IR homework 2 answers

Unbiased Learning-to-rank

2a According to their experimental results, How successful is IPS in addressing bias in click data? In the presence of high degrees of bias, how the performance of their model could be improved?

0.0 / 15.0 points

- [ ] **+8** points
  They compare a naive SVM-rank and an IPS version of SVM-rank. Using IPS weighting results in outperforming naive SVM-rank for different levels of bias. The results indicate IPS is more robust to increase of noise than naive SVM-rank.

- [ ] **+7** points
  In the presence of high degrees of bias, the performance of IPS is improved when more interaction data is used, while this is not the case for naive SVM-rank.

- [ ] **+10** points
  Partially answered the question, but not complete

- [x] **0** points
  None of the above

Unbiased Learning-to-rank

2b One of the implicit biases that are ignored as a result of their IPS formulation is the bias caused by implicitly treating non-clicked items as not relevant. Discuss when this implicit bias is problematic?

0.0 / 25.0 points

- [ ] **+25** points
  The problem occurs when non-relevant items are included in the loss function. For example, in the pointwise loss, non-relevant items also constitute the loss. Any other scenario where training makes use of non-relevant items as well as the relevant ones is accepted as a valid answer.
- [ ] **+14** points
  Mentioned and analyzed possible reasons for bias, but does not mention the real problem is the loss and training with this kind of data.
- [ ] **+5** points
  The answer is related to the question but does not really answer it.
- [x] **0** points
  None of the above

Unbiased Learning-to-rank

2c Propose a simple method to correct for the implicit bias of non-clicks.

0.0 / 20.0 points Bonus

+20 points
Answer containing some form of propensity with some reasonable arguments is accepted. For example, non-relevant scores can be estimated by (1-c)/(1-p). This estimation may not unbiased.

+12 points
Reasonable to some extent but not entirely accurate, or the solution will lead to a bad user experience or is not simple to implement practically.

+8 points
The answer is related to the question but does not really answer it.

0 points
None of the above

LTR with IPS

3a Explain the LTR loss function in Thorsten et al. [1] that can be unbiased using the IPS formula and discuss what is the property of that loss function that allows for IPS correction.

0.0 / 15.0 points

+10 points

They use the sum of the ranks of the relevant results as loss:
$\Delta\left(\mathbf{y} \mid \mathbf{x}_i, r_i\right) = \sum_{y \in \mathbf{y}} \operatorname{rank}(y \mid \mathbf{y}) \cdot r_i(y)$

+5 points

The property of that loss that allows for IPS is that it is linearly decomposable, meaning that each document contributes to the score individually, i.e. there is no "joint" contribution to the loss from multiple documents. This is necessary for the proof of unbiasedness described in the paper.

+4 points

The answer is sort of related to the question, but isn't actually answering it.

0 points

None of the above

LTR with IPS

3b Try to provide an IPS corrected formula for each of the three LTR loss functions that you have seen and implemented in the computer assignment. If a loss function cannot be adapted in the IPS formula, discuss the possible reasons.

0.0 / 40.0 points

+10 points

Pointwise fully correct:
Either $(c/q - y)^2$ instead of $(r - y)^2$, or $(1/p - y)^2$ only for clicked items, or indicating that it cannot be adopted because of the non-relevant scores in the loss. (all the three answers are acceptable)

+5 points

Pointwise partially correct

+15 points

Pairwise fully correct:
Unbiased pointwise loss is similar to propensity weighted SVM-rank from the paper: $C_T = \sum_{\text{i is clicked}} \sum_{\text{j is not clicked}} \frac{C_{ij}}{q_i q_j}$ where q_i and q_j are propensities of respective documents. Another correct alternative is $\sum_{i,j \in \mathcal{P}} \frac{C_{ij}}{q_i q_j}$ where $\mathcal{P}$ is the set of documents considered for a given query.

+7 points

Pairwise partially correct

+15 points

Extensions to IPS

4a The IPS in Thorsten et al. [1] works with binary clicks. How can it be extended to the graded user feedback, e.g., a 0 to 5 rating scenario in a recommendation.

0.0 / 20.0 points

☐ **+20** points

Graded rating can be treated just the same as binary clicks in IPS. They also can first be mapped into relevance probabilities, and then used with IPS. Another acceptable answer is: we can define different propensities for different rating levels, e.g have 5 propensity values for each position in a 0 to 5 rating scenario.

