Unbiased Offline Evaluation for Learning to Rank with Business Rules

MATEJ JAKIMOV, Amazon Music, Germany
ALEXANDER BUCHHOLZ, Amazon Music, Germany
YANNIK STEIN, Amazon Music, Germany
THORSTEN JOACHIMS, Cornell University, USA

For industrial learning-to-rank (LTR) systems, it is common that the output of a ranking model is modified, either as a results of post-processing logic that enforces business requirements, or as a result of unforeseen design flaws or bugs present in real-world production systems. This poses a challenge for deploying off-policy learning and evaluation methods, as these often rely on the assumption that rankings implied by the model's scores coincide with displayed items to the users. Further requirements for reliable offline evaluation are proper randomization and correct estimation of the propensities of displaying each item in any given position of the ranking, which are also impacted by the aforementioned post-processing. We investigate empirically how these scenarios impair off-policy evaluation for learning-to-rank models. We then propose a novel correction method based on the Birkhoff-von-Neumann decomposition that is robust to this type of post-processing. We obtain more accurate off-policy estimates in offline experiments, overcoming the problem of post-processed rankings. To the best of our knowledge this is the first study on the impact of real-world business rules on offline evaluation of LTR models.

CCS Concepts: • **Information systems → Evaluation of retrieval results**; **Recommender systems**; **Learning to rank**.

Additional Key Words and Phrases: learning-to-rank, off-policy evaluation, business rules, Birkhoff-von-Neumann decomposition

ACM Reference Format:
Matej Jakimov, Alexander Buchholz, Yannik Stein, and Thorsten Joachims. 2023. Unbiased Offline Evaluation for Learning to Rank with Business Rules. In . ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A/B testing is the gold standard for evaluating learning-to-rank (LTR) systems in industrial recommender systems. However, reliable offline evaluation of new ranking policies is often even more important as it allows researchers to evaluate a large number of possible ranking policies without deployment to production. By finding the most promising ranking policies offline, the risk of suboptimal user experience is reduced.

Off-policy evaluation of new policies on historic data requires adequate strategies to deal with biases coming from the way users interact with the system, i.e., commonly referred to as *click models* [6]. Previous work showed that estimators leveraging randomization of rankings such as in the Item-Position Model [12] (IPM) can be more accurate compared to methods assuming a stronger model of user behavior such as the Position-Bias Model (PBM) [5, 8]. If only a subset of candidates is presented to users, randomization is necessary to avoid selection bias [15].

Typical architectures of industrial LTR system consist of at least four layers: candidate generation, ranking, post-processing and presentation layer. Many LTR systems in industry [16] use some kind of business rules that intentionally

Manuscript submitted to ACM

modify the ranking in the post-processing step. For example, marketing considerations (or sponsored content) might justify to increase exposure of certain items by displaying them in a top position. In case randomization and propensity computation is performed in the ranking layer, the propensities no longer represent the actual probabilities of presenting an item at some position. This propensity distortion can bias off-policy estimation.

2 BACKGROUND

Off-policy evaluation methods allow us to estimate a reward r (i.e., number of clicks, conversions, Discounted Cumulative Gain, etc.) for a new policy π, called *target policy*, provided observations $(x_i, y_i, \{r_{i,1}^{\pi_0}, \ldots, r_{i,n}^{\pi_0}\})$, indexed by observation i, generated by another policy π_0, called *logging policy*. As the space of possible rankings $y_i \sim \pi(x_i)$ of length n is usually very large ($O(n!)$), a common assumption is that the reward r_i for the whole ranking y_i can be decomposed into a sum of rewards $r_{i,j}$, one for each item j in the ranking [1]. We also assume that the logging policy ensures a non-zero probability of observing a reward for every item for which the target policy has a non-zero probability of observing a reward. This assumption is called the *full-support assumption*. We then obtain an unbiased estimate of the reward r_i for π using observations $(x_i, y_i, \{r_{i,1}^{\pi_0}, \ldots, r_{i,n}^{\pi_0}\})$ generated by logging policy π_0:

$$\mathbb{E}_\pi[r_i] = \sum_{j=1}^{n} w(y_{i,j}) \cdot \lambda(j) \cdot r_{i,j}^{\pi_0} \tag{1}$$

where $r_{i,j}^{\pi_0}$ is the reward for item $y_{i,j}$ (e.g. 1 if clicked), $w(\cdot)$ is a weight determined by the used estimator (e.g. PBM, IPM [12] or INTERPOL [5]) and $\lambda(\cdot)$ is a function of displayed position that defines the metric of interest (e.g. $\lambda(j) = log(1+j)^{-1}$ for DCG, see [1]).

