ULTRE framework: a framework for Unbiased Learning to Rank Evaluation based on simulation of user behavior

Yurou Zhao¹, Jiaxin Mao¹, Qingyao Ai²

¹ Renmin University of China, China
² University of Utah, USA
Outline

• Background
  – Unbiased Learning to Rank (ULTR)
  – User simulation-based evaluation approach of evaluating ULTR models and its limitations

• ULTRE (Unbiased Learning to Rank Evaluation) framework
  – two affiliated evaluation protocols

• Experiments

• Conclusion and Future works

• NTCIR16–ULTRE task (Advertisement time)
Background

• Unbiased Learning to Rank (ULTR):
  – Users’ interaction with search systems can reflect their implicit relevance feedback for search results
  – Cheap but biased
  – ULTR: Learning an unbiased ranker from biased user feedback
  – Counterfactual/Offline LTR and Online LTR
Background

• The evaluation of ULTR
  – Ideally, we should train and evaluate the ULTR model on real search logs and online search systems
  – Not possible for academic researchers due to a lack of access to query logs and online systems
  – Most existing ULTR studies utilize a simulation-based evaluation approach
Background

• Simulation–based evaluation of ULTR

User behavior model → Simulated clicks → ULTR models

Trad. LTR dataset w/ relevance annotation → Evaluate
Background

• Limitations with the evaluation of ULTR
  – No standard evaluation settings or shared evaluation benchmarks for the ULTR community
  – Most studies only use a single user simulation model
    • may not fully capture the diverse patterns of real user behavior
    • may introduce systematic biases into the comparison among ULTR models
ULTRE Framework

Step 1: Real click logs

Step 2: Click Simulators

Step 3: Training queries

Step 4: Simulated clicks

Step 5: Synthetic train sets

Step 6: Have participants received 100% user impressions?

Yes: Train an online ULTR model

No: Train an offline or online ULTR model?

Step 7: Evaluation Results

Trained ULTR models:
- Online models: DBGD, PDGD, ...
- Offline models: SVMRank+IPW, DNN+DLA, ...
ULTRE Framework - Stage 1

Stage 1: Simulation of clicks

Step 1: Real click logs
Step 2: Click Simulators
Step 3: Training queries
Step 4: Simulated clicks
Step 5: Synthetic train sets
Step 6: Have participants received 100% user impressions?
   Yes: Train an online ULTR model
   No: Train an offline or online ULTR model?
Step 7: Evaluation Results
ULTRE Framework - Stage 1

**Step 1**
Real click logs

**Step 2**
Click Simulators

**Step 3**
Training queries

**Step 4**
Simulated clicks

**Step 5**
Synthetic train sets
ULTRE Framework - Stage 2

Stage 2: Training of ULTR models

- **Step 1**: Real click logs
- **Step 2**: Click Simulators
- **Step 3**: Training queries
- **Step 4**: Simulated clicks
- **Step 5**: Synthetic train sets
- **Step 6**: Online or offline ULTR model
- **Step 7**: Evaluation Results

**User Behavior Modes**
- PBM
- DCM
- UBM
- MCM

**Stage 2: Training of ULTR models**

- **No**: Have participants received 100% user impressions?
  - Train an online ULTR model

- **Yes**: Trained ULTR models
  - Online models: DBGD, PDGD, ...
  - Offline models: SVMRank+IPW, DNN+DLA, ...
  - Train an offline or online ULTR model
ULTRE Framework - Stage 3

Stage 3: Evaluation of ULTR models
Application of ULTRE framework—Unbiased Learning to Rank Evaluation Task (ULTRE) in NTCIR-16

• Provide a shared evaluation task and benchmark for the evaluation of different ULTR
• Two evaluation protocols for the ULTRE task
  – Describe how the task organizers (i.e. TOs) interact with the participants and work together to evaluate the ULTR models
  – Offline protocol for offline/counterfactual LTR models
  – Online protocol for online LTR models
**Evaluation protocol for offline ULTR models**

**Step 1:** TOs generate simulated click logs for all training queries
- Use four click models (PBM/DCM/UBM/MCM)
- Train and calibrate the click models with real click logs

**Step 2:** Participants train ULTR models with simulated click logs and submit the ranking lists (runs) for validation/test queries

