Sample-Free Almost-Exact Estimation of Plackett-Luce Propensities for Off-Policy Ranking Estimators

Norman Knyazev
Radboud University
Nijmegen, The Netherlands
norman.knyazev@ru.nl

Harrie Oosterhuis
Radboud University
Nijmegen, The Netherlands
harrie.oosterhuis@ru.nl

1 Introduction

Modern learning to rank (LTR) approaches often rely on historical user interaction data such as clicks due to their relative abundance compared to other types of interactions [14–16, 21]. However, user click data is also known to be affected by various forms of statistical biases, e.g. position bias [25, 32], whereby documents that were more likely to be ranked higher by the production (logging) policy are also more likely to be observed and thus interacted with by users [25]. As such, off-policy learning and evaluation methods rely on the information about the logging policy to correct for such forms of bias [6, 13, 25].

Importantly, in most cases, including under the widely-used position-based model (PBM) click-model [5, 6], this requires the knowledge of the placement probabilities of each document across every position. However, for most ranking policies the only known exact solution is to iterate over all possible rankings, summing up the probabilities of those where the document is observed at the desired rank. This, however, is generally not feasible unless the number of documents is very low [22, 23].

To our knowledge, the only commonly used alternative is to approximate these probabilities by sampling a large number of rankings and counting the occurrences of every document at each position [10, 23]. However, in practice, sampling enough rankings for a sufficiently good probability estimate may not be computationally feasible, especially if these probabilities cannot be pre-computed or need to be re-computed multiple times [3, 7, 22]. On the other hand, sampling enough rankings may be especially difficult for queries with a high number of candidate documents or where the logging policy assigns high probabilities only to a small subset of documents at each rank. In both cases many rank-document pairs may instead end up without a single relevant sample.

To address this issue, we propose a new efficient way to calculate these placement probabilities for the popular Plackett-Luce (PL) ranking model [4, 7, 22, 23, 27]. We show that under this model, due to its connection to Gumbel distribution, the placement probabilities can also be seen as the expectation of a Poisson binomial random variable over the document scores. By leveraging this interpretation and the known connection of Poisson binomial to convolutions [1, 9] along with numerical integration, we propose a novel approximation approach for estimating the probability of a document being placed at a ranking position. Importantly, we argue that despite being an approximation, our approach is effectively exact when evaluating it over a practically feasible number of points. Our experiments also confirm the significant gains in accuracy and the efficiency of our proposed method compared to the standard sampling approach. Altogether, our method opens the door for a new direction of more exact propensity estimation; and thereby, we hope that it motivates for a wider use of PL policies for off-policy ranking evaluation and optimization applications.

CONSEQUENCES '25, Prague, Czech Republic

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Background

For a query q, the main task of LTR is to select K of the candidate documents $d \in D$ and present them in a form of an ordered ranking $y = [y_1, \dots, y_K]$ [19]. The quality of a stochastic ranking policy π assigning a probability to every ranking y measured by the expected number of clicks $c \in \{0, 1\}$ can be expressed by $\Delta_{\text{LTR}}(q) = \mathbb{E}_{y\sim\pi,c}\big[\sum_{k=1}^{K} c(y_k)\big]$. In practice, this quantity is often estimated using historical user interaction data with rankings produced by production (logging) policy π of the form $\{q, y, c, \pi_0\}_{i=1}^{n}$ [16, 21]. Importantly, not all documents in the ranking y may actually be observed by the user. For instance, under the commonly used PBM, the probability that the user observes document d depends on $\Pr_\pi(d = y_k)$, i.e., the probability of policy π placing d in each position $k \in \{1, \dots, K\}$, and the probability of $O(k) \in \{0, 1\}$ – the user observing the contents of that position. Then using inverse propensity scoring (IPS) a possible unbiased off-policy estimate is [16, 25]:

$$\hat{\Delta} = \sum_{i,k=1}^{n,K} w(y_k, k)c(y_k), w(d, k) = \frac{\sum_{k=1}^{K} \Pr(O(k))\Pr_\pi(d = y_k)}{\sum_{k=1}^{K} \Pr(O(k))\Pr_{\pi_0}(d = y_k)}. \quad (1)$$

