Unbiased Learning to Rank: Counterfactual and Online Approaches

Harrie Oosterhuis* **Rolf Jagerman*** **Maarten de Rijke*,****

April 21, 2020

*University of Amsterdam

**Ahold Delhaize

oosterhuis@uva.nl, rolf.jagerman@uva.nl, derijke@uva.nl

The Web Conference 2020 Tutorial

Part 3: Online Learning to Rank

Online Learning to Rank: Overview

This part will cover the following topics:

- **Online Evaluation**
  - Comparing rankers through interleaving.
- **Dueling Bandit Gradient Descent**
  - Learning to rank as an interactive dueling bandit problem.
- **Pairwise Differentiable Gradient Descent**
  - Learning to rank through unbiased pairwise optimization.
- **Comparison of PDGD and DBGD**
  - Theoretical differences and empirical comparisons.

Related Work: Bandits for Ranking

Ranking as a K-Armed Bandit

In the past, ranking has been modelled as a K-armed bandit (Busa-Fekete and Hüllermeier, 2014).

These methods aim to find the **optimal ranking for a single query**.

Ranking bandit methods include:

- **Upper confidence bounds** on relevances per document (Kveton et al., 2015).
- **Divide and conquer**: split documents in groups so that there are high-confidence relevance differences between groups (Lattimore et al., 2018).
- **Click-through-rate estimation** per document similar to counterfactual LTR (Lagrée et al., 2016).

Ranking Bandits and Learning to Rank

The goal of ranking bandit algorithms is:

- the **optimal ranking** for a **single query**.

The results from these algorithms do **not generalize** to other queries, i.e., there is **no resulting ranking model**.

Advantage: rankings **not limited by features** (Zoghi et al., 2016).
Disadvantage: **learning from scratch** for every new query.

Very different from the **goal** of LTR as defined for this **tutorial**:

- to find a **ranking model** that **generalizes** well across user queries.

Online Evaluation

Online Evaluation: Introduction

We have seen:

- Counterfactual evaluation corrects for position bias in historical logs by explicitly modelling the user's examination probabilities.

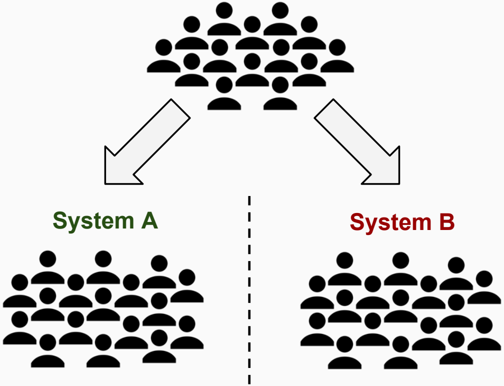
One way of getting these **explicit probabilities** is through **randomization**.

Alternatively, older methods use **randomization** to **directly perform evaluation**:

- A/B testing
- Interleaving

They answer the **question**: **Should ranker A be preferred over ranker B?**

Online Evaluation: A/B testing



A/B testing **randomizes system exposure to users** to measure differences.

Online Evaluation: Interleaving

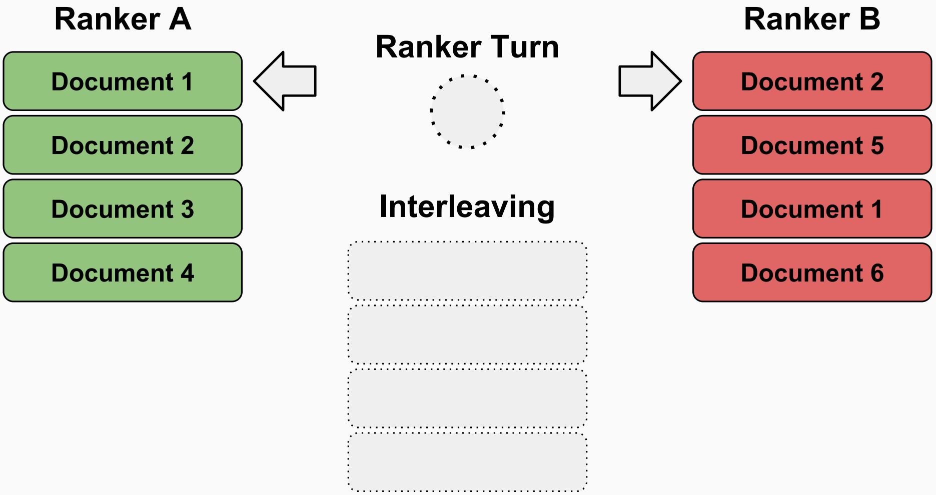
A/B testing is powerful and widely applicable, it is **not specific for rankings**.

Specific aspects of interactions in rankings can be used for **more efficient comparisons**.

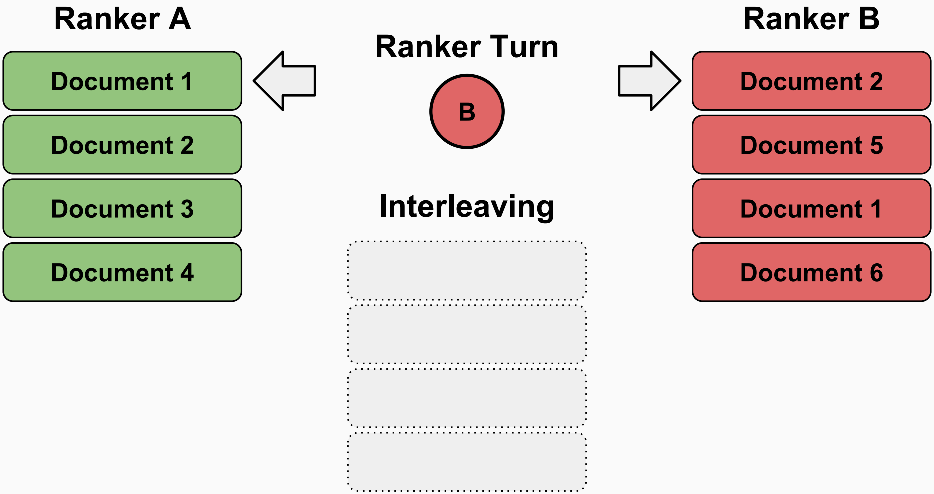
Interleaving (Joachims, 2003):

- Take the two rankings for a query from two rankers .
- Create an **interleaved ranking**, based on both rankings.
- **Clicks** on an interleaved ranking provide **preference signals** between rankers.

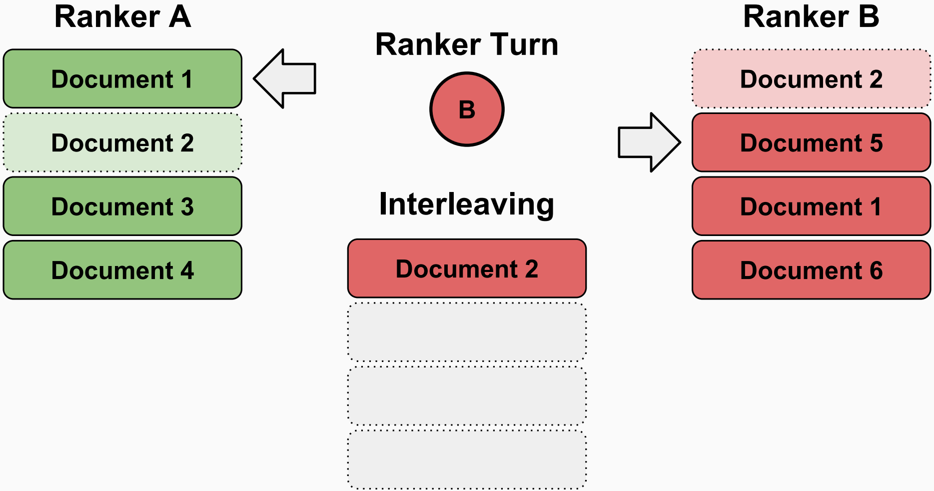
Online Evaluation: Team-Draft Interleaving



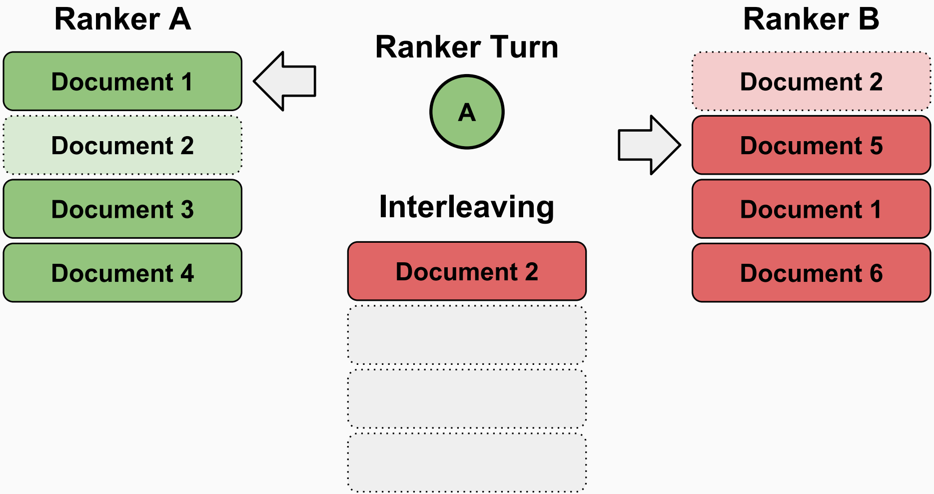
Online Evaluation: Team-Draft Interleaving



