Unbiased Learning to Rank from User Interactions

Harrie Oosterhuis

December 5, 2018

University of Amsterdam

oosterhuis@uva.nl

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

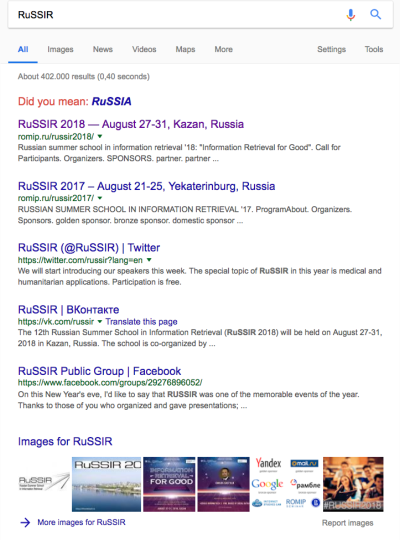
Introduction

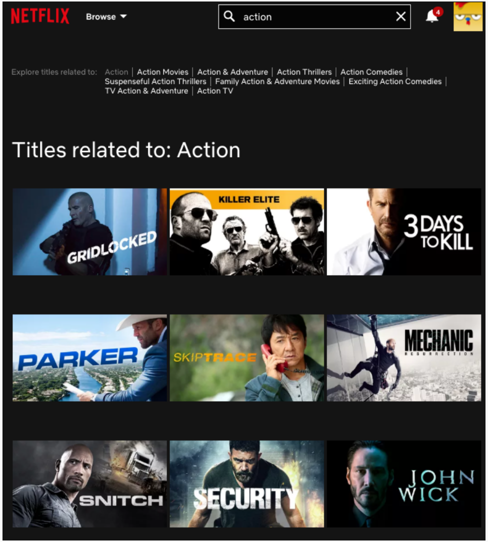
Learning to Rank in Information Retrieval

Learning to Rank is vital to informational retrieval:

- Key component for **search** and **recommendation**.
- Directly impacts user experience.

Ranking in Information Retrieval





Learning to Rank in Information Retrieval

Learning to Rank is vital to informational retrieval:

- Key component for **search** and **recommendation**.
- Directly impacts user experience.

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:

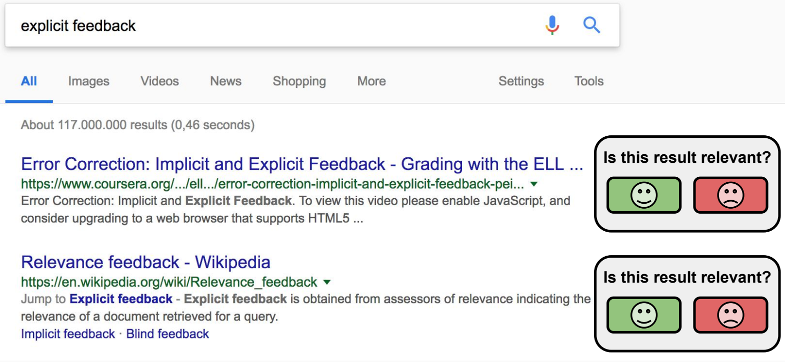
- Interactions are **virtually free** if you have users.
- User **behaviour** is indicative of their **preferences**.

Learning from User Interactions: Advantages

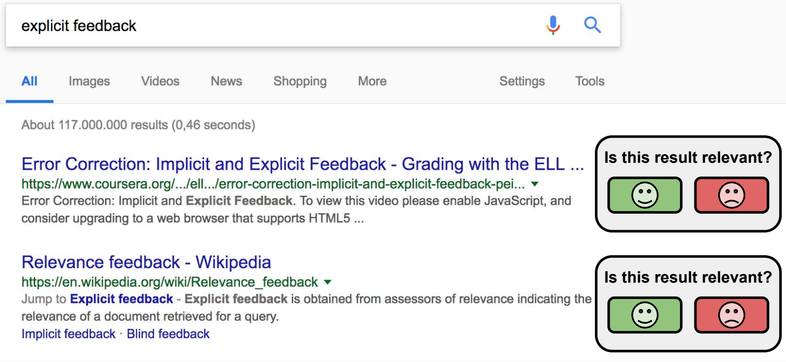
Learning from user interactions solves the problems of annotations:

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

Explicit Feedback for Search



Explicit Feedback for Search



This approach is **rarely used** in search:

- People **hate giving feedback** like this.
- It is also very **vulnerable to abuse**.

