of explanations and domains.

Keywords
recommender systems, explainable recommendations, evaluation study, crowdsourcing

1. Introduction

These days, recommender systems are an integral part of many areas in people's lives. They are capable of influencing people's choice of films, clothing, tourist destinations, food- and health-related habits, and much more. As a rule, such algorithms leverage previous users' actions in order to show current users items that can potentially be interesting to them. Content that is personalized in such a way is attractive to users, driving them to interact more with various online stores, dating services, music, videos, and others.

Recent works highlight the main problems of merchant recommendations. For instance, some banks provide merchant reward systems [1, 2, 3]. When bank clients make transactions with

WSDM 2023 Crowd Science Workshop on Collaboration of Humans and Learning Algorithms for Data Labeling, March 3, 2023, Singapore
d.i.krasilnikov@tinkoff.ru (D. Krasilnikov); o.a.lashinin@tinkoff.ru (O. Lashinin); m.r.tsygankov@tinkoff.ru (M. Tsygankov); m.ananyeva@tinkoff.ru (M. Ananyeva); scitator@gmail.com (S. Kolesnikov)
0009-0009-6864-392X (D. Krasilnikov); 0000-0001-8894-9592 (O. Lashinin); 0000-0002-9885-2230 (M. Ananyeva); 0000-0002-4820-987X (S. Kolesnikov)

particular merchants, they receive cashback automatically. With many such offers available, personalization of the rewards section improves user profit [1]. In order to do so, it is possible to find the most relevant offers for each individual user based on their transaction history. For example, some recent works demonstrated experiment results on different real-world transaction datasets [4, 5, 6]. These works proved that recommendation models are capable of accurately predict users' future behavior by analysing past transactions. However, it is important to note providing explanations for such merchant recommendations is not well studied yet.

In this paper, we research the problem of offering users explanations as well as merchant recommendations. Formally, we have a dataset **D** with transaction histories of anonymous users with a number of merchants. Our task is to not only suggest the most appropriate merchants for each user, but to also explain each personalized suggestion.

There are various types of explanations [7] for results provided by recommender systems. One of them is item-based explanations, where textual patterns consist of a few items, connected by certain conditions. For instance, we can show the user the following message: ”We recommend you merchant Y because you purchased from merchant X”. This communication can be fully defined by the (X, Y) merchant pair. Both of them should be represented in the dataset **D**. The item Y is received from a recommender model **M** built on historical transactions. The item X must be in the history of the user for whom we are providing the recommendation explanation. Otherwise, the statement in the communication will obviously be wrong.

We chose this method due to a number of advantages. Firstly, it does not require additional knowledge about merchants. Secondly, it is quite simple to implement in an interface for testing with real users. Finally, there are many different approaches and heuristics to retrieve the (X, Y) merchant pairs. However, not all pairs may be valid. For example, a merchant pair consisting of a bar and children's store can be perceived negatively by real users. To avoid such situations, we suggest pre-screening some merchant pairs using crowdsourcing platforms. If some of them look questionable together when considered by real users, then they should be filtered out using additional labeling. This idea is illustrated in Figure 1.

In this paper, we research the validation of X → Y pairs for further use in recommender systems for real clients. We provide an extensive survey for 23,000 merchant pairs. 400 crowdworkers share their opinions as if they were seeing these pairs in a scenario with real recommender systems. Based on the results of these surveys, we evaluate 9 approaches for explainable recommendations. This helps us estimate the quality of recommendation explanations in offline experiments based on the feelings of real people. Specifically, the main contribution of this paper can be listed as follows:

- We have asked 400 real people on a crowdsourcing platform to evaluate 23,000 pairs of explanations. We share our results in an anonymized dataset.
- We describe a new way to evaluate explanations of merchant recommendations based on users' feedback. Our approach makes it possible to separate the development of algorithms and the evaluation of explanation quality into independent processes. As a result, we can collect user feedback once and then use it many times for different approaches.
- We provide the results of experiments with 9 recommender algorithms as well as heuristics that generate the most appropriate explanation pairs (X, Y). We demonstrate how the

