Revisiting Offline Evaluation for Implicit-Feedback Recommender Systems

Olivier Jeunen
University of Antwerp, Belgium

ABSTRACT

Recommender systems are typically evaluated in an offline setting. A subset of the available user-item interactions is sampled to serve as test set, and some model trained on the remaining data points is then evaluated on its performance to predict which interactions were left out. Alternatively, in an online evaluation setting, multiple versions of the system are deployed and various metrics for those systems are recorded. Systems that score better on these metrics, are then typically preferred. Online evaluation is effective, but inefficient for a number of reasons. Offline evaluation is much more efficient, but current methodologies often fail to accurately predict online performance. In this work, we identify three ways to improve and extend current work on offline evaluation methodologies. More specifically, we believe there is much room for improvement in temporal evaluation, off-policy evaluation, and moving beyond using just clicks to evaluate performance.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Evaluation of retrieval results*;

KEYWORDS

Offline evaluation; counterfactual evaluation; implicit feedback

ACM Reference format:
Olivier Jeunen. 2019. Revisiting Offline Evaluation for Implicit-Feedback Recommender Systems. In *Proceedings of Thirteenth ACM Conference on Recommender Systems, Copenhagen, Denmark, September 16–20, 2019 (RecSys '19),* 5 pages.
https://doi.org/10.1145/3298689.3347069

1 INTRODUCTION

Traditionally, recommender systems research focused on *rating* prediction from *explicit* feedback. The best known example of this setting is probably the Netflix Prize competition, where researchers were challenged to predict which ratings users had given to certain movies, based on millions of other user-item-rating triplets [3]. User-item pairs that were predicted to have high ratings, were then assumed to make up good recommendations. In recent years, a shift has occurred towards *item* prediction from *implicit* feedback [15, 29, 31, 38]. These systems no longer rely on a set of explicitly generated ratings, but can learn to infer personal preferences from logged feedback such as click behaviour on a news website, listening behaviour on a music streaming service, video watches on video streaming websites, and many more. As logged feedback is much easier to obtain than explicit ratings, these systems are gaining more and more popularity. Netflix has even moved on from their star rating system, favouring binary preference expressions [11].

Implicit-feedback recommender systems can be evaluated either off- or on-line. In the offline setting, the data is split into a training and testing subset, as is often the case in classical supervised learning contexts. Models learn from the training set, and are evaluated on their ability to predict the samples that are part of the test set. Those models that perform best on some chosen metric, are then assumed to be the optimal performers in an online setting as well. Online evaluation methods often relate to some form of A/B-testing: multiple different models are deployed, and their performance is measured according to some Key Performance Indicator (KPI), such as click-through rate (CTR), sales revenue, dwell time, retention rate, and so on. The biggest advantage of online evaluation methods is that they are very effective: interaction between users and the systems is directly measured, and if done properly, online experiments provide a fair and unambiguous view of system performance. They are, however, much more expensive than offline alternatives for a number of reasons [1, 10, 33]. Because of this, offline evaluation methods that can accurately predict online performance remain imperative. However, multiple recent works show time and again that offline evaluation results from traditional procedures are often contradictory compared to the results of live A/B-tests [2, 9, 32]. The scientific aspirations of this research are to identify which aspects are at the root cause of the fact that current offline evaluation procedures are often ineffective, and alleviate these aspects during the training as well as the evaluation phase of live recommender system deployment. Specifically, we wish to research novel offline evaluation procedures that are much more tightly coupled with the inherent characteristics of these live recommender systems, such as dynamic deployment, user interaction, temporal information, and self-induced presentation bias in the data. The rest of this work broadly corresponds to the following research questions:

(1) What is the importance and the role of temporal information in an offline evaluation stage? How do we handle the sequential order of events and the absolute time between interactions and predictions correctly?
(2) What impact does a live recommender system, in place during data collection, have on the resulting logged feedback? How do we mitigate biases they induce?
(3) Can we use more information about user (in-)activity, beyond just clicks, during the offline evaluation process? How do we exploit information about impressions, dwell time, scrolling,... to effectively distinguish between *missing* and *negative* feedback?