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**:

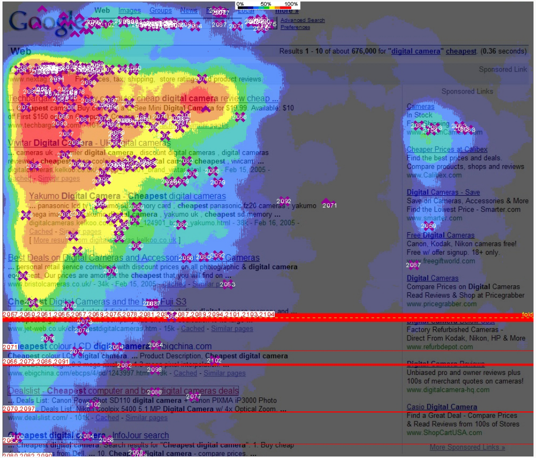
- **Noise:**
  - Users click for **unexpected reasons**.
  - Often clicks occur **not because** of relevancy.
  - Often clicks do not occur **despite** of relevancy.

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**.

Online Learning to Rank:

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

Learning from Historical Data

Learning from Historical Data: Introduction

Unbiased learning from historical user-interaction logs:

- Given that a search system has active users,
  interaction logs are **easy to obtain** and with **no significant cost**.

Learning from Historical Data: Introduction

Unbiased learning from historical user-interaction logs:

- Given that a search system has active users,
  interaction logs are **easy to obtain** and with **no significant cost**.
- From logged interactions a **model of user behaviour** can be learned.

Learning from Historical Data: Introduction

Unbiased learning from historical user-interaction logs:

- Given that a search system has active users, interaction logs are **easy to obtain** and with **no significant cost**.
- From logged interactions a **model of user behaviour** can be learned.
- If the **biases** in user interactions can be **modelled accurately**, they can also be **adjusted for effectively**.

Learning from Historical Data: Introduction

Unbiased learning from historical user-interaction logs:

- Given that a search system has active users,
  interaction logs are **easy to obtain** and with **no significant cost**.
- From logged interactions a **model of user behaviour** can be learned.
- If the **biases** in user interactions can be **modelled accurately**,
  they can also be **adjusted for effectively**.

History of this approach:

- **User modelling**: click models that predict user behaviour (Chuklin et al., 2015).
- **Unbiased learning**: applying them to learning to rank (Wang et al., 2016).
- **Counter-factual learning**: recasting the approach with counter-factual learning theory (Joachims et al., 2017).

Lambda-IPS: User Modelling

Influential method by Joachims et al. (2017), assumes **user clicks** can be **modelled** by the probability:

$$
\begin{aligned}
&P(\textit{clicked}(d) \mid \textit{relevance}(d), \textit{position}(d)) \\
&\qquad = P(\textit{clicked}(d) \mid \textit{relevance}(d), \textit{observed}(d)) \times P(\textit{observed}(d) \mid \textit{position}(d)).
\end{aligned}
$$

Lambda-IPS: User Modelling

Influential method by Joachims et al. (2017), assumes **user clicks** can be **modelled** by the probability:

$$\begin{aligned} &P(\mathit{clicked}(d) \mid \mathit{relevance}(d), \mathit{position}(d)) \\ &\quad = P(\mathit{clicked}(d) \mid \mathit{relevance}(d), \mathit{observed}(d)) \times P(\mathit{observed}(d) \mid \mathit{position}(d)). \end{aligned}$$

Ranking should be based on the **unbiased part**:

$$P(\mathit{clicked}(d) \mid \mathit{relevance}(d), \mathit{observed}(d)).$$

Lambda-IPS: User Modelling

Influential method by Joachims et al. (2017), assumes **user clicks** can be **modelled** by the probability:

$$\begin{aligned} &P(\textit{clicked}(d) | \textit{relevance}(d), \textit{position}(d)) \\ &\qquad = P(\textit{clicked}(d) | \textit{relevance}(d), \textit{observed}(d)) \times P(\textit{observed}(d) | \textit{position}(d)). \end{aligned}$$

Ranking should be based on the **unbiased part**:

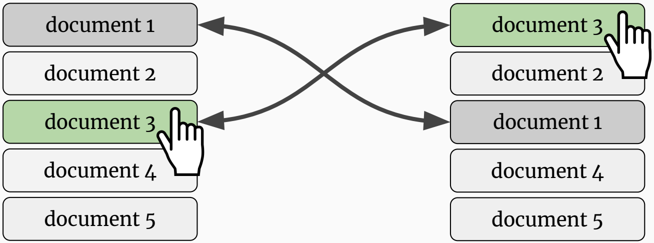
$$P(\textit{clicked}(d) | \textit{relevance}(d), \textit{observed}(d)).$$

This can be **estimated from clicks** if we know the **effect of position bias**:

$$P(\textit{observed}(d) | \textit{position}(d)).$$

Lambda-IPS: Estimating Observance Probability

We can estimate the **ratio** between **observance probabilities** by **swapping document pairs**:



The difference in click probabilities is **only affected** by the **observance probability**:

$$
\begin{aligned}
&P(\textit{clicked}(d) \mid \textit{relevance}(d), \textit{position}(d)) \\
&\quad = P(\textit{clicked}(d) \mid \textit{relevance}(d), \textit{observed}(d)) \times P(\textit{observed}(d) \mid \textit{position}(d)).
\end{aligned}
$$

Lambda-IPS: Conclusion

Joachims et al. (2017) **model observance probabilities** using the **formula**:

$$P(\mathit{observed}(d)|\mathit{position}(d)) = \left(\frac{1}{\mathit{position}(d)}\right)^{\eta}, \tag{1}$$

where η is a learned parameter.