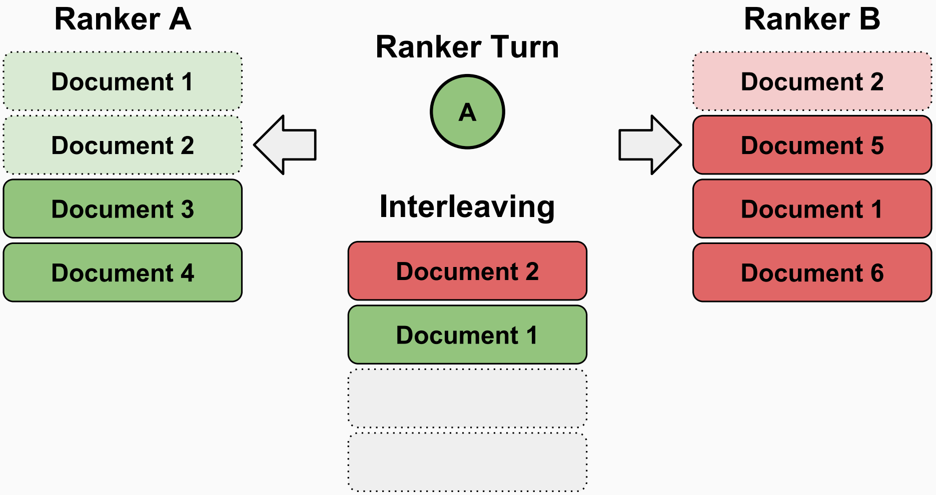
Online Evaluation: Team-Draft Interleaving



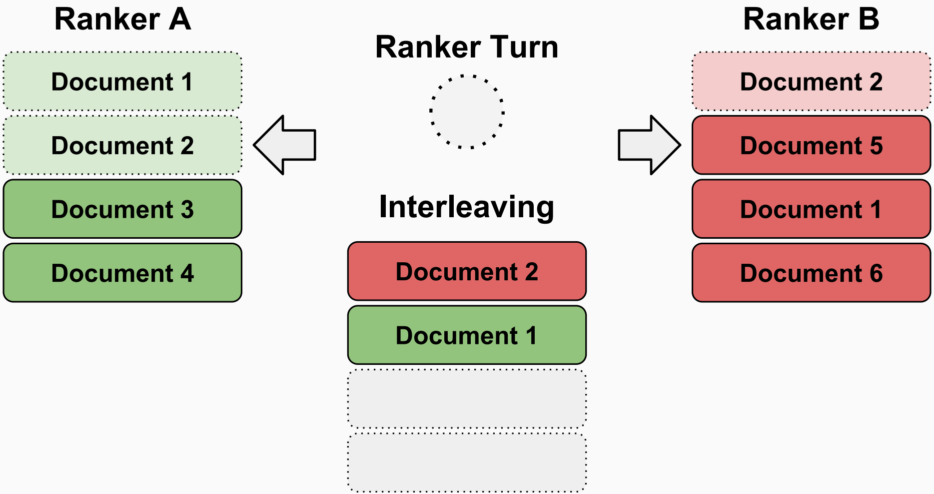
Online Evaluation: Team-Draft Interleaving



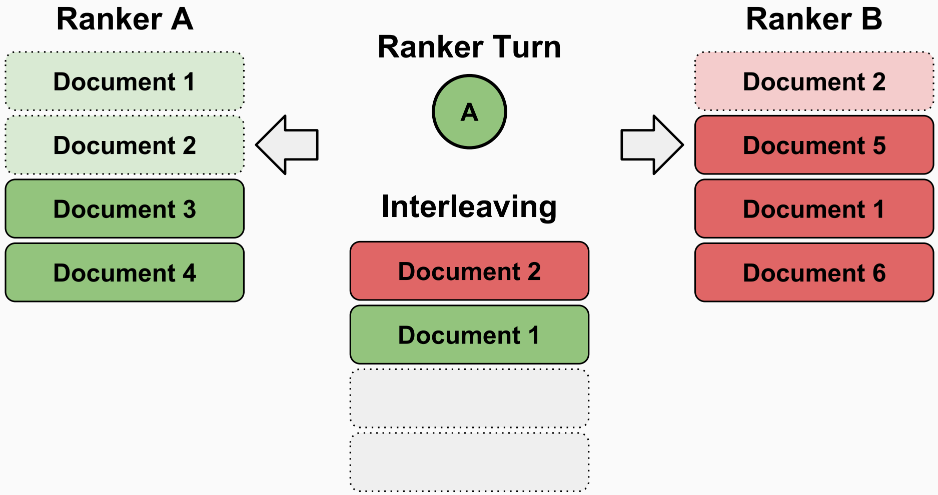
Online Evaluation: Team-Draft Interleaving



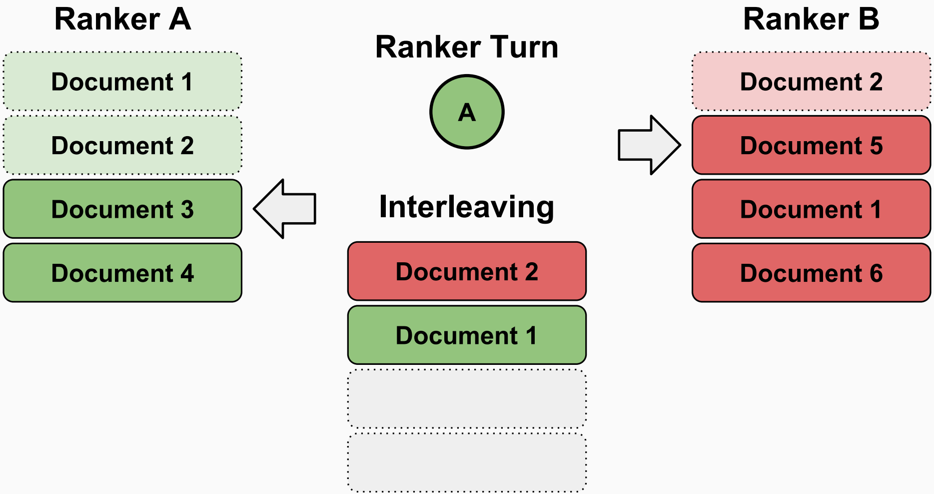
Online Evaluation: Team-Draft Interleaving



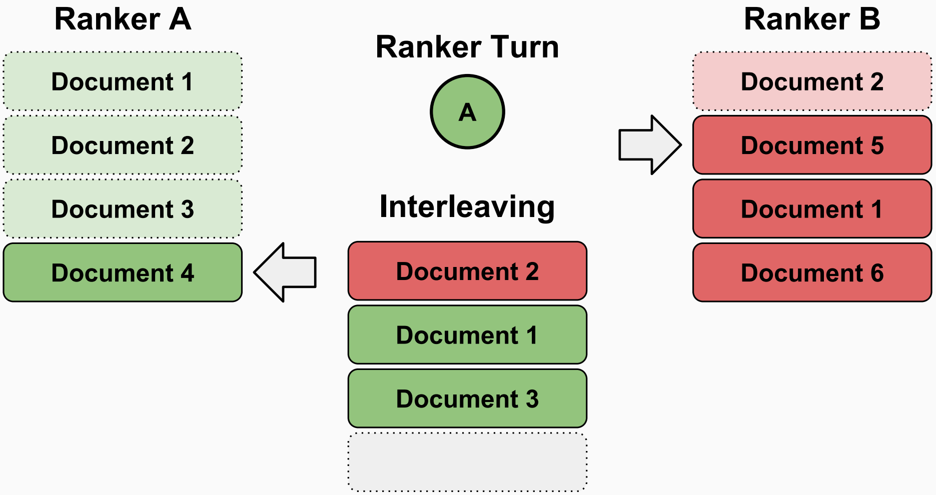
Online Evaluation: Team-Draft Interleaving



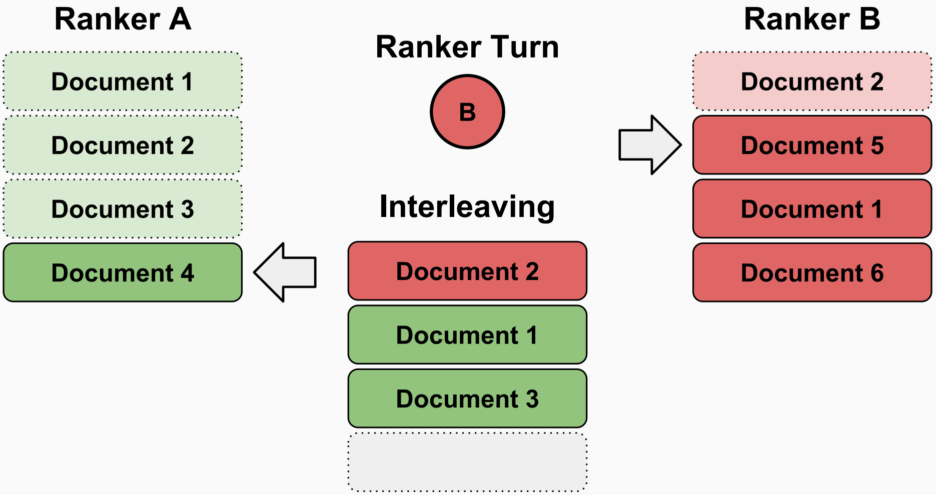
Online Evaluation: Team-Draft Interleaving



