Unbiased Learning-to-Rank with Biased Feedback

Thorsten Joachims, Adith Swaminathan, Tobias Schnabel
International Conference on Web Search and Data Mining, 2017

Presenter: Thorsten Joachims
Dept of Computer Science, Cornell University
Interaction Logs: Search Engine

• Context $x$:
  – Query

• Action $y$:
  – Ranking

• Reward/Loss $\Delta(y|x)$:
  – Search cost
  – Information gained

• Feedback:
  – Clicks on SERP
Interaction Logs: Online Retail

- **Context $x$:**
  - Category
- **Action $y$:**
  - Tile Layout
- **Reward/Loss $\Delta(y|x)$:**
  - Search cost
  - Product utility
- **Feedback:**
  - Purchases
Interaction Logs: Streaming Media

• **Context** $x$:
  – User

• **Action** $y$:
  – Carousel layout

• **Reward/Loss** $\Delta(y|x)$:
  – Search cost
  – Enjoyment

• **Feedback**:
  – Plays
Learning-to-Rank from Clicks

Query Distribution
\( x_i \sim P(X) \)

Deployed Ranker
\( \tilde{y}_i = \pi_0(x_i) \)

Presented \( \bar{y}_n \)

Click

B

C

Click

E

F

G

Learning Algorithm

New Ranker
\( \pi(x) \)

Should perform better than
\( \pi_0(x) \)
Eye-Tracking

Detect and record where and what people look at

- **Fixations**: ~200-300ms; information is acquired

- **Saccades**: extremely rapid movements between fixations

- **Pupil dilation**: size of pupil indicates interest, arousal

“Scanpath” output depicts pattern of movement throughout screen. Black markers represent fixations.
How Many Links do Users View?

Total number of abstracts viewed per page

Total number of abstracts viewed

frequency

Total number of abstracts viewed

1 2 3 4 5 6 7 8 9 10

Total number of abstracts viewed per page
In Which Order are Results Viewed?

Users tend to read the results in order.
Examination Curve from Eyetracking

- Time spent in each result by frequency of doc selected
- Graph showing the position, exposure, and feedback relationship

Position ➔ Exposure ➔ Feedback

[Granka et al., 2007]
Outline

• Learning-to-Rank from User Interactions
  – Find new ranking policy $\pi$ that selects $y$ with better $\delta$

• Batch Learning-to-Rank from Partial Labels
  – Learning from partial and biased feedback
  – Learning Principle: Unbiased Partial-Information ERM
  – Learning Algorithm: Propensity SVM-Rank

• Propensity Estimation for Ranking
  – Break confounding through position randomization
  – Intervention Harvesting
  – Contextual propensity models
Evaluating Rankings

Deployed Ranker
\( \bar{y} = \pi_0("SVM") \)

New Ranker to Evaluate
\( y = \pi("SVM") \)

Manually Labeled

Presented \( \bar{y} \)
- A
- B
- D
- E
- F
- G

Click

New \( y \)
- F
- G
- D
- C
- E
- A
- B

1
2
3
4
Evaluation with Missing Judgments

- Loss: $\Delta(y|r)$
  - Relevance labels $r_i \in \{0,1\}$
  - This talk: rank of relevant documents
    $\Delta(y|r) = \sum_i \text{rank}(i|y) \cdot r_i$

- Assume:
  - Click implies observed and relevant:
    $(c_i = 1) \leftrightarrow (o_i = 1) \land (r_i = 1)$

- Problem:
  - No click can mean not relevant OR not observed
    $(c_i = 0) \leftrightarrow (o_i = 0) \lor (r_i = 0)$

→ Understand observation mechanism

[Joachims et al., 2017]
Inverse Propensity Score Estimator

- Observation Propensities $Q(o_i = 1|x, \bar{y}, r)$
  - Random variable $o_i \in \{0, 1\}$ indicates whether relevance label $r_i$ for $y$ is observed

- Inverse Propensity Score (IPS) Estimator:

$$\hat{\Delta}(y|r, o) = \sum_{i: c_i = 1} \frac{\text{rank}(i|y)}{Q(o_i = 1|\bar{y}, r)}$$

- Unbiasedness: $E_o[\hat{\Delta}(y|r, o)] = \Delta(y|r)$

**Presented $\bar{y}$**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

[Horvitz & Thompson, 1952] [Rubin, 1983] [Zadrozny et al., 2003] [Langford, Li, 2009] [Joachims et al., 2017]
ERM for Partial-Information LTR

- Unbiased Empirical Risk:
  \[ \hat{R}_{IPS}(\pi) = \frac{1}{N} \sum_{(x, y, c) \in S, i: c_i = 1} \sum \frac{\text{rank}(i|\pi(x))}{Q(o_i = 1|\bar{y}, r)} \]

- ERM Learning:
  \[ \hat{\pi} = \arg\min_{\pi} \hat{R}_{IPS}(\pi) \]

- Questions:
  - How do we optimize this empirical risk in a practical learning algorithm?
  - How do we define and estimate the propensity model \( Q(o_i = 1|\bar{y}, r) \)?
Propensity-Weighted SVM Rank

• Data:

\[ S = (x_j, d_j, D_j, q_j)^n \]

• Training QP:

\[
\begin{align*}
    w^* &= \text{argmin}_{w, \xi \geq 0} \frac{1}{2} w \cdot w + \frac{C}{n} \sum_j \frac{1}{q_j} \sum_i \xi_j \\
    \forall \bar{d}_i \in D_1: w \cdot [\phi(x_1, d_1) - \phi(x_1, \bar{d}_i)] &\geq 1 - \xi_1 \\
    \vdots \\
    \forall \bar{d}_i \in D_n: w \cdot [\phi(x_n, d_n) - \phi(x_n, \bar{d}_i)] &\geq 1 - \xi_n
\end{align*}
\]

