Reusing Historical Interaction Data for Faster Online Learning to Rank for IR (Abstract)

Katja Hofmann
k.hofmann@uva.nl

Anne Schuth
a.g.schuth@uva.nl

Shimon Whiteson
s.a.whiteson@uva.nl

Maarten de Rijke
derijke@uva.nl

ISLA, University of Amsterdam

ABSTRACT

We summarize the findings from Hofmann et al. [6]. Online learning to rank for information retrieval (IR) holds promise for allowing the development of "self-learning" search engines that can automatically adjust to their users. With the large amount of e.g., click data that can be collected in web search settings, such techniques could enable highly scalable ranking optimization. However, feedback obtained from user interactions is noisy, and developing approaches that can learn from this feedback quickly and reliably is a major challenge. In this paper we investigate whether and how previously collected (historical) interaction data can be used to speed up learning in online learning to rank for IR. We devise the first two methods that can utilize historical data (1) to make feedback available during learning more reliable and (2) to preselect candidate ranking functions to be evaluated in interactions with users of the retrieval system. We evaluate both approaches on 9 learning to rank data sets and find that historical data can speed up learning, leading to substantially and significantly higher online performance. In particular, our pre-selection method proves highly effective at compensating for noise in user feedback. Our results show that historical data can be used to make online learning to rank for IR much more effective than previously possible, especially when feedback is noisy.

1. INTRODUCTION

In recent years, learning to rank methods have become popular in information retrieval (IR) as a means of tuning retrieval systems. However, most current approaches work *offline*, meaning that manually annotated data needs to be collected beforehand, and that, once deployed, the system cannot continue to adjust to user needs, unless it is retrained with additional data. An alternative setting is *online* learning to rank, where the system learns directly from interactions with its users. These approaches are typically based on reinforcement learning techniques, meaning that the system tries out new ranking functions (also called rankers), and learns from feedback inferred from users' interactions with the presented rankings. In contrast to offline learning to rank approaches, online approaches do not require any initial training material, but rather automatically improve rankers while they are being used.

A main challenge that online learning to rank for IR approaches have to address is to learn as quickly as possible from the limited quality and quantity of feedback that can be inferred from user interactions. In this paper we address this challenge by proposing the first two online learning to rank algorithms that can reuse previously collected (historical) interaction data to make online learning more reliable and faster.

DIR 2013, April 26, 2013, Delft, The Netherlands.

2. METHOD

We model online learning to rank for IR as a cycle of interactions between users and retrieval system. Users submit queries to which the system responds with ranked result lists. The user interacts with the result lists, and these interactions allow the search engine to update its ranking model to improve performance over time. We address the problem of learning a ranking function that generalizes over queries and documents, and assume that queries are independent of each other, and of previously presented results.

Learning in this setting is implemented as stochastic gradient descent to learn a weight vector $\mathbf{w}$ for a linear combination of ranking features. Ranking features $\mathbf{X}$ encode the relationship between a query and the documents in a document collection (e.g., tf-idf, PageRank, etc.). Given a weight vector $\mathbf{w}$, and ranking features $\mathbf{X}$ candidate documents are scored using $\mathbf{s} = \mathbf{wX}$. Sorting the documents by these scores results in a result list for the given $\mathbf{w}$. Our baseline method learns weight vectors using the *dueling bandit gradient descent* (DBGD, [8]) algorithm. This algorithm maintains a current best weight vector, and learns by generating candidate weight vectors that are compared to the current best. When a candidate is found to improve over the current best weight vector, the weights are updated.

User feedback is interpreted using interleaved comparison methods [7]. These methods can infer unbiased relative feedback about ranker quality from implicit feedback, such as user clicks. In particular, they combine the result lists produced by the two rankers into one result ranking, which is then shown to the user. Clicks on the documents contributed by each ranker can then be interpreted as votes for that ranker. Our baseline interleaved comparison methods are Balanced Interleave (BI) and Team Draft (TD) [7]. Our extensions for reusing historical data are enabled by Probabilistic Interleave (PI) [4].

Based on DBGD and PI, we can now define our two approaches for reusing historical data to speed up online learning to rank.

Reliable Historical Comparison (RHC). RHC is based on the intuition that repeating comparisons on historical data should provide additional information to complement live comparisons, which can make estimates of relative performance more reliable. This is expected to reduce noise and lead to faster learning. Reusing historical interaction data for additional comparisons is possible using PI, but estimates may be biased. To remove bias, we use importance sampling as proposed in [5]. We combine the resulting historical estimates with the original live estimate using the Graybill-Deal estimator [2]. This combined estimator weights the two estimates by the ratio of their variances.

