CS6501 IR Paper Presentation

Differentiable Unbiased Online Learning to Rank

Harrie Oosterhis, Maarten de Rijke

Shengyuan Piao, Zice Wei, Peiyi Yang, Fangzhou Xu

April 8th, 2021

Outline

- Introduction
- Related Work
- Method
- Experiments
- Results and Analysis

Introduction

- **Online Learning to Rank** (OLTR) Method
  - **Pro**: Does not require an existing ranker of decent quality
  - **Con**: Not good at handling bias and noise
  - Paper proposed **Pairwise Differentiable Gradient Descent** (PDGD) Method
    - Unbiased OLTR methods
    - Does not rely on sampling models for exploration, but models ranking as probability distributions over documents.
- Answer Three Research Questions:
  - **RQ1**: Does Using PDGD result in significantly better performance than Multileave Gradient Descent (MGD)?
  - **RQ2**: Is the gradient estimation of PDGD unbiased?
  - **RQ3**: Is PDGD capable of effectively optimizing different types of ranking models?

Related Work

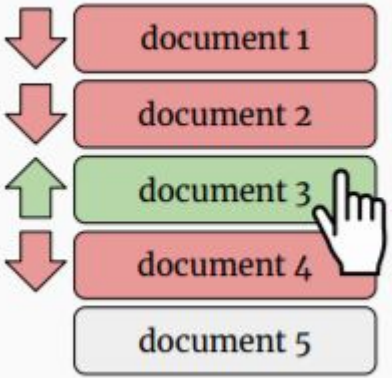
- **Dueling Bandit Gradient Descent** (DBGD)
  - OLTR method
  - Interleaving
- **Multileave Gradient Descent** (MGD)
  - Vastly speeds up the learning rate of DBGD (better user experience)
  - However, huge computational costs, large number of ranker have to be applied
- **Pairwise Loss Method**
  - It injects the ranking from the current model with randomly sampled documents.
  - After each impression, a pairwise loss is constructed from inferred preferences between documents.

Method - Pairwise update is biased

(Pairwise update: rank the relevance of 2 docs according to relevance and user reaction(=clicking))

Some preferences are more likely to be inferred due to position bias (e.g. people only look at top 10 returned docs)

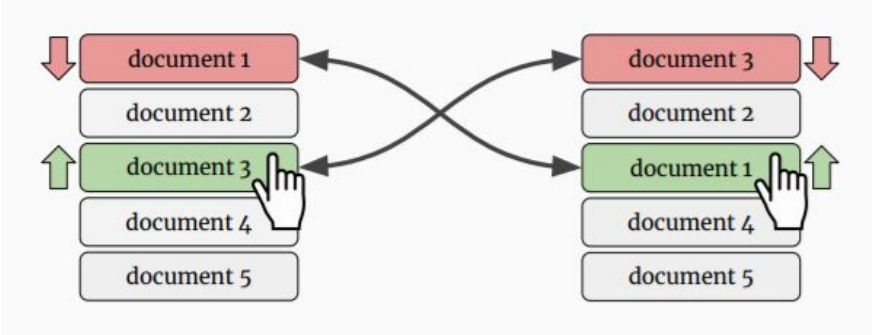
Result: Model **learns biased preferrences**



Method - Capturing the bias with reversed pair

Let R* (di , dj , R) be R but with the positions of di and dj **swapped**:

pref inferred from R:
d3 >click d1



pref inferred from R*:
d1 >click d3

Intuition:
Ideally, for a preference di>dj inferred from ranking R, and if both documents are **equally relevant**, then the opposite preference dj>di is **equally likely** to be inferred from R* (di , dj , R).