Importantly, off-policy estimation based on both the PBM and other click models [17, 24, 30, 31] relies on the availability of correct placement probabilities $\Pr_\pi(d = y_k)$ [5, 26], which are difficult to estimate for most stochastic ranking models [3, 7, 22, 29]. Under the PL ranking model, for documents with (logit) scores m_d and where $\mathbb{1}[d \notin y_{1:i-1}]$ denotes d not being placed in a partial ranking spanning positions 1 to $i-1$, the probability of full ranking y is:

$$\pi_{\text{PL}}(y) = \prod_{i=1}^{K} \pi_{\text{PL}}(y_i \mid y_{1:i-1}) = \prod_{i=1}^{K} \frac{e^{m_{y_i}} \mathbb{1}[y_i \notin y_{1:i-1}]}{\sum_{d'=1}^{|D|} e^{m_{d'}} \mathbb{1}[d' \notin y_{1:i-1}]}. \quad (2)$$

Importantly, under the PL model the placement probability of y_k depends (solely) on the previously placed documents $y_{1:k-1}$. As such, obtaining the closed-form document-position marginal probability $\Pr(d = y_k) = \sum_{y_{1:k-1}} \pi(y_{1:k-1}) \sum_{y_k} \pi(y_k \mid y_{1:k-1})$ requires enumerating over all possible rankings $y_{1:k}$.

Notably, previous works on the optimization of various PL objectives leverage the connection of the PL distribution to the Gumbel

distribution [3, 8, 20, 22]. In particular, a Gumbel(m, β) random variable with location m and spread β is defined between $[-\infty, \infty]$ with the probability density function (PDF) f and cumulative distribution function (CDF) F of Gumbel$(m_j, 1)$ respectively being:

$$\begin{aligned} f_j(x) &= f(x; m_j, 1) = e^{m_j - x - e^{m_j - x}}, \\ F_j(x) &= F(x; m_j, 1) = e^{-e^{m_j - x}}. \end{aligned} \tag{3}$$

The well-known Gumbel-max trick [12, 18], used for efficient sampling from softmax and Plackett-Luce distributions, leverages the fact that these distributions can also be represented in terms of Gumbel$(m_j, 1)$ variables centered on $|D|$ document scores m_j. The softmax sampling probability of a document $\Pr(d) = \Pr(y_1 = d)$ is then equal to the probability of taking $|D|$ samples $\forall j : G_j \sim$ Gumbel$(m_j, 1)$ and G_d being the highest [12]. Following similar intuition, Ma et al. [20] show that under the PL model, the probability that all documents from set Y are placed before all documents from a disjoint complementary set Y^C s.t. $Y \cup Y^C = D$ is:

$$\Pr(Y \prec Y^C) = \int_{-\infty}^{\infty} \prod_{i \in Y} \big(1 - F_i(x)\big) f(x; \log(\sum_{j \in Y^C} e^{m_j}), 1) dx. \tag{4}$$

Intuitively, $f(x; \log(\sum_{j \in Y^C} e^{m_j}), 1)$ represents the distribution of the highest Gumbel sample $G_j \in Y^C$ and is itself a Gumbel variable, whilst $\prod_{i \in Y}(1 - F_i(x))$ represents the probability that *all* $G_i \in Y$ exceed it. We now leverage this Gumbel-based intuition to derive our estimator of $\Pr(d = y_{k+1})$.

3 Method: Placement Propensity Estimation

We now propose a novel approach to accurately and efficiently estimate the document-position placement probabilities under the PL model. To this end, we leverage the connection between the Plackett-Luce and Gumbel distributions.

We first note that the result in Equation 4 of Ma et al. [20] can also be extended to three mutually exclusive sets: $Y_{k\setminus d}$ of size k that does not include d; its complement $Y^C_{k\setminus d}$ that similarly excludes d; and $\{d\}$ itself. Then the probability $\Pr(Y_{k\setminus d} \prec d \prec Y^C_{k\setminus d})$ that d is placed after all documents in $Y_{k\setminus d}$ and before all documents in $Y^C_{k\setminus d}$, is equal to the probability that all $G_i \in Y_{k\setminus d}$ are higher than G_d *and* that all $G_j \in Y^C_{k\setminus d}$ are lower than G_d:

$$\Pr(Y_{k\setminus d} \prec d \prec Y^C_{k\setminus d}) = \int_{-\infty}^{\infty} f_d(x) \prod_{i \in Y_{k\setminus d}} \big(1 - F_i(x)\big) \prod_{j \in Y^C_{k\setminus d}} F_j(x) dx. \tag{5}$$