Lambda-IPS: Conclusion

Joachims et al. (2017) **model observance probabilities** using the **formula**:

$$P(\mathit{observed}(d)|\mathit{position}(d)) = \left(\frac{1}{\mathit{position}(d)}\right)^{\eta}, \tag{1}$$

where η is a learned parameter.

Then roughly speaking:

- $P(\mathit{observed}(d)|\mathit{position}(d))$ can be inferred from clicks on **swapped pairs**.

Lambda-IPS: Conclusion

Joachims et al. (2017) **model observance probabilities** using the **formula**:

$$P(\mathit{observed}(d)|\mathit{position}(d)) = \left(\frac{1}{\mathit{position}(d)}\right)^{\eta}, \tag{1}$$

where η is a learned parameter.

Then roughly speaking:

- $P(\mathit{observed}(d)|\mathit{position}(d))$ can be inferred from clicks on **swapped pairs**.
- $P(\mathit{clicked}(d)|\mathit{relevance}(d), \mathit{position}(d))$ can be estimated from **click-logs**.

Lambda-IPS: Conclusion

Joachims et al. (2017) **model observance probabilities** using the **formula**:

$$P(\textit{observed}(d)|\textit{position}(d)) = \left(\frac{1}{\textit{position}(d)}\right)^{\eta}, \tag{1}$$

where η is a learned parameter.

Then roughly speaking:

- $P(\textit{observed}(d)|\textit{position}(d))$ can be inferred from clicks on **swapped pairs**.
- $P(\textit{clicked}(d)|\textit{relevance}(d), \textit{position}(d))$ can be estimated from **click-logs**.
- $P(\textit{clicked}(d)|\textit{relevance}(d), \textit{observed}(d))$ can be derived **unbiasedly**.

Lambda-IPS: Conclusion

Joachims et al. (2017) **model observance probabilities** using the **formula**:

$$P(\mathit{observed}(d) | \mathit{position}(d)) = \left(\frac{1}{\mathit{position}(d)} \right)^{\eta}, \tag{1}$$

where η is a learned parameter.

Then roughly speaking:

- $P(\mathit{observed}(d) | \mathit{position}(d))$ can be inferred from clicks on **swapped pairs**.
- $P(\mathit{clicked}(d) | \mathit{relevance}(d), \mathit{position}(d))$ can be estimated from **click-logs**.
- $P(\mathit{clicked}(d) | \mathit{relevance}(d), \mathit{observed}(d))$ can be derived **unbiasedly**.

Counter-factual learning provides a methodology to do this.

Learning from Historical Data: Conclusion

Unbiased learning to rank from historical data:

1. **Infer a user model** from randomized rankings (e.g. swapped pairs).

Learning from Historical Data: Conclusion

Unbiased learning to rank from historical data:

1. **Infer a user model** from randomized rankings (e.g. swapped pairs).
2. **Gather user-interaction logs** using the production ranker (**no randomization required**).

Learning from Historical Data: Conclusion

Unbiased learning to rank from historical data:

1. **Infer a user model** from randomized rankings (e.g. swapped pairs).
2. **Gather user-interaction logs** using the production ranker (**no randomization required**).
3. Learn an **unbiased ranking model** by applying **counter-factual learning**, using the user model and interaction logs.

Learning from Historical Data: Conclusion

Unbiased learning to rank from historical data:

1. **Infer a user model** from randomized rankings (e.g. swapped pairs).
2. **Gather user-interaction logs** using the production ranker (**no randomization required**).
3. Learn an **unbiased ranking model** by applying **counter-factual learning**, using the user model and interaction logs.

Note that:

- Once an accurate user model is obtained, **randomization is no longer needed.**

Learning from Historical Data: Conclusion

Advantages:

- Can **learn unbiasedly** from **user interactions**.
- Learned ranking models **match user preferences** closer than annotations.
- Requires **no randomization** after an accurate user model is learned.

Learning from Historical Data: Conclusion

Advantages:

- Can **learn unbiasedly** from **user interactions**.
- Learned ranking models **match user preferences** closer than annotations.
- Requires **no randomization** after an accurate user model is learned.

Disadvantages:

- **Requires an accurate user model**, which may not always be feasible.
- **No comparison** with *online learning to rank* has been performed.

This is still a **very new and active** area of research.