Method - Unbiasing pairwise update

The **ratio** between the probability of the ranking R and the reversed pair ranking R* **indicates the bias between the two directions**:

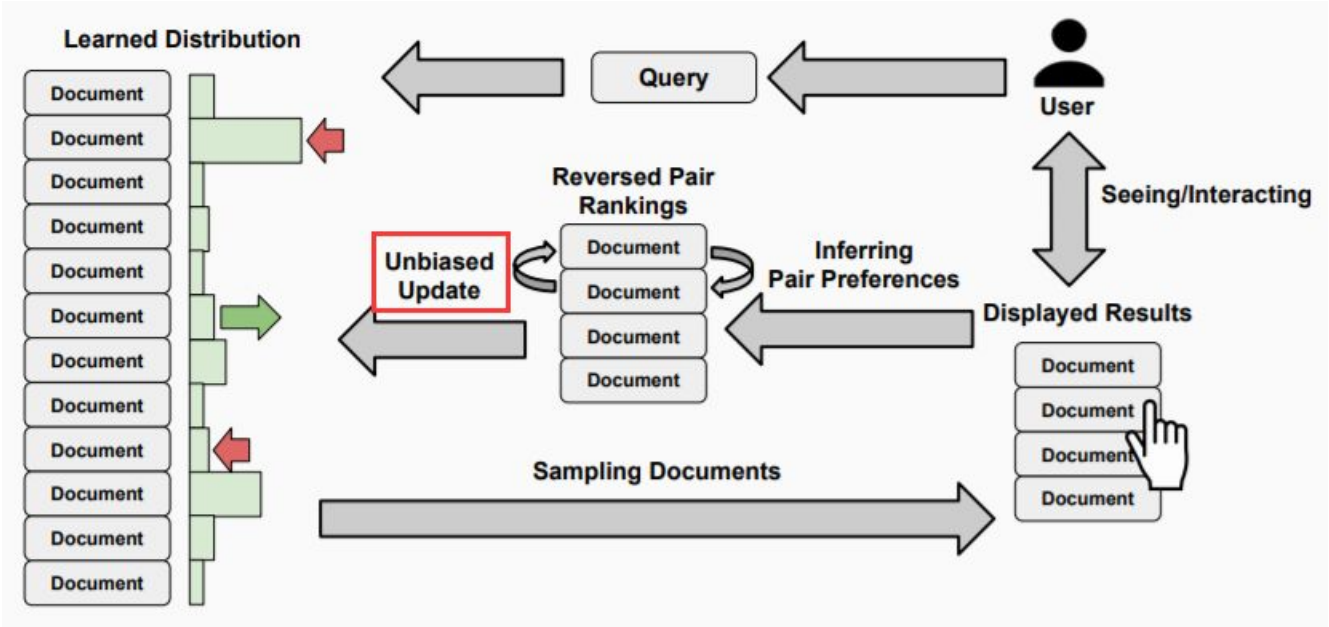
$$\rho(d_i, d_j, R) = \frac{P(R^*(d_i, d_j, R)|f, D)}{P(R|f, D) + P(R^*(d_i, d_j, R)|f, D)}.$$

This ratio is used to **reweight the found biased preference** and therefore **unbias the gradient estimation during model update:**

$$\nabla f_\theta(\cdot) \approx \sum_{d_i >_\mathbf{c} d_j} \rho(d_i, d_j, R) \nabla P(d_i \succ d_j | D, \theta).$$

(RQ2/unbiasedness answered - see paper for proof of this equation)

Method - Workflow



Experiments

Setup

- 5 datasets: sets of queries with each query having a corresponding preselected document set and relevance labels from 0-4
- 3 baseline models: DBGD, MGD, pairwise
- 2 types of models: linear & non-linear (=neural networks)

Metrics

- offline performance/final convergence: average NDCG@10 of the ranking model over the queries in the held-out test-set
- online performance/ranking quality during training: cumulative discounted NDCG@10 of the rankings displayed during training

Experiments

User behavior (clicking/stopping) is modelled as probability over relevance label R

Table 2: Instantiations of Cascading Click Models [10] as used for simulating user behavior in experiments.

