Unbiased Learning-to-Rank with Biased Feedback*

Thorsten Joachims[1], **Adith Swaminathan**[2], **Tobias Schnabel**[1]
[1] Cornell University, Ithaca, NY
[2] Microsoft Research, Redmond, WA
tj@cs.cornell.edu, adswamin@microsoft.com, tbs49@cornell.edu

Abstract

Implicit feedback (e.g., clicks, dwell times, etc.) is an abundant source of data in human-interactive systems. While implicit feedback has many advantages (e.g., it is inexpensive to collect, user-centric, and timely), its inherent biases are a key obstacle to its effective use. For example, position bias in search rankings strongly influences how many clicks a result receives, so that directly using click data as a training signal in Learning-to-Rank (LTR) methods yields sub-optimal results. To overcome this bias problem, we present a counterfactual inference framework that provides the theoretical basis for unbiased LTR via Empirical Risk Minimization despite biased data. Using this framework, we derive a propensity-weighted ranking SVM for discriminative learning from implicit feedback, where click models take the role of the propensity estimator. Beyond the theoretical support, we show empirically that the proposed learning method is highly effective in dealing with biases, that it is robust to noise and propensity model mis-specification, and that it scales efficiently. We also demonstrate the real-world applicability of our approach on an operational search engine, where it substantially improves retrieval performance.

1 Introduction

Batch training of retrieval systems requires annotated test collections that take substantial effort and cost to amass. While economically feasible for web search, eliciting relevance annotations from experts is infeasible or impossible for most other ranking applications (e.g., personal collection search, intranet search). For these applications, implicit feedback from user behavior is an attractive source of data. Unfortunately, existing approaches for Learning-to-Rank (LTR) from implicit feedback – and clicks on search results in particular – have several limitations or drawbacks.

First, the naïve approach of treating a click/no-click as a positive/negative relevance judgment is severely biased. In particular, the order of presentation has a strong influence on where users click [Joachims *et al.*, 2007]. This presentation bias leads to an incomplete and skewed sample of relevance judgments that is far from uniform, thus leading to biased learning-to-rank.

Second, treating clicks as preferences between clicked and skipped documents has been found to be accurate [Joachims, 2002; Joachims *et al.*, 2007], but it can only infer preferences that oppose the presented order. This again leads to severely biased data, and learning algorithms trained with these preferences tend to reverse the presented order unless additional heuristics are used [Joachims, 2002].

Third, probabilistic click models (see Chuklin *et al.* [2015]) have been used to model how users produce clicks. By estimating latent parameters of these generative click models, one can infer the relevance of a given document for a given query. However, inferring reliable relevance judgments typically requires that the same query is seen multiple times, which is unrealistic in many retrieval settings (e.g., personal collection search) and for tail queries.

Fourth, allowing the LTR algorithm to randomize what is presented to the user, like in online learning algorithms [Raman *et al.*, 2013; Hofmann *et al.*, 2013] and batch learning from bandit feedback (BLBF) [Swaminathan and Joachims, 2015] can overcome the problem of bias in click data in a principled manner. However, requiring that rankings be actively perturbed during system operation whenever we collect training data decreases ranking quality and, therefore, incurs a cost compared to observational data collection.

In this paper we present a theoretically principled and empirically effective approach for learning from observational implicit feedback that can overcome the limitations outlined above. By drawing on counterfactual estimation techniques from causal inference [Imbens and Rubin, 2015] and work on correcting sampling bias at the query level [Wang *et al.*, 2016], we first develop a provably unbiased estimator for evaluating ranking performance using biased feedback data. Based on this estimator, we propose a Propensity-Weighted Empirical Risk Minimization (ERM) approach to LTR, which we implement efficiently in a new learning method we call *Propensity SVM-Rank*. While our approach uses a click model, the click model is merely used to assign propensities to clicked results in hindsight, not to extract aggregate relevance judgments. This means that our Propensity SVM-Rank

*Invited IJCAI18 summary based on WSDM17 Best-Paper Award.

does not require queries to repeat, making it applicable to a large range of ranking scenarios. Finally, our methods can use observational data and we do not require that the system randomizes rankings during data collection, except for a small pilot experiment to estimate the propensity model.

