The Online Approach to Learning to Rank

Harrie Oosterhuis

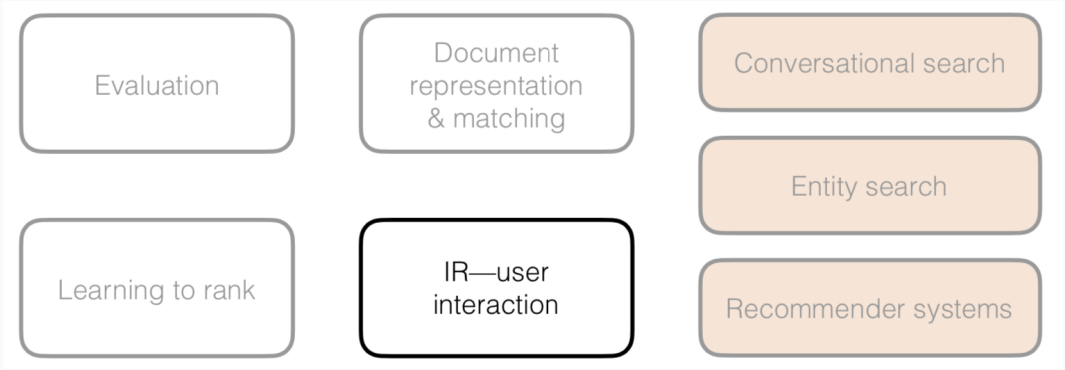
March 2, 2020

University of Amsterdam

oosterhuis@uva.nl

Introduction

Course Structure



Introduction

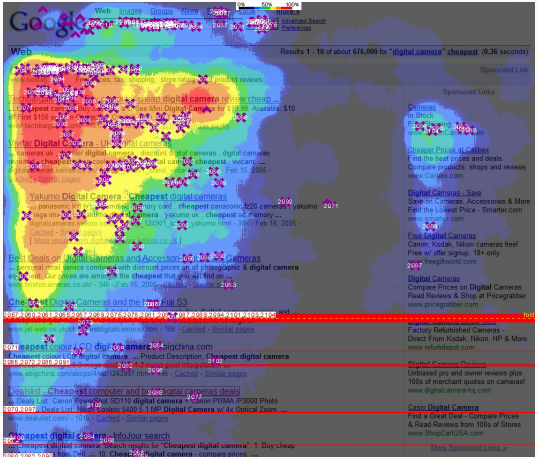
Last week we discussed **counterfactual evaluation/learning to rank**:

- Correcting for position bias with **inverse propensity scoring**.

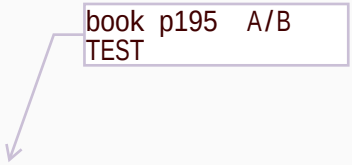
Today we will look at older **online evaluation/learning to rank**:

- Correcting for position bias through **randomization**.

Remember Position Bias?



Source: http://www.mediative.com/



A/B testing

Online Evaluation: Interleaving

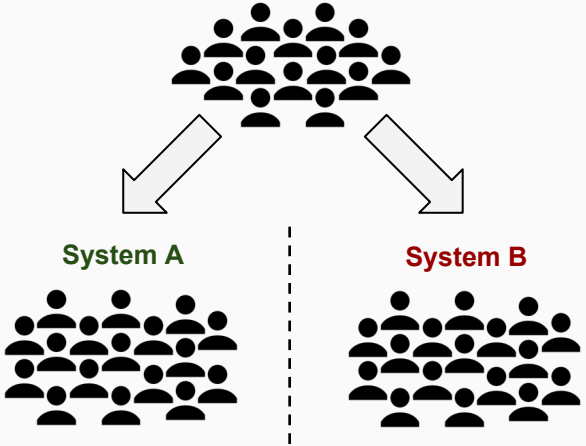
Often we need to answer the **question**:

Is ranker A be preferred over ranker B?

A/B testing is one of the most robust ways of testing this.

A/B testing

Split the users in two groups, one group is given system **A**, the other system **B**.
The **differences in behaviour** allows for a comparison of the systems.



A/B testing: Advantages/Disadvantages

Split the users in two groups, one group is given system **A**, the other system **B**.

Advantages:

- **Straightforward** and common method, also outside of IR.
- Can test **many aspects** of user behaviour.

Disadvantages:

- **Inefficient**, requires a lot of user data.
- Tests have to run for a **long time**.
- Need to recognize individual users.

Online Evaluation: The Idea

Online Evaluation

Online Evaluation methods have **control over what to display** to the user.

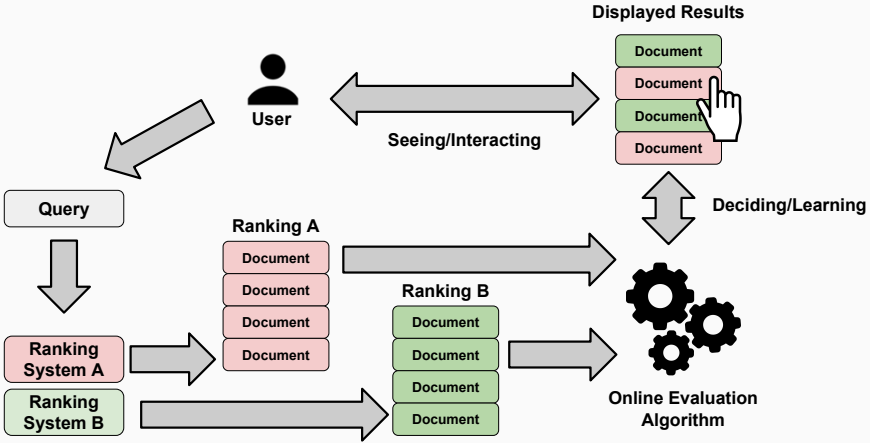
At the same time they:

- **Decide what results to display to the user.**
- **Infer preferences** from user interactions with the **chosen results**.

These methods can be **much more efficient**,
because they have (some) **control over what data is gathered**.

Online Evaluation: Visualization

Online Evaluation methods have **control over what to display** to the user.



Online Evaluation: Requirements

An **online evaluation** algorithm should:

- Give **reliable** and **correct** results:
  - **Guarantee** to provide correct comparison outcomes.
    i.e. theoretical proof.
  - **Robust** to interaction **noise**.
  - Handle **biases** in user interactions.
- Provide **good user experience**:
  - Methods should **not interfere** with user task.

Team-Draft Interleaving

Online Evaluation: Interleaving

To answer the **question**:

Is ranker A be preferred over ranker B?

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

Interleaving (Joachims, 2003):

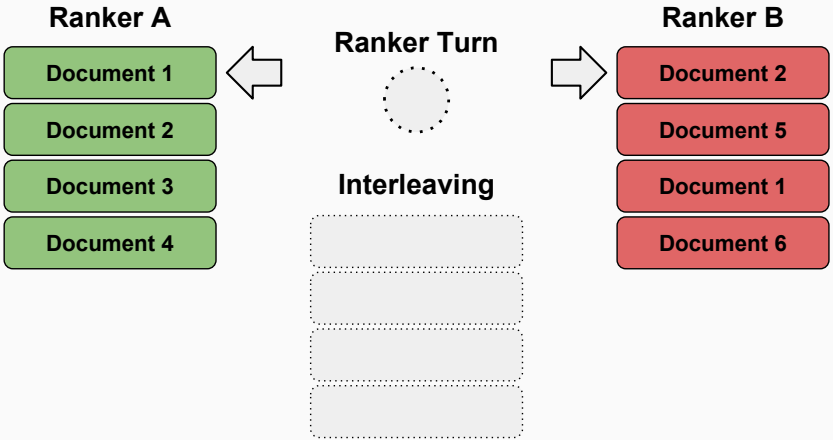
- Take the two rankings for a query from two rankers .
- Create an **interleaved ranking**, based on both rankings.
- **Clicks** on an interleaved ranking provide **preference signals** between rankers.

Team-Draft Interleaving: Algorithm Simple

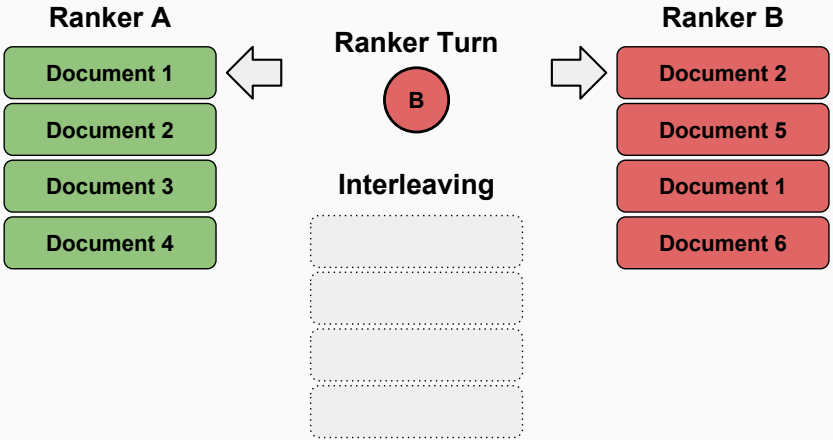
In plain English:

1. Until k documents are placed:
2.
   1. Randomly choose ranker **A** or **B**.
   2. Let **chosen ranker** place its **next** *unplaced* document.
   3. Let **other ranker** place its **next** *unplaced* document.
   4. Remember which ranker placed which document.
3. Present interleaving to user, observe clicks.
4. Ranker with the most clicks on its placed document wins.

