# Optimizing Ranking Systems from User Interactions

**Harrie Oosterhuis**

February 22, 2019

University of Amsterdam

oosterhuis@uva.nl

https://staff.fnwi.uva.nl/h.r.oosterhuis

Based on *Differentiable Unbiased Online Learning to Rank* (Oosterhuis and de Rijke, 2018).

# Introduction

# Learning to Rank in Information Retrieval

**Ranking systems** are the basis for **search** and most **recommendation**.

**Learning to rank** enables the optimization of ranking systems:

- Directly improves user experience.
- Increase engagements, conversions, sales, views, etc.

# Learning to Rank in Information Retrieval

**Ranking systems** are the basis for **search** and most **recommendation**.

**Learning to rank** enables the optimization of ranking systems:

- Directly improves user experience.
- Increase engagements, conversions, sales, views, etc.

Traditionally learning to rank uses **annotated datasets**:

- **Relevance annotations** for query-document pairs provided by **human judges**.

# Problems with Supervised Approach

Some of the most substantial limitations of **annotated datasets** are:

- **expensive** to make (Qin and Liu, 2013; Chapelle and Chang, 2011).

# Problems with Supervised Approach

Some of the most substantial limitations of **annotated datasets** are:

- **expensive** to make (Qin and Liu, 2013; Chapelle and Chang, 2011).
- **unethical** to create in **privacy-sensitive settings** (Wang et al., 2016).

# Problems with Supervised Approach

Some of the most substantial limitations of **annotated datasets** are:

- **expensive** to make (Qin and Liu, 2013; Chapelle and Chang, 2011).
- **unethical** to create in **privacy-sensitive settings** (Wang et al., 2016).
- **impossible** for small scale problems e.g. **personalization**.

# Problems with Supervised Approach

Some of the most substantial limitations of **annotated datasets** are:

- **expensive** to make (Qin and Liu, 2013; Chapelle and Chang, 2011).
- **unethical** to create in **privacy-sensitive settings** (Wang et al., 2016).
- **impossible** for small scale problems e.g. **personalization**.
- **stationary**, cannot capture **future changes in relevancy** (Lefortier et al., 2014).

# Problems with Supervised Approach

Some of the most substantial limitations of **annotated datasets** are:

- **expensive** to make (Qin and Liu, 2013; Chapelle and Chang, 2011).
- **unethical** to create in **privacy-sensitive settings** (Wang et al., 2016).
- **impossible** for small scale problems e.g. **personalization**.
- **stationary**, cannot capture **future changes in relevancy** (Lefortier et al., 2014).
- **not necessarily aligned with actual user preferences** (Sanderson, 2010),

  i.e. annotators and users often disagree.

# Learning from User Interactions

# Learning from User Interactions: Advantages

**Learning from user interactions** solves the problems of annotations:

- Interactions are **virtually free** if you have users.
- User **behaviour** is indicative of their **preferences**.

# Learning from User Interactions: Advantages

**Learning from user interactions** solves the problems of annotations:

- Interactions are **virtually free** if you have users.
- User **behaviour** is indicative of their **preferences**.
- Interactions give **implicit feedback**.

# Learning from User Interactions: Difficulties

User interactions bring their **own difficulties**:

- **Noise:**
  - Users click for **unexpected reasons**.
  - Often clicks occur **not because** of relevancy.

# Learning from User Interactions: Difficulties

User interactions bring their **own difficulties**:

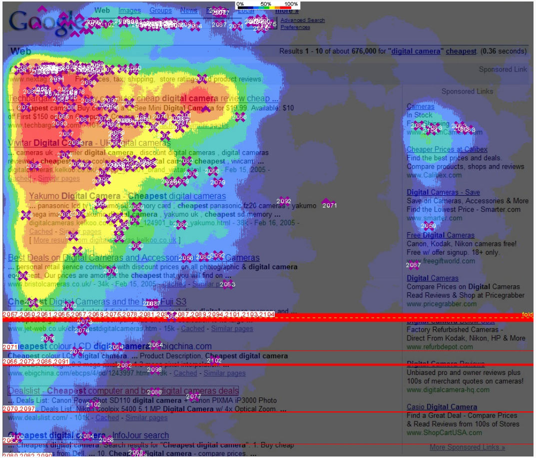
- **Noise:**
  - Users click for **unexpected reasons**.
  - Often clicks occur **not because** of relevancy.
  - Often clicks do not occur **despite** of relevancy.

# Learning from User Interactions: Difficulties

User interactions bring their **own difficulties**:

- **Noise:**
  - Users click for **unexpected reasons**.
  - Often clicks occur **not because** of relevancy.
  - Often clicks do not occur **despite** of relevancy.
- **Bias:** Interactions are affected by **factors other than relevancy**:
  - **Position bias:** **Higher ranked** documents get more attention.
  - **Selection bias:** Interactions are **limited** to the **presented** documents.
  - **Presentation bias:** Results that are **presented different** will be **treated different**.
  - ...

# The Golden Triangle




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

# Learning from User Interactions: Goal

**Goal of unbiased learning to rank from user interactions:**

- Learn the **relevance preferences** of the user from their interactions.
- **Avoid** being **biased by other factors** that influence interactions.

