

Reducing Popularity Influence by Addressing Position Bias

Andrii Dzhoha^{1,*†}, Alexey Kurennoy^{1,†}, Vladimir Vlasov^{1,†} and Marjan Celikik^{1,†}

¹Zalando SE, Berlin, Germany

Abstract

Position bias poses a persistent challenge in recommender systems, with much of the existing research focusing on refining ranking relevance and driving user engagement. However, in practical applications, the mitigation of position bias does not always result in detectable short-term improvements in ranking relevance. This paper provides an alternative, practically useful view of what position bias reduction methods can achieve. It demonstrates that position debiasing can spread visibility and interactions more evenly across the assortment, effectively reducing a skew in the popularity of items induced by the position bias through a feedback loop. We offer an explanation of how position bias affects item popularity. This includes an illustrative model of the item popularity histogram and the effect of the position bias on its skewness. Through offline and online experiments on our large-scale e-commerce platform, we show that position debiasing can significantly improve assortment utilization, without any degradation in user engagement or financial metrics. This makes the ranking fairer and helps attract more partners or content providers, benefiting the customers and the business in the long term.

Keywords

Recommender Systems, feedback loop, position bias, popularity

1. Introduction

From a long-term strategy standpoint, a modern e-commerce platform aims to be a one-stop shop for all platform-related shopping needs. That requires offering vast product selections, making it challenging for customers to find products aligned with their preferences and current needs. To address this challenge, e-commerce platforms deploy personalized recommender and ranking systems that nowadays play a central role in the customer shopping experience [1]. Yet, there is an obstacle down this path: the effectiveness of those personalization systems is reduced by a naturally present feedback loop [2]. As items ranked higher receive more user attention, the production recommender model creates a skew in the collected user interaction data in favor of itself. This skew then impacts subsequent models as they are trained on the collected data, creating a repetitive cycle and reinforcing suboptimal model behaviors. For example, it makes filter bubbles [3] and echo chambers in e-commerce [4] more persistent.

Note that the key driver behind the described feedback loop is the tendency of users to attend to some positions in the layout more than to others. This phenomenon is referred to as *position bias* [2, 5]. Position bias can lead to a lack of interaction with highly relevant items that are ranked low.

The literature on position bias largely focuses on improving the relevance of ranking and the associated theories and experiments predict gains in user engagement when the respective methods are deployed [6, 7]. We argue however that depending on the strength of the position bias and the effectiveness of the respective debiasing method, one may not observe such gains in the short term. On one hand, the strength of the position bias might not be sufficient to impact relevance significantly. On the other – mitigating position bias is known to be difficult as the associated methods often lack robustness to data sparsity [8], exhibit high variance [9, 10], or suffer from interleaving biases [11].

Yet, as this paper shows, debiasing has another poten-

tial benefit apart from driving engagement. Namely, it can spread visibility and interactions more evenly across the assortment, reducing the extra skew in the popularity of items incurred by the position bias.

Most relevant to our work are studies that have looked into how the feedback loop impacts popularity [12, 13, 14]. In simulations, they show that the feedback loop tends to make already popular items even more popular and less popular items even less popular, creating a “rich get richer” effect. Effectively, it means that recorded sets of user interactions become more homogeneous [15, 16].

In this paper, we offer a theoretical view of how the feedback loop driven by the position bias affects item popularity. We propose a model to quantify the skew in the popularity of items and the effect of the position bias on that skew. Through offline and online experiments on our e-commerce platform, we demonstrate that position debiasing can effectively spread visibility and interactions more evenly across the assortment while leaving user engagement and financial metrics intact.

2. Recommender system and popularity of items

Consider a recommender system serving a stream of incoming user requests. In response to each of those requests, the system displays a sequence of recommended items taken from a larger item vocabulary (assortment). Presented with that sequence, the user observes some of its elements and interacts with those observed items that he or she finds relevant. Over any fixed time frame, this process generates a number of interactions. Naturally, different items in the vocabulary receive different number of interactions. We use the term *popularity* to refer to the share of interactions accumulated by a given item.