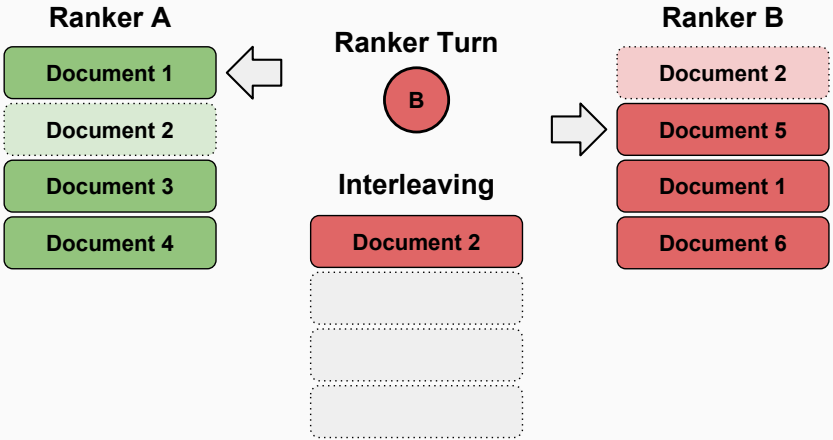
Team-Draft Interleaving: Visualization



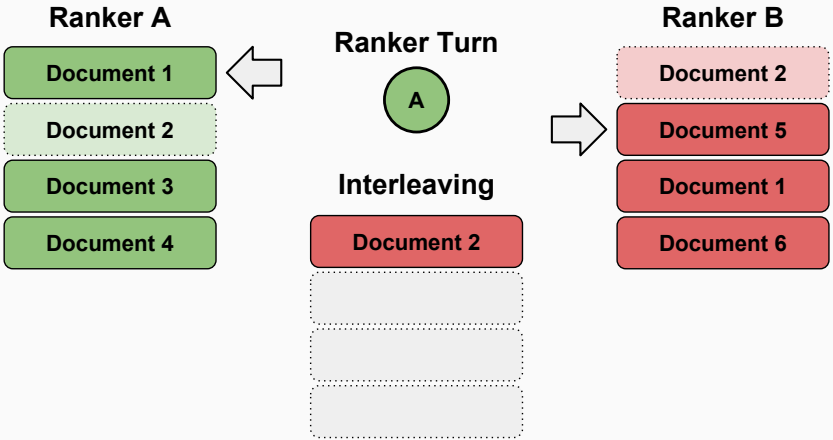
Team-Draft Interleaving: Visualization



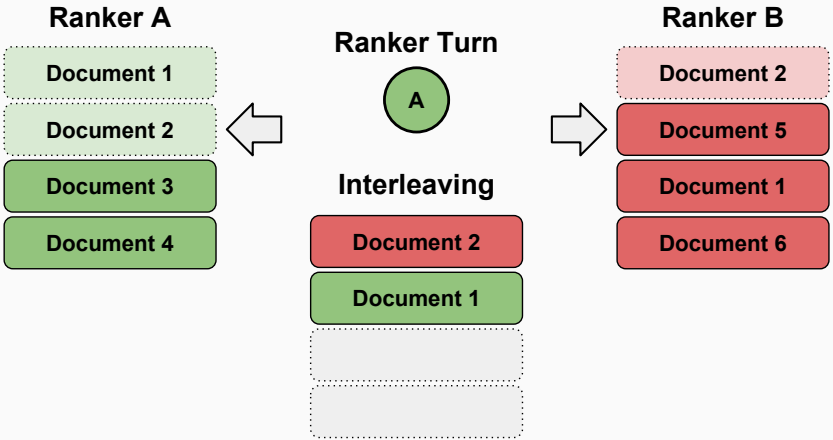
Team-Draft Interleaving: Visualization



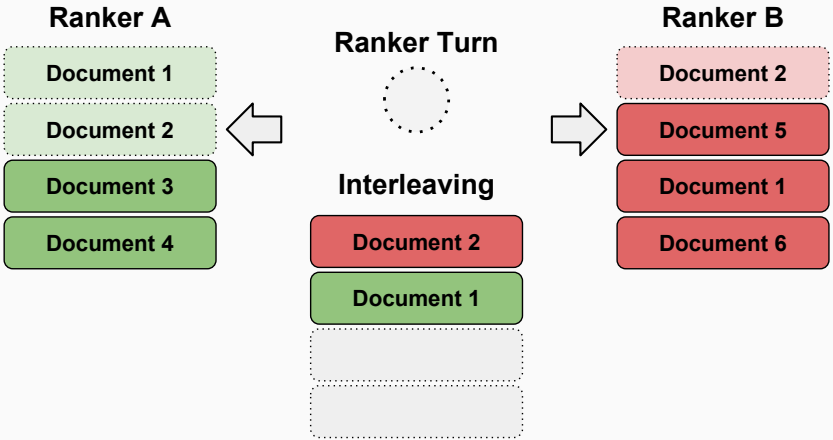
Team-Draft Interleaving: Visualization



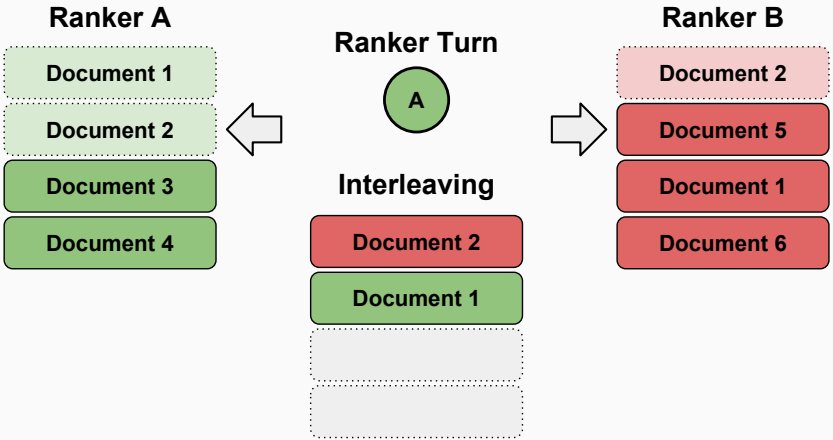
Team-Draft Interleaving: Visualization



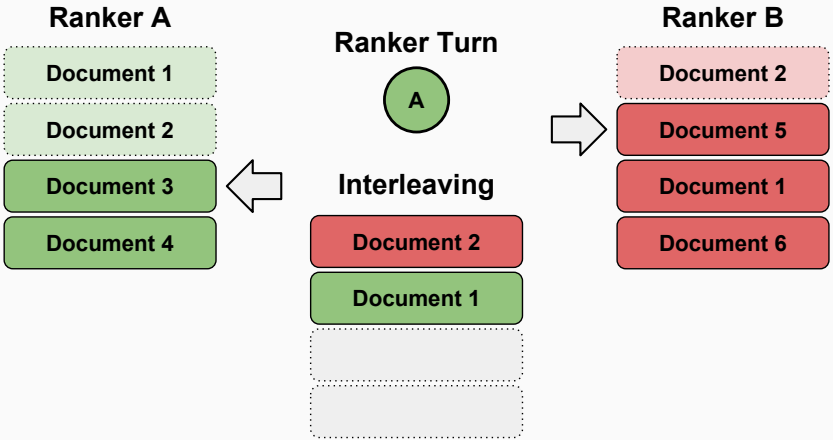
Team-Draft Interleaving: Visualization



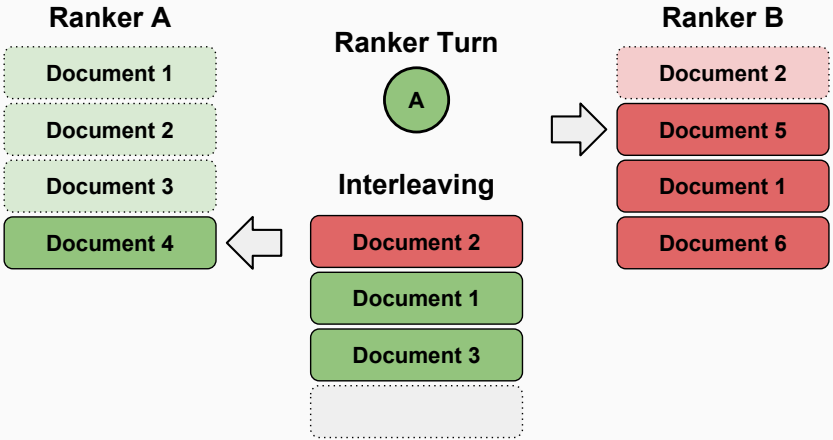
Team-Draft Interleaving: Visualization



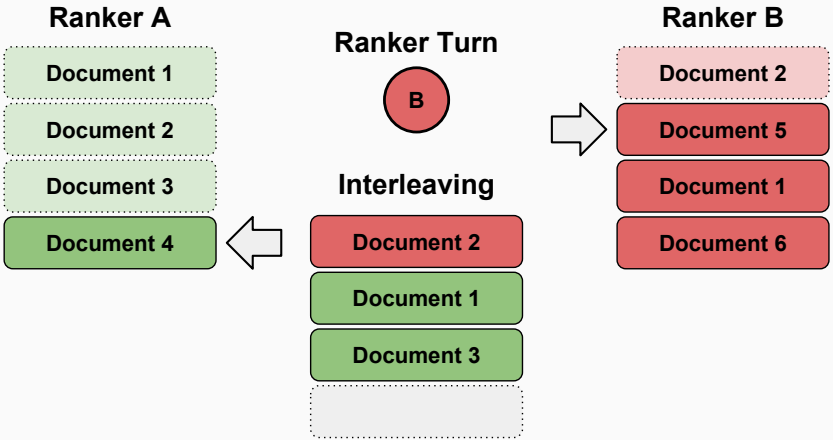
Team-Draft Interleaving: Visualization



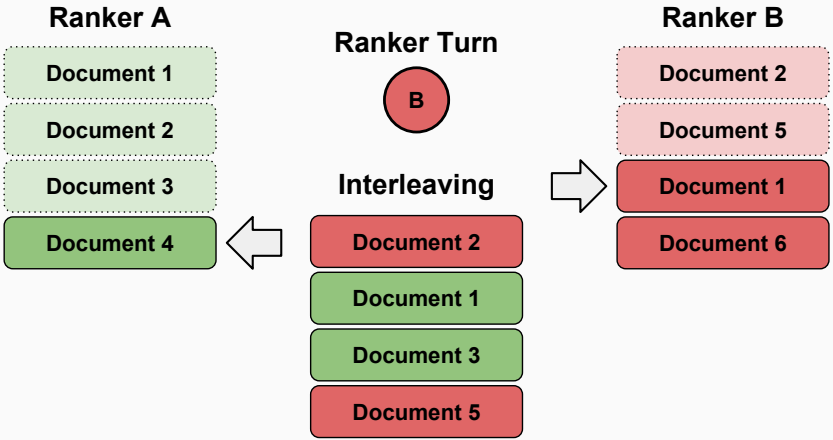
Team-Draft Interleaving: Visualization



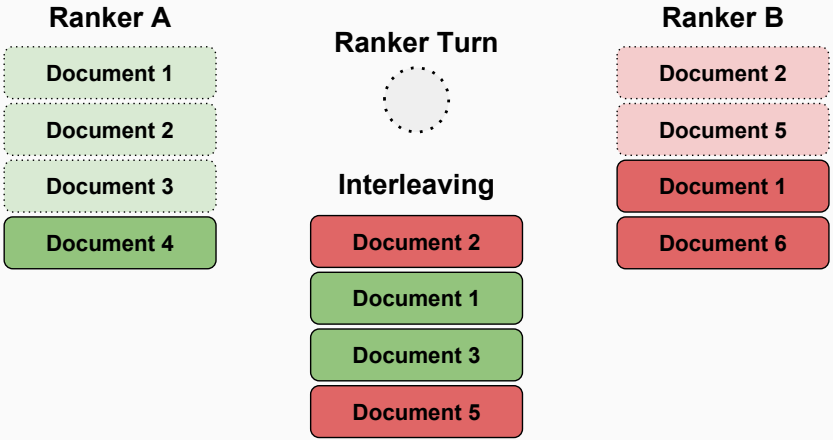
Team-Draft Interleaving: Visualization



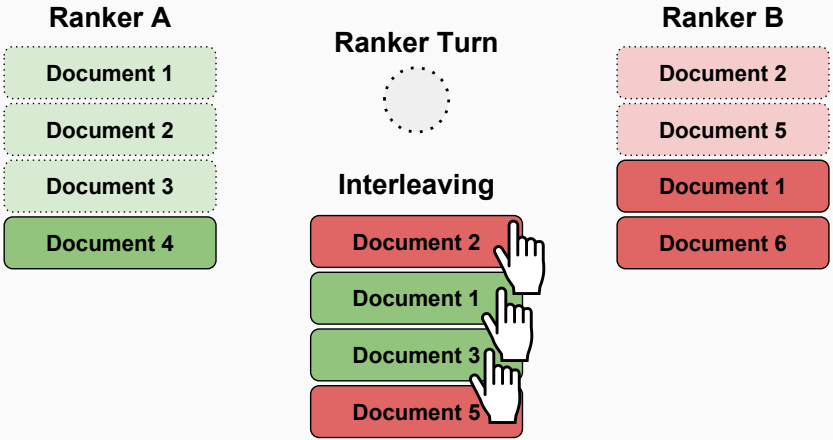
Team-Draft Interleaving: Visualization



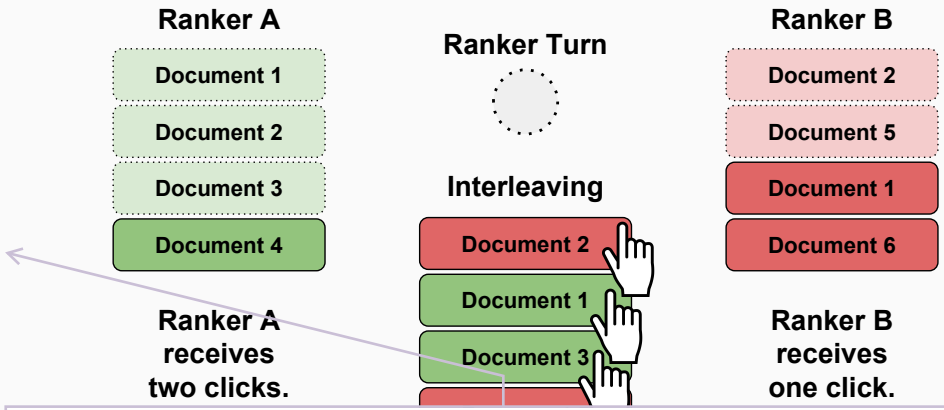
Team-Draft Interleaving: Visualization



Team-Draft Interleaving: Visualization



Team-Draft Interleaving: Visualization

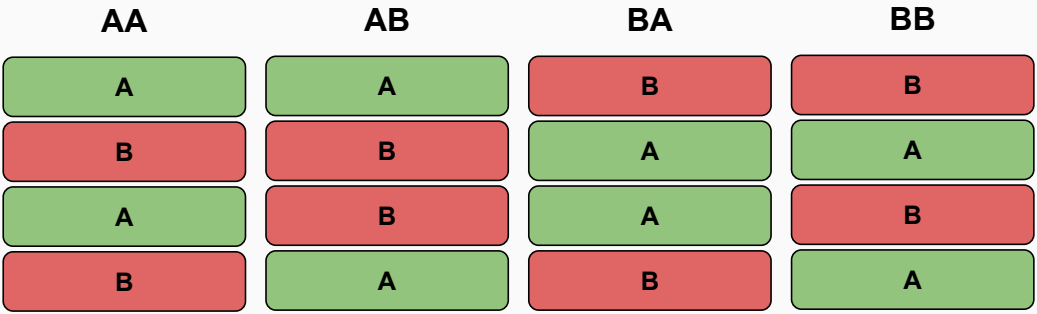


ranker A receives more clicks than B, so we conclude ranker A is better than B. interleaving is for evaluate how good/ bad the new ranker is.
to remove noise, we repeat this experiment many times, each time flip a coin from rank1, generate a new interleaving list. compute mean

Team-Draft Interleaving: Resolved Example

Team-Draft interleaving finds **no preferences** under **random** clicks.

All possible assignments:



Team-Draft Interleaving: Comparison to A/B testing

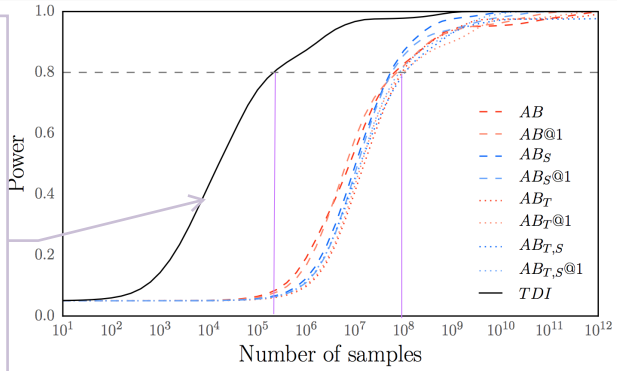
From (Schuth et al., 2015), power is an indication of sensitivity.

compare diff between two methods

AB test needs 10^8 samples

Term Draft Interleaving nees 10^5 samples.

so TDI is more efficient than ABtest. so interleaving is more effective than AB test



igure 1: Power as a function of sample size, computed using e observed effect sizes for 38 interleaving and AB comparons, averaged over all comparisons (assuming two-sided t-test with $p = 0.05$, as described in Section 4.2).

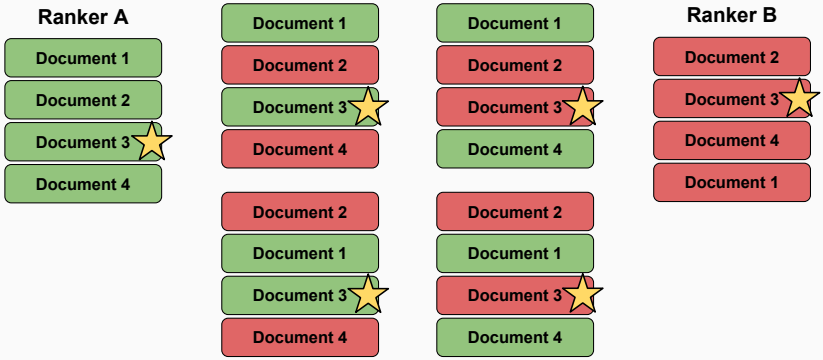
Team-Draft Interleaving: Properties

Properties of Team-Draft interleaving:

- **User experience**:
  - User **hardly affected** by method.
  - Experience will not be worse than that of the worst ranker.
- **Correctness**:
  - Ranker that places relevant documents **usually wins**.
  - **Correct outcomes not guaranteed.**

Team-Draft Interleaving: Problematic Example

Note this example, where document 3 is the **only relevant one**.



Ranker **B** should win, but **in expectation no preference will be found**.

Probabilistic Interleaving

Probabilistic Interleaving

Introduced by Hofmann et al. (2011) designed around the **fidelity conditions**.

Treats rankers as **probability distributions** over a set of documents.

Probabilistic Interleaving: Rankers as Probability Distributions

A ranker **A** with the ranking:

$$R_A(D) = [d_1, d_2, \ldots d_N] \tag{1}$$

then let $rank(d, R_A)$ be the rank of d in R_A.

