Counterfactual Learning to Rank from User Interactions

Harrie Oosterhuis

June 23, 2020

University of Amsterdam
oosterhuis@uva.nl
https://staff.fnwi.uva.nl/h.r.oosterhuis

Based on the SIGIR 2019 tutorial made with Rolf Jagerman and Maarten de Rijke.

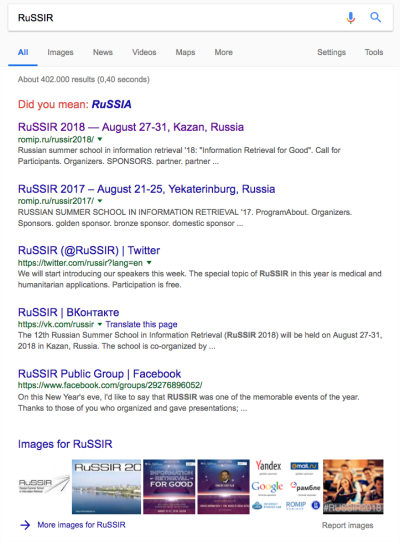
Introduction

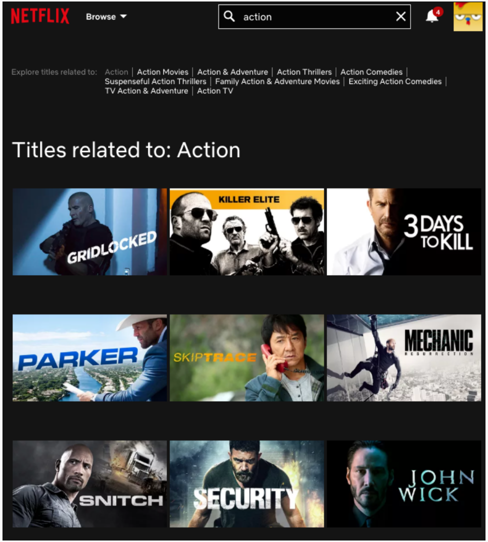
Learning to Rank in Information Retrieval

Learning to Rank is vital to informational retrieval:

- Key component for **search** and **recommendation**.

Ranking in Information Retrieval





Learning to Rank in Information Retrieval

Learning to Rank is a **core task** in informational retrieval:

- Key component for **search** and **recommendation**.

Traditionally learning to rank is **supervised** through **annotated datasets**:

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

Limitations of the Annotated Datasets

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 **behavior** is indicative of their **preferences**.

User interactions also bring their **own difficulties**:

- 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.
  - 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.
  - **Item selection bias:** Interactions are **limited** to the **presented** documents.
  - **Presentation bias:** Results that are **presented differently** will be **treated differently**.
  - ...

The Golden Triangle

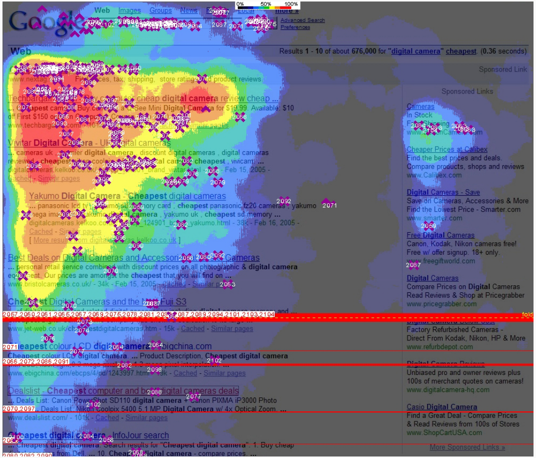


Image source: `http://www.mediative.com/`

Learning from User Interactions: Goal

Goal of unbiased learning to rank:

- Optimize a ranker w.r.t. **relevance preferences** of users from their interactions.
- **Avoid** being **biased by other factors** that influence interactions.

Counterfactual Evaluation

Counterfactual Evaluation: Introduction

Evaluation is incredibly **important before deploying** a ranking system.