Online Learning to Rank

Online Learning to Rank: Concept

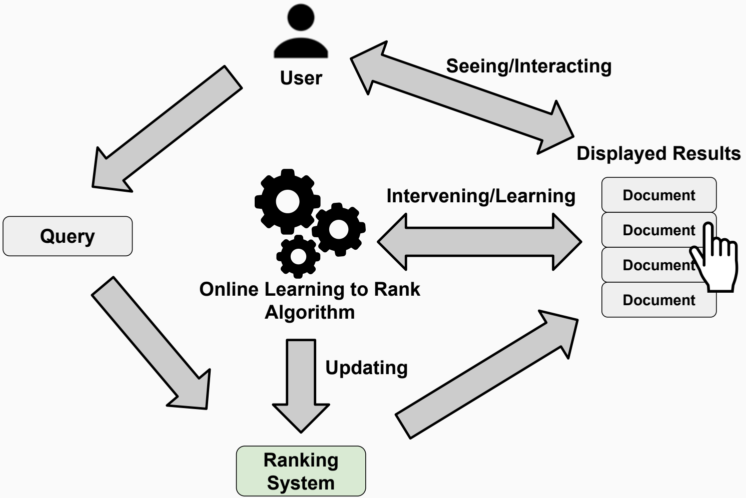
Online Learning to Rank methods have **control over what to display** to the user Yue and Joachims (2009).

Simultantiously they:

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

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

Online Learning to Rank: Visualization



Online Learning to Rank: Advantages

Online learning to rank methods have the **potential advantages**:

- Learn the **true preferences of users** (unlike annotator approaches).

Online Learning to Rank: Advantages

Online learning to rank methods have the **potential advantages**:

- Learn the **true preferences of users** (unlike annotator approaches).
- More **responsive** by **immediately adapting** to users.

Online Learning to Rank: Advantages

Online learning to rank methods have the **potential advantages**:

- Learn the **true preferences of users** (unlike annotator approaches).
- More **responsive** by **immediately adapting** to users.

Online methods also bring a **large risk**:

Online Learning to Rank: Advantages

Online learning to rank methods have the **potential advantages**:

- Learn the **true preferences of users** (unlike annotator approaches).
- More **responsive** by **immediately adapting** to users.

Online methods also bring a **large risk**:

- **Unreliable** methods could **severely worsen the user experience immediately**.

Online Learning to Rank: Properties

Advantages:

- Online methods learn **faster** by **intervening**, (than learning from historical data).
- By **immediately adapting** they can be more **responsive**.

Online Learning to Rank: Properties

Advantages:

- Online methods learn **faster** by **intervening**, (than learning from historical data).
- By **immediately adapting** they can be more **responsive**.
- More suitable for cases where **fewer user interactions** are available.

Online Learning to Rank: Properties

Advantages:

- Online methods learn **faster** by **intervening**, (than learning from historical data).
- By **immediately adapting** they can be more **responsive**.
- More suitable for cases where **fewer user interactions** are available.

Disadvantages:

- The algorithm requires **exploration**, and needs (some) **control** over search results.

Online Learning to Rank: Properties

Advantages:

- Online methods learn **faster** by **intervening**, (than learning from historical data).
- By **immediately adapting** they can be more **responsive**.
- More suitable for cases where **fewer user interactions** are available.

Disadvantages:

- The algorithm requires **exploration**, and needs (some) **control** over search results.
- **No comparisons** performed with *learning from historical data*.

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

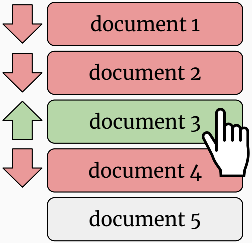
Pairwise Differentiable Gradient Descent optimizes a **Plackett Luce** ranking model, this models a **probabilistic distribution over documents**.

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

$$P(d|D, \theta) = \frac{\exp^{f(\mathbf{d},\theta)}}{\sum_{d' \in D} \exp^{f(\mathbf{d}',\theta)}} \tag{2}$$

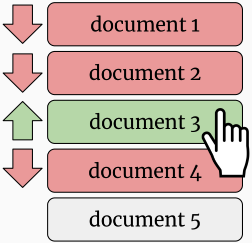
Bias in Pairwise Inference

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**:

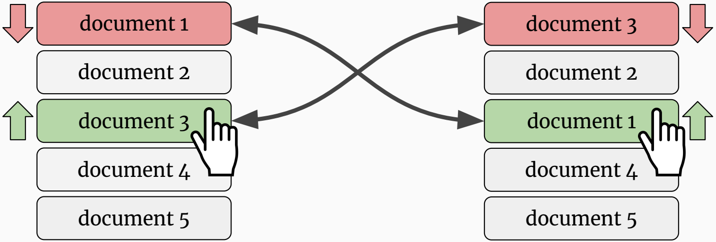


This approach is **biased**:

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

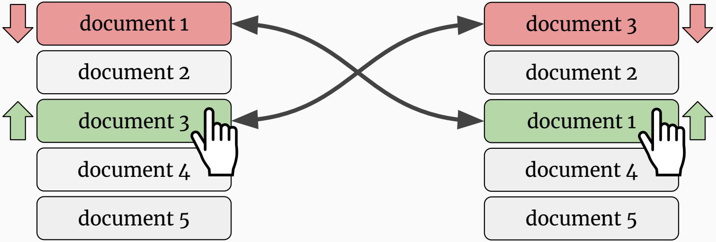
Reversed Pair Rankings

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**:

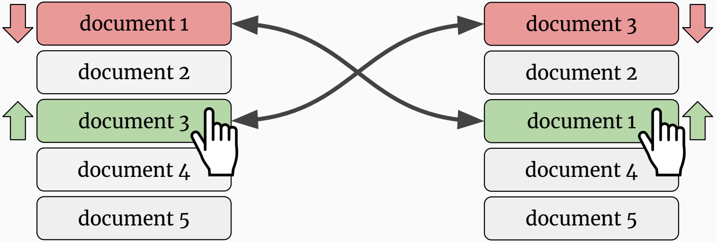


We assume:

- 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**:

$$\rho(d_i, d_j, R) = \frac{P(R^*(d_i, d_j, R)|f, D)}{P(R|f, D) + P(R^*(d_i, d_j, R)|f, D)} \tag{3}$$

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

$$\nabla f(\cdot, \theta) \approx \sum_{d_i >_{\mathbf{c}} d_j} \rho(d_i, d_j, R) \nabla P(d_i \succ d_j | D, \theta). \tag{4}$$

Unbiasedness of Pairwise Differentiable Gradient Descent

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

$$E[\nabla f(\cdot,\theta)] = \sum_{d_i,d_j} \alpha_{ij}(f'(\mathbf{d_i},\theta) - f'(\mathbf{d_j},\theta)). \tag{5}$$

Unbiasedness of Pairwise Differentiable Gradient Descent

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

$$E[\nabla f(\cdot,\theta)] = \sum_{d_i,d_j} \alpha_{ij}(f'(\mathbf{d_i},\theta) - f'(\mathbf{d_j},\theta)). \tag{5}$$

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

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

$$d_i >_{rel} d_j \Leftrightarrow \alpha_{ij} > 0, \tag{7}$$

$$d_i <_{rel} d_j \Leftrightarrow \alpha_{ij} < 0. \tag{8}$$

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:

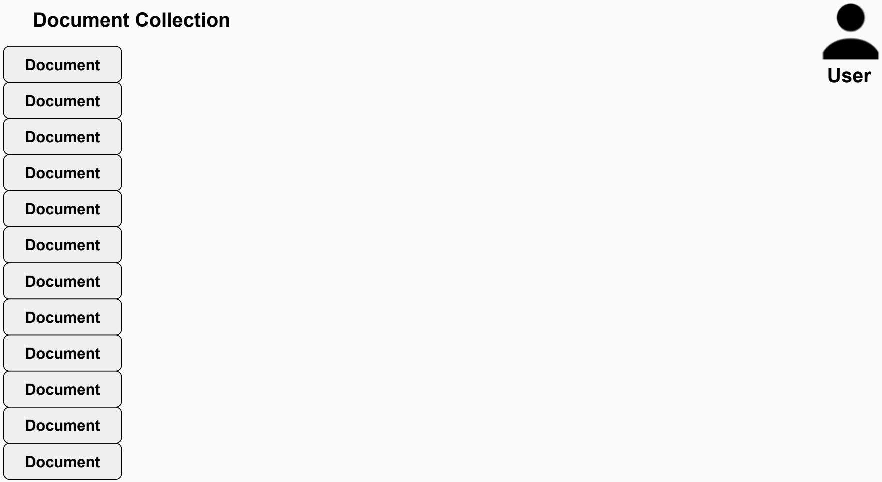
1. Wait for a user query.
2. **Sample** (without replacement) a **ranking** R from the document distribution:

$$P(d|D,\theta_t) = \frac{\exp^{f(\mathbf{d},\theta_t)}}{\sum_{d' \in D} \exp^{f(\mathbf{d}',\theta_t)}}. \tag{9}$$

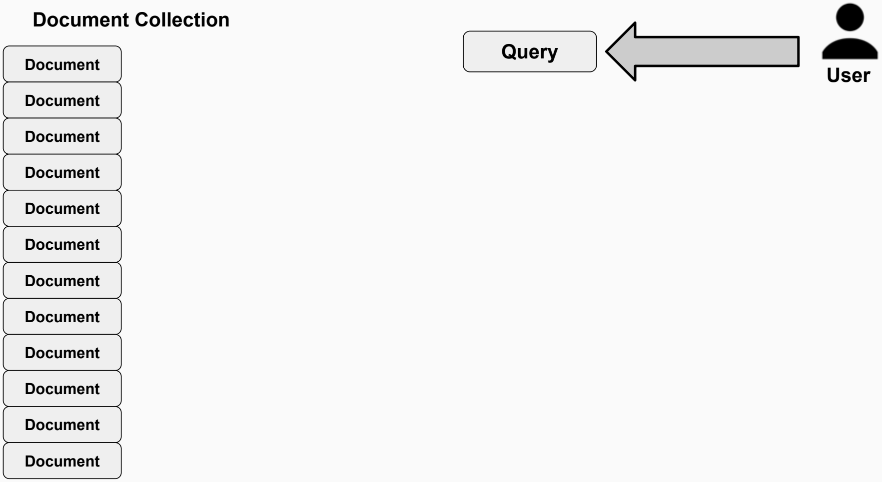
3. **Display** the ranking R to the user.
4. **Infer document preferences** from the **user clicks**: $\mathbf{c}$.
5. **Update** model according to the **estimated (unbiased) gradient**:

$$\nabla f(\cdot,\theta) \approx \sum_{d_i >_{\mathbf{c}} d_j} \rho(d_i, d_j, R) \nabla P(d_i \succ d_j | D, \theta). \tag{10}$$

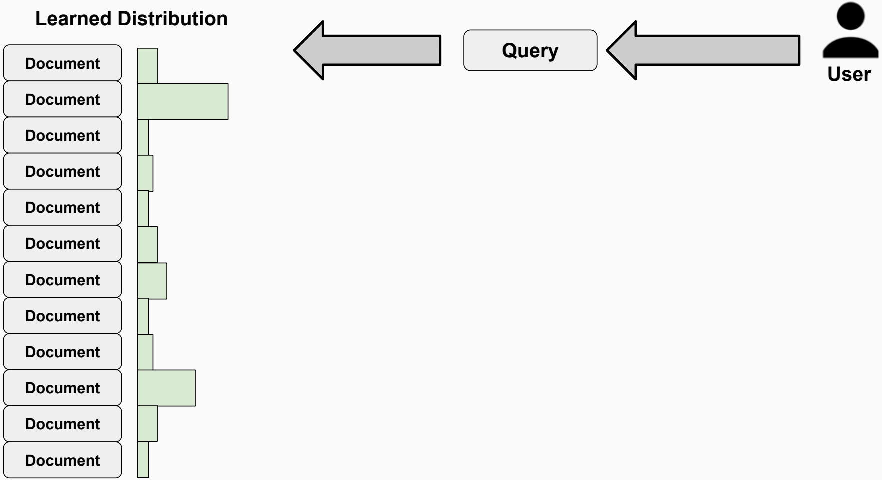
Pairwise Differentiable Gradient Descent: Visualization



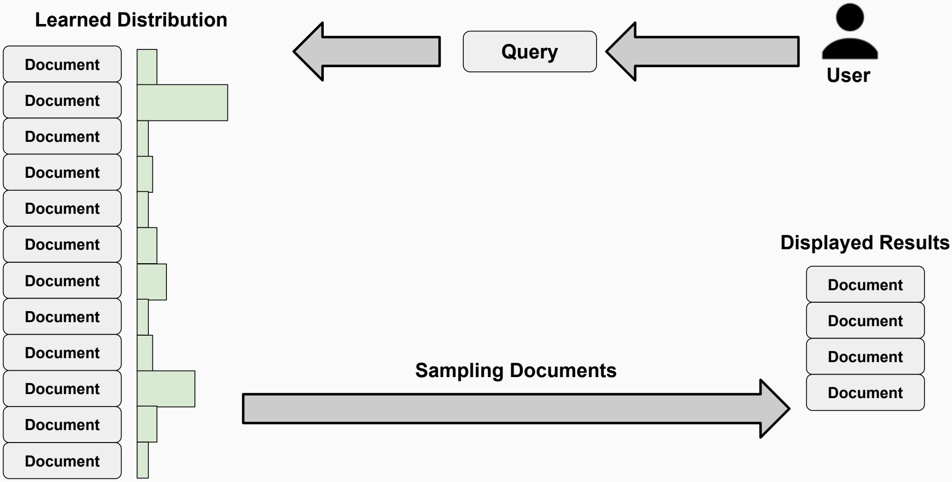
Pairwise Differentiable Gradient Descent: Visualization



