Learning Ranking Functions with SVMs

CS4780/5780 – Machine Learning
Fall 2013
Thorsten Joachims
Cornell University

http://www.cs.cornell.edu/People/tj/publications/joachims_02c.pdf

Adaptive Search Engines

- Traditional Search Engines
  - One-size-fits-all
  - Hand-tuned retrieval function
- Hypothesis
  - Different users need different retrieval functions
  - Different collections need different retrieval functions
- Machine Learning
  - Learn improved retrieval functions
  - User Feedback as training data

Overview

- How can we get training data for learning improved retrieval functions?
  - Explicit vs. implicit feedback
  - Absolute vs. relative feedback
  - User study with eye-tracking and relevance judgments

- What learning algorithms can use this training data?
  - Ranking Support Vector Machine
  - User study with meta-search engine

Final Course Projects

- Now
  - Start thinking of project ideas, anything relevant to the course goes
  - Start recruiting team members
- Oct 22
  - Submit project proposal as group of 3-4 students
- Oct 24
  - Submit peer feedback for proposals
- Nov 21
  - Submit status report
- Dec 5
  - Project poster presentations (evening)
- Dec 11
  - Submit final project report
- Dec 18
  - Submit peer reviews of reports

Sources of Feedback

- Explicit Feedback
  - Overhead for user
  - Only few users give feedback
  - => not representative
- Implicit Feedback
  - Queries, clicks, time, mousing, scrolling, etc.
  - No Overhead
  - More difficult to interpret

Feedback from Clickthrough Data

Relative Feedback:
Clicks reflect preference between observed links.

(3 < 2), (7 < 2), (7 < 4), (7 < 5), (7 < 6)

Absolute Feedback:
The clicked links are relevant to the query.

Rel(1), NotRel(2), Rel(3), NotRel(4), NotRel(5), NotRel(6), Rel(7)
### User Study: Eye-Tracking and Relevance

- **Scenario**
  - WWW search
  - Google search engine
  - Subjects were not restricted
  - Answer 10 questions
- **Eye-Tracking**
  - Record the sequence of eye movements
  - Analyze how users scan the results page of Google
- **Relevance Judgments**
  - Ask relevance judges to explicitly judge the relevance of all pages encountered
  - Compare implicit feedback from clicks to explicit judgments

### What is Eye-Tracking?

- **Device** to 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

### How Many Links do Users View?

- **Total number of abstracts viewed per page**
- **Mean**: 3.07  **Median/Mode**: 2.00

### In Which Order are the Results Viewed?

- **Instance of arrival to each result**
- **Mean fixation value of arrival**
- **Rank of result**
- **# times result selected**
- **Time spent in abstract**

- Users tend to read the results in order

### Looking vs. Clicking

- **# times result selected**
- **Time spent in abstract**

- Users view links one and two more thoroughly / often
- Users click most frequently on link one

### Do Users Look Below the Clicked Link?

- **Viewed Rank**
- **Clicked Rank**

- Users typically do not look at links below before they click (except maybe the next link)
How do Clicks Relate to Relevance?

- Experiment (Phase II)
  - Additional 16 subjects
  - Manually judged relevance
    - Abstract
    - Page
- Manipulated Rankings
  - Normal: Google’s ordering
  - Swapped: Top Two Swapped
  - Reversed: Ranking reversed
- Experiment Setup
  - Same as Phase I
  - Manipulations not detectable

Presentation Bias

Hypothesis: Order of presentation influences where users look, but not where they click!

<table>
<thead>
<tr>
<th></th>
<th>l₁, l₂</th>
<th>l₃, l₄</th>
<th>l₅, l₆</th>
<th>l₇, l₈</th>
<th>l₉, l₁₀</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;normal&quot;</td>
<td>15</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>36</td>
</tr>
<tr>
<td>rel(l₁) &gt; rel(l₂)</td>
<td>16</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>rel(l₁) &lt; rel(l₂)</td>
<td>19</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>29</td>
<td>85</td>
</tr>
<tr>
<td>rel(l₁) = rel(l₂)</td>
<td>45</td>
<td>33</td>
<td>4</td>
<td>3</td>
<td>8</td>
<td>85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>l₁, l₂</th>
<th>l₃, l₄</th>
<th>l₅, l₆</th>
<th>l₇, l₈</th>
<th>l₉, l₁₀</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;swapped&quot;</td>
<td>11</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>28</td>
</tr>
<tr>
<td>rel(l₁) &gt; rel(l₂)</td>
<td>17</td>
<td>10</td>
<td>7</td>
<td>2</td>
<td>36</td>
<td>50</td>
</tr>
<tr>
<td>rel(l₁) &lt; rel(l₂)</td>
<td>36</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>50</td>
<td>114</td>
</tr>
</tbody>
</table>

=> All but “Click > Earlier Click” appear accurate

Quality-of-Context Bias

Hypothesis: Clicking depends only on the link itself, but not on other links.

