Robust Generalization and Safe Query-Specialization in Counterfactual Learning to Rank

Harrie Oosterhuis\textsuperscript{1}, Maarten de Rijke\textsuperscript{2,3}

April 20, 2021

Radboud University\textsuperscript{1}, University of Amsterdam\textsuperscript{2}, Ahold Delhaize\textsuperscript{3}
harrie.oosterhuis@ru.nl, derijke@uva.nl
https://twitter.com/HarrieOos
A Dilemma
Should item D be placed on top?
Feature-Based Model (trained on 10,000,000 interactions)

Item A
0 clicks

Item B
0 clicks

Item C
0 clicks

Item D
10 clicks

Should item D be placed on top?
Feature-Based Model (trained on 10,000,000 interactions)

Should item D be placed on top?
Dilemma

Feature-Based Model
(trained on 10,000,000 interactions)

Item A
0 clicks

Item B
0 clicks

Item C
0 clicks

Item D
1,000 clicks

Should item D be placed on top?
Dilemma

Feature-Based Model
(trained on 10,000,000 interactions)

Item A
0 clicks

Item B
0 clicks

Item C
0 clicks

Item D
10,000 clicks

Should item D be placed on top?
Introduction: Counterfactual Learning to Rank
Counterfactual Learning to Rank:

- **Learning from clicks** while correcting for interaction biases caused during gathering of the data (Joachims et al., 2017).

We correct for Position Bias (Craswell et al., 2008) using the policy-aware approach (Oosterhuis and de Rijke, 2020):

\[
\hat{R}_{IPS}(d) = \frac{1}{N} \sum_{i=1}^{N} \frac{c_i(d)}{\rho_d}.
\]
The estimator $\hat{R}_{IPS}$ can be used to unbiasedly estimate a **ranking loss**, e.g. DCG-loss.

For instance, we can optimize a **linear model**:

![Graph showing NDCG vs. Average Number of Interactions per Query]

Performance on the Yahoo! Webscope LTR dataset with simulated biased and noisy clicks.
A tabular model simply stores the $\hat{R}_{IPS}$ values and ranks accordingly:

Performance on the Yahoo! Webscope LTR dataset with simulated biased and noisy clicks.
Feature-Based Models:

- **Generalized performance**: Robust over all queries.
- Can be applied to **previously unseen queries**.
- Performance is often **limited** by the quality of the available **features**.

Tabular Models:

- **Specialized performance**: Independent behavior per query.
- **Cannot** be applied to **previously unseen queries**.
- Performance is **not limited by features**, can learn any possible ranking behavior.
Model Trade-Off Visualized

Performance on the Yahoo! Webscope LTR dataset with simulated biased and noisy clicks.
Performance on the Yahoo! Webscope LTR dataset with simulated biased and very noisy clicks.
We want to have both:

- the **safe robust behavior** of feature-based models,
- the **high-performance at convergence** of tabular models,

and avoid

- the **detrimental initial performance** of tabular models.
Performance Bounds
Jagerman et al. (2020) introduce the Safe Exploration Algorithm:
Jagerman et al. (2020) introduce the Safe Exploration Algorithm:
Jagerman et al. (2020) introduce the Safe Exploration Algorithm:
We introduce a new approach that bounds relative performance:
We introduce a new approach that bounds relative performance:
We introduce a new approach that bounds relative performance:
We introduce a new approach that bounds relative performance:
We introduce a new approach that bounds relative performance:
The Generalization and Specialization Framework
Optimization Phase

- train a feature-based model on data over all queries,
- compute the values for the tabular model.

Serving Procedure

- choose between the logging policy and feature-based model according to bound computed over all data,
- then choose between tabular model and previous choice according to bound computed only on data for the specific query.
GENSPEC overview

Logging policy $\pi_0$ to Rankings/Queries

Dividing clickdata per query

Users

Feature-based model $\pi_0$ to Tabular model $\pi_D$

Queries $q = 1$ to $q = 5$

Deployed $\pi_1$ to $\pi_5$

Users

Rankings/Queries

Dividing clickdata per query
Experimental Results
Performance on the Yahoo! Webscope Dataset

Experiment with simulated biased and noisy clicks.
Performance on the Yahoo! Webscope Dataset

Experiment with simulated biased and very noisy clicks.
Conclusion
Different models have different advantages and risks:

- **feature-based:** robust generalized performance,
- **tabular:** high performance at convergence, initial detrimental performance.

We introduced the **Generalization and Specialization (GENSPEC) framework:**

- optimizes two models for **generalization** and **specialization**,
- uses performance bounds to **safely choose** to deploy per query.

We can have both **robust generalization** and **safe query-specialization** in counterfactual learning to rank.

Continue our work: [https://github.com/Harrie0/2021WWW-GENSPEC](https://github.com/Harrie0/2021WWW-GENSPEC)


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