Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness

Harrie Oosterhuis

July 12, 2021

Radboud University
harrie.oosterhuis@ru.nl
https://twitter.com/HarrieOos

Goal of this work:

- Optimize a **Plackett-Luce** (PL) model for **relevance** or **fairness** ranking metrics,
- with an unbiased method (no heuristic or bounding),
- in a **computationally efficient** way (avoid combinatorial problems).

Contribution: PL-Rank

- A novel **sampling-based** method for quickly estimating PL gradients.
- Derivation to prove the estimation is unbiased.

Motivation

Traditionally a ranking **model** m tries to **score** items $d \in D$ in **order of relevance**:

$$d \succ_{\text{relevance}} d' \rightarrow m(d) > m(d').$$

Traditionally a ranking **model** m tries to **score** items $d \in D$ in **order of relevance**:

$$d \succ_{\text{relevance}} d' \rightarrow m(d) > m(d').$$

In recent years, **probabilistic ranking models** have been argued for:

Fairness (Singh and Joachims, 2019; Diaz et al., 2020)

- a deterministic ranking will give **most attention to a single item**, even if there are (almost) **equally relevant** items.
- A stochastic ranking model can more **fairly distribute** exposure over items.

Traditionally a ranking **model** m tries to **score** items $d \in D$ in **order of relevance**:

$$d \succ_{\text{relevance}} d' \rightarrow m(d) > m(d').$$

In recent years, **probabilistic ranking models** have been argued for:

Fairness (Singh and Joachims, 2019; Diaz et al., 2020)

- a deterministic ranking will give **most attention to a single item**, even if there are (almost) **equally relevant** items.
- A stochastic ranking model can more **fairly distribute** exposure over items.

Exploration (Hofmann et al., 2011; Oosterhuis and de Rijke, 2021)

- when learning from user clicks, a stochastic ranking model can **try various rankings** according to its uncertainty.

For any ranking y, an **arbitrary ranking metric** uses the weights per rank θ_k, the **relevance** of the items $P(R = 1 \mid q, d) = \rho_d$, and the **policy** π with the probability of a **ranking** $\pi(y \mid q)$:

$$\mathcal{R}(q) = \sum_{y \in \pi} \pi(y \mid q) \sum_{k=1}^{K} \theta_k P(R = 1 \mid q, y_k) = \sum_{y \in \pi} \pi(y) \sum_{k=1}^{K} \theta_k \rho_{y_k} = \mathbb{E}_y \left[\sum_{k=1}^{K} \theta_k \rho_{y_k} \right].$$

This is taken in expectation over a query distribution:

$$\mathcal{R} = \mathbb{E}_q[\mathcal{R}(q)] = \sum_{q \in \mathcal{Q}} P(q) \mathcal{R}(q).$$

This description applies to **well-known metrics**: **precision@k, recall@k, DCG, ARP**.

Background: Plackett-Luce Models

A Plackett-Luce model (Plackett, 1975; Luce, 2012) assumes the **probability of selecting** an item d is determined by the value of it compared to the **sum of values** over all items:

$$P(d \mid D) = \frac{\text{value of item } d}{\sum_{d' \in D} \text{value of item } d'}.$$

A Plackett-Luce model (Plackett, 1975; Luce, 2012) assumes the **probability of selecting** an item d is determined by the value of it compared to the **sum of values** over all items:

$$P(d \mid D) = \frac{\text{value of item } d}{\sum_{d' \in D} \text{value of item } d'}.$$

A **SoftMax** function is an instance of a Plackett-Luce model, where the exponential function ensures **positive non-zero** values:

$$P(d \mid D) = \frac{e^{m(d)}}{\sum_{d' \in D} e^{m(d')}}.$$

A Plackett-Luce ranking model is repeatedly applied to the **unplaced items**:

$$\pi(d \mid y_{1:k}, D) = \underbrace{\overbrace{\frac{\mathbb{1}[d \notin y_{1:k}]e^{m(d)}}{\sum_{d' \in D \setminus y_{1:k}} e^{m(d')}}}^{\text{item score if not placed}}}_{\text{sum of all unplaced item scores}}.$$