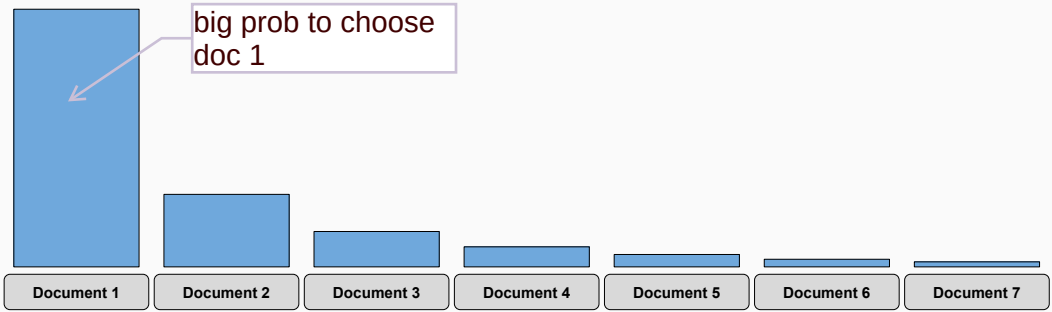
The distribution for ranker **A** is modelled by (with $\tau \in \mathbb{R}_{>0}$):

$$P(d|D, R_A) = P_A(d) = \frac{\frac{1}{rank(d,R_A)^\tau}}{\sum_{d' \in D} \frac{1}{rank(d',R_A)^\tau}}$$

doc d在 rank list RA 中的排位越高，则 prob of choosing d 越大

Renormalize after each document is removed, i.e. remove sampled document from D.

Probabilistic Interleaving: Rankers as Probability Distributions



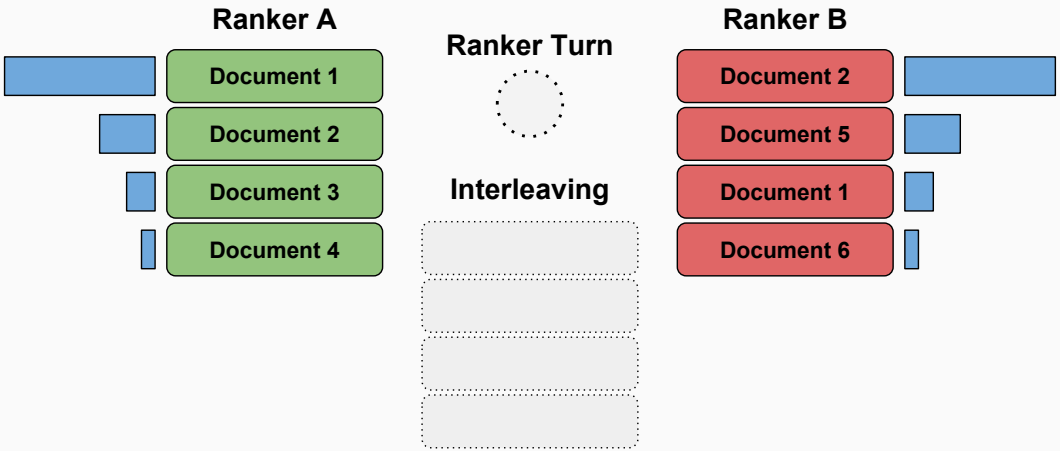
Example of a possible **document distribution**.

Probabilistic Interleaving: Proto-Method

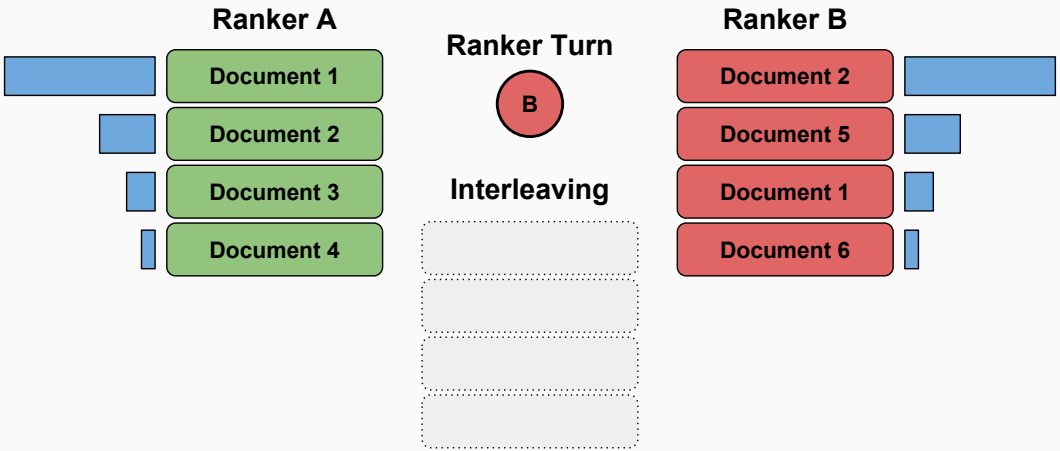
Consider this proto-version of Probabilistic Interleaving:

1. Compute P_A and P_B from ranker **A** and **B** respectively.
2. Repeat until k documents placed:
   1. Randomly choose P_A or P_B and **sample a document** d.
   2. Place d and remember whether **A** or **B** was chosen.
   3. Renormalize P_A or P_B after removing d.
3. Display to user and observe clicks.
4. Ranker with the most clicked documents wins comparison.

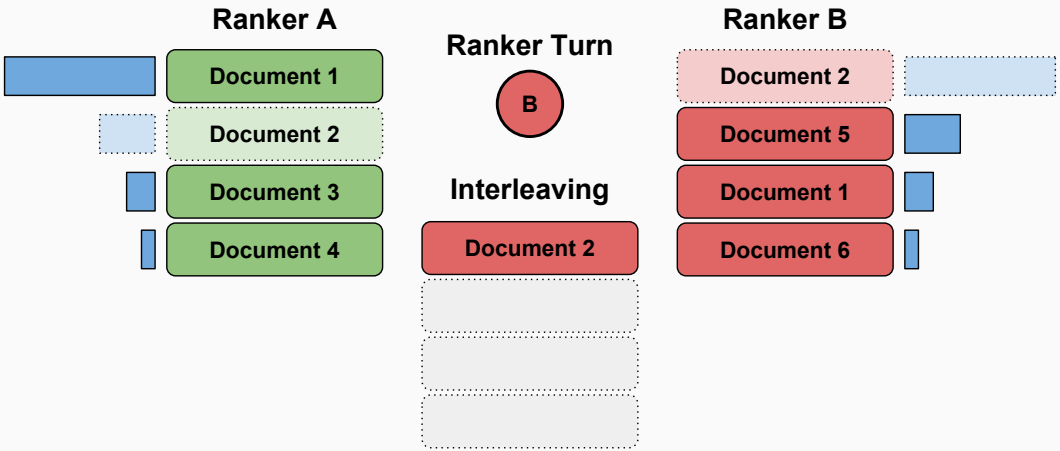
Probabilistic Interleaving: Proto-Method Visualization



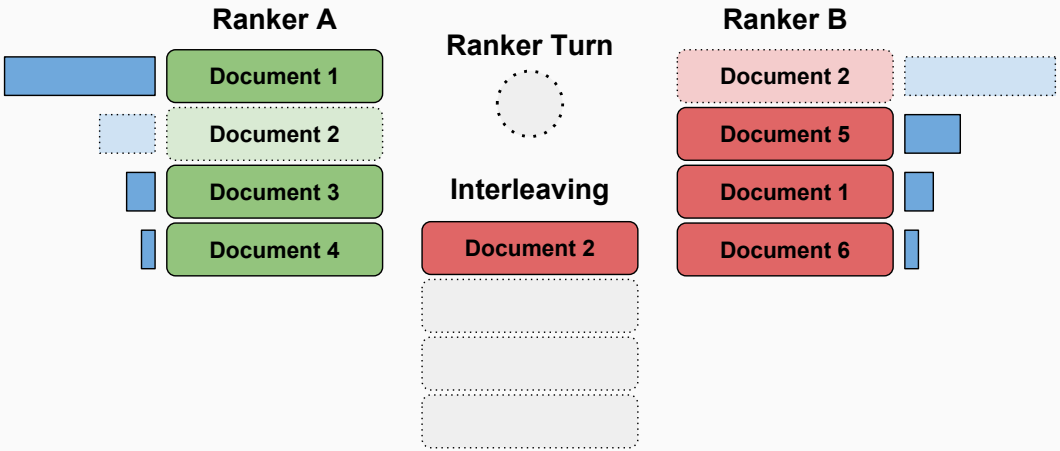
Probabilistic Interleaving: Proto-Method Visualization



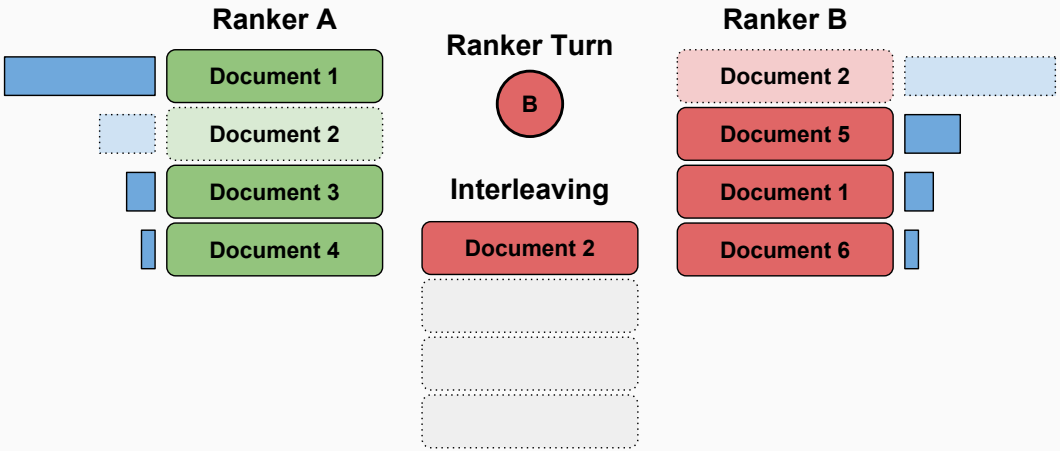
Probabilistic Interleaving: Proto-Method Visualization



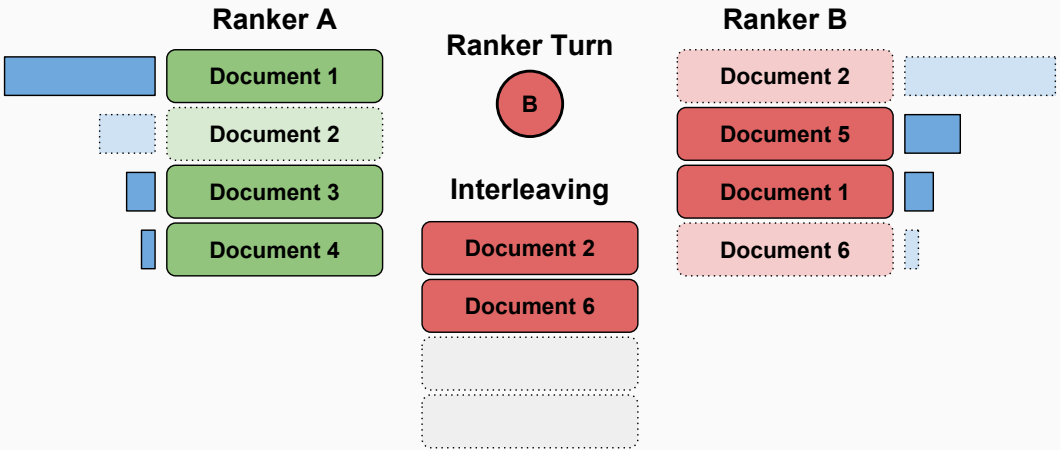
Probabilistic Interleaving: Proto-Method Visualization



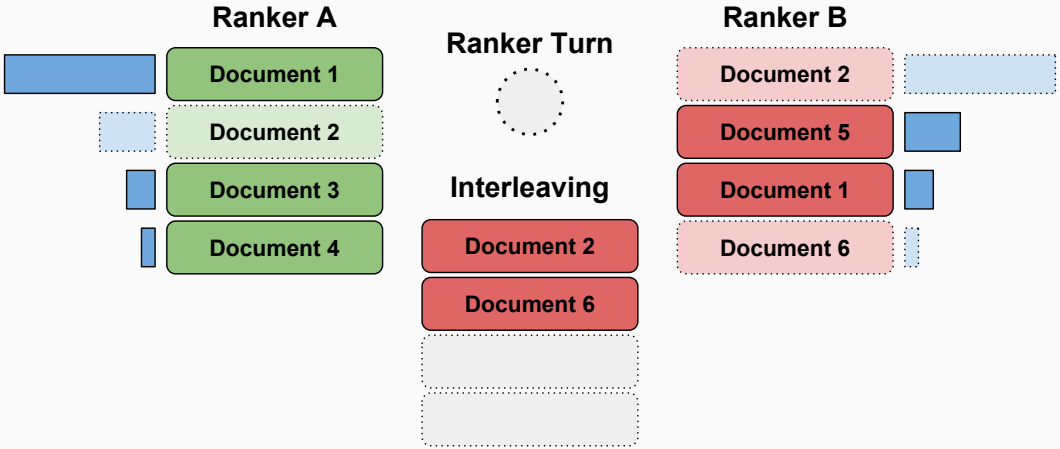
Probabilistic Interleaving: Proto-Method Visualization



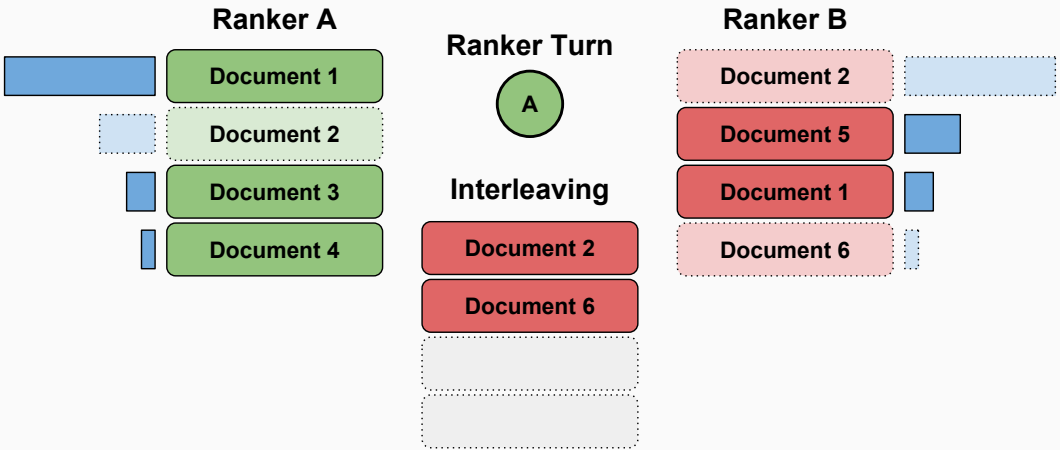
Probabilistic Interleaving: Proto-Method Visualization