<table>
<thead>
<tr>
<th></th>
<th>Rank of clicked link as sorted by relevance judges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal + Swapped</td>
<td>2.67</td>
</tr>
<tr>
<td>Reversed</td>
<td>3.27</td>
</tr>
</tbody>
</table>

=> Users click on less relevant links, if they are embedded between irrelevant links.

Are Clicks Absolute Relevance Judgments?

- Clicks depend not only on relevance of a link, but also
  - On the position in which the link was presented
  - The quality of the other links

=> Interpreting Clicks as absolute feedback extremely difficult!

Strategies for Generating Relative Feedback

- "Click > Skip Above"
  - (3>2), (5>2), (5>4)
- "Last Click > Skip Above"
  - (5>2), (5>4)
- "Click > Earlier Click"
  - (3>1), (5>1), (5>3)
- "Click > Skip Previous"
  - (3>2), (5>4)
- "Click > Skip Next"
  - (1>2), (3>4), (5>6)

Comparison with Explicit Feedback

<table>
<thead>
<tr>
<th>Explicit Feedback Data Strategy</th>
<th>Abstracts Phase I &quot;normal&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-Judge Agreement</td>
<td>89.5</td>
</tr>
<tr>
<td>Click &gt; Skip Above</td>
<td>80.8 ± 3.6</td>
</tr>
<tr>
<td>Last Click &gt; Skip Above</td>
<td>83.1 ± 3.8</td>
</tr>
<tr>
<td>Click &gt; Earlier Click</td>
<td>67.2 ± 12.3</td>
</tr>
<tr>
<td>Click &gt; Skip Previous</td>
<td>82.3 ± 7.3</td>
</tr>
<tr>
<td>Click &gt; No Click Next</td>
<td>84.1 ± 4.9</td>
</tr>
</tbody>
</table>

=> All but “Click > Earlier Click” appear accurate
Is Relative Feedback Affected by Bias?

<table>
<thead>
<tr>
<th>Explicit Feedback Data Strategy</th>
<th>Abstracts Phase II</th>
<th>&quot;normal&quot;</th>
<th>&quot;swapped&quot;</th>
<th>&quot;reversed&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click &gt; Skip Above</td>
<td>88.0 ± 9.5</td>
<td>79.6 ± 8.9</td>
<td>83.0 ± 6.7</td>
<td></td>
</tr>
<tr>
<td>Last Click &gt; Skip Above</td>
<td>89.7 ± 9.8</td>
<td>77.9 ± 9.9</td>
<td>84.6 ± 6.9</td>
<td></td>
</tr>
<tr>
<td>Click &gt; Earlier Click</td>
<td>75.0 ± 25.8</td>
<td>36.8 ± 22.9</td>
<td>28.6 ± 27.5</td>
<td></td>
</tr>
<tr>
<td>Click &gt; Skip Previous</td>
<td>88.9 ± 24.1</td>
<td>80.0 ± 18.0</td>
<td>79.5 ± 15.4</td>
<td></td>
</tr>
<tr>
<td>Click &gt; No Click Next</td>
<td>75.6 ± 14.5</td>
<td>66.7 ± 13.1</td>
<td>70.0 ± 15.7</td>
<td></td>
</tr>
</tbody>
</table>

⇒ Significantly better than random in all conditions, except “Click > Earlier Click”

How Well Do Users Judge Relevance Based on Abstract?

<table>
<thead>
<tr>
<th>Explicit Feedback Data Strategy</th>
<th>Abstracts Phase II</th>
<th>Pages all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-Judge Agreement</td>
<td>82.5</td>
<td>86.4</td>
</tr>
<tr>
<td>Click &gt; Skip Above</td>
<td>83.1 ± 4.4</td>
<td>78.2 ± 5.6</td>
</tr>
<tr>
<td>Last Click &gt; Skip Above</td>
<td>83.8 ± 4.6</td>
<td>80.9 ± 5.1</td>
</tr>
<tr>
<td>Click &gt; Earlier Click</td>
<td>46.9 ± 13.9</td>
<td>64.3 ± 15.4</td>
</tr>
<tr>
<td>Click &gt; Skip Previous</td>
<td>81.6 ± 9.5</td>
<td>80.7 ± 9.6</td>
</tr>
<tr>
<td>Click &gt; No Click Next</td>
<td>70.4 ± 8.0</td>
<td>67.4 ± 8.2</td>
</tr>
</tbody>
</table>