However, with the **limitations of annotated datasets**,
can we **evaluate** a ranker **without deploying** it or **annotated data**?

Counterfactual Evaluation:
Evaluate a new ranking function f_θ using **historical interaction data** (e.g., clicks) collected from a previously deployed ranking function f_{deploy}.

Counterfactual Evaluation: Full Information

If we **know** the **true relevance labels** ($y(d_i)$ for all i), we can compute any additive linearly decomposable IR metric.

In this talk we will assume relevance is binary:

$$y(d_i) \in \{0, 1\},$$

and minimize the **Average Relevant Position**:

$$\Delta(f_\theta, D, y) = \sum_{d_i \in D} \mathit{rank}(d_i \mid f_\theta, D) \cdot y(d_i).$$

Counterfactual Evaluation: Full Information

$y(d_1) = 1$	Document d_1
$y(d_2) = 0$	Document d_2
$y(d_3) = 0$	Document d_3
$y(d_4) = 1$	Document d_4
$y(d_5) = 0$	Document d_5

Counterfactual Evaluation: Partial Information

We often do not know the true relevance labels ($y(d_i)$), but can only observe implicit feedback in the form of, e.g., clicks:

- A click c_i on document d_i is a **biased and noisy indicator** that d_i is relevant
- A missing click does **not** necessarily indicate non-relevance

Counterfactual Evaluation: Clicks

$y(d_1) = 1$	Document d_1	$c_1 = 1$
$y(d_2) = 0$	Document d_2	$c_2 = 0$
$y(d_3) = 0$	Document d_3	$c_3 = 1$
$y(d_4) = 1$	Document d_4	$c_4 = 0$
$y(d_5) = 0$	Document d_5	$c_5 = 0$

Counterfactual Evaluation: Clicks

Remember that there are many reasons why a click on a document may **not** occur:

- **Relevance**: the document may not be relevant.
- **Observance**: the user may not have examined the document.
- **Miscellaneous**: various random reasons why a user may not click.

Some of these reasons are considered to be:

- **Noise**: averaging over many clicks will remove their effect.
- **Bias**: averaging will **not** remove their effect.

Counterfactual Evaluation: Examination User Model

If we **only** consider **examination** and **relevance**, a user click can be modelled by:

- The probability of document d_i **being examined** $(o_i = 1)$ in a ranking R:

$$P(o_i = 1 \mid R, d_i)$$

- The probability of a **click** $c_i = 1$ on d_i given its **relevance** $y(d_i))$ and whether it was **examined** o_i:

$$P(c_i = 1 \mid o_i, y(d_i))$$

- **Clicks only occur on examined documents**, thus the probability of a click in ranking R is:

$$P(c_i = 1 \wedge o_i = 1 \mid y(d_i), R) = P(c_i = 1 \mid o_i = 1, y(d_i)) \cdot P(o_i = 1 \mid R, d_i)$$

Counterfactual Evaluation: Naive Estimator

A **naive way** to estimate is to assume clicks are a unbiased relevance signal:

$$\hat{\Delta}_{\mathit{NAIVE}}(f_\theta, D, c) = \sum_{d_i \in D} \mathit{rank}(d_i \mid f_\theta, D) \cdot c_i.$$

Even if **no click noise** is present: $P(c_i = 1 \mid o_i = 1, y(d_i)) = y(d_i)$, this estimator is **biased** by the examination probabilities:

$$\begin{aligned} \mathbb{E}_o[\hat{\Delta}_{\mathit{NAIVE}}(f_\theta, D, c)] &= \mathbb{E}_o\left[\sum_{d_i : o_i = 1 \wedge y(d_i) = 1} \mathit{rank}(d_i \mid f_\theta, D)\right] \\ &= \sum_{d_i : y(d_i) = 1} P(o_i = 1 \mid R, d_i) \cdot \mathit{rank}(d_i \mid f_\theta, D). \end{aligned}$$

Counterfactual Evaluation: Naive Estimator Bias

The biased estimator **weights documents** according to their **examination probabilities** in the ranking R displayed during **logging**:

$$\mathbb{E}_o[\hat{\Delta}_{\mathit{NAIVE}}(f_\theta, D, c)] = \sum_{d_i : y(d_i)=1} P(o_i = 1 \mid R, d_i) \cdot \mathit{rank}(d_i \mid f_\theta, D).$$

Documents at higher ranks are more likely to be examined: **position bias**.

