Jadidinejad, A. H., Macdonald, C. and Ounis, I. (2020) Using Exploration to Alleviate Closed-Loop Effects in Recommender Systems. In: 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), Xi'an, China, 25-30 Jul 2020, pp. 2025-2028. ISBN 9781450380164.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

© Association for Computing Machinery 2020. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), Xi'an, China, 25-30 Jul 2020, pp. 2025-2028. ISBN 9781450380164. http://dx.doi.org/10.1145/3397271.3401230.

http://eprints.gla.ac.uk/215383/

Deposited on: 16 June 2020

Enlighten – Research publications by members of the University of Glasgow
http://eprints.gla.ac.uk

Using Exploration to Alleviate Closed Loop Effects in Recommender Systems

Amir H. Jadidinejad, Craig Macdonald, Iadh Ounis
University of Glasgow
firstname.lastname@glasgow.ac.uk

ABSTRACT

Recommendation systems are often trained and evaluated based on users' interactions obtained through the use of an existing, already deployed, recommendation system. Hence the deployed recommendation systems will recommend some items and not others, and items will have varying levels of exposure to users. As a result, the collected feedback dataset (including most public datasets) can be skewed towards the particular items favored by the deployed model. In this manner, training new recommender systems from interaction data obtained from a previous model creates a feedback loop, i.e. a closed loop feedback. In this paper, we first introduce the closed loop feedback and then investigate the effect of closed loop feedback in both the training and offline evaluation of recommendation models, in contrast to a further exploration of the users' preferences (obtained from the randomly presented items). To achieve this, we make use of open loop datasets, where randomly selected items are presented to users for feedback. Our experiments using an open loop Yahoo! dataset reveal that there is a strong correlation between the deployed model and a new model that is trained based on the closed loop feedback. Moreover, with the aid of exploration we can decrease the effect of closed loop feedback and obtain new and better generalizable models.

ACM Reference Format:
Amir H. Jadidinejad, Craig Macdonald, Iadh Ounis. 2020. Using Exploration to Alleviate Closed Loop Effects in Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401230

1 INTRODUCTION

Recommendation systems benefit from the collaborative effect in that an effective recommendation model that predicts items of interest for a user can be obtained by examining the historical interactions of user and items. However, in reality, the historical interactions of users can be affected by any previously deployed recommender system. For instance, users may not leave feedback on items as they have not been *exposed* to them. In this way, training a recommender model on historical interactions, obtained from a previous recommender system, forms a closed loop feedback (aka bandit feedback[1]). Indeed, this feedback loop may reinforce the users' historical behavior [3, 16].

The problem of exposure bias has been well-known for some time in the evaluation of both search engines [11] and recommender systems [19]. In search scenarios, clickthrough impressions are known to be weak signals of relevance, and thus test collections for offline learning and evaluation often encompass explicit relevance judging (to identify if the clicked document is definitely relevant to the user's information need) as well as pooling (to identify other relevant items not retrieved by the single system) [11]. However, due to the subjective nature of the users' information needs in recommender systems, such explicit judging is not possible. The evaluation of both search and recommender systems can thus benefit from online evaluation, in the form of A/B testing or interleaving [17].

In contrast, our aim is to learn a better recommendation model from the observed closed loop feedback collected by the deployed system. To overcome the challenge of closed loop feedback, some recommender systems exploit a bandit-based approach, where *exploitation* – the presentation of items that the system is more confident about – is mixed with the *exploration* of items it is less confident about, in order to obtain the users' feedback about items that they would not otherwise be exposed to [13]. Indeed, the recommender systems community is increasingly concerned with reinforcement learning techniques to learn from such biased user feedback [2, 19], as exemplified by the recent REVEAL workshops at RecSys [6].

