Optimizing Search Engines using Clickthrough Data

By Thorsten Joachims

R222 Presentation by Kaitlin Cunningham
23 January 2017
What is clickthrough data?

- Triplet: (q, r, c)

- Premise: set c conveys some information about user preferences
Pairwise preferences

\[ link_3 <_{\tau^*} link_2 \quad link_7 <_{\tau^*} link_2 \quad (1) \]
\[ link_7 <_{\tau^*} link_4 \]
\[ link_7 <_{\tau^*} link_5 \]
\[ link_7 <_{\tau^*} link_6 \]

Algorithm 1. (Extracting Preference Feedback from Clickthrough)

For a ranking \((link_1, link_2, link_3, \ldots)\) and a set \(C\) containing the ranks of the clicked-on links, extract a preference example

\[ link_i <_{\tau^*} link_j \]

for all pairs \(1 \leq j < i\), with \(i \in C\) and \(j \notin C\).
A new learning algorithm

- Optimal (target) ranking \( r^* \) v. system ranking \( r_{f(q)} \)

- Kendall’s \( \tau \):

\[
\tau(r_a, r_b) = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{\binom{m}{2}}
\]  

- Expected Kendall’s \( \tau \):

\[
\tau_P(f) = \int \tau(r_{f(q)}, r^*) d\Pr(q, r^*)
\]
SVM algorithm for learning

- Empirical risk minimization approach:

\[
\tau_S(f) = \frac{1}{n} \sum_{i=1}^{n} \tau(f(q_i), r_i^*).
\]

\[
\forall (d_i, d_j) \in r_1^* : \bar{w} \Phi(q_1, d_i) > \bar{w} \Phi(q_1, d_j)
\]

\[
\ldots
\]

\[
\forall (d_i, d_j) \in r_n^* : \bar{w} \Phi(q_n, d_i) > \bar{w} \Phi(q_n, d_j)
\]

**Optimization Problem 1. (Ranking SVM)**

minimize: \[
V(\bar{w}, \bar{\xi}) = \frac{1}{2} \bar{w} \cdot \bar{w} + C \sum \xi_{i,j,k}
\]

subject to:

\[
\forall (d_i, d_j) \in r_1^* : \bar{w} \Phi(q_1, d_i) \geq \bar{w} \Phi(q_1, d_j) + 1 - \xi_{i,j,1}
\]

\[
\ldots
\]

\[
\forall (d_i, d_j) \in r_n^* : \bar{w} \Phi(q_n, d_i) \geq \bar{w} \Phi(q_n, d_j) + 1 - \xi_{i,j,n}
\]

\[
\forall i \forall j \forall k : \xi_{i,j,k} \geq 0
\]
Using partial feedback

- Replace \( r^* \) with \( r' \):

\[
\text{OPTIMIZATION PROBLEM 2. (RANKING SVM (PARTIAL))}
\]

\[
\text{minimize: } V(\bar{w}, \bar{\xi}) = \frac{1}{2} \bar{w} \cdot \bar{w} + C \sum \xi_{i,j,k} \quad (21)
\]

\[
\text{subject to:}
\]

\[
\forall (d_i, d_j) \in r'_1 : \bar{w} \Phi(q_1, d_i) > \bar{w} \Phi(q_1, d_j) + 1 - \xi_{i,j,1} \\
\ldots
\]

\[
\forall (d_i, d_j) \in r'_n : \bar{w} \Phi(q_n, d_i) > \bar{w} \Phi(q_n, d_j) + 1 - \xi_{i,j,n} \\
\forall i \forall j \forall k : \xi_{i,j,k} \geq 0 \quad (23)
\]
Experiment: Offline

- Training set: 112 queries over one month to Google and MSNSearch through “Striver”

- Feature mapping $\Phi(q, d)$:
  - 38 rank-based features
  - 3 query/content features
  - ~20 000 popularity attribute features

- Extracted pairwise preferences using Alg 1

- 50 constraints added
Offline: Results

- Test error decreases to ~10% with 80 training queries (out of 112)

✓ Proof of concept

Figure 4: Generalization error of the Ranking SVM depending on the size of the training set. The error bars show one standard error.
Experiment: Interactive Online

- Training set: 260 queries from 20 users over less than a month
- Evaluation period of ~2 weeks
- Compared against Google, MSNSearch and Toprank
Interactive Online: Results

<table>
<thead>
<tr>
<th>Comparison</th>
<th>more clicks on learned</th>
<th>less clicks on learned</th>
<th>tie (with clicks)</th>
<th>no clicks</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned vs. Google</td>
<td>29</td>
<td>13</td>
<td>27</td>
<td>19</td>
<td>88</td>
</tr>
<tr>
<td>Learned vs. MSNSearch</td>
<td>18</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>40</td>
</tr>
<tr>
<td>Learned vs. Toprank</td>
<td>21</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>52</td>
</tr>
</tbody>
</table>

Table 2: Pairwise comparison of the learned retrieval function with Google, MSNSearch, and the non-learning meta-search ranking. The counts indicate for how many queries a user clicked on more links from the top of the ranking returned by the respective retrieval function.

- Users clicked on more links from the learned retrieval function than the other search engines
- Learned function improves retrieval
Discussion

- Personalised retrieval functions which can be tailored to small homogenous groups or individual users

- Function doesn’t rely on explicit relevance judgements

- Question: What are the computational demands of training using clickthrough data?
Critique

- Theory well-placed in context of other measures and research
- Well-reasoned explanations throughout
Critique

- Little to no discussion about the constraints

- No discussion about the relevance/influence of the tied clicks or no clicks in the online experiment

- Experiments based on homogenous user base:
  - How diverse were the queries in the training and testing periods

- Hypothesise the effect of scaling up the number of queries