For the **PBM** we define $w_{PBM}(y_j)$ as a ratio of position-biases b for positions where the target ($rank^{\pi}$) and logging policy ($rank^{\pi_0}$) placed the item y_j: $w_{PBM}(y_j) = \frac{b_{rank^{\pi}(y_j)}}{b_{rank^{\pi_0}(y_j)}}$. The set of position-biases, one for each position, is called position-bias curve. It is possible to estimate the position-bias curve from logged data using Expectation-Maximization [19], but estimating position-bias curve from randomized data tends to be more accurate [2, 17]. While the PBM estimator usually provides estimates with low variance, it is sensitive to imperfect estimation of the position-bias curve which can lead to bias in the estimated reward [5].

The **IPM** [12] defines [1] $w_{IPM}(y_j) = \frac{\mathbb{1}(rank^{\pi_0}(y_j)=rank^{\pi}(y_j)}{\mathbb{P}_{\pi_0}(rank^{\pi_0}(y_j)=rank^{\pi}(y_j))}$. This model thus requires non-zero probability of an item being displayed at any position by π_0. Typically, the IPM is less prone to bias at the cost of increased variance [12].

INTERPOL [5] interpolates between the PBM and IPM model by using the PBM for local *windows* of neighboring items, whereas across windows the IPM model is used. This allows to trade-off some bias for variance and minimize the MSE of the offline evaluation compared to the plain PBM or IPM estimators.

Randomization Schemes and Birkhoff-von-Neumann Decomposition. The estimators introduced above require π_0 to be stochastic and assume knowledge of probabilities of placing items j at position k in the form of a doubly-stochastic propensity matrix $P_{j,k}$. There are several options for obtaining suitable logging policies π_0 that respect the full-support assumption $P_{j,k} > 0$. One is to use a LTR policy that is inherently stochastic, such as a policy based on nested softmax sampling [4, 14]. However, computing propensities requires computationally expensive Monte Carlo estimation via sampling of possible rankings. Another option is to use a deterministic ranker and apply a randomization scheme to the resulting ranking, such as randomization based on Birkhoff-von-Neumann decompositions (BvN) [10, 18].

[1] For simplicity we assume that the target policy is deterministic.

The BvN scheme decomposes a predetermined propensity matrix P into a set of permutation matrices $\Pi_1, \ldots, \Pi_M$ that can be applied to deterministic rankings, each with some probability $p_1, \ldots, p_M$, so that: $P = \sum_{m=1}^{M} p_m \Pi_m$ and $\sum_{m=1}^{M} p_m = 1$. Let n be the number of available items for ranking, then the BvN decomposition of size at most n^2 can be computed with time complexity $O(n^{4.5})$. See [7, 10] for more details.

Business Rules and Other Modifications to Ranking. In practice the ranking provided by LTR algorithm is often post-processed, both intentionally, in a form of *business rules*, and unintentionally due to unexpected behavior in the processing pipeline (i.e., software bugs).

One of the most common type of business rules is enforcement of visibility for certain items by always displaying them at the top positions, i.e. *pinning*. Another common type of intentional post-processing is diversification [11].

As an example of unintentional modification to ranking, imagine a system that only presents an item to the user if the system successfully fetches the corresponding image. If the item is newly added to the system, it is possible that the image is not yet present in a cache and the request for the image times out. We only consider *pinning* in the rest of the paper.

3 PROPOSED SOLUTION

If the ranking is post-processed between the randomization step and the presentation layer, the propensity matrix $P_{j,k}$ of the randomization scheme no longer represents the true probabilities of an item j being displayed at position k. This can lead to a biased estimate of the reward. We assume that it is known what business rules were applied and how they would have been applied to alternative rankings. Then the propensity matrix $P_{j,k}$ can be corrected and a new propensity matrix $P'_{j,k}$, describing the actual propensities after application of business rules, can be derived. In general we could use Monte Carlo estimation of P' (see Algorithm 2 from Appendix C), however such approach is computationally expensive and introduces another source of variance in the off-policy evaluation.