If we rank the items in the vocabulary by their popularity and then plot the popularity as in Figure 1a (starting from the most popular item), we will obtain a histogram that can be typically approximated with a long-tailed distribution.

In this and subsequent sections, we will consider the cases with and without the presence of the position bias and argue about its effect on the skew of the popularity histogram.

First, consider a hypothetical case where position bias does not exist. In this setting, the user interacts with a rec-

RobustRecSys: Design, Evaluation, and Deployment of Robust Recommender Systems Workshop @ RecSys 2024, 18 October, 2024, Bari, Italy

✉ andrew.dzhoha@zalando.de (A. Dzhoha);

alexey.kurennoy@zalando.ie (A. Kurennoy);

vladimir.vlasov@zalando.de (V. Vlasov); marjan.celikik@zalando.de

(M. Celikik)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

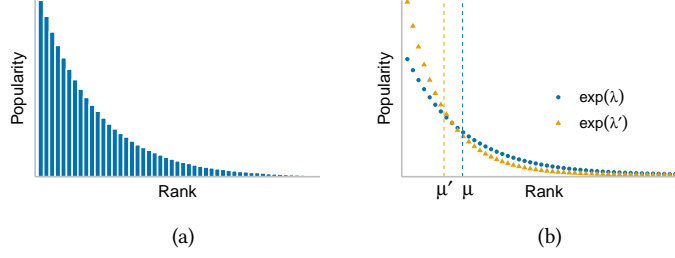


Figure 1: Distribution of popularity of items. (a) Example of the histogram. (b) Discretized exponential distribution. The distribution with parameter λ' has a stronger bias towards favoring more popular items illustrated by the expected values μ and μ' of the rank.

ommended item according to the probability of its relevance to the user. For that, we denote by D (without prime symbol) the interaction data that represents the current iteration that maintains its natural bias in popularity and *remains unaffected by the feedback loop*. This data is used by the recommender system to train and serve a model, resulting in interactions that may be confounded by the exposure mechanism – the feedback loop. Consequently, data D' is obtained in the subsequent iteration.

2.1. Interaction model without position bias

We follow the Plackett-Luce ranking model [17, 18] to describe the interaction model. Consider a situation where we display an item i given some presented ranking to the user u . Let p_{ui} denote the actual relevance of item i to the user u (i.e. the probability of the user interacting with an item provided the item has been observed). To model user engagement with recommended items R without position bias, we interpret the selection of an item at time step t from a presented ranking subset of items $r_t \in R$ as sampling from the probability mass function $f_I(i | r_t)$ of a random variable I_t . This random variable represents the selected item at time step t . Formally, this is defined as follows:

$$f_I(i | r_t) = \frac{p_{ui}}{\sum_{j \in r_t} p_{uj}}, \quad i \in r_t, \quad (1)$$

$$I_t \sim f_I(i | r_t).$$

These interactions (I_t) together with the recommended items R on the current iteration will constitute data D' on the next iteration.

We assume that the presented ranking, along with its interactions, can be viewed as a resampling process. In the absence of feedback loop effects, these observations are expected to produce data with distributions that are approximately equivalent, $D' \approx D$, neglecting randomness and user behavior/traffic changes.

This leads us to a key observation of how we model the effect of position bias on the popularity histogram, which has remained unaffected by the feedback loop until now. Specifically, we note that the collected interaction data can be viewed as a sample from a distribution over the vocabulary.

3. Interaction model within the feedback loop

A feedback loop is a mechanism that makes past ranking results appear in the data as more aligned with user preferences than they are. It is called a loop because once a

deployed model induces a skew in the data, it affects models trained on that data. Those models get deployed, and the whole phenomenon repeats. These iterations are illustrated in Figure 2.