Position bias causes **logging-policy-confirming** behavior:

- Documents displayed at **higher ranks during logging** are incorrectly considered as **more relevant**.

Inverse Propensity Scoring

Counterfactual Evaluation: Inverse Propensity Scoring

Counterfactual evaluation accounts for bias using **Inverse Propensity Scoring (IPS)**:

$$\hat{\Delta}_{\mathit{IPS}}(f_\theta, D, c) = \sum_{d_i \in D} \frac{\mathit{rank}(d_i \mid f_\theta, D)}{P(o_i = 1 \mid R, d_i)} \cdot c_i,$$

- $\mathit{rank}(d_i \mid f_\theta, D)$: (weighted) rank of document d_i by ranker f_θ,
- c_i: observed click on the document in the log,
- $P(o_i = 1 \mid R, d_i)$: examination probability of d_i in ranking R displayed during logging.

This is an **unbiased estimate** of any additive linearly decomposable IR metric (Joachims et al., 2017).

Counterfactual Evaluation: Proof of Unbiasedness

If no click noise is present, this provides an **unbiased estimate**:

$$
\begin{aligned}
\mathbb{E}_o[\hat{\Delta}_{\mathit{IPS}}(f_\theta, D, c)] &= \mathbb{E}_o\left[\sum_{d_i \in D} \frac{\mathit{rank}(d_i \mid f_\theta, D)}{P(o_i = 1 \mid R, d_i)} \cdot c_i\right] \\
&= \mathbb{E}_o\left[\sum_{d_i : o_i = 1 \wedge y(d_i) = 1} \frac{\mathit{rank}(d_i \mid f_\theta, D)}{P(o_i = 1 \mid R, d_i)}\right] \\
&= \sum_{d_i : y(d_i) = 1} \frac{P(o_i = 1 \mid R, d_i) \cdot \mathit{rank}(d_i \mid f_\theta, D)}{P(o_i = 1 \mid R, d_i)} \\
&= \sum_{d_i \in D} \mathit{rank}(d_i \mid f_\theta, D) \cdot y(d_i) \\
&= \Delta(f_\theta, D, y).
\end{aligned}
$$

Propensity-weighted Learning to Rank

Propensity-weighted Learning to Rank (LTR)

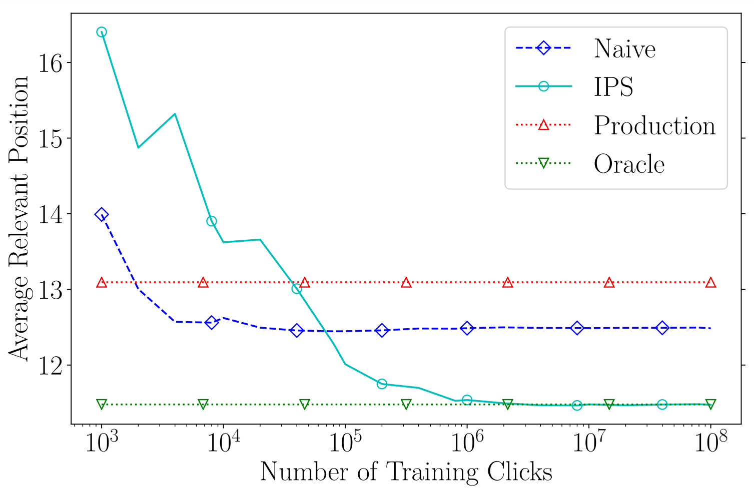
The inverse-propensity-scored estimator can unbiasedly estimate performance:

$$\hat{\Delta}_{\mathit{IPS}}(f_\theta, D, c) = \sum_{d_i \in D} \frac{\mathit{rank}(d_i \mid f_\theta, D)}{P(o_i = 1 \mid R, d_i)} \cdot c_i.$$

Similar to the **standard ranking objective** but **weighted** per document, can be optimized with **small adjustments** to **standard learning to rank methods** (Agarwal et al., 2019).