A crucial observation is that $\Pr(Y_{k\setminus d} \prec d \prec Y^C_{k\setminus d})$ represents the probability of a specific partially sorted ranking $Y_{k\setminus d} \prec d \prec Y^C_{k\setminus d}$ where d is in position $k+1$. Therefore, the probability of observing any partial ranking with d in position $k+1$ is simply the sum over all possible $Y_{k\setminus d} \in \mathbb{Y}_{k\setminus d}$. That, however, is exactly the probability of d being placed in position $k+1$ under the PL distribution, i.e. its $|D| - k + 1$'th order statistic:

$$\begin{aligned} \Pr(y_{k+1} = d) &= \sum_{Y_{k\setminus d}} \Pr(Y_{k\setminus d} \prec d \prec Y^C_{k\setminus d}) \qquad (6) \\ &= \int_{-\infty}^{\infty} f_d(x) \sum_{Y_{k\setminus d}} \prod_{i \in Y_{k\setminus d}} \big(1 - F_i(x)\big) \prod_{j \in Y^C_{k\setminus d}} F_j(x) dx. \qquad (7) \end{aligned}$$

Importantly, in constrast to the closed-form solution in Section 2, $Y_{k\setminus d}$ refers to a set of documents rather than a specific ranking, and thus the relative order of its elements does not matter. In practice this means that we do not need to consider all document permutations and only their combinations.

Nevertheless, explicitly iterating over all possible sets of size $k < K$ may still be prohibitively expensive for most common ranking sizes K. To this end, we note that for a fixed value of x we can treat each $G_j > G_d$ as a success in a Bernoulli trial with its own $p_j(x) = 1 - F_j(x)$ where all trials are independent (conditional on x). The summation in Equation 6 then reflects the probability of the observed number of successes S in $|D| - 1$ such trials being k:

$$\Pr(S = k \mid d, x) = \sum_{Y_{k\setminus d}} \prod_{i \in Y_{k\setminus d}} p_i(x) \prod_{j \in Y^C_{k\setminus d}} \big(1 - p_j(x)\big). \tag{8}$$

S is thus a Poisson binomial variable whose success probabilities p may be different for each x. Combining Equations 7 and 8 and integrating over all values of $x = G_d$ then yields:

$$\Pr(y_{k+1} = d) = \int_{-\infty}^{\infty} f_d(x) \sum_{y_{1:k\setminus d}} \prod_{i \in y_{1:k\setminus d}} p_i(x) \prod_{j \in y^C_{1:k\setminus d}} \big(1 - p_j(x)\big) dx \tag{9}$$

$$= \int_{-\infty}^{\infty} f_d(x) \Pr(S = k \mid d, x) dx = \mathbb{E}_{G_d}\Big[\Pr(S = k \mid d, G_d)\Big]. \tag{10}$$

As such, $\Pr(y_{k+1} = d)$ is the expected value of the Poisson binomial probability of k documents scoring above d. This interpretation is significant, as for a given value of x the Poisson binomial distribution can be efficiently obtained without explicitly iterating over all possible sets.

In particular, as suggested by Fernandez and Williams [9], Poisson binomial distribution with trial success probabilities $p_1, \ldots, p_{|D|}$ can also be seen as a convolution $c^{1:|D|} = *_{i=1}^{|D|}[1 - p_i, p_i]$ over probability vectors $\forall [1 - p_i, p_i]$. To that end, we use the Direct Convolution (DC) algorithm of Biscarri et al. [1], who showed that the convolution of the first $k-1$ documents $c^{1:k-1}$ with $[1 - p_k, p_k]$ is $[(1-p_k) \cdot c_1^{1:k-1}] \oplus \big((1-p_k) \cdot c_{2:k}^{1:k-1} + p_k \cdot c_{1:k-1}^{1:k-1}\big) \oplus [p_k \cdot c_k^{1:k-1}]$. Here $\oplus$ denotes the concatenation of two lists and $c_{m:n}^{1:k-1}$ denotes the entries of $c^{1:k-1}$ between indices m and n. Leveraging this recursive relation and by noting that at each step we only need to keep up to $K+1$ first entries of each convolution, we can obtain the Poisson binomial probabilities for the first K ranks $c_{1:K+1}^{1:|D|}$ in $O(|D|K)$ time.

