Improving Quality of Training Data for Learning to Rank Using Click-Through Data

Jingfang Xu, Chuanliang Chen, Gu Xu, Hang Li, Elbio Abib
Learning to Rank

• Relevance label:
  – Assigned by human judges
  – Prone to contain errors

• This work: training data quality
Training Data Quality

Talk Outline

How Training Data Quality Affects Learning to Rank

Label Prediction Using Click-through Data

Improving Training Data Quality Using Click-through Data
How Training Data Quality Affects Learning to Rank
Simulation on LETOR Datasets

Training data

- \(<Q1, U1, Relevant>\>
- \(<Q2, U2, Irrelevant>\>
- \(<Q3, U3, Relevant>\>
- \(<Q4, U4, Irrelevant>\>
- \(<Q5, U5, Relevant>\>

Error rate: 20%

Test data

- \(<Q6, U6, Irrelevant>\>
- \(<Q7, U7, Relevant>\>
- \(<Q8, U8, Irrelevant>\>

Learning with high quality data

Learning with noisy data

Performance comparison

- \(<Q4, U4, Relevant>\>
Performance Degrades When Error Rate Increases

Relative decrease in MAP

<table>
<thead>
<tr>
<th>Error Rate</th>
<th>TD2004</th>
<th>HP2004</th>
<th>NP2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>12%</td>
<td>36%</td>
<td>29%</td>
</tr>
<tr>
<td>30%</td>
<td>46%</td>
<td>70%</td>
<td>74%</td>
</tr>
</tbody>
</table>
Similar Results on Other Algorithms

RankBoost

RankSVM

AdaRank

SVMMAP
Label Prediction Using Click-Through Data
Relevance Label Prediction

Query

Search Engine

Document

User click

N occurrences

Click distribution

Click-through Pattern

Position

Relevance Label
Relevance Label Prediction (cont’)

\[
Pr(y|x) = p((y_1, y_2, \ldots, y_n) \mid (x_1, x_2, \ldots, x_n))
\]

\(x\): click-through pattern

\(y\): relevance label

Supervised Learning Problem (Carterette & Jones 2007)
Two Dependency Models

Sequential dependency model

- Dependency between labels of adjacent document pairs

Full dependency model

- Dependency between labels of any document pairs
Two Dependency Models (cont’)

Sequential dependency Model

\[
Pr_\theta(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda^i_k f_k(y_{i-1}, y_i, x) + \sum_{i,k} \mu^i_k g_k(y_i, x)\right)
\]

\[
Z(x) = \sum_y \exp\left(\sum_{i,k} \lambda^i_k f_k(y_{i-1}, y_i, x) + \sum_{i,k} \mu^i_k g_k(y_i, x)\right)
\]

Full dependency Model

\[
Pr_\theta(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,j,k} \lambda^{i,j}_k f_k(y_i, y_j, x) + \sum_{i,k} \mu^i_k g_k(y_i, x)\right)
\]

\[
Z(x) = \sum_y \exp\left(\sum_{i,j,k} \lambda^{i,j}_k f_k(y_i, y_j, x) + \sum_{i,k} \mu^i_k g_k(y_i, x)\right)
\]
Learning

Learning = Maximum Likelihood Estimation

$$\theta^* = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} \sum_{m=1}^{M} \log(\text{Pr}_\theta(y^m|x^m))$$

Sequential dependency model

- Dynamic programming (L-BFGS)

Full dependency model

- Solution space is huge and thus calculation of $Z(x)$ is difficult
- Approximate with Gibbs Sampling
Prediction

Find most likely label sequence

\[ y^* = \arg \max_y \Pr_{\theta^*}(y|x) \]

- Sequential dependency model
  - Viterbi algorithm
- Full dependency model
  - Solution space is huge
  - Quadratic programming relaxation method (Ravikumar & Lafferty, 2006)
# Major Features

