Learning to Rank with Selection Bias in Personal Search

Xuanhui Wang, Micheal Bendersky, Donald Metzler, Marc Najork

Google Inc. Mountain View, CA

SIGIR, 2016

Outline

Background

Why selection bias on personal search

Problem Formulation

Selection Bias Problem

Proposed Methods

How to solve the bias problem

Experiments

The performance of the proposed methods

Outline

Background

Click-through data

Background

Click-through data

- Pros: natural, abundant and continuously renewable.
- Cons: noisy, biased(position bias, presentation bias, trust bias)

Background

Click-through data

- Pros: natural, abundant and continuously renewable.
- Cons: noisy, biased(position bias, presentation bias, trust bias)
  - Cascade model: probability of documents at position i
  - Assume access to large quantities of click data for each document given the query

Background

Click-through data

- Pros: natural, abundant and continuously renewable.
- Cons: noisy, biased(position bias, presentation bias, trust bias)
  - Cascade model: probability of documents at position i
  - Assume access to large quantities of click data for each document given the query

Personal search

- Email search, desktop search, on-device search.
- Properties
  - Each user has access only to their own private document corpus
  - Relevance judgment restricted by privacy

Outline

Learning to Rank

- Let $Q = (q, \{x_1, ..., x_n\})$ denote a query string q and its set of result documents. Let $P(Q)$ denote the probability of observing query Q in the universe $\mathcal{Q}$.
- The goal of learning-to-rank is to find a scoring function $f(x)$ that can minimize the loss function defined as:

$$L(f) = \int_{Q \in \mathcal{Q}} l(Q, f) dP(Q)$$

where $l(Q, f)$is the incurred loss of scoring function f applied to query Q.

- Let $x_i \succ_Q x_j$ denote all pairs x_i, x_j of result documents in Q for which x_i is more relevant than x_j ➔ Pair-wise loss

$$l(Q, f) = \sum_{x_i \succ_Q x_j} max(0, f(x_j) - f(x_i))^2$$

Learning to Rank

- The loss function

$$L(f) = \int_{Q \in \mathcal{Q}} l(Q, f) dP(Q)$$

- In practice, we form a uniformly random sample $\mathcal{U} = \{Q \in \mathcal{Q} : Q \sim \mathcal{Q}\}$

$$L_{\mathcal{U}}(f) = \frac{1}{|\mathcal{U}|} \sum_{Q \in \mathcal{U}} l(Q, f)$$

Selection Bias Problem

- Two ways to obtain relevance estimates for $\mathcal{U}$
  - Explicit judgments from human raters
  - Implicit judgments, such as click-through data
- Click-through data: biased and noisy.
  - Queries without clicks provide no useful information for optimization pair-wise loss function

$$l(Q, f) = \sum_{x_i \succ_Q x_j} max(0, f(x_j) - f(x_i))^2$$

Selection Bias Problem

- Let $\mathcal{S}$ denote the collection of queries with clicks.

Selection Bias Problem

- Let $\mathcal{S}$ denote the collection of queries with clicks.
  - $\mathcal{S}$ is biased. Formally, let $\hat{P}(Q)$ denote the probability mass of query Q in $\mathcal{S}$, then $\hat{P}(Q) \neq P(Q)$

Selection Bias Problem

- Let $\mathcal{S}$ denote the collection of queries with clicks.
  - $\mathcal{S}$ is biased. Formally, let $\hat{P}(Q)$ denote the probability mass of query Q in $\mathcal{S}$, then $\hat{P}(Q) \neq P(Q)$