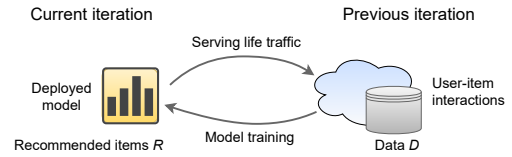


Figure 2: Model's life cycle within the feedback loop.

The most prominent driver of the feedback loop is position bias. Now, we present the case where the user behavior exhibits the position bias while interacting with the recommendation model, which is trained on data D . This means that the users may not observe all of the positions in the recommended list and may also pay different attention to different positions. In the remainder of this section, we elaborate on the concept of position bias and define it formally. Then, we extend the interaction model defined in (1).

3.1. Position bias

Let $\mathbb{1}_C$ be the indicator of the event that the user interacted with the item i (e.g. clicked on it), $\mathbb{1}_O$ – the indicator of the event that the user observed the displayed item i , and $\text{pos}(i)$ – the position at which the item was displayed. Define the relevance variable, $\mathbb{1}_V \sim \text{Bernoulli}(p_{ui})$, as the indicator of the event that the item is relevant to the user. Under the standard *examination hypothesis* [19], a displayed item receives a click if and only if the user observes it and finds relevant, i. e.

$$\mathbb{1}_C = \mathbb{1}_O \cdot \mathbb{1}_V. \quad (2)$$

From Eq. (2) it immediately follows that

$$\mathbb{P}(\mathbb{1}_C = 1 | \text{pos}(i) = k) \leq \mathbb{P}(\mathbb{1}_V = 1 | \text{pos}(i) = k).$$

Then the position bias can be defined as the following ratio:

$$\text{bias}(k) := \mathbb{P}(\mathbb{1}_C = 1 | \text{pos}(i) = k) / \mathbb{P}(\mathbb{1}_V = 1 | \text{pos}(i) = k).$$

From Eq. (2), it follows that

$$\mathbb{P}(\mathbb{1}_C = 1 | \text{pos}(i) = k) = \mathbb{P}(\mathbb{1}_C \mathbb{1}_V = 1 | \text{pos}(i) = k),$$

hence by the definition of conditional probability, the position bias is the probability of being clicked conditional on being relevant:

$$\text{bias}(k) = \mathbb{P}(\mathbb{1}_C = 1 | \mathbb{1}_V = 1, \text{pos}(i) = k).$$

Finally, assuming O and V are independent events conditional on the position, we have that

$$\begin{aligned} \mathbb{P}(\mathbb{1}_C = 1 \mid \text{pos}(i) = k) = \\ \mathbb{P}(\mathbb{1}_O = 1 \mid \text{pos}(i) = k) \mathbb{P}(\mathbb{1}_V = 1 \mid \text{pos}(i) = k), \end{aligned}$$

and the bias is just the probability of observing the item i displayed at the k -th position,

$$\text{bias}(k) = \mathbb{P}(\mathbb{1}_O = 1 \mid \text{pos}(i) = k).$$

The position bias is commonly modeled as follows [6]:

$$\text{bias}(k) \propto k^{-\beta}, \quad (3)$$

where the parameter β controls the severity of bias.

3.2. Interaction model

To reason about the feedback loop dynamics of user behavior and algorithmic recommendations built on observations D , we define a model of how users engage with recommended items R considering the presence of position bias. For that, we extend the model (1) with the position bias defined earlier as follows:

$$\begin{aligned} f_I(i \mid \mathbf{r}_t) = \frac{\text{bias}(\text{pos}(i \mid \mathbf{r}_t)) \cdot p_{ui}}{\sum_{j \in \mathbf{r}_t} \text{bias}(\text{pos}(j \mid \mathbf{r}_t)) \cdot p_{uj}}, \quad i \in \mathbf{r}_t, \\ I_t \sim f_I(i \mid \mathbf{r}_t), \end{aligned} \quad (4)$$

where the item's selection is done by the user u at time step t given a presented ranking $\mathbf{r}_t \in R$. This way, position bias introduces extra skew into sampling from the underlying distribution of the user judgment. The interactions (I_t) will produce biased data D' with distributions different from D .