■ **0** points

None of the above

Extensions to IPS

4b One of the issues with IPS is its high variance. Explain the issue and discuss what can be done to reduce the variance of IPS.

0.0 / 20.0 points

+20 points

Small values for propensities, e.g. for lower-ranked positions, magnify the variance of the clicks, causing IPS to have a high variance for small datasets. The simplest workaround is to use propensity clipping, e.g. max(0.01, p_i). (Answers such as using doubly robust approaches are also acceptable.)

0 points

None of the above

Interleaving

5 Please discuss how these approaches differ from each other:

0.0 / 30.0 points

+30 points

In each of these methods, a single list of documents is created and presented to the user. Assume that we have n rankers, then there would be n list of documents for the user's given query. In TDI, one of the rankers is initially randomly chosen to start the process by putting the highest preferred document w.r.t it's ranking in the list. Then, the next ranker will select another available document based on its ordered priority. This process continues as the rankers take turns. A mapping showing which document in the list belongs to which ranking is recorded. Next, the ranked list of documents is presented to the user that might interact with some (or none) of them. The ranking that attracts more clicks to its selected document is preferred. In order to have a complete comparison, this process should be repeated $n(n-1)$ times and the ranking who wins the most is chosen as the better ranking. Conversely, in TDM, rankings are selected randomly one after each other and their list of documents is presented to the user, and the best ranker is selected by comparing the number of clicks over these rankings. Accordingly, this process works with only a single query (in practice this is done with a sufficient number of queries), but if there are more rankings than the number of slots in the interleaved list, some of the rankings will not be shown to the user.

Multileave Gradient Descent (MGD)

6 How does MGD perform w.r.t ranking performance and convergence w.r.t DBGD and why these improvements occur?

0.0 / 20.0 points

+20 points

Both of these approaches are based on using online evaluation solutions, though the difference lies in the degree they explore for better rankers. In other words, DBGD performs interleaved comparisons to infer pairwise comparisons between two rankers while MGD extends these comparisons to a set of n rankers and infers preferences among them.

The experimental results indicate that MGD outperforms DBGD in terms of ranking performance especially when there is noisy feedback (that is more in line with practical cases), though this improvement is not significant in perfect feedback scenarios. In terms of convergence, it is observed that although both of these approaches converge to the same points, MGD reaches optimal point faster than DBGD. In addition, an increasing number of ranking candidates helps with better convergence, especially in noisy feedback scenarios.

PDGD and MGD

7 What are the two major advantages of PDGD over MGD?

0.0 / 30.0 points

+15 points

1) PDGD is able to vary exploration per query and is able to have high confidence in its ranking per query, while MGD is incapable of this and the level of its exploration is only affected by the norm of its linear model.

+15 points

2) PDGD can be employed for any differentiable scoring function while MGD is not suitable for non-linear functions.

0 points

None of the above

Counterfactuals and online learning

8 How do counterfactual and online learning to rank approaches perform compare to each other w.r.t ranking performance and user experience?

0.0 / 30.0 points

+30 points

Performance: Comparing these models is heavily dependent on the level of feedback noise and bias in the data. If high-level biases such as selection or position bias exist OLTR approaches outperform CLTR approaches.

Though CLTR approaches should be preferred if the level of position bias and noise is not significant and selection bias is not present.
User experience: OLTR approaches can damage user experience in the beginning severely, though they outperform CLTR methods in the long term. The decision on choosing between them is also dependent on the application.

0 points

None of the above

Counterfactual LTR

9a Counterfactual LTR assumes a relation between a click on a document and the document's relevance. What is this assumption?

0.0 / 20.0 points

+20 points

A document is more likely to be clicked if it is more relevant.

0 points

None of the above

Counterfactual LTR

9b Give two situations where this assumption does not hold.

Model answer

0.0 / 15.0 points

+7.5 points

Over/under estimation of relevance by the user, e.g., click-bait websites have titles that make people overestimate the relevance of the site, thus there may be many clicks despite little relevance.

+7.5 points

No need to click the document, e.g. just the title or the snippet of a website can answer a user's need, thus no longer requiring them to click on the link to be helpful.

+7.5 points

Dependencies on other documents in the ranking: if a previous result answers the user's need, there may be no need to continue looking at other documents.

+7.5 points

Misclicks or similar 'mistakes' by the user.

+7.5 points

'Trust' in the search engine where a user clicks the first document in the ranking.