If the BvN randomization scheme is used then it is possible to obtain an exact P' with complexity $O(nM)$, where n is the number of ranked items and M is the size of the BvN decomposition. We represent a ranking y by a matrix Y where $Y_{j,k} = 1$ if item j was ranked at position k and 0 otherwise. We assume that the deterministic part of the logging policy π_0^{ranker} ranks items by their index, i.e. $Y = I$. We define a function $B(Y)$ that returns a permutation matrix corresponding to the application of all business rules to the ranking Y. To obtain P', we iterate over all permutations from the BvN decomposition that the logging policy used, and apply the business rules to the permuted rankings: $P' = \sum_{m=1}^{M} B(\Pi_m) \cdot p_m \Pi_m$, as explained in Algorithm 1. Thereby we recover the corrected propensity matrix P'.

Algorithm 1 Estimation using BvN

Require: observation x_i, business rules B, BvN decomposition from logging policy $\Pi_1, \ldots, \Pi_M, p_1, \ldots, p_M$ and the deterministic ranker from logging policy π_0^{ranker}

```
P'_{j,k} ← 0
Y ← π_0^ranker(x_i)                ▷ Ranking for the observation x_i provided by deterministic ranker
for m ∈ [1 ... M] do               ▷ Iterating over the BvN decomposition used by the logging policy
    Y^m ← Π_m · Y                  ▷ Permute the ranking by Π_m
    Y' ← B(Y^m) Y^m                ▷ Permute the ranking Y^m by applying the business rules
    P' ← P' + p_m Y'               ▷ π_0 could have applied Π_m with probability p_m, add p_m for each item j and its final position
end for
```

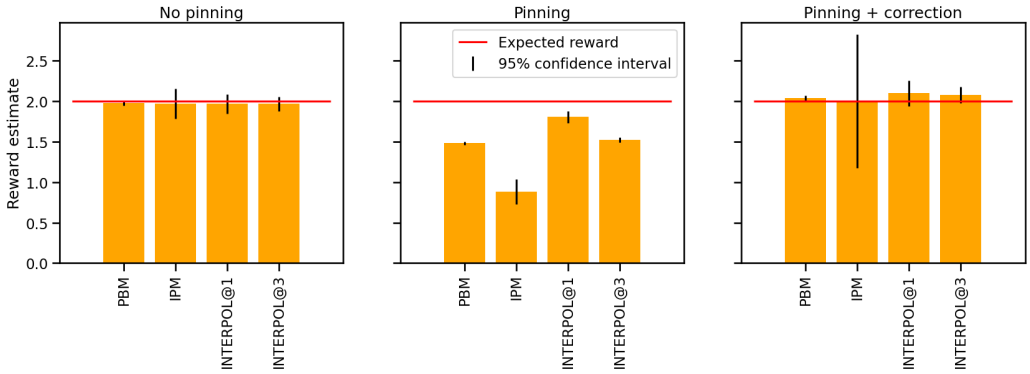

Fig. 1. We show the effect of pinning an item to the first position on accuracy of off-policy estimators: PBM (with position-bias curve estimated by *PA-IH* algorithm from [17]), IPM and INTERPOL with *window size* set to 1 and 3 respectively. In the middle graph, the ranking is first randomized and propensities are estimated. Then one of the items with low relevance is moved to the first position with 95% probability, which invalidates the propensities. In the right graph we apply pinning with 95% probability and then post-process propensity matrix using Algorithm 1.

Deterministic vs. Stochastic Business Rules. In case business rules are applied with probability 1, corrected propensity matrix can contain values $P'_{j,k} \in \{0, 1\}$ and violate the full-support assumption from Section 2. This can be avoided by ensuring that modifications to the ranking are only applied with some probability. This is the case for many unintentional modifications due to flaws in the system. For business rules it is often acceptable to apply them only with high probability. An algorithm that incorporates stochasticity of business rules can be found in the Appendix, see Algorithm 3 in Appendix C. This algorithm generalizes Algorithm 1 and will be used in our experiments.

4 EXPERIMENTS

To evaluate the effectiveness of our solution we investigate the effect of pinning an item to the first position in the logging policy. We follow the setup of [5]. A full description is given in Appendix B. The left graph in the Figure 1 shows accurate estimates when no pinning is applied, as the expected reward (red line) of the target policy is within the 95% confidence interval for the mean of the reward. In the middle graph the pinning is applied. As expected, this biases the results. When propensities are corrected by using Algorithm 3, the results are more accurate and we recover the expected reward. More detailed results are in Appendix D.

