Intervention Harvesting for Context-Dependent Examination-Bias Estimation

Zhichong Fang
Tsinghua University
Beijing, China
fzc14@mails.tsinghua.edu.cn

Aman Agarwal
Cornell University
Ithaca, NY, USA
aa2398@cornell.edu

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

ABSTRACT

Accurate estimates of examination bias are crucial for unbiased learning-to-rank from implicit feedback in search engines and recommender systems, since they enable the use of Inverse Propensity Score (IPS) weighting techniques to address selection biases and missing data. Unfortunately, existing examination-bias estimators are limited to the Position-Based Model (PBM), where the examination bias may only depend on the rank of the document. To overcome this limitation, we propose a Contextual Position-Based Model (CPBM) where the examination bias may also depend on a context vector describing the query and the user. Furthermore, we propose an effective estimator for the CPBM based on intervention harvesting. A key feature of the estimator is that it does not require disruptive interventions but merely exploits natural variation resulting from the use of multiple historic ranking functions. Real-world experiments on the ArXiv search engine and semi-synthetic experiments on the Yahoo Learning-To-Rank dataset demonstrate the superior effectiveness and robustness of the new approach.

CCS CONCEPTS

• **Information systems → Learning to rank**.

KEYWORDS

examination bias; unbiased learning-to-rank; propensity estimation

ACM Reference Format:
Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention Harvesting for Context-Dependent Examination-Bias Estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331238

1 INTRODUCTION

While implicit feedback (e.g., clicks, dwell time) is an abundant and attractive source of data in most information-retrieval applications (e.g., personal search, email search, recommendation), its use for learning-to-rank (LTR) is challenging due to its biased nature. To address this bias problem, Joachims et al. [19] proposed a counterfactual inference approach, providing an unbiased LTR framework via Empirical Risk Minimization. A key requirement for the effectiveness of this approach is an accurate estimate of the examination bias, which describes how likely a user is to discover a particular result. For example, a result is less likely to be discovered at position 10 than at position 1. Estimates of the examination bias enable the use of Inverse Propensity Score (IPS) weighting techniques, which make modeling and estimating examination bias equivalent to propensity estimation for unbiased LTR.

There are two key limitations of existing propensity estimation methods for LTR [3, 19, 27]. First, existing methods are restricted to the Position-Based Model (PBM) [9], which only models how examination changes with the rank of the result. Second, existing methods treat all queries uniformly, even though the examination bias is likely to vary from query to query. For example, users may examine results in navigational queries (i.e., search queries entered with the intention of finding a particular website or webpage) differently compared to informational queries (i.e., search queries for a broad topic for which there could be thousands of relevant results). To overcome these limitations, a naive approach would be to train a separate PBM for each context – say one for navigational and one for informational queries – simply by partitioning the data. However, this is feasible only when there is a small number of discrete contexts, and it does not apply to cases where contexts are described by arbitrary feature vectors. The latter is a highly desirable use case, since it is natural to represent the context by features describing the query (e.g., query length), features describing the candidate set (e.g., size), and features describing the user (e.g., age).

In this paper, we address these limitations of the PBM and present a new Contextual Position-Based Model (CPBM) that greatly extends the expressiveness of the PBM. Instead of having a single examination parameter for each rank that is shared among all queries, we show how the CPBM can model examination dependent on arbitrary context vectors through a deep network. Furthermore, we present an AllPairs estimator [3] for learning CPBM models from log data. For training, our estimator harvests implicit interventions that are already available in most operational systems. In particular, the estimator only requires (not necessarily randomized) log data from at least two ranking functions that were deployed on the system in the past. The resulting deep network can then be used to compute context-dependent propensities for LTR algorithms like [1, 2, 19]. We evaluate the fidelity of the CPBM model and the effectiveness of the estimator in real-world experiments on the ArXiv full-text search engine and in semi-synthetic experiments on the Yahoo Learning-to-Rank Challenge dataset [7].

SIGIR '19, July 21–25, 2019, Paris, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331238

2 RELATED WORK

In most information retrieval systems, large amounts of implicit feedback are logged automatically and serve as an attractive source of training data. However, it is known that this type of data suffers from various biases due to both the system and the user, such as position bias [17], presentation bias [22] and trust bias [18].

To handle biases in a principled way, Joachims et al. [19] introduced an unbiased learning-to-rank framework, which is a consistent learning approach despite biased feedback. It relies on IPS weighting first developed in causal inference and survey sampling [15, 24]. IPS has been commonly adopted for unbiased evaluation and learning [1, 2, 12, 21, 25]. However, because the propensity in the unbiased LTR setting represents the unknown bias with which a user examines a document, this propensity needs to be estimated.

Existing propensity-estimation methods for LTR are based on the Position-Based Model (PBM) [23]. The most effective methods use randomized interventions [19, 26], which unfortunately degrade the user's search experience. To avoid such interventions, Wang et al. [27] proposed a regression-based Expectation-Maximization (EM) algorithm, and Ai et al. [4] proposed a learning algorithm that learns propensity models together with the ranking model. Unfortunately, both approaches involve learning an accurate relevance model, which is just as hard as the LTR problem itself. The approach of Agarwal et al. [3] avoids both randomized interventions and relevance modeling by exploiting click data from multiple loggers as implicit interventions. In our work, we extend their approach to the Contextual Position-Based Model (CPBM) for improved accuracy.

Beyond the PBM, many other click models for ranked search exist. However, they were designed for inferring relevance, not propensities. One example is the Cascade model [10], where users scan documents top-down until a relevant document is found. Built upon the PBM and the Cascade model, more complex models like UBM [13], DBN [8], CCM [14] and CSM [6] were proposed to infer relevance judgments from click logs. It is an open question in how far these models can be adapted for propensity estimation as well.

3 THE CONTEXTUAL POSITION-BASED MODEL

Modeling the examination bias is crucial for learning to rank from implicit feedback, since it confounds the feedback signal. We start by reviewing the Position-Based Model, as it is arguably the simplest model for correcting the examination bias in learning to rank from implicit feedback. As shown by Joachims et al. [19], the parameters of the PBM can serve as propensity estimates, enabling the use of IPS weighting for unbiased learning-to-rank.

The PBM captures that the rank of a result has a strong influence on whether a result is examined (i.e. viewed and evaluated as a prerequisite for any subsequent feedback like a click or a rating) by a user, where higher-ranked results are typically more likely to be examined than results further down the ranking. Suppose that for a particular query q, result d is displayed at position k. Let C be the random variable corresponding to a user clicking on d, and let E be the random variable denoting whether the user examines d. Then according to the Position-Based Model [9],

$$\Pr(C = 1|q, d, k) = \Pr(E = 1|k)\,\mathrm{rel}(q, d), \tag{1}$$

where $\mathrm{rel}(q, d) \in \{0, 1\}$ is the binary relevance of document d for query q.