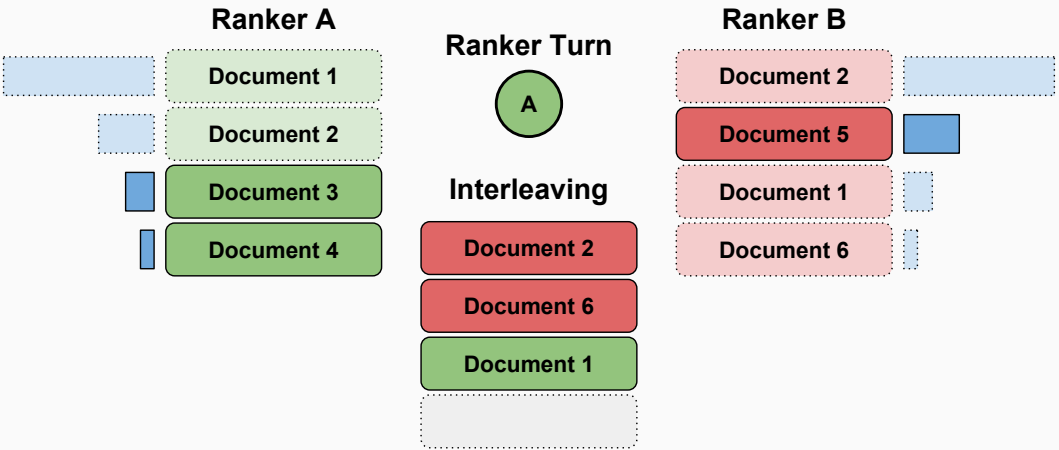
Probabilistic Interleaving: Proto-Method Visualization



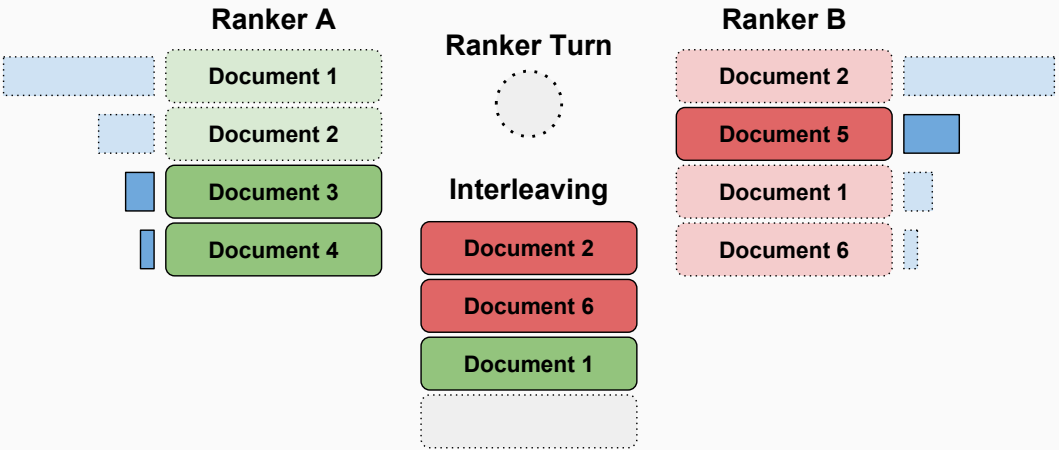
Probabilistic Interleaving: Proto-Method Visualization



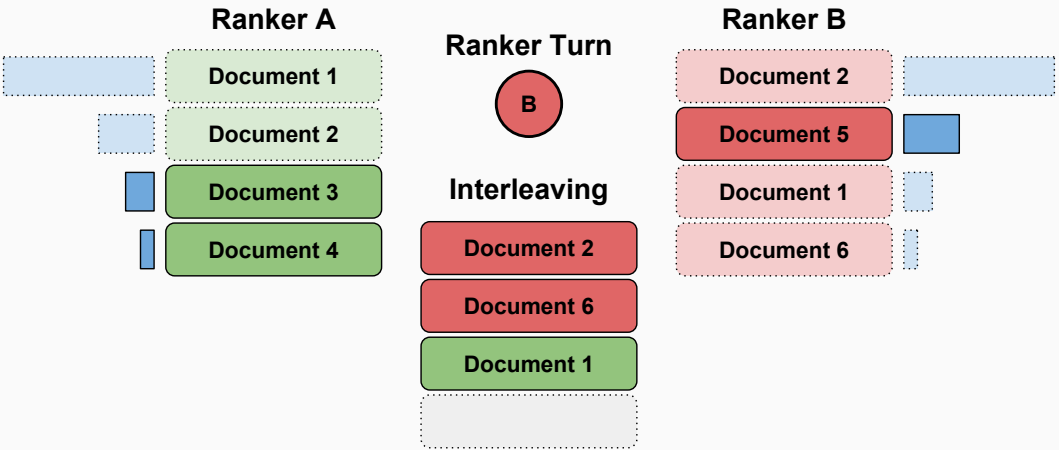
Probabilistic Interleaving: Proto-Method Visualization



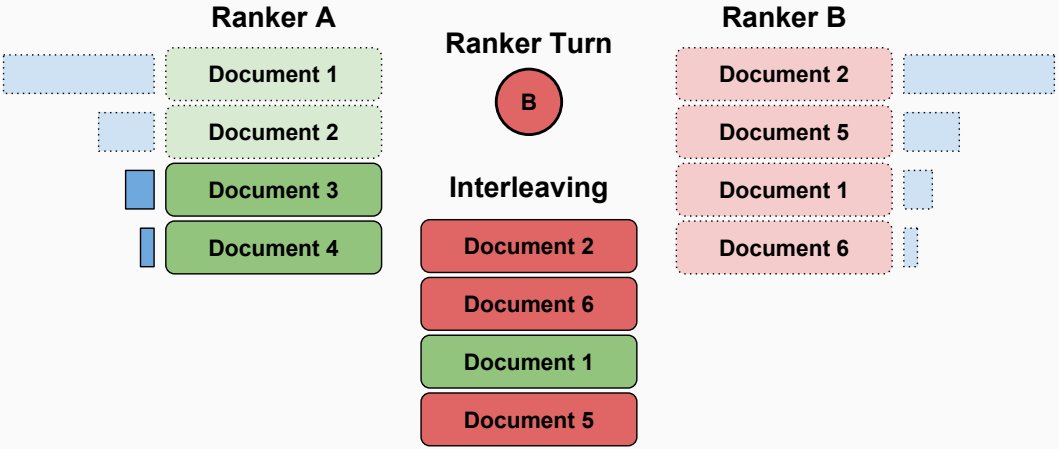
Probabilistic Interleaving: Proto-Method Visualization



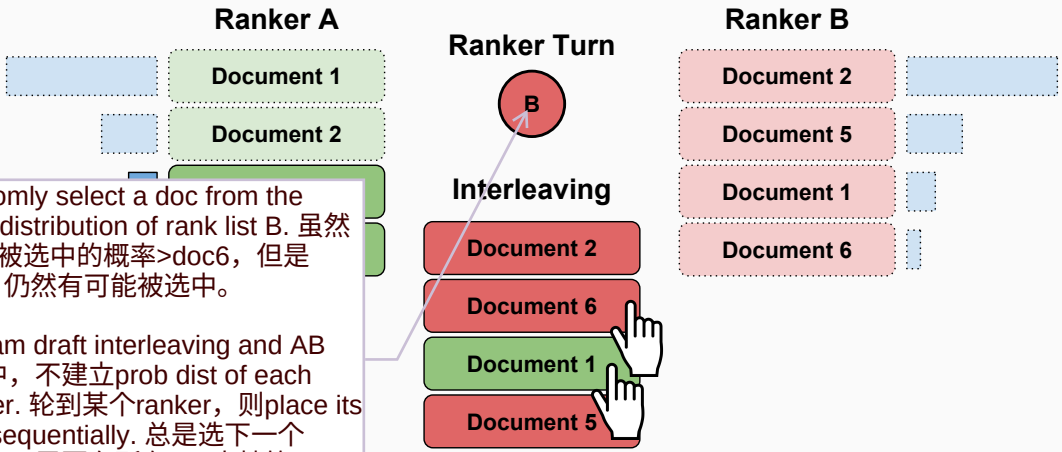
Probabilistic Interleaving: Proto-Method Visualization



Probabilistic Interleaving: Proto-Method Visualization



Probabilistic Interleaving: Proto-Method Visualization



randomly select a doc from the prob distribution of rank list B. 虽然doc5被选中的概率>doc6，但是doc6 仍然有可能被选中。

in team draft interleaving and AB test中，不建立prob dist of each ranker. 轮到某个ranker，则place its doc sequentially. 总是选下一个doc，不需要在所有doc中抽签

Probabilistic Interleaving: Proto-Method Fidelity

Does this method have Fidelity?

1. Could a ranker have an **advantage** due to factors **other than relevance**?
   - Every ranker is equally likely to place at every rank.
   - Thus expected number of clicks is equal under random interactions.
2. Will an **unambiguous winner** always win in expectation?
   - Every ranker is equally likely to place at every rank.
   - Dominating ranker is more likely to place relevant documents at each rank.
   - If relevant documents are more likely to be clicked, then the dominating ranker wins in expectation.

Quite trivial to show that **this method has fidelity**.

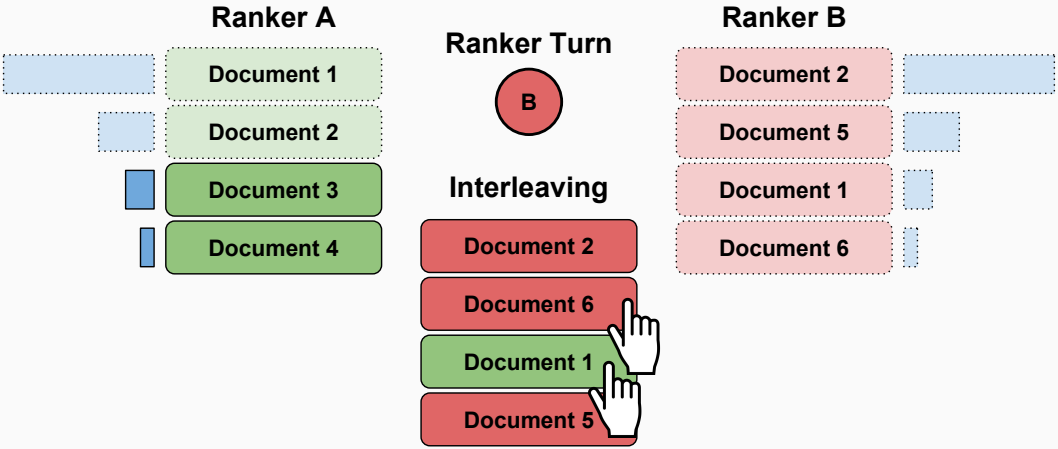
Probabilistic Interleaving: Marginalization

The user **does not see** which ranker placed what documents,
thus their behaviour will **not be affected** by document assignments.

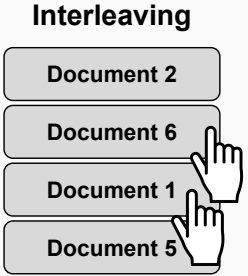
Probabilistic interleaving takes the proto-method and **marginalizes over the assignments**:

- Instead of using the outcome based on the *true* ranker assignment, calculate the **expected outcome over all possible assignments**.

Probabilistic Interleaving: Expected Outcome Visualized



Probabilistic Interleaving: Expected Outcome Visualized



Probabilistic Interleaving: Expected Outcome

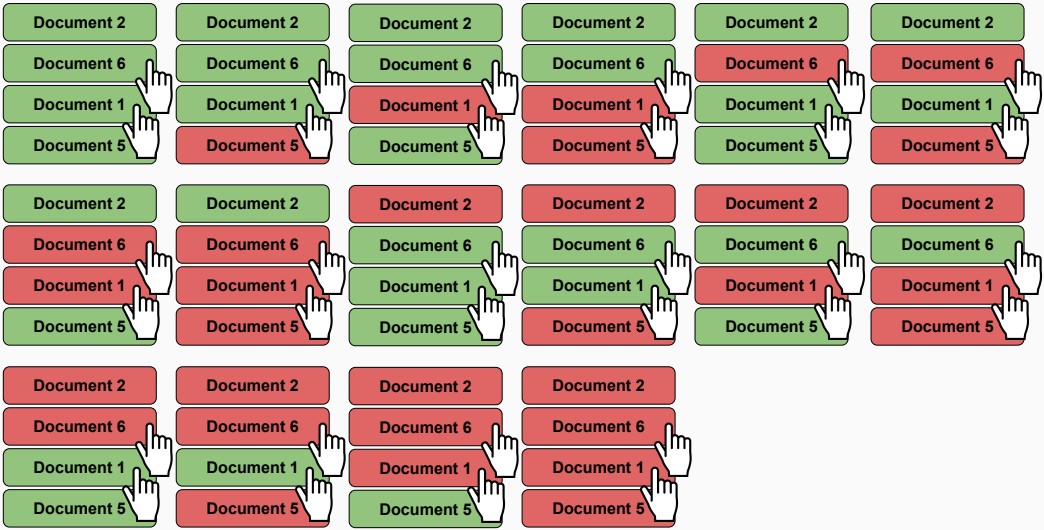
For rankings R_A, R_B, the interleaved list L, assignments T, and clicks c,
the **outcome of a comparison** can be noted as:

$$O(R_A, R_B, L, T, c) \in \{-1, 0, 1\} \tag{3}$$

Since clicks are **independent** of the assignment T,
we can **marginalize over all possible assignments** to reach an **expected outcome**:

$$E[O(R_A, R_B, L, c)] = \sum_T P(T|R_A, R_B, L) O(R_A, R_B, L, T, c) \tag{4}$$

Probabilistic Interleaving: Expected Outcome Visualized



Probabilistic Interleaving: Placement Probability

Thus we can calculate the placement probability for each document.:

$$P(T_i = A) = \frac{1}{2} \tag{5}$$

$$P(L_i = d|T_i = A) = P_A(d) = \frac{\frac{1}{rank(d,R_A)^\tau}}{\sum_{d' \in D} \frac{1}{rank(d',R_A)^\tau}} \tag{6}$$

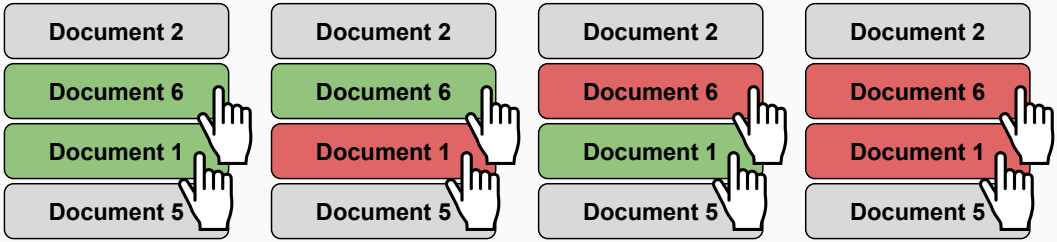
$$P(T_i = A|L_i = d) = \frac{P_A(d)}{P_A(d) + P_B(d)} \tag{7}$$

Two important observations:

- The outcome of a comparison is **only dependent on the clicked documents**.
- The assignment of a document is **not dependent on other assignments**.