• Loss Bound:

\[ \forall w: \text{rank}(d, \text{sort}(w \cdot \phi(x, d))) \leq \sum_i \xi_i + 1 \]

[Herbrich at al., 1999] [Joachims et al., 2002] [Joachims et al., 2017]
Position-Based Propensity Model

• Model:

\[ P(c_i = 1|r_i, \text{rank}(i|\bar{y})) = q_{\text{rank}}(i|\bar{y}) \cdot [r_i = 1] \]

• Assumptions
  – Examination only depends on rank
  – Click reveals relevance if rank is examined

<table>
<thead>
<tr>
<th>Presented ( \bar{y} )</th>
<th>( Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( q_1 )</td>
</tr>
<tr>
<td>B</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>C</td>
<td>( q_3 )</td>
</tr>
<tr>
<td>D</td>
<td>( q_4 )</td>
</tr>
<tr>
<td>E</td>
<td>( q_5 )</td>
</tr>
<tr>
<td>F</td>
<td>( q_6 )</td>
</tr>
<tr>
<td>G</td>
<td>( q_7 )</td>
</tr>
</tbody>
</table>

[Richardson et al., 2007] [Chuklin et al., 2015] [Wang et al., 2016]
Experiments

• Yahoo Web Search Dataset
  – Full-information dataset
  – Binarized relevance labels
• Generate synthetic click data based on
  – Position-based propensity model
    with \( q_r = \left( \frac{1}{r} \right)^\eta \)
  – Baseline “deployed” ranker to generate \( \bar{y} \)
  – 33% noisy clicks on irrelevant docs

<table>
<thead>
<tr>
<th>Presented ( \bar{y} )</th>
<th>( Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( q_1 )</td>
</tr>
<tr>
<td>B</td>
<td>( q_2 )</td>
</tr>
<tr>
<td>\text{Click}</td>
<td>( q_3 )</td>
</tr>
<tr>
<td>D</td>
<td>( q_4 )</td>
</tr>
<tr>
<td>\text{Click}</td>
<td>( q_5 )</td>
</tr>
<tr>
<td>E</td>
<td>( q_6 )</td>
</tr>
<tr>
<td>G</td>
<td>( q_7 )</td>
</tr>
</tbody>
</table>
Scaling with Training Set Size

[Joachims et al., 2017]
Scaling with Training Set Size

[Joachims et al., 2017]
Severity of Presentation Bias

$q_r = \left(\frac{1}{r}\right)^\eta$

[Joachims et al., 2017]
Increasing Click Noise

[Joachims et al., 2017]
Misspecified Propensities

\[ q_r = \left( \frac{1}{r} \right)^\eta \]

[Joachims et al., 2017]
• Learning-to-Rank from User Interactions
  – Find new ranking policy \( \pi \) that selects \( y \) with better \( \delta \)
• Batch Learning-to-Rank from Partial Labels
  – Learning from partial and biased feedback
  – Learning Principle: Unbiased Partial-Information ERM
  – Learning Algorithm: Propensity SVM-Rank
• Propensity Estimation for Ranking
  – Control for relevance through position randomization
Position-Based Propensity Model

• Model:

\[ P(c_i = 1|r_i, \text{rank}(i|\bar{y})) = q_{\text{rank}}(i|\bar{y}) \cdot [r_i = 1] \]

• Assumptions
  – Examination only depends on rank
  – Click reveals relevance if rank is examined

[Richardson et al., 2007] [Chuklin et al., 2015] [Wang et al., 2016] [Joachims et al., 2017]
Examination Curve from Eyetracking

[Granka et al., 2007]
Estimating the Propensities

- Idea: Randomization to control for relevance
  → Swap Interventions

\[ E(c_1|T1) = q_1 \cdot E(r_1 = 1|\text{rank}(d|\bar{y}) = 1) \]

\[ E(c_3|T2) = q_3 \cdot E(r_1 = 1|\text{rank}(d|\bar{y}) = 1) \]

\[ \frac{q_1}{q_k} = \frac{E(c_1|T1)}{E(c_k|T2)} \]

[Wang et al., 2016; Joachims et al., 2017]
Real-World Experiment

- Arxiv Full-Text Search
  - Run Swap(1,r) experiment to estimate $q_r$
  - Collect training clicks using production ranker
  - Train naïve / propensity SVM-Rank (1000 features)
  - A/B tests via interleaving

<table>
<thead>
<tr>
<th>Interleaving Experiment</th>
<th>Propensity SVM-Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>wins</td>
</tr>
<tr>
<td>against Prod</td>
<td>87</td>
</tr>
<tr>
<td>against Naive SVM-Rank</td>
<td>95</td>
</tr>
</tbody>
</table>
Conclusions and Discussion

• Learning to Rank from User Interactions
• Batch Learning-to-Rank from Partial Labels
  – Find new ranker $\pi$ that selects $y$ with improved rank metric
  – Positive-only feedback on subset of items
  – Correct for biased feedback due to bias in user exposure
  – Estimate propensities by controlling for relevance through swap interventions
• What is still missing?
  – Improve on simplistic propensity model
  – How to deal with zero propensities
  – Biases that do not work through exposure (e.g. Trust Bias)
  – Other learning algorithms and ranking metrics
  – Etc.