	$P(click = 1 \mid R)$					$P(stop = 1 \mid click = 1, R)$				
R	0	1	2	3	4	0	1	2	3	4
perf	0.0	0.2	0.4	0.8	1.0	0.0	0.0	0.0	0.0	0.0
nav	0.05	0.3	0.5	0.7	0.95	0.2	0.3	0.5	0.7	0.9
inf	0.4	0.6	0.7	0.8	0.9	0.1	0.2	0.3	0.4	0.5

Results and Analysis

1. Convergence of ranking models
2. User experience during training
3. Answering RQ1 & RQ3

Results and Analysis

Convergence of ranking models

PDGD learns faster than existing OLTR methods

Table 3: Offline performance (NDCG) for different instantiations of CCM (Table 2). The standard deviation is shown in brackets, bold values indicate the highest performance per dataset and click model, significant improvements over the DBGD, MGD and pairwise baselines are indicated by △ ($p < 0.05$) and ▲ ($p < 0.01$), no losses were measured.

	MQ2007	MQ2008	MSLR-WEB10k	Yahoo	istella
			perfect		
DBGD (linear)	0.483 (0.023)	0.683 (0.024)	0.331 (0.010)	0.684 (0.010)	0.448 (0.014)
DBGD (neural)	0.463 (0.025)	0.670 (0.026)	0.319 (0.014)	0.676 (0.016)	0.429 (0.017)
MGD (linear)	0.494 (0.022)	0.690 (0.019)	0.333 (0.003)	0.714 (0.002)	0.496 (0.004)
Pairwise (linear)	0.479 (0.022)	0.674 (0.017)	0.315 (0.003)	0.709 (0.001)	0.252 (0.002)
PDGD (linear)	**0.511 (0.017)** ▲▲▲▲	**0.699 (0.024)** ▲▲▲▲	0.427 (0.005) ▲▲▲▲	**0.736 (0.004)** ▲▲▲▲	0.573 (0.004) ▲▲▲▲
PDGD (neural)	0.509 (0.020) ▲▲▲▲	0.698 (0.024) ▲▲▲▲	**0.430 (0.006)** ▲▲▲▲	0.733 (0.005) ▲▲▲▲	**0.575 (0.006)** ▲▲▲▲
			navigational		
DBGD (linear)	0.461 (0.025)	0.670 (0.025)	0.319 (0.011)	0.661 (0.023)	0.401 (0.015)
DBGD (neural)	0.430 (0.033)	0.646 (0.031)	0.304 (0.019)	0.649 (0.029)	0.382 (0.024)
MGD (linear)	0.426 (0.020)	0.662 (0.015)	0.321 (0.003)	0.706 (0.009)	0.405 (0.004)
Pairwise (linear)	0.476 (0.022)	0.677 (0.018)	0.312 (0.003)	0.696 (0.004)	0.209 (0.002)
PDGD (linear)	**0.496 (0.019)** ▲▲▲▲	**0.695 (0.021)** ▲▲▲▲	**0.406 (0.015)** ▲▲▲▲	**0.725 (0.005)** ▲▲▲▲	**0.540 (0.008)** ▲▲▲▲
PDGD (neural)	0.493 (0.020) ▲▲▲▲	0.692 (0.019) ▲▲▲▲	0.386 (0.019) ▲▲▲▲	0.722 (0.006) ▲▲▲▲	0.532 (0.011) ▲▲▲▲
			informational		
DBGD (linear)	0.411 (0.036)	0.631 (0.036)	0.299 (0.017)	0.620 (0.035)	0.360 (0.028)
DBGD (neural)	0.383 (0.047)	0.595 (0.053)	0.276 (0.033)	0.603 (0.040)	0.316 (0.057)
MGD (linear)	0.406 (0.021)	0.647 (0.036)	0.318 (0.003)	0.676 (0.043)	0.387 (0.005)
Pairwise (linear)	0.478 (0.022)	0.677 (0.018)	0.311 (0.003)	0.690 (0.006)	0.183 (0.001)
PDGD (linear)	**0.487 (0.021)** ▲▲▲▲	**0.690 (0.022)** ▲▲▲▲	**0.368 (0.025)** ▲▲▲▲	**0.713 (0.008)** ▲▲▲▲	**0.532 (0.010)** ▲▲▲▲
PDGD (neural)	0.483 (0.022) ▲▲▲	0.686 (0.022) ▲▲▲▲	0.355 (0.021) ▲▲▲▲	0.709 (0.009) ▲▲▲▲	0.525 (0.012) ▲▲▲▲

Results and Analysis

Convergence of ranking models

- PDGD has an improved point of final convergence and learns faster compared to DBGD and MGD
- Needs more data to achieve fully converge.
- Speed-quality tradeoff.