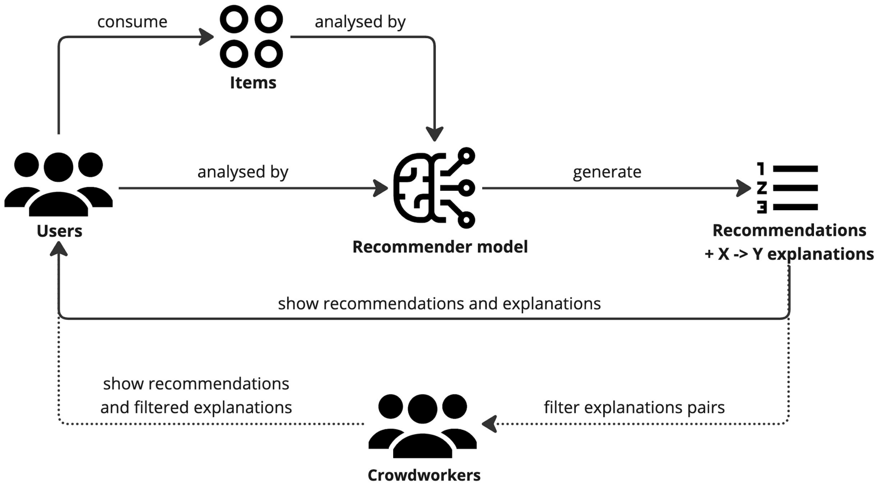

Figure 1: In order to provide explanations for the (X, Y) merchant pair, we suggest validating such pairs using crowdsourcing platforms. This will help avoid questionable pairs and possible negative feedback from users.

ranking quality of these pairs matches the opinions of real people.

2. Related Work

Explanations based on X → Y pairs can be considered a relatively well-studied topic in the research community. For instance, iALS [8] is capable of not only generating high-quality recommendations [9], but can also provide X → Y explanations. To do so, its optimization algorithm learns the weights of item-item similarities. Moreover, iALS takes into account the significance of interactions to each user. Another work [10] demonstrated a way of mining post-hoc explanations. This method is based on association rules and can be applied to any latent factor recommender model. A recent study [11] used a causal rule learning model to retrieve personal post-hoc explanations. Finally, there are studies that analyze the influence of input data on output scores [12, 13]

Crowdsourcing can be very helpful in the field of explainable recommendations. For instance, crowdworkers can generate textual explanations [14], provide information about cold start items [15] and evaluate the explanations offered for recommendations [16]. In a recent work [17], crowdworkers helped improve the quality of recommendations via a human-in-the-loop framework. The crowdsourced opinions helped increase accuracy of personalized suggestions. Moreover, according to [18, 19], crowdworkers are capable of evaluating different explanations in flexible experiment settings.

To the best of our knowledge, explainable merchant recommendations are not well-studied in the broad research community. Recent works tend to consider only the quality of merchant recommendations [4, 5, 6]. However, we are of the opinion that the explainability of such models is a topic that is worth investigating in future works.

Table 1
Descriptive statistics of the transaction dataset.

Version	#users	#merchants	#transactions	#density
Original	88800	5400	12M	2.51%
After preprocessing	88500	350	10M	33.32%

3. Dataset Collection

In this section, we describe our approach to collecting and processing data.

The TTRS dataset (Tinkoff Transaction Recommender System) served as the source for all of our data. This open-source dataset with detailed statistics was provided in [5]. TTRS contains real clients' transactions with some merchants such as brands, retail chains and services. In the open-source version of the dataset, each transaction contains the user id, merchant id[1], merchant category and transaction timestamp. As this dataset was provided by us, we can enrich it with additional information. Concretely, merchant names of brands were used in our crowdsoursing experiments.

We chose only a set *TOP* of top 350 most popular merchants. According to original data, these merchants accounted for about 87% of all transactions. The specifications of this dataset can be found in Table 1.

Additionally, we create a square sparse matrix C with a shape of (350, 350) to represent the "relevance" of items. In order to collect this data, we sampled around 23,000 unique (X, Y) pairs randomly, where $X \neq Y$ and $X, Y \in TOP$. This number was determined by our study's limited budget. Since it is possible to create 349 * 349 = 121,801 pairs in total, we have selected about 19% of all possible pairs.