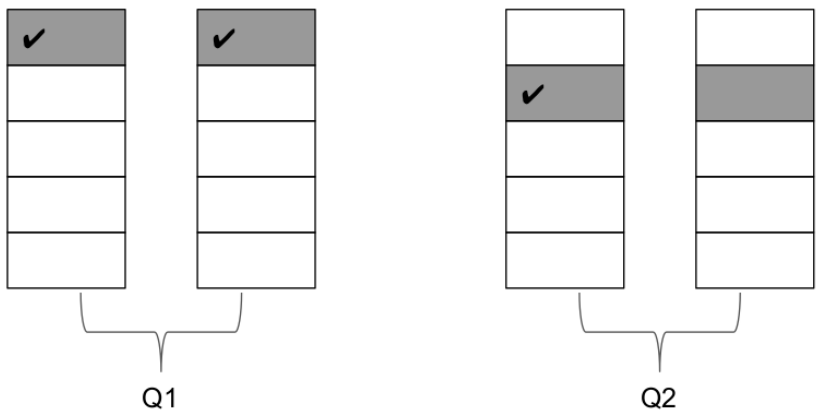

Figure 1: An illustration of selection bias in click data. The shaded documents are the relevant ones. A check mark means the document is clicked.

$P(Q_1) = P(Q_2)$ for set $\mathcal{U}$

Suppose the relevant document for Q_2 is clicked half of the time when the query is issued.

$\hat{P}(Q_1) = 2\hat{P}(Q_2)$ for set $\mathcal{S}$

Inverse Propensity Weighting

- $\hat{P}(Q)$ as the propensity score of Q, then let

$$w_Q = \frac{P(Q)}{\hat{P}(Q)}$$

- The empirical loss function

$$L_{\mathcal{S}}(f) = \frac{1}{|\mathcal{S}|} \sum_{Q \in \mathcal{S}} \frac{P(Q)}{\hat{P}(Q)} l(Q, f) = \frac{1}{|\mathcal{S}|} \sum_{Q \in \mathcal{S}} w_Q \cdot l(Q, f)$$

- w_Q?

Outline

Global Bias Model

- The global bias model, like the standard position bias model, which assumes the bias is a function of the position within the ranked list.
  - c_{xi}^{Q}: the probability of receiving a click.
    - For query Q, document x is shown at position i.
  - r_{x}^{Q}: the probability of relevance of x to Q.
  - b_i: the bias at position i.
    - How likely a user is to examine the document at this position.

$$c_{xi}^{Q} = r_{x}^{Q} \cdot b_i$$

Global Bias Model

- In the randomized data, the probability of showing $x \in Q$ is the same for all positions. $P(x|Q, i_1) = P(x|Q, i_2)$ for all $1 \leq i_1, i_2 \leq n$.

$$\int_{x \in Q} r_x^Q dP(x|Q, i_1) = \int_{x \in Q} r_x^Q dP(x|Q, i_2)$$

$$b_i = \frac{\sum_{Q \in \mathcal{R}} \int_{x \in Q} c_{xi} dP(x|Q, i)}{\sum_{Q \in \mathcal{R}} \int_{x \in Q} r_x dP(x|Q, i)} \propto \sum_{Q \in \mathcal{R}} \int_{x \in Q} c_{xi} dP(x|Q, i)$$

