

Learning-to-Rank at the Speed of Sampling: Plackett-Luce Gradient Estimation With Minimal Computational Complexity

Harrie Oosterhuis

July 12, 2022

Radboud University

harrie.oosterhuis@ru.nl

<https://twitter.com/HarrieOos>



Contribution: PL-Rank-3

- A novel algorithm for estimating **Plackett-Luce** (PL) ranking model gradients,
- with **minimal computational complexity**: bounded by underlying sorting algorithm.

PL-Rank-2 algorithm approximates the following gradient with a **computational complexity** of $\mathcal{O}(N \cdot D \cdot K)$.

$$\begin{aligned}
 \frac{\delta}{\delta f} \mathcal{R}(q) \approx & \frac{1}{N} \sum_{d \in \mathcal{D}} \overbrace{\left[\frac{\delta}{\delta f} f(d) \right]}^{\text{grad. w.r.t. score}} \sum_{i=1}^N \overbrace{\left(\sum_{k=\text{rank}(d, y^{(i)})+1}^K \theta_k \rho_{y_k^{(i)}} \right)}^{\text{future reward after placement}} \\
 & + \underbrace{\sum_{k=1}^{\text{rank}(d, y^{(i)})} \pi(d \mid y_{1:k-1}^{(i)}) \left(\theta_k \rho_d - \sum_{x=k}^K \theta_x \rho_{y_x^{(i)}} \right)}_{\text{expected direct reward minus the risk of placement}}.
 \end{aligned}$$



Novel PL-Rank-3 computes the **same approximation** with $\mathcal{O}(N \cdot (D + K))$ given N sampled rankings.

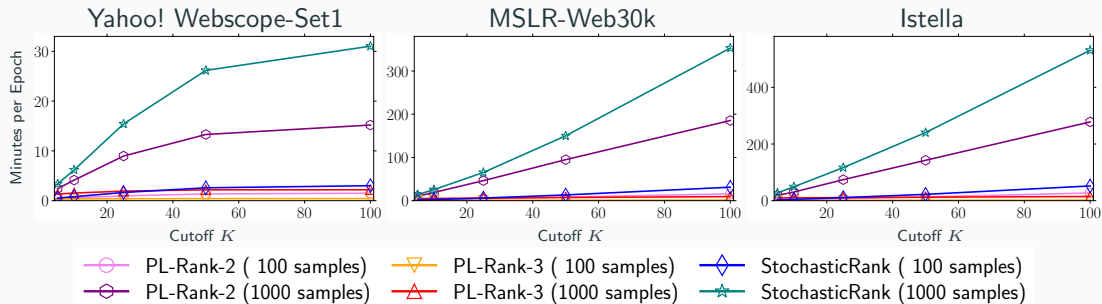
Using the following property:

$$\sum_{k=1}^{\text{rank}(d,y)} \pi(d \mid y_{1:k-1}) = e^{f(d)} \left(\sum_{k=1}^{\text{rank}(d,y)} \frac{1}{\sum_{d' \in D \setminus y_{1:k-1}} e^{f(d')}} \right).$$

We define three **new vectors** of size K : PR_y , DR_y and RI_y ; which enables:

$$\frac{\delta \mathcal{R}(q)}{\delta f} = \sum_{d \in \mathcal{D}} \left[\frac{\delta}{\delta f} f(d) \right] \mathbb{E}_y \left[PR_{y,d} + e^{f(d)} (\rho_d DR_{y,d} - RI_{y,d}) \right].$$

Given a ranking y , PR_y , DR_y and RI_y are pre-computed in **linear time**: $\mathcal{O}(K)$.



<i>method name</i>	<i>pairwise</i>	<i>ranking-based</i>	<i>metric-based</i>	<i>sample approximation</i>	<i>rank-based exposure</i>	<i>computational complexity</i>	<i>notes</i>
Pointwise						D	not an LTR loss
SoftMax Cross-Entropy						D	
Pairwise	✓					D^{2*}	memory efficient
Listwise/ListMLE		✓				DK	
SoftRank	✓	?	✓			D^3	
ApproxNDCG	✓	?	✓			D^2	proven bound
LambdaRank/Loss	✓	✓	✓			$D^{2*} + D \log(D)$	proven bound
StochasticRank	?	✓	✓	✓	✓	DK	policy-gradient
PL-Rank-1/2		✓	✓	✓	✓	DK	policy-gradient
PL-Rank-3 (ours)		✓	✓	✓	✓	$D + K \log(D)$	policy-gradient

Paper, slides, video and code:

<https://harrieo.github.io/publication/2022-sigir-short>