While $\Pr(E = 1|k)$ can be used as an estimate of the examination propensity [19], it is a rather simplistic model since it assumes that examination does not vary across queries. However, it is implausible that navigational queries share the same propensity curve with informational queries, and we will validate in our experiments that such dependencies exist in real-world search engines. More broadly, we argue that examination behavior not only varies across queries, but that it varies across contexts x more generally. This context x includes the query itself and features describing the query (e.g., query length), features describing the candidate set (e.g., size), and features describing the user (e.g., age). To be able to model these dependencies, we propose a new model – called the Contextual PBM (CPBM) – where the examination propensity can depend on the observed context x in addition to the position as follows.

$$\Pr(C = 1|x, d, k) = \Pr(E = 1|k, x)\,\mathrm{rel}(x, d). \tag{2}$$

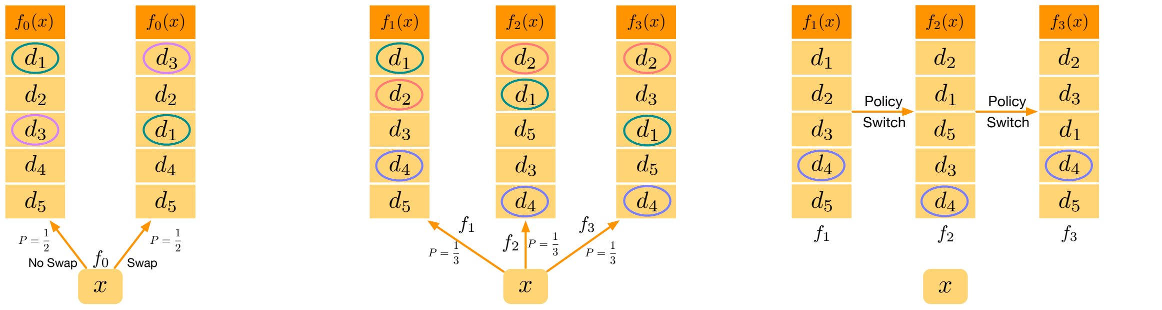
Since the context x contains all the information about its corresponding query q, we can drop the query q from our notation. Through its dependence on context x, the CPBM can represent different propensity curves $\Pr(E = 1|k, x)$ w.r.t. position k for each query context x, instead of assuming that all queries share the same examination curve $\Pr(E = 1|k)$ like in the PBM.

4 ESTIMATING CPBM MODELS

While the increased expressiveness of the CPBM is clearly desirable, it raises several challenges when estimating the model from the data. In particular, instead of just estimating k_{max} scalar parameters $\Pr(E = 1|k)$ like in the PBM, where k_{max} is the maximum length of the presented rankings (say 10 or 20), the CPBM requires estimating a context-dependent propensity model $\Pr(E = 1|k, x)$, which in the following will be represented as a neural network. Furthermore, estimating $\Pr(E = 1|k, x)$ is challenging since we typically do not observe ground truth for $\mathrm{rel}(x, d)$ such that it is difficult to attribute the lack of a feedback signal to a lack of examination or a lack of relevance. After reviewing the shortcomings of a naive generative modeling approach in the next subsection, we will exploit the fact that randomized interventions can be used to control for relevance. In particular, we will show how reusing logged click data from multiple ranking functions provides such intervention data for the CPBM under reasonable assumptions, eliminating the need for explicit interventions that affect the user experience.

4.1 Generative Modeling

The first thought one may have is to estimate a CPBM via a standard generative-modeling approach with both examination and relevance as latent variables. In fact, Wang et al. [27] have proposed such an approach for the simpler problem of estimating the parameters of the PBM. Let $\mathcal{L} = \{(x^j, d^j, k^j, c^j)|j \in [N]\}$ be a sample of N observations with one tuple for each context-document pair (x^j, d^j), indicating with k^j the position of d^j in the ranking and with $c^j \in \{0, 1\}$ whether it was clicked. Extending the approach of Wang et al. [27] to the CPBM, the conditional log likelihood



(a) Swap Intervention between positions 1 and 3.

(b) Intervention Harvesting (A/B Test).

(c) Intervention Harvesting (switch in production rankers).

Figure 1: Illustration of Swap Interventions and of Intervention Harvesting.

objective (conditioned on the observed queries and rankings) is

$$\log \Pr(\mathcal{L}) = \sum_{j \in \mathcal{L}} \Big[c^j \log \left(p_{k^j}(x^j) r(x^j, d^j) \right) + (1 - c^j) \log \left(1 - p_{k^j}(x^j) r(x^j, d^j) \right) \Big],$$

where $p_k(x) := \Pr(E = 1|k, x)$ is a context-dependent propensity model and $r(x, d) := \mathrm{rel}(x, d)$ is a document-dependent relevance model. Both relevance and examination are latent, and even for the simpler PBM model it was found that the propensity estimates can be far off [3]. A key shortcoming of this approach is that it requires learning the relevance $\mathrm{rel}(x, d)$ of all individual documents without any direct supervision, which is just as difficult as the learning-to-rank problem itself. This means that the relevance model will typically be misspecified and thus bias the propensity estimates.

4.2 Explicit Swap Interventions

To overcome the need for modeling the unobserved relevance of all query-document pairs, we will employ an interventional approach that controls for relevance at each position. To start, let us first review how explicit interventions have been used for estimating $p_k := \Pr(E = 1|k)$ in the PBM [19, 26]. The PBM requires estimating a single vector $p = [p_1, p_2, ..., p_{k_{max}}]$ with p_k for each position $k \in [1, k_{max}]$. In this case, randomly swapping results at positions k and k' before presenting the ranking [19] makes the expected relevance of results at the two positions equal. An illustrative example is given in Figure 1a for $k = 1$ and $k' = 3$. Through the randomized swap, documents d_1 and d_3 have a 50% probability of being presented either at position $k = 1$ or at position $k' = 3$. So, over the distribution of queries that are subject to this randomized swapping, the distribution of documents in position $k = 1$ is identical to the distribution of documents in $k = 3$, and thus is their expected relevance. This randomized control for relevance resolves the ambiguity in attributing the lack of clicks to either a lack of relevance or a lack of observation.

More formally, denote with $C^k_{k,k'}$ and $C^{k'}_{k,k'}$ the random variables indicating clicks on positions k and k' respectively for the set of training queries where the results at positions k and k' are swap-randomized with probability $q = 0.5$. Since the results are swapped uniformly, the expected relevance at positions k and k' is controlled to be equal at these positions, and thus expected click-through rates reveal the relative propensities via

$$\frac{p_k}{p_{k'}} = \frac{\mathrm{E}[C^k_{k,k'}]}{\mathrm{E}[C^{k'}_{k,k'}]}.$$

This means that the ratio of the observed click-through rates is a consistent estimator of the relative propensities p_k and $p_{k'}$ under the PBM [19]. Note that knowing the relative propensities with respect to a single "anchor" position (e.g. $\frac{p_k}{p_1}$) is sufficient, since the counterfactual ERM learning objective is invariant to multiplicative scaling [19].