Probabilistic Interleaving: Smarter Marginalization

Thus we only have to consider the possible assignments of clicked documents:



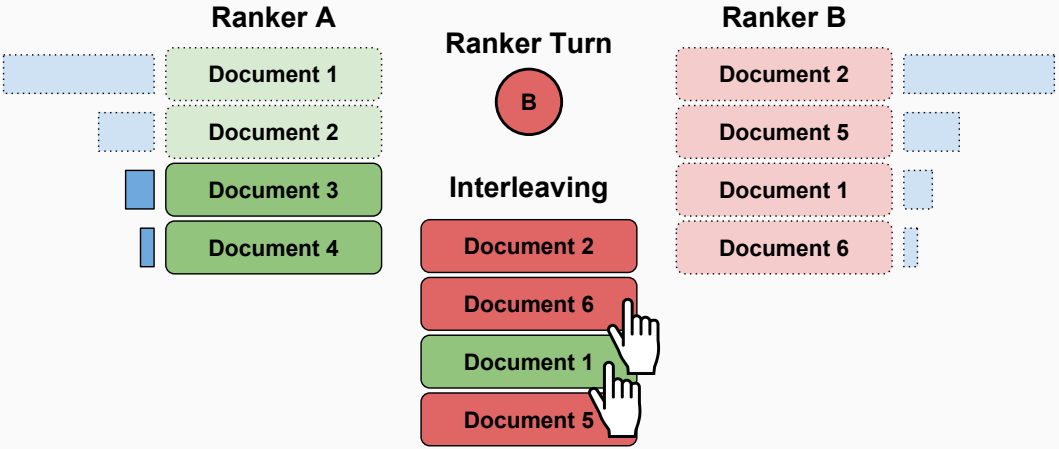
Bringing the complexity from 2^k to 2^c.

Probabilistic Interleaving: Method

This gives us the following method:

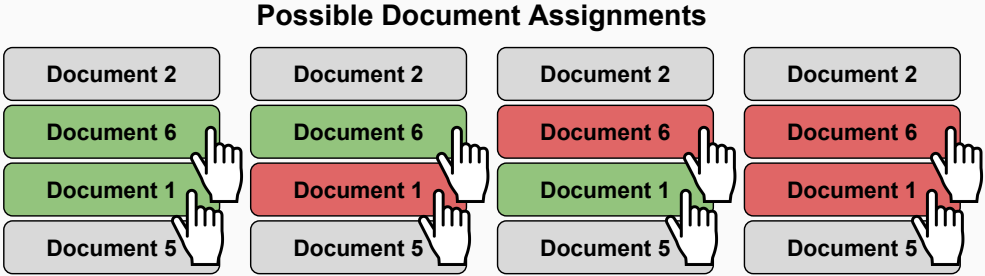
1. Compute P_A and P_B from ranker **A** and **B** respectively.
2. Repeat until k documents placed:
   1. Randomly choose P_A or P_B and sample a document d.
   2. Place d **without remembering** whether **A** or **B** was chosen.
   3. Renormalize P_A or P_B after removing d.
3. Display to user and observe clicks.
4. Calculate the **expected outcome** marginalizing over all possible placements.
5. Expected winner wins the comparison.

Probabilistic Interleaving: Visualization



Probabilistic Interleaving: Visualization

Probabilistic Interleaving: Visualization



Probabilistic Interleaving: Properties

Properties of Probabilistic Interleaving:

- **Correctness**:
  - Correct outcomes guaranteed w.r.t. fidelity.
  - **Marginalization does not affect the expected outcomes.**
  - Thus if proto-method has fidelity, so has this method.
- **User experience**:
  - User experience not guaranteed.
  - **Every possible ranking can be displayed**, albeit with different probabilities.

Optimized Interleaving

Optimized Interleaving

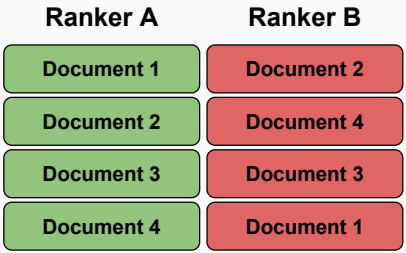
Introduced by Radlinski and Craswell (2013) casts interleaving as an **optimization problem**.

Interleavings should **only contain top-documents** from rankers, i.e. rankers should always add their top document.

First step: determine the set of **allowed interleavings**.

Optimized Interleaving: Allowed Interleavings

Allowed Interleavings



Optimized Interleaving: Allowed Interleavings

Allowed Interleavings

Document 1	Document 2

if len of final result=1
then there are only two possible results

在rankerA,B中，不在第一排位的doc，不可能被放在result中的第一排位。

probabilistic interleaving 会出现 bottom ranked doc is put on the top rank in the result. we dont want this to happen

Ranker A	Ranker B
Document 1	Document 2
Document 2	Document 4
Document 3	Document 3
Document 4	Document 1

Optimized Interleaving: Allowed Interleavings

Allowed Interleavings

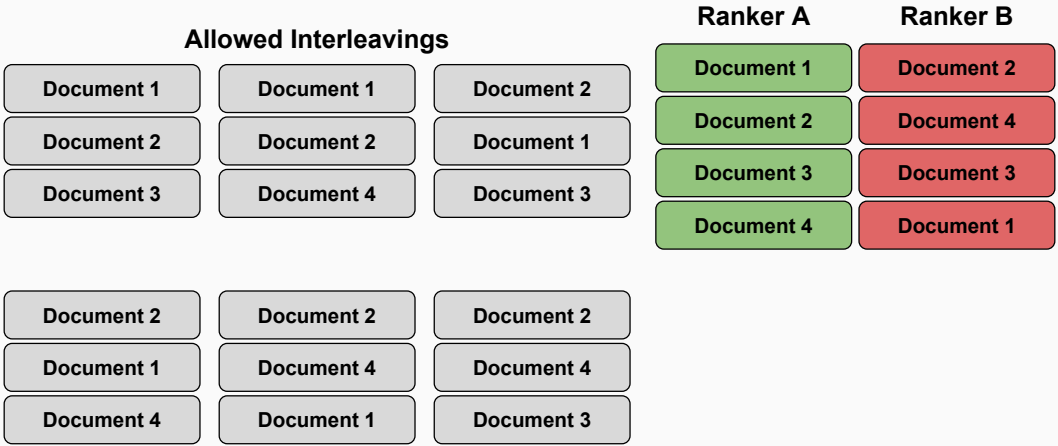
Document 1	Document 2	Document 2
Document 2	Document 1	Document 4

Ranker A	Ranker B
Document 1	Document 2
Document 2	Document 4
Document 3	Document 3
Document 4	Document 1

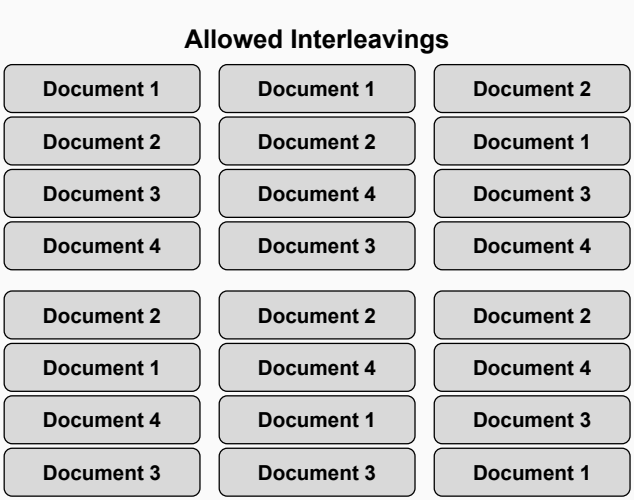
if len of final result=1
then there are only two possible results.

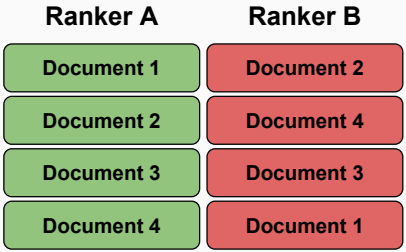
in result list, doc 4 一定ranked after doc2. 因为 both ranker A and B agrees on this

Optimized Interleaving: Allowed Interleavings



Optimized Interleaving: Allowed Interleavings





Optimized Interleaving: Scoring Function

Optimized interleaving can use **different scoring functions** that meet its requirements. A click on a document d gives a **preference score** determined by how its ranked.

Common choices are:

不管用哪个score公式，总之，每一个prob of a click must have a weight. this weight is computed by scoring func

1. **Linear Rank Difference**:

$$s(d, R_A, R_B) = \delta_d = \mathit{rank}(d, R_A) - \mathit{rank}(d, R_B) \tag{8}$$

2. **Inverse Rank Difference**:

$$s(d, R_A, R_B) = \delta_d = \frac{1}{\mathit{rank}(d, R_A)} - \frac{1}{\mathit{rank}(d, R_B)} \tag{9}$$

Optimized Interleaving: Scoring Function Example

Given two rankers where $R_A = [1, 2, 3, 4]$ and $R_B = [2, 4, 3, 1]$ and using the *Linear Rank Difference* scoring function, we see:

Interleaving	δ_{L_1}	δ_{L_2}	δ_{L_3}	δ_{L_4}
$[1, 2, 3, 4]$	3	-1	0	-2
$[1, 2, 4, 3]$	3	-1	-2	0
$[2, 1, 3, 4]$	-1	3	0	-2
$[2, 1, 4, 3]$	-1	3	-2	0
$[2, 4, 1, 3]$	-1	-2	3	0
$[2, 4, 3, 1]$	-1	-2	0	3

Optimized Interleaving: Position Bias Assumption

Let's assume that a user is **only position biased**, that means that only the position determines the **click probability**:

$$P(click(position))$$

Optimized Interleaving: Scoring Function Example

Given two rankers where $R_A = [1, 2, 3, 4]$ and $R_B = [2, 4, 3, 1]$ and using the *Linear Rank Difference* scoring function, **what distribution over interleavings should be chosen?**

Interleaving	δ_{L_1}	δ_{L_2}	δ_{L_3}	δ_{L_4}	$E[O]$	p_L
$[1, 2, 3, 4]$	3	-1	0	-2	$3P(click(1)) - P(click(2)) - 2P(click(4))$	
$[1, 2, 4, 3]$	3	-1	-2	0	$3P(click(1)) - P(click(2)) - 2P(click(3))$	
$[2, 1, 3, 4]$	-1	3	0	-2	$-P(click(1)) + 3P(click(2)) - 2P(click(4))$	
$[2, 1, 4, 3]$	-1	3	-2	0	$-P(click(1)) + 3P(click(2)) - 2P(click(3))$	
$[2, 4, 1, 3]$	-1	-2	3	0	$-P(click(1)) - 2P(click(2)) + 3P(click(3))$	
$[2, 4, 3, 1]$	-1	-2	0	3	$-P(click(1)) - 2P(click(2)) + 3P(click(4))$	

Optimized Interleaving: Optimization for Bias

If we take p_L for the **probability** of interleaving L being **displayed**, then the **expected outcome** can be written as:

$$E[O] = \sum_{L \in \mathcal{L}} \left(p_L \sum_{i=1}^{|L|} P(click(i)) s(L_i, R_A, R_B) \right) = 0 \tag{10}$$

This becomes a **linear optimization** (or linear programming) task to find a p_L to meet this **requirement**.

Optimized Interleaving: Scoring Function Example

根据 上一张slide sum =0 (linear progrmaming)计算得出PL
we have 25% chance to show the 2nd interleaving. 35% chance to show the 4th interleaving

Given two rankers where $R_A = [1, 2, 3, 4]$ and $R_B = [2, 4, 3, 1]$ and using the *Linear Rank Difference* scoring function, a possible solution is:

Interleaving	δ_{L_1}	δ_{L_2}	δ_{L_3}	δ_{L_4}	$E[O]$	p_L
$[1, 2, 3, 4]$	3	-1	0	-2	$3P(click(1)) - P(click(2)) - 2P(click(4))$	0%
$[1, 2, 4, 3]$	3	-1	-2	0	$3P(click(1)) - P(click(2)) - 2P(click(3))$	25%
$[2, 1, 3, 4]$	-1	3	0	-2	$-P(click(1)) + 3P(click(2)) - 2P(click(4))$	0%
$[2, 1, 4, 3]$	-1	3	-2	0	$-P(click(1)) + 3P(click(2)) - 2P(click(3))$	35%
$[2, 4, 1, 3]$	-1	-2	3	0	$-P(click(1)) - 2P(click(2)) + 3P(click(3))$	40%
$[2, 4, 3, 1]$	-1	-2	0	3	$-P(click(1)) - 2P(click(2)) + 3P(click(4))$	0%

3 表示 doc1 is ranked 3 positions higher in RA than in RB

0 表示 doc3 is ranked at the same position in RA and RB

Optimized Interleaving: Scoring Function Example

Given two rankers where $R_A = [1, 2, 3, 4]$ and $R_B = [2, 4, 3, 1]$ and using the *Linear Rank Difference* scoring function, a possible solution is:

Interleaving	δ_{L_1}	δ_{L_2}	δ_{L_3}	δ_{L_4}	$E[O]$	p_L
$[1, 2, 4, 3]$	3	-1	-2	0	$3P(click(1)) - P(click(2)) - 2P(click(3))$	25%
$[2, 1, 4, 3]$	-1	3	-2	0	$-P(click(1)) + 3P(click(2)) - 2P(click(3))$	35%
$[2, 4, 1, 3]$	-1	-2	3	0	$-P(click(1)) - 2P(click(2)) + 3P(click(3))$	40%

$$P(click(1))(3 \times 0.25 - 1 \times 0.35 - 1 \times 0.4) = 0$$
$$P(click(2))(-0.25 + 3 \times 0.35 - 2 \times 0.4) = 0$$
$$P(click(3))(-2 \times 0.25 - 2 \times 0.35 + 3 \times 0.4) = 0$$

gaurantees: no preference on random-position bias.