An important consideration, however, is that when simultaneously calculating $\Pr(S = k \mid d, x)$ for all $|D|$ documents in a query, each document d has to be excluded from its own convolution, as d cannot appear in either $Y_{k\setminus d}$ nor $Y^C_{k\setminus d}$. To avoid explicitly computing a size $|D| - 1$ convolution $|D|$ times, and inspired by the use of divide-and-conquer algorithms for polynomial multiplication [11], we thus propose the following scheme. We start by calculating the convolution in both directions, i.e. $c^{1:|D|}$ and $c^{|D|:1}$, whilst caching all intermediate results, e.g. $c^{1:3}$ and $c^{|D|:5}$. The convolution $c^{1:|D|\setminus d}$ that excludes d is then exactly $c^{1:d-1} * c^{|D|:d+1}$. The entry associated with k successes can then be obtained by summing over the entries of $k+1$'th anti-diagonal of $(c^{1:d-1} \otimes c^{|D|:d+1})$, where $\otimes$ denotes the outer product. We can use the above approach to simultaneously calculate $\Pr(S = k \mid d, x)$ for all D and for all $k \leq K$, resulting in the computational complexity of $O(|D|K^2)$. We note that it may also be possible

to leverage the relation $c_k^{1:|D|\setminus d} = (c_k^{1:|D|} - p_d \cdot c_{k-1}^{1:|D|\setminus d})/(1-p_d)$ to further improve the computational complexity of our approach to $O(|D|K)$. However, in practice the high numerical stability of our approach, which avoids division, may be preferred across most K.

With the ability to efficiently calculate $\Pr(S = k \mid d, x)$, we then return to the other aspect of Equation 10. In particular, $\Pr(d = y_{k+1})$ requires integrating over all possible values x that G_d can take. However, in practice almost all of the probability density of all f_d will be concentrated between $[a, b] = [\min_{m_j} + c_1, \max_{m_j} + c_2]$, where a reasonable choice of c_1 and c_2 may be some low and high quantiles of a Gumbel$(0, 1)$ distribution (s.t. $c_1 < 0 < c_2$):

$$\Pr(y_{k+1} = d) \approx \int_{a=\min_{m_j}+c_1}^{b=\max_{m_j}+c_2} f_d(x) \Pr(S = k \mid d, x) dx. \quad (11)$$

The form $\int_a^b g(x)$ is crucial, as integrals of that form can be efficiently approximated with Gauss-Legendre quadrature using the rule: $\int_{-1}^{1} g(x)dx \approx \sum_{i=1}^{N} w_i g(x_i)$. The integration points x_i and associated weights w_i are the first N roots of the N'th Legendre polynomial and can be pre-computed once and shared across the whole dataset [2]. Using this form to represent Equation 10 and applying change of variables to change the integration limits from $[-1, 1]$ to $[a, b]$ then allows us to obtain our final estimator:

$$\Pr(y_{k+1} = d) \approx \alpha \sum_{i=1}^{N} w_i f_d(\alpha x_i + \beta) \Pr(S = k \mid d, \alpha x_i + \beta), \quad (12)$$

where $\alpha = (b-a)/2$ and $\beta = (a+b)/2$. Importantly, Gauss-Legendre quadrature is highly accurate and the approximation is exact where the approximated function $g(x)$ is a polynomial of degree $2(N-1)$ or lower [28]. Whilst Equation 11 itself is not a polynomial; our observations indicate that the region spanning its values that are meaningfully above 0 can usually be well approximated using polynomials of degree < 20, suggesting that for a sufficiently great N Equation 12 is sufficient to closely approximate $P(y_k = d)$. In practice, we observe that our approximation closely resembles the true value $\Pr(y_{k+1} = d)$ and in our experiments does not significantly change beyond $N = 200$ points used to simultaneously calculate $\forall d, k : \Pr(y_{k+1} = d)$. We note that the optimal N may differ in case where m_j are spaced far apart, as in that case f_d at many points may be close to 0, leading to fewer points being available to correctly approximate the heavier regions of f_d.