While this ratio estimator is a sensible approach for the PBM, it is not directly applicable to the Contextual PBM even if we only need relative propensity estimates. In particular, a simple ratio of the observed click-through rates at different ranks will yield $\frac{\mathrm{E}_x[p_k(x)]}{\mathrm{E}_x[p_{k'}(x)]}$, where the expectation is over contexts. This is not the estimate we seek for the CPBM, since we need estimates of each specific $p_k(x)$ (up to multiplicative scaling) to de-bias clicked examples at position k under context x. To get such context-dependent propensity estimates, we will introduce a different estimator below. Furthermore, we will show how to avoid explicit swap interventions by harvesting implicit interventions. As illustrated below, such implicit interventions are typically available in large quantities and do not come at the expense of user experience related to randomly swapping results.

4.3 Intervention Harvesting for the CPBM

Instead of explicitly swapping results, Agarwal et al. [3] have recently shown for the PBM how interventions similar to explicit swaps can be harvested from data that is readily available in most operational systems. We will extend this approach to the CPBM and derive an intervention-harvesting estimator for the CPBM that does not require explicit swap interventions, nor does it require a document-specific relevance model that would be difficult to fit. Instead, our estimator merely needs to model how the *average relevance* over all queries and documents at a position – not the context-document specific relevance – changes with context.

As input for our estimator, suppose we have data from m historic rankers $F = \{f_1, ..., f_m\}$. Each ranker f maps a query context x to a ranking $f(x)$ of the candidate set of documents. Let $\mathrm{rk}(d|f(x))$ denote the rank of document d in the ranking. Let n_i be the number of queries that f_i processed, and let $\mathcal{L} = \{(x^j, d^j, k^j, c^j) | j \in [N]\}$ be the aggregated click log over all the rankers, with one tuple for each context-document pair. We require that the distribution of contexts is stationary, or specifically that there is no dependency between the context and the choice of ranking function f_i [3, 20],

$$\forall f_i : \Pr(\mathcal{X}|f_i) = \Pr(\mathcal{X}) \Rightarrow \forall x \in \mathcal{X} : \Pr(f_i|x) = \Pr(f_i). \tag{3}$$

This condition is fulfilled in at least two situations – namely in A/B tests and under stationary $\Pr(\mathcal{X})$. In data from A/B tests, where users are randomly assigned to one of the rankers, the condition is fulfilled by design. An example is shown in Figure 1b. For a given context x, the ranking functions f_1, f_2 and f_3 are each chosen completely randomized with equal probability $\frac{1}{3}$. By choosing one of the three rankers, we implicitly conduct a number of interventions. For example, document d_1 is randomized to be displayed in positions 1, 2, or 3 with equal probability, and document d_2 is displayed in position 1 with probability $\frac{2}{3}$ and in position 2 with probability $\frac{1}{3}$. Figure 1c shows that a similar randomization holds when the production ranker gets updated from f_1 to f_2 under stationary $\Pr(\mathcal{X})$. Stationarity implies that the probability of a context x is equal before and after the update, and thus d_4 has twice the probability of being shown in position 4 than in position 5 in this toy example with 3-time steps.

To exploit this readily available intervention data for estimating the CPBM, let's first focus on a fixed pair of positions k, k'. The key idea of intervention harvesting for the CPBM is to control for the varying average relevance of results displayed in positions k, k' for context x by restricting to the set of queries that, for an appropriate choice of ranker from $F = \{f_1, ..., f_m\}$, could have been placed either at k or k'. To this effect, we define interventional sets

$$S_{k,k'} := \{(x, d) : \exists f, f' \ \mathrm{rk}(d|f(x)){=}k \wedge \mathrm{rk}(d|f'(x)){=}k'\} \tag{4}$$

as the sets of (x, d) pairs that receive "treatments" k or k' under different rankers. Specifically, a context-document pair (x, d) is included in $S_{k,k'}$, if for the context x some ranker $f \in F$ puts the document d at position k and another ranker $f' \in F$ puts it at position k'. This is akin to a virtual swap intervention at positions k and k', albeit only with a single document. Based on these definitions, the toy example in Figure 1b produces interventional sets such that $(x, d_1) \in S_{1,2}, S_{1,3}, S_{2,3}$, $(x, d_2) \in S_{1,2}$, $(x, d_4) \in S_{4,5}$, etc. Note that the set includes all possible queries that may be sampled, not only those that are actually sampled in one or more rankers' logs. Furthermore, note that the feedback signals of (x, d) from some rankers might remain counterfactual and unobserved. Illustrating this using the toy example in Figure 1b, each ranking of $F = \{f_1, f_2, f_3\}$ was a potential choice, but only one of those rankings was presented to the user – say the ranking of f_1. In this way, we only observe the feedback for d_1 at position 1, but not at the other positions.

To account for the fact that not all interventions within an interventional set $S_{k,k'}$ have the same probability, we define the following weighting function that is proportional to the treatment-assignment probability. It can either be computed from the known assignment probabilities in an A/B test, or for consecutive policy deployments via

$$q_k(x, d) := \frac{\sum_{i=1}^m n_i \mathbb{1}[\mathrm{rk}(d|f_i(x)) = k]}{\sum_{i=1}^m n_i}. \tag{5}$$

For the example in Figure 1b, we have $q_1(x, d_1) = q_2(x, d_1) = q_3(x, d_1) = \frac{1}{3}$, $q_4(x, d_4) = \frac{1}{3}$, $q_5(x, d_4) = \frac{2}{3}$, etc.

4.4 AllPairs Estimator for the CPBM

Now that we have extracted intervention data and its assignment mechanism from existing logs, we can tackle the question of defining an estimator for the CPBM using this data. The key challenge compared to analogous estimators for the PBM [3] lies in modeling the dependence on context. We start by constructing the following feedback labels for each $(x^j, d^j, k^j, c^j) \in \mathcal{L}$ by correcting the non-uniform assignment mechanism to the uniform intervention distribution in each interventional set [3].

$$\hat{c}^j_{k,k'}(k) := \mathbb{1}_{[(x^j, d^j) \in S_{k,k'}]} \mathbb{1}_{[k^j = k]} \frac{c^j}{q_k(x^j, d^j)}$$

$$\hat{\neg c}^j_{k,k'}(k) := \mathbb{1}_{[(x^j, d^j) \in S_{k,k'}]} \mathbb{1}_{[k^j = k]} \frac{1 - c^j}{q_k(x^j, d^j)}$$

This can be thought of as an IPS weighted class label. For the PBM without a dependence on context x, in expectation (over the choice of ranker and query) $\hat{c}^j_{k,k'}(k)$ is proportional to the product of examination propensity p_k and average relevance $r_{k,k'} := \Pr(\mathrm{rel}(x, d) = 1 | (x, d) \in S_{k,k'})$ [3], just like for the explicit swap interventions mentioned above. However, when there is a dependency on context x for both the examination probabilities $p_k(x)$ and the average relevance $r_{k,k'}(x) := \Pr(\mathrm{rel}(x, d) = 1 | (x, d) \in S_{k,k'}, x)$, unbiasedness w.r.t. the query distribution no longer holds and there is generally no small number of individual parameters p_k and $r_{k,k'}$ that could be estimated exhaustively. To overcome this problem, we exploit that unbiasedness still holds for each individual context, and we introduce a context-dependent examination model $h(k, x)$ for $p_k(x)$ and a context-dependent average relevance model $g(k, k', x)$ for $r_{k,k'}(x)$ to compactly capture the variation across contexts. In the experiments in this paper, we model both $h(k, x)$ and $g(k, k', x)$ as neural networks.