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09.
https://doi.org/10.1145/3298689.3347069

2 TEMPORAL EVALUATION

In its simplest form, supervised learning systems are evaluated on a hold-out test set. A subset of the available data is randomly sampled and held out when training a model. Then, the model is evaluated on its performance when predicting the labels for the unseen, held-out test set. k-fold cross validation and bootstrap sampling are the most well known and widely used methods [20]. As the recommender systems field emerged from the broader machine learning field, adaptations specific to the recommender systems use case were proposed. Leave-one-out cross-validation (LOOCV) is a commonly recurring scheme in the literature, where one item for every user is randomly sampled to be part of the test set. The training set then consists of all remaining user-item pairs. Every model generates a set of top-N recommendations, and those that can rank the missing sampled items highest in the set of recommendations are assumed to be the best performers in an online environment as well. This process is repeated with different random seeds and samples, and results are averaged in an attempt to reduce variance from different runs. While this technique has been used widely and recently to present new models as the state of the art [7, 8, 13, 24–27, 41, 43], it entirely disregards the sequential nature of user-item interactions. It should come as no surprise that positively rewarding the prediction of past interactions from future data leads to a distorted picture of algorithmic performance in an online environment. Because of this, temporal evaluation has recently gained traction as well [17, 18, 39]. Zhao et al. [44] use a temporal leave-one-out scheme that we will refer to as last-one-out: instead of randomly sampling an item for every user, the last item is left out for every user. By doing this, no information about future preferences of a user can be used by the model when generating recommendations for said user, avoiding look-ahead bias. However, as the model is still trained on future interactions from other users, time-constraints remain violated and biases remain inevitable. Li et al. [23] propose an evaluation protocol called *replay*, aimed towards contextual-bandit news recommenders. Their work is extended into StreamingRec, a recently introduced offline evaluation framework for news recommenders [18]. They focus on model recency and incremental updates, which are of vital importance in the news domain. Nevertheless, incremental learning is not trivial for many state-of-the-art algorithms, nor is it always necessary. Further work for broader recommendation domains still needs to be conducted.

A novel temporal evaluation technique was proposed by Jeunen et al. [17], using a sliding window technique to adhere to the chronological ordering of interactions in the data, and aggregating multiple measurements to provide a robust estimate (Sliding-Window Evaluation, or SW-EVAL). The authors show that taking the sequentiality of the data into account at evaluation time has a significant impact on evaluation results, in terms of (1) absolute values of evaluation metrics, (2) ratios of evaluation metric values among competing algorithms, and (3) rankings of evaluation metric values among competing algorithms. Figure 1 provides some visual intuition into the differences of these methods. Here, $u, v \in \mathcal{U}$ are users and $a, b, c, d, e \in \mathcal{I}$ are items they have interacted with. The x-axis represents time. In this trivial example, the test windows used by SW-EVAL hold only one interaction per user. However, this parameter t_Δ can be tuned freely. We find that SW-EVAL has room

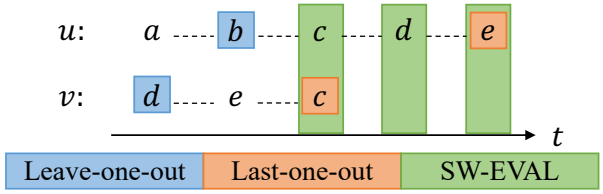

Figure 1: A visual representation of the differences between several offline evaluation procedures. $u, v \in \mathcal{U}$ are users, $a, b, c, d, e \in \mathcal{I}$ are items they have interacted with, and the x-axis represents time.

for several extensions which could prove beneficial as future work. Currently, all user-item interactions that occur within the test window are taken into account with equal importance, and those that occur later are entirely ignored. Because of this, the impact of the window size t_Δ should be further investigated. Furthermore, a discounted importance function could be used to place higher weight on interactions that occur near the prediction time (i.e. the start of the test window), and lower weight on interactions that occur later in time. Additionally, until now, SW-EVAL has only been evaluated on a single dataset, with a modest selection of traditional baseline algorithms and the Recall@k metric. More thorough analysis needs to be done, by exploring the impact temporal evaluation has on clearly time-dependent use-cases (e.g. news recommendation) versus those that might seem less time-dependent on first look (e.g. movie recommendation). We wish to investigate the impact on various types of metrics as well, such as Mean Reciprocal Rank (MRR) or Normalised Discounted Cumulative Gain (NDCG). Finally, to validate the effectiveness of the evaluation procedure, a correlation study comparing results with those obtained from live A/B-tests needs to be conducted as well.