5 CONCLUSIONS AND FUTURE WORK

We show that a randomization scheme based on the Birkhoff-von-Neumann decomposition enables the correction of otherwise biased propensity matrices. This allows practitioners to achieve unbiaed off-policy evaluation when the ranking was post-processed by business rules, as often occurs in real-world production systems. We also show that such post-processing can considerably bias off-policy estimation if no correction is made. To the best of our knowledge, this is the first contribution investigating the effect of business rules (or other modifications to the ranking) on the quality of off-policy evaluation. In the future we will investigate additional types of modifications, such as dropping certain items, diversification and scenarios where not all ranked items are presented to the user (as in the limited visibility setting [15]). We also plan to study in more detail how modifications to the ranking impact accuracy of off-policy evaluation and off-policy learning in the real-world setting, using data from a deployed production system.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 5–14. https://doi.org/10.1145/3331184.3331202

[2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 474–482. https://doi.org/10.1145/3289600.3291017

[3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 385–394.

[4] Alexander Buchholz, Jan Malte Lichtenberg, Giuseppe Di Benedetto, Yannik Stein, Vito Bellini, and Matteo Ruffini. 2022. Low-variance estimation in the Plackett-Luce model via quasi-Monte Carlo sampling. In *SIGIR 2022 Workshop on Reaching Efficiency in Neural Information Retrieval*. https://www.amazon.science/publications/low-variance-estimation-in-the-plackett-luce-model-via-quasi-monte-carlo-sampling

[5] Alexander Buchholz, Ben London, Giuseppe Di Benedetto, and Thorsten Joachims. 2022. Off-policy evaluation for learning-to-rank via interpolating the item-position model and the position-based model. In *CONSEQUENCES+REVEAL 2022*. https://www.amazon.science/publications/off-policy-evaluation-for-learning-to-rank-via-interpolating-the-item-position-model-and-the-position-based-model

[6] Aleksandr Chuklin, Ilya Markov, and Maarten De Rijke. 2022. *Click models for web search*. Springer Nature.

[7] John E. Hopcroft and Richard M. Karp. 1973. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM J. Comput.* 2, 4 (1973), 225–231. https://doi.org/10.1137/0202019 arXiv:https://doi.org/10.1137/0202019

[8] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil) *(SIGIR '05)*. Association for Computing Machinery, New York, NY, USA, 154–161. https://doi.org/10.1145/1076034.1076063

[9] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*. 781–789.

[10] Janardhan Kulkarni, Euiwoong Lee, and Mohit Singh. 2017. Minimum birkhoff-von neumann decomposition. In *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 343–354.

[11] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-based systems* 123 (2017), 154–162.

[12] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S. Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline Evaluation of Ranking Policies with Click Models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1685–1694. https://doi.org/10.1145/3219819.3220028

[13] Ben London and Thorsten Joachims. 2022. Control variate diagnostics for detecting problems in logged bandit feedback. In *RecSys 2022 Workshop: CONSEQUENCES – Causality, Counterfactuals and Sequential Decision-Making*. https://www.amazon.science/publications/control-variate-diagnostics-for-detecting-problems-in-logged-bandit-feedback

[14] Harrie Oosterhuis. 2022. Learning-to-Rank at the Speed of Sampling: Plackett-Luce Gradient Estimation with Minimal Computational Complexity. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2266–2271. https://doi.org/10.1145/3477495.3531842

[15] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 489–498. https://doi.org/10.1145/3397271.3401102

[16] Tomas Rehorek, Ondrej Biza, Radek Bartyzal, Pavel Kordik, Ivan Povalyev, and O Podstavek. 2018. Comparing offline and online evaluation results of recommender systems. In *In Proceedings of the REVEAL workshop at RecSyS conference (RecSyS 2018)*.

[17] Matteo Ruffini, Vito Bellini, Alexander Buchholz, Giuseppe Di Benedetto, and Yannik Stein. 2022. Modeling Position Bias Ranking for Streaming Media Services. In *Companion Proceedings of the Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 72–76. https://doi.org/10.1145/3487553.3524210

[18] Lequn Wang and Thorsten Joachims. 2021. User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval* (Virtual Event, Canada) *(ICTIR '21)*. Association for Computing Machinery, New York, NY, USA, 23–41. https://doi.org/10.1145/3471158.3472260

[19] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. 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*. 610–618.