A Plackett-Luce ranking model is repeatedly applied to the **unplaced items**:

$$\pi(d \mid y_{1:k}, D) = \frac{\overbrace{\mathbb{1}[d \notin y_{1:k}]e^{m(d)}}^{\text{item score if not placed}}}{\underbrace{\sum_{d' \in D \setminus y_{1:k}} e^{m(d')}}_{\text{sum of all unplaced item scores}}}.$$

The probability of a ranking is the product over each item placement:

$$\pi(y) = \prod_{k=1}^{K} \pi(y_k \mid y_{1:k-1}, D).$$

We can sample from a Plackett-Luce ranking model by sampling Gumbel Noise: $\zeta_d \sim$ *Gumbel*, and sorting according to $m(d) + \zeta_d$ (Bruch et al., 2020).

The prevalent approach in existing work (Singh and Joachims, 2019; Bruch et al., 2020) uses **policy-gradients** with the log-trick (Williams, 1992):

$$\frac{\delta}{\delta m}\pi(y) = \pi(y)\left[\frac{\delta}{\delta m}\log(\pi(y))\right].$$

Given N samples from π: $y^{(i)} \sim \pi$, the gradient can be **unbiasedly** estimated:

$$\frac{\delta}{\delta m}\mathcal{R}(q) \approx \frac{1}{N}\sum_{i=1}^{N} \underbrace{\left[\frac{\delta}{\delta m}\log(\pi(y^{(i)}))\right]}_{\text{gradient w.r.t. log prob. of full ranking}} \underbrace{\left(\sum_{k=1}^{K}\theta_k \rho_{y_k^{(i)}}\right)}_{\text{observed reward}}.$$

Method: PL-Rank

A **reward before** rank k should **not influence** the probabilities of the ranking **after** k:

$$\mathcal{R}(q) = \sum_{y \in \pi} \pi(y) \sum_{k=1}^{K} \theta_k \rho_{y_k} = \sum_{k=1}^{K} \theta_k \sum_{y \in \pi} \pi(y) \rho_{y_k} = \sum_{k=1}^{K} \theta_k \sum_{y_{1:k} \in \pi} \pi(y_{1:k}) \rho_{y_k}.$$

A **reward before** rank k should **not influence** the probabilities of the ranking **after** k:

$$\mathcal{R}(q) = \sum_{y \in \pi} \pi(y) \sum_{k=1}^{K} \theta_k \rho_{y_k} = \sum_{k=1}^{K} \theta_k \sum_{y \in \pi} \pi(y) \rho_{y_k} = \sum_{k=1}^{K} \theta_k \sum_{y_{1:k} \in \pi} \pi(y_{1:k}) \rho_{y_k}.$$

Given N samples from π: $y^{(i)} \sim \pi$, the gradient can be **unbiasedly** estimated:

$$\frac{\delta}{\delta m} \mathcal{R}(q) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \underbrace{\left[\frac{\delta}{\delta m} \log(\pi(y_k^{(i)} | y_{1:k-1}^{(i)})) \right]}_{\text{log prob. of item placement at } k} \underbrace{\sum_{x=k}^{K} \theta_x \rho_{y_x^{(i)}}}_{\text{reward received after } k}.$$

Using the fact that π is a Plackett-Luce model, we can estimate the gradient using:

$$\frac{\delta}{\delta m}\mathcal{R}(q) \approx \sum_{d \in D} \overbrace{\left[\frac{\delta}{\delta m} m(d)\right]}^{\text{grad. w.r.t. score}} \frac{1}{N} \sum_{i=1}^{N} \Bigg(\overbrace{\left(\sum_{k=\text{rank}(d, y^{(i)})}^{K} \theta_k \rho_{y_k^{(i)}} \right)}^{\text{reward following placement}} \\ - \underbrace{\sum_{k=1}^{\text{rank}(d, y^{(i)})} \pi(d \mid y_{1:k-1}^{(i)}) \left(\sum_{x=k}^{K} \theta_x \rho_{y_x^{(i)}} \right)}_{\text{risk imposed by placement probability}} \Bigg).$$

Given N samples, this can be computed in $\mathcal{O}(N \cdot K \cdot D)$.

