Bridging Offline-Online Evaluation with a Time-dependent and Popularity Bias-free Offline Metric for Recommenders

Petr Kasalický*, Rodrigo Alves and Pavel Kordík

Faculty of Information Technology, Czech Technical University in Prague, Thákurova 9, Prague, 160 00, Czech Republic

Abstract

The evaluation of recommendation systems is a complex task. The offline and online evaluation metrics for recommender systems are ambiguous in their true objectives. The majority of recently published papers benchmark their methods using ill-posed offline evaluation methodology that often fails to predict true online performance. Because of this, the impact that academic research has on the industry is reduced. The aim of our research is to investigate and compare the online performance of offline evaluation metrics. We show that penalizing popular items and considering the time of transactions during the evaluation significantly improves our ability to choose the best recommendation model for a live recommender system. Our results, averaged over five large-size real-world live data procured from recommenders, aim to help the academic community to understand better offline evaluation and optimization criteria that are more relevant for real applications of recommender systems.

Keywords

recall, click-through rate, leave-last-one-out, evaluation, popularity, bias

1. Introduction

The evaluation of recommendation systems is a complex task. One of the primary reasons is that several evaluation metrics represent distinct properties of a recommendation algorithm (RA). For instance, RMSE is a reliable marker of how a RA predicts a user-to-item taste [1], whereas Recall reflects how well the same algorithm retrieves a relevant list of items [2]. Therefore, it is crucial to identify metrics that effectively evaluate and describe the target of a particular recommender system (RS).

A comprehensive search of the relevant literature revealed that the most prevalent method is *offline* evaluation of static feedback, where data are collected from a real-world RS dataset [3, 4]. In offline evaluation, recommendation algorithms are trained on a training subset of data [5], and then their recommendations are evaluated on test data using metrics such as Recall@k [6], Precision@k [7], NDCG@k [8], MAP@k [9] and HR@k [10].

However, conventional offline evaluation of RS has significant disadvantages. For instance,

2nd Edition of EvalRS: a Rounded Evaluation of Recommender Systems, August 6 - August 10, 2023, Long Beach, CA, USA

*Corresponding author.

kasalpe1@fit.cvut.cz (P. Kasalický); rodrigo.alves@fit.cvut.cz (R. Alves); pavel.kordik@fit.cvut.cz (P. Kordík)

0000-0001-6438-366X (P. Kasalický); 0000-0001-7458-5281 (R. Alves); 0000-0003-1433-0089 (P. Kordík)

CEUR Workshop Proceedings (CEUR-WS.org)

the interaction observation follows a non-uniform distribution [11, 12, 13]: certain items have more interaction observations solely because the RS exposed them to more users. In addition to observation bias, standard offline evaluation techniques are agnostic to the fact that recommendation algorithms perform in a live environment where it is only possible to use interactions made in the past to predict future interactions: the commonly employed leave-one-out-cross-validation (LOOCV) method [14, 15] does not track the user's behavior over time.

Several approaches have been proposed to address these drawbacks. Bias in the test data caused by the missing-not-at-random (MNAR) problem can be partially suppressed by sampling [16, 17] or by using popularity-stratified recall [13], which gives a higher reward for recommending less popular items. The unbiased evaluation method for simulating bandit algorithms introduced in [18] was experimentally verified in [19] and found to work only for Top-1 recommendations.

Regarding time-aware validation processes, [20] proposed leave-last-one-out-cross-validation (LLOOCV), where the validation set only considers the latest iteration of each user. Although there is a clear past-future distinction for each user, in LLOOCV there is no such distinction between users. [21] offered a more comprehensive solution called k-fold Sliding Window Evaluation (SW-EVAL), which analyzes the recommendation algorithm based only on interactions that occurred after the conclusion of the training data.

Unfortunately, despite the significant progress made by the RS community to debiasing offline evaluation, the techniques often do not adequately reflect the nature of *online* recommender systems. An optimal evaluation technique should review the entire system and consider the platform's high-level objectives, such as the number of clicks on items, the number of items purchased, the number of adverts viewed, the customer lifetime value, etc. [22]. Such metrics are unsuitable for the offline setting because a RS's environment is domain-dependent, dynamic, and constantly changing. Furthermore, the results are frequently unreproducible because they are based on the monitoring of implicit interactions of a real user or asking them to use pop-up surveys for explicit ratings of the recommended items [23].