The final computational complexity of our new approach is $O(N|D|K^2)$, compared to $O(D!)$ for the naive approach described in Section 2. Finally, we note that our estimator is fully differentiable w.r.t. $\forall m_j$ and can thus be used to optimize objectives such as Δ_{LTR}, NDCG or exposure without the need for manual derivation of gradient functions [22]. However, the scope of this abstract is limited to evaluating the accuracy of the estimated propensities.

4 Results: Accuracy of Estimated Propensities

We now evaluate the accuracy of our proposed propensity estimation method and contrast it with that of the sampling approach. In particular, we examine the estimation accuracy of both methods, as measured by the mean absolute error (MAE) over all document-position combinations, where the true values are approximated by the sampling-based approach with 10^8 sampled rankings. MAE is calculated over 320 simulated queries with $|D| = 200$ documents.

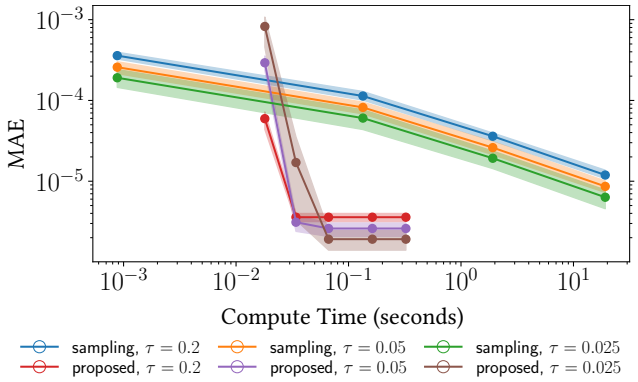

Figure 1: MAE of the sampling-based and proposed models under varied per-batch compute time. Results are averages over 320 independent queries, the difference between the best and worst runs denoted by shading.

In all cases the logits m_d are sampled from the uniform distribution, where to evaluate the impact on the performance for different policies, we further divide the logits by the temperature factor $\tau \in \{0.2, 0.05, 0.025\}$. The sampling-based method is evaluated using $N \in [10^4, \ldots, 10^7]$ samples, while our method evaluates the sum in Equation 12 over $N \in \{50, 100, 200, 500, 1000\}$ points using the lowest and the highest 0.01'st percentiles of the standard Gumbel distribution as c_1 and c_2 respectively.

Figure 1 shows the relationship between the MAE and the average time to compute the propensities for a batch of data in seconds on an $A100$ GPU, with the shaded regions denoting the lowest and the highest average error across runs. We can observe that both our proposed as well as the sampling-based method can be used to estimate document propensities, with both approaches also improving when the number of samples or points is increased. Nevertheless, our method is generally able to achieve over an order of magnitude lower average error compared to the sampling approach when using $N = 100$ points (requiring 0.02 seconds). In fact, our method achieves its optimum performance at just $N = 200$ (0.07 seconds), suggesting high data-efficiency and in line with our discussion in Section 3. Moreover, our method outperforms the sampling approach even when the latter runs for almost 19 seconds per batch, i.e. over 270 times as long. As this improvement is also consistent across different logit distributions, the above results overall suggest that the proposed method is highly suited for propensity estimation under the PL model.

5 Conclusion

Off-policy estimators in learning to rank rely on document placement probabilities to correct for various forms of statistical bias. However, obtaining a sufficiently good propensity estimate may be computationally challenging. In this work we propose a novel method for estimating these probabilities under the Plackett-Luce (PL) ranking model. We show that our approach is both highly accurate and more efficient compared to the existing sampling-based approach. Our method thus reduces the time cost of propensity estimation and opens the door for more accurate estimators under PL policies in off-policy learning to rank.

References

[1] William Biscarri, Sihai Dave Zhao, and Robert J. Brunner. 2018. A Simple and Fast Method for Computing the Poisson Binomial Distribution Function. *Computational Statistics & Data Analysis* 122 (2018), 92–100.

[2] Ignace Bogaert. 2014. Iteration-Free Computation of Gauss-Legendre Quadrature Nodes and Weights. *SIAM JOURNAL ON SCIENTIFIC COMPUTING* 36, 3 (2014), A1008–A1026. hdl:1854/LU-5683230

[3] Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. 2020. A Stochastic Treatment of Learning to Rank Scoring Functions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 61–69.