Optimized Interleaving: Properties

Properties of Optimized Interleaving:

- **User experience**:
  - **Strongest guarantees** of all interleaving methods.
- **Correctness**:
  - Method has **fidelity** (if optimized for it),
    as long as the linear optimization is successful.
  - Proven by **brute-forcing** that there is **always a solution** for top-10 rankings.
  - Can be correct **under other definitions as well**.

Online Learning to Rank: Dueling Bandit Gradient Descent

Dueling Bandit Gradient Descent: Introduction

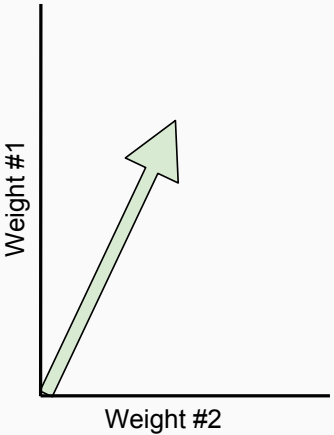
Introduced by Yue and Joachims (2009) as the **first online learning to rank** method.

Intuition:

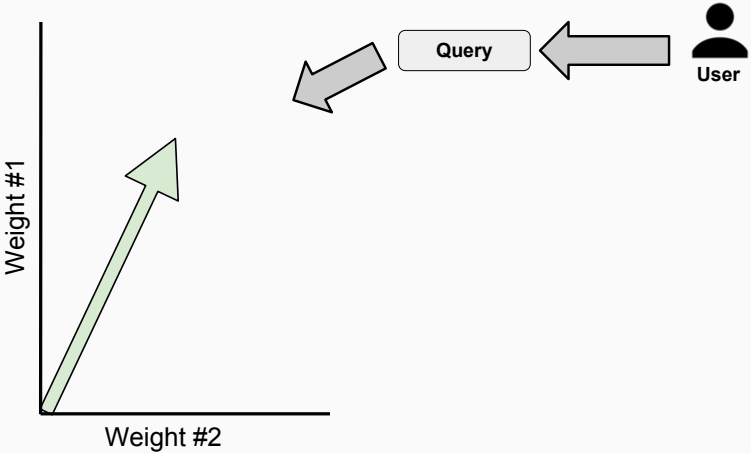
- if **online evaluation** can tell us if a **ranker is better** than another, then we can use it to **find an improvement** of our system.

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

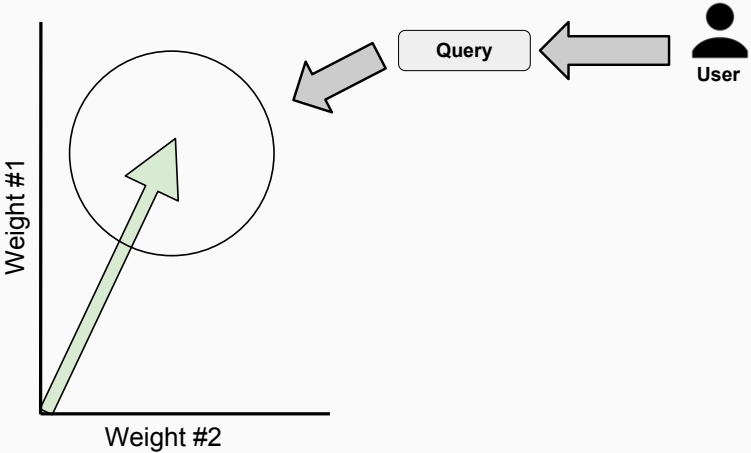
Dueling Bandit Gradient Descent: Visualization



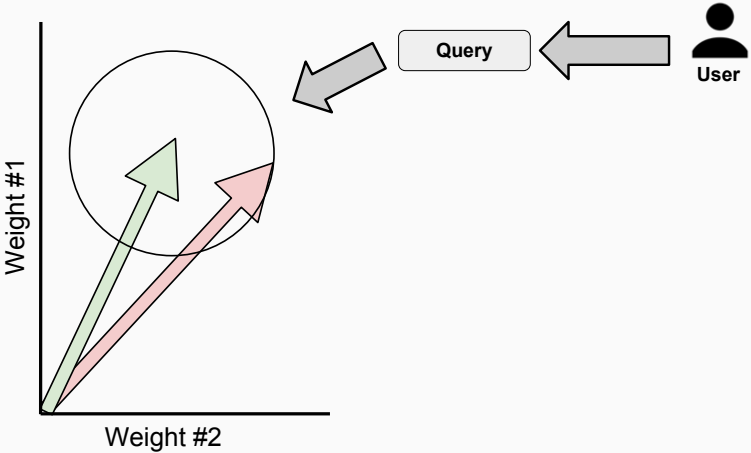
Dueling Bandit Gradient Descent: Visualization



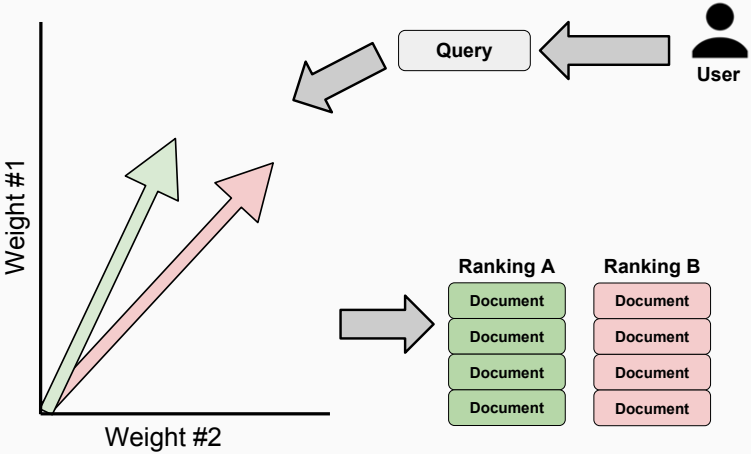
Dueling Bandit Gradient Descent: Visualization



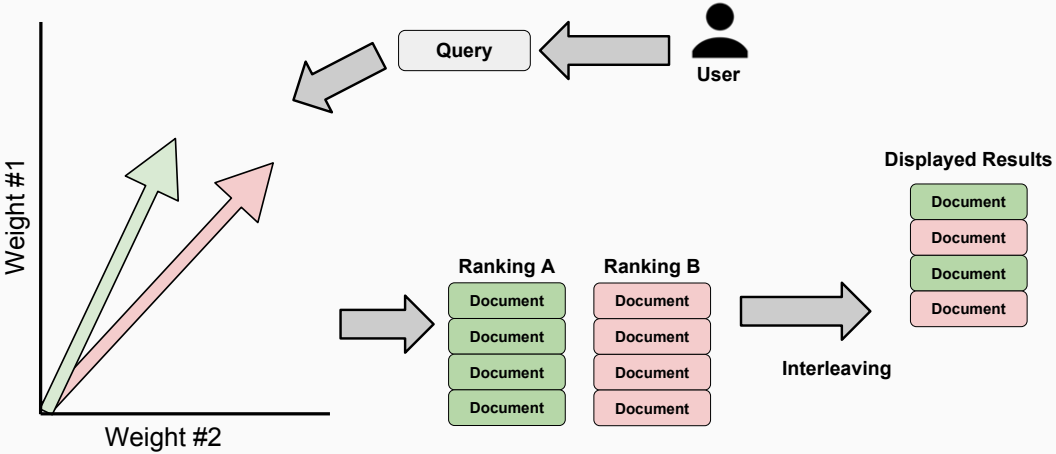
Dueling Bandit Gradient Descent: Visualization



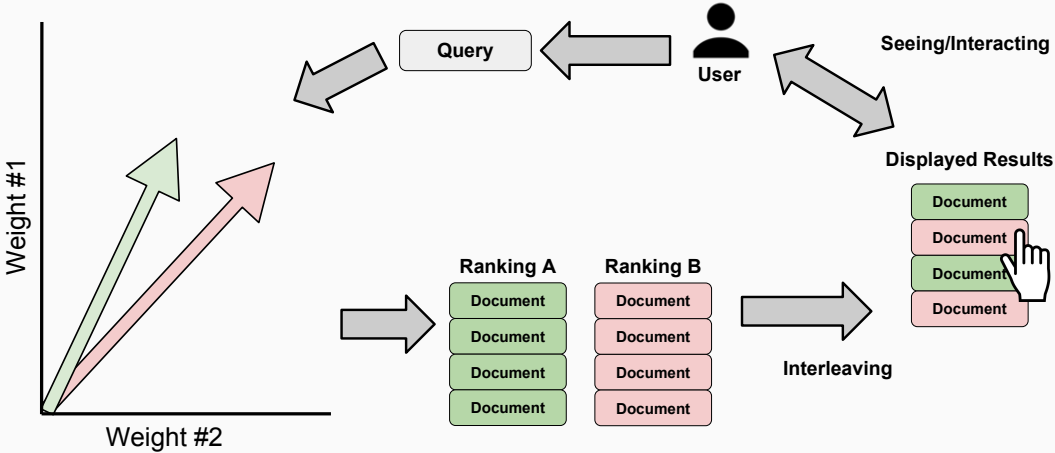
Dueling Bandit Gradient Descent: Visualization



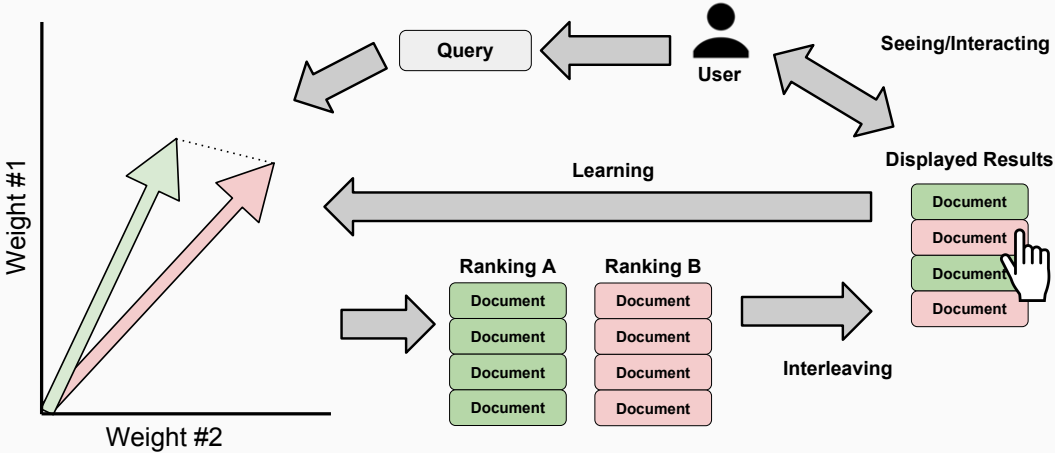
Dueling Bandit Gradient Descent: Visualization



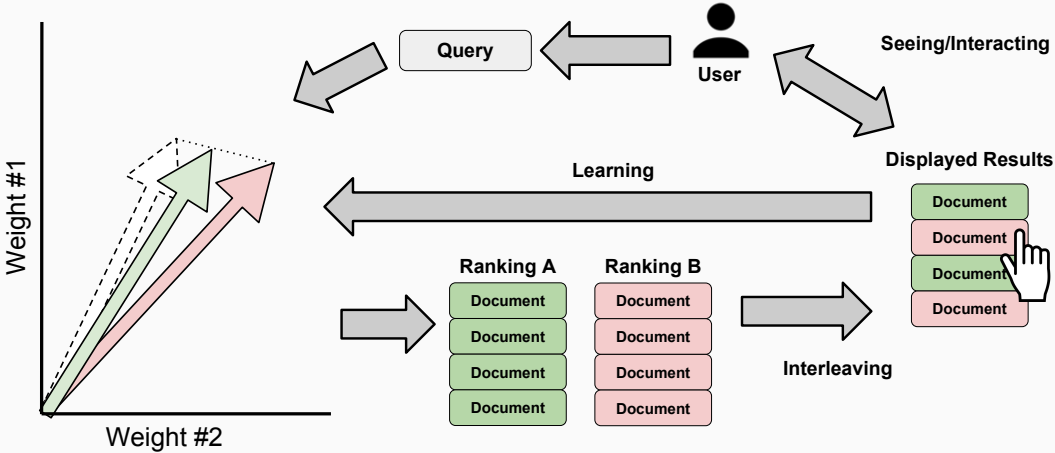
Dueling Bandit Gradient Descent: Visualization



Dueling Bandit Gradient Descent: Visualization



Dueling Bandit Gradient Descent: Visualization



Dueling Bandit Gradient Descent: Properties

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

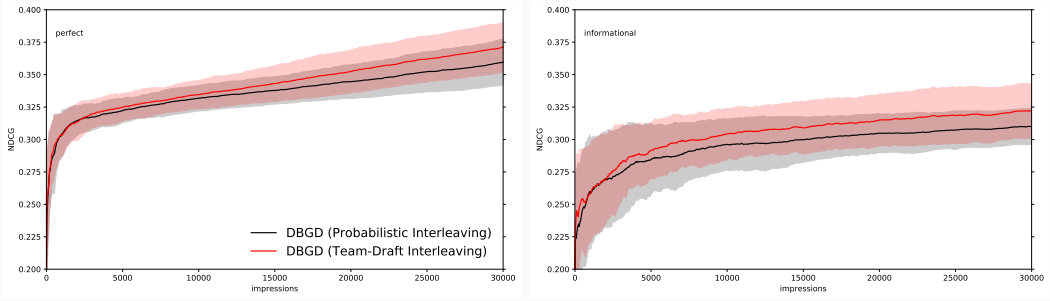
- There is a **single optimal** set of parameters: θ^*.
- The **utility space** w.r.t. θ is **convex** and **smooth**, i.e., small changes in θ lead to small changes in user experience.