This paper fulfills the research objectives outlined for the EvalRS 2023 workshop [24] by experimentally exploring the relationship between offline and online metrics. We note that this comparison type is crucial for real-world recommenders' commercial success. In real scenarios, it is possible to compare a large number of models by using offline metrics. However, to avoid compromising the system's accuracy, only a limited number of models can be examined online. Our first step is to investigate which offline evaluation criteria enables to navigate towards architectures and parametrizations with better online performance. More specifically, we are interested in verifying whether a model with the highest Recall@k (when evaluated offline) also produces the highest CTR (when evaluated online). We also verified the impact of including popularity-penalization and time aspect on the CTR-Recall@k relation. Finally, we introduce a new offline evaluation metric more adaptive to live environments. Our metric, so-called $recall@K^{\beta}_{LLOO}$, simultaneously incorporates popularity penalization and time dependency of interactions.

2. Related work

Due to the complexity of performing online evaluations, comparative studies between online and offline metrics are scarce. Added to that, existing solutions typically analyze a single dataset. For instance, in [25] the authors compare offline and online metrics on the Swiss news website *swissinfo.ch*. They constructed a metric titled Success@k and used it to fit the model. The value of Success@k in the (offline) validation set is then compared to the CTR in the online environment. As a result, they hypothesize that RAs that are dominant based on offline metrics are no longer as effective in the online context since they favor popular items. In contrast, the RA that recommends random items does significantly better in the online evaluation than in the offline evaluation because they encourage content exploration. As a follow-up, in [26] they proposed a model predicting online performance based on five different offline metrics but did not find a universal formula for predicting online performance. A similar mismatch between offline and online evaluation is shown for individual domains. For example, in [27], the authors used the MovieLens dataset and evaluated the online performance of RA based on feedback from 100 users invited to their experiment. The behavior of Docear users receiving recommendations of research articles was investigated in [28, 19] with the conclusion that offline evaluations are probably not suitable to evaluate recommender systems in this domain. An experiment involving 4287 users of a travel agency was conducted in [29]. However, according to the findings reported in [30], a general cross-domain comparison is missing. A study from Netflix [31] from 2021 describes the same issue regarding deep-learning models. They identify a ”mismatch in offline and online settings” as one of the unresolved practical challenges of current recommendation systems. The authors of the study give a hint that Netflix uses its own offline bias-suppressing metric for more corresponding offline evaluation. However, they failed to describe essential details, such as how to use contextual bandit techniques to remove various biases. The researchers in [32] addressed the challenge of offline evaluation in industry recommender systems by investigating behavioral principles and developing RecList, an open-source tool that employs NLP techniques to assess the effectiveness of a RA in common recommendation scenarios. All of the previously referenced research compare offline to online metrics without taking popularity bias and time-dependence into account. Differently, we investigate how a correction for popularity bias [13] and a validation set consisting purely of interactions that happened after the end of the training data [21] affect the optimization of online metrics. We also analyzed multiple datasets.

2.1. Contributions

The main contributions of this paper are listed as follows:

- We introduced a new offline evaluation criterion that included both (1) popularity penalization and (2) time aspect of interactions ($recall@K^{\beta}_{LLOO}$, see Section 3.1).
- We conducted a large-scale experiment on real-world datasets that shows our criterion has an advantage in predicting online performance over the popular criteria like recall@N used in the RecSys community.

3. Method

In our experiment, we are examining whether (1) incorporating a time dimension into an offline evaluation approach, and (2) reducing popularity bias by assigning less weight to errors of frequently interacted items, reduces the disparity between offline and online metrics. In theory, one expects that offline metrics (when employed to cross-validate models) result in improved online performance in live environments.

Basic notation: Denote the set of interactions between users and items by $F = \{f_1, \dots, f_p\}$. A single interaction is denoted as $f_j \in (U \times I \times \mathbb{Z}_t)$ for $\forall j \in \{1, \dots, p\}$, where $U = \{u_1, \dots, u_m\}$ is a set of all RS users, $I = \{i_1, \dots, i_n\}$ is a set of items which can be recommended, $\mathbb{Z}_t$ is the set of integers expressing the timestamp when the interaction was performed and p is a total number of interactions. Then we represent the set of interactions between the item i and the user u as

$$F_{u,i} = \{(u_j, i_j, t_j) \mid (u_j, i_j, t_j) \in F \,:\, u_j = u \wedge i_j = i\}.$$

Finally, define the set of relevant items for the user u as $N_u \subset I$. We assume that items that are relevant for a given user can be extracted from explicit and/or implicit feedback. Thus, we considered the item i relevant to the user u if u implicitly (e.g., view, click) interacted with i.