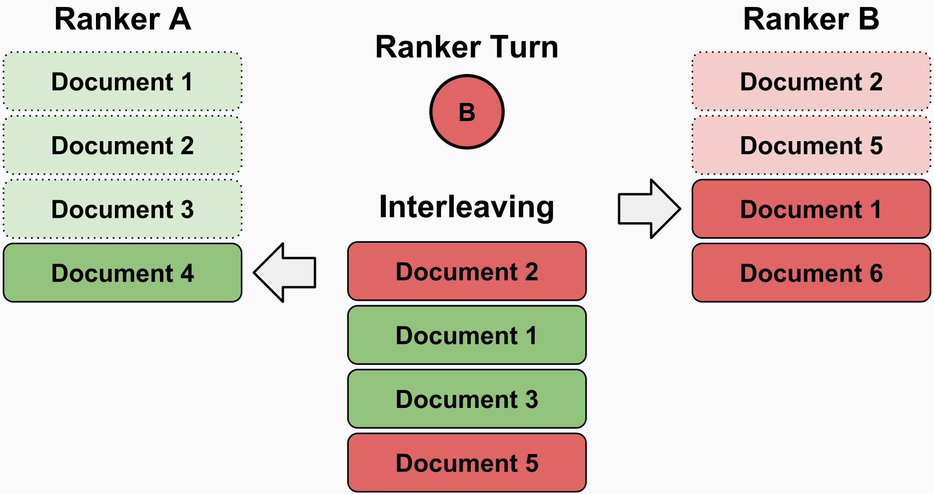
Online Evaluation: Team-Draft Interleaving



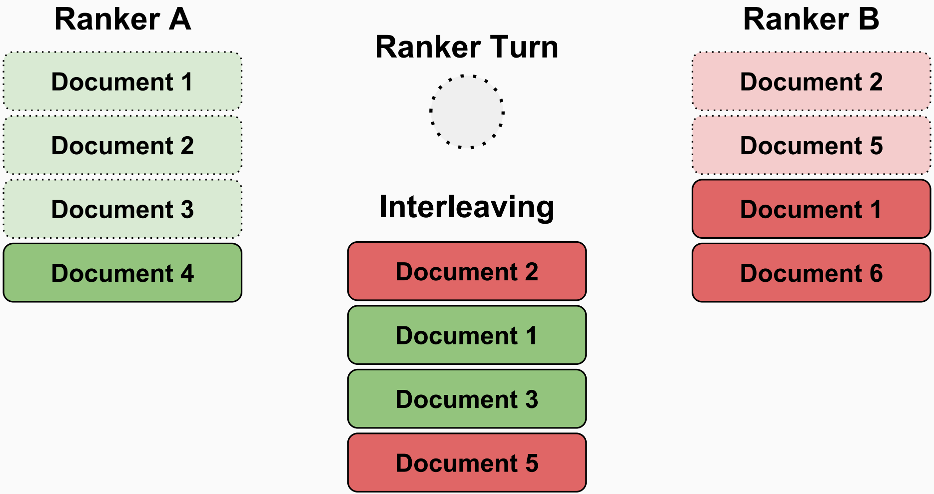
Online Evaluation: Team-Draft Interleaving



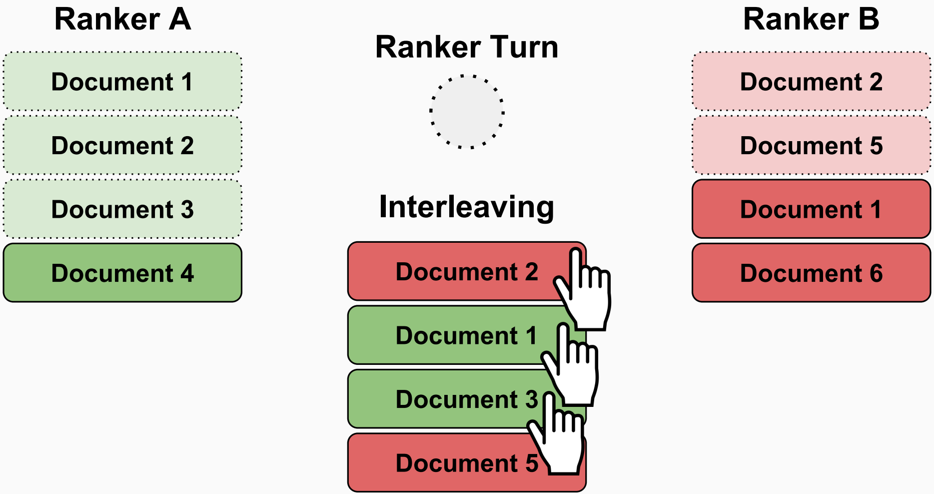
Online Evaluation: Team-Draft Interleaving



Online Evaluation: Team-Draft Interleaving



Online Evaluation: Team-Draft Interleaving



Online Evaluation: Team-Draft Interleaving

Ranker A

- Document 1
- Document 2
- Document 3
- Document 4

Ranker A receives two clicks.

Ranker Turn

Interleaving

- Document 2
- Document 1
- Document 3
- Document 5

Ranker B

- Document 2
- Document 5
- Document 1
- Document 6

Ranker B receives one click.

Online Evaluation: Interleaving

The idea behind interleaving:

- **Randomize display positions** of documents to deal with position bias.
- Limit randomization to **maintain user experience**.

Team-Draft Interleaving (Radlinski et al., 2008) is **affected by position bias**:

- Similar rankers can be inferred equal when a preference should be found.

Other interleaving methods are **proven** to be **unbiased**[1]:

- **Probabilistic Interleaving** (Hofmann et al., 2011)
- **Optimized Interleaving** (Radlinski and Craswell, 2013)

[1] Different definition of unbiased than the first part of this tutorial.

Online Evaluation: Interleaving

Interleaving requires **magnitudes fewer interactions** for a reliable preference than A/B testing (Chapelle et al., 2012; Yue et al., 2010).

Unlike counterfactual evaluation, interleaving **is interactive**.

- It is not effective on historical data (Hofmann et al., 2013).

Efficiency comes from:

- displaying the **most important documents** first,
- and looking for **relative differences**.

Providing a reliable, efficient and interactive evaluation methodology.

Dueling Bandit Gradient Descent

Dueling Bandit Gradient Descent: Introduction

Introduced by Yue and Joachims (2009) as the **first online learning to rank** method.

Intuition:

- if **online evaluation** can tell us if a **ranker is better** than another, then we can use it to **find an improvement** of our system.

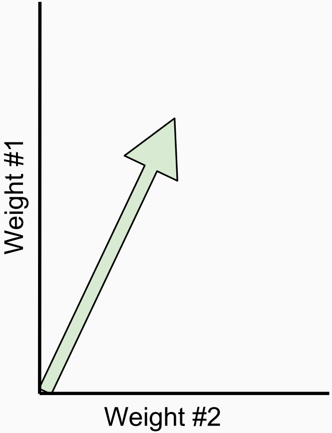
By **sampling model variants** and **comparing** them with **interleaving**, the *gradient* of a model w.r.t. user satisfaction can be **estimated**.

Dueling Bandit Gradient Descent: Method

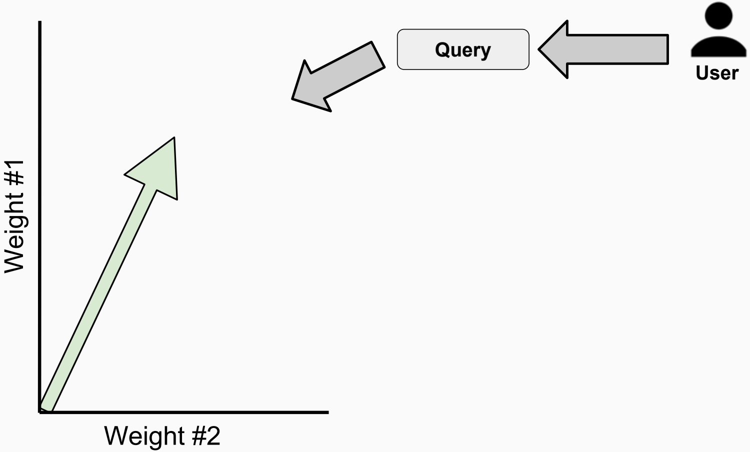
Start with the **current** ranking model **parameters**: θ_b.
Then indefinitely:

1. Wait for a user query.
2. **Sample a random direction** from the unit sphere: u, (thus $|u| = 1$).
3. Compute the **candidate ranking model** $\theta_c = \theta_b + u$, (thus $|\theta_b - \theta_c| = 1$).
4. Get the **rankings** of θ_b and θ_c.
5. **Compare** θ_b and θ_c using interleaving.
6. If θ_c wins the **comparison**:
    - **Update** the current model: $\theta_b \leftarrow \theta_b + \eta(\theta_c - \theta_b)$