To conduct our survey, we used an internal crowdsourcing platform at Tinkoff. Since machine learning algorithms require a large amount of labeled data [20], machine learning developers can use this platform for their needs. The Tinkoff crowdsourcing platform makes it possible to label datasets based on people's opinions. We used this platform to ask crowdworkers the following question: "We recommend you merchant Y because you bought from merchant X. **Based on this explanation, do you think the model works correctly?**". This question is preceded by a brief description of both merchants. In addition, respondents were given instructions before starting to answer the questions. The instruction requested crowdworkers to imagine a situation in which they actually made a purchase from merchant X. Afterwards, they received a communication recommending merchant Y. The wording of the question was chosen based on the analysis of previous works [18, 19, 16].

The basic intuition is that if the (X, Y) pair seems illogical, the crowdworker will answer in the negative. If the pair is perceived as logical by the respondent, then we will get a positive answer from them.

During the experiment, we asked 400 people to respond to our questions. To improve the quality of the experiment, we asked five different crowdworkers to answer every question. This made it possible for us to determine how many votes are needed to consider the recommendation explanation correct (3 out of 5, 4 out of 5, or 5 out of 5). We received 2,350 pairs where at least 3

[1] Both user id and merchant id columns are anonymized

Table 2
Notation used in the paper.

U	a set of users
I	a set of items
u, i, t	a particular user/item/timestamp respectively
X	an item from the user's history, used to explain a recommendation
Y	an item to be recommended for the user
$B_{u,i,t}$	a set of transactions from the dataset
$S_{u,i,t}$	a sample of $B_{u,i,t}$ with distinct (u, i) pairs
$K(u)$	a number of items consumed by the user u
$r_{u,i,t}$	an ordered set of transactions made by the user u, with timestamps
I_u	an unordered set of items consumed by the user u
M	a fitted recommender model
E	an approach ranking (X, Y) explanation pairs
C	a matrix which contains averaged answers from respondents
$s_{u,X,Y}$	a set of scores for ranking (X, Y) pairs for a user u

out of 5 people said that the model works correctly. Thus, the cell of matrix $C[X][Y]$ equals 1 if 3 out of 5 respondents label the pair as correct. Otherwise, $C[X][Y]$ is 0.

4. Evaluating Explanation Quality

In this section, we describe a new method of evaluating X → Y explanations. We summarize our notation in Table 2. Let's assume that we have a set of users U and a set of items $I = \{i_1, \dots, i_{|I|}\}$. We have a training part of a dataset which can be represented by the set of transactions $B_{u,I,t} = \{B_{u_k,i_k,t_k}\}$, where u_k, i_k, t_k are the user, item and timestamp respectively. For simplicity, we will consider only the following subset $S_{u,me,t} \subset \{B_{u_k,i_k,t_k}\}$ such that it contains only the maximum timestamps for each (u, i) pair. This makes it so that the recommender system should only predict new merchants for users. The modeling of recurring transactions remains to be researched in future works.

A user u interacted with a set of $K(u)$ unique items $r_{u,i,t} = \{B_{u,1,t_1}, \dots, B_{u,|K(u)|,t_{K(u)}}\}$, ordered by timestamps. Let's assume that there is a method E that can generate explainability scores $s_{u,X,Y} = \{s_{u,1,Y}, \dots, s_{u,|K(u)|,Y}\}$. If we have items i and j, and $s_{u,i,Y} < s_{u,j,Y}$, we will be explaining the recommendation of Y with item j. It is important to have valid pairs with higher scores and invalid (illogical) pairs with the lower scores.

Therefore, it is possible to compute quality ranking metrics under $s_{u,X,Y}$ for some subset of (X, Y) pairs and users from U. The proposed approach is described in detail in Algorithm 1. The key idea of this method is to take all the items user u interacted with. Then, we leave only the pairs of merchants that meet the following conditions: (a) Y was recommended, (b) X is in the user's purchase history, (c) (X, Y) pair is validated by the respondents. If for user u there are at least two or more candidates x_k for explaining each recommended item Y, we can sort these candidates by $s_{u,x_k,Y}$. Finally, quality ranking metrics compare the sorted lists of candidates and people's opinions. Higher metric values prove that a method E can effectively retrieve explanation pairs, while low values may indicate that users might dislike certain explanations because they find them incorrect.