3.1. Offline metrics

We will first present offline metrics, which will then be compared to online measurements. Regarding the cross-validating split procedure, we are considering LOOCV and LLOOCV. The most common way to compute recall measured using LOOCV is expressed by:

$$recall@K_{LOO} = \frac{\sum\limits_{u \in U} \sum\limits_{i \in N_u} \mid \{i\} \cap Top_K(N_u \setminus \{i\}) \mid}{\sum\limits_{u \in U} \mid N_u \mid}, \tag{1}$$

where $Top_K(M)$ is a recommendation algorithm that selects K items based on items in the set $M \subset I$ recommendable items. Conversely, recall measured using LLOOCV can be computed by

$$recall@K_{LLOO} = \frac{\sum\limits_{u \in U} \sum\limits_{(i_1, t_1) \in F_u} \mid \{i_1\} \cap Top_K(Q_{t_1}) \mid}{\sum\limits_{u \in U} \mid N_u \mid}, \tag{2}$$

with

$$Q_{t_1} = \{i_2 \mid (i_2, t_2) \in F_u \,:\, t_2 < t_1\}$$

where F_u is a list of interactions of user u defined as $F_u = \{(i_j, t_j) \mid (u_j, i_j, t_j) \in F \,:\, u_j = u\}$ and the expression $\{i_2 \mid (i_2, t_2) \in F_u \,:\, t_2 < t_1\}$ represents the set of items that were interacted by the user u before timestamp t_1.

Including popularity-penalization from [13] and incorporating it into Eq. (1), we get

$$recall@K^{\beta}_{LOO} = \sum_{u \in U} w^{\beta}(u) \frac{\sum\limits_{i \in N_u} \mid \{i\} \cap Top_K(N_u \setminus \{i\}) \mid \ p(i)^{-\beta}}{\sum\limits_{i \in N_u} p(i)^{-\beta}} \tag{3}$$

and Eq. (2) turns to

$$recall@K^{\beta}_{LLOO} = \sum_{u \in U} w^{\beta}(u) \frac{\sum\limits_{(i_1,t_1)\in F_u} \mid \{i_1\} \cap Top_K(Q_{t_1}) \mid \;\; p(i)^{-\beta}}{\sum\limits_{i \in N_u} p(i)^{-\beta}} \tag{4}$$

where $w^{\beta}(u) \in [0, 1]$ is a weight of user u. The sum of weights for all users must sum up to one, $\sum\limits_{u \in U} w^{\beta} = 1$, with suggested:

$$w^{\beta}(u) = \frac{1}{\mid U \mid} \frac{\sum\limits_{i \in N_u} p(i)^{-\beta}}{\sum\limits_{v \in U} \sum\limits_{i \in N_v} p(i)^{-\beta}}$$

and $p(i) \in [0, 1]$ denotes relative popularity of item i. Note that for $\beta = 0$, Eq. (3) and (1) are equivalent, and Eq. (4) with (2) as well.

Remark: In [13], the authors explain the effect of the hyperparameter β for offline evaluation but no longer show whether penalizing popularity makes recall a more appropriate metric for the online environment. [21] proposed that reducing bias by penalization of popularity β can lead to a better offline evaluation, but with no experimental verification.

3.2. Online evaluation

Another approach to evaluating a RA is based on feedback from users interacting with a live recommender system. More specifically, we work on a scenario where items are recommended to a user, and the RS collects the user's reactions to the recommended items. An example of a reaction is when the user clicks on the recommended item.

The most widely used online metric (given its universality) is the click-through rate (CTR). CTR can be seen as the ratio between the accepted recommendations and all offered recommendations [28]. We consider that the user accepted the recommendation if he clicked on at least one of the recommended items.