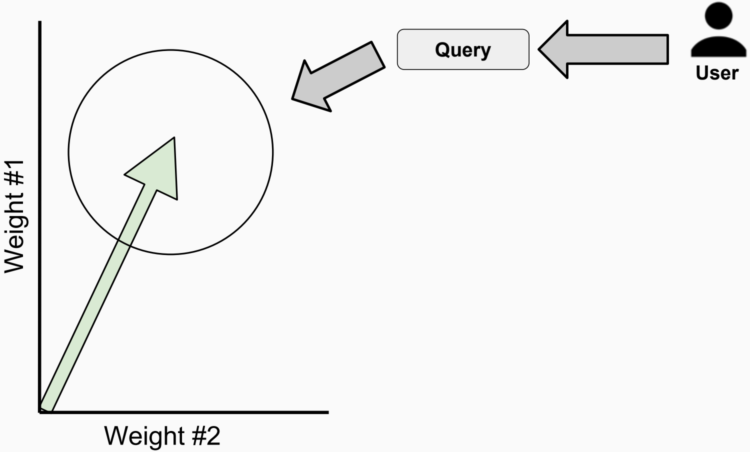
Dueling Bandit Gradient Descent: Visualization



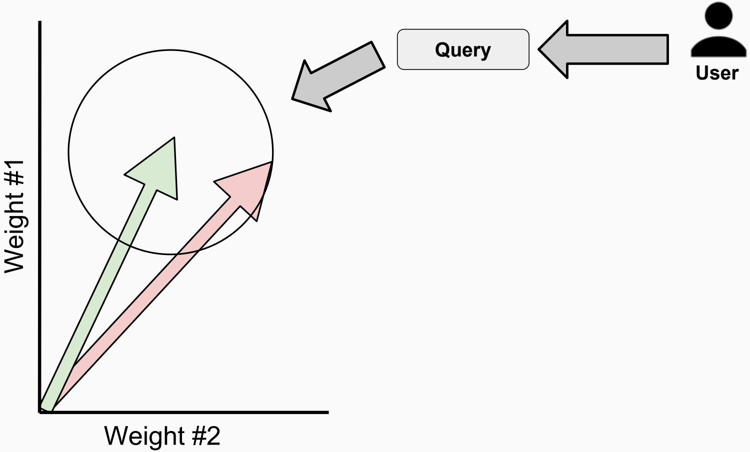
Dueling Bandit Gradient Descent: Visualization



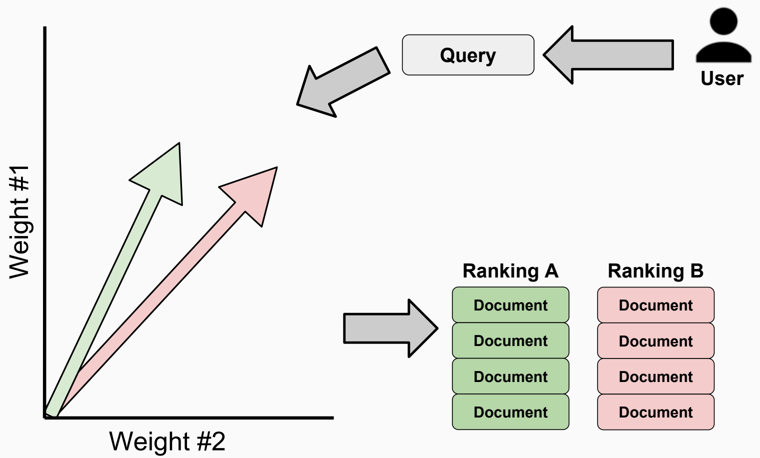
Dueling Bandit Gradient Descent: Visualization



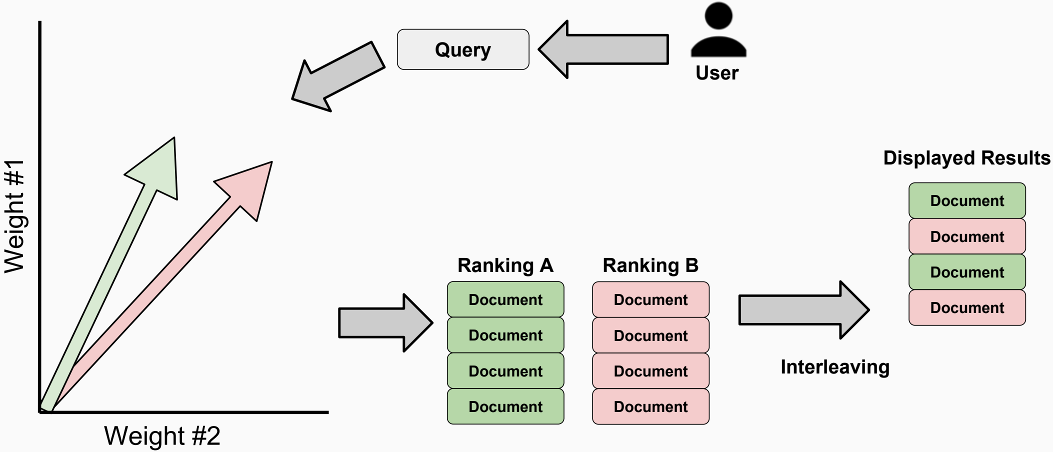
Dueling Bandit Gradient Descent: Visualization



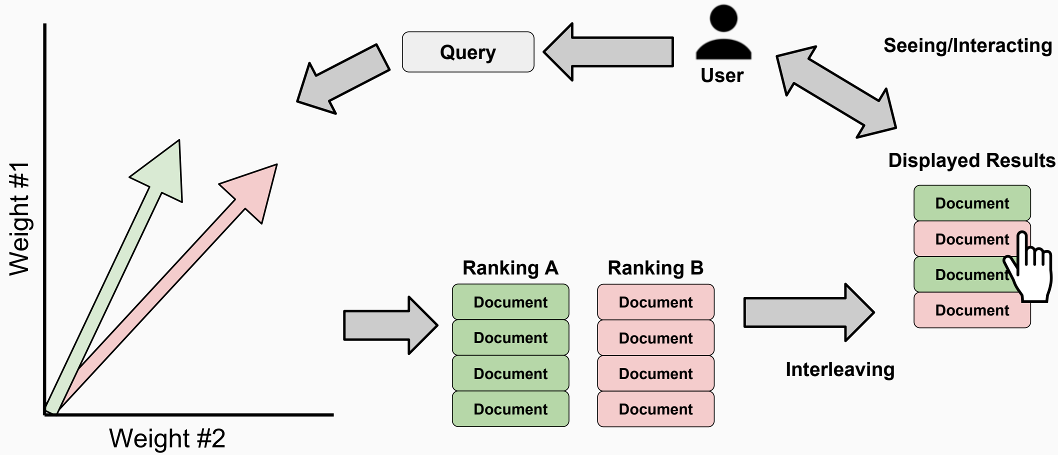
Dueling Bandit Gradient Descent: Visualization



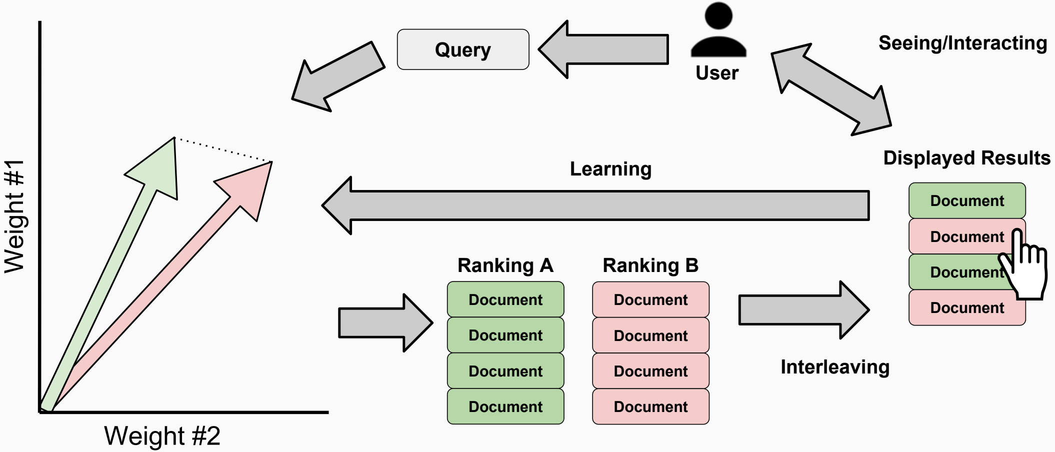
Dueling Bandit Gradient Descent: Visualization



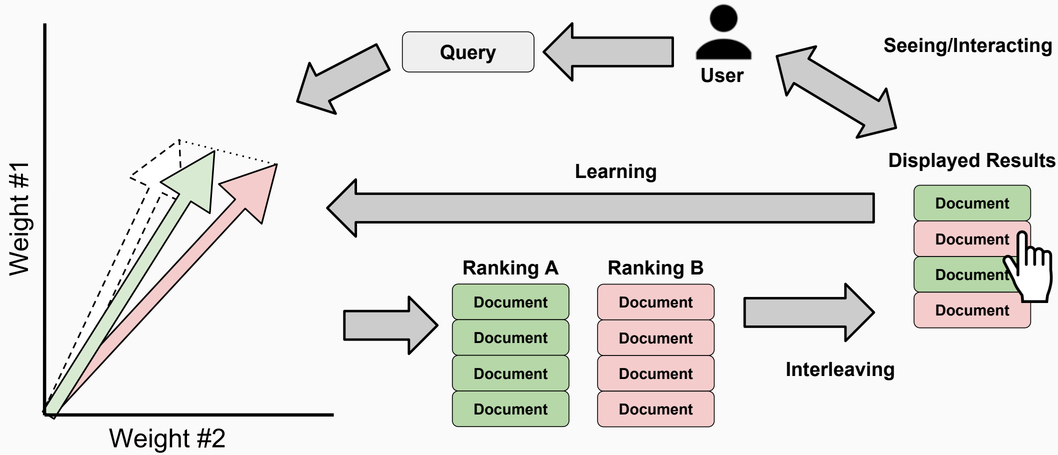
Dueling Bandit Gradient Descent: Visualization