# Learning from User Interactions: Conclusion

**Learning from Historical Interactions:**

- Learn/estimate a **model of user behaviour** including their biases.
- Learn from historical data while **adjusting** for these **biases**.
- See: (Wang et al., 2018b; Joachims et al., 2017; Ai et al., 2018)

**Online Learning to Rank:**

- Algorithms that can **intervene** during the learning process.
- Handle biases by having **control over displayed results**.

# Dueling Bandit Gradient Descent

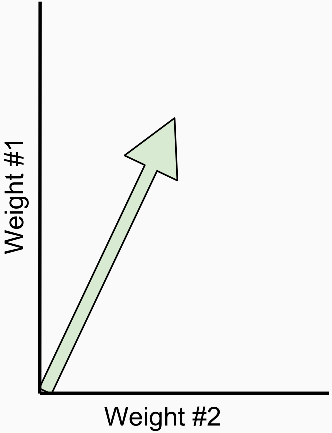
# Dueling Bandit Gradient Descent: Introduction

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

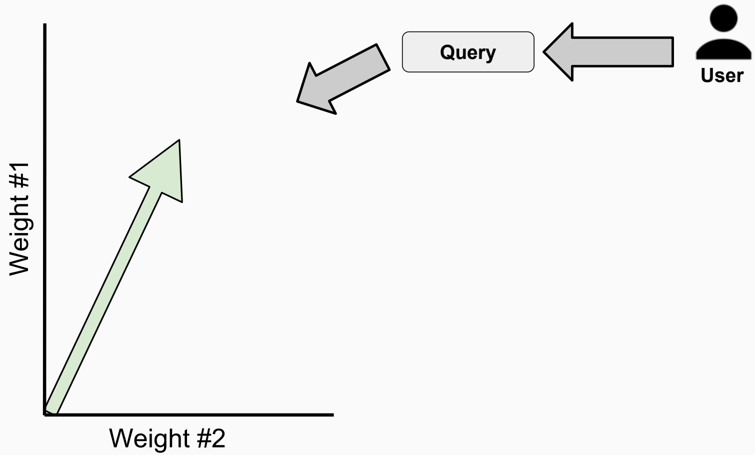
**Intuition:**

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

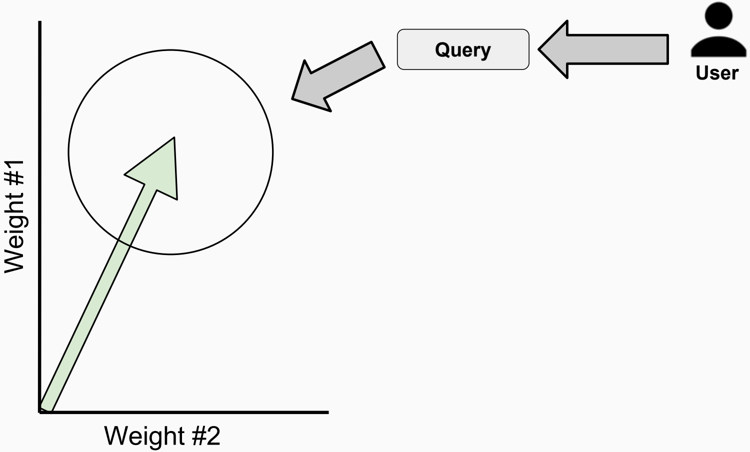
# Dueling Bandit Gradient Descent: Visualization



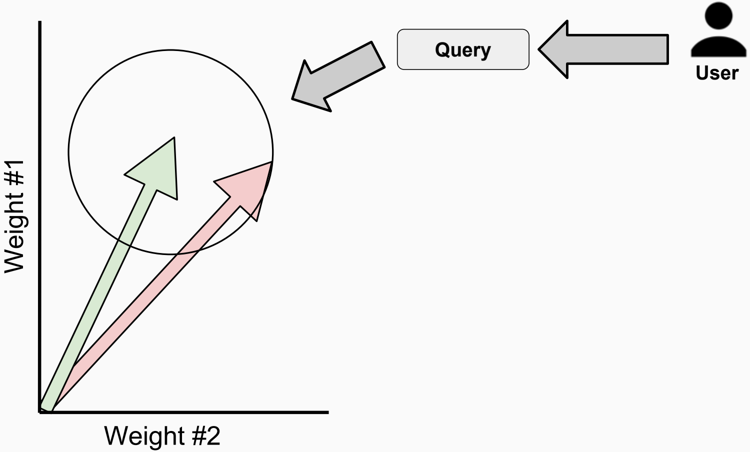

# Dueling Bandit Gradient Descent: Visualization



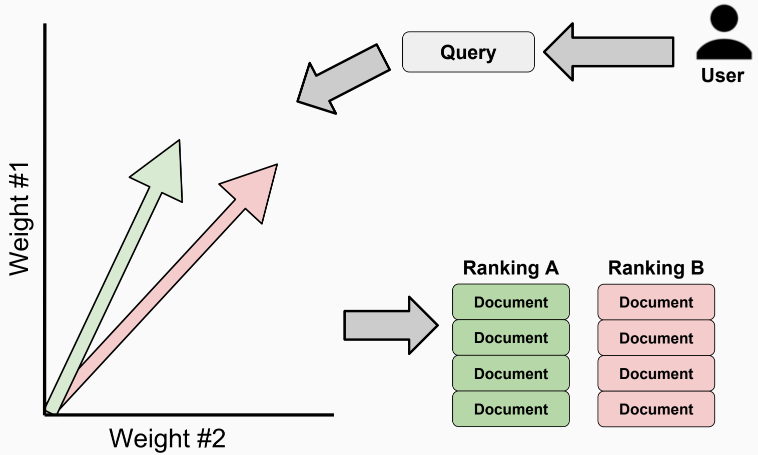
# Dueling Bandit Gradient Descent: Visualization