We investigate the impact of closed loop feedback on both the training and evaluation of recommendation models. Without the application of online bandit-based or reinforcement learning approaches, the classical offline training and evaluation approaches of collaborative filtering models suffer from the closed loop effect. Our contributions in this paper are hence two-folds: Firstly, we propose a novel methodology in order to assess the effect of closed loop feedback on both the training and evaluation of recommendation models. Compared to previous research [5, 15, 16] that are based on simulation frameworks, our approach is based on sampling from a real-world randomized dataset[2]. The proposed methodology provides a new perspective to analyze the closed loop effect on both the training and evaluation of recommender systems. Secondly, our experiments based on the real-world randomized dataset reveal that the evaluation based on closed loop feedback datasets is not compatible with that of the randomized open loop feedback when the deployed model has no exploration (i.e. it only exploits the highest relevant item to expose to the user), at least for the dataset and models in our experiments. On the other hand, we show that if the deployed model explores a broader range of items, the closed loop effect is decreased and the evaluation based on closed loop feedback is more compatible with the randomized open loop feedback dataset.

[1] Bandit feedback is the term mostly used in Reinforcement Learning.

[2] In the randomized dataset, each item is exposed randomly.

SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China*, https://doi.org/10.1145/3397271.3401230.

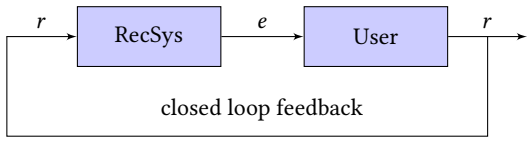


Figure 1: Closed loop feedback in recommendation systems.

2 RELATED WORK

Counterfactual Evaluation: Learning from bandit feedback [9] is a well-studied topic in Reinforcement Learning. Inverse Propensity Scoring (IPS) is a well-known approach that allows the new model to be evaluated/learned independently from the deployed model. The key idea of IPS is to weight samples based on the propensities of the observed items using importance sampling, where propensity refers to the probability of item i being shown to the user u at the point of data collection. This is a theoretically unbiased way to learn a new model based on the feedback collected from the deployed model [14]. Practically, these models need a comprehensive and stochastic deployed model that covers all possible items with accurate logged propensities [19]. Hence, despite the theoretical underpinning of the propensity-based methods, it is not easy, in reality, to satisfy the above constraints.

Algorithmic Confounding: Chaney et al. [3] showed that algorithmic confounding occurs when a recommendation system platform attempts to model user behavior without accounting for the observed recommendations. Based on simulation, they showed that closed loop feedback (1) causes homogenization of user behavior while decreasing the utility and (2) amplifies the impact of recommendation models on the distribution of item consumption. In addition, Sun et al. [15, 16] proposed strategies based on active learning to debias the effect of feedback loops. These are the most related works to ours, but they are based on synthetic datasets. We propose a novel methodology to leverage real-world randomized datasets.

Unbiased Learning to Rank: Users' feedback in recommendation systems is reminiscent of clickthrough data in information retrieval (IR). Previous research [11] has shown that there is a strong dependency between the documents presented to the user, and those for which the system receives feedback, i.e. higher-ranked documents obtain more clicks (position/presentation bias). As a result, such feedback does not reliably reflect retrieval quality [11]. Researchers in IR have proposed novel approaches for both learning [7] and evaluating [11] from user's clickthrough data that specifically factored for position bias in search results. However, due to the subjective nature of the users' information needs in recommender systems, these models are not directly applicable.

3 CLOSED LOOP FEEDBACK

The term *feedback* refers to a situation in which two (or more) dynamic systems are strongly connected together such that each system influences the other and their dynamics [1]. Figure 1 shows the closed loop information flow of a recommendation system. The deployed recommender system (RecSys) component filters (or personalizes) items for the target user (e.g. by suggesting a ranked list of items) depicted as exposure (e). As a result, the user has a partial view of the world, modulated by the RecSys component [10]. The user's recorded preferences towards those ranked items (depicted as r) are leveraged as training data to develop the next-generation of the recommendation model. These interactions are subject to *selection bias* exposed by the RecSys component [16, 17, 19].

Input: deployed RecSys model M; randomized exposure R
Output: closed loop feedback dataset
$D_M = \langle users, items, clicks \rangle$
initialize $D_M \leftarrow \emptyset$;
//iterate through the randomized exposures R;
foreach $e = \langle user, item, click \rangle \in R$ **do**
  //predict based on the previous observed history;
  if $M(D_M, user) == item$ **then**
    $D_M \leftarrow D_M \cup e$;
    //update the model's parameters ;
    $fit(M, e)$;
  end
end

Algorithm 1: closed loop feedback sampling from a randomized dataset (R) based on a deployed model (M).

The main challenge is that both the User and the RecSys components depicted in Figure 1 form a feedback loop: they are strongly connected together such that each component influences the other and their dynamics, i.e. the system presented in Figure 1 is a dynamic system where a simple causal reasoning about the system is difficult because each component influences the other, leading to a *closed loop feedback*. On the other hand, if the RecSys component in Figure 1 is a *random* model, then the interconnection between the User and the RecSys components is removed and the RecSys component has no effect on the collected feedback dataset known as *open loop feedback*.

The modeling of the users' preferences without taking into account the closed loop effect has a negative effect on both users and recommendation models. From the user's perspective, during different iterations of closed loop feedback, the RecSys component systematically restricts the perception of the user by recommending personalized items. This might create a monoculture where, during different iterations of closed loop feedback, the perception of the user becomes narrower, aka filter bubble [3, 10]. On the other hand, from the system's perspective, the recommendation models are trained based on the users feedback. During different iterations of closed loop feedback, the observed feedback is hence systematically biased towards some particular items. As a result, the observed feedback that is used for both the training and evaluation of new models becomes restricted to a highly skewed distribution of items [16]. For instance, if popular items are overexposed by the RecSys component in Figure 1, the user's perceptions may be restricted to those popular items (i.e. the user's perspective). As a result, the collected feedback dataset will be skewed towards the popular items too (i.e. the system's perspective), raising questions about the quality of machine learning models being trained and evaluated based on closed loop feedback [2, 4]. In this paper, we focus on the system's perspective. Our aim is to show the impact of the deployed model on both the training and evaluation of a new model based on closed loop feedback. For this purpose, we propose a novel methodology in the following section.

4 METHODOLOGY

Our objective is to show the impact of closed loop feedback in both the training and evaluation of recommendation models. Therefore, we aim to answer the following two research questions:

RQ1: *Is the evaluation of models based on closed loop feedback compatible with the corresponding open loop feedback?* Our hypothesis is

that the deployed model (the RecSys component in Figure 1) plays a key role when collecting closed loop feedback. In this research question, we evaluate the effectiveness of various recommendation models with and without closed loop feedback. Our aim is to demonstrate the relationship between the deployed model (M) and the new evaluated model (M').

RQ2: *How does exploration affect the closed loop feedback?* Our hypothesis is that the confounding role of the deployed model can be decreased if the deployed model explores broader ranges of items instead of exploiting highly relevant items. In this research question, we add exploration to the deployed recommendation model and analyze the effect of closed loop feedback when the deployed model involves exploration compared to the situation when the deployed model only exploits the items with the highest relevance scores.

Most current public datasets for developing and evaluating recommender systems have been collected from a deployed system. These datasets cover only the User component in Figure 1, i.e. the deployed recommender system used is not known. Even if the deployed model is identified, designing a rigorous experiment in this dynamic system is not straightforward. Therefore, we cannot investigate the closed loop effect based on the current public datasets. On the other hand, randomized datasets are promising for analyzing closed loop feedback. In a randomized dataset, items are *randomly* selected for being exposed to the user, i.e. the RecSys component depicted in Figure 1 is a *random* recommendation model.

In order to simulate the effect of closed loop feedback, we propose a novel methodology based on Algorithm 1 that samples a set of closed loop feedback D_M for a particular deployed model M from the random exposures (R). For each ⟨user, item, click⟩ tuple in the randomized dataset, the deployed model first predicts the user's preferences (r in Figure 1) based on the current state of the model M. If the output of the model is equivalent to the observed random outcome, the corresponding ⟨user, item, click⟩ will be leveraged as a closed loop feedback and the deployed model M will be updated based on the observed feedback. Since each user has been exposed to a random item, Algorithm 1 samples a subset of exposures reinforced by the deployed model M. On the other hand, if the given deployed model M is a *random* model, then the collected feedback data will form an *open loop feedback* dataset. As a result, we will be able to contrast the effectiveness of a new model on both closed and open loop scenarios. Li et al. [8] proposed a similar approach for the unbiased offline evaluation of contextual-bandit models. However, our proposed methodology provides a new perspective to assess the effect of closed loop feedback, which is not possible to investigate based on the current public datasets.

5 EXPERIMENTAL SETUP

We use the Yahoo! front page news dataset (Yahoo! R6B) [8], which is a well-known randomized dataset commonly used in the field of counterfactual learning and evaluation. A unique property of this dataset is that the displayed news articles were randomly sampled from the pool of candidate articles, and the user's feedback (i.e. clicks) were collected for each random exposure. The dataset contains 15 days of random exposure of news articles to the users. We randomly sample two consecutive days, leverage the first day to initialize the model M based on randomized open loop feedback and then apply Algorithm 1 to collect closed loop feedback (D_M) for the second day.[3] We use 80% of the feedback for training and 20% for evaluation. We repeat our experiments 10 times and report the average of the results. We evaluate the following models: A simple baseline model that recommends a random item among all available items (**Random**); An unpersonalized model that selects the top-20 most popular items and suggests them to each and every user regardless of their preferences (**Popularity**); Bayesian Personalized Ranking (**BPR**) [12] is a well-known pair-wise ranking model. The BPR model is trained based on uniform negative sampling, i.e. we randomly sample items not interacted with as negative instances for each user; Weighted Approximate-Rank Pairwise (**WARP**) [18] is another pair-wise ranking prediction model. Unlike BPR, the negative items are not chosen by random sampling: they are chosen among those negative items that would violate the desired ranking given the current state of the model. The dimensions of the user and item representations were set to 64 and each model was trained using the Adam optimizer (the learning rate is 10^{-3}) with a batch size of 256. For a particular deployed model M, we collect a closed loop feedback dataset (D_M) based on Algorithm 1. The collected closed loop feedback dataset is leveraged to train and evaluate a new model (M') in the same manner as for the deployed model (M). In addition, we evaluate the models based on the commonly used normalized Discounted Cumulative Gain (nDCG@20) metric.

6 RESULTS AND DISCUSSION

Table 1 shows the average of 10 different experiments with 95% confidence intervals for two different settings: (I) when we only exploit the best item suggested by the deployed model and (II) when we choose the item by randomly sampling from the distribution of relevance scores[4] associated with each candidate item. The rows and columns in Table 1 correspond to the deployed model (M) and the evaluated model (M'), respectively.

RQ1: Table 1 (I) shows the dependency between the deployed model and the corresponding evaluated model based on closed loop feedback when the deployed model only exploits the item with the highest relevance score (i.e. no exploration). The ground truth (open loop evaluation) is the 'Random' deployed model where there is no correlation between the deployed model and the evaluated model. From the table, we observe that the evaluation of models based on closed loop feedback is not compatible with the open loop feedback (e.g. BPR outperforms other models based on closed loop feedback while WARP is markedly better than BPR based on open loop feedback). Focusing on BPR and WARP, we observe that when the BPR is the deployed model, the performance of BPR is markedly higher than WARP, while when WARP is the deployed model, the performance of these two models are close to each other. Our experiments reveal that there is an inconsistency between the closed loop and open loop (random) evaluation. However, the study of the dependency between the deployed model and the corresponding evaluated model needs a further investigation (e.g. with various types of recommendation models).

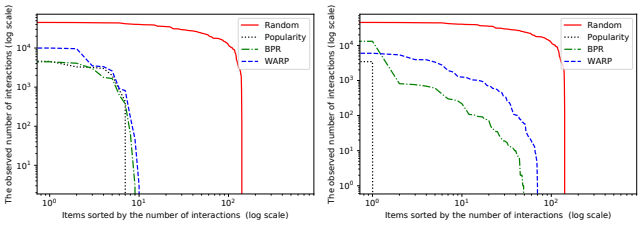
RQ2: Figure 2 shows the items' coverage of the deployed model with and without exploration. The figure shows that random sampling based on the items' relevance scores, instead of choosing the

[3] This allows us to initialize the model with a proper prior knowledge and then to investigate the effect of closed loop feedback. Otherwise, different instances of the deployed model will not coverage.

[4] We use the softmax function to map the relevance scores to a probability distribution.

Table 1: Evaluation based on NDCG@20 with (II) or without (I) exploration. 95% confidence intervals are shown with the ± symbol. Rows correspond to the deployed model while columns correspond to the new model, which is trained and evaluated based on the closed loop feedback collected by the deployed model.

	(I) Exploitation				(II) Exploitation+Exploration			
	Random	Popularity	BPR	WARP	Random	Popularity	BPR	WARP
Random	0.0005 ± 0.000	0.009 ± 0.001	0.222 ± 0.025	**0.240 ± 0.030**	0.0005 ± 0.000	0.009 ± 0.000	0.212 ± 0.026	**0.228 ± 0.027**
Popularity	0.0007 ± 0.000	0.049 ± 0.005	**0.840 ± 0.070**	0.838 ± 0.071	0.0007 ± 0.000	0.053 ± 0.013	**0.815 ± 0.081**	0.783 ± 0.099
BPR	0.0006 ± 0.000	0.045 ± 0.010	**0.852 ± 0.082**	0.839 ± 0.099	0.0007 ± 0.000	0.048 ± 0.004	0.796 ± 0.055	**0.797 ± 0.054**
WARP	0.0007 ± 0.000	0.046 ± 0.012	**0.804 ± 0.086**	0.802 ± 0.077	0.0006 ± 0.000	0.035 ± 0.007	0.583 ± 0.069	**0.584 ± 0.067**



(a) Exploitation **(b) Exploitation + Exploration**

Figure 2: Items' coverage of the deployed model without (a) and with (b) exploration. Both axes are log scaled for clarity.

item with the highest relevance score, leads to a better coverage of the items' consumption. Table 1 (II) shows the corresponding results. Compared to the situation where the deployed model only exploits the item with the highest relevance score (exploitation), the evaluation of different models based on closed loop feedback is more compatible with the open loop evaluation (i.e. when the deployed model is 'Random'). This answers RQ2; exploration does decrease the effect of closed loop feedback. However, the difference between the nDCG@20 values are small. For example, when the deployed model is BPR, although the effectiveness of BPR is not markedly higher than WARP (as we observe in the exploitation setting), the difference between the nDCG@20 absolute values of BPR and WARP is small ($\Delta_{nDCG@20} = 0.001$). A further analysis of Table 1 reveals that the performances of all models are degraded in the exploration setting. We conjecture that leveraging a more effective exploration strategy (e.g. Thompson sampling) [13] rather than random sampling is important to balance between exploitation based on closed loop feedback and exploration without degrading the systems' performances. We leave the choice of the exploration strategy for future work.

7 CONCLUSIONS

Recommendation systems are an instance of dynamic systems where a simple causal reasoning about the users' preferences is difficult because both the recommender and the user have a direct influence on each other. In this paper, we introduced the notion of closed loop feedback and proposed a novel methodology to analyze the closed loop effect based on randomized datasets. Our experiments revealed that when the deployed model has no exploration, the collected closed loop feedback is highly skewed towards a subgroup of items and the training and evaluation of recommendation models based on closed loop feedback is not compatible with the random open loop situation. However, when the deployed model involves exploration, the evaluation of recommendation models based on closed loop feedback becomes more compatible with the random open loop scenario. We know that the current public datasets are collected from a deployed model, but we are not aware of the deployed model's specifications. If the deployed model has no exploration[5] then any new model that is trained and evaluated based on closed loop feedback datasets will suffer from the closed loop effect.

ACKNOWLEDGMENTS

Work as part of EPSRC grant EP/ R018634/1: Closed-Loop Data Science for Complex, Computationally- & Data-Intensive Analytics.

REFERENCES

[1] K. Astrom and R. Murray. 2008. *Feedback Systems: An Introduction for Scientists and Engineers.* Princeton University Press.
[2] R. Cañamares, M. Redondo, and P. Castells. 2019. Multi-Armed Recommender System Bandit Ensembles. In *Proceedings of RecSys.*
[3] A. Chaney, B. Stewart, and B. Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *Proceedings of RecSys.*
[4] A. Jadidinejad, C. Macdonald, and I. Ounis. 2019. How Sensitive is Recommendation Systems' Offline Evaluation to Popularity?. In *Workshop on Reinforcement and Robust Estimators for Recommendation (REVEAL).*
[5] R. Jiang, S. Chiappa, T. Lattimore, A. György, and P. Kohli. 2019. Degenerate Feedback Loops in Recommender Systems. In *Proceedings of AIES.*
[6] T. Joachims, A. Swaminathan, Y. Raimond, O. Koch, and F. Vasile. 2018. REVEAL 2018: Offline Evaluation for Recommender Systems. In *Proceedings of RecSys.*
[7] T. Joachims, A. Swaminathan, and T. Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of WSDM.*
[8] L. Li, W. Chu, J. Langford, and X. Wang. 2011. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *Proceedings of WSDM.*
[9] J. McInerney, B. Lacker, S. Hansen, K. Higley, H. Bouchard, A. Gruson, and R. Mehrotra. 2018. Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits. In *Proceedings of RecSys.*
[10] T. Nguyen, P. Hui, F.M. Harper, L. Terveen, and J. Konstan. 2014. Exploring the Filter Bubble: The Effect of Using Recommender Systems on Content Diversity. In *Proceedings of WWW.*
[11] F. Radlinski, M. Kurup, and T. Joachims. 2008. How Does Clickthrough Data Reflect Retrieval Quality?. In *Proceedings of CIKM.*
[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of UAI.*
[13] T. Schnabel, P. Bennett, S. Dumais, and T. Joachims. 2018. Short-Term Satisfaction and Long-Term Coverage: Understanding How Users Tolerate Algorithmic Exploration. In *Proceedings of WSDM.*
[14] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proceedings of ICML.*
[15] W. Sun, S. Khenissi, O. Nasraoui, and P. Shafto. 2019. Debiasing the Human-Recommender System Feedback Loop in Collaborative Filtering. In *Companion Proceedings of WWW.*
[16] W. Sun, O. Nasraoui, and P. Shafto. 2018. Iterated Algorithmic Bias in the Interactive Machine Learning Process of Information Filtering. In *Proceedings of KDIR.*
[17] X. Wang, M. Bendersky, D. Metzler, and M. Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *Proceedings of SIGIR.*
[18] J. Weston, H. Yee, and R. Weiss. 2013. Learning to Rank Recommendations with the K-order Statistic Loss. In *Proceedings of RecSys.*
[19] L. Yang, Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-not-at-random Implicit Feedback. In *Proceedings of RecSys.*

[5] This is plausible since these models are designed to serve users in production.