2 Full-Info Learning to Rank

Before we derive our approach for LTR from biased implicit feedback, we first review the conventional problem of LTR from editorial judgments. In conventional LTR, we are given a sample $\boldsymbol{X}$ of i.i.d. queries $\boldsymbol{x}_i \sim \mathrm{P}(\boldsymbol{x})$ for which we assume the relevances $\mathrm{rel}(\boldsymbol{x}, y)$ of all documents y are known. Since all relevances are assumed to be known, we call this the Full-Information Setting. The relevances can be used to compute the *loss* $\Delta(\boldsymbol{y}|\boldsymbol{x})$ (e.g., negative DCG) of any ranking $\boldsymbol{y}$ for query $\boldsymbol{x}$. Aggregating the losses of individual rankings by taking the expectation over the query distribution, we can define the overall *risk* of a ranking system S that returns rankings $S(\boldsymbol{x})$ as

$$R(S) \quad = \quad \int \Delta(S(\boldsymbol{x})|\boldsymbol{x}) \, d\,\mathrm{P}(\boldsymbol{x}). \tag{1}$$

The goal of learning is to find a ranking function $S \in \mathcal{S}$ that minimizes $R(S)$ for the query distribution $\mathrm{P}(\boldsymbol{x})$. Since $R(S)$ cannot be computed directly, it is typically estimated via the *empirical risk*

$$\hat{R}(S) \quad = \quad \frac{1}{|\boldsymbol{X}|} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \Delta(S(\boldsymbol{x}_i)|\boldsymbol{x}_i).$$

A common learning strategy is *Empirical Risk Minimization (ERM)* [Vapnik, 1998], which corresponds to picking the system $\hat{S} \in \mathcal{S}$ that optimizes the empirical risk

$$\hat{S} \quad = \quad \mathrm{argmin}_{S \in \mathcal{S}} \left\{ \hat{R}(S) \right\},$$

possibly subject to some regularization in order to control overfitting. There are several LTR algorithms that follow this approach (see Liu [2009]), and we use SVM-Rank [Joachims, 2002] as a representative algorithm in this paper.

The relevances $\mathrm{rel}(\boldsymbol{x}, y)$ are typically elicited via expert judgments. Apart from being expensive and often infeasible (e.g., in personal collection search), expert judgments come with at least two other limitations. First, it is clearly impossible to get explicit judgments for all documents, and pooling techniques [Sparck-Jones and van Rijsbergen, 1975] often introduce bias. The second limitation is that expert judgments $\mathrm{rel}(\boldsymbol{x}, y)$ have to be aggregated over all intents that underlie the same query string, and it can be challenging for a judge to properly conjecture the distribution of query intents to assign an appropriate $\mathrm{rel}(\boldsymbol{x}, y)$.

3 Partial-Info Learning to Rank

Learning from implicit feedback has the potential to overcome the above-mentioned limitations of full-information LTR. By drawing the training signal directly from the user, it naturally reflects the user's intent, since each user acts upon their own relevance judgement subject to their specific context and information need. It is therefore more appropriate to talk about query instances $\boldsymbol{x}_i$ that include contextual information about the user, instead of query strings $\boldsymbol{x}$. For a given query instance $\boldsymbol{x}_i$, we denote with $\mathrm{r}_i(y)$ the user-specific relevance of result y for query instance $\boldsymbol{x}_i$. One may argue that what expert assessors try to capture with $\mathrm{rel}(\boldsymbol{x}, y)$ is the mean of the relevances $\mathrm{r}_i(y)$ over all query instances that share the query string. Relying on implicit feedback instead for learning allows us to remove a lot of guesswork about what the distribution of users meant by a query.