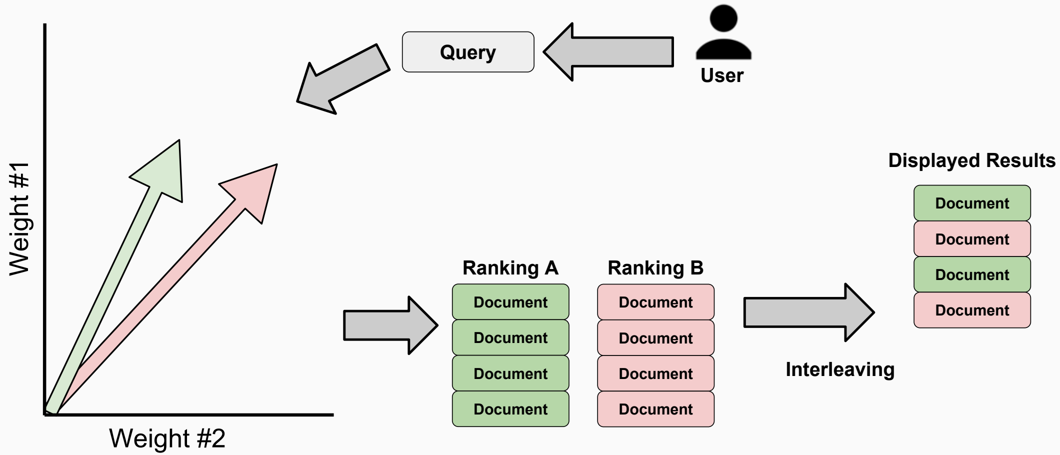
# Dueling Bandit Gradient Descent: Visualization



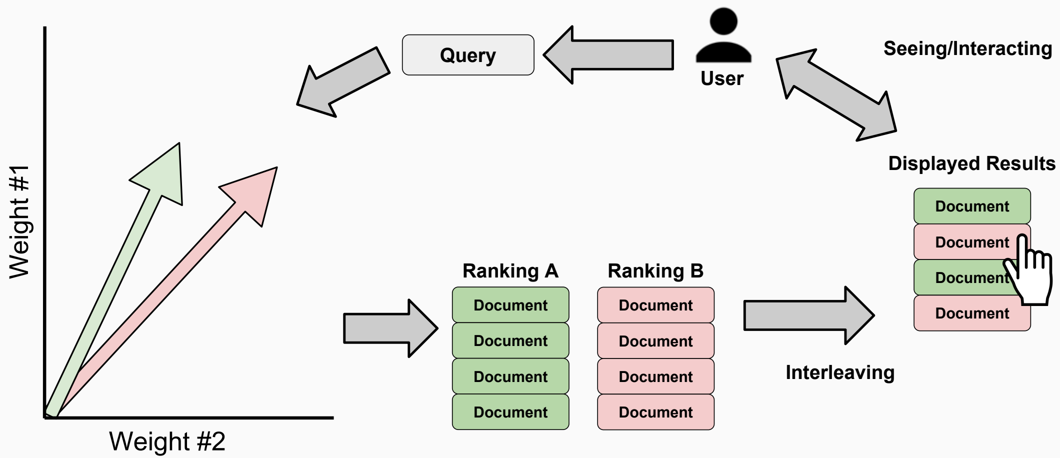
## Dueling Bandit Gradient Descent: Visualization



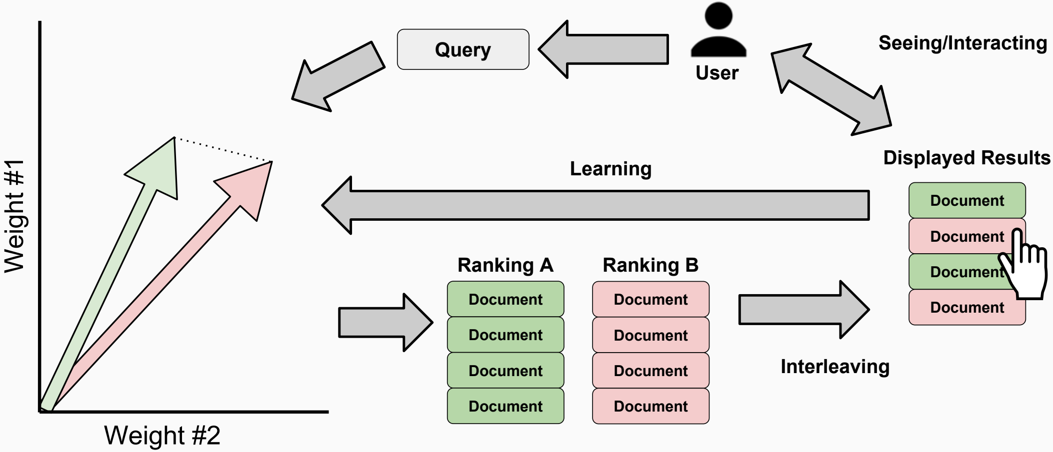
# Dueling Bandit Gradient Descent: Visualization