4. Skew in the distribution of popularity of items

In this section, we demonstrate and quantify the skew in the popularity histogram based on interaction data D . As we explained earlier, we view such interaction data as a sample from a distribution over the vocabulary of items. A common approach to model such a distribution is by discretizing a continuous density, like exponential distribution, which gives a suitable approximation to a discrete empirical distribution. We will use the assumption of an exponential distribution.

4.1. Sample selection bias

Let us first turn to the case where the position bias does not exist and let λ be the rate of the exponential distribution that approximates the popularity histogram or, equivalently, the item sampling distribution in that case.

Define the random variable X as the popularity rank of an item. Now, the interaction model (1) without position bias results in sampling a random variable X_t at time step t from an exponential distribution with the rate parameter λ . As a result, the current iteration produces observations (X_t) along with interactions (I_t). These observations are expected to yield a similar distribution of popularity of items.

Next consider the interaction model (4) with position bias. The interpretation of the interaction data as an item sample applies in this scenario too but the sampling probabilities

change. We encounter biased sampling of X_t because the sampling distribution is different from the target population $\exp(\lambda)$. This is known as the sample selection bias [20]. In this scenario, the probability density function of X can be described as follows using the weighted exponential distribution:

$$f_X(x) = \frac{w_{\text{bias}}(x)\lambda e^{-\lambda x}}{\int_0^\infty w_{\text{bias}}(x)\lambda e^{-\lambda x} dx} \propto \lambda' e^{-\lambda' x}, \quad x > 0, \quad (5)$$

where x corresponds to the popularity rank of an item and $w_{\text{bias}}(x)$ is a weighting function that describes the effect of the position bias on the popularity at different ranks.

Since the bias is typically a monotonically decreasing function of the position and popular items are generally shown in earlier positions, it is natural to expect that $\lambda' > \lambda$ or, in other words, that the popularity histogram has a greater skew when the position bias is present. Such positive skewness can be demonstrated by estimating a monotonic density under selection bias sampling [21].

4.2. Influence on popularity

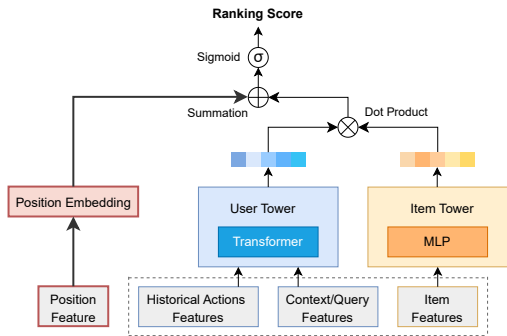
Without correcting for the bias, the distribution of popularity of items will follow $\exp(\lambda')$ from Eq. (5), exhibiting a stronger bias towards more popular items as depicted in Figure 1b. A commonly used method to correct for sample selection bias is the inverse probability-weighted method originated from Horvitz and Thompson [22], akin to the inverse propensity scores method used in counterfactual learning-to-rank [7].

In this manner, we show how the feedback loop driven by the position bias affects item popularity. Addressing position bias mitigates its influence on popularity, effectively reducing the skew. The validity of this popularity amplification is supported by simulations conducted in [12], where the sampling process, derived from user interactions, can be seen as a form of sample selection bias. If we further assume that the deployment of a debiased model would make the popularity histogram look closer to the hypothetical case in which the position bias is absent, we should expect the position debiasing to reduce the skew in the recorded popularity histogram. This is exactly what we observed in our debiasing experiment that we present next.

At last, we quantify the skew in the popularity of items induced by the feedback loop as the relative change $(\lambda' - \lambda) / \lambda$ between the actual parameter λ and the biased parameter λ' of Eq. (5). The magnitude of the change reflects the strength of the feedback loop's influence on popularity.