Dueling Bandit Gradient Descent: Visualization



Dueling Bandit Gradient Descent: Visualization



Dueling Bandit Gradient Descent: Properties

Yue and Joachims (2009) prove that under the **assumptions**:

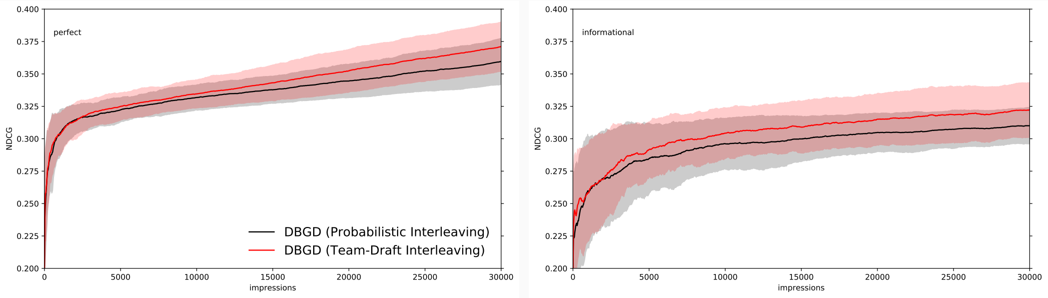
- There is a **single optimal** set of parameters: θ^*.
- The **utility space** w.r.t. θ is **smooth**,
  i.e., small changes in θ lead to small changes in user experience.

Then Dueling Bandit Gradient Descent is **proven** to have a **sublinear regret**:

- The algorithm will **eventually** approximate the ideal model.
- The duration of time is effected by the number of parameters of the model, the smoothness of the space, the unit chosen, etc.

Dueling Bandit Gradient Descent: Visualization

Simulations based on offline datasets: **user behavior** is based on the **annotations**.
As a result, we can **measure** how close the **model** is getting to their **satisfaction**.



Simulated results on the MSLR-WEB10k dataset,
a perfect user (left) and an informational user (right).

Image credits: (Oosterhuis, 2018).

Reusing Historical Interactions

Reusing Historical Interactions

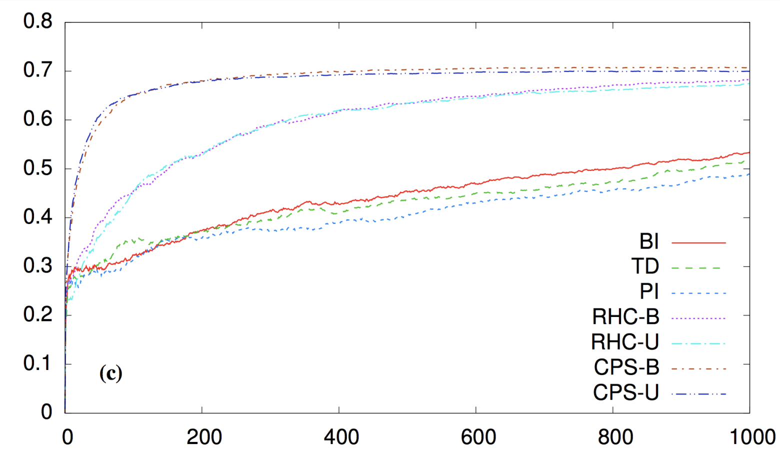
Hofmann et al. (2013) introduced the idea of **guiding exploration** by **reusing previous interactions**.

Intuition: if **previous interactions** showed that a **direction is unfruitful** then we should **avoid it in the future**.

Candidate Pre-Selection:

- Sample a **large number** of rankers to create a **candidate set**.
- **Compare two** candidate rankers based on a **historical interaction**.
- **Remove loser** from candidate set.
- **Repeat** until a **single candidate** is left.

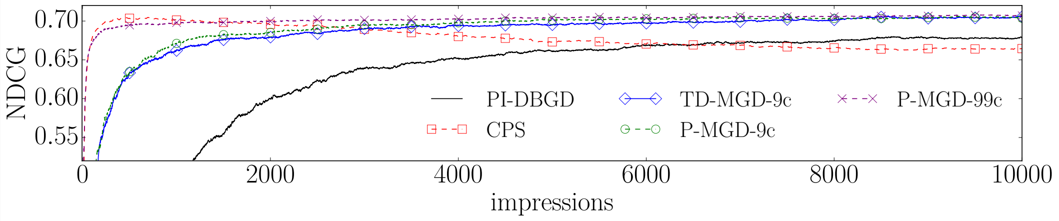
Reusing Historical Interactions: Performance



Simulated results on the NP2003 dataset.

Image credits: graph from (Hofmann et al., 2013).

Reusing Historical Interactions: Long Term Performance



Simulated results on the NP2003 dataset.

Remember, in the online setting the **performance cannot be measured**, thus **early-stopping is unfeasible**.

Image credits: graph from (Oosterhuis et al., 2016).

Reusing Historical Interactions: Other Work

Besides Hofmann et al. (2013) **other work** has also tried **reusing historical interactions** for online learning to rank: (Zhao and King, 2016; Wang et al., 2018a).

The problem with these works is that:

- they **do not consider the long-term convergence**.
- they were **not evaluated** on the **largest available industry datasets**.

As a result, it is **still unclear** whether we can **reliably reuse historical interactions** during online learning.

Multileave Gradient Descent

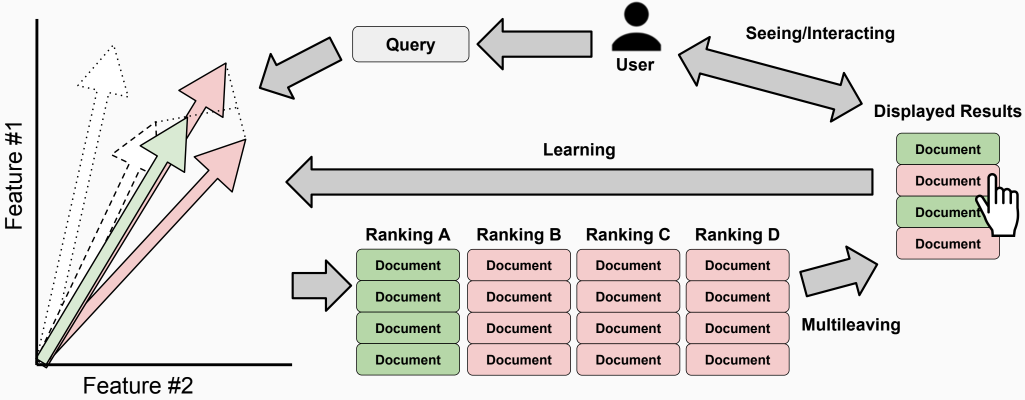
Multileave Gradient Descent

The introduction of **multileaving** in online evaluation allowed for **multiple rankers being compared simultaneously** from a single interaction.

A **natural extension** of Dueling Bandit Gradient Descent is to combine it with multileaving, resulting in **Multileave Gradient Descent** (Schuth et al., 2016).

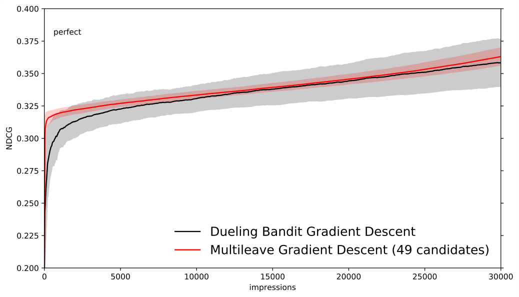
Multileaving allows comparisons with **multiple candidate rankers**, **increasing** the **chance of finding an improvement**.

Multileave Gradient Descent: Visualization



Multileave Gradient Descent: Results

Results on the MSRL10k dataset under simulated users:



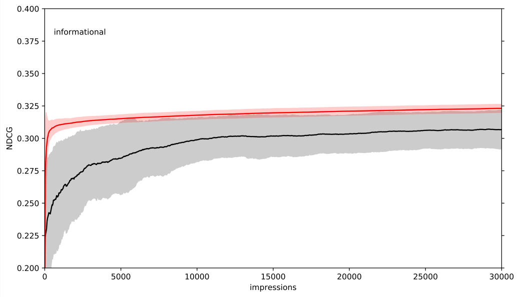


Image credits: (Oosterhuis, 2018).

Multileave Gradient Descent: Conclusion

Properties of Multileave Gradient Descent:

- **Vastly speeds up the learning rate** of Dueling Bandit Gradient Descent.
  - Much better user experience.
- Instead of **limiting (guiding) exploration**, it is done more **efficiently**.
- **Huge computational costs**, large number of rankers have to be applied.