With these definitions in place, we can now formulate an extremum estimator similar to a maximum-likelihood criterion. We call this the AllPairs estimator for the CPBM. It combines the flexibility of the observational generative modeling approach with the robustness of the interventional methods, specifically the intervention harvesting approach previously used for estimating the PBM [3].

$$\begin{aligned} \hat{h}_{CPBM} := \operatorname*{argmax}_{h,g} \sum_{j \in \mathcal{L}} \sum_{k \neq k'} \Big[& \hat{c}^j_{k,k'}(k) \log \Big(h(k, x^j) g(k, k', x^j) \Big) \\ & + \hat{\neg c}^j_{k,k'}(k) \log \Big(1 - h(k, x^j) g(k, k', x^j) \Big) \Big] \end{aligned} \tag{6}$$

Here, $h(k, x)$ and $g(k, k', x)$ are constrained to $(0, 1)$ by using a sigmoid output layer on both networks. While the AllPairs estimator has syntactic similarity with the generative maximum-likelihood objective from [27], both are fundamentally different. Notably, AllPairs uses interventional data to control for the unobserved document relevance, while the generative model is purely observational. This allows the average relevance model $g(k, k', x)$ in AllPairs to be substantially simpler than the individual relevance model $g(q, d)$ in generative modeling. In particular, $g(k, k', x)$ in AllPairs does not model the relevance of an individual document to a query, but merely how the average relevance of documents in positions k and k' changes with context. As such, $g(k,k',x)$ does not require document-level relevance features, but merely takes the context x and the positions as input. In the experiments, we find that the average relevance at a specific position k does not change much with context x, and that even replacing the neural relevance model $g(k,k',x)$ with k_{max} choose 2 context-independent parameters $r_{k,k'}$ performs quite well.

We now further justify the use of the objective in Equation (6) by showing that it is equivalent to a weighted version of Cross-Entropy Maximization where the weights adjust for the varying amounts of interventional data available across position pairs k, k'. This relates the AllPairs objective to optimizing the KL-divergence between model and data, and it implies two practical advantages. First, for this type of objective, it is well known that training neural networks via backpropagation is effective. Second, this objective provides an attractive method for information aggregation, mitigating the noisiness and sparsity of click data.

Proposition 1. *Under the condition in (3) and i.i.d. contexts $x \sim \Pr(\mathcal{X})$, the objective in Equation (6) is equivalent to the following weighted form of Cross-Entropy,*

$$\sum_{x\in\mathcal{X}}\sum_{k\neq k'}\hat{N}_{k,k'}(x)\Big[\hat{y}_{k,k'}(k,x)\log\left(y_{k,k'}(k,x)\right)$$
$$+\hat{\neg}y_{k,k'}(k,x)\log\left(1-y_{k,k'}(k,x)\right)\Big]$$

of the random variables $y_{k,k'}(k,x)$ and their empirical counterparts $\hat{y}_{k,k'}(k,x)$, $\hat{\neg}y_{k,k'}(k,x)$ weighted with $\hat{N}_{k,k'}(x)$, where

$$y_{k,k'}(k,x) := h(k,x)g(k,k',x) = p_k(x)r_{k,k'}(x)$$
$$\hat{N}_{k,k'}(x) := \sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\mathbb{1}_{[(x^j,d^j)\in S_{k,k'}]}$$
$$\hat{y}_{k,k'}(k,x) := \frac{\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\hat{c}^j_{k,k'}(k)}{\hat{N}_{k,k'}(x)}$$
$$\hat{\neg}y_{k,k'}(k,x) := \frac{\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\hat{\neg}c^j_{k,k'}(k)}{\hat{N}_{k,k'}(x)},$$

and

$$\mathbb{E}[\hat{y}_{k,k'}(k,x)] = y_{k,k'}(k,x)$$
$$\mathbb{E}[\hat{\neg}y_{k,k'}(k,x)] = 1 - y_{k,k'}(k,x).$$

Proof. First, we rewrite the objective as follows:

$$\sum_{j\in\mathcal{L}}\sum_{k\neq k'}\Big[\hat{c}^j_{k,k'}(k)\log\left(y_{k,k'}(k,x^j)\right)+\hat{\neg}c^j_{k,k'}(k)\log\left(1-y_{k,k'}(k,x^j)\right)\Big]$$
$$=\sum_{x\in\mathcal{X}}\sum_{j\in\mathcal{L}}\sum_{k\neq k'}\mathbb{1}_{[x^j=x]}\Big[\hat{c}^j_{k,k'}(k)\log\left(y_{k,k'}(k,x^j)\right)$$
$$+\hat{\neg}c^j_{k,k'}(k)\log\left(1-y_{k,k'}(k,x^j)\right)\Big]$$
$$=\sum_{x\in\mathcal{X}}\sum_{k\neq k'}\Bigg[\Bigg(\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\hat{c}^j_{k,k'}(k)\Bigg)\log\left(y_{k,k'}(k,x^j)\right)$$
$$+\Bigg(\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\hat{\neg}c^j_{k,k'}(k)\Bigg)\log\left(1-y_{k,k'}(k,x^j)\right)\Bigg]$$
$$=\sum_{x\in\mathcal{X}}\sum_{k\neq k'}\hat{N}_{k,k'}(x)\Big[\hat{y}_{k,k'}(k,x)\log\left(y_{k,k'}(k,x^j)\right)$$
$$+\hat{\neg}y_{k,k'}(k,x)\log\left(1-y_{k,k'}(k,x^j)\right)\Big]$$

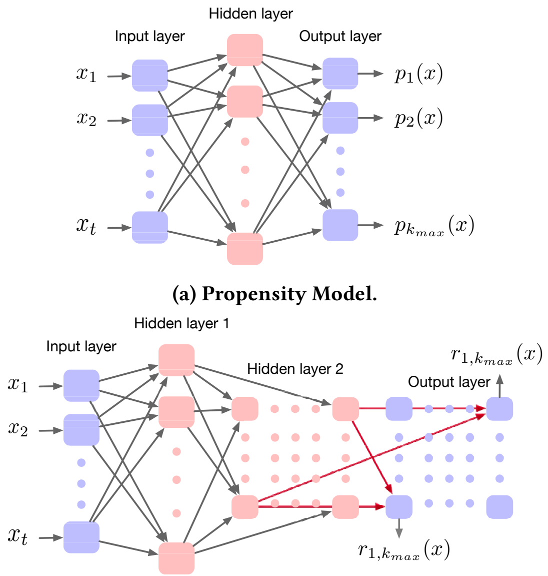
Next, we are going to prove that $\mathbb{E}[\hat{y}_{k,k'}(k,x)] = y_{k,k'}(k,x)$ and $\mathbb{E}[\hat{\neg}y_{k,k'}(k,x)] = 1-y_{k,k'}(k,x)$, which is required by Cross-Entropy.