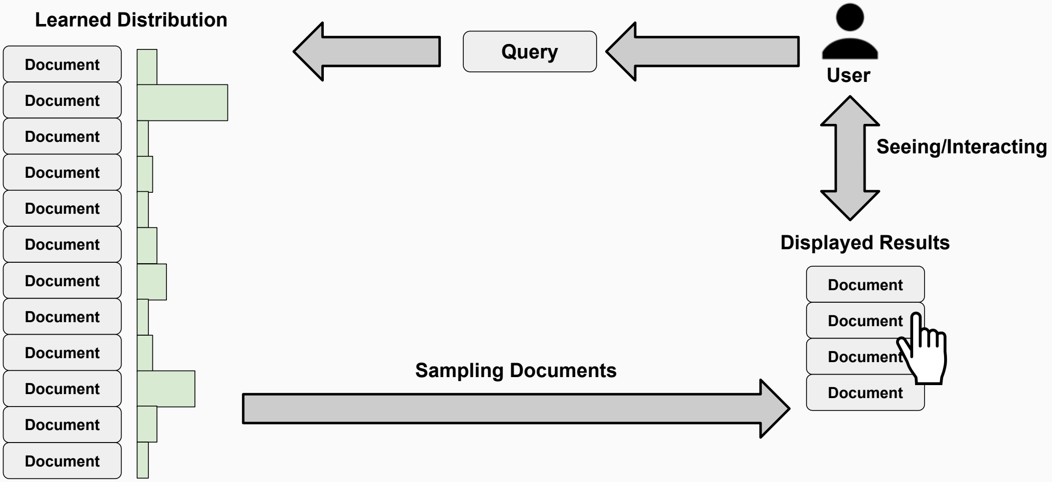
Pairwise Differentiable Gradient Descent: Visualization



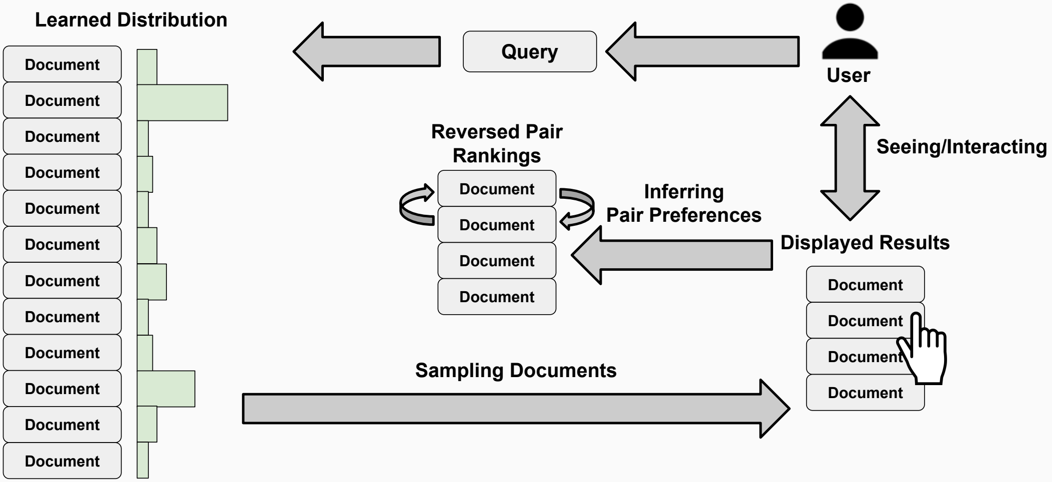
Pairwise Differentiable Gradient Descent: Visualization



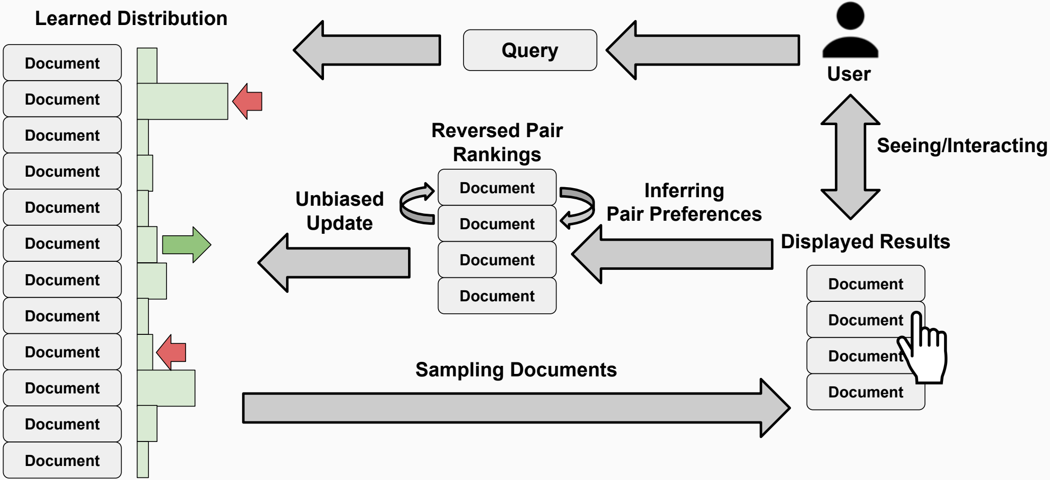
Pairwise Differentiable Gradient Descent: Visualization



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**.