Problems with Dueling Bandit Gradient Descent

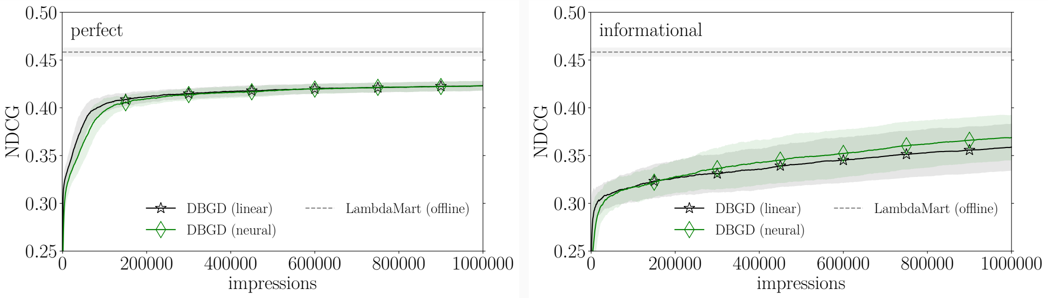
Problems with Dueling Bandit Gradient Descent

A **problem** with Dueling Bandit Gradient Descent and **all its extensions**:

- Their **performance at convergence** is **much worse** than offline approaches, even **under ideal user interactions**.

DBGD problems: Empirical

Results on the MSRL10k dataset under simulated users:



How is this possible, if it has **proven sub-linear regret**?

Image credits: (Oosterhuis, 2018).

Problems with the Dueling Bandit Gradient Descent Bounds

Remember the **regret** of Dueling Bandit Gradient Descent made **two assumptions**:

- There is a **single optimal model**: θ^*.
- The **utility space is smooth** w.r.t. to the model weights θ.

These **assumptions do not hold** for all models that are used in practice (Oosterhuis and de Rijke, 2019).

To prove this we use the fact that **the utility u is scale invariant** w.r.t. a ranking function $f_\theta(\cdot)$:

$$\forall \theta, \quad \forall \alpha \in \mathbb{R}_{>0}, \quad u(f_\theta(\cdot)) = u(\alpha f_\theta(\cdot)).$$

DBGD Assumptions: Single Optimal Model

First assumption: There is a **single optimal model**: θ^*.

For any linear or neural model:

- if θ^* has the **optimal performance**,
- then $\theta' = \alpha\theta$ has the **same performance**, *(linear model)* or multiplying the final weight matrix with α, *(neural model)*.

Therefore, there can **never** be a **single optimal model** θ^*.

DBGD Assumptions: Smoothness

Second assumption: The **utility space is smooth** w.r.t. to the model weights θ:

$$\exists L \in \mathbb{R}, \quad \forall (\theta_a, \theta_b) \in \mathcal{W}, \quad |u(\theta_a) - u(\theta_b)| < L\|\theta_a - \theta_b\|.$$

Since a **linear model** is **scale invariant**:

$$\forall \alpha \in \mathbb{R}_{>0}, \quad |u(\theta_a) - u(\theta_b)| = |u(\alpha\theta_a) - u(\alpha\theta_b)|,$$
$$\forall \alpha \in \mathbb{R}_{>0}, \quad \|\alpha\theta_a - \alpha\theta_b\| = \alpha\|\theta_a - \theta_b\|.$$

Thus the smoothness assumption can be rewritten as:

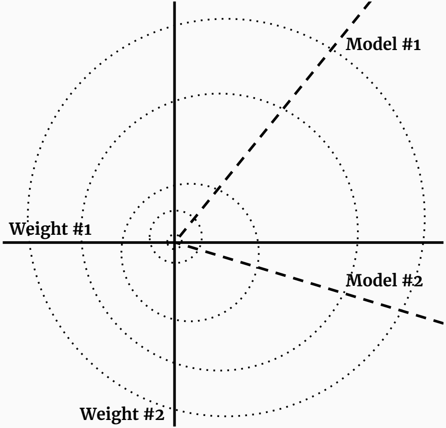
$$\exists L \in \mathbb{R}, \quad \forall \alpha \in \mathbb{R}_{>0}, \quad \forall (\theta_a, \theta_b) \in \mathcal{W}, \quad |u(\theta_a) - u(\theta_b)| < \alpha L\|\theta_a - \theta_b\|.$$

This condition is **impossible to be true** (proof can be extended for neural networks).

DBGD Assumptions: Smoothness Visualization

Intuition behind the **smoothness problem** for linear ranking models:

- **Every model** in a **line** from the origin in any direction is **equivalent**.
- **Any sphere** around the origin contains **every possible ranking model**[a].
- The **distance** between the *best* and the *worst* model becomes **infinitely small** near the origin.



[a] Except for the trivial random model on the origin.

DBGD Problems: Conclusion

Theoretical properties:

- Currently, no **sound regret bounds proven**.

Empirical observations:

- Methods do **not approach optimal performance**.
- Neural models have no advantage over linear models.

Possible solutions:

- Extend the algorithm (the last decade of research) or introduce new model.
- **Find an approach different to the bandit approach.**

Pairwise Differentiable Gradient Descent

Pairwise Differentiable Gradient Descent

We recently introduced **Pairwise Differentiable Gradient Descent** (Oosterhuis and de Rijke, 2018b):

- Very different from previous Online Learning to Rank methods,
  that relied on sampling model variations similar to evolutionary approaches.

Intuition:

- A **pairwise** approach can be made **unbiased**, while being **differentiable**,
  without relying on online evaluation method or the sampling of models.

Plackett Luce Model

Pairwise Differentiable Gradient Descent optimizes a **Plackett Luce** ranking model, this models a **probabilistic distribution over documents**.

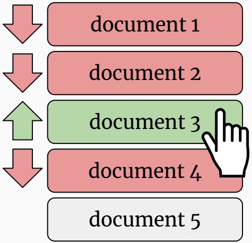
With the ranking scoring model $f_\theta(\mathbf{d})$ the distribution is:

$$P(d|D,\theta) = \frac{\exp^{f_\theta(\mathbf{d})}}{\sum_{d' \in D} \exp^{f_\theta(\mathbf{d'})}}.$$

Confidence is explicitly modelled and **exploration** depends on the **available documents**, thus it **naturally varies per query** and even within the ranking.

Bias in Pairwise Inference

Similar to existing pairwise methods (Oosterhuis and de Rijke, 2017; Joachims, 2002), Pairwise Differentiable Gradient Descent infers **pairwise document preferences from user clicks**:

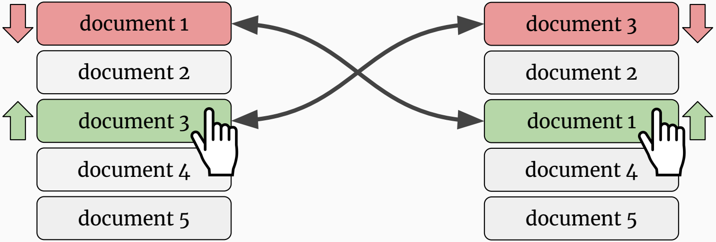


This approach is **biased**:

- Some preferences are **more likely to be inferred** due to **position/selection bias**.

Reversed Pair Rankings

Let $R^*(d_i, d_j, R)$ be R but with the **positions** of d_i and d_j **swapped**:



We assume:

- For a preference $d_i \succ d_j$ inferred from ranking R, if both are **equally relevant** the opposite preference $d_j \succ d_i$ is **equally likely** to be inferred from $R^*(d_i, d_j, R)$.

Then scoring **as if** R and R^* are **equally likely to occur** makes the gradient **unbiased**.

Unbiasing the Pairwise Update

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

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

We use this ratio to **unbias the gradient estimation**:

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

Unbiasedness of Pairwise Differentiable Gradient Descent

Under the reversed pair ranking assumption, we prove that **the expected estimated gradient** can be written as:

$$E[\nabla f_\theta(\cdot)] = \sum_{d_i, d_j} \alpha_{ij} (f'_\theta(\mathbf{d_i}) - f'_\theta(\mathbf{d_j})).$$

Where the weights α_{ij} will **match the user preferences** in expectation:

$$\begin{aligned} d_i =_{rel} d_j &\Leftrightarrow \alpha_{ij} = 0, \\ d_i >_{rel} d_j &\Leftrightarrow \alpha_{ij} > 0, \\ d_i <_{rel} d_j &\Leftrightarrow \alpha_{ij} < 0. \end{aligned}$$

Thus the estimated gradient is **unbiased w.r.t. document pair preferences**.