However, when using implicit feedback as a relevance signal, unobserved feedback is an even greater problem than missing judgments in the pooling setting. In particular, implicit feedback is distorted by presentation bias, and it is not missing completely at random [Little and Rubin, 2002]. To nevertheless derive well-founded learning algorithms, we adopt the following counterfactual model. It closely follows [Schnabel *et al.*, 2016], which unifies several prior works on evaluating information retrieval systems.

For concreteness and simplicity, assume that relevances are binary, $\mathrm{r}_i(y) \in \{0, 1\}$, and our performance measure of interest is the sum of the ranks of the relevant results

$$\Delta(\boldsymbol{y}|\boldsymbol{x}_i, \mathrm{r}_i) \quad = \quad \sum_{y \in \boldsymbol{y}} \mathrm{rank}(y|\boldsymbol{y}) \cdot \mathrm{r}_i(y). \tag{2}$$

Analogous to (1), we can define the risk of a system as

$$R(S) \quad = \quad \int \Delta(S(\boldsymbol{x})|\boldsymbol{x}, \mathrm{r}) \, d\,\mathrm{P}(\boldsymbol{x}, \mathrm{r}). \tag{3}$$

In our counterfactual model, there exists a true vector of relevances r_i for each incoming query instance $(\boldsymbol{x}_i, \mathrm{r}_i) \sim \mathrm{P}(\boldsymbol{x}, \mathrm{r})$. However, only a part of these relevances is observed for each query instance, while typically most remain unobserved. In particular, given a presented ranking $\bar{\boldsymbol{y}}_i$ we are more likely to observe the relevance signals (e.g., clicks) for the top-ranked results than for results ranked lower in the list. Let o_i denote the 0/1 vector indicating which relevance values were revealed, $o_i \sim \mathrm{P}(o|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$. For each element of o_i, denote with $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$ the marginal probability of observing the relevance $\mathrm{r}_i(y)$ of result y for query $\boldsymbol{x}_i$, if the user was presented the ranking $\bar{\boldsymbol{y}}_i$. We refer to this probability value as the *propensity* of the observation. We discuss how o_i and Q can be obtained more in Section 4.

Using this counterfactual modeling setup, we can get an unbiased estimate of $\Delta(\boldsymbol{y}|\boldsymbol{x}_i, \mathrm{r}_i)$ for any new ranking $\boldsymbol{y}$ (typically different from the presented ranking $\bar{\boldsymbol{y}}_i$) via the inverse propensity scoring (IPS) estimator [Horvitz and Thompson, 1952; Rosenbaum and Rubin, 1983; Imbens and Rubin, 2015]

$$\begin{aligned}\hat{\Delta}_{IPS}(\boldsymbol{y}|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, o_i) \; &= \sum_{y:o_i(y)=1} \frac{\mathrm{rank}(y|\boldsymbol{y}) \cdot \mathrm{r}_i(y)}{Q(o_i(y)=1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)} \\ &= \sum_{\substack{y:o_i(y)=1 \\ \bigwedge \mathrm{r}_i(y)=1}} \frac{\mathrm{rank}(y|\boldsymbol{y})}{Q(o_i(y)=1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)}.\end{aligned}$$

This is an unbiased estimate of $\Delta(\boldsymbol{y}|\boldsymbol{x}_i, \mathrm{r}_i)$ for any $\boldsymbol{y}$, if $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i) > 0$ for all y that are relevant $\mathrm{r}_i(y) = 1$ (but not necessarily for the irrelevant y). The proof for this is quite straightforward and can be found in the original paper [Joachims *et al.*, 2017].

