Learning to Rank
from heuristics to theoretic approaches

Hongning Wang
Congratulations

• Job Offer
  – Design the ranking module for Bing.com
How should I rank documents?

Answer: Rank by relevance!
Relevance ?!
The Notion of Relevance

Relevance

\[ \Delta(\text{Rep}(q), \text{Rep}(d)) \]

Similarity

\[ P(r=1|q,d) \quad r \in \{0,1\} \]

Probability of Relevance

\[ P(d \rightarrow q) \text{ or } P(q \rightarrow d) \]

Probabilistic inference

Relevance constraints

[ Fang et al. 04 ]

Div. from Randomness

(Amati & Rijsbergen 02)

Different inference system

Prob. concept space model

(Wong & Yao, 95)

Inference network model

(Turtle & Croft, 91)

Different inference system

Prob. concept space model

(Wong & Yao, 95)

Inference network model

(Turtle & Croft, 91)

Doc generation

Learn. To Rank

(Joachims 02, Berges et al. 05)

Regression Model

(Fuhr 89)

Generative Model

Query generation

LM approach

(Lafferty & Zhai, 01a)

Classical prob. Model

(Robertson & Sparck Jones, 76)

Prob. distr. model

(Wong & Yao, 89)

Vector space model

(Salton et al., 75)

Different rep & similarity

...
Relevance Estimation

• Query matching
  – Language model
  – BM25
  – Vector space cosine similarity

• Document importance
  – PageRank
  – HITS
Did I do a good job of ranking documents?

- IR evaluations metrics
  - Precision
  - MAP
  - NDCG

Documents as geometric objects: how to rank documents for full-text ...
www.michaelnielsen.org/.../documents-as-geometric-objects-how-to-...
Jul 7, 2011 – In this post I explain the basic ideas of how to rank different documents according to their relevance. The ideas used are very beautiful.

Information Retrieval: Ranking Documents
ciir.cs.umass.edu/~strohman/slides/IR-Intro-Ranking.pdf
File Format: PDF/Adobe Acrobat - View as HTML
Web features, implicit relevance indicators. • Evaluating ranking quality. • Test collections. • Quality metrics. • Training systems to rank documents better. 10 ...

Lucene.net - Lucene: How to rank documents according to the ...
stackoverflow.com/.../lucene-how-to-rank-documents-according-to-t...
1 answer - Mar 3
Top answer: This will require some work, but you can achieve this using payloads. See answers to this very similar question: How to get a better Lucene/Solr score ...

The Anatomy of a Search Engine
infoclab.stanford.edu/~backrub/google.html
We use font size relative to the rest of the document because when searching, you do not want to rank otherwise identical documents differently just because ...
Take advantage of different relevance estimator?

• Ensemble the cues
  – Linear?
    • \( \alpha_1 \times BM25 + \alpha_2 \times LM + \alpha_3 \times PageRank + \alpha_4 \times HITS \)
  – Non-linear?
    • Decision tree
      \( \{ \alpha_1 = 0.4, \alpha_2 = 0.2, \alpha_3 = 0.1 \} \rightarrow \{ M@P = 0.12, NDCG = 0.5 \} \)
      \( \{ \alpha_1 = 0.1, \alpha_2 = 0.1, \alpha_3 = 0.5 \} \rightarrow \{ M@P = 0.18, NDCG = 0.7 \} \)
What if we have thousands of features?

• Is there any way I can do better?
  – Optimizing the metrics automatically!

Where to find those tree structures?

How to determine those $\alpha$s?
Rethink the task

Given: (query, document) pairs represented by a set of relevance estimators, a.k.a., features

\[ f(q, \{d\}_{i=1}^D) \rightarrow \text{ordered } \{d\}_{i=1}^D \]

Needed: a way of combining the estimators

Criterion: optimize IR metrics

– P@k, MAP, NDCG, etc.

<table>
<thead>
<tr>
<th>DocID</th>
<th>BM25</th>
<th>LM</th>
<th>PageRank</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1.6</td>
<td>1.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>0002</td>
<td>2.7</td>
<td>1.9</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>
Machine Learning

- Input: \( \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\} \), where \( X_i \in \mathbb{R}^N \), \( Y_i \in \mathbb{R}^M \)
- Object function: \( O(Y', Y) \)
- Output: \( f(X) \rightarrow Y \), such that \( f = \text{argmax}_{f' \subset F} O(f'(X), Y) \)

Classification

\( O(Y', Y) = \delta(Y' = Y) \)

Regression

\( O(Y', Y) = -||