Remark: A recommender system can observe implicit and explicit CTRs. An explicit CTR is calculated if the RS has clear evidence that a user clicked on a particular item as a result of the recommendation. The implicit CTR can be calculated based on interactions if the RS does not have explicit knowledge that the item has been clicked through as a result of the recommendation.

Formally, $C : Z_t \times U \times \{0, 1\}^I$ defines a set of recommendations where each recommendation is represented by a timestamp $t \in Z_t$, user $u \in U$ and by a set of recommended items $I' \subset I$. When assuming that the set of interactions F contains only "clicked-type" interactions, then the implicit CTR can be calculated as:

$$iCTR(d) = \frac{\sum\limits_{(t,u,I')\in C} sgn(\mid I' \cap F_u(t, d) \mid)}{\mid C \mid}, \tag{5}$$

where $F_u(t, d)$ is a set of items interacted by the user u within time $[t, t + d]$ defined as

$$F_u(t, d) = \{i_j \mid (i_j, t_j) \in F_u \; : \; (t_j >= t) \wedge (t + d >= t_j)\},$$

and d is a parameter determining how long after the recommendation the user has to interact with the recommended item to mark the recommendation as successful.

3.3. Methodology of the experiment

The goal of our experiment is to find out how different versions of $recall@K^{\beta}_{LOO}$ and $recall@K^{\beta}_{LLOO}$ as validation metric relates to CTR.

Due to the limited resources to conduct online experiments, to perform our analysis, we first select a backbone algorithm: the item-*k*NN algorithm [33] with the similarity between items measured by the cosine similarities of latent space embeddings. The latent space embeddings are created using matrix factorization from implicit feedback [34] with different data preprocessing and hyperparameters (such as latent space size and regularization) to ensure diverse performance according to offline metrics. Second, we train our model and measure its performance using different versions of recall (i.e., $recall@K^{\beta}_{LOO}$ and $recall@K^{\beta}_{LLOO}$ with β as a hyperparameter). Third, our model is deployed to an RS with live interactions, and thus the CTR can be measured. Note that the RS is constantly receiving new interactions from users. Therefore, we re-trained the model to include them periodically.

Once the individual steps of the experiment are fulfilled and implemented, the experiment is performed on several datasets. The results of the experiment are measured CTRs along with the number of users that interacted with each model. The RS splits users for each model equally. Once the user was assigned to the model during the first recommendation, the same model generated any further recommendations.

Subsequently, the vector of CTRs can be taken for the dataset A and its $L = 5$ deployed models $S^A = (S^A_1, S^A_2, \dots, S^A_5)$. Similarly, one $E^A = (E^A_1, E^A_2, \dots, E^A_5)$ vector is measured for each combination of hyperparameters VAL, β, k. The entire list of vectors of recalls is measured according to their hyperparameters $EL^A = \{E^A_{VAL_1,\beta_1,k_1}, E^A_{VAL_2,\beta_2,k_2}, \dots, E^A_{VAL_V,\beta_V,k_V}\}$, where V is the total number of combinations of hyperparameters and VAL_j, β_j, k_j are the individual hyperparameter values.

Inspired by practical application, we are interested in whether the best model according to offline metrics for a given dataset is also the best according to online metrics. In other words, we find out what the chances are that if we choose the best model according to recall, it will be the best model according to CTR. This can be measured using Recall@1. To distinguish between $recall@K^{\beta}_{LLOO}$ measuring the quality of item recommendations and Recall@1 measuring whether the best model according to offline metrics is also the best model in online evaluation, we will explicitly denote the latter as **Model Selection Recall (MSR)**. We define **MSR** as the ratio of how many times the best model according to offline metrics has been selected by offline metrics, namely recalls with different hyperparameters (β, VAL).

4. Experiments

In this section, we will describe the used datasets and present the results of our large-scale experiment.

Table 1
Description of the total number of items and users in the datasets along with the average number of users and interactions per model used to measure iCTR

Dataset	Description	Interactions	Users	Items	Users per model	Interactions per model
A	Liquor e-shop	2.8m	1.3m	3k	7.0k	22.7k
B	Pet store	16.8m	6.9m	20k	68.8k	146.3k
C	Fashion e-shop	30.4m	3.1m	19.3k	6.3k	19.1k
D	Supplier of African goods	13.9m	70k	1k	19.4k	67.0k
E	Videostreaming service	30.7m	1.8m	5.3k	19.6k	69.7k

4.1. Datasets and collected data

Commonly stable research datasets (such as *MovieLens* or *Last.fm*) cannot be used in our experiment since live users are required for online evaluation. Because of that, our work was performed by using real datasets with live customers. For each model and each dataset, iCTR with a parameter $d = 10$ (minutes) was measured over 18 days. The number of users and the number of their recommendations participating in iCTR measurement depend on the data set and other external circumstances and are shown in Table 1.