3 DEBIASING LOGGED FEEDBACK

Most recommender systems datasets are collected from logged user-item interactions on some website: be it e-commerce, news, media, and so on. Moreover, most of these websites have some form of live recommender system running that influences user behaviour. As user behaviour is logged *after* these users have been exposed to algorithmic recommendations, this generates severe biases in the resulting datasets [5, 36]. When these datasets are subsequently used to train models that are in turn deployed on the website, vicious feedback loops can occur [34]. The biases present in these datasets pose a significant challenge when the data is used to evaluate other competing algorithms in an offline manner. Under the presence of A/B tests, where multiple algorithms influence behaviour for different subsets of the data [1], or for fast-changing environments that suffer from concept drift [16], these issues become even more poignant. This problem of predicting how a certain algorithm would have performed when we only have data that originates from a different algorithm, is a specific instance of the more general problem of counterfactual evaluation in the reinforcement learning literature. Importance weighting or inverse propensity scoring (IPS) is a well-known statistical technique, often used in these types of situations [4]. In this setting, a stochastic logging policy π_0 is assumed, that assigns a probability to an action, given

some context: $\pi_0(i|x)$. Here, the action corresponds to recommending an item $i \in \mathcal{I}$, and the context is a feature vector that can be of arbitrary dimensionality d: i.e. $x \in \mathbb{R}^d$. When evaluating a new target policy π_t on data collected under π_0, the weights for samples (i, x) are then given by

$$w(i|x) = \frac{\pi_t(i|x)}{\pi_0(i|x)}. \quad (1)$$

It can be proven that, under mild assumptions, this weighting function results in an unbiased estimator. Its variance, however, is essentially unbounded. When π_0 and π_t differ greatly, the ratio of their assigned probabilities will allow for some samples to be weighted disproportionately. Multiple techniques have been proposed to alleviate these issues, such as normalising or capping the weights [10].

However, for the case where π_0 or π_t are deterministic policies (i.e. $\pi(i|x) \in \{0, 1\}$), these weighting techniques lose all practical value. Moreover, even when the policies are truly stochastic, the propensity scores for every recommendation-context pair (i, x) need to be known beforehand. Although this is somewhat trivial for the target policy, this is certainly not the case for the logging policy[1]. Yang et al. [42] tackle this issue by modelling popularity bias as an exponential function to mimic the typical long-tail distribution. They then use this model to estimate propensity scores, and use those in turn to debias their evaluation procedure. The model is user- and context-independent, and assumes a single propensity score for every item i. Furthermore, they assume the probability of a user interacting with an item, is independent from whether that item has actually been recommended to and impressed upon the given user. Naturally, these assumptions do not hold in real-world situations. We identify two possible directions of future research:

(1) By clustering users based on either meta-information (location, age-group, et cetera) or their historical sequence of logged items, the propensity of user-item pairs could more accurately be modelled in a cluster-local model. These models could then, stand-alone or in combination with a global model following the paradigm of [8], be used to improve upon their offline evaluation procedure, in combination with the SW-EVAL procedure.
(2) The independence assumption between impressions and interactions is clearly oversimplified for real-world environments and data. For datasets that include impression information, this effect can be studied. When properly quantified, it can be used to improve upon the accuracy of the estimated propensities. Examples of such datasets include the Outbrain [28] and Plista [19] datasets. We discuss the benefit of using datasets enriched with this information in Section 4.1.

We believe more realistic propensity scores can be estimated through more nuanced modelling, which can in turn help in improving the effectiveness of IPS-based estimators in settings where propensities are unknown beforehand. In the case where not a single item, but a list of N items is recommended, Chen et al. [6] propose an adaptation of the IPS estimator. As propensity scores for the logging policy are also unknown beforehand in their setting, they propose to learn these alongside with the target policy via a recurrent neural network.