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

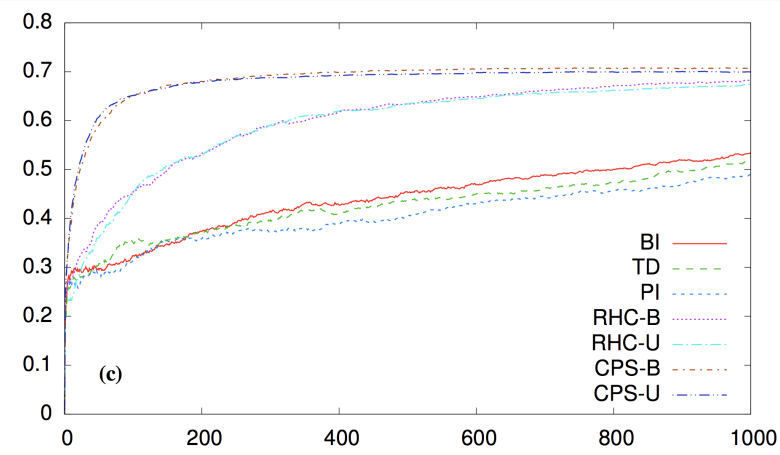
Hofmann et al. (2013) introduced the idea of **guiding exploration** by **reusing previous interactions**.

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

Hofmann et al. (2013) introduced the **Candidate Pre-Selection** method:

- Sample a **large number** of rankers to create a **candidate set**.
- Use historical interactions to select the **most promising candidate** for DBGD.

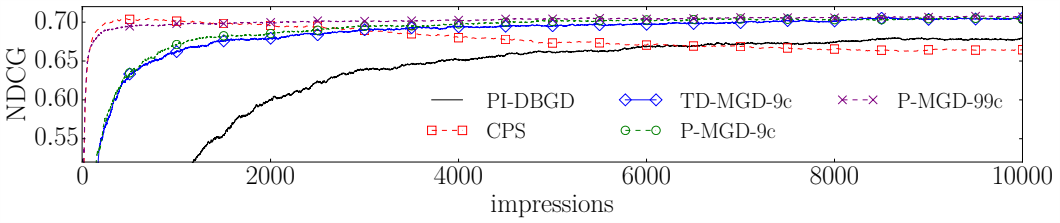
Reusing Historical Interactions: Performance



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., 2018).

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

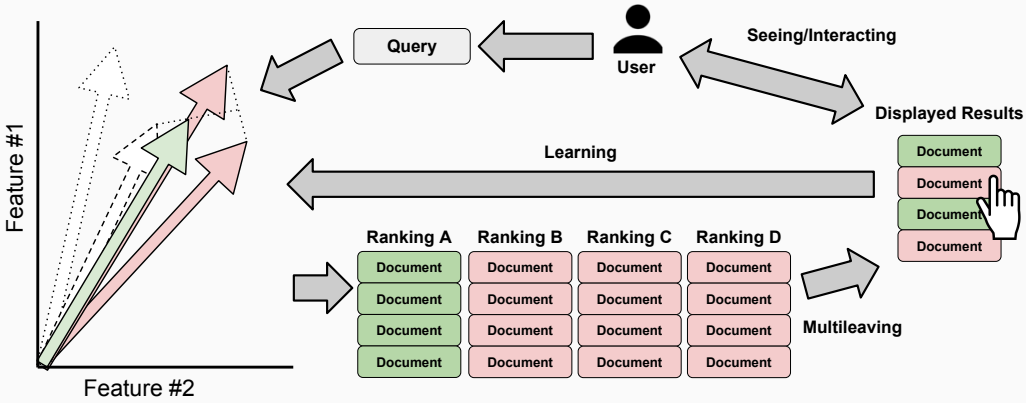
Multileave Gradient Descent

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).

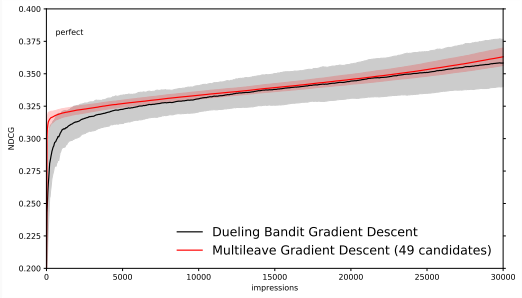
Multileaving allows comparisons with **multiple candidate rankers**, **increasing** the **chance of finding an improvement**.

Multileave Gradient Descent: Visualization



Multileave Gradient Descent: Results

Results on the MSRL10k dataset under simulated users:



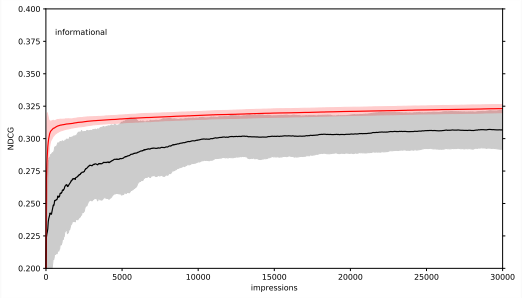


Image credits: (Oosterhuis, 2018).

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

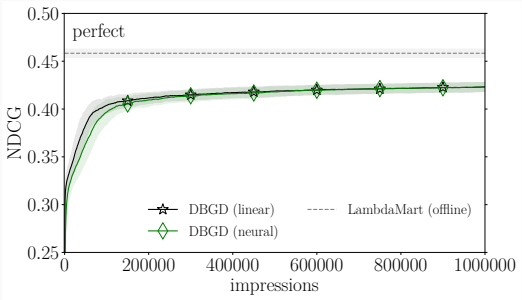
Problems with Dueling Bandit Gradient Descent

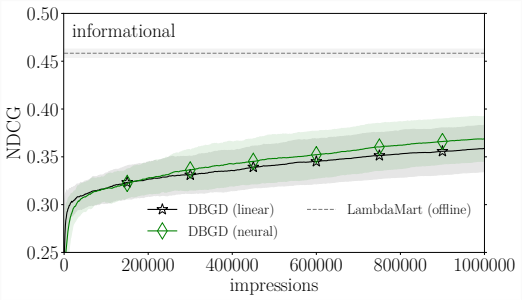
A **problem** with Dueling Bandit Gradient Descent and **all its extensions**:

- 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).

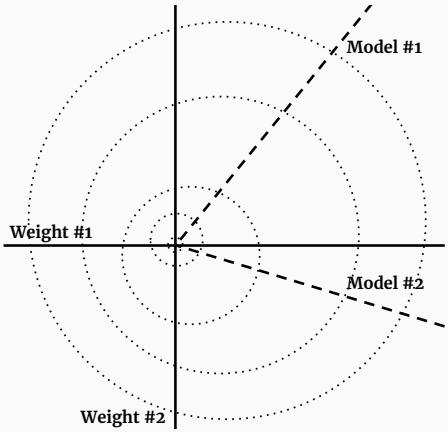
To prove this we use the fact that **the utility u is scale invariant** w.r.t. a ranking function $f_\theta(\cdot)$:

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

DBGD Assumptions: 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, 2018):

- 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**.

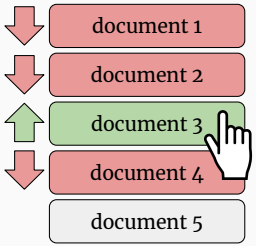
With the ranking scoring model $f_\theta(\mathbf{d})$ the distribution is:

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

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**:

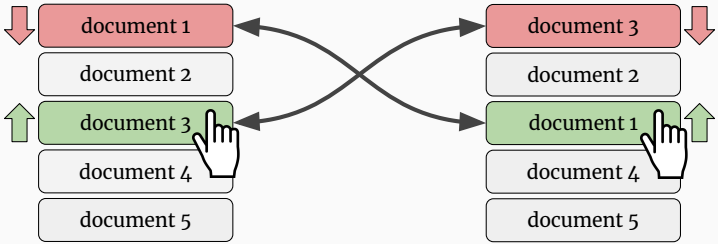


This approach is **biased**:

- 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:

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

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:

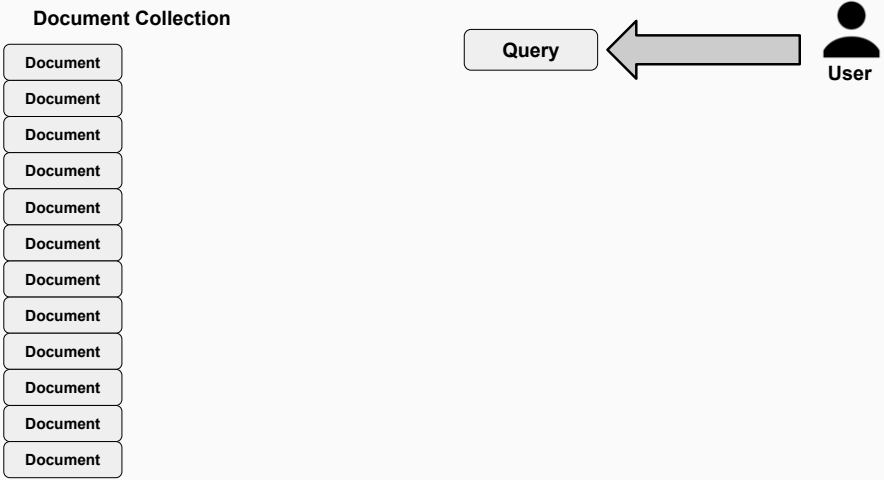
$$P(d|D,\theta_t) = \frac{\exp^{f_{\theta_t}(\mathbf{d})}}{\sum_{d' \in D} \exp^{f_{\theta_t}(\mathbf{d}')}}.$$

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**:

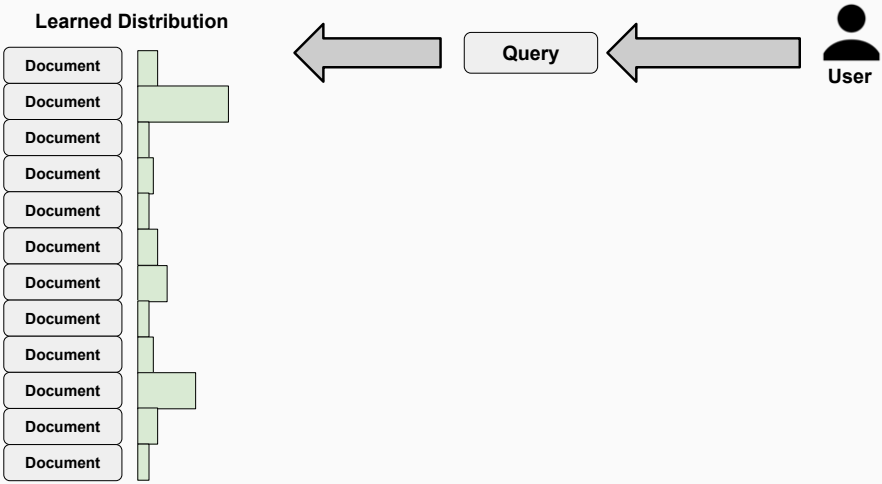
$$\nabla f_{\theta_t}(\cdot) \approx \sum_{d_i >_{\mathbf{c}} d_j} \rho(d_i, d_j, R) \nabla P(d_i \succ d_j | D, \theta_t).$$

Pairwise Differentiable Gradient Descent: Visualization

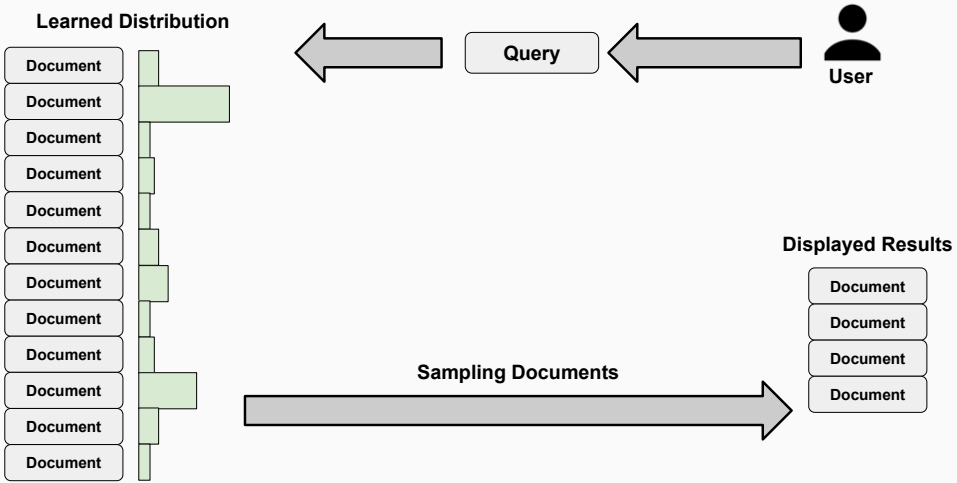
Pairwise Differentiable Gradient Descent: Visualization



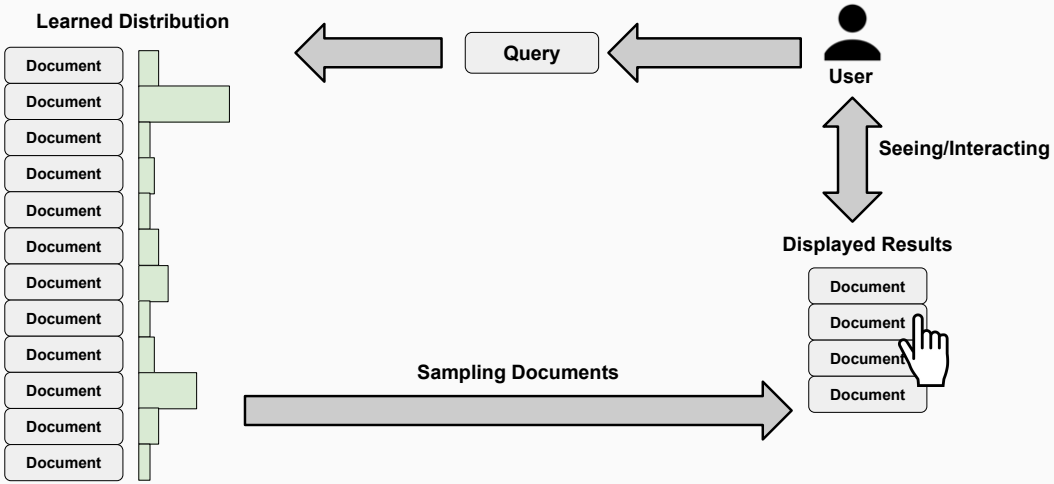
Pairwise Differentiable Gradient Descent: Visualization