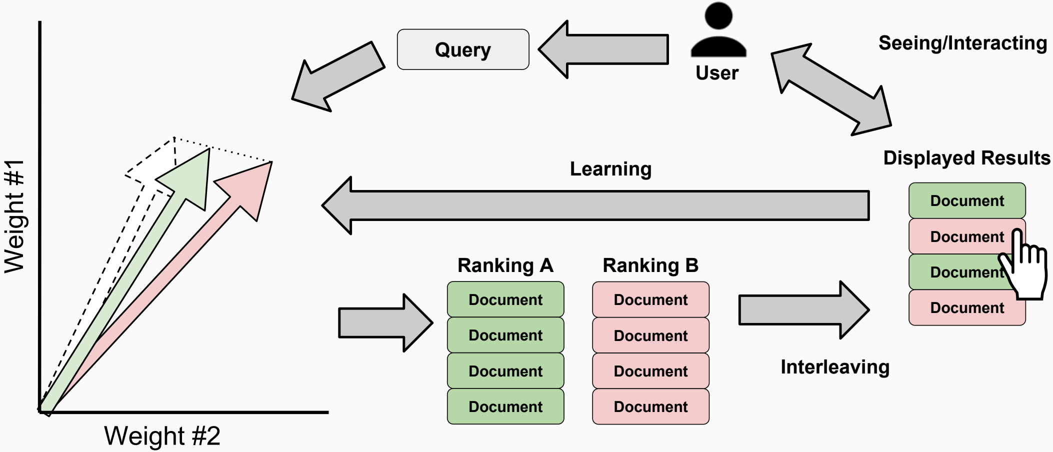
# Dueling Bandit Gradient Descent: Visualization



# Dueling Bandit Gradient Descent: Visualization



# Dueling Bandit Gradient Descent: Visualization



# Dueling Bandit Gradient Descent: Properties

**Basis** of the online learning to rank **field**,
virtually all existing methods are **extensions** of this algorithm
(Schuth et al., 2016; Hofmann et al., 2013; Zhao and King, 2016; Wang et al., 2018a).

**Problems** with Dueling Bandit Gradient Descent:

- A **considerable gap** between *offline* learning to rank performance, even for subsequent extensions of method.
- Ineffective at optimizing **non-linear models**.

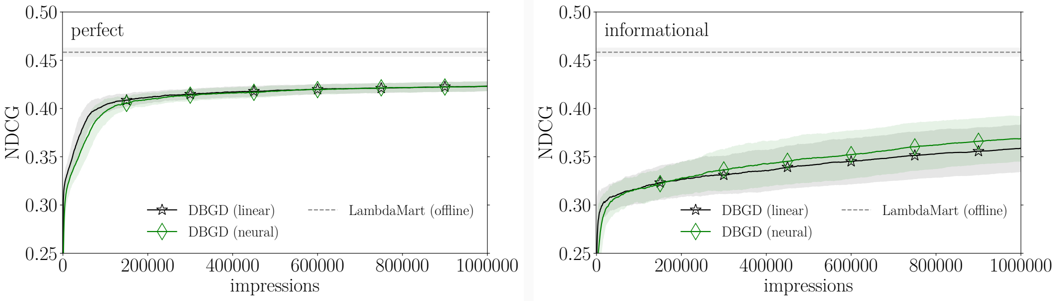
# Dueling Bandit Gradient Descent: Properties

**Basis** of the online learning to rank **field**,
virtually all existing methods are **extensions** of this algorithm
(Schuth et al., 2016; Hofmann et al., 2013; Zhao and King, 2016; Wang et al., 2018a).

**Problems** with Dueling Bandit Gradient Descent:

- A **considerable gap** between *offline* learning to rank performance, even for subsequent extensions of method.
- Ineffective at optimizing **non-linear models**.
- **No proven regret bounds for ranking problems** (Oosterhuis and de Rijke, 2019).

# Dueling Bandit Gradient Descent: Results




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

# 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.

**Intuition:**

- A **pairwise** approach can be made **unbiased**, while being **differentiable**, without relying on online evaluation methods 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 $f(\mathbf{d})$ the distribution is:

$$P(d|f, D) = \frac{\exp^{f(\mathbf{d})}}{\sum_{d' \in D} \exp^{f(\mathbf{d'})}} \tag{1}$$

# Plackett Luce Model

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

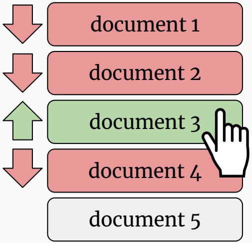
With the ranking scoring model $f(\mathbf{d})$ the distribution is:

$$P(d|f, D) = \frac{\exp^{f(\mathbf{d})}}{\sum_{d' \in D} \exp^{f(\mathbf{d'})}} \tag{1}$$

Unlike DBGD, **confidence is explicitly modelled** and **exploration naturally varies** per query and even within the ranking.