Preliminary results from SW-EVAL on logged feedback originating from different logging policies confirm that such biases are indeed present, and significantly impact offline evaluation results when not taken into account properly [17]. As future work, we wish to incorporate such a weighting procedure into the evaluation method, even when logging policy propensities are unknown in advance. Recent work used an adapted IPS-estimator to evaluate an algorithm for music playlist recommendation in an offline way [12]. They show that their offline evaluation results correlate with their online results, obtained through a series of live A/B tests. Multi-armed bandit models were used, with logged propensities. However, their work deals with an idealised environment, making it unclear whether the methodology can be readily applied other domains. Extending their work for generality, to include larger sets of available items, and towards ranking-based metrics is a promising direction for future research.

[1] For the problem of computational advertising, a dataset containing logged propensities has been released by Lefortier et al. [22].

4 BEYOND JUST CLICKS

4.1 Missing vs Negative Feedback

A well-known issue in implicit-feedback recommender system literature is the difficulty in distinguishing between *missing* and *negative* feedback: if a user-item pair (u, i) is missing from the dataset, does this mean that u disliked i, or was simply unaware of it? Many different algorithms have been proposed to address this problem, e.g. by sampling negatives or focusing only on positive preference expressions [15, 29, 31]. These works are limited, however, in that they only take a single type of feedback into account: clicks, sales, likes,... while typically much more interaction data is available to the entities providing the recommendations. The combination of clicks, add-to-cart actions and sales, page dwell times, *not* interacting with an impression, et cetera are all rich but largely neglected feedback signals. The work of Wan and McAuley [40] focuses on monotonic behaviour chains, where different levels of feedback are all jointly taken into account into the resulting model. We wish to study whether these different types of feedback can be used to distinguish between missing and negative feedback, in order to improve offline evaluation accuracy. If we know a user clicked on an item i, but later in the same session clicked on and bought an alternative item i', this might be interpreted as a negative feedback signal for the (u, i)-pair. Analogously, if i is shown as a recommendation to u without any resulting interaction, certain conclusions might be appropriate. After 1 impression, one can give the benefit of the doubt. After a large number of impressions, however, we might be able to infer a negative relation between u and i. This issue has been tackled in the modelling stage of the recommender system, by effectively discounting the score for the item and pushing it downwards the top-N list as it is recommended again and again [21]. However, this information is mostly neglected when evaluating new models. In a broader sense, the problem of interpreting user *in*-action to better understand how users interact with live recommender systems in the movie domain was studied by Zhao et al. [45]. They identify various reasons why a user would *not* click on a given recommendation, and try to infer from context which of those reasons might explain a given sample. Further research for

better understanding user interactions in various other domains would very much help with answering this research question in more general environments.

When the distinction between *missing* and *negative* feedback is facilitated, more involved objective functions such as Generalised Area under the Curve (GAUC) from the two-class collaborative filtering field [30, 35], can be exploited for use in the positive-only use-case [38]. Furthermore, multiple pairwise learning algorithms for recommender systems utilise the information that a given user has interacted with item i, but not with item j, to model preferences and similarities [14, 31]. These methods assume that the relation between u and j is less strong than u and i, and that u effectively prefers item i over j. Although this is a plausible assumption for datasets with large numbers of items, it cannot be assured. By sampling known, or assumed with high probability, *negative* items instead of *missing* items, we believe the performance of these existing algorithms can be significantly boosted.