Method: PL-Rank-1

Using the fact that π is a Plackett-Luce model, we can estimate the gradient using:

$$\frac{\delta}{\delta m}\mathcal{R}(q) \approx \sum_{d \in D} \overbrace{\left[\frac{\delta}{\delta m} m(d)\right]}^{\text{grad. w.r.t. score}} \frac{1}{N} \sum_{i=1}^{N} \Bigg(\overbrace{\left(\sum_{k=\text{rank}(d, y^{(i)})}^{K} \theta_k \rho_{y_k^{(i)}} \right)}^{\text{reward following placement}}$$
$$- \underbrace{\sum_{k=1}^{\text{rank}(d,y^{(i)})} \pi(d \mid y_{1:k-1}^{(i)}) \left(\sum_{x=k}^{K} \theta_x \rho_{y_x^{(i)}} \right)}_{\text{risk imposed by placement probability}} \Bigg).$$

Given N samples, this can be computed in $\mathcal{O}(N \cdot K \cdot D)$.

Flaw: items that are **not** in the top-K of any of the N **sampled** rankings will **always** have a **negative** gradient.

We can avoid the flaw while maintaining the $\mathcal{O}(N \cdot K \cdot D)$ complexity:

$$\frac{\delta}{\delta m}\mathcal{R}(q) \approx \sum_{d \in D} \overbrace{\left[\frac{\delta}{\delta m} m(d)\right]}^{\text{grad. w.r.t. score}} \frac{1}{N}\sum_{i=1}^{N} \overbrace{\left(\sum_{k=\text{rank}(d,y^{(i)})+1}^{K} \theta_k \rho_{y_k^{(i)}}\right)}^{\text{future reward after placement}}$$

$$+ \underbrace{\sum_{k=1}^{\text{rank}(d,y^{(i)})} \pi(d \mid y_{1:k-1}^{(i)}) \left(\theta_k \rho_d - \sum_{x=k}^{K} \theta_x \rho_{y_x^{(i)}}\right)}_{\text{expected direct reward minus the risk of placement}}.$$

Fairness in exposure generally use rank-based exposure:

$$\mathcal{E}(q, d) = \mathbb{E}_y\left[\sum_{k=1}^{K} \theta_k \mathbb{1}[y_k = d]\right] = \sum_{y \in \pi} \pi(y) \sum_{k=1}^{K} \theta_k \mathbb{1}[y_k = d].$$

Fairness in exposure generally use rank-based exposure:

$$\mathcal{E}(q,d) = \mathbb{E}_y\left[\sum_{k=1}^{K} \theta_k \mathbb{1}[y_k = d]\right] = \sum_{y\in\pi} \pi(y) \sum_{k=1}^{K} \theta_k \mathbb{1}[y_k = d].$$

In this paper, we use a (novel) **pairwise disparity-based fairness** metric:

$$\mathcal{F}(q) = \sum_{d_1\in D} \sum_{d_2\in D\setminus d_1} \left(\mathcal{E}(q,d_1)\rho_{d_2} - \mathcal{E}(q,d_2)\rho_{d_1}\right)^2.$$

Fairness in exposure generally use rank-based exposure:

$$\mathcal{E}(q,d) = \mathbb{E}_y\left[\sum_{k=1}^{K} \theta_k \mathbb{1}[y_k = d]\right] = \sum_{y\in\pi} \pi(y) \sum_{k=1}^{K} \theta_k \mathbb{1}[y_k = d].$$

In this paper, we use a (novel) **pairwise disparity-based fairness** metric:

$$\mathcal{F}(q) = \sum_{d_1\in D} \sum_{d_2\in D\setminus d_1} \left(\mathcal{E}(q,d_1)\rho_{d_2} - \mathcal{E}(q,d_2)\rho_{d_1}\right)^2.$$