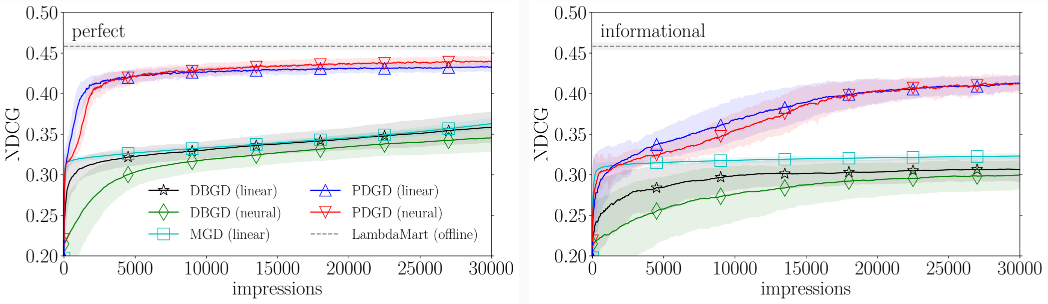
Experiments **repeated** under **varying levels** of noise and bias.

Pairwise Differentiable Gradient Descent: Results

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

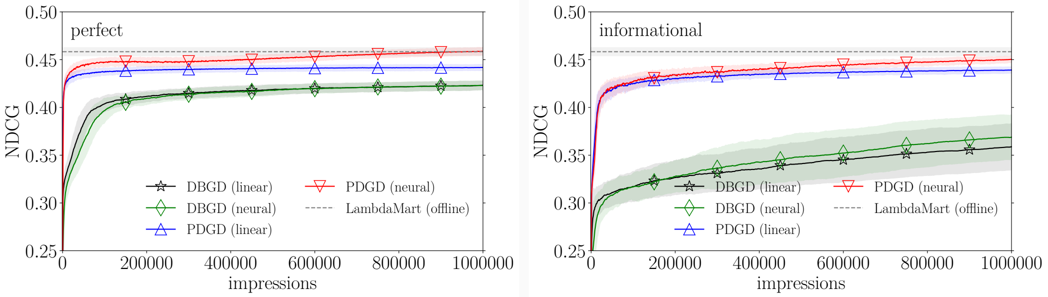
- **Large improvements** in performance of **convergence** under all levels of noise.
- Much **faster learning (better user experience)** under all levels of noise.

Pairwise Differentiable Gradient Descent: Results Short Term



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

Pairwise Differentiable Gradient Descent: Results Long Term



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

Conclusion

Conclusion

Take-away messages:

- **Supervised approaches** to learning to rank are **limited**.
    - **Annotations often disagree** with user preferences.

Conclusion

Take-away messages:

- **Supervised approaches** to learning to rank are **limited**.
    - **Annotations often disagree** with user preferences.
- User interactions solve this problem but bring **noise and biases**.

Conclusion

Take-away messages:

- **Supervised approaches** to learning to rank are **limited**.
    - **Annotations often disagree** with user preferences.
- User interactions solve this problem but bring **noise and biases**.
- **Counter-factual approaches** allow for **unbiased** learning to rank:
    - If an **accurate user model** can be learned, we can **adjust for biases**.
    - **Only** require randomization to **infer a user model**.

Conclusion

Take-away messages:

- **Supervised approaches** to learning to rank are **limited**.
    - **Annotations often disagree** with user preferences.
- User interactions solve this problem but bring **noise and biases**.
- **Counter-factual approaches** allow for **unbiased** learning to rank:
    - If an **accurate user model** can be learned, we can **adjust for biases**.
    - **Only** require randomization to **infer a user model**.
- **Online approaches** allow for **unbiased** and **responsive** learning to rank:
    - **Immediately adapt** to user behaviour, require **fewer interactions** to learn.
    - Perform **randomization** at each step, though limited.

Conclusion

Take-away messages:

- **Supervised approaches** to learning to rank are **limited**.
  - **Annotations often disagree** with user preferences.
- User interactions solve this problem but bring **noise and biases**.
- **Counter-factual approaches** allow for **unbiased** learning to rank:
  - If an **accurate user model** can be learned, we can **adjust for biases**.
  - **Only** require randomization to **infer a user model**.
- **Online approaches** allow for **unbiased** and **responsive** learning to rank:
  - **Immediately adapt** to user behaviour, require **fewer interactions** to learn.
  - Perform **randomization** at each step, though limited.
- Clear, that different situations suit different approaches.

Conclusion

Take-away messages:

- **Supervised approaches** to learning to rank are **limited**.
  - **Annotations often disagree** with user preferences.
- User interactions solve this problem but bring **noise and biases**.
- **Counter-factual approaches** allow for **unbiased** learning to rank:
  - If an **accurate user model** can be learned, we can **adjust for biases**.
  - **Only** require randomization to **infer a user model**.
- **Online approaches** allow for **unbiased** and **responsive** learning to rank:
  - **Immediately adapt** to user behaviour, require **fewer interactions** to learn.
  - Perform **randomization** at each step, though limited.
- Clear, that different situations suit different approaches.
- Unclear, when which approach is preferred (still missing a direct comparison).

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.

A. Chuklin, I. Markov, and M. d. Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015.

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.

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.

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, 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.

Acknowledgments

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