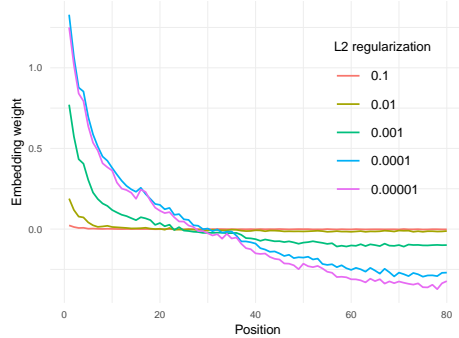
5. Experiments and results

To address position bias and counteract the extra skew in the popularity of items influenced by the feedback loop, we integrate position-aware learning [23] into the ranking model that powers both Browse and Search use cases of our e-commerce platform's catalog. This approach models positional information as a feature during training, allowing the model to separate the impact of item position from its actual relevance. Due to its simplicity, this method is widely used in practical applications [24]. In contrast, another common approach – using inverse propensity weighting transformations during training – often results in challenges



(a) The model architecture with the position branch.

Figure 3: Position-aware learning.



(b) Embedding weight per position after training.

related to the accuracy of the transformations [23] and high variance [9].

Dataset. Our dataset consists of a sample of catalog sessions. Each session consists of articles displayed in response to the customer’s browse or search request. This set of articles is combined with contextual data (e.g. market, device type, browsing category, etc), user history prior to the time of the request (previous product clicks, add-to-cart and add-to-wishlist events, purchases), and information about which of the displayed received an interaction from the customer. The training dataset consists of 250 million sessions, involving 71 million unique customers across 25 markets, with an average history length of 24 actions. We split the sessions temporarily to create the training and test datasets to ensure no data leakage.

Base Model. We performed debiasing on top of an existing catalog ranking model. This model has a two-tower architecture [25, 26]. It did not have any mechanisms to remove position bias prior to our intervention. The training objective of the model is to rank a given set of items. This task is modeled as a pointwise prediction problem using binary cross entropy, where we predict the probability of a customer performing a positive action. Within this model, the user tower utilizes historical action sequences and contextual data to produce a user embedding, while the item tower represents item embeddings. Then, these embeddings are combined using a dot product operation to produce a score per item. We use early stopping as a regularization technique to halt training when parameter updates no longer yield improvements on a validation set. Figure 3a depicts the final architecture, which includes the baseline model along with a newly added shallow position branch.

Methodology. Using the position-aware approach [24, 27], we add position information as a feature, allowing the model to disentangle the influence of item position from the true relevance of the probability that a user would engage with an item. During training, the model is conditioned on positions, while during serving, it becomes position-independent by setting a default value. To prevent potential negative effects of correlation with other features, we isolate this positional feature from the remaining features by using a shallow position branch. Additionally, to prevent overfitting on position information, we apply L2 regularization to the position embeddings. Figure 3b illustrates the position embedding weights for each position based on

the regularization applied. Higher weights indicate greater overfitting to position information during training. Adding the position branch did not significantly change the training dynamics. It slightly increased the loss, but the number of epochs required to reach saturation remained the same.

During the offline evaluation, we use inverse-propensity-weighted NDCG to measure relevance and an average recommendation popularity metric to detect improvements in popularity skew. In the online experiment, we also measure the effective catalog size. The main metrics are described as follows:

- Recall@k [28]: Proportion of all relevant items within top-k items.
- Inverse Propensity Score weighted NDCG (IPS-NDCG@k) [7]: Assesses ranking effectiveness by considering the position of relevant items within the top-k list. All attributed items are considered as relevant and their relevance is weighted based on inverse propensity scores. The propensities were modeled as Eq. (3), and their severity β was estimated using the Expectation-Maximization algorithm [29]. It is essentially a correction for bias in Eq. (5).
- Average Recommendation Popularity within top k items (ARP@k) [30]: Measures the average popularity of recommended items in each list. For any item in the list, popularity is computed by the number of interactions accumulated in the preceding days. A higher ARP indicates a greater propensity for popular items within the recommendations.
- Effective Catalog Size (ECS@X) [31]: Measures the share of items that constitute X% of all interactions aiming at describing the distribution of interactions.