PL-Rank can be applied to any **rank-based exposure metric** where:

$$\frac{\delta \mathcal{F}(q)}{\delta m} = \sum_{d\in D} \frac{\delta \mathcal{F}(q)}{\delta \mathcal{E}(q,d)} \frac{\delta \mathcal{E}(q,d)}{\delta m(d)}.$$

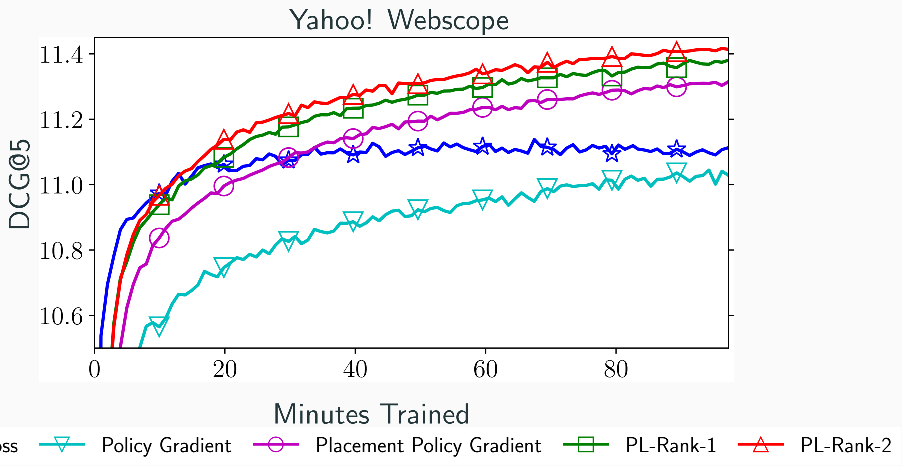
Fairness in exposure generally use rank-based exposure:

$$\mathcal{E}(q,d) = \mathbb{E}_y\left[\sum_{k=1}^{K}\theta_k \mathbb{1}[y_k = d]\right] = \sum_{y\in\pi}\pi(y)\sum_{k=1}^{K}\theta_k \mathbb{1}[y_k = d].$$

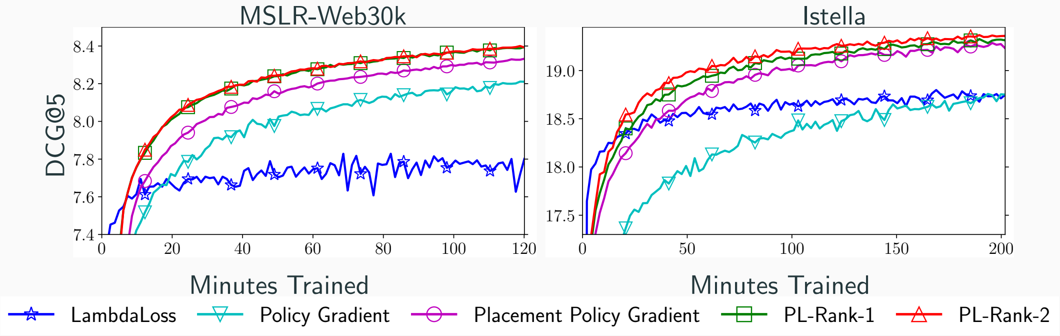
PL-Rank can be used to optimize a rank-based exposure metric $\mathcal{F}$:

$$\begin{aligned}\frac{\delta}{\delta m}\mathcal{F}(q) = \sum_{d\in D}\left[\frac{\delta}{\delta m}m(d)\right]\mathbb{E}_y\Bigg[&\left(\sum_{k=\text{rank}(d,y)+1}^{K}\theta_k\left[\frac{\delta\mathcal{F}(q)}{\delta\mathcal{E}(q,y_k)}\right]\right)\\ &+\sum_{k=1}^{\text{rank}(d,y)}\pi(d\,|\,y_{1:k-1})\left(\theta_k\left[\frac{\delta\mathcal{F}(q)}{\delta\mathcal{E}(q,d)}\right]-\sum_{x=k}^{K}\theta_x\left[\frac{\delta\mathcal{F}(q)}{\delta\mathcal{E}(q,y_x)}\right]\right)\Bigg].\end{aligned}$$