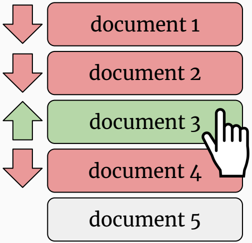
# Bias in Pairwise Inference

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



# 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**:

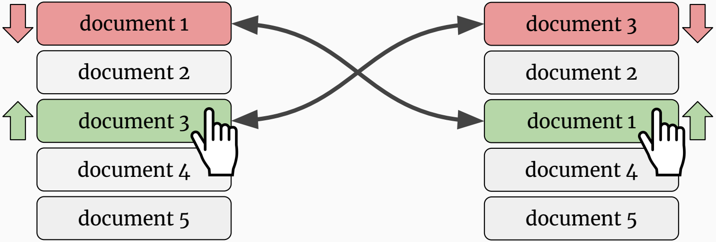



This approach is **biased**:

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

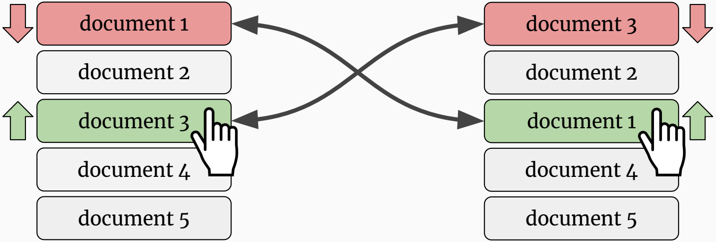
# Reversed Pair Rankings

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



# Reversed Pair Rankings

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

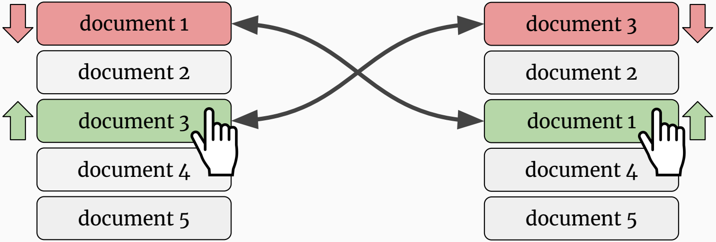



We assume:

- 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 $R^*(d_i, d_j, R)$.

# 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**:

$$\rho(d_i, d_j, R) = \frac{P(R^*(d_i, d_j, R) | f, D)}{P(R | f, D) + P(R^*(d_i, d_j, R) | f, D)}. \tag{2}$$

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

$$\nabla f \approx \sum_{d_i >_{\mathbf{c}} d_j} \rho(d_i, d_j, R) \nabla P(d_i \succ d_j | f, D). \tag{3}$$

# Unbiasedness of Pairwise Differentiable Gradient Descent

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

$$E[\nabla f] = \sum_{d_i, d_j} \alpha_{ij}(f'(\mathbf{d_i}) - f'(\mathbf{d_j})). \tag{4}$$

# Unbiasedness of Pairwise Differentiable Gradient Descent

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

$$E[\nabla f] = \sum_{d_i, d_j} \alpha_{ij}(f'(\mathbf{d_i}) - f'(\mathbf{d_j})). \tag{4}$$

Where the weights $\alpha_{ij}$ will **match the user preferences** in expectation:

$$d_i =_{rel} d_j \Leftrightarrow \alpha_{ij} = 0, \tag{5}$$

$$d_i >_{rel} d_j \Leftrightarrow \alpha_{ij} > 0, \tag{6}$$

$$d_i <_{rel} d_j \Leftrightarrow \alpha_{ij} < 0. \tag{7}$$

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

## Pairwise Differentiable Gradient Descent: Method

Start with initial model $f_t$.

Then indefinitely:

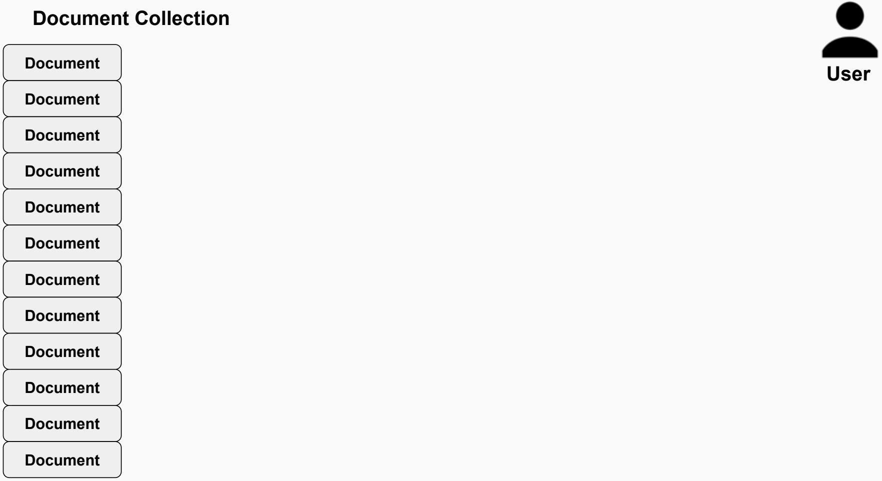
1. Wait for a user query.
2. **Sample** (without replacement) a **ranking** $R$ from the document distribution:

$$P(d|D, f_t) = \frac{\exp^{f_t(\mathbf{d})}}{\sum_{d' \in D} \exp^{f_t(\mathbf{d}')}}. \tag{8}$$