5.1. Offline experiments

We offline evaluate the model on a holdout set containing instances (catalog sessions) from the subsequent day, ensuring that the evaluation data remains unseen during the training phase. Table 1 presents the results of a grid search for L2 regularization, identifying the best model for further online experiments with an L2 value of 0.001. After tuning the regularization, the offline evaluation showed no statistically significant improvement in IPS-NDCG (relevance), but did demonstrate a notable improvement of -4.34% in ARP, with a statistically significant difference (p-value < 0.05). Further adjustments in either direction did not improve the results.

Additionally, we included the performance of random and popularity-based baselines.

Table 1

Results of grid search for L2 regularization compared to the baseline model without position-aware learning. Best results for each metric are highlighted with an underscore.

Model	Recall@6	ARP@6	IPS-NDCG@6
L2 = 0.001	<u>-0.15%</u>	<u>-4.34%</u>	<u>0.16%</u>
L2 = 0.0001	-0.97%	-8.66%	-1.43%
L2 = 0.00001	-1.64%	-14.68%	-2.70%
Popularity	-28.94%	129.10%	-14.78%
Random	-49.51%	<u>-48.19%</u>	-27.11%

5.2. Online experiment

We conducted an online A/B test on the ranking use case, where we allocated equal traffic splits among variants over several weeks to achieve the minimum detectable effect for the success KPI, with a p-value < 0.05. The findings from offline experiments were consistent with the outcomes of the A/B test:

- No statistically significant changes in the main KPIs, customer engagement and financial metrics, or either of the guardrail KPIs: net merchandise value after return per user and discovery return days per user.
- Decrease (i.e. improvement) in the popularity metric by 5.7% in ARP@6.
- Increase (i.e. improvement) in the catalog items utilization by 3.1% in ECS@10.

The results are further detailed in Table 2 and Table 3.

Table 2

The relative change of the average recommendation popularity (ARP@k).

k	top-6	top-12	top-24	top-84
ARP	-5.7%	-5.4%	-4.9%	-2.9%

The experiments demonstrate that position debiasing can effectively spread visibility and interactions more evenly across the assortment, maintaining user engagement and financial metrics even when the strength of the position bias is not sufficient to impact relevance significantly. This makes the ranking fairer and helps attract more partners or content providers, benefiting the customers and the business in the long term.

At last, following the deployment of the debiased model, we calculated the skew in the popularity of items as the relative change between the distribution parameters before and after the model rollout, as defined in the previous section. We approximated the distributions using exponential distribution with parameters, determined through maximum likelihood distribution fitting. The rollout of the debiased model resulted in a 2.5% reduction in skew, indicating a shift towards a more balanced distribution across items. It's important to note that this comparison in skew is not rigorous due to inherent daily variations in factors such as user traffic patterns, behavior distribution, introduction of new items, and other sources of randomness. As a result, it serves as a supplementary metric to the main KPIs.

6. Conclusion

The paper has demonstrated that debiasing can spread visibility and interactions more evenly across the assortment without hurting user engagement and financial metrics. This makes the ranking fairer and helps attract more partners or content providers, benefiting the customers and the business in the long term. We have provided a theoretical explanation of how the feedback loop, influenced by position bias, impacts popularity. Through experiments on our e-commerce platform, we have showcased these findings.