An interesting property of $\hat{\Delta}_{IPS}(\boldsymbol{y}|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, o_i)$ is that only those results y with $[o_i(y) = 1 \wedge \mathrm{r}_i(y) = 1]$ (i.e. clicked results, as we will see later) contribute to the estimate. We therefore only need the propensities $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$ for relevant results. Since we will eventually need to estimate the propensities $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$, an additional requirement for making $\hat{\Delta}_{IPS}(\boldsymbol{y}|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, o_i)$ computable while remaining unbiased is that the propensities only depend on observable information (i.e., unconfoundedness, see Imbens and Rubin [2015]).

Having a sample of N query instances $\boldsymbol{x}_i$, recording the partially-revealed relevances r_i as indicated by o_i, and the propensities $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$, the empirical risk of a system is simply the IPS estimates averaged over query instances:

$$\hat{R}_{IPS}(S) = \frac{1}{N}\sum_{i=1}^{N} \sum_{\substack{y:o_i(y)=1 \\ \bigwedge \mathrm{r}_i(y)=1}} \frac{\mathrm{rank}(y|S(\boldsymbol{x}_i))}{Q(o_i(y)=1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)}. \quad (4)$$

Since $\hat{\Delta}_{IPS}(\boldsymbol{y}|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, o_i)$ is unbiased for each query instance, the aggregate $\hat{R}_{IPS}(S)$ is also unbiased for $R(S)$ from (3),

$$\mathbb{E}[\hat{R}_{IPS}(S)] \quad = \quad R(S).$$

Furthermore, it is easy to verify that $\hat{R}_{IPS}(S)$ converges to the true $R(S)$ under mild additional conditions (i.e., propensities bounded away from 0) as we increase the sample size N of query instances. So, we can perform ERM using this propensity-weighted empirical risk,

$$\hat{S} \quad = \quad \mathrm{argmin}_{S\in\mathcal{S}} \left\{ \hat{R}_{IPS}(S) \right\}.$$

Finally, using standard results from statistical learning theory [Vapnik, 1998], consistency of the empirical risk paired with capacity control implies consistency also for ERM. In intuitive terms, this means that given enough training data, the learning algorithm is guaranteed to find the best system in $\mathcal{S}$.

3.1 Position-Based Propensity Model

The previous section showed that the propensities of the observations $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$ are the key component for unbiased LTR from biased observational feedback. To derive propensities of observed clicks, we consider a straightforward examination model analogous to Richardson *et al.* [2007], where a click on a search result depends on the probability that a user examines a result (i.e., $\mathrm{e}_i(y)$) and then decides to click on it (i.e., $\mathrm{c}_i(y)$) in the following way:

$$P(\mathrm{e}_i(y) = 1|\,\mathrm{rank}(y|\bar{\boldsymbol{y}})) \cdot P(\mathrm{c}_i(y) = 1|\,\mathrm{r}_i(y), \mathrm{e}_i(y) = 1).$$

In this model, examination depends only on the rank of y in $\bar{\boldsymbol{y}}$. So, $P(\mathrm{e}_i(y) = 1|\,\mathrm{rank}(y|\bar{\boldsymbol{y}}_i))$ can be represented by a vector of examination probabilities p_r, one for each rank r, which are precisely the propensities $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$. These examination probabilities can model presentation bias found in eye-tracking studies [Joachims *et al.*, 2007], where users are more likely to see results at the top of the ranking than those further down.

Under this propensity model, we can simplify the IPS estimator from (4) by substituting p_r as the propensities and by using $\mathrm{c}_i(y) = 1 \leftrightarrow [o_i(y) = 1 \ \wedge \ \mathrm{r}_i(y) = 1]$

$$\hat{R}_{IPS}(S) = \frac{1}{n}\sum_{i=1}^{n} \sum_{y:\mathrm{c}_i(y)=1} \frac{\mathrm{rank}(y|S(\boldsymbol{x}_i))}{p_{\mathrm{rank}(y|\bar{\boldsymbol{y}}_i)}}. \quad (5)$$

$\hat{R}_{IPS}(S)$ is an unbiased estimate of $R(S)$ under the position-based propensity model if $p_r > 0$ for all ranks. While absence of a click does not imply that the result is not relevant (i.e., $\mathrm{c}_i(y) = 0 \not\rightarrow \mathrm{r}_i(y) = 0$), the IPS estimator has the nice property that such explicit negative judgments are not needed to compute an unbiased estimate of $R(S)$ for the loss in (2). Similarly, while absence of a click leaves us unsure about whether the result was examined (i.e., $\mathrm{e}_i(y) = ?$), the IPS estimator only needs to know the indicators $o_i(y) = 1$ for results that are also relevant (i.e., clicked results).