$$\mathbb{E}[\hat{y}_{k,k'}(k,x)\hat{N}_{k,k'}(x)]$$
$$=\mathbb{E}[\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\mathbb{1}_{[(x^j,d^j)\in S_{k,k'}]}\mathbb{1}_{[k^j=k]}\frac{c^j}{q_k(x^j,d^j)}]$$
$$=\sum_{i=1}^{m}\sum_{j=1}^{n_i}\Pr(x)\sum_{d\in\Omega(x)}\mathbb{1}_{[(x,d)\in S_{k,k'}]}\mathbb{1}_{[\mathrm{rk}(d|f_i(x))=k]}\frac{\mathbb{E}[c(d)]}{q_k(x,d)}$$
$$=\Pr(x)\sum_{i=1}^{m}\sum_{j=1}^{n_i}\sum_{d\in\Omega(x)}\mathbb{1}_{[(x,d)\in S_{k,k'}]}\mathbb{1}_{[\mathrm{rk}(d|f_i(x))=k]}\frac{p_k(x)\,\mathrm{rel}(x,d)}{q_k(x,d)}$$
$$=p_k(x)\Pr(x)\sum_{d\in\Omega(x)}\mathbb{1}_{[(x,d)\in S_{k,k'}]}\,\mathrm{rel}(x,d)\frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i}\mathbb{1}_{[\mathrm{rk}(d|f_i(x))=k]}}{q_k(x,d)}$$
$$=p_k(x)\Pr(x)\sum_{d\in\Omega(x)}\mathbb{1}_{[(x,d)\in S_{k,k'}]}\,\mathrm{rel}(x,d)\frac{\sum_{i=1}^{m}n_i\mathbb{1}_{[\mathrm{rk}(d|f_i(x))=k]}}{q_k(x,d)}$$
$$=p_k(x)\Pr(x)\sum_{d\in\Omega(x)}\mathbb{1}_{[(x,d)\in S_{k,k'}]}\,\mathrm{rel}(x,d)\sum_{i=1}^{m}n_i$$
$$=p_k(x)\mathbb{E}[\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\mathbb{1}_{[(x^j,d^j)\in S_{k,k'}]}\,\mathrm{rel}(x^j,d^j)]$$
$$=p_k(x)r_{k,k'}(x)N_{k,k'}(x)$$
$$=y_{k,k'}(k,x)N_{k,k'}(x)$$

where $N_{k,k'}(x) = \mathbb{E}[\sum_{j\in\mathcal{L}}\mathbb{1}_{[x^j=x]}\mathbb{1}_{[(x^j,d^j)\in S_{k,k'}]}] = \mathbb{E}[\hat{N}_{k,k'}(x)]$. Then we have $\mathbb{E}[\hat{y}_{k,k'}(k,x)] = \frac{\mathbb{E}[\hat{y}_{k,k'}(k,x)\hat{N}_{k,k'}(x)]}{\mathbb{E}[\hat{N}_{k,k'}(x)]} = y_{k,k'}(k,x)$. Similarly, $\mathbb{E}[\hat{\neg}y_{k,k'}(k,x)] = 1 - y_{k,k'}(k,x)$. Note that we make the reasonable assumption that user click behavior is independent of the context sampling and ranker choice process, and thus $\hat{y}_{k,k'}(k,x)$ and $\hat{N}_{k,k'}(x)$ are independent random variables, so that $\mathbb{E}[\hat{y}_{k,k'}(k,x)\hat{N}_{k,k'}(x)] = \mathbb{E}[\hat{y}_{k,k'}(k,x)]\mathbb{E}[\hat{N}_{k,k'}(x)]$. □

4.5 Neural Network Model for the CPBM

We employ neural networks for modeling both the context-dependent propensities $h(k,x)$ as well as the context-dependent average relevance $g(k,k',x)$. The respective multi-layer perceptron (MLP) architectures are shown in Figure 2. Both networks take as input the context features $x \in \mathbb{R}^t$, and output the examination propensity vector $p(x) \in \mathbb{R}^{k_{max}}$ and the average relevance matrix $r(x) \in$



(a) Propensity Model.

(b) Relevance Model.

Figure 2: The architecture of the multilayer perceptrons.

$\mathbb{R}^{k_{max} \times k_{max}}$ respectively. The hidden layer in the propensity MLP is a traditional sigmoid-activated dense layer, which learns a weight matrix $\mathrm{W_p} \in \mathbb{R}^{t \times k_{max}}$ and bias vector $\mathrm{b_p} \in \mathbb{R}^{k_{max}}$ to produce a propensity vector $p(x) = \sigma(\mathrm{W_p}x + \mathrm{b_p})$. The average-relevance model is less standard, and its first hidden layer in the relevance MLP learns a 3d weight array $\mathrm{W_r} \in \mathbb{R}^{t \times k_{max} \times k_{max}}$ and bias matrix $\mathrm{b_r} \in \mathbb{R}^{k_{max} \times k_{max}}$ to produce an initial relevance matrix $\widetilde{r}(x) = \sigma(\mathrm{W_r}x + \mathrm{b_r})$. To ensure the symmetry of the relevance matrix, the second hidden layer of the relevance MLP computes $r(x) = (\widetilde{r}(x)^T + \widetilde{r}(x))/2$.

We conjecture that improvements to these models could further improve results. First, other neural networks may be good alternatives. For instance, in terms of the sequential examination process, we could iteratively output the propensities $p_1(x), p_2(x), ..., p_{max}(x)$ using a recurrent neural network (RNN), where the input sequence consists of repeated context features x. Second, embedding the position k as a feature would give rise to different network architectures. For example, the position could be encoded in a one-hot feature vector $\mathbf{k} \in \mathbb{R}^{k_{max}}$, which could then be concatenated to the context features x to predict the examination propensity $p_k(x)$.

5 EMPIRICAL EVALUATION

We empirically evaluate the effectiveness and robustness of our method through real-world experiments on the ArXiv Full-Text Search [1] and through semi-synthetic experiments on the Yahoo Learning-To-Rank Challenge corpus (set 1) [7]. The ArXiv experiments verify real-world relevance and applicability, while the synthetic experiments enable the evaluation of the method over a wide range of scenarios.

Table 1: Size of data sets from ArXiv.

Type	Swap intervention		A/B Test Harvesting	
	Clicks	Queries	Clicks	Queries
Complex	32,108	24,460	41,638	27,072
Simple	15,296	36,659	22,915	50,443

5.1 Real-World Evaluation: ArXiv Search

To verify that contextual effects on the propensity exist in real-world settings and to show that these can be estimated using intervention harvesting and the AllPairs estimator for the CPBM, we conducted a series of experiments on the ArXiv Full-Text Search. To get reliable propensity estimates that can serve as a gold-standard, we fielded explicit swap intervention in addition to an A/B test that we use for intervention harvesting. Specifically, we assigned equal probability of accepting an incoming query to these two mechanisms. For intervention harvesting, we used three ranking functions $\{f_1, f_2, f_3\}$ and chose uniformly at random between them for half of the incoming queries. For the other half, we also chose one of these ranking functions at random but inserted an explicit swap intervention between rank 1 and rank $k \in \{1, 2, ..., 21\}$. These explicit swap interventions were then used to get a gold-standard estimate of the propensities via the methods in [19]. To avoid any confounding due to changes in the query distribution, data for all conditions was collected in parallel between May 14, 2018 and December 13, 2018. In total, 138,600 queries and 112,000 clicks were collected, with about 61,100 queries for the explicit intervention and the rest for the intervention harvesting. For the following experiments, the data was randomly divided into a training set with 80 % of the data, a validation set with 10 %, and a test set with the remaining 10 %. In all experiments, the hyper-parameters of the neural networks in the CPBM were selected via cross-validation.