References

- [1] P. M. Alamdari, N. J. Navimipour, M. Hosseinzadeh, A. A. Safaei, A. Darwesh, A systematic study on the recommender systems in the e-commerce, *IEEE Access* 8 (2020) 115694–115716. doi:10.1109/ACCESS.2020.3002803.
- [2] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, X. He, Bias and debias in recommender system: A survey and future directions, *ACM Trans. Inf. Syst.* 41 (2023). URL: <https://doi.org/10.1145/3564284>. doi:10.1145/3564284.
- [3] H. Steck, Calibrated recommendations, in: *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18*, Association for Computing Machinery, New York, NY, USA, 2018, p. 154–162. URL: <https://doi.org/10.1145/3240323.3240372>. doi:10.1145/3240323.3240372.
- [4] Y. Ge, S. Zhao, H. Zhou, C. Pei, F. Sun, W. Ou, Y. Zhang, Understanding echo chambers in e-commerce recommender systems, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, Association for Computing Machinery, New York, NY, USA, 2020, p. 2261–2270. URL: <https://doi.org/10.1145/3397271.3401431>. doi:10.1145/3397271.3401431.
- [5] A. Collins, D. Tkaczyk, A. Aizawa, J. Beel, Position Bias in Recommender Systems for Digital Libraries, 2018, pp. 335–344. doi:10.1007/978-3-319-78105-1_37.
- [6] T. Joachims, A. Swaminathan, T. Schnabel, Unbiased learning-to-rank with biased feedback, in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17*, Association for Computing Machinery, New York, NY, USA, 2017, p. 781–789. URL: <https://doi.org/10.1145/3018661.3018699>. doi:10.1145/3018661.3018699.
- [7] Z. Ovaisi, R. Ahsan, Y. Zhang, K. Vasilaky, E. Zhelleva, Correcting for selection bias in learning-to-rank systems, in: *Proceedings of The Web Conference 2020, WWW '20*, Association for Computing Machinery, New York, NY, USA, 2020, p. 1863–1873. URL: <https://doi.org/10.1145/3366423.3380255>. doi:10.1145/3366423.3380255.
- [8] X. Wang, M. Bendersky, D. Metzler, M. Najork, Learning to rank with selection bias in personal search, in: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16*, Association for Computing Machinery, New York, NY, USA, 2016, p. 115–124. URL: <https://doi.org/10.1145/2911451.2911537>. doi:10.1145/2911451.2911537.

Table 3

The relative change of the effective catalog size (ECS@X).

X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ECS	3.1%	2.2%	1.8%	1.4%	1.3%	1.1%	1.0%	0.8%	0.5%