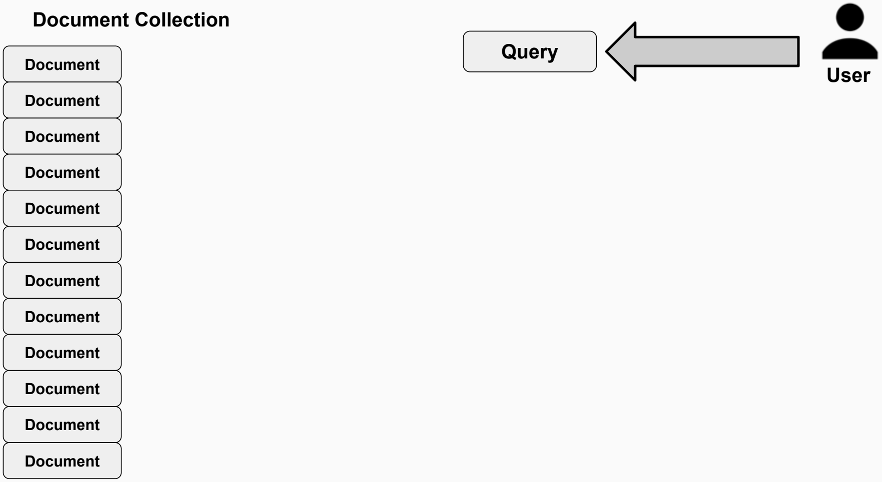
3. **Display** the ranking $R$ to the user.
4. **Infer document preferences** from the **user clicks**: $\mathbf{c}$.
5. **Update** model according to the **estimated (unbiased) gradient**:

$$\nabla f_t \approx \sum_{d_i >_{\mathbf{c}} d_j} \rho(d_i, d_j, R) \nabla P(d_i \succ d_j | D, f_t). \tag{9}$$

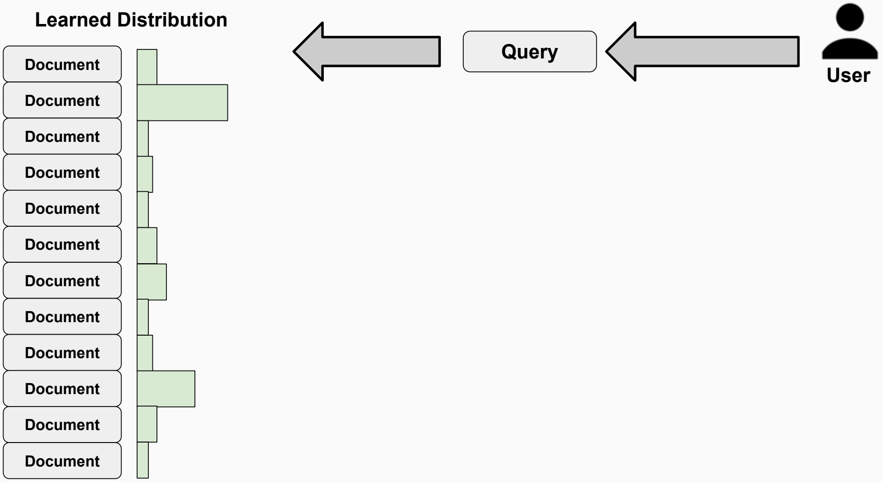
# Pairwise Differentiable Gradient Descent: Visualization



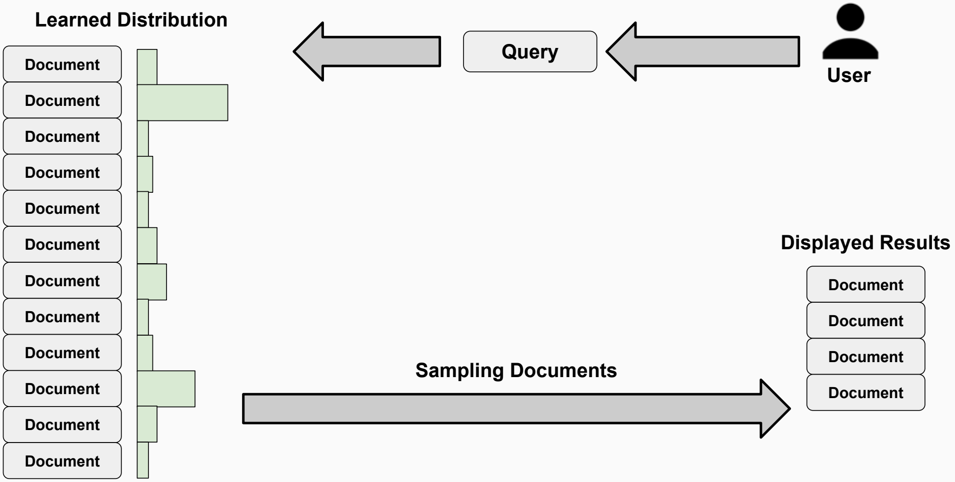
# Pairwise Differentiable Gradient Descent: Visualization