<table>
<thead>
<tr>
<th><strong>Vertex Features</strong></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClickthroughRate ((r_1, r_2))</td>
<td>Whether clickthrough rate of document is in range of ([r_1, r_2])</td>
</tr>
<tr>
<td>DwellTime ((t_1, t_2))</td>
<td>Whether time users spend on document is in range of ([t_1, t_2])</td>
</tr>
<tr>
<td>LastClick ((p_1, p_2))</td>
<td>Whether probability of document’s being the last click of session is in range of ([p_1, p_2])</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Edge Features</strong></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClickthroughRateDiff ((r_1, r_2))</td>
<td>Whether the diff between clickthrough rates of two documents is in range of ([r_1, r_2])</td>
</tr>
<tr>
<td>DwellTimeDiff ((t_1, t_2))</td>
<td>Whether the diff between times users spends on two documents is in range of ([t_1, t_2])</td>
</tr>
<tr>
<td>LastClickDiff ((p_1, p_2))</td>
<td>Whether the diff between probabilities of two documents’ being the last click of a session is in range of ([p_1, p_2])</td>
</tr>
</tbody>
</table>
Experiment on Label Prediction

Data Set

- Search log of a commercial search engine in Oct. 2008
- 1500 queries, 141 million impressions and 129 million clicks
- Query-document pairs judged by 3 well-trained judges
- 900 queries for training, 600 queries for testing

Baseline method

- Non-dependency model (Carterette & Jones 2007)
  \[ \Pr(y|x) = p(y_1|x_1)p(y_2|x_2)\ldots p(y_n|x_n) \]

Evaluation measure

- Correlation between predicted labels and human labels
Experimental Result on Label Prediction

Comparison between Three Methods

<table>
<thead>
<tr>
<th>Model</th>
<th>Correlation</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDM</td>
<td>0.64</td>
<td>-</td>
</tr>
<tr>
<td>SDM</td>
<td>0.69</td>
<td>+7.8%*</td>
</tr>
<tr>
<td>FDM</td>
<td>0.74</td>
<td>+15.6%*</td>
</tr>
</tbody>
</table>

NDM: non-dependency model
SDM: sequential dependency model
FDM: full dependency model

- SDM and FDM outperform NDM
  - Considering conditional dependency is necessary

- FDM outperforms SDM
  - Increasing scope of dependency is necessary
Improving Training Data Quality Using Click-through Data
Labeling Error Creation

Judgment Error

- Random error
  - Caused by careless miss
  - Equally change to other labels

- Real error
  - Caused by misunderstanding/low proficiency
  - More likely change to close labels
  - Estimated from Mturk (low quality judgment)

Confusion Matrix estimated from Mturk

<table>
<thead>
<tr>
<th></th>
<th>Perfect</th>
<th>Excellent</th>
<th>Good</th>
<th>Fair</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect</td>
<td>55%</td>
<td>19%</td>
<td>15%</td>
<td>9%</td>
<td>2%</td>
</tr>
<tr>
<td>Excellent</td>
<td>10%</td>
<td>26%</td>
<td>26%</td>
<td>26%</td>
<td>12%</td>
</tr>
<tr>
<td>Good</td>
<td>7%</td>
<td>11%</td>
<td>25%</td>
<td>34%</td>
<td>23%</td>
</tr>
<tr>
<td>Fair</td>
<td>4%</td>
<td>9%</td>
<td>31%</td>
<td>23%</td>
<td>33%</td>
</tr>
<tr>
<td>Bad</td>
<td>5%</td>
<td>3%</td>
<td>8%</td>
<td>20%</td>
<td>64%</td>
</tr>
</tbody>
</table>
Labeling Error Detection

Detection Method

Predict labels using click-through data
If $P(\text{predicted label} | \text{current label}) > \text{threshold}$, then detect as error

Experimental Result

- FDM and SDM outperform NDM
- FDM outperforms SDM
Experimental Results on Labeling Error Detection

RankSVM (detection precision = 0.7)

SVM-MAP (detection precision = 0.7)

RankSVM (0.8)

SVM-MAP (0.8)
Conclusion

• Labeling errors in training data significantly degrade performance of learning to rank
• Automatically predicting relevance labels using click-through data
  – Sequential dependency model
  – Full dependency model
• Error correction significantly improves performance of learning to rank
Thank you!