**Step 3:** TOs evaluate the runs
- Show the results on validation set on the leaderboard
- Release the official results on test set in the final report
Evaluation protocol for online UTLR models

Step 1: Participants submit the ranking lists for training queries
  • Specify that they want to receive x% of impressions

Step 2: TOs sample training queries and generate simulated clicks on the ranking lists submitted by participants

Step 3: Participants update their models with the simulated clicks

Repeat Step 1-Step 3 until participants receive 100% of impressions

Step N: TOs evaluate results on validation/test set
  • Show the results on validation set on the leaderboard
  • Release the official results on test set in the final report
Experiments

• RQ: Can we evaluate existing ULTR models properly with the ULTRE framework?
Experiment--Evaluating ULTR models with the ULTRE framework

• Dataset (corresponds to the traditional LTR dataset in ULTRE framework)
  – Based on SogouSRR [Zhang et al. 2008]
  – 1,211 unique queries with 10 successfully crawled results
  – 1011 for training, 100 for validation and 100 for testing

• Simulation Setup
  – Train LambdaMART with 1% data randomly sampled from the original training set (with 5-level relevance annotations) and use it as the offline production ranker
  – Follow the click–simulation process in the ULTRE framework.

• Model Setup and Evaluation
  – Evaluation metrics: nDCG@5
# Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>PBM</th>
<th>DCM</th>
<th>UBM</th>
<th>MCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production ranker (Baseline)</td>
<td></td>
<td></td>
<td>0.7815</td>
<td></td>
</tr>
<tr>
<td>Full-info (Skyline)</td>
<td></td>
<td></td>
<td>0.8182</td>
<td></td>
</tr>
<tr>
<td>PBM–IPS (IPW)</td>
<td>0.8064</td>
<td>0.7826</td>
<td>0.8017</td>
<td>0.7647</td>
</tr>
<tr>
<td>CM–IPS</td>
<td>0.7894</td>
<td>0.8050</td>
<td>0.7932</td>
<td>0.7778</td>
</tr>
<tr>
<td>DLA</td>
<td>0.8119</td>
<td>0.8173</td>
<td>0.8107</td>
<td>0.7932</td>
</tr>
</tbody>
</table>
## Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>PBM</th>
<th>DCM</th>
<th>UBM</th>
<th>MCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production ranker (Baseline)</td>
<td></td>
<td></td>
<td><strong>0.7815</strong></td>
<td></td>
</tr>
<tr>
<td>Full-info (Skyline)</td>
<td></td>
<td></td>
<td><strong>0.8182</strong></td>
<td></td>
</tr>
<tr>
<td>PBM-IPS (IPW)</td>
<td><strong>0.8064</strong></td>
<td>0.7826</td>
<td>0.8017</td>
<td>0.7647</td>
</tr>
<tr>
<td>CM-IPS</td>
<td>0.7894</td>
<td><strong>0.8050</strong></td>
<td>0.7932</td>
<td>0.7778</td>
</tr>
<tr>
<td>DLA</td>
<td>0.8119</td>
<td>0.8173</td>
<td>0.8107</td>
<td>0.7932</td>
</tr>
</tbody>
</table>

When the used behavior models used in click simulation and the correction method of bias are consistent, the results are better than the case in which they don’t agree.
Evaluation results

<table>
<thead>
<tr>
<th></th>
<th>PBM</th>
<th>DCM</th>
<th>UBM</th>
<th>MCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production ranker (Baseline)</td>
<td></td>
<td></td>
<td>0.7815</td>
<td></td>
</tr>
<tr>
<td>Full-info (Skyline)</td>
<td></td>
<td></td>
<td>0.8182</td>
<td></td>
</tr>
<tr>
<td>PBM–IPS (IPW)</td>
<td>0.8064</td>
<td>0.7826</td>
<td>0.8017</td>
<td>0.7647</td>
</tr>
<tr>
<td>CM–IPS</td>
<td>0.7894</td>
<td>0.8050</td>
<td>0.7932</td>
<td>0.7778</td>
</tr>
<tr>
<td>DLA</td>
<td>0.8119</td>
<td>0.8173</td>
<td>0.8107</td>
<td>0.7932</td>
</tr>
</tbody>
</table>

DLA is more robust and more adaptive to the change of user behavior assumption used in the click simulation due to its unification of learning propensity weights (used to correct bias in click data) and leaning ranking models.
Conclusion and Future work

• Contribution
  – Propose the ULTRE framework that aims to improve the simulation approach used in previous ULTR evaluation
  – Experiments show that ULTRE framework can provide simulated–based training sets of both quality and diversity and enables us to conduct a thorough and relatively objective comparison of different ULTR models.