The datasets have been selected to include different domains. In addition to e-commerce services, a video streaming platform is represented. Also, dataset D is very different from other e-commerce customers as it is a B2B business with a few products and customers with an unconventional high number of interactions. Another parameter by which the datasets were selected is the number of recommendations per day. The datasets with small traffic could not be selected since the traffic needs to be divided between models and an estimate of CTR needs to be as accurate as possible.

4.2. Used hyperparameters and resulting MSR

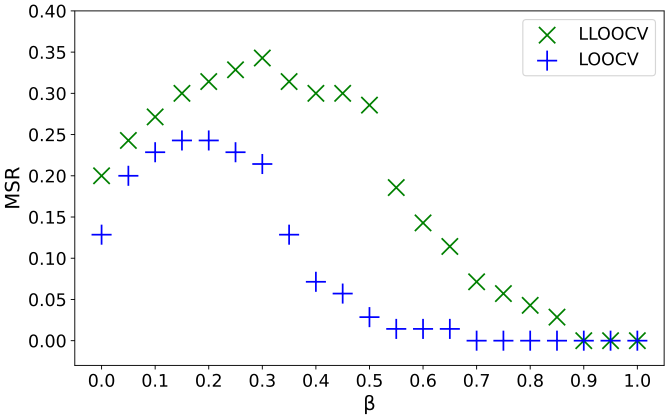
The values of hyperparameters for which recall was measured were $k \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 50\}$ and $\beta \in \{0.05j \mid j \in \{0, 1, \ldots, 20\}\}$. The *VAL* hyperparameter specifies whether is used Equation (3) or (4) for the cross-validation method. Figure 1 shows the relation between iCTR and recall for selected βs and both values of *VAL* hyperparameter. It can be seen that the inclusion of the time aspect using the LLOCV technique led to an increase in MSR, which means the best online model was more often selected using the offline metric. The same effect can be seen when using the penalization of popularity by the beta hyperparameter. However, it is not true that the more popularity is penalized, the better the offline metric is.

The best metric found with LLOOCV and $\beta = 0.30$ has MSR$= 34.29\%$ and therefore, it is better in choosing the best online model than commonly used LOOCV technique with $\beta = 0$ (i.e., without popularity penalization) with MSR$= 12.86\%$.

5. Conclusion

We researched various solutions to better address the known issues of offline metrics with the aim of reducing the gap between offline and online metrics. Then we performed a large-scale

Figure 1: Including a time aspect in the evaluation using the LLOCV approach increases the chance that the selected model will be the best online model (measured by MSR)



experiment that examined the effect of penalizing popularity and including the time aspect of recommendations on the relation between recall and CTR. The experiment was conducted on five commercial datasets covering multiple domains (such as e-commerce, video streaming). It was shown that, in general, maximizing recall does not always lead to maximizing CTR. Popularity-stratified recall proved to be a good approach for reducing the MNAR problem leading to better model selection for an online environment but measured using offline metrics. Offline evaluation with an emphasis on the time aspect of recommendations using leave-last-one-out cross-validation also showed that it can lead to an improvement in MSR. Using the proposed method, we were able to improve the selection of the best online model from the original MSR= 12.86% to MSR= 34.29%.

5.1. Future work

The relation between recall and CTR was investigated on five datasets and each contained five different models. To measure the relation more accurately, it would be advisable to have more models, but this greatly prolongs the experiment (since there is a limit traffic of new users) and increases the already exhaustive engineering effort. Similarly, it would be interesting to include datasets from other domains, such as news or bazaars, as these are very specific to the changing popularity of the item. As a final improvement, the SW-EVAL method could be used for stricter adherence to the time aspect. Another extension advisable for the future is to examine how the best possible offline metric relates to dataset attributes such as number of users, number of items, sparsity of the interaction matrix, and domain of the dataset. This requires a large meta-study based on experiments with dozens of real datasets for a long period of time.