$$w_Q = \frac{P(Q)}{\hat{P}(Q)} \propto \frac{1}{b_i}$$

Global Bias Model

- $n = 4$

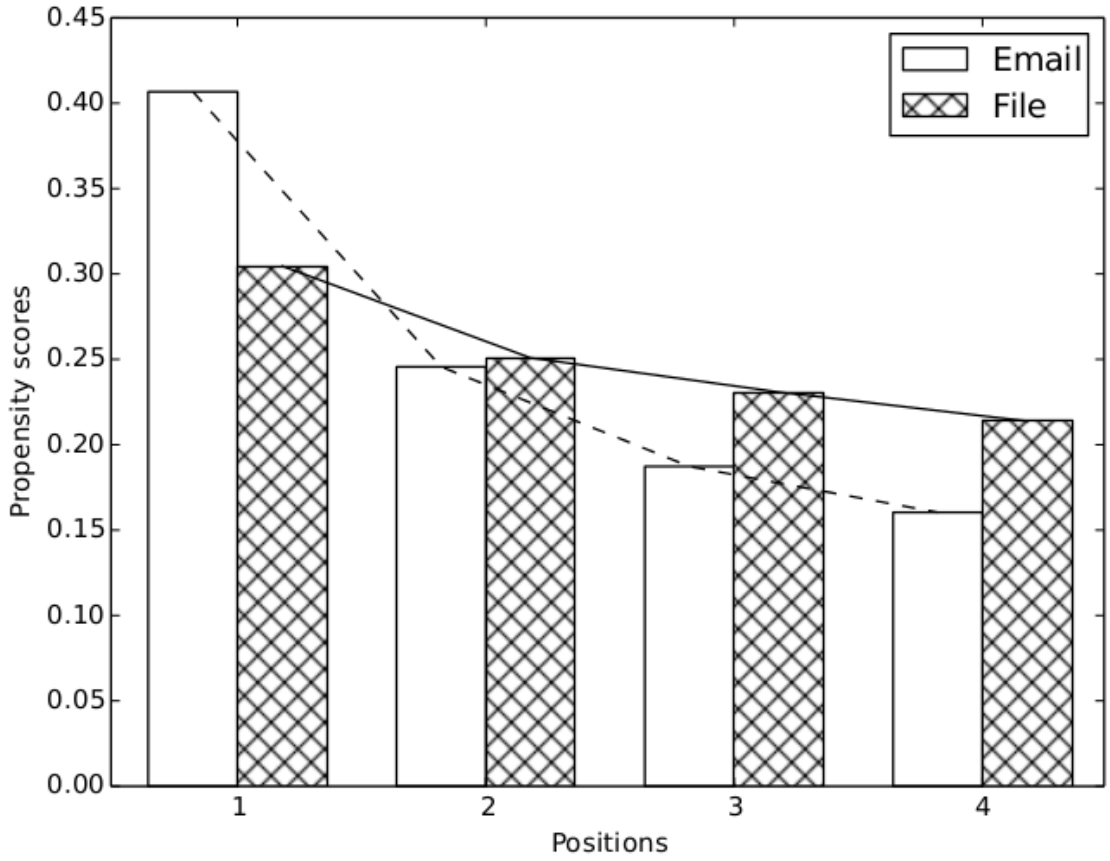

Figure 2: The position bias propensity scores for user emails and cloud storage files.

Segmented Bias Model

- Different segments of queries have different position biases.
- Basic idea
  - Partition queries into a few segments → apply the global model to each segment → a specific bias model for each segment.

Segmented Bias Model

- Segment
  - Categories assigned to each email: *promotional*, *social*.
  - Each email can be associated with multiple labels.
    → select a single label for each query.
  - Similar to IDF(Inverse Document Frequency).

$$IQF(t) \propto \frac{1}{|\{Q : t \in Q\}|}$$

$$t(Q) = arg \max_{t \in Q} \{IQF(t)\}$$

$$w_Q \propto \frac{1}{b_i^{t(Q)}}$$

Generalized Bias Model

- A generalized query-dependent bias model?
  - Data sparseness: each query Q needs thousands of data points.
  - Different documents for the same query in private search.
- Multi-class logistic regression.
  - **Label**: a query instance belongs to class i if we have the clicked position i.
  - **Features**: for each query Q, construct a feature vector $v(Q)$
  - **Training**: n logistic regression models, each for a single position.

$$b_i^Q = \frac{1}{1 + exp(\beta_i \cdot v(Q))}$$

  - **Prediction**: each query has n prediction values corresponding to n positions, which is the position bias.

Outline

Experimental Design

- Email Data Sets
  - Regular Data
    - Collected from click logs to learn a score function
  - Randomized Data
    - Randomly permuted the top search results of email search queries to estimate the bias.
- Learning-to-Rank Algorithm
  - Aim to train the adjustment $\delta(x)$ over the base score $s(x)$.

$$f(x) = s(x) + \delta(x)$$

  - Ranking features for $\delta(x)$
    - Email categories.
    - User interactions.

Perplexity on Randomized Data

- Perplexity.

$$perplexity = 2^{-\frac{1}{N}\sum_{o=1}^{N} log_2 p_o}$$

- A lower perplexity score means the model is better at predicting the observations.
- p_o is the predicted bias probability for the sample.
- $n = 4$.