Pairwise Differentiable Gradient Descent: Method

Start with initial model θ_t, then indefinitely:

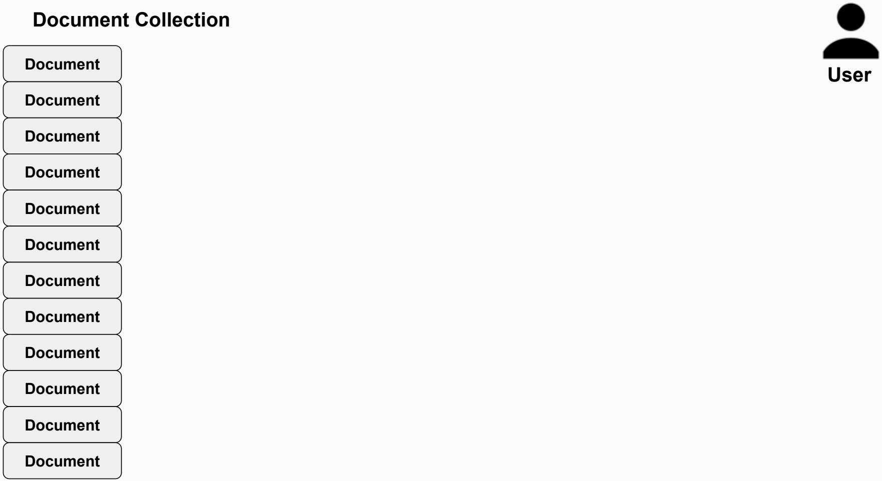
1. Wait for a user query.
2. **Sample** (without replacement) a **ranking** R from the document distribution:

$$P(d|D, \theta_t) = \frac{\exp^{f_{\theta_t}(\mathbf{d})}}{\sum_{d' \in D} \exp^{f_{\theta_t}(\mathbf{d'})}}.$$

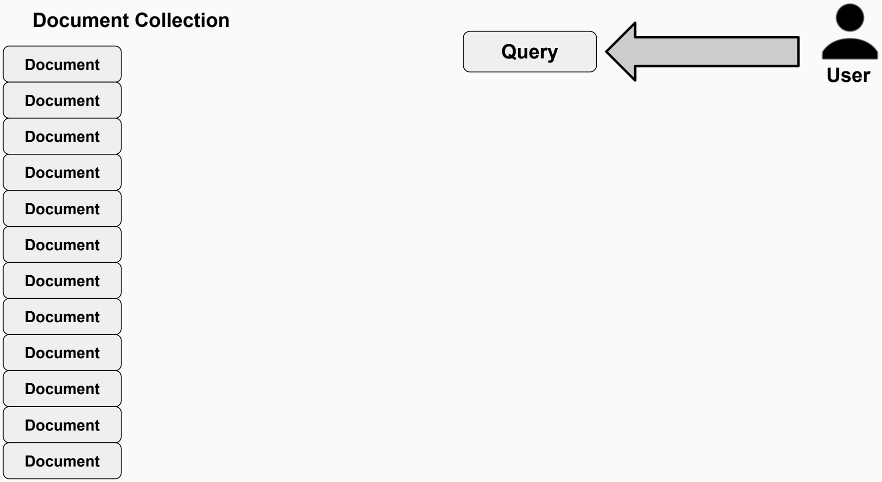
3. **Display** the ranking R to the user.
4. **Infer document preferences** from the **user clicks**: $\mathbf{c}$.
5. **Update** model according to the **estimated (unbiased) gradient**:

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

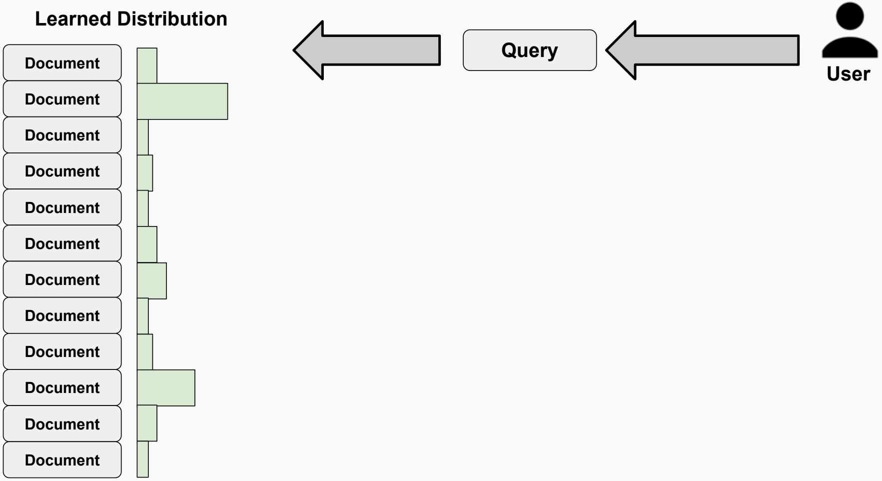
Pairwise Differentiable Gradient Descent: Visualization



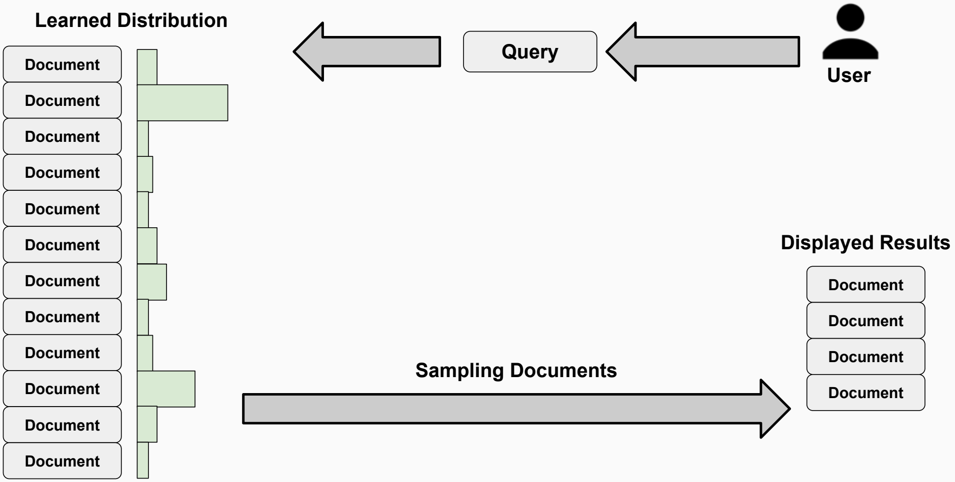
Pairwise Differentiable Gradient Descent: Visualization



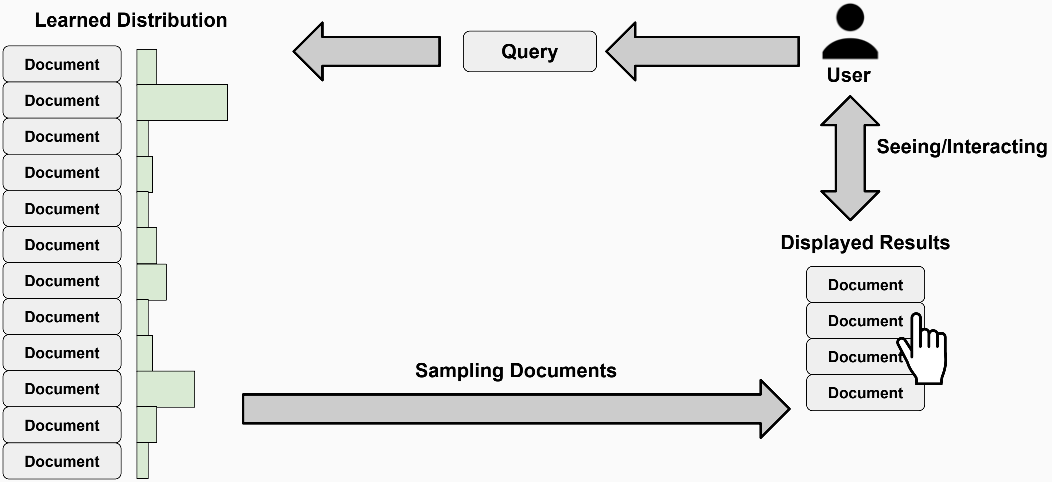
Pairwise Differentiable Gradient Descent: Visualization



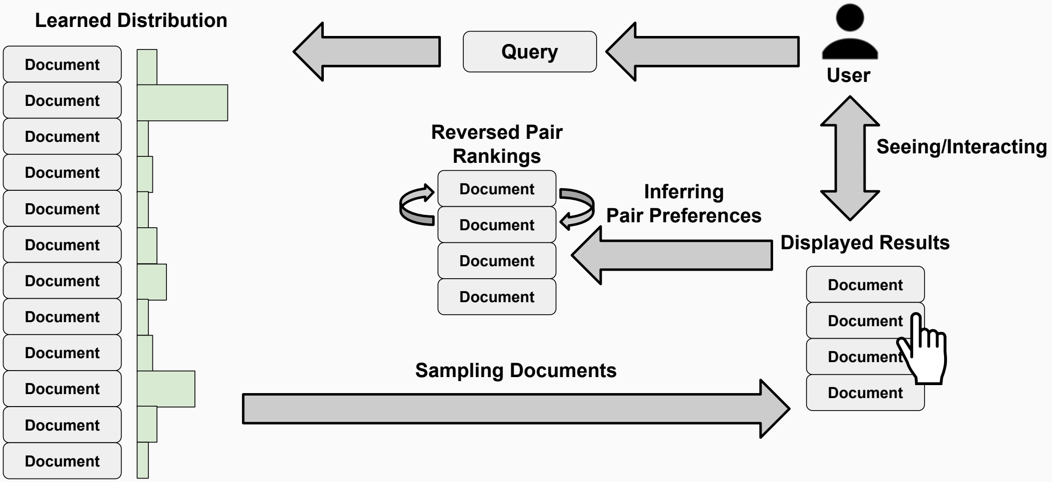
Pairwise Differentiable Gradient Descent: Visualization