Acknowledgments

This work was supported by the Grant Agency of the Czech Technical University in Prague (SGS23/210/OHK3/3T/18). The experiment was carried out in cooperation with Recombee.

References

[1] A. Ledent, R. Alves, M. Kloft, Orthogonal Inductive Matrix Completion, IEEE Transactions on Neural Networks and Learning Systems (2021) 1–12. doi:10.1109/TNNLS.2021.3106155, arXiv:2004.01653 [cs, stat].

[2] T. Silveira, M. Zhang, X. Lin, Y. Liu, S. Ma, How good your recommender system is? A survey on evaluations in recommendation, International Journal of Machine Learning and Cybernetics 10 (2019) 813–831. doi:10.1007/s13042-017-0762-9.

[3] J. Ni, J. Li, J. McAuley, Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 188–197. URL: https://www.aclweb.org/anthology/D19-1018. doi:10.18653/v1/D19-1018.

[4] D. Wannigamage, M. Barlow, E. Lakshika, K. Kasmarik, Steam Games Dataset : Player count history, Price history and data about games, volume 1, 2020. URL: https://data.mendeley.com/datasets/ycy3sy3vj2/1. doi:10.17632/ycy3sy3vj2.1, publisher: Mendeley Data.

[5] A. Gunawardana, G. Shani, S. Yogev, Evaluating Recommender Systems, Springer US, New York, NY, 2022, pp. 547–601. URL: https://doi.org/10.1007/978-1-0716-2197-4_15. doi:10.1007/978-1-0716-2197-4_15.

[6] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, Z. Wang, Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems, 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, pp. 968–977. doi:10.1145/3292500.3330836.

[7] N. Sachdeva, G. Manco, E. Ritacco, V. Pudi, Sequential Variational Autoencoders for Collaborative Filtering, Technical Report arXiv:1811.09975, arXiv, 2018. ArXiv:1811.09975 [cs, stat] type: article.

[8] H. Steck, Embarrassingly Shallow Autoencoders for Sparse Data, in: The World Wide Web Conference, WWW ’19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 3251–3257. doi:10.1145/3308558.3313710.

[9] C. Yang, L. Bai, C. Zhang, Q. Yuan, J. Han, Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1245–1254. URL: https://doi.org/10.1145/3097983.3098094. doi:10.1145/3097983.3098094.

[10] Y. Tay, L. Anh Tuan, S. C. Hui, Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking, in: Proceedings of the 2018 World Wide Web Conference, WWW ’18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, pp. 729–739. doi:10.1145/3178876.3186154.

[11] B. M. Marlin, R. S. Zemel, S. Roweis, M. Slaney, Collaborative filtering and the missing at random assumption, in: Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, UAI’07, AUAI Press, Arlington, Virginia, USA, 2007, pp. 267–275.

[12] B. M. Marlin, R. S. Zemel, Collaborative prediction and ranking with non-random missing data, in: Proceedings of the third ACM conference on Recommender systems, RecSys '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 5–12. doi:10.1145/1639714.1639717.

[13] H. Steck, Item popularity and recommendation accuracy, in: Proceedings of the fifth ACM conference on Recommender systems, RecSys '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 125–132. doi:10.1145/2043932.2043957.

[14] X. Li, J. She, Collaborative Variational Autoencoder for Recommender Systems, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Halifax NS Canada, 2017, pp. 305–314. doi:10.1145/3097983.3098077.

[15] J. Vinagre, A. M. Jorge, J. Gama, Evaluation of recommender systems in streaming environments, Silicon Valley, United States, 2014. doi:10.13140/2.1.4381.5367, arXiv:1504.08175 [cs].

[16] D. Carraro, D. Bridge, A sampling approach to Debiasing the offline evaluation of recommender systems, Journal of Intelligent Information Systems 58 (2022) 311–336. URL: https://link.springer.com/10.1007/s10844-021-00651-y. doi:10.1007/s10844-021-00651-y.

[17] D. Carraro, D. Bridge, Debiased offline evaluation of recommender systems: a weighted-sampling approach, in: Proceedings of the 35th Annual ACM Symposium on Applied Computing, SAC '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1435–1442. URL: https://doi.org/10.1145/3341105.3375759. doi:10.1145/3341105.3375759.