Propensity-weighted LTR: Results

Simulated results on the Yahoo! Webscope dataset (Chapelle and Chang, 2011) .



Estimating Position Bias

Estimating Position Bias

Recall that position bias is a form of bias where higher positioned results are more likely to be observed and therefore clicked.

Assumption: The **observation probability** only depends on the rank of a document:

$$P(o_i = 1 \mid i).$$

The objective is now to **estimate**, for each rank i, the propensity $P(o_i = 1 \mid i)$.

Estimating Position Bias

RandTop-n Algorithm:

Document d_1	Document d_3	Document d_2	Rank 1
Document d_2	Document d_4	Document d_1	Rank 2
Document d_3	Document d_1	Document d_4	Rank 3
Document d_4	Document d_2	Document d_3	Rank 4

Estimating Position Bias

Uniformly **randomizing** the top n results may negatively impacts users during logging.

There are various methods that minimize the impact to the user:

- **RandPair:** Choose a pivot rank k and only swap a random other document with the document at this pivot rank (Joachims et al., 2017).
- **Interventional Sets:** Exploit inherent "randomness" in data coming from multiple rankers (e.g., A/B tests in production logs) (Agarwal et al., 2017).

Also methods that estimate bias without any randomization:

- **Expectation-Maximization approach** (Wang et al., 2018),
- **Dual Learning Objective** (Ai et al., 2018).

Applying Counterfactual LTR in Practice

Overview of Application Process

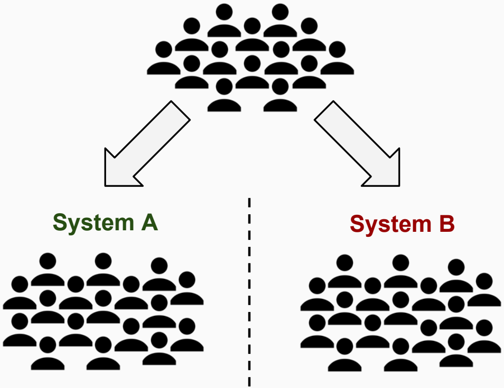
Recommended steps to apply counterfactual LTR:

- A/B testing
- Interaction Logging
- Position bias estimation
- Counterfactual LTR
- Post-deployment evaluation

A/B Testing

Randomly assign a percentage of **users** to system B and the rest to system A.

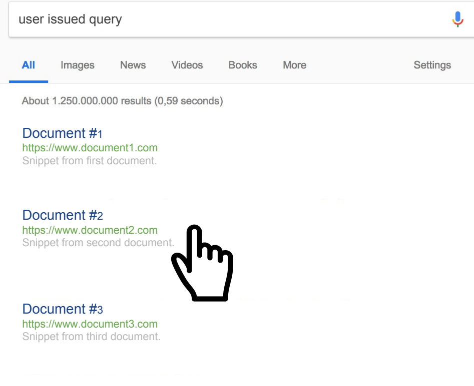
The differences in performance per group can **reliably compare A to B**.



Interaction Logging

Log every interaction that takes place and its context:

- **Actions taken by user:**
  - Query issued, clicks, purchases, dwell-time, . . .
- **Actions taken by system:**
  - Items displayed, layout, descriptions displayed, prices offered, . . .
- **Item information:**
  - Item features, popularity, category info, entity linking, . . .
- **Contextual information:**
  - User info, time & date, mobile/web interface, . . .



Disclaimer: I'm not a lawyer, check these decision with your legal/ethics department.

Position Bias Estimation

A position bias model needs to be inferred before counterfactual learning or evaluation.