4.2 Impression-data for Presentation Bias

Impression-data further has its use in the off-policy evaluation setting described in Section 3. Assume we have a dataset consisting of logged feedback from multiple loggers (e.g. collected from a live A/B test): $\mathcal{D} = \mathcal{D}_{\pi_0} \cup \ldots \cup \mathcal{D}_{\pi_n}$, where $\mathcal{D}_\pi$ refers to the subset of $\mathcal{D}$ that was generated under policy π, and $\mathcal{D}$ is a set of user-item-timestamp triplets. This is the same setting as tackled in the work of Agarwal et al. [1], where they propose two provably unbiased variants of IPS that limit the variance in these environments. However, it is assumed that all propensities are known beforehand, and the loggers are effectively stochastic multi-armed contextual bandits. When this is not the case, and the policies are possibly unknown, applying IPS-like methods gets troublesome. To this end, we propose a model-agnostic way of quantifying the biases among policies. By computing the overlap in their generated recommendations, we believe the presentation bias that is inherent to the data can be normalised and alleviated. We denote the set of impressions generated under the various logging policies by $\mathcal{R} = \mathcal{R}_{\pi_0} \cup \ldots \cup \mathcal{R}_{\pi_n}$. For every context-vector x in the logging data, a set of top-N recommendations is logged. As $\mathcal{D}$ and $\mathcal{R}$ usually originate from a live A/B-test, the context-vectors x will be disjoint for different logging policies. Suppose we want to evaluate a new policy π_t on $\mathcal{D}$; we can now compute $\mathcal{R}_{\pi_t}$ for every logged impression: the set of recommendations that *would* have been shown, had π_t been in effect. Through $\mathcal{R}_{\pi_i}$, we can quantify the similarity between a logging policy π_i and the target π_t. Although more sophisticated measures can be used, a simple first option is to use the Jaccard index to compute the intersection of the generated recommendations. Assuming $\mathcal{R}_{\pi_i}$ and $\mathcal{R}_{\pi_t}$ hold the same number of generated top-N recommendations, for an identical set of context-vectors x, the similarity between these two policies then becomes

$$\text{sim}(\pi_i, \pi_t) = \frac{|\mathcal{R}_{\pi_i} \cap \mathcal{R}_{\pi_t}|}{|\mathcal{R}_{\pi_i}|}. \tag{2}$$

If the logging and target policy are identical, i.e. $\pi_i = \pi_t$, offline evaluation of π_t on $\mathcal{D}_{\pi_i}$ is effectively an *online* evaluation, as $\mathcal{R}_{\pi_t}$ does not hold hypothetical recommendations, but those that were actually shown to the user. On the other hand, if π_i and π_t generate entirely disjoint sets of recommendations, evaluating π_t on $\mathcal{D}_{\pi_i}$ will yield heavily biased results that disfavour π_t. Intuitively, the effect of this bias is correlated with $\text{sim}(\pi_i, \pi_t)$. A similarity of 0.5, indicates that 50% of the recommendations were actually shown at the time of data collection[2]. For a given evaluation function $f(\mathcal{D}, \mathcal{R}, \pi_t)$ that intends to evaluate the recommendations of π_t using $\mathcal{D}$, we propose a variant f', obtained by partitioning the data according to the logging policies and normalising according to their similarity with the target policy:

$$f'(\mathcal{D}, \mathcal{R}, \pi_t) = \frac{1}{n} \sum_{i=0}^{n} \frac{f(\mathcal{D}_{\pi_i}, \mathcal{R}_{\pi_i}, \pi_t)}{\text{sim}(\pi_i, \pi_t)}. \tag{3}$$

This formula can be decomposed even further, when normalising on a sample-by-sample basis instead of once for every logging policy. We intend to investigate this approach further from a theoretical basis, as well as empirically. Ideally, we wish to validate whether this approach can debias results obtained through off-policy evaluation, in situations where classical IPS weighting is not straightforward. This can be achieved by studying the correlation between results of our proposed approach, and those obtained through live A/B tests. Analogous to the IPS estimator presented in Equation 1, our proposed Equation 3 is prone to several of the same pitfalls. As $\text{sim}(\pi_i, \pi_t)$ approaches 0, these estimators will tend to over-compensate. It might prove beneficial, as is the case in IPS, to clip similarities to a given range or normalise them, with the goal of decreasing the variance. Since these approaches share the same intuitions, many of the proposed extensions to IPS can be studied with regards to their applicability in this setting [4, 10, 12, 16]. The work of Steck [37] on tackling popularity bias is also based on modelling bias, and normalising it in the evaluation phase. When our proposed approach is theoretically and experimentally fine-tuned and validated, interesting future work directions might open up on deriving novel learning and optimisation procedures that are unbiased as well. Furthermore, by studying the overlap between $\mathcal{D}$ and $\mathcal{R}$, we can disambiguate *organic* user behaviour (i.e. behaviour that was *not* instigated by a recommendation) and *influenced* user behaviour (i.e. behaviour that *was* instigated by a recommendation). We believe a better understanding of these different types of user behaviour will be useful to the research field as well.