	Uniform	Global	Segmented	Generalized
Mean	4.0	3.7360	3.7337	**3.7336**
95% CI	/	±0.0202	±0.0201	±0.0197

Table 3: Perplexity on the randomized data with 95% Confidence Interval.

Offline Evaluation on Regular Data

- Mean Reciprocal Rank(MRR) ➔ Weighted MRR.

$$MRR = \frac{1}{|\mathcal{S}|} \sum_{Q \in \mathcal{S}} \frac{1}{rank_Q} \qquad MRR = \frac{1}{\sum_{Q \in \mathcal{S}} w_Q} \sum_{Q \in \mathcal{S}} w_Q \frac{1}{rank_Q}$$

- Different position bias prediction models ➔ Different w_Q.
  - Normalize raw MRR values by the smallest value.

	Weighted MRR in Eq 7		
	Global w_Q	Segmented w_Q	Generalized w_Q
NoCorrection	1.0055	1.0055	1.0000
Global	1.0154	-	-
Segmented	-	1.0156	-
Generalized	-	-	1.0101
Improvement	0.9822%	0.9968%	1.0099%

Table 4: Offline evaluation on the regular data based on the weighted MRR. The number in this table is normalized by the smallest MRR.

Unbiased Offline Evaluator

Algorithm 1 Offline Evaluator

Input: scoring function f; randomized data $\mathcal{R}$; evaluation metric M on top k: M_k.
Output: evaluation value of f.

1: Set matched data collection $\mathcal{R}_s := \emptyset$
2: **for** $Q = (q, \langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle)$ in $\mathcal{R}$ **do**
3: Let $\langle \mathbf{x}_{j_1}, ..., \mathbf{x}_{j_n} \rangle$ be $\langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle$ re-ranked by f
4: **if** $\langle j_1, ..., j_k \rangle = \langle 1, ..., k \rangle$ **then**
5: $\mathcal{R}_s := \mathcal{R}_s \cup Q$
6: **end if**
7: **end for**
8: **return** $M_k(\mathcal{R}_s)$

- $\mathcal{R}_s$: selected subset of queries.
- Evaluate weighted MRR on the top-k results
  $MRR = \frac{1}{\sum_{Q \in \mathcal{S}} w_Q} \sum_{Q \in \mathcal{S}} w_Q \frac{1}{rank_Q}$

Unbiased Offline Evaluator

- $MRR = \frac{1}{\sum_{Q \in \mathcal{S}} w_Q} \sum_{Q \in \mathcal{S}} w_Q \frac{1}{rank_Q}$

k	$\lvert\mathcal{R}_s\rvert$	Global	Segmented	Generalized
1	19.8K	0.94%	**1.01%**	0.97%
2	6.7K	1.08%	**1.28%**	1.20%
3	3.3K	1.58%	1.67%	**1.68%**
4	3.3K	1.37%	**1.44%**	1.41%

Table 5: Comparison of different position bias prediction methods using the unbiased offline evaluator. We report the relative improvement over the NoCorrection baseline.

Online Experiments on Live Traffic

- A/B testing
  - One half as control: the NoCorrection model
  - One half as treatment: one of the bias prediction model.

MRR			
Baseline	Global	Segmented	Generalized
NoCorrection	0.67%***	0.88%***	0.79%***
Global	-	0.21%*	0.12%
CTR			
Baseline	Global	Segmented	Generalized
NoCorrection	0.46%***	0.71%***	0.62%***
Global	-	0.25%**	0.15%

Table 6: Comparison of different bias prediction methods using online experiments. We report the relative improvement over the NoCorrection baseline. Notation *, ** and * means the difference is significant at level 0.1, 0.05 and 0.01 respectively.**

Conclusion

- The infeasibility of using existing click models in personal search.
- A novel approach to overcome the selection bias.
  - Three models to to estimate the selection bias.
  - Address selection bias using inverse propensity weighting.
- Offline and online experiments.

Questions?