[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. *ArXiv* abs/2205.06024 (2022).

[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. arXiv:2210.09512 [cs]

[6] Alexander Buchholz, Ben London, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, and Thorsten Joachims. 2024. Counterfactual Ranking Evaluation with Flexible Click Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1200–1210.

[7] Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. 2020. Evaluating Stochastic Rankings with Expected Exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 275–284. doi:10.1145/3340531.3411962

[8] Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. 2020. *Evaluating Stochastic Rankings with Expected Exposure.* Association for Computing Machinery, New York, NY, USA, 275–284.

[9] Manuel Fernandez and Stuart Williams. 2010. Closed-Form Expression for the Poisson-Binomial Probability Density Function. *IEEE Trans. Aerospace Electron. Systems* 46, 2 (2010), 803–817.

[10] Philipp Hager, Onno Zoeter, and Maarten de Rijke. 2025. Unidentified and Confounded? Understanding Two-Tower Models for Unbiased Learning to Rank. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*. Association for Computing Machinery, 347–357.

[11] William Hart and Andrew Novocin. 2011. Practical Divide-and-Conquer Algorithms for Polynomial Arithmetic. In *Computer Algebra in Scientific Computing*, Vladimir P. Gerdt, Wolfram Koepf, Ernst W. Mayr, and Evgenii V. Vorozhtsov (Eds.). Springer, 200–214.

[12] Iris A. M. Huijben, Wouter Kool, Max B. Paulus, and Ruud J. G. van Sloun. 2023. A Review of the Gumbel-max Trick and Its Extensions for Discrete Stochasticity in Machine Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2023), 1353–1371.

[13] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When People Change Their Mind: Off-Policy Evaluation in Non-stationary Recommendation Environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 447–455.

[14] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 133–142.

[15] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data as Implicit Feedback. In *ACM SIGIR Forum*, Vol. 51. Acm New York, NY, USA, 4–11.

[16] 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.

[17] Haruka Kiyohara, Yuta Saito, Tatsuya Matsuhiro, Yusuke Narita, Nobuyuki Shimizu, and Yasuo Yamamoto. 2022. Doubly Robust Off-Policy Evaluation for Ranking Policies under the Cascade Behavior Model. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 487–497. doi:10.1145/3488560.3498380

[18] Wouter Kool, Herke Van Hoof, and Max Welling. 2019. Stochastic Beams and Where To Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 3499–3508.

[19] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[20] Jiaqi Ma, Xinyang Yi, Weijing Tang, Zhe Zhao, Lichan Hong, Ed Chi, and Qiaozhu Mei. 2021. Learning-to-Rank with Partitioned Preference: Fast Estimation for the Plackett-Luce Model. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. PMLR, 928–936.

[21] Harrie Oosterhuis. 2020. *Learning from User Interactions with Rankings: A Unification of the Field.* Ph. D. Dissertation. Informatics Institute, University of Amsterdam.

[22] Harrie Oosterhuis. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[23] 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*. ACM, 2266–2271.

[24] Harrie Oosterhuis. 2023. Doubly Robust Estimation for Correcting Position Bias in Click Feedback for Unbiased Learning to Rank. *ACM Trans. Inf. Syst.* 41, 3, Article 61 (Feb. 2023), 33 pages. doi:10.1145/3569453

[25] Harrie Oosterhuis and Maarten de Rijke. 2020. Policy-Aware Unbiased Learning to Rank for Top-k Rankings. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 489–498.

[26] Harrie Oosterhuis and Maarten de Rijke. 2021. Unifying Online and Counterfactual Learning to Rank. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM'21)*. ACM.

[27] Robin L. Plackett. 1975. The Analysis of Permutations. *Journal of The Royal Statistical Society Series C-applied Statistics* 24 (1975), 193–202.

[28] Lloyd N. Trefethen. 2022. Exactness of Quadrature Formulas. *SIAM Rev.* 64, 1 (2022), 132–150.

[29] Aleksei Ustimenko and Liudmila Prokhorenkova. 2020. StochasticRank: Global Optimization of Scale-Free Discrete Functions. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 9669–9679.

[30] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade Model-Based Propensity Estimation for Counterfactual Learning to Rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2089–2092.

[31] Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

[32] 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*. ACM, 610–618.