Learning-to-Rank at the Speed of Sampling:

Plackett-Luce Gradient Estimation With Minimal Computational Complexity

Harrie Oosterhuis

Radboud University Nijmegen

Introduction

Plackett-Luce (PL) ranking models provide optimizable probability distributions over rankings.
PL-Rank-2, the most efficient existing method for optimizing PL models, approximates gradients based on N samples with complexity of $\mathcal{O}(N \cdot D \cdot K)$ where D is the number of items to rank and K the ranking size.

- We introduce a novel algorithm that computes the same gradient approximation in $\mathcal{O}(N \cdot (D + K \cdot \log(D)))$.

Background: PL-Rank-2

For a ranking model π, metric-rank-weights θ and item relevances ρ, we wish to maximize the following reward:

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

where the ranking probabilities $\pi(y)$ are from a PL model:

$$\pi(y) = \prod_{d \in y} \pi(d \,|\, y_{1:k-1}, D), \; \pi(d \,|\, y_{1:k-1}, D) = \frac{e^{f(d)} \mathbb{1}[d \not\in y_{1:k-1}]}{\sum_{d' \in D \setminus y_{1:k-1}} e^{f(d')}}.$$

Oosterhuis (2021) proved the following equality:

$$\frac{\delta \mathcal{R}(q)}{\delta f(d)} = \mathbb{E}_y \Bigg[\Bigg(\sum_{k=\text{rank}(d,y)+1}^{K} \theta_k \rho_{y_k} \Bigg) + \sum_{k=1}^{\text{rank}(d,y)} \pi(d \mid y_{1:k-1}) \Bigg(\theta_k \rho_d - \sum_{x=k}^{K} \theta_x \rho_{y_x} \Bigg) \Bigg],$$

and the PL-Rank-2 algorithm to approximate it in $\mathcal{O}(N \cdot D \cdot K)$.

Method: PL-Rank-3

We define new vectors that can be computed in $\mathcal{O}(N \cdot K)$:

$$PR_{y,i} = \sum_{k=i}^{\min(i,K)} \theta_k \rho_{y_k}, \qquad PR_{y,d} = PR_{y,rank(d,y)+1},$$

$$RI_{y,i} = \sum_{k=1}^{\min(i,K)} \frac{PR_{y,k}}{\sum_{d' \in D \setminus y_{1:k-1}} e^{f(d')}}, \qquad RI_{y,d} = RI_{y,\text{rank}(d,y)},$$

$$DR_{y,i} = \sum_{k=1}^{\min(i,K)} \frac{\theta_k}{\sum_{d' \in D \setminus y_{1:k-1}} e^{f(d')}}, \qquad DR_{y,d} = DR_{y,\text{rank}(d,y)}.$$

Given the values of these vectors, the gradient can be simplified:

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

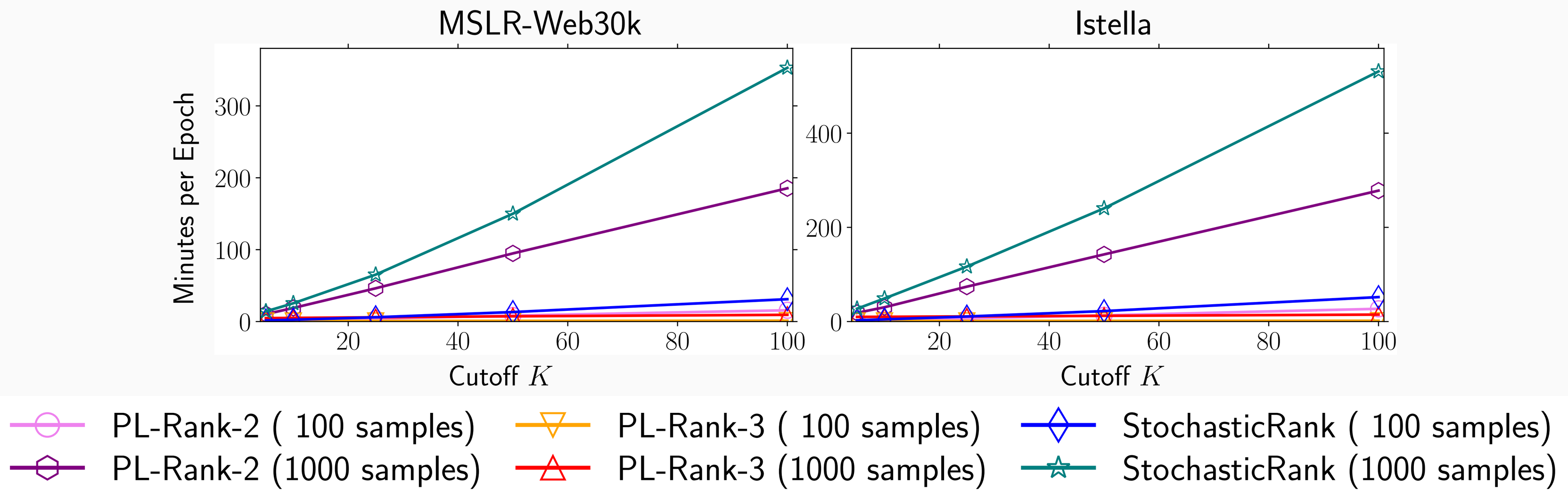
By first pre-computing the vector values, gradient estimation can be done in $\mathcal{O}(N \cdot (D + K))$ given N sampled rankings.
If we take into account the sampling procedure, the final complexity becomes: $\mathcal{O}(N \cdot (D + K \cdot \log(D)))$, which is the complexity of the underlying sorting procedure.

Experimental Setup

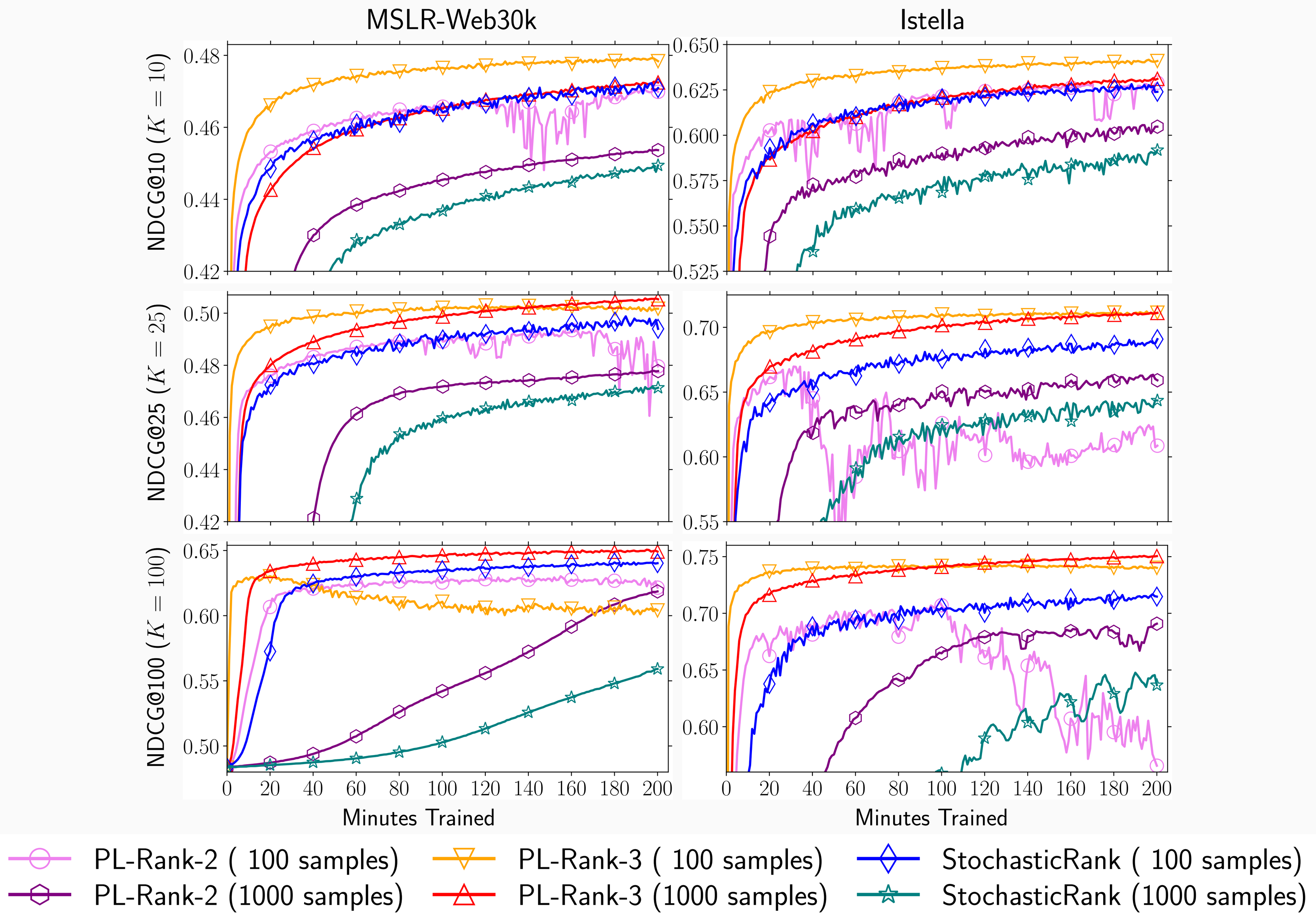
Our experiments optimize the $DCG@K$ of neural ranking models on the Yahoo! Webscope-Set1 (Chapelle and Chang, 2011), MSLR-Web30k (Qin and Liu, 2013) and Istella (Dato et al., 2016) datasets. We compare PL-Rank-3 with PL-Rank-2 (Oosterhuis, 2021) and StochasticRank (Ustimenko and Prokhorenkova, 2020) implemented in Numpy and Tensorflow. Experiments were all performed on AMD EPYC™ 7H12 CPUs.