Algorithm 1: The algorithm proposed for evaluating explanation pairs

```
Data: a ground truth matrix $C$, a set of transaction $r_{u,i,t}$ for each user from $U$,
      a trained recommender model $M$, a method $E$ for explanation
      generation, selected quality ranking metrics.
Result: Calculated ranking quality metrics
foreach user $u \in U$ do
    generate top-K recommendations $Y_j$ with model $M$;
    foreach $y_j \in Y_j$ do
        compute $s_{u,x_k,y_j} = \{s_{u,x_q,y_j} \mid q = 1, \dots, K(u) \wedge C[x_q][y_j] \in \{0,1\}\}$;
        if $|s_{u,x_k,y_j}| = 0$ then
            continue;
        else
            let $g_{u,x_k} = \{C[i][y_j] \mid i \in$ items from $r_{u,i,t}\}$ ;
            sort $s_{u,x_k,y_j}$ and $g_{u,x_k}$ by $s_{u,x_k,y_j}$ ;
            foreach metric $\in$ metrics do
                calculate metric($s_{u,x_k,y_j}$, $g_{u,x_k}$);
            end
        end
    end
end
```

5. Methods to Rank Explanation Pairs

In this section, we briefly describe the methods of generating explanation pairs which we included in our work.

- **Random**. This method simply generates random scores $s_{u,X,Y}$. It is included in order to calculate the relative improvement of other approaches.
- **Chrono**. Some sequential recommenders assume that a user's future interactions are caused by their recent interactions [21, 22]. Chrono is a heuristic approach that works under this assumption. Specifically, it gives higher scores for the most recent items. The last recent item has the highest score $s_{u,x_{K(u)},Y}$ because of the assumption that future test interactions are mostly caused by the last interaction. Formally, let $s_{u,i,Y} = \frac{1}{\underset{i \in I_u}{rank}(\underset{date \in t_i}{\max}(date))}$. Here t_i is all possible dates for user u.
- **MostPop**. This baseline ranks items according to their popularity in the training part of the dataset. This popularity is defined as the total number of transactions.
- **PersonTop**. In this approach, we calculate the personal frequencies of user interactions with every item. The more a user purchased from a certain merchant, the higher the value of $s_{u,i,Y}$. If a user buys from two different merchants an equal number of times, the order between them is defined by their popularity, similar to MostPop.

- **Similar Category MostPop**. Merchants in our dataset have categories. People may consider it reasonable if they see merchants X and Y from one category. Therefore, we can assign higher scores to merchants from the same category and lower to those from different ones. Formally, $s_i = \begin{cases} 2 + mp_i, \text{ if X and Y belong to the same category} \\ mp_i, \text{ if X and Y are from different categories} \end{cases}$
  Where $mp_i \in [0, 1]$ is the score from MostPop.
- **Implicit ALS** [8]. This model not only shows competitive performance on top-n recommendations [9], but is capable of generating explanations for recommendations.
- **Similar Category + iALS**. This method is similar to **Similar Category MostPop** with items sorted according to iALS scores.
- **Association Rules**. This method makes it possible to generate explanations for any recommender model [10]. There are different metrics to compute the rule. In our work, we include confidence, support and leverage.
- **EASE** [23]. This approach is a shallow autoencoder that has an item-item weight matrix $W[X][Y]$. This matrix can be considered a method to calculate $s_{u,X,Y}$. Formally, $s_{u,X,Y} = W[X][Y]$,

It is important to note that **Random**, **MostPop**, **Association Rules**, **EASE** do not depend on $r_{u,i,t}$. They rank all I_u items, and the relative order of items does not change if new interactions are made. Alternatively, **Chrono**, **PersonTop**, **iALS**, **Similar Category + iALS** take into account the set of I_u and rank explanation pairs in person.

6. Experiments

We use the feedback collected from the crowdworkers to evaluate the explanation quality of different heuristics and algorithms. The choice of recommender model that provides recommendations is not the focus of our work. Therefore, we take an MF-based method iALS [8], which is a powerful recommender model for the top-n recommendation task [9, 24] that is capable of generating explanations for its recommendations. We used the last month of user transactions as a test set and the penultimate month as a validation set to determine the best model hypeparameters.

To evaluate the ranking quality, we use standard metrics such as Recall@K, NDCG@K, MAP@K. If the list of candidates is smaller than a particular K, we simply pad the end of the list with zeros. Furthermore, we do not calculate ranking metrics for lists of candidates if they lack any examples that can create a valid (X, Y) pair.