Do real-world propensity curves actually depend on context? We first verify that the propensity curves in ArXiv do indeed depend on context. To this effect, we introduce a single binary context feature that characterizes each query as either complex (denoted by 1) or simple (denoted by 0).

Complex queries are those that contain some logical operators from the Boolean query language supported by the search engine, such as "OR" and "AND", while simple queries are the remainder. The numbers of queries and clicks are given in Table 1. We then use the gold-standard propensity estimator from [19] to learn two PBM models from the swap intervention data, one for complex and one for simple queries.

Figure 3 shows that the two propensity curves are indeed substantially different. The shaded region for each curve depicts a 95 % confidence interval run on 1000 bootstrap samples. One possible interpretation is that complex queries are often used as more of a "lookup" rather than a search, and thus the first few results typically either match or the user reformulates. On the other hand, simple queries are often part of an exploratory search, such that users go further down the ranking.

Can AllPairs learn context-dependent propensity curves? Now that we know that contextual dependencies exist in real-world propensity curves, we can verify whether the AllPairs estimator

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

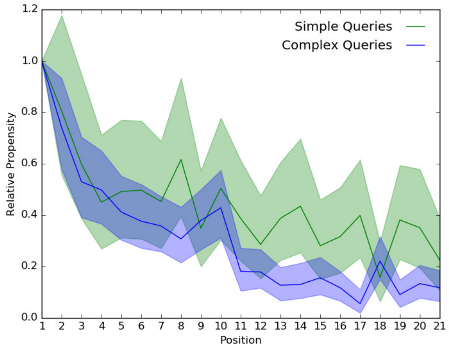


Figure 3: Propensity curves for simple and complex queries on ArXiv estimated as two PBM via swap interventions.

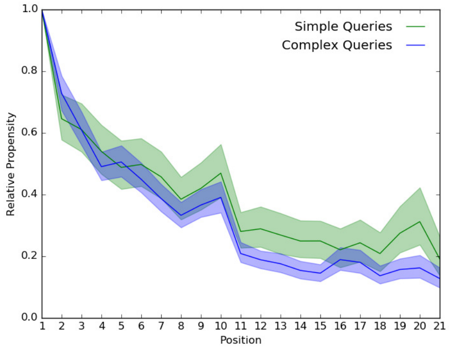


Figure 4: Propensity curves for simple and complex queries on ArXiv estimated as a CPBM via intervention harvesting.

with the neural CPBM model can accurately estimate these curves. Figure 4 show the propensity curves estimated by the AllPairs estimator on the intervention harvesting data from Table 1 using the neural model with only the single input feature. The curves closely match the gold standard in Figure 3, indicating that the CPBM can accurately learn these curves with a single neural network model. In addition, AllPairs achieves much improved error bars. This is to be expected, given that AllPairs makes more efficient use of the data than the ratio-estimates from [19].

Can AllPairs learn CPBM models with many context features? While it is infeasible to introduce additional features and learn separate PBM for each combination, adding context features to our neural CPBM model is straightforward. We will now explore in how far different groups of context features improve the predictive accuracy of the CPBM. Since we no longer have a gold-standard propensity curve to compare against, we instead use the AllPairs objective evaluated on a test set as our measure of predictive performance – similar to evaluating log-likelihood on a test set. We explore the following groups of context features:

(1) category: whether the query is specified as a category and its corresponding specified category ((binary $\{0, 1\}$, 10 features in total)
(2) query_len: whether the length of the query is greater than $X \in \{1, 2, 5, 10, 15, 20, 25, 30, 35, 40\}$ (binary $\{0, 1\}$, 10 features in total)
(3) ord_in_session: whether the order of the query in its session is greater than $X \in \{1, 2, 5, 10, 15\}$ ((binary $\{0, 1\}$, 5 features in total)
(4) #results: whether the number of results for each query is greater than $X \in \{1, 2, 5, 10, 15, 20, 50, 100, 150, 200\}$ (binary $\{0, 1\}$, 10 features in total)
(5) result_dist: the category distribution of each query (lies in $[0, 1]^{35}$ with sum to 1, 35 features in total)

Other reasonable features can also be taken into consideration, like query performance predictors [5, 11].

Table 2 shows the test-set performance. The baseline is a PBM model trained according to [3], which is essentially a CPBM model without features and a relevance model that explicitly represents each pairwise relevance. The table shows that the CPBM improves on the PBM in terms of predictiveness across all feature groups. The "category" and "query_len" features appear to have the largest influence on the propensity curve. However, the best predictive accuracy is achieved when all features are included in the CPBM. This verifies that the CPBM can make use of complex features to improve the fit of the propensity model.

5.2 Robustness Analysis: Yahoo LTR Challenge

We now turn to experiments on semi-synthetic data. Using a semi-synthetic setup combines the external validity of using a real-world dataset with the ability to fully explore a range of different settings for evaluating robustness.

Our semi-synthetic click data is based on the Yahoo LTR Challenge dataset. It contains manual relevance assessments as ground truth and we follow the given train/validation/test splits, but filter out queries that have no relevant documents. To generate click data for intervention harvesting, we learned two ranking functions by running SVM-Rank [16] on two small randomly sampled subsets of the training queries. To control the ranker similarity, 22 queries were the same for both rankers and each ranker independently sampled 92 additional queries. The remaining (roughly 11,400) queries of the training set were used to generate synthetic click data based on these two ranking functions.

To generate the click data via a CPBM, we need a model for the context features and an examination model. For context features, each query was mapped to a 10-dimensional feature vector x, concatenated by two parts: relevant part $[x_1, x_2, ..., x_i]$ and random part $[x_{i+1}, x_{i+2}, ..., x_{10}]$, and we use parameter $\zeta = \frac{i}{10}$ to control the dependency between relevance and context. For the relevant part, the important features which contribute to the relevance modeling were selected in the following way: we first used an SVM-Rank to get a one-sweep click log on the training split. Then we trained k_{max} logistic models $r_k(x), k \in [1, k_{max}]$, which denotes the average relevance at position k. Let the coefficient of each feature x_j among the given query-document feature vector in each model r_m be u_{jm}, we assigned each feature a score $s_j = \max_j |u_{jm}|$. We randomly selected i features from a candidate set which contains features x_j ranked in top-30 s_j list. At last, the relevant part was the average of the vector representations of all relevant results on those selected i features. For the random part, we drew $[x_{i+1}, x_{i+2}, ..., x_{10}]$ from the normal distribution $\mathcal{N}(0, \sigma^2)$. To keep the performance of the PBM stable with increasing ζ, σ was tuned to be 0.35.