A RELATED WORK

The impact of business rules and other ranking modifications on off-policy evaluation and learning has been acknowledged in recent work, see for example [13, 16] and hence creates a clear need for addressing this issues in real world

production systems. [13] approaches this problem by detecting potential issues that come from biased propensities. Our work provides a practical solution to de-bias propensities with the aim of enabling unbiased offline evaluation.

Unbiased offline evaluation has been carefully studied over recent years (see for example [3, 9]), but mostly focuses on biases coming from user behavior, not from intentional modifications of the ranking output as our work does.

We build on work investigating the use of randomization of rankings based on Birkhoff-von-Neumann decompositions, see [18] and the use of propensities for unbiased off-policy evaluation as in [1, 5, 12, 17].

B SIMULATIONS

We generated dataset with 50,000 rankings using following scheme:

(1) each ranking has 10 rankable items encoded as one-hot vectors u_j where $j \in \{0, \ldots, 9\}$
(2) random standard normal noise is added to all vectors u_j
(3) for each item, relevance vector v is generated where items $\{v_1, v_2, v_4, v_7\}$ have all dimensions set to 1 and rest of items have all dimensions set to -1
(4) items are ranked by score computed as $u_j \cdot v_j$
(5) ranking is randomized by using a BvN decomposition of propensity matrix which keeps the item on the position assigned by ranker with probability 0.95 and with probability $\frac{0.05}{9}$ it's placed on any other position
(6) optionally, depending on the experiment, some item is pinned/moved to another position
(7) optionally, depending on the experiment, the propensity matrix is corrected using algorithm 1 or 3
(8) clicks are simulated assuming a PBM with position-bias inversely proportional to the item position k: $\frac{1}{k}$ and relevance defined as $\mathbb{1}_{u_j \cdot v_j > 0}$

Then we evaluated a policy that assigns items $[7, 0, 3, 1]$ to the top and items $[2, 4]$ to the bottom of the ranking in the given order. Rest of the items are ranked arbitrarily. Expected reward (number of clicks per ranking) is known and equal to 2. The position-bias curve was estimated using *PA-IH* algorithm from [17] using an SGD optimizer with decreasing learning rate. Results are shown in Figure 3 in Appendix D.

C VARIANTS OF THE ALGORITHM

Algorithm 2 Monte Carlo Estimation

Require: observation x_i, business rules B and the ranker from logging policy π_0^{ranker}
  $P'_{j,k} \leftarrow 0$
  for L samples **do**
    $Y \leftarrow \pi_0^{\text{ranker}}(x_i)$ ▷ Sample ranking Y for observation x_i provided by stochastic or randomized ranker
    $Y' \leftarrow B(Y) \cdot Y$ ▷ Apply business rules to ranking Y
    $P' \leftarrow P' + \frac{1}{L} Y'$ ▷ Sum occurrences of the items at their positions after applying business rules
  end for

Algorithm 3 Estimation using BvN with stochastic business rules

Require: observation x_i, set of applicable business rules B, BvN decomposition from logging policy $\Pi_1, \ldots, \Pi_M, p_1, \ldots, p_M$ and the deterministic ranker from logging policy π_0^{ranker}

```
P'_{j,k} ← 0
Y ← π_0^ranker(x_i)                ▷ Ranking for the observation x_i provided by deterministic ranker
for m ∈ [1 ... M] do               ▷ Iterating over the BvN decomposition used by the logging policy
    Y^m ← Π_m · Y                  ▷ Permute the ranking by Π_m
    for S ∈ P(B) do                ▷ where P(B) is the power set of all applicable business rules B.
        Y' ← S(Y^m)Y^m             ▷ Permute the ranking Y^m by applying the subset of business rules
        P' ← P' + p_m ℙ(S|x_i) Y'  ▷ Where ℙ(S) is a probability of applying all business rules from S simultaneously
    end for
end for
```

D ADDITIONAL RESULTS

We analyzed several additional settings of pinning (Figure 2) and provide corresponding fitted position-bias curves in Figure 3:

- Item with low/high average relevance pinned to the first/last position (rows in Figures 2 and 3)
- Columns correspond to:
  - Pinning applied with 100% probability with no correction of propensity matrix
  - Pinning applied with 95% probability with no correction of propensity matrix
  - Pinning applied with 100%, correction was made assuming that pinning was applied with 95% probability
  - Pinning applied with 95%, correction was made assuming that pinning was applied with 95% probability

We can see that assuming stochastic pinning when the actual pinning was applied with 100% probability often leads to biased estimates, highlighting the importance of having only stochastic business rules. We can also see that INTERPOL estimator is more robust than PBM estimator, as PBM estimator sometimes provides slightly biased results with high confidence. This is caused by the fact that PBM estimator is sensitive to precise estimation of position-bias curve, which is impossible due to inherent variance in the estimation of position-bias curve.

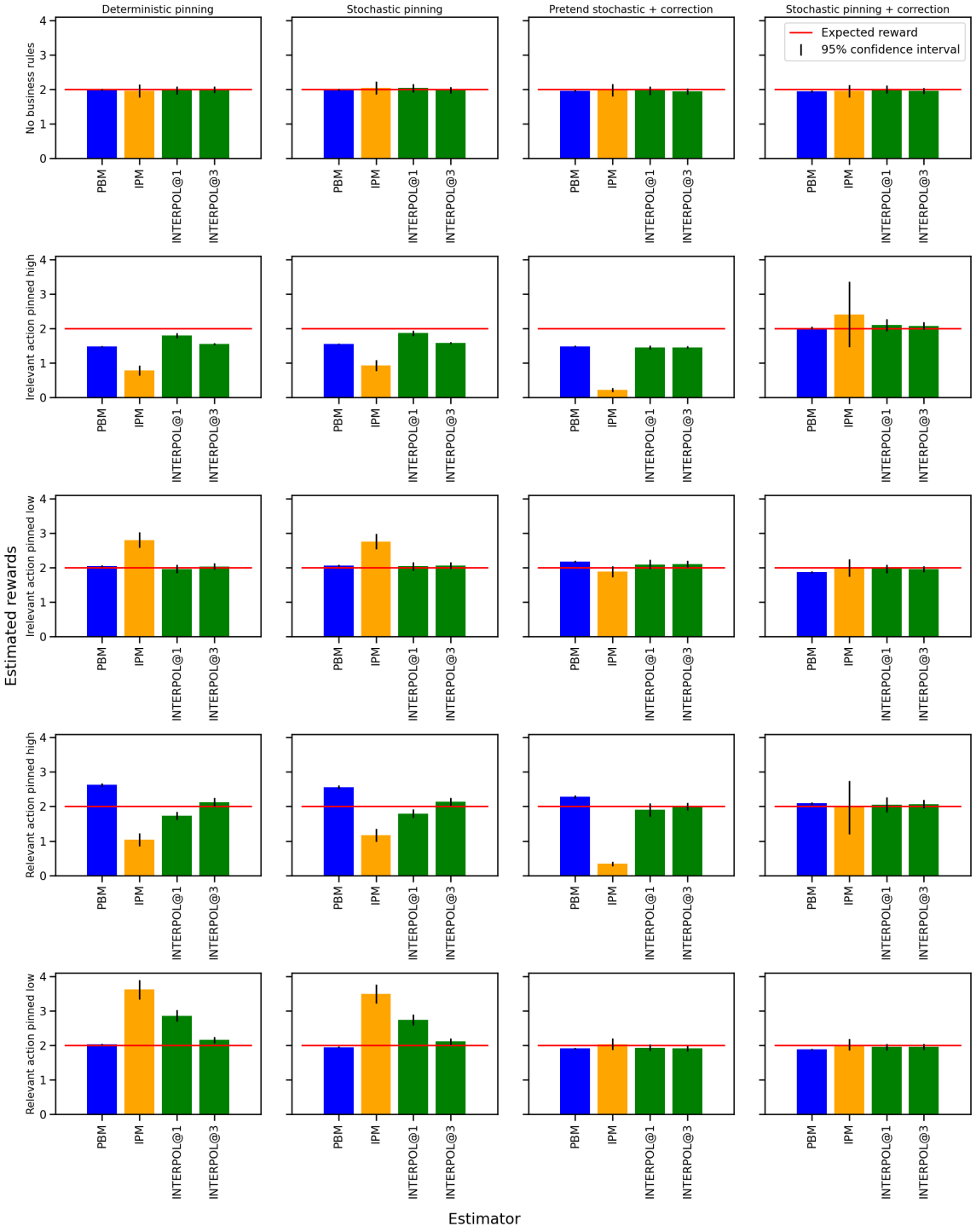

Fig. 2. Estimates for additional settings of pinning.