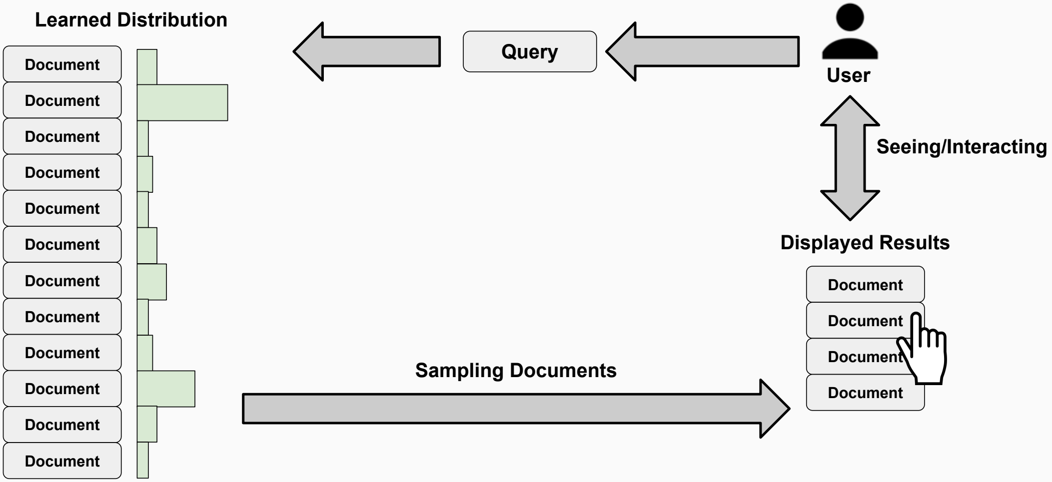
# Pairwise Differentiable Gradient Descent: Visualization



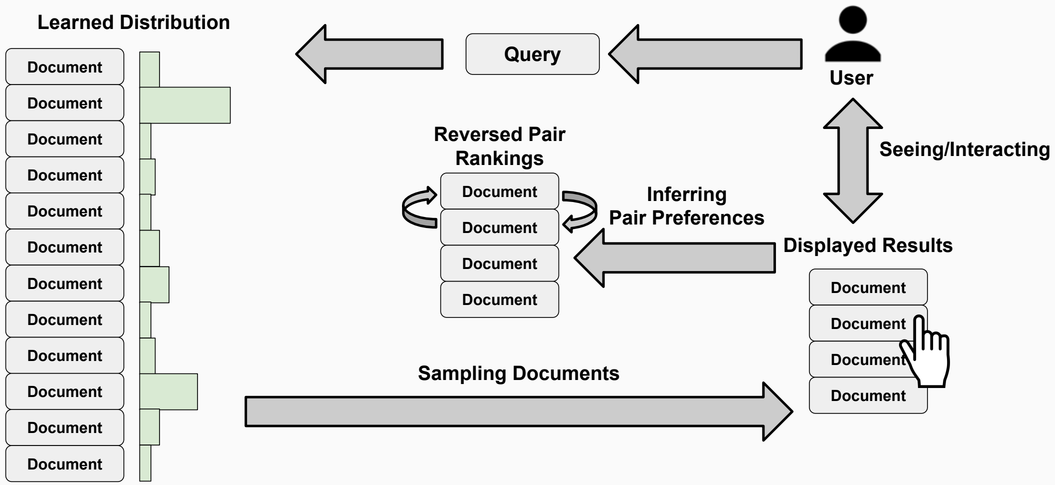
# Pairwise Differentiable Gradient Descent: Visualization



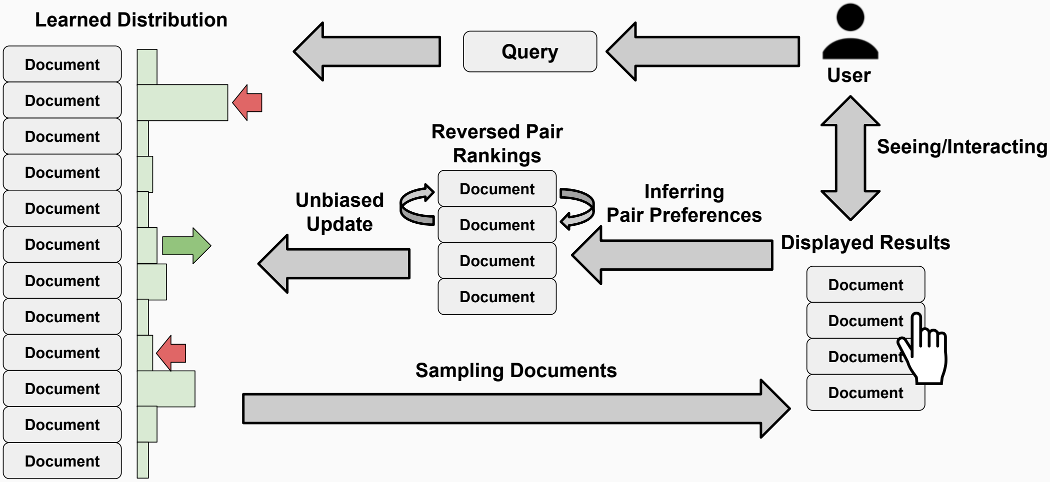
# Pairwise Differentiable Gradient Descent: Visualization



# Pairwise Differentiable Gradient Descent: Visualization



# Pairwise Differentiable Gradient Descent: Visualization



# Experimental Results

# Experimental Setup

Comparison of Pairwise Differentiable Gradient Descent with **previous Online Learning to Rank methods**.