3.2 Propensity SVM-Rank

We now derive a concrete learning method that implements propensity-weighted LTR. It is based on SVM-Rank [Joachims, 2002; Joachims, 2006], but we conjecture that propensity-weighted versions of other LTR methods can be derived as well.

Consider a dataset of n examples of the following form. For each query-result pair $(\boldsymbol{x}_j, y_j)$ that is clicked, we compute the propensity $q_i = Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$ of the click according to our click propensity model. We also record the candidate set Y_j of all results for query $\boldsymbol{x}_j$. Typically, Y_j contains a few hundred documents – selected by a stage-one ranker [Wang *et al.*, 2011] – that we aim to rerank. Note that each click generates a separate training example, even if multiple clicks occur for the same query.

Given this propensity-scored click data, we define Propensity SVM-Rank as a generalization of conventional SVM-Rank. Propensity SVM-Rank learns a linear scoring function $f(\boldsymbol{x}, y) = w \cdot \phi(\boldsymbol{x}, y)$ that can be used for ranking results, where w is a weight vector and $\phi(\boldsymbol{x}, y)$ is a feature vector that describes the match between query $\boldsymbol{x}$ and result y.

Propensity SVM-Rank optimizes the following objective,

$$\begin{aligned}
\hat{w} \ = \ & \mathrm{argmin}_{w,\xi} \ \frac{1}{2} w \cdot w + \frac{C}{n}\sum_{j=1}^{n} \frac{1}{q_j} \sum_{y\in Y_j} \xi_{jy} \\
s.t. \quad & \forall y \in Y_1 \setminus \{y_1\} : w \cdot [\phi(\boldsymbol{x}_1, y_1) - \phi(\boldsymbol{x}_1, y)] \geq 1 - \xi_{1y} \\
& \qquad \vdots \\
& \forall y \in Y_n \setminus \{y_n\} : w \cdot [\phi(\boldsymbol{x}_n, y_n) - \phi(\boldsymbol{x}_n, y)] \geq 1 - \xi_{ny} \\
& \forall j \forall y : \xi_{jy} \geq 0.
\end{aligned}$$

C is a regularization parameter that is typically selected via cross-validation. The training objective optimizes an upper bound on the regularized IPS estimated empirical risk of (5), since each line of constraints corresponds to the rank of a relevant document (minus 1).

In particular, for any feasible (w, ξ)

$$\begin{aligned}\text{rank}(y_i|\boldsymbol{y})-1 &= \sum_{y\neq y_i} \mathbb{1}_{w\cdot[\phi(\boldsymbol{x}_i,y)-\phi(\boldsymbol{x}_i,y_i)]>0} \\ &\leq \sum_{y\neq y_i} \max(1-w\cdot[\phi(\boldsymbol{x}_i,y_i)-\phi(\boldsymbol{x}_i,y)],0) \\ &\leq \sum_{y\neq y_i} \xi_{iy}.\end{aligned}$$

We can solve this type of Quadratic Program efficiently via a one-slack formulation [Joachims, 2006], and we are using SVM-Rank with appropriate modifications to include IPS weights $1/q_j$. The code is available online[1].

In the empirical evaluation, we compare against the naive application of SVM-Rank, which minimizes the rank of the clicked documents while ignoring presentation bias. In particular, Naive SVM-Rank sets all the q_i uniformly to the same constant (e.g., 1).