Results: Mean Minutes per Epoch

	method	N	$K=5$	$K=10$	$K=25$	$K=50$	$K=100$
Yahoo!	Stoc.Rank	100	0.52	0.80	1.64	2.58	3.01
		1000	3.38	6.18	15.41	26.17	31.03
	PL-Rank-2	100	0.43	0.58	0.96	1.35	1.50
		1000	2.41	4.13	8.96	13.30	15.19
	PL-Rank-3	100	**0.33**	**0.34**	**0.38**	**0.40**	**0.40**
		1000	**1.33**	**1.53**	**1.92**	**2.18**	**2.19**
MSLR	Stoc.Rank	100	1.55	2.53	6.01	13.23	31.12
		1000	14.15	25.46	65.25	150.00	353.01
	PL-Rank-2	100	1.19	1.75	4.04	8.36	15.71
		1000	10.34	18.91	46.27	94.81	185.42
	PL-Rank-3	100	**0.74**	**0.77**	**0.86**	**1.00**	**1.20**
		1000	**4.77**	**5.09**	**5.93**	**7.31**	**9.37**
Istella	Stoc.Rank	100	2.79	4.61	10.75	22.25	51.94
		1000	27.34	49.07	116.73	240.40	531.09
	PL-Rank-2	100	1.96	2.87	6.72	13.25	27.07
		1000	18.49	30.37	74.09	142.55	278.01
	PL-Rank-3	100	**1.24**	**1.26**	**1.34**	**1.46**	**1.72**
		1000	**9.93**	**10.14**	**10.87**	**12.20**	**14.75**



Results: DCG/Training-Time Learning Curves



Conclusion

PL-Rank-3 is the first algorithm for PL-ranking optimization with the same computational complexity as sorting algorithms.

Resources: `https://github.com/HarrieO/2022-SIGIR-plackett-luce`

References

O. Chapelle and Y. Chang. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research*, 14:1–24, 2011.

D. Dato, C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, N. Tonellotto, and R. Venturini. Fast ranking with additive ensembles of oblivious and non-oblivious regression trees. *ACM Transactions on Information Systems (TOIS)*, 35(2):Article 15, 2016.

H. Oosterhuis. Computationally efficient optimization of plackett-luce ranking models for relevance and fairness. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 1023–1032. ACM, 2021.

T. Qin and T.-Y. Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

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

Acknowledgements

This work was partially supported by the Google Research Scholar Program and made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-1748. All content represents the opinion of the author, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Radboud University