Simulations based on the annotated learning-to-rank datasets.

- **Largest available industry datasets**: MSLR-Web10k, Yahoo Webscope, Istella.

User behaviour simulated using **cascading click models**.

Experiments **repeated** under **varying levels** of noise and bias.

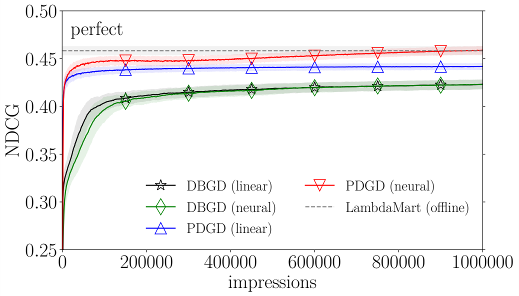
# Pairwise Differentiable Gradient Descent: Results

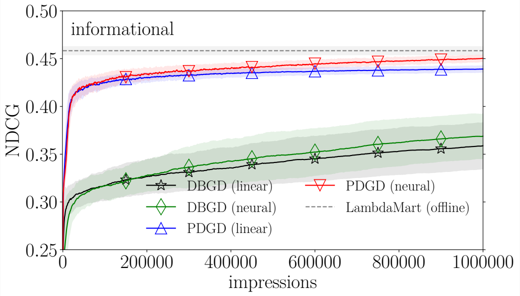
Results **across all datasets** (MSLR-Web10k, Yahoo Webscope, Istella) we observe:

- **Large improvements** in performance of **convergence** under all levels of noise.
- Much **faster learning (better user experience)** under all levels of noise.

Findings further generalized in follow-up work (Oosterhuis and de Rijke, 2019).

# Pairwise Differentiable Gradient Descent: Results Long Term




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

# Conclusion

## Conclusion

With the introduction of **Pairwise Differentiable Gradient Descent** we have:

## Conclusion

With the introduction of **Pairwise Differentiable Gradient Descent** we have:

- the **first online method** to convergence **near offline levels of performance**.

# Conclusion

With the introduction of **Pairwise Differentiable Gradient Descent** we have:

- the **first online method** to convergence **near offline levels of performance**.
- **considerably faster learning** (user experience) than before.

# Conclusion

With the introduction of **Pairwise Differentiable Gradient Descent** we have:

- the **first online method** to convergence **near offline levels of performance**.
- **considerably faster learning** (user experience) than before.
- **performance** of an **non-linear model** to **exceed** linear model.

# Conclusion

With the introduction of **Pairwise Differentiable Gradient Descent** we have:

- the **first online method** to convergence **near offline levels of performance**.
- **considerably faster learning** (user experience) than before.
- **performance** of an **non-linear model** to **exceed** linear model.
- **computational efficiency much greater** than most previous methods.

# Conclusion

With the introduction of **Pairwise Differentiable Gradient Descent** we have:

- the **first online method** to convergence **near offline levels of performance**.
- **considerably faster learning** (user experience) than before.
- **performance** of an **non-linear model** to **exceed** linear model.
- **computational efficiency much greater** than most previous methods.
- currently **no proven regret bounds**.

# Future Directions for Online Learning to Rank

Now that performance is on the level of offline learning to rank:

- **The time is ripe for real-world experiments.**

# Future Directions for Online Learning to Rank

Now that performance is on the level of offline learning to rank:

- **The time is ripe for real-world experiments.**
- We can compare online learning to rank **to offline/historical approaches**.

# Future Directions for Online Learning to Rank

Now that performance is on the level of offline learning to rank:

- **The time is ripe for real-world experiments.**
- We can compare online learning to rank **to offline/historical approaches**.
- See its effectiveness for **personalization**.

## Future Directions for Online Learning to Rank

Now that performance is on the level of offline learning to rank:

- **The time is ripe for real-world experiments.**
- We can compare online learning to rank **to offline/historical approaches**.
- See its effectiveness for **personalization**.
- **Many different models, settings, and extensions possible.**

# Future Directions for Online Learning to Rank

Now that performance is on the level of offline learning to rank:

- **The time is ripe for real-world experiments.**
- We can compare online learning to rank **to offline/historical approaches**.
- See its effectiveness for **personalization**.
- **Many different models, settings, and extensions possible.**

**Please continue on our work:**

`https://github.com/HarrieO/OnlineLearningToRank`.

# References i


Q. Ai, K. Bi, C. Luo, J. Guo, and W. B. Croft. Unbiased learning to rank with unbiased propensity estimation. *arXiv preprint arXiv:1804.05938*, 2018.

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

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, A. Swaminathan, and T. Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 781–789. ACM, 2017.

# References ii


D. Lefortier, P. Serdyukov, and M. de Rijke. Online exploration for detecting shifts in fresh intent. In *CIKM 2014: 23rd ACM Conference on Information and Knowledge Management*. ACM, November 2014.

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 the online setting. In *ECIR*. ACM, 2019.

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

M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.

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*, 2018a.

X. Wang, M. Bendersky, D. Metzler, and M. Najork. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 115–124. ACM, 2016.

X. Wang, N. Golbandi, M. Bendersky, D. Metzler, and M. Najork. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 610–618. ACM, 2018b.

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.