5 CONCLUSION

In this paper, we have presented the key differences between on- and offline evaluation methodologies for implicit feedback recommender systems. We have motivated the need for more effective offline evaluation strategies that are successful in predicting online performance. To this end, we have formulated and discussed three main research objectives: (1) temporal evaluation, (2) off-policy or counterfactual evaluation, and (3) the use of more involved interaction data to improve upon currently existing offline evaluation methods. For each of those areas, we summarised and discussed the state-of-the-art, identifying shortcomings and proposing promising areas for future work. In the near future, we wish to incorporate the above-mentioned extensions into our SW-EVAL approach [17].

[2] A possible extension would be to include the rank of the item in the recommendation list, instead of representing the recommendations as bags-of-items. In this case, Jaccard index could be replaced by e.g. Spearman's rank correlation coefficient.

REFERENCES

[1] A. Agarwal, S. Basu, T. Schnabel, and T. Joachims. 2017. Effective Evaluation Using Logged Bandit Feedback from Multiple Loggers. In *Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '17)*. ACM, 687–696.

[2] J. Beel, M. Genzmehr, S. Langer, A. Nürnberger, and B. Gipp. 2013. A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation. In *Proc. of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys '13)*. 7–14.

[3] J. Bennett, S. Lanning, et al. 2007. The Netflix prize. In *Proc. of the KDD cup and workshop*, Vol. 2007. 35.

[4] L. Bottou, J. Peters, J. Quiñonero-Candela, D. Charles, D. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research* 14, 1 (2013), 3207–3260.

[5] A. Chaney, B. Stewart, and B. Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 224–232.

[6] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proc. of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, 456–464.

[7] E. Christakopoulou and G. Karypis. 2016. Local Item-Item Models For Top-N Recommendation. In *Proc. of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, 67–74.

[8] E. Christakopoulou and G. Karypis. 2018. Local Latent Space Models for Top-N Recommendation. In *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, 1235–1243.

[9] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. 2014. Offline and Online Evaluation of News Recommender Systems at Swissinfo.Ch. In *Proc. of the 8th ACM Conference on Recommender Systems (RecSys '14)*. 169–176.

[10] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, 198–206.

[11] C. A. Gomez-Uribe and N. Hunt. 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manage. Inf. Syst.* 6, 4, Article 13 (Dec. 2015), 19 pages.

[12] A. Gruson, P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, and B. Carterette. 2019. Offline Evaluation to Make Decisions About Playlist Recommendation Algorithms. In *Proc. of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, New York, NY, USA, 420–428.

[13] R. He, W. Kang, and J. McAuley. 2017. Translation-based Recommendation. In *Proc. of the 11th ACM Conference on Recommender Systems (RecSys '17)*. ACM, 161–169.

[14] C. Hsieh, L. Yang, Y. Cui, T. Lin, S. Belongie, and D. Estrin. 2017. Collaborative Metric Learning. In *Proc. of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, 193–201.

[15] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM '08)*. 263–272.

[16] R. Jagerman, I. Markov, and M. de Rijke. 2019. When People Change Their Mind: Off-Policy Evaluation in Non-stationary Recommendation Environments. In *Proc. of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, 447–455.

[17] O. Jeunen, K. Verstrepen, and B. Goethals. 2018. Fair Offline Evaluation Methodologies for Implicit-feedback Recommender Systems with MNAR Data. In *Proc. of the REVEAL 18 Workshop on Offline Evaluation for Recommender Systems (RecSys '18)*.

[18] M. Jugovac, D. Jannach, and M. Karimi. 2018. Streamingrec: A Framework for Benchmarking Stream-based News Recommenders. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 269–273.

[19] B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz. 2013. The Plista Dataset. In *Proc. of the 2013 International News Recommender Systems Workshop and Challenge (NRS '13)*. ACM, 16–23.

[20] R. Kohavi et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of the 1995 International Joint Conference on Artificial Intelligence*, Vol. 14. 1137–1145.

