Unbiased Learning-to-Rank with Biased Feedback

Thorsten Joachims
Cornell University, Ithaca, NY
tj@cs.cornell.edu

Adith Swaminathan
Cornell University, Ithaca, NY
adith@cs.cornell.edu

Tobias Schnabel
Cornell University, Ithaca, NY
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. In contrast to most conventional approaches to de-bias the data using click models, this allows training of ranking functions even in settings where queries do not repeat. 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 misspecification, 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 [11]. 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 [9, 11], 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 [9].

Third, probabilistic click models (see [4]) have been used to model how users produce clicks, and they can take position and context biases into account. 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 [16, 6] and batch learning from bandit feedback (BLBF) [24] 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 [8], 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 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.

Under review

ACM ISBN 123-4567-24-567/08/06…$15.00

DOI: 10.475/123_4

When deriving our approach, we provide theoretical justification for each step, leading to a rigorous end-to-end approach that does not make unspecified assumptions or employs heuristics. This provides a principled basis for further improving components of the approach (e.g., the click propensity model, the ranking performance measure, the learning algorithm). We present an extensive empirical evaluation testing the limits of the approach on synthetic click data, finding that it performs robustly over a large range of bias, noise, and misspecification levels. Furthermore, we field our method in a real-world application on an operational search engine, finding that it is robust in practice and manages to substantially improve retrieval performance.

2. RELATED WORK

There are two groups of approaches for handling biases in implicit feedback for learning-to-rank. The first group assumes the feedback collection step is fixed, and tries to interpret the observationally collected data so as to minimize bias effects. Approaches in the second group intervene during feedback collection, trying to present rankings that will lead to less biased feedback data overall.

Approaches in the first group commonly assume some model of user behavior in order to explain bias effects. For example, in a cascade model [5], users are assumed to sequentially go down a ranking and click on a document if it is relevant. Clicks, under this model, let us learn preferences between skipped and clicked documents. Learning from these relative preferences lowers the impact of some biases [9]. Other click models ([5, 3, 1], also see [4]) have been proposed, and are trained to maximize log-likelihood of observed clicks. In these click modeling approaches, performance on downstream learning-to-rank algorithms is merely an afterthought. In contrast, we separate click propensity estimation and learning-to-rank in a principled way and we optimize for ranking performance directly. Our framework allows us to plug-and-play more sophisticated user models in place of the simple click models we use in this work.

The key technique used by approaches in the second group to obtain more reliable click data are randomized experiments. For instance, randomizing documents across all ranks lets us learn unbiased relevances for each document, and swapping neighboring pairs of documents [15] lets us learn reliable pairwise preferences. Similarly, randomized interleaving can detect preferences between different rankers reliably [2]. Different from online learning via bandit algorithms and interleaving [29, 21], batch learning from bandit feedback (BLBF) [24] still uses randomization during feedback collection, and then performs offline learning. Our problem formulation can be interpreted as being half way between the BLBF setting (loss function is unknown and no assumptions on loss function) and learning-to-rank from editorial judgments (components of ranking are fully labeled and loss function is given) since we know the form of the loss function but labels for only some parts of the ranking are revealed. All approaches that use randomization suffer from two limitations. First, randomization typically degrades ranking quality during data collection; second, deploying non-deterministic ranking functions introduces bookkeeping overhead. In this paper, the system can be deterministic and we merely exploit and model stochasticity in user behavior. Moreover, our framework also allows (but does not require) the use of randomized data collection in order to mitigate the effect of biases and improve learning.

Our approach uses inverse propensity scoring (IPS), originally employed in causal inference from observational studies [18], and more recently also in whole page optimization [28], IR evaluation with manual judgments [19], and recommender evaluation [12, 20]. We use randomized interventions similar to [5, 23, 27] to estimate propensities in a position discount model. Unlike the uniform ranking randomization of [27] (with its high performance impact) or swapping adjacent pairs as in [5], we swap documents in different ranks to the top position randomly as in [23]. See Section 5.3 for details.

Finally, our approach is similar in spirit to [27], where propensity-weighting is used to correct for selection bias when discarding queries without clicks during learning-to-rank. The key insight of our work is to recognize that inverse propensity scoring can be employed much more powerfully, to account for position bias, trust bias, contextual effects, document popularity etc. using appropriate click models to estimate the propensity of each click rather than the propensity for a query to receive a click as in [27].

3. 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)* [25], 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 [14]), and we use SVM-Rank [9] 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, since it is clearly impossible to get explicit judgments for all documents, pooling techniques [22] are used such that only the most promising documents are judged. While cutting down on judging effort, this introduces an undesired *pooling bias* because all unjudged documents are typically assumed to be irrelevant. 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 $\mathrm{rel}(\boldsymbol{x}, y)$.

4. 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, so, using implicit feedback for learning is able 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 [13]. To nevertheless derive well-founded learning algorithms, we adopt the following counterfactual model. It closely follows [19], 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

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

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

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

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 will discuss how o_i and Q can be obtained in Section 5.

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 [7, 18, 8]

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

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).

$$\begin{aligned} &\mathbb{E}_{o_i}[\hat{\Delta}_{IPS}(\boldsymbol{y}|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, o_i)] \\ &\quad=\; \mathbb{E}_{o_i}\left[\sum_{y: o_i(y)=1} \frac{\mathrm{rank}(y|\boldsymbol{y}) \cdot \mathrm{r}_i(y)}{Q(o_i(y)=1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)}\right] \\ &\quad=\; \sum_{y \in \boldsymbol{y}} \mathbb{E}_{o_i}\left[\frac{o_i(y) \cdot \mathrm{rank}(y|\boldsymbol{y}) \cdot \mathrm{r}_i(y)}{Q(o_i(y)=1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i))}\right] \\ &\quad=\; \sum_{y \in \boldsymbol{y}} \frac{Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i) \cdot \mathrm{rank}(y|\boldsymbol{y}) \cdot \mathrm{r}_i(y)}{Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)} \\ &\quad=\; \sum_{y \in \boldsymbol{y}} \mathrm{rank}(y|\boldsymbol{y}) \, \mathrm{r}_i(y) \\ &\quad=\; \Delta(\boldsymbol{y}|\boldsymbol{x}_i, \mathrm{r}_i). \end{aligned}$$

The second step uses linearity of expectation, and the fourth step uses $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i) > 0$.

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 [8]).

To define the empirical risk to optimize during learning, we begin by collecting a sample of N query instances $\boldsymbol{x}_i$, recording the partially-revealed relevances r_i as indicated by o_i, and the propensities $Q(o_i(y) = 1|\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i, \mathrm{r}_i)$ for the observed relevant results in the ranking $\bar{\boldsymbol{y}}_i$ presented by the system. Then, the empirical risk of a system is simply the IPS estimates averaged over query instances:

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

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),

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

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,

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

Finally, using standard results from statistical learning theory [25], consistency of the empirical risk paired with capacity control implies consistency also for ERM. In intuitive