- [9] Y. Saito, S. Yaginuma, Y. Nishino, H. Sakata, K. Nakata, Unbiased recommender learning from missing-not-at-random implicit feedback, in: Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 501–509. URL: <https://doi.org/10.1145/3336191.3371783>. doi:10.1145/3336191.3371783.
- [10] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, T. Joachims, Recommendations as treatments: Debiasing learning and evaluation, in: M. F. Balcan, K. Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, PMLR, New York, New York, USA, 2016, pp. 1670–1679. URL: <https://proceedings.mlr.press/v48/schnabel16.html>.
- [11] A. Vardasbi, H. Oosterhuis, M. de Rijke, When inverse propensity scoring does not work: Affine corrections for unbiased learning to rank, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1475–1484. URL: <https://doi.org/10.1145/3340531.3412031>. doi:10.1145/3340531.3412031.
- [12] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, R. Burke, Feedback loop and bias amplification in recommender systems, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2145–2148. URL: <https://doi.org/10.1145/3340531.3412152>. doi:10.1145/3340531.3412152.
- [13] Z. Zhu, Y. He, X. Zhao, J. Caverlee, Popularity bias in dynamic recommendation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2439–2449. URL: <https://doi.org/10.1145/3447548.3467376>. doi:10.1145/3447548.3467376.
- [14] D. Jannach, L. Lerche, I. Kamehkhosh, M. Jugovac, What recommenders recommend: an analysis of recommendation biases and possible countermeasures, *User Modeling and User-Adapted Interaction* 25 (2015) 427–491. URL: <https://doi.org/10.1007/s11257-015-9165-3>. doi:10.1007/s11257-015-9165-3.
- [15] R. Jiang, S. Chiappa, T. Lattimore, A. György, P. Kohli, Degenerate feedback loops in recommender systems, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 383–390. URL: <https://doi.org/10.1145/3306618.3314288>. doi:10.1145/3306618.3314288.
- [16] A. J. B. Chaney, B. M. Stewart, B. E. Engelhardt, How algorithmic confounding in recommendation systems increases homogeneity and decreases utility, in: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 224–232. URL: <https://doi.org/10.1145/3240323.3240370>. doi:10.1145/3240323.3240370.
- [17] R. D. Luce, The choice axiom after twenty years, *Journal of mathematical psychology* 15 (1977) 215–233. doi:10.1016/0022-2496(77)90032-3.
- [18] R. L. Plackett, The analysis of permutations, *Journal of the Royal Statistical Society Series C: Applied Statistics* 24 (1975) 193–202. doi:10.2307/2346567.
- [19] N. Craswell, O. Zoeter, M. Taylor, B. Ramsey, An experimental comparison of click position-bias models, in: Proceedings of the 2008 international conference on web search and data mining, 2008, pp. 87–94. doi:10.1145/1341531.1341545.
- [20] B. Zadrozny, Learning and evaluating classifiers under sample selection bias, in: Proceedings of the Twenty-First International Conference on Machine Learning, ICML '04, Association for Computing Machinery, New York, NY, USA, 2004, p. 114. URL: <https://doi.org/10.1145/1015330.1015425>. doi:10.1145/1015330.1015425.
- [21] J. Qin, Length Biased Sampling, *Multiplicative Censoring and Survival Analysis*, Springer Singapore, Singapore, 2017, pp. 519–558. URL: https://doi.org/10.1007/978-981-10-4856-2_25. doi:10.1007/978-981-10-4856-2_25.
- [22] D. G. Horvitz, D. J. Thompson, A generalization of sampling without replacement from a finite universe, *Journal of the American Statistical Association* 47 (1952) 663–685. URL: <http://www.jstor.org/stable/2280784>.
- [23] X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, F. Sun, Model ensemble for click prediction in bing search ads, in: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, p. 689–698. URL: <https://doi.org/10.1145/3041021.3054192>. doi:10.1145/3041021.3054192.
- [24] H. Guo, J. Yu, Q. Liu, R. Tang, Y. Zhang, Pal: a position-bias aware learning framework for ctr prediction in live recommender systems, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 452–456. URL: <https://doi.org/10.1145/3298689.3347033>. doi:10.1145/3298689.3347033.
- [25] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 191–198. URL: <https://doi.org/10.1145/2959100.2959190>. doi:10.1145/2959100.2959190.
- [26] J. Yang, X. Yi, D. Zhiyuan Cheng, L. Hong, Y. Li, S. Xiaoming Wang, T. Xu, E. H. Chi, Mixed negative sampling for learning two-tower neural networks

- in recommendations, in: Companion Proceedings of the Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 441–447. URL: <https://doi.org/10.1145/3366424.3386195>. doi:10.1145/3366424.3386195.
- [27] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, E. Chi, Recommending what video to watch next: a multitask ranking system, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 43–51. URL: <https://doi.org/10.1145/3298689.3346997>. doi:10.1145/3298689.3346997.
- [28] Y.-M. Tamm, R. Damdinov, A. Vasilev, Quality metrics in recommender systems: Do we calculate metrics consistently?, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 708–713. URL: <https://doi.org/10.1145/3460231.3478848>. doi:10.1145/3460231.3478848.
- [29] X. Wang, N. Golbandi, M. Bendersky, D. Metzler, M. Najork, Position bias estimation for unbiased learning to rank in personal search, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 610–618. URL: <https://doi.org/10.1145/3159652.3159732>. doi:10.1145/3159652.3159732.
- [30] H. Yin, B. Cui, J. Li, J. Yao, C. Chen, Challenging the long tail recommendation, Proc. VLDB Endow. 5 (2012) 896–907. URL: <https://doi.org/10.14778/2311906.2311916>. doi:10.14778/2311906.2311916.
- [31] C. A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Trans. Manage. Inf. Syst. 6 (2016). URL: <https://doi.org/10.1145/2843948>. doi:10.1145/2843948.