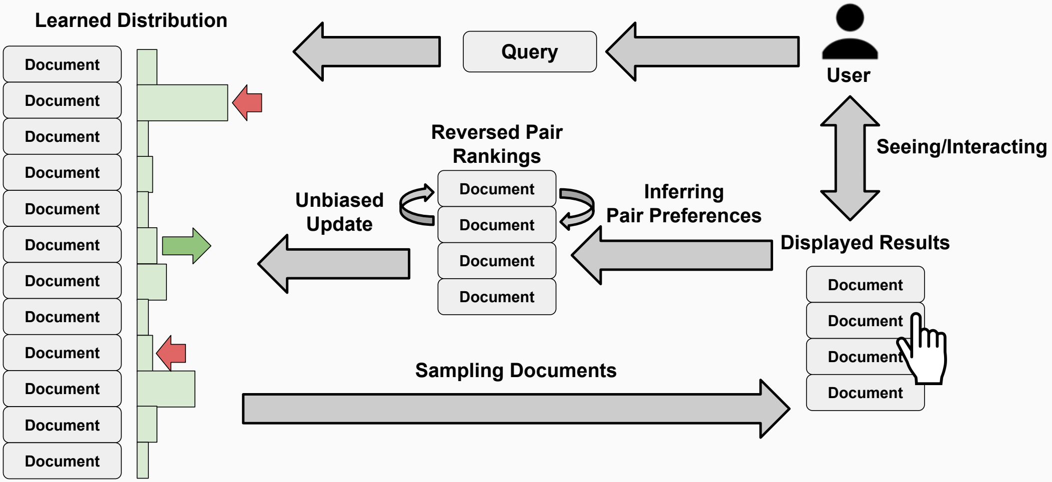
Pairwise Differentiable Gradient Descent: Visualization



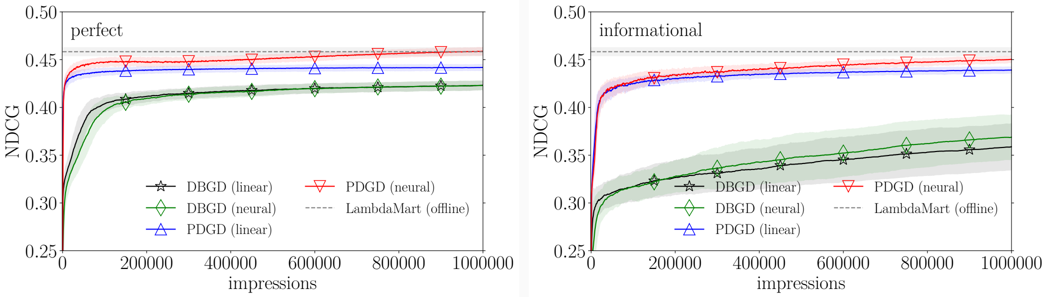
Pairwise Differentiable Gradient Descent: Visualization



Pairwise Differentiable Gradient Descent: Visualization



Pairwise Differentiable Gradient Descent: Results Long Term



Results of simulations on the MSLR-WEB10k dataset, a perfect user (left) and an informational user (right).

Image credits: (Oosterhuis and de Rijke, 2018b).

Comparison of Online Methods

Empirical Comparison: Introduction

Recent most generalized comparison so far (Oosterhuis and de Rijke, 2019).

Simulations based on **largest available industry datasets**:

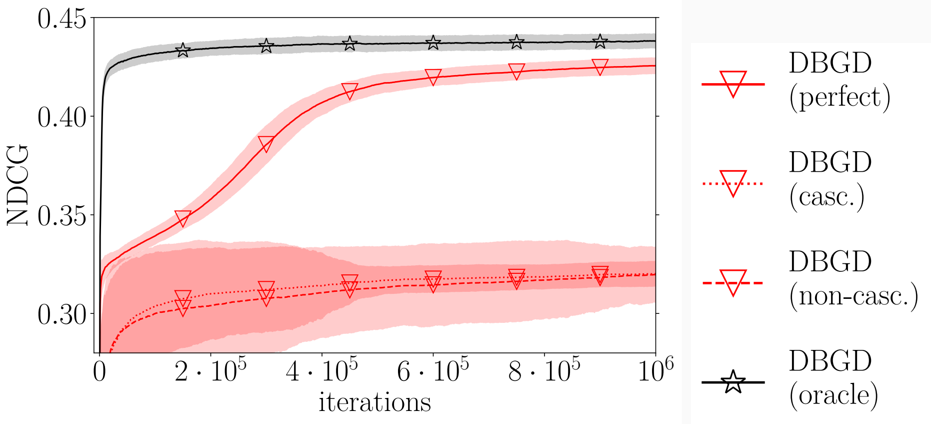
- MSLR-Web10k, Yahoo Webscope, Istella.

Simulated behavior ranging from:

- **ideal:** no noise, no position bias,
- **extremely difficult:** mostly noise, very high position bias.

Dueling Bandit Gradient Descent with an **oracle instead of interleaving**, to see the **maximum potential** of better interleaving methods.

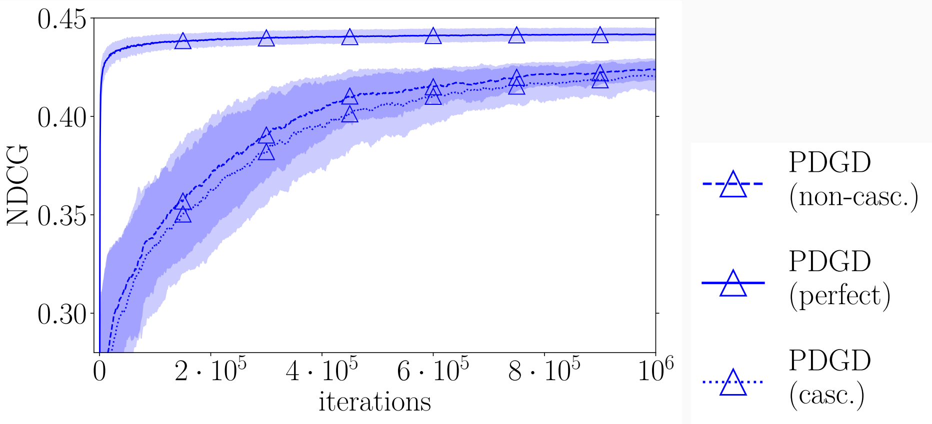
Empirical Comparison: DBGD



Results of simulations on the MSLR-WEB10k dataset.

Image credits: (Oosterhuis and de Rijke, 2019).

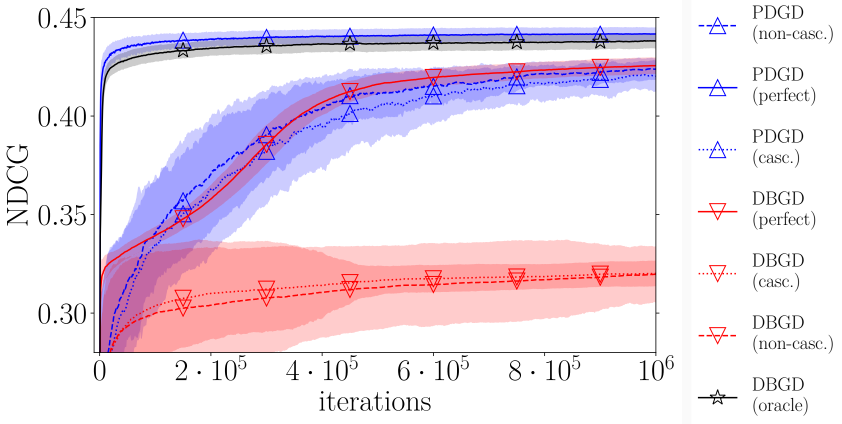
Empirical Comparison: PDGD



Results of simulations on the MSLR-WEB10k dataset.

Image credits: (Oosterhuis and de Rijke, 2019).

Empirical Comparison: All



Results of simulations on the MSLR-WEB10k dataset.

Image credits: (Oosterhuis and de Rijke, 2019).

Empirical Comparison: Conclusion

Dueling Bandit Gradient Descent (DBGD):

- **Unable** to reach **optimal performance** in **ideal** settings.
- Strongly affected by noise and position bias.

Pairwise Differentiable Gradient Descent (PDGD):

- **Capable** of reaching **optimal performance** in ideal settings.
- **Robust** to noise and position bias.
- Considerably **outperforms** DBGD in **all tested experimental settings**.

Theoretical Comparison

Dueling Bandit Based Approaches:

- Sublinear regret bounds proven,
  unsound for ranking problems as commonly applied.
- *Single update steps* are as **unbiased** as its **interleaving method**.

The Differentiable Pairwise Based Approach:

- **No regret bounds** proven.
- *Single update steps* are unbiased w.r.t. **pairwise document preferences**.

For the common ranking problem, neither approach has a theoretical advantage.

The Future for Online Learning to Rank

The **theory** for Online Learning to Rank is **inadequate** and needs **re-evaluation**.

The Dueling Bandit approach appears to be **lacking for optimizing ranking systems**.

Novel alternative approaches have high potential:

- Pairwise Differential Gradient Descent is a clear example.

Comparison of Online LTR with Supervised LTR

Comparison of Online LTR with Supervised LTR

Supervised LTR:

- Uses **manually annotated labels**.
- Optimization is a widely studied and very effective w.r.t. evaluation on annotated labels.
- Often unavailable for practitioners.

Online LTR:

- Learns from **direct interaction**:
  - Debiases by **randomization**.
- Ineffective when applied to historical data.
- Unbiased w.r.t. pairwise preferences.
- Not guaranteed to be unbiased w.r.t. ranking metrics.