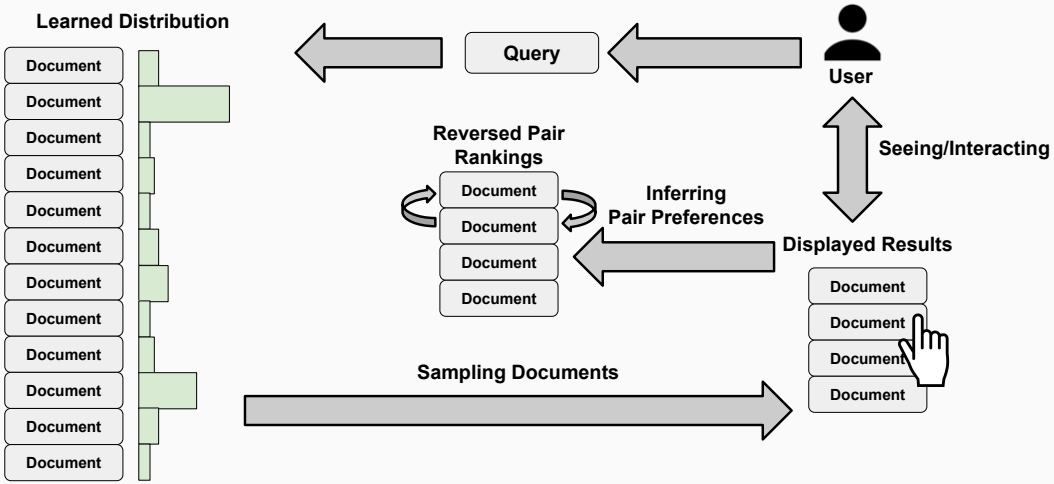
Pairwise Differentiable Gradient Descent: Visualization



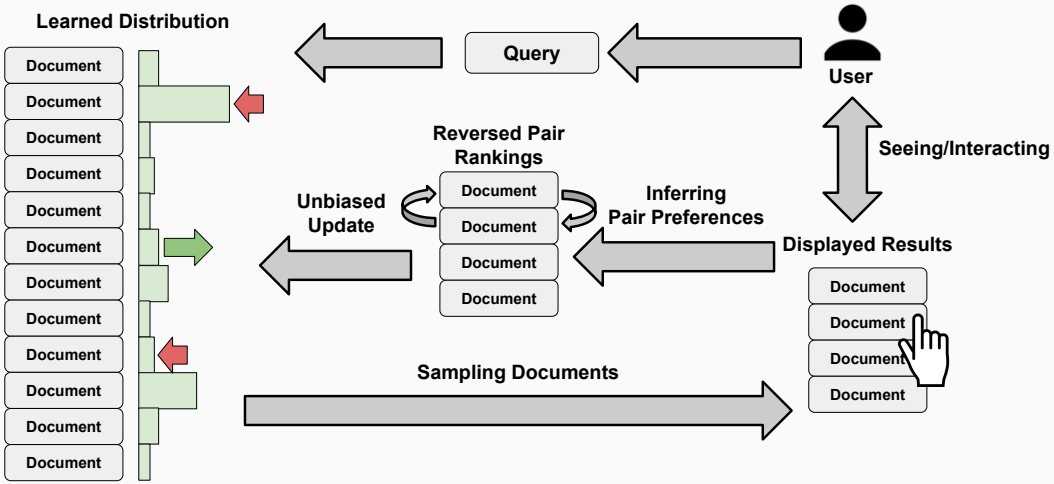
Pairwise Differentiable Gradient Descent: Visualization



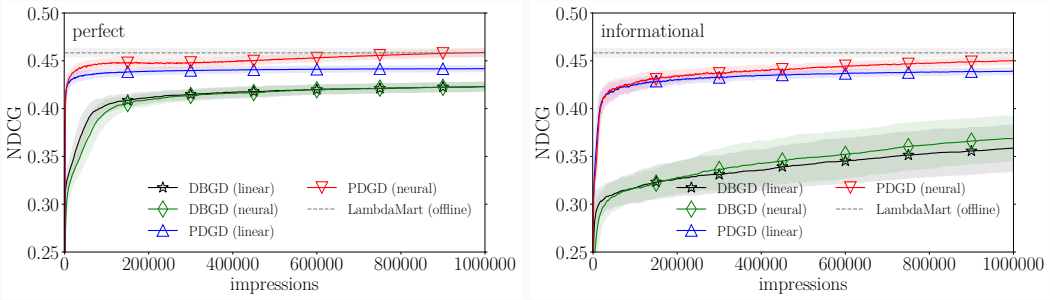
Pairwise Differentiable Gradient Descent: Visualization



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, 2018).

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**:

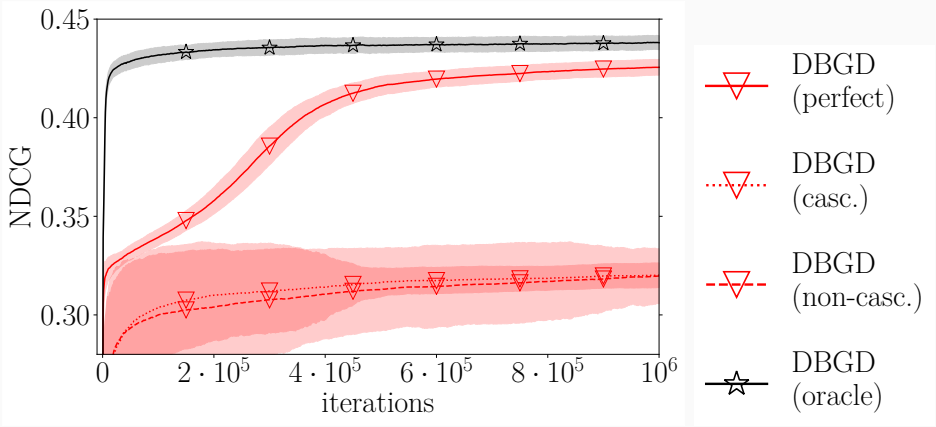
- MSLR-Web10k, Yahoo Webscope, Istella.

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.

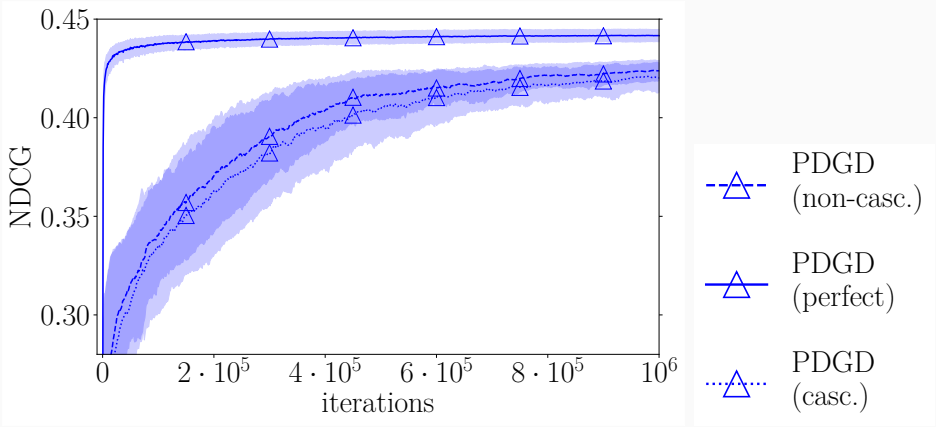
Empirical Comparison: DBGD



Results of simulations on the MSLR-WEB10k dataset.

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

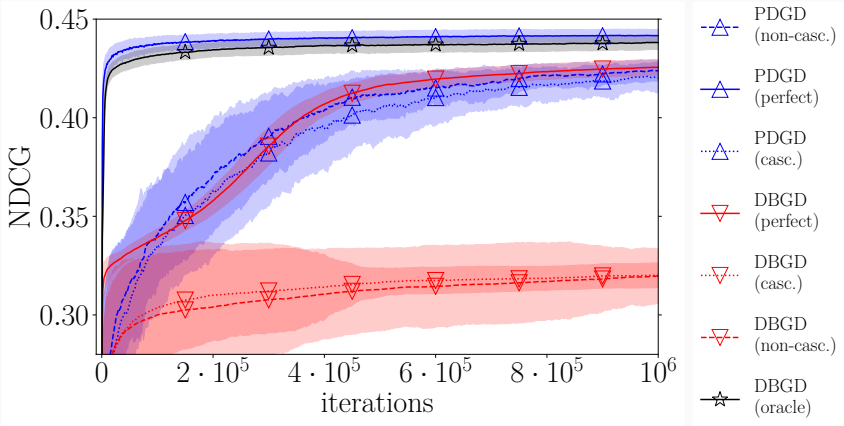
Empirical Comparison: PDGD



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.

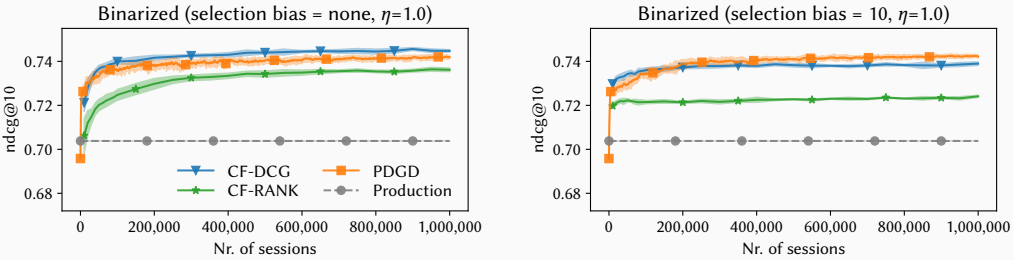
What about Counterfactual Learning to Rank?

Empirical Comparison

Single empirical comparison so far (Jagerman et al., 2019).

Using the simulated setup common in unbiased learning to rank, we apply both **Inverse Propensity Scoring** and **Pairwise Differentiable Gradient Descent.**

Empirical Comparison: Item-Selection-Bias

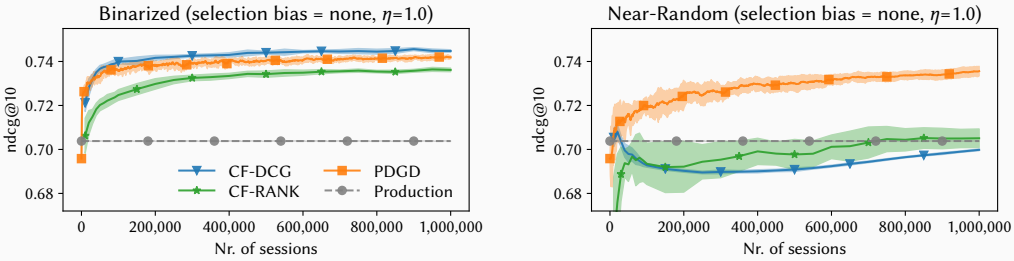


Little interaction noise, no item-selection-bias (left) and at rank ten (right).

The effect of item-selection-bias is greater on the counterfactual method than on the online method.

Image credits: (Jagerman et al., 2019).

Empirical Comparison: Interaction Noise



Little interaction noise (left) and near-random interaction noise (right).

The effect of interaction noise is **substantial** on the counterfactual method and very limited on the online method.

Image credits: (Jagerman et al., 2019).

Empirical Comparison: Conclusion

Counterfactual Learning to Rank:

- Slightly higher performance under:
  - no item-selection-bias,
  - little interaction noise.
- Very affected by high interaction noise.

Online Learning to Rank:

- More reliable performance across settings.
- Handles item-selection bias well.
- More robust to noise

Overall the empirical results suggest that **Online Learning to Rank** is more reliable.

Theoretical Comparison: Conclusion

Counterfactual Learning to Rank:

- Explicit position bias model.
- Proven to unbiasedly optimize ranking metrics.
- Can be interactive.
- Applicable to any historical interactions.

Online Learning to Rank:

- No explicit user model.
- Not proven to unbiasedly optimize ranking metrics.
- Only effective when interactive.
- Not applicable to all historical interactions.

In theory **Counterfactual Learning to Rank** has all the advantageous properties.

Conclusion

Conclusion

- **Online approaches** allow for **unbiased** and **responsive** learning to rank:
  - **Immediately adapt** to user behavior.
  - Perform **randomization** at each step, though limited.
- The Online Learning to Rank field seems to be in **trouble**:
  - **Theoretical guarantees** are **unsound** for the standard ranking problem.
  - Dueling Bandit Gradient Descent method is **unable to solve toy-problems**.
- **Comparison** with the **Counterfactual** approach:
  - **Empirically:** Online methods appear to be more reliable.
  - **Theoretically:** Counterfactual methods are much more advantageous.

Questions and Answers

Thank you for listening!

References i

K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 249–258. ACM, 2011.

K. Hofmann, A. Schuth, S. Whiteson, and M. de Rijke. Reusing historical interaction data for faster online learning to rank for ir. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 183–192. ACM, 2013.

R. Jagerman, H. Oosterhuis, and M. de Rijke. To model or to intervene: A comparison of counterfactual and online learning to rank from user interactions. In *42nd International ACM SIGIR Conference on Research & Development in Information Retrieval*, page (to appear). ACM, 2019.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

T. Joachims. Evaluating retrieval performance using clickthrough data. In J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining*. Physica Verlag, 2003.

References ii

H. Oosterhuis. Learning to rank and evaluation in the online setting. 12th Russian Summer School in Information Retrieval (RuSSIR 2018), 2018.

H. Oosterhuis and M. de Rijke. Sensitive and scalable online evaluation with theoretical guarantees. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 77–86. ACM, 2017.

H. Oosterhuis and M. de Rijke. Differentiable unbiased online learning to rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1293–1302. ACM, 2018.

H. Oosterhuis and M. de Rijke. Optimizing ranking models in an online setting. In *Advances in Information Retrieval*, pages 382–396, Cham, 2019. Springer International Publishing.

H. Oosterhuis, A. Schuth, and M. de Rijke. Probabilistic multileave gradient descent. In *European Conference on Information Retrieval*, pages 661–668. Springer, 2016.

References iii

F. Radlinski and N. Craswell. Optimized interleaving for online retrieval evaluation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 245–254. ACM, 2013.

A. Schuth, K. Hofmann, and F. Radlinski. Predicting search satisfaction metrics with interleaved comparisons. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 463–472. ACM, 2015.

A. Schuth, H. Oosterhuis, S. Whiteson, and M. de Rijke. Multileave gradient descent for fast online learning to rank. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 457–466. ACM, 2016.

H. Wang, R. Langley, S. Kim, E. McCord-Snook, and H. Wang. Efficient exploration of gradient space for online learning to rank. *arXiv preprint arXiv:1805.07317*, 2018.

Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.

T. Zhao and I. King. Constructing reliable gradient exploration for online learning to rank. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1643–1652. ACM, 2016.

Acknowledgments

All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their employers and/or sponsors.