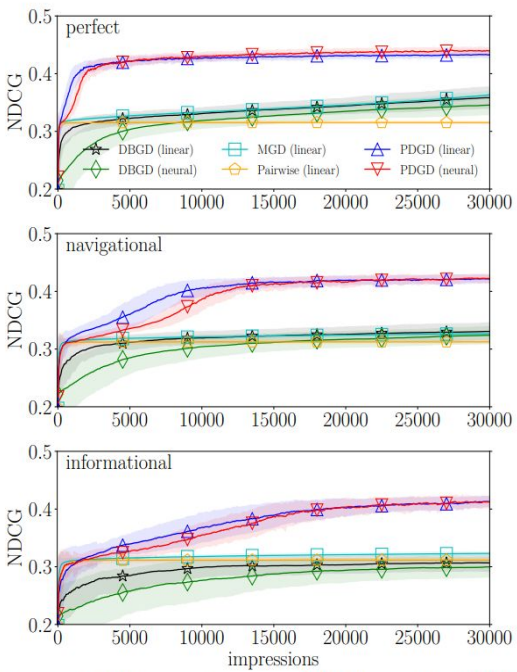


Figure 2: Offline performance (NDCG) on the MSLR-WEB10k dataset under three different click models, the shaded areas indicate the standard deviation.

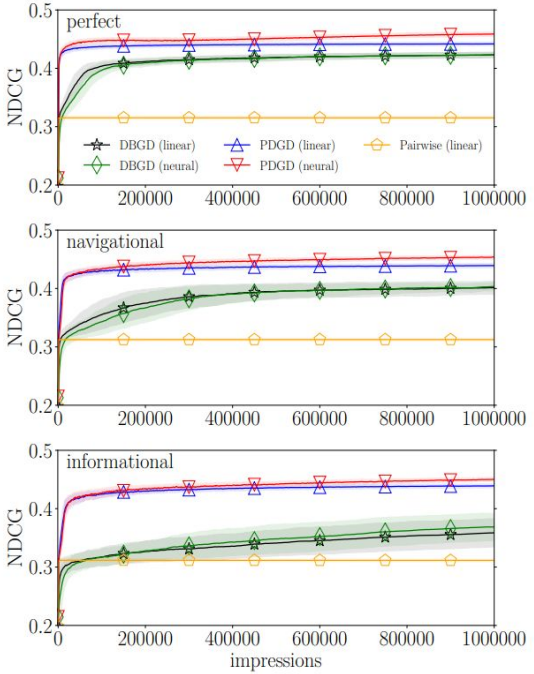


Figure 3: Long-term offline performance (NDCG) on the MSLR-WEB10k dataset under three click models, the shaded areas indicate the standard deviation.

Results and Analysis

User experience during training

- Online performance of DBGD and MGD are close; MGD has a higher online performance.
- Pairwise baseline has a lower online performance
- PDGD has significant improvements.

Table 4: Online performance (Discounted Cumulative NDCG, Section 4.4) for different instantiations of CCM (Table 2). The standard deviation is shown in brackets, bold values indicate the highest performance per dataset and click model, significant improvements and losses over the DBGD, MGD and pairwise baselines are indicated by △ ($p < 0.05$) and ▲ ($p < 0.01$) and by ▽ and ▼, respectively.