6.1. Results

The results of our experiments are provided in Table 3. The rows are sorted by Recall@1. Since it is difficult to explain the recommendation of any merchant Y by the most popular merchant X, **MostPop** clearly performs the poorest at ordering explanations. As **AR**$_{support}$ uses pairs of items, its results are slightly superior to **MostPop**. **Chrono**, **AR**$_{confidence}$, **SimCatMostPop**, **EASE** and **PersonParty** take into account some logical heuristics, including

Table 3
Explanation quality of different methods. N@K/R@K/M@K are NDCG@K/Recall@K/MNAP@K respectively. The best value is **boldfaced**.

	N@1	N@3	N@10	R@1	R@3	R@10	M@1	M@5	M@10
MostPop	0.34	0.54	0.63	0.22	0.65	0.84	0.34	0.46	0.52
Random	0.37	0.53	0.62	0.23	0.63	0.82	0.37	0.45	0.51
$AR_{support}$	0.38	0.55	0.64	0.24	0.65	0.84	0.38	0.47	0.53
Chrono	0.4	0.55	0.64	0.25	0.64	0.83	0.4	0.47	0.53
$AR_{confidence}$	0.39	0.55	0.64	0.25	0.64	0.83	0.39	0.47	0.53
SimCatMostPop	0.4	0.56	0.64	0.26	0.64	0.83	0.4	0.48	0.54
EASE	0.42	0.58	0.66	0.27	0.F66	0.84	0.42	0.5	0.56
PersonPop	0.42	0.58	0.66	0.27	0.67	0.85	0.42	0.5	0.56
$AR_{leverage}$	0.46	0.59	0.66	0.29	0.66	0.84	0.46	0.51	0.56
SimCatIALS	0.5	0.62	0.7	0.3	0.7	0.87	0.5	0.54	0.6
IALS	**0.58**	**0.71**	**0.76**	**0.37**	**0.77**	**0.91**	**0.58**	**0.64**	**0.68**

different item-to-item relations or the notion that customers frequently purchase from particular merchants. Therefore, these approaches produced relatively good results. **AR**$_{leverage}$ provided quality results because it takes into account both X and Y separately in addition to the pair X → Y. However, iALS-based models take the top of the leaderboard. The most accurate ranking was produced by **implicit ALS**. An interesting point to consider is that this method was able to generate rankings that were more accurate than category-based sorting. **SimCatIALS** performed worse than **iALS**, possibly due to the fact that two merchants in the same category can have a different target audience.

It is important to note that the best-performing **iALS** method has a Recall@1 of 0.37. It means that in 63% of cases, this model retrieves a pair (X, Y) that is labeled as incorrect by crowdworkers. On the other hand, this result is about 20% better on the Recall@1 metric than the results of MostPop.

7. Limitations and Future Work

Our research has some limitations that we plan to overcome in future works. Firstly, the number of available merchants can be very large and it can be expensive to label most of the provided item pairs. This problem can be potentially addressed if we can find a way to predict the respondent's answers based on partial data labeling. Secondly, we did not study the use of unlabeled pairs. For instance, the people's opinions can be predicted by factorising the matrix C. Finally, we considered only a small set of approaches for ranking explanation candidates. Approaches with casual explanations [25] is something to consider in future work.

8. Conclusion

In this paper, we studied the validation of explanations for merchant recommendations. We validated the explanation pairs using a crowdsourcing platform. This made it possible for us to attempt a new approach to evaluating the quality of explanations of recommendations in the offline scenario. We also considered 9 different approaches for generating explanations and compared them based on the data gathered from crowdworkers. The results of our experiments have shown that even well-known approaches may generate invalid explanations that are considered illogical by real users. We hope that this method will allow researchers to develop explainable models and test them in an offline scenario based on data gathered from crowdsourcing.

References

[1] N. Ranjbar Kermany, L. Pizzato, T. Min, C. Scott, A. Leontjeva, A multi-stakeholder recommender system for rewards recommendations, in: Proceedings of the 16th ACM Conference on Recommender Systems, RecSys '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 484–487. URL: https://doi.org/10.1145/3523227.3547388. doi:10.1145/3523227.3547388.