⇒ clicks based on abstracts reflect relevance of the page well

Learning Retrieval Functions from Pairwise Preferences

- Idea: Learn a ranking function, so that number of violated pair-wise training preferences is minimized.
- Form of Ranking Function: sort by
  \[ U(q,d_i) = w_1 \# \text{(of query words in title of } d_i) + w_2 \# \text{(of query words in anchor)} + \ldots + w_n \text{(page-rank of } d_i) = w \cdot \Phi(q,d_i) \]
- Training: Select \( w \) so that if user prefers \( d_i \) to \( d_j \) for query \( q \), then \( U(q,d_i) > U(q,d_j) \)

Ranking Support Vector Machine

- Find ranking function with low error and large margin
  \[
  \min_w \frac{1}{2} \|w\|^2 + C \sum_i \xi_i
  \text{ s.t. } \Phi(q,d_i) \cdot w \geq U(q,d_i) + 1 - \xi_i
  \ldots
  \Phi(q,d_j) \cdot w \geq U(q,d_j) + 1 - \xi_j
  \]
- Properties
  - Convex quadratic program
  - Non-linear functions using Kernels
  - Implemented as part of SVM-light
  - http://svmlight.joachims.org

Experiment

- Meta-Search Engine “Striver”
  - Implemented meta-search engine on top of Google, MSNSearch, Altavista, Hotbot, Excite
  - Retrieve top 100 results from each search engine
  - Re-rank results with learned ranking functions
- Experiment Setup
  - User study on group of ~20 German machine learning researchers and students
  - ⇒ homogeneous group of users
  - Asked users to use the system like any other search engine
  - Train ranking SVM on 3 weeks of clickthrough data
  - Test on 2 following weeks

Which Ranking Function is Better?

Balanced Interleaving

\[ f_1(u,q) \rightarrow r_1 \]

\[ f_2(u,q) \rightarrow r_2 \]

Interleaving \( r_1, r_2 \)

Model of User: Better retrieval functions is more likely to get more clicks.

Invariant: For all \( k \), top \( k \) of balanced interleaving is union of top \( k \) of \( r_1 \) and top \( k \) of \( r_2 \) with \( r_1, r_2 \) ± 1

Interpretation: \( r_1 > r_2 \) \( \iff \) clicks(top(\( r_1 \))) > clicks(top(\( r_2 \)))
Results

<table>
<thead>
<tr>
<th>Ranking A</th>
<th>Ranking B</th>
<th>A better</th>
<th>B better</th>
<th>Tie</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned</td>
<td>Google</td>
<td>29</td>
<td>13</td>
<td>27</td>
<td>69</td>
</tr>
<tr>
<td>Learned</td>
<td>MSNSearch</td>
<td>18</td>
<td>4</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>Learned</td>
<td>Toprank</td>
<td>21</td>
<td>9</td>
<td>11</td>
<td>41</td>
</tr>
</tbody>
</table>

Result:
- Learned > Google
- Learned > MSNSearch
- Learned > Toprank

Toprank: rank by increasing minimum rank over all 5 search engines

Learned Weights

- Weight  Feature
- 0.60    cosine between query and abstract
- 0.48    ranked in top 10 from Google
- 0.24    cosine between query and the words in the URL
- 0.24    doc ranked at rank 1 by exactly one of the 5 engines
- 0.22    host has the name “citeseer”
- 0.17    country code of URL is “.de”
- 0.16    ranked top 1 by HotBot
- 0.15    country code of URL is “.fi”
- 0.17    length of URL in characters
- 0.32    not ranked in top 10 by any of the 5 search engines
- 0.38    not ranked top 1 by any of the 5 search engines

Conclusions

- Clickthrough data can provide accurate feedback
  - Clickthrough provides relative instead of absolute judgments
- Ranking SVM can learn effectively from relative preferences
  - Improved retrieval through personalization in meta search
- Current and future work
  - Exploiting query chains
  - Other implicit feedback signals
  - Adapting intranet search for ArXiv.org
  - Recommendation
  - Robustness to “click-spam”
  - Learning and micro-economic theory for interactive learning with preference
  - Further user studies to get better models of user behavior

Feedback across Query Chains