• Future work
  – Adopt neural user behavior models such as Context–aware Click Simulator (CCS) [Zhang et al. 2019] for the click simulation
  – Implement online service and Compare online ULTR models under the ULTRE framework
Unbiased Learning to Rank Evaluation Task (ULTRE) in NTCIR-16 (NII Test Collection for IR Systems)

- ULTRE task is a pilot task in NTCIR–16 (http://research.nii.ac.jp/ntcir/ntcir–16/)
- Provide a shared benchmark and evaluation service for ULTR
- NTCIR16–ULTRE task official website: http://ultre.online/

Looking for your participation!
Thanks!

Q&A
Construct Click Simulators

• Use four user simulation models
  – Position-Based Model (PBM) [Craswell et al. 2008]: a click model that assumes the click probability of a search result only depends on its relevance and its ranking position.
  – Dependent Click Model (DCM) [Guo et al. 2009]: a click model that is based on the cascade assumption that the user will sequentially examine the results list and find attractive results to click until she feels satisfied with the clicked result.
  – User Browsing Model (UBM) [Dupret et al. 2008]: a click model that assumes the examination probability on a search result depends on its ranking position and the distance to the last clicked result.
  – Mobile Click Model (MCM) [Mao et al. 2018]: a click model that considers the click necessity bias (i.e. some vertical results can satisfy users’ information need without a click) in user clicks.

• Train and calibrate the user simulation models with real query logs
Experiments

• RQ: How do different click simulators perform in predicting clicks and generating synthetic click logs?
Examining click simulators

- Dataset (corresponds to the *Real click logs* in the ULTRE framework)
  - real search log dataset released by Chinese commercial search engine Sogou.com (1.6 million sessions, 1211 unique queries)

- Evaluation metrics
  - LogLikelihood (LL) and Perplexity (PPL)
  - Reverse/Forward PPL [Dai et al. 2021]
Performance on predicting clicks

<table>
<thead>
<tr>
<th></th>
<th>LL</th>
<th>PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCM</td>
<td>-0.1848</td>
<td>1.2363</td>
</tr>
<tr>
<td>PBM</td>
<td>-0.1721</td>
<td>1.2059</td>
</tr>
<tr>
<td>UBM</td>
<td>-0.1513</td>
<td>1.2029</td>
</tr>
<tr>
<td>MCM</td>
<td>-0.1503</td>
<td>1.1787</td>
</tr>
</tbody>
</table>
Quality of generated click logs

Reverse PPL: the PPL of a surrogate model (an intermediary to evaluate the similarity between the generated samples and the real data samples) that is trained on generated samples and evaluate on real data.

Forward PPL: the PPL of a surrogate model that is trained on real data and evaluated on generated samples.

<table>
<thead>
<tr>
<th></th>
<th>Surrogate DCM</th>
<th>Surrogate PBM</th>
<th>Surrogate UBM</th>
<th>Surrogate MCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real data</td>
<td>1.2363</td>
<td>1.2059</td>
<td>1.2029</td>
<td>1.1787</td>
</tr>
<tr>
<td>DCM samples</td>
<td>–</td>
<td>1.2374</td>
<td>1.2350</td>
<td>1.2191</td>
</tr>
<tr>
<td>PBM samples</td>
<td>1.2824</td>
<td>–</td>
<td>1.2061</td>
<td>1.2053</td>
</tr>
<tr>
<td>UBM samples</td>
<td>1.2409</td>
<td>1.2055</td>
<td>–</td>
<td>1.1802</td>
</tr>
<tr>
<td>MCM samples</td>
<td>1.2388</td>
<td>1.2061</td>
<td>1.2031</td>
<td>–</td>
</tr>
</tbody>
</table>