Candidate Pre-Selection (CPS). Our second approach for reusing historical data to speed up online learning to rank for IR uses historical data to improve candidate generation. Instead of randomly

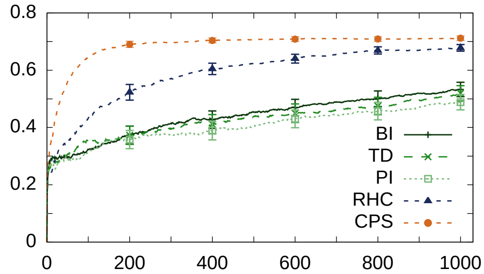

Figure 1: Offline performance in NDCG (vertical axis, computed on held-out test queries after each learning step) on *NP-2003* data set, for the *informational* click model over 1K queries.

generating a candidate ranker to test in each comparison, it generates a pool of candidate rankers, and selects the most promising one using historical data. We hypothesize that historical data can be used to identify promising rankers, and that the increased quality of candidate rankers can speed up learning.

3. EXPERIMENTS AND RESULTS

Our experiments are designed to investigate whether online learning to rank for IR can be sped up by using historical data. They are based on an existing simulation framework, which combines fully annotated learning to rank data sets with probabilistic user models to simulate user interactions with a search engine that learns online.

We conduct our experiments on the 9 data sets provided as LETOR 3.0 and 4.0. These data sets implement retrieval tasks that range from navigational (e.g., home page finding) to informational (e.g., literature search). They range in size from 50 to 1700 queries, 45 to 64 features, and up to 1000 judged documents per topic. Starting a data set, we simulate user queries by uniform sampling from the provided queries. After the retrieval system returns a ranked result list, user feedback is generated using the Dependent Click Model (DCM) [3], an extension of the Cascade Model [1] that has been shown to be effective in explaining users' click behavior in web search. We instantiate the user model with three levels of noise. The *perfect* click model provides reliable feedback. The *navigational* and *informational* model reflect two types of search tasks. Our experiments compare and contrast three baseline runs (BI: *balanced interleave*, TD: *team draft*, and PI: *probabilistic interleave*) and our proposed methods for reusing historical interaction data, RHC and CPS. Over all data sets, we find that the performance of the baseline methods substantially degrades with noise as expected. Comparing the performance of these baseline methods to that of RHC and CPS answers our research question of whether reusing historical interaction data can compensate for this noise.

Performance for our method RHC confirms our hypothesis. Under perfect user feedback, the method's performance is equivalent to that of the baseline methods that use live data only. However, its relative performance improves with increased noise. Under the informational click model, the method performs significantly better than the best baseline method on five of the nine data sets. Performance is still equivalent on two data sets, and decreases on the remaining two. Best performance under all click models is achieved by our CPS method. While we expected performance improvements under noisy click feedback, this method achieves significant improvements over the baseline methods even when click feedback is perfect. We attribute this improvement to the more exhaustive local exploration enabled by this approach. Performance improvements are highest under noisy feedback. An example is shown in Figure 1. This graph shows the offline performance in terms of NDCG on the held-out test folds for the data set NP2003 over the number of iterations (queries). We see that the baseline methods BI, TD, and PI learn slowly as the amount of available feedback increases. RHC, learning is significantly and substantially faster, because complementing comparisons with historical data makes feedback for learning more reliable. Finally, CPS is able to compensate for most of the noise in user feedback, leading to significantly faster learning.

4. CONCLUSION

In this paper, we investigated whether and how historical data can be reused to speed up online learning to rank for IR. We proposed the first two online learning to rank approaches that can reuse historical interaction data. RHC uses historical interaction data to make feedback inferred from user interactions more reliable. CPS uses this data to preselect candidate rankers so that the quality of the rankers compared in live interactions is improved.

We found that both proposed methods can improve the reliability of online learning to rank for IR under noisy user feedback. Best performance was observed using the CPS method, which can outperform all other methods significantly and substantially under all levels of noise. Performance gains of CPS were particularly high when click feedback was noisy. This result demonstrates that CPS is effective in compensating for noise in click feedback.

This work is the first to show that historical data can be used to significantly and substantially improve online performance in online learning to rank for IR. These methods are expected to make online learning with noisy feedback more reliable and therefore more widely applicable.

Acknowledgements. This research was partially supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP and BILAND projects funded by the CLARIN-nl program, the Dutch national program COMMIT, by the ESF Research Network Program ELIAS, and the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences.

References

[1] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM '08*, pages 87–94, 2008.

[2] F. Graybill and R. Deal. Combining unbiased estimators. *Biometrics*, 15(4):543–550, 1959.

[3] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *WSDM '09*, pages 124–131, 2009.

[4] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM '11*, pages 249–258, 2011.

[5] K. Hofmann, S. Whiteson, and M. de Rijke. Estimating interleaved comparison outcomes from historical click data. In *CIKM '12*, 2012.

[6] K. Hofmann, A. Schuth, S. Whiteson, and M. de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In *WSDM '13*, pages 183–192, 2013.

[7] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM '08*, 2008.

[8] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML'09*, pages 1201–1208, 2009.