Table 2: Objective on the test set for the PBM and the CPBM when including each feature group and for all features.

Model	PBM	CPBM					
		category	query_len	ord_in_session	#results	result_dist	All
Objective	-13926.18	-12622.96	-12674.8	-13205.21	-13241.28	-12901.94	-12306.52
Increment (vs. PBM)	-	1303.22	1251.38	720.97	684.90	1024.24	1619.66

Table 3: Relative decrease in the relative error of CPBM vs. PBM (#Training queries = 113590, $\eta = 0.5$).

	PBM	CPBM	Improvement
RelError	0.478700	0.169443	64.60%

For the examination model we chose $\Pr(E = 1|k, x) = \frac{1}{k^{\max(w \cdot x + 1, 0)}}$. The parameter vector w was drawn from a uniform distribution over the half-open interval $[-\eta, \eta)$, and we normalized the weight to $\sum_{i=1}^{10} w_i = 0$ by subtracting the average weight from each position.

The parameter η controls how much examination varies with context. In the extreme case of $\eta = 0$, there is no context dependency, and context dependency grows as η increases. We also incorporated click noise into the simulation by setting the probability of clicking on an irrelevant result to $\epsilon_- = 0.1$. We chose the maximum number of positions to be $k_{max} = 10$.

To evaluate the accuracy of the propensity estimates on a test sample $\mathcal{D} = \{x^j | j \in [M]\}$, we adopted the following relative error measure where $\hat{p}_k(x) = \frac{h(k, x^j)}{h(1, x^j)}$ are the estimated relative propensities and $p_k(x) = \frac{\Pr(E=1|k, x^j)}{\Pr(E=1|1, x^j)}$ are the true relative propensities are known by construction.

$$RelError(h) = \frac{1}{M} \sum_{j \in \mathcal{D}} \frac{1}{k_{max}} \sum_{k=1}^{k_{max}} \left| 1 - \frac{\hat{p}_k(x^j)}{p_k(x^j)} \right|$$

This measure evaluates the accuracy of the estimates in terms of their use as inverse relative propensity weights, which will be their primary function. The relative error reported below is evaluated on the test set, and error bars indicate the standard deviation estimated over 6 independent runs (except in Figure 8 as described below).

In our implementation of the AllPairs estimator, the propensity model and the relevance model were both implemented by a multilayer perceptron (described in Section 4.5), whose parameters were selected via cross-validation.

How much more accurate is the CPBM compared to the PBM?. Table 3 shows the RelError of the CPBM and the PBM on test data, where both are trained using the AllPairs estimator using a large amount of click data for training (113, 590 training queries). It can be thought of as the asymptotic performance of the respective model. The table shows that the CPBM improves substantially over the PBM, more than halving the error. This verifies that the AllPairs estimator can effectively learn context-dependent propensity curves from harvested interventions. Note that the CPBM had no knowledge of the true functional form of the examination model that was used to generate the clicks, but had to approximate it using the neural network model.

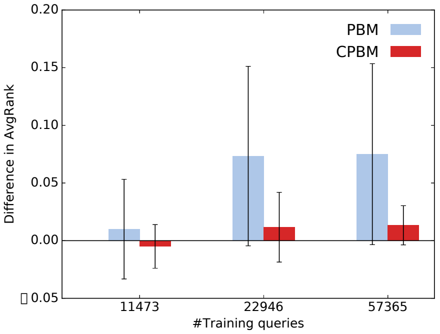


Figure 5: Difference in AvgRank compared to the true propensity model ($\eta = 10$, $\zeta = 1$).

Does the CPBM improve learning-to-rank performance? In practice, the propensities coming from the CPBM will typically be used for learning new ranking functions from the de-biased click data. We now evaluate whether the CPBM model improves learning performance compared to using the propensities from the PBM.

We trained a Clipped Propensity SVM-Rank [19] for each of the following three propensity models: PBM estimated via AllPairs, CPBM estimated via AllPairs, and – as gold standard – the true propensities used during synthetic data generation. All hyperparameters were picked via cross-validation. For rank $r > 21$, we impute the propensity $p_r(x) = p_{21}(x)$. Following [19], we measure test-set ranking performance via the average sum of the ranks of the relevant results across the queries in the test set $\mathcal{D}$,

$$AvgRank(f) = \frac{1}{M} \sum_{j \in \mathcal{D}} \sum_{d \in \Omega(x^j)} \mathrm{rk}(d | f(x^j)) \, \mathrm{rel}(x^j, d).$$

Figure 5 shows ranking performance relative to the performance of the Propensity SVM-Rank that has access to the true propensities. For sufficiently large data set sizes, the performance when using the CPBM propensities appears closer to the gold-standard performance than when using the PBM. This is to be expected, since the training objective the Propensity SVM-Rank is known to be biased for the misspecified propensities of the PBM, so that more data no longer translates into better learning performance.

How much data is needed to learn a CPBM? So far, we have used large amounts of training data to study the asymptotic performance of AllPairs for the CPBM. But how much data is really needed? Figure 6 compares the error of the three models across a wide range of training data sizes. The figure shows that a much smaller number of training examples suffices to get good accuracy. In particular, the relative error decreases quickly and asymptotes at about 5,700 training queries. Furthermore, Figure 6 shows that

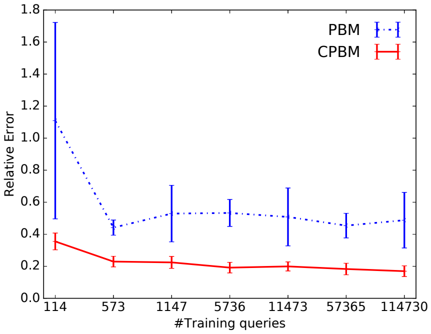


Figure 6: Relative error with increasing number of training queries ($\eta = 0.5$, $\zeta = 1$).

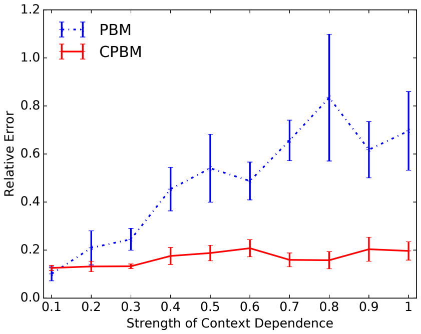


Figure 7: Relative error with increasing strength of context dependence η (#Training queries = 57365, $\zeta = 1$).

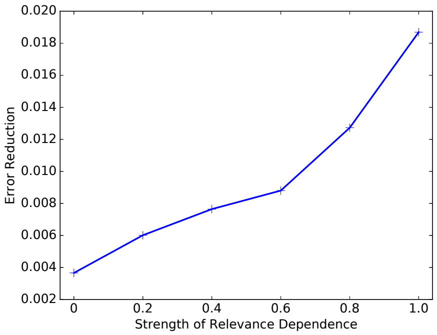


Figure 8: Error reduction by incorporating a relevance model with increasing strength of relevance dependence ζ (#Training queries = 57365, $\eta = 1$).

the CPBM dominates the PBM across the whole range of data-set sizes, even when the amount of click data is quite small.