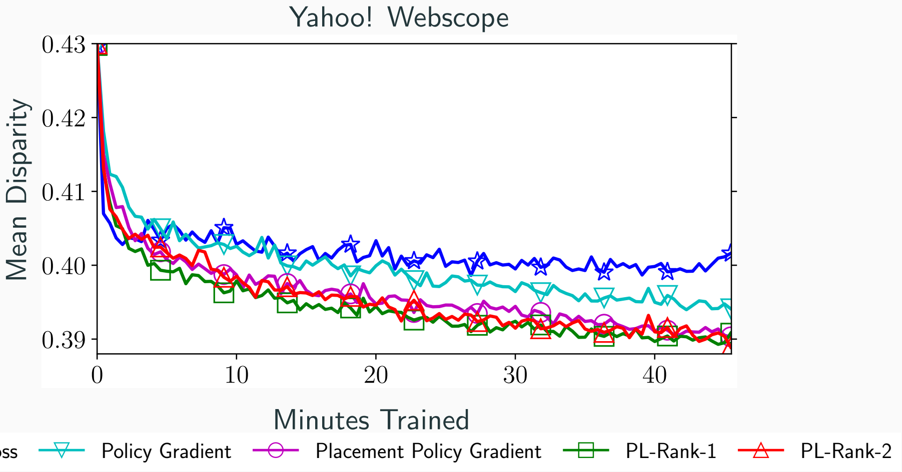
Experimental Results


Yahoo! Webscope
DCG@5
11.4
11.2
11.0
10.8
10.6
0
20
40
60
80
Minutes Trained
LambdaLoss
Policy Gradient
Placement Policy Gradient
PL-Rank-1
PL-Rank-2

Results on All Datasets



Results: Exposure Fairness



Conclusion

PL-Rank: a novel LTR method for Plackett-Luce models:

- **unbiased sample-based** gradient estimation (no heuristic or bounding),
- **computationally efficient** (avoids combinatorial problems).
- applicable to **relevance** and **fairness** ranking metrics.

Continue our work: `https://github.com/HarrieO/2021-SIGIR-plackett-luce`

Future Work: StochasticRank

The **StochasticRank** algorithm (Ustimenko and Prokhorenkova, 2020) uses sampled noise to **stochastically smooth** a ranking function.

This algorithm has strong theoretical properties and could also be applied to Plackett-Luce models with comparable computational complexity.

Very promising direction for finding computationally efficient, effective and broadly applicable LTR.

A. Ustimenko and L. Prokhorenkova. Stochasticrank: Global optimization of scale-free discrete functions. In International Conference on Machine Learning, pages 9669–9679. PMLR, 2020.

S. Bruch, S. Han, M. Bendersky, and M. Najork. A stochastic treatment of learning to rank scoring functions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 61–69, 2020.

F. Diaz, B. Mitra, M. D. Ekstrand, A. J. Biega, and B. Carterette. *Evaluating Stochastic Rankings with Expected Exposure*, page 275–284. Association for Computing Machinery, New York, NY, USA, 2020.

K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM*, pages 249–258. ACM, 2011.

R. D. Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

H. Oosterhuis and M. de Rijke. Unifying online and counterfactual learning to rank. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM'21)*. ACM, 2021.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

A. Singh and T. Joachims. Policy learning for fairness in ranking. In *Advances in Neural Information Processing Systems*, pages 5426–5436, 2019.

A. Ustimenko and L. Prokhorenkova. Stochasticrank: Global optimization of scale-free discrete functions. In *International Conference on Machine Learning*, pages 9669–9679. PMLR, 2020.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.