[21] P. Lee, L. Lakshmanan, M. Tiwari, and S. Shah. 2014. Modeling Impression Discounting in Large-scale Recommender Systems. In *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '14)*. ACM, 1837–1846.

[22] D. Lefortier, A. Swaminathan, X. Gu, T. Joachims, and M. de Rijke. 2016. Large-scale validation of counterfactual learning methods: A test-bed. *arXiv preprint arXiv:1612.00367* (2016).

[23] L. Li, W. Chu, J. Langford, and X. Wang. 2011. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *Proc. of the 4th ACM International Conference on Web Search and Data Mining (WSDM '11)*. ACM, 297–306.

[24] X. Li and J. She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '17)*. ACM, 305–314.

[25] X. Ning and G. Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proc. of the 2011 IEEE 11th International Conference on Data Mining (ICDM '11)*. IEEE Computer Society, 497–506.

[26] Y. Ning, Y. Shi, L. Hong, H. Rangwala, and N. Ramakrishnan. 2017. A Gradient-based Adaptive Learning Framework for Efficient Personal Recommendation. In *Proc. of the 11th ACM Conference on Recommender Systems (RecSys '17)*. ACM, 23–31.

[27] R. Otunba, R. Rufai, and J. Lin. 2017. MPR: Multi-Objective Pairwise Ranking. In *Proc. of the 11th ACM Conference on Recommender Systems (RecSys '17)*. ACM, 170–178.

[28] Outbrain. 2017. Kaggle Click Prediction Dataset. https://www.kaggle.com/c/outbrain-click-prediction/data. (2017).

[29] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. 2008. One-Class Collaborative Filtering. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM '08)*. 502–511.

[30] B. Paudel, T. Haas, and A. Bernstein. 2017. Fewer Flops at the Top: Accuracy, Diversity, and Regularization in Two-Class Collaborative Filtering. In *Proc. of the 11th ACM Conference on Recommender Systems (RecSys '17)*. ACM, 215–223.

[31] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, 452–461.

[32] M. Rossetti, F. Stella, and M. Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proc. of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, 31–34.

[33] G. Shani and A. Gunawardana. 2011. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, 257–297.

[34] A. Sinha, D. Gleich, and K. Ramani. 2016. Deconvolving Feedback Loops in Recommender Systems. In *Proc. of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. Curran Associates Inc., 3251–3259.

[35] D. Song and David A. Meyer. 2015. Recommending Positive Links in Signed Social Networks by Optimizing a Generalized AUC. In *Proc. of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 290–296.

[36] H. Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '10)*. ACM, 713–722.

[37] H. Steck. 2011. Item Popularity and Recommendation Accuracy. In *Proc. of the 5th ACM Conference on Recommender Systems (RecSys '11)*. ACM, 125–132.

[38] K. Verstrepen, K. Bhaduriy, B. Cule, and B. Goethals. 2017. Collaborative Filtering for Binary, Positiveonly Data. *SIGKDD Explor. Newsl.* 19, 1 (Sept. 2017), 1–21.

[39] J. Vinagre, A. Jorge, and J. Gama. 2015. Evaluation of recommender systems in streaming environments. *CoRR* abs/1504.08175 (2015). arXiv:1504.08175 http://arxiv.org/abs/1504.08175

[40] M. Wan and J. McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 86–94.

[41] J. Yang, C. Chen, C. Wang, and M. Tsai. 2018. HOP-rec: High-order Proximity for Implicit Recommendation. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 140–144.

[42] L. Yang, Y. Cui, Yuan X., C. Wang, S. Belongie, and D. Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-not-at-random Implicit Feedback. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 279–287.

[43] Y. Zhang, H. Lu, W. Niu, and J. Caverlee. 2018. Quality-aware Neural Complementary Item Recommendation. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 77–85.

[44] Q. Zhao, J. Chen, M. Chen, S. Jain, A. Beutel, F. Belletti, and E. H. Chi. 2018. Categorical-attributes-based Item Classification for Recommender Systems. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 320–328.

[45] Q. Zhao, M. C. Willemsen, G. Adomavicius, F. M. Harper, and J. A. Konstan. 2018. Interpreting User Inaction in Recommender Systems. In *Proc. of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, 40–48.