[2] W. Neussner, E. Ginina, N. Kryvinska, et al., Novel approaches to increasing customer loyalty: Example of "cashback" in austria, J Fin Mark. 2022; 6 (2): 1-14. 2 J Fin Mark 2022 Volume 6 Issue 2 (2022).

[3] R. Poleshchuk, Increasing bank customers' loyalty through innovative loyalty programs (2022).

[4] X. Chen, A. Reibman, S. Arora, Sequential recommendation model for next purchase prediction, arXiv preprint arXiv:2207.06225 (2022).

[5] S. Kolesnikov, O. Lashinin, M. Pechatov, A. Kosov, Ttrs: Tinkoff transactions recommender system benchmark, arXiv preprint arXiv:2110.05589 (2021).

[6] M. Du, R. Christensen, W. Zhang, F. Li, Pcard: Personalized restaurants recommendation from card payment transaction records, in: The World Wide Web Conference, 2019, pp. 2687–2693.

[7] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, L. Getoor, Personalized explanations for hybrid recommender systems, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, 2019, pp. 379–390.

[8] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: 2008 Eighth IEEE international conference on data mining, Ieee, 2008, pp. 263–272.

[9] S. Rendle, W. Krichene, L. Zhang, Y. Koren, Revisiting the performance of ials on item recommendation benchmarks, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 427–435.

[10] G. Peake, J. Wang, Explanation mining: Post hoc interpretability of latent factor models for recommendation systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2060–2069.

[11] S. Xu, Y. Li, S. Liu, Z. Fu, X. Chen, Y. Zhang, Learning post-hoc causal explanations for recommendation, arXiv preprint arXiv:2006.16977 (2020).

[12] W. Cheng, Y. Shen, L. Huang, Y. Zhu, Incorporating interpretability into latent factor models via fast influence analysis, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 885–893. URL: https://doi.org/10.1145/3292500.3330857. doi:10.1145/3292500.3330857.

[13] V. W. Anelli, A. Bellogín, T. Di Noia, F. M. Donini, V. Paparella, C. Pomo, An analysis of local explanation with lime-rs (2022).

[14] S. Chang, F. M. Harper, L. G. Terveen, Crowd-based personalized natural language explanations for recommendations, in: Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 175–182.

[15] D.-G. Hong, Y.-C. Lee, J. Lee, S.-W. Kim, Crowdstart: Warming up cold-start items using crowdsourcing, Expert Systems with Applications 138 (2019) 112813. URL: https://www.sciencedirect.com/science/article/pii/S0957417419305093. doi:https://doi.org/10.1016/j.eswa.2019.07.030.

[16] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, L. Getoor, Personalized explanations for hybrid recommender systems, in: Proceedings of the 24th International Conference on Intelligent User Interfaces, 2019, pp. 379–390.

[17] A. Ghazimatin, S. Pramanik, R. Saha Roy, G. Weikum, Elixir: learning from user feedback on explanations to improve recommender models, in: Proceedings of the Web Conference 2021, 2021, pp. 3850–3860.

[18] K. Balog, F. Radlinski, Measuring recommendation explanation quality: The conflicting goals of explanations, in: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, 2020, pp. 329–338.

[19] X. Chen, Y. Zhang, J.-R. Wen, Measuring ”why” in recommender systems: a comprehensive survey on the evaluation of explainable recommendation, arXiv preprint arXiv:2202.06466 (2022).

[20] A. Drutsa, V. Farafonova, V. Fedorova, O. Megorskaya, E. Zerminova, O. Zhilinskaya, Practice of efficient data collection via crowdsourcing at large-scale, arXiv preprint arXiv:1912.04444 (2019).

[21] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE international conference on data mining (ICDM), IEEE, 2018, pp. 197–206.

[22] R. He, J. McAuley, Fusing similarity models with markov chains for sparse sequential recommendation, in: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, 2016, pp. 191–200.

[23] H. Steck, Embarrassingly shallow autoencoders for sparse data, arXiv preprint arXiv:1905.03375 (2019).

[24] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, D. Jannach, A troubling analysis of reproducibility and progress in recommender systems research, ACM Transactions on Information Systems (TOIS) 39 (2021) 1–49.

[25] S. Xu, Y. Li, S. Liu, Z. Fu, Y. Ge, X. Chen, Y. Zhang, Learning causal explanations for recommendation, in: The 1st International Workshop on Causality in Search and Recommendation, 2021.