	MQ2007	MQ2008	MSLR-WEB10k	Yahoo	istella
			perfect		
DBGD (linear)	675.7 (21.8)	843.6 (40.8)	533.6 (15.6)	1159.3 (31.6)	589.9 (19.2)
DBGD (neural)	602.7 (58.1)	776.9 (67.4)	481.2 (53.0)	1135.7 (41.3)	494.3 (60.5)
MGD (linear)	689.6 (15.3)	858.6 (40.6)	558.7 (6.4)	1203.9 (9.9)	670.9 (8.6)
Pairwise (linear)	458.4 (13.3)	616.6 (25.8)	345.3 (4.6)	1027.2 (9.2)	64.5 (2.1)
PDGD (linear)	**797.3** (17.3) ▲ ▲ ▲ ▲	**959.7** (43.4) ▲ ▲ ▲ ▲	**691.4** (12.3) ▲ ▲ ▲ ▲	**1360.3** (10.8) ▲ ▲ ▲ ▲	**957.5** (9.4) ▲ ▲ ▲ ▲
PDGD (neural)	743.7 (18.8) ▲ ▲ ▲ ▲	925.4 (43.3) ▲ ▲ ▲ ▲	619.2 (13.6) ▲ ▲ ▲ ▲	1319.6 (10.1) ▲ ▲ ▲ ▲	834.0 (22.2) ▲ ▲ ▲ ▲
			navigational		
DBGD (linear)	638.6 (29.7)	816.9 (42.0)	508.2 (21.6)	1129.9 (32.2)	538.2 (29.0)
DBGD (neural)	573.7 (68.4)	740.3 (69.7)	465.8 (52.0)	1116.0 (45.7)	414.3 (96.2)
MGD (linear)	635.9 (14.7)	824.5 (34.0)	538.1 (7.6)	1181.7 (20.0)	593.2 (9.7)
Pairwise (linear)	459.9 (12.9)	618.6 (25.2)	347.3 (5.4)	1031.2 (9.0)	72.6 (2.2)
PDGD (linear)	**703.0** (17.9) ▲ ▲ ▲ ▲	**903.1** (40.7) ▲ ▲ ▲ ▲	**578.1** (16.0) ▲ ▲ ▲ ▲	**1298.4** (33.4) ▲ ▲ ▲ ▲	**704.1** (33.5) ▲ ▲ ▲ ▲
PDGD (neural)	560.9 (14.6) ▼ ▽ ▼ ▲	788.7 (38.5) ▼ ▲ ▼ ▲	448.1 (12.3) ▼ ▼ ▼ ▲	1176.1 (17.0) ▲ ▲ ▽ ▲	390.2 (35.1) ▼ ▼ ▼ ▲
			informational		
DBGD (linear)	584.2 (41.1)	757.4 (56.9)	477.2 (32.2)	1110.0 (37.0)	436.8 (57.4)
DBGD (neural)	550.8 (75.7)	720.9 (79.0)	444.7 (60.9)	1091.2 (48.6)	322.9 (121.0)
MGD (linear)	618.8 (21.7)	815.1 (44.5)	540.0 (7.7)	1159.1 (40.0)	581.8 (10.7)
Pairwise (linear)	462.6 (14.4)	619.6 (25.0)	349.7 (6.6)	1034.1 (9.0)	77.0 (2.4)
PDGD (linear)	**704.8** (30.5) ▲ ▲ ▲ ▲	**907.9** (42.0) ▲ ▲ ▲ ▲	**567.3** (36.5) ▲ ▲ ▲ ▲	**1266.7** (50.0) ▲ ▲ ▲ ▲	**731.5** (80.0) ▲ ▲ ▲ ▲
PDGD (neural)	594.6 (23.0) △ ▲ ▼ ▲	818.3 (39.6) ▲ ▲ ▲	470.1 (19.4) ▽ ▲ ▼ ▲	1178.1 (22.8) ▲ ▲ ▲ ▲	484.3 (64.8) ▲ ▲ ▼ ▲

Results and Analysis

Answering RQ1 & RQ3

- RQ1 Does using PDGD result in significantly better performance than MGD?
  - **PDGD outperforms existing methods (model convergence & user experience)**

- RQ3 Is PDGD capable of effectively optimizing different types of ranking models?
  - **PDGD is applicable to different ranking models and effective for both linear and non-linear models.**

Conclusion

1. PDGD outperforms existing methods both in terms of model convergence and user experience during learning.
2. The gradient estimation of PDGD is unbiased.
3. PDGD is applicable to different ranking models and effective for both linear and non-linear models.

Q&A