How does the strength of context dependence affect the CPBM? We explore the behavior of the estimators when we vary the strength of context dependence via η. Results are shown in Figure 7, where the CPBM outperforms or at least matches the PBM across the whole range. As expected, the error of the PBM increases as the strength of context dependence increases. In contrast, the CPBM can capture the context dependence effectively.

How important is it to incorporate a relevance model in the estimator? Figure 8 shows the error reduction between the estimators under the CPBM with and without a context-dependent relevance model. For the CPBM with a context-dependent relevance model, we use the neural-network relevance model $g(k, k', x)$, and for the other one we simply use context-independent parameters $r_{k,k'}$ for each pair of ranks. To ensure statistical stability, we reran the experiment 20 times. The error reduction provided by context-dependent relevance model increases when the context has increasing influence on the relevance profile. With maximum decrease in error of only 0.02, the context-dependent relevance model provides only a mild improvement to the accuracy of the estimates. This highlights the desirable fact that the relevance model $g(k, k', x)$ can be far less crucial than the query-document relevance model $g(q, d)$ in generative models.

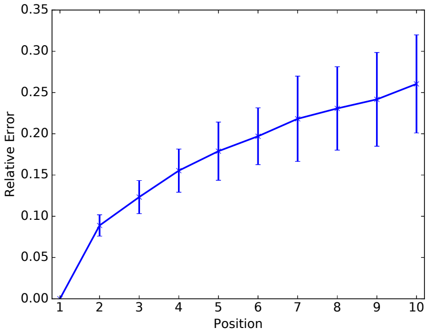


Figure 9: Relative error of the CPBM at different positions in the ranking (#Training queries = 114730, $\eta = 0.5$, $\zeta = 1$).

How accurate is the estimate at different positions in the ranking? Figure 9 shows the relative error of the CPBM at different positions in the ranking. As expected, the relative error increases with position, because lower-ranked documents have a smaller chance of receiving clicks and thus have less training data from intervention harvesting. Furthermore, the examination propensities at lower ranks are generally smaller, such that absolute deviations in the propensity estimates lead to larger contributions to our relative error metric.

6 CONCLUSIONS

We introduced the Contextual Position-Based Model (CPBM) to better capture the examination bias in interaction feedback from rankings. The CPBM captures how examination changes with context, and we developed an estimator for learning a CPBM from implicit feedback data. The key idea is to harvest interventions from the logs of multiple historic rankers, which provides experimental control to eliminate confounding of relevance on examination. Plugging a neural network model into the estimator, we show how the CPBM and the estimator can effectively learn context-dependent examination models in simulation experiments and real-world experiments.

ACKNOWLEDGMENTS

This research was supported in part by NSF Awards IIS-1615706 and IIS-1513692, as well as a gift from Google. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A General Framework for Counterfactual Learning-to-Rank. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*.

[2] Aman Agarwal, Ivan Zaitsev, and Thorsten Joachims. 2018. Counterfactual Learning-to-Rank for Additive Metrics and Deep Models. In *ICML Workshop on Machine Learning for Causal Inference, Counterfactual Prediction, and Autonomous Action (CausalML)*.

[3] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Interventions. In *International Conference on Web Search and Data Mining (WSDM)*.

[4] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *The 41st International ACM SIGIR Conference on Research ; Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 385–394. https://doi.org/10.1145/3209978.3209986

[5] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. 2004. Query difficulty, robustness, and selective application of query expansion. In *European conference on information retrieval*. Springer, 127–137.

[6] Alexey Borisov, Martijn Wardenaar, Ilya Markov, and Maarten de Rijke. 2018. A Click Sequence Model for Web Search. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 45–54. https://doi.org/10.1145/3209978.3210004

[7] Olivier Chapelle and Yi Chang. 2011. Yahoo! Learning to Rank Challenge Overview. In *Proceedings of the Learning to Rank Challenge (Proceedings of Machine Learning Research)*, Olivier Chapelle, Yi Chang, and Tie-Yan Liu (Eds.), Vol. 14. PMLR, Haifa, Israel, 1–24. http://proceedings.mlr.press/v14/chapelle11a.html

[8] Olivier Chapelle and Ya Zhang. 2009. A Dynamic Bayesian Network Click Model for Web Search Ranking. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, New York, NY, USA, 1–10. https://doi.org/10.1145/1526709.1526711

[9] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7, 3 (2015), 1–115.

[10] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-bias Models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08)*. ACM, New York, NY, USA, 87–94. https://doi.org/10.1145/1341531.1341545

[11] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting Query Performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)*. ACM, New York, NY, USA, 299–306. https://doi.org/10.1145/564376.564429

[12] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11)*. Omnipress, USA, 1097–1104. http://dl.acm.org/citation.cfm?id=3104482.3104620

[13] Georges E. Dupret and Benjamin Piwowarski. 2008. A User Browsing Model to Predict Search Engine Click Data from Past Observations.. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, USA, 331–338. https://doi.org/10.1145/1390334.1390392

[14] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click Chain Model in Web Search. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, New York, NY, USA, 11–20. https://doi.org/10.1145/1526709.1526712

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

[16] Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. ACM, New York, NY, USA, 133–142. https://doi.org/10.1145/775047.775067

[17] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data As Implicit Feedback, In ACM SIGIR Forum. *SIGIR Forum* 51, 1, 4–11. https://doi.org/10.1145/3130332.3130334

[18] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search. *ACM Trans. Inf. Syst.* 25, 2, Article 7 (April 2007). https://doi.org/10.1145/1229179.1229181

[19] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. ACM, New York, NY, USA, 781–789. https://doi.org/10.1145/3018661.3018699

[20] John Langford, Alexander Strehl, and Jennifer Wortman. 2008. Exploration Scavenging. In *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*. ACM, New York, NY, USA, 528–535. https://doi.org/10.1145/1390156.1390223

[21] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual Estimation and Optimization of Click Metrics in Search Engines: A Case Study. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. ACM, New York, NY, USA, 929–934. https://doi.org/10.1145/2740908.2742562

[22] Maeve O'Brien and Mark T Keane. 2006. Modeling result-list searching in the World Wide Web: The role of relevance topologies and trust bias. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*. Citeseer, 1–881.

[23] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-through Rate for New Ads. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 521–530. https://doi.org/10.1145/1242572.1242643

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

[25] Adith Swaminathan and Thorsten Joachims. 2015. Batch Learning from Logged Bandit Feedback Through Counterfactual Risk Minimization. *J. Mach. Learn. Res.* 16, 1 (Jan. 2015), 1731–1755. http://dl.acm.org/citation.cfm?id=2789272.2886805

[26] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '16)*. ACM, New York, NY, USA, 115–124. https://doi.org/10.1145/2911451.2911537

[27] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. ACM, New York, NY, USA, 610–618. https://doi.org/10.1145/3159652.3159732