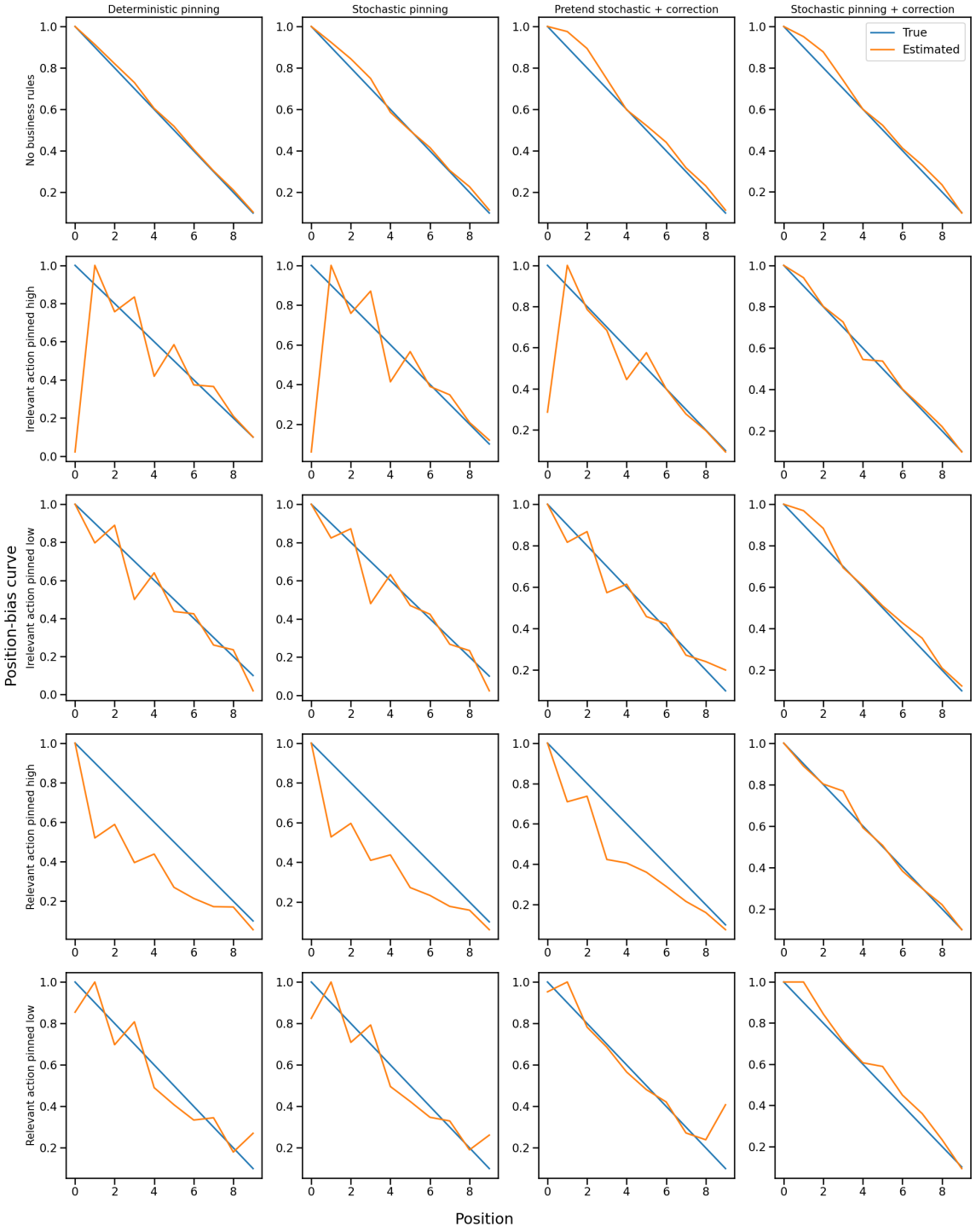

Fig. 3. Estimated position-bias curves corresponding to results in figure 2.