[18] L. Li, W. Chu, J. Langford, X. Wang, Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 297–306. URL: https://doi.org/10.1145/1935826.1935878. doi:10.1145/1935826.1935878.

[19] J. Beel, S. Langer, A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems, in: S. Kapidakis, C. Mazurek, M. Werla (Eds.), Research and Advanced Technology for Digital Libraries, Springer International Publishing, Cham, 2015, pp. 153–168.

[20] Q. Zhao, J. Chen, M. Chen, S. Jain, A. Beutel, F. Belletti, E. H. Chi, Categorical-attributes-based item classification for recommender systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 320–328. doi:10.1145/3240323.3240367.

[21] O. Jeunen, K. Verstrepen, B. Goethals, Fair Offline Evaluation Methodologies for Implicit-Feedback Recommender Systems with MNAR Data, Vancouver, Canada, 2018, p. 8. URL: http://adrem.uantwerpen.be/bibrem/pubs/OfflineEvalJeunen2018.pdf.

[22] T. Řehořek, Manipulating the Capacity of Recommendation Models in Recall-Coverage Optimization, Ph.D. thesis, Czech Technical University, 2019. URL: https://dspace.cvut.cz/handle/10467/81823, accepted: 2019-04-05T11:19:10Z Publisher: České vysoké učení technické v Praze. Vypočetní a informační centrum.

[23] R. Cañamares, P. Castells, A. Moffat, Offline evaluation options for recommender systems, Information Retrieval Journal 23 (2020) 387–410. doi:10.1007/s10791-020-09371-3.

[24] F. Bianchi, P. J. Chia, C. Greco, C. Pomo, G. Moreira, D. Eynard, F. Husain, J. Tagliabue, Evalrs 2023. well-rounded recommender systems for real-world deployments, 2023. arXiv:2304.07145.

[25] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, A. Huber, Offline and online evaluation of news recommender systems at swissinfo.ch, in: Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14, ACM Press, Foster City, Silicon Valley, California, USA, 2014, pp. 169–176. doi:10.1145/2645710.2645745.

[26] A. Maksai, F. Garcin, B. Faltings, Predicting online performance of news recommender systems through richer evaluation metrics, in: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 179–186. URL: https://doi.org/10.1145/2792838.2800184. doi:10.1145/2792838.2800184.

[27] M. Rossetti, F. Stella, M. Zanker, Contrasting Offline and Online Results when Evaluating Recommendation Algorithms, in: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 31–34. URL: https://doi.org/10.1145/2959100.2959176. doi:10.1145/2959100.2959176.

[28] J. Beel, M. Genzmehr, S. Langer, A. Nürnberger, B. Gipp, A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation, in: Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, RepSys '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 7–14. doi:10.1145/2532508.2532511.

[29] L. Peska, P. Vojtas, Off-line vs. On-line Evaluation of Recommender Systems in Small E-commerce, Proceedings of the 31st ACM Conference on Hypertext and Social Media (2020) 291–300. doi:10.1145/3372923.3404781, arXiv: 1809.03186.

[30] O. Jeunen, Revisiting offline evaluation for implicit-feedback recommender systems, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 596–600. doi:10.1145/3298689.3347069.

[31] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, J. Basilico, Deep Learning for Recommender Systems: A Netflix Case Study, AI Magazine 42 (2021) 7–18. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/18140. doi:10.1609/aimag.v42i3.18140, number: 3.

[32] P. J. Chia, J. Tagliabue, F. Bianchi, C. He, B. Ko, Beyond ndcg: Behavioral testing of recommender systems with reclist, WWW '22 Companion, Association for Computing Machinery, New York, NY, USA, 2022, p. 99–104. URL: https://doi.org/10.1145/3487553.3524215. doi:10.1145/3487553.3524215.

[33] A. N. Nikolakopoulos, X. Ning, C. Desrosiers, G. Karypis, Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer US, New York, NY, 2022, pp. 39–89. URL: https://doi.org/10.1007/978-1-0716-2197-4_2. doi:10.1007/978-1-0716-2197-4_2.

[34] Y. Hu, Y. Koren, C. Volinsky, Collaborative Filtering for Implicit Feedback Datasets, 2008, pp. 263–272. doi:10.1109/ICDM.2008.22.