Most efficient with randomization during logging:

- Random shuffle top-n.
- Randomly swap pairs of items.
- Apply different rankers during the same period of time (Automatically happens when A/B testing).

Less efficient but non-intrusive with no randomization:

- Estimate through Expectation-Maximization or a dual learning objective.

Remember that **bias depends on the ranking layout**,
i.e. layout changes $\rightarrow$ bias model may need to be updated.

Performing Counterfactual Learning to Rank

Optimize using a counterfactual learning to rank method, the bias model and any logged data (no randomization needed).

The following choices have to be made:

- The choice of **features** the ranking model uses
  (logged data may limit your choices.).
- What **ranking model** to use? e.g. linear model, neural model, ...
- **Model parameters:** number of layers, activation functions, ...
- **Optimization parameters:** learning rate, regularization weight, ...

All these choices can be made using unbiased evaluation,
massive speed boost to research and development.

Post-deployment Evaluation

Never blindly trust anything you may deploy to users:

- Before fully deploying a model,
  deploy to a small percentage and evaluate with A/B testing.

Errors can always sneak into the results of counterfactual evaluation:

- Bugs in code for counterfactual evaluation or learning,
  or any other part of the pipeline.
- Bias model may be incorrect or outdated.
- Explicit or implicit assumptions can be false for your users and application.

Conclusion

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**.
- **Counterfactual approaches** allow for **unbiased** learning to rank:
    - By **modelling users' position bias**, we can **remove its effect** during learning.
    - **Only** requires randomization to **infer a user model**.
- Counterfactual evaluation **predicts improvements** to your system **without deployment**.

Final Message

Final message:

- Remember that unbiased LTR means **unbiased LTR w.r.t. position bias**, **always expect that there are more biases than we are aware of.**

Thank you for listening!

Notation

Notation Used in the Slides i

Definition	**Notation**	**Example**
Query	q	–
Candidate documents	D	–
Document	$d \in D$	–
Ranking	R	$(R_1, R_2, \ldots, R_n)$
Document at rank i	R_i	$R_i = d$
Relevance	$y : D \to \mathbb{N}$	$y(d) = 2$
Ranker model with weights θ	$f_\theta : D \to \mathbb{R}$	$f_\theta(d) = 0.75$
Click	$c_i \in \{0, 1\}$	–
Observation	$o_i \in \{0, 1\}$	–
Rank of d when f_θ ranks D	$\textit{rank}(d \mid f_\theta, D)$	$\textit{rank}(d \mid f_\theta, D) = 4$

Notation Used in the Slides ii

Differentiable upper bound on *rank*$(d, \mid f_\theta, D)$	$\overline{\mathit{rank}}(d, \mid f_\theta, D)$	–
Average Relevant Position metric	*ARP*	–
Discounted Cumulative Gain metric	*DCG*	–
Precision at k metric	*Prec*$@k$	–
A performance measure or estimator	Δ	–

Resources i

- Tensorflow Learning to Rank, allows for inverse propensity scoring:
  `https://github.com/tensorflow/ranking`
- Inverse Propensity Scored Rank-SVM:
  `https://www.cs.cornell.edu/people/tj/svm_light/svm_proprank.html`
- Data and code for comparing counterfactual and online learning to rank
  `http://github.com/rjagerman/sigir2019-user-interactions`

References i

A. Agarwal, S. Basu, T. Schnabel, and T. Joachims. Effective evaluation using logged bandit feedback from multiple loggers. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 687–696. ACM, 2017.

A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims. A general framework for counterfactual learning-to-rank. In *42nd International ACM SIGIR Conference on Research & Development in Information Retrieval*, page (to appear). ACM, 2019.

Q. Ai, K. Bi, C. Luo, J. Guo, and W. B. Croft. Unbiased learning to rank with unbiased propensity estimation. In *Proceedings of the 41st International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 385–394. ACM, 2018.

O. Chapelle and Y. Chang. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research*, 14:1–24, 2011.

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.

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.

Acknowledgments

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