4 Empirical Evaluation

The original paper [Joachims *et al.*, 2017] takes a two-pronged approach to evaluation. First, it uses synthetically generated click data to explore the behavior of our methods over the whole spectrum of presentation bias severity, click noise, and propensity mis-specification. Due to space constraints, we do not include these experiments here, but focus on a real-world experiment that evaluates our approach on an operational search engine using real click-logs from live traffic. In particular, we examine the performance of Propensity SVM-Rank when learning a new ranking function for the Arxiv Full-Text Search[2] based on real-world click logs from this system. The search engine uses a linear scoring function $f(\boldsymbol{x}, y) = w \cdot \phi(\boldsymbol{x}, y)$. The Query-document features $\phi(\boldsymbol{x}, y)$ are represented by a $1000-$dimensional vector, and the production ranker used for collecting training clicks employs a hand-crafted weight vector w (denoted Prod). Observed clicks on rankings served by this ranker over a period of 21 days provide implicit feedback data for LTR as outlined in Section 3.2.

To estimate the propensity model, we consider the simple position-based model of Section 3.1 and we collect new click data via randomized interventions for 7 days as detailed in [Joachims *et al.*, 2017] with landmark rank $k = 1$. In short, before presenting the ranking, we take the top-ranked document and swap it with the document at a uniformly-at-random chosen rank $j \in \{1, \dots 21\}$. The ratio of observed click-through rates (CTR) on the formerly top-ranked document now at position j vs. its CTR at position 1 gives a noisy estimate of p_j/p_1 in the position-based click model. We additionally smooth these estimates by interpolating with the overall observed CTR at position j (normalized so that $CTR@1 = 1$). This yields p_r that approximately decay with rank r, and the smallest $p_r \simeq 0.12$. For $r > 21$, we impute $p_r = p_{21}$. Since the original paper appeared, another technique for propensity estimation from observational data has been proposed that could also be used [Wang *et al.*, 2018].

[1] http://www.joachims.org/svm_light/svm_proprank.html

[2] http://search.arxiv.org:8081/

Interleaving Experiment	Propensity SVM-Rank wins	loses	ties
against Prod	87	48	83
against Naive SVM-Rank	95	60	102

Table 1: Per-query balanced interleaving results for detecting relative performance between the hand-crafted production ranker used for click data collection (Prod), Naive SVM-Rank and Propensity SVM-Rank.

We partition the click-logs into a train-validation split: the first 16 days are the train set and provide 5437 click-events, while the remaining 5 days are the validation set with 1755 click events. The hyper-parameter C is picked via cross validation. We use the IPS estimator for Propensity SVM-Rank, and the naive estimator with $Q(o(y) = 1|\boldsymbol{x}, \bar{\boldsymbol{y}}, \mathrm{r}) = 1$ for Naive SVM-Rank. With the best hyper-parameter settings, we re-train on all 21 days worth of data to derive the final weight vectors for either method.

We fielded these learnt weight vectors in two online interleaving experiments [Chapelle *et al.*, 2012], the first comparing Propensity SVM-Rank against Prod and the second comparing Propensity SVM-Rank against Naive SVM-Rank. The results are summarized in Table 1. We find that Propensity SVM-Rank significantly outperforms the hand-crafted production ranker that was used to collect the click data for training (two-tailed binomial sign test $p = 0.001$ with relative risk 0.71 compared to null hypothesis). Furthermore, Propensity SVM-Rank similarly outperforms Naive SVM-Rank, demonstrating that even a simple propensity model provides benefits on real-world data (two-tailed binomial sign test $p = 0.006$ with relative risk 0.77 compared to null hypothesis). Note that Propensity SVM-Rank not only significantly, but also substantially outperforms both other rankers in terms of effect size – and the synthetic data experiments suggest that additional training data will further increase its advantage.

5 Conclusions and Future

This paper introduced a principled approach for learning-to-rank under biased feedback data. Drawing on counterfactual modeling techniques from causal inference, we present a theoretically sound Empirical Risk Minimization framework for LTR. We instantiate this framework with a propensity-weighted ranking SVM. Real-world experiments on a live search engine show that the approach leads to substantial retrieval improvements.

Beyond the specific learning methods and propensity models we propose, this paper may have even bigger impact for its theoretical contribution of developing the general counterfactual model for LTR, thus articulating the key components necessary for LTR under biased feedback. First, the insight that propensity estimates are crucial for ERM learning opens a wide area of research on designing better propensity models. Second, the theory demonstrates that LTR methods should optimize propensity-weighted ERM objectives, raising the question of which other learning methods can be adapted. Third, we conjecture that propensity-weighted ERM

approaches can be developed also for pointwise and listwise LTR methods using techniques from Schnabel *et al.* [2016].

Beyond learning from implicit feedback, propensity-weighted ERM techniques may prove useful even for optimizing offline IR metrics on manually annotated test collections. First, they can eliminate pooling bias, since the use of sampling during judgment elicitation puts us in a controlled setting where propensities are known (and can be optimized [Schnabel *et al.*, 2016]) by design. Second, propensities estimated via click models can enable click-based IR metrics like click-DCG to better correlate with test set DCG.

This work was supported in part through NSF Awards IIS-1247637, IIS-1513692, IIS-1615706, and a gift from Bloomberg. We thank Maarten de Rijke, Alexey Borisov, Artem Grotov, and Yuning Mao for valuable feedback and discussions.

References

[Chapelle *et al.*, 2012] Oliver Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30(1):6:1–6:41, 2012.

[Chuklin *et al.*, 2015] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2015.

[Hofmann *et al.*, 2013] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In *International Conference on Web Search and Data Mining (WSDM)*, pages 183–192, 2013.

[Horvitz and Thompson, 1952] Daniel Horvitz and Donovan Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

[Imbens and Rubin, 2015] Guido Imbens and Donald Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

[Joachims *et al.*, 2007] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2), April 2007.

[Joachims *et al.*, 2017] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *ACM Conference on Web Search and Data Mining (WSDM)*, pages 781–789, 2017.

[Joachims, 2002] Thorsten Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

[Joachims, 2006] Thorsten Joachims. Training linear SVMs in linear time. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD)*, pages 217–226, 2006.

[Little and Rubin, 2002] Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. John Wiley, 2002.

[Liu, 2009] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, March 2009.

[Raman *et al.*, 2013] Karthik Raman, Thorsten Joachims, P. Shivaswamy, and Tobias Schnabel. Stable coactive learning via perturbation. In *International Conference on Machine Learning (ICML)*, pages 837–845, 2013.

[Richardson *et al.*, 2007] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *International Conference on World Wide Web (WWW)*, pages 521–530. ACM, 2007.

[Rosenbaum and Rubin, 1983] Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[Schnabel *et al.*, 2016] Tobias Schnabel, Adith Swaminathan, Peter Frazier, and Thorsten Joachims. Unbiased comparative evaluation of ranking functions. In *ACM International Conference on the Theory of Information Retrieval (ICTIR)*, pages 109–118, 2016.

[Sparck-Jones and van Rijsbergen, 1975] Karen Sparck-Jones and Cornelis J. van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical report, University of Cambridge, 1975.

[Swaminathan and Joachims, 2015] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research (JMLR)*, 16:1731–1755, Sep 2015. Special Issue in Memory of Alexey Chervonenkis.

[Vapnik, 1998] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, Chichester, GB, 1998.

[Wang *et al.*, 2011] Lidan Wang, Jimmy J. Lin, and Donald Metzler. A cascade ranking model for efficient ranked retrieval. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 105–114, 2011.

[Wang *et al.*, 2016] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Learning to rank with selection bias in personal search. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2016.

[Wang *et al.*, 2018] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. Position bias estimation for unbiased learning to rank in personal search. In *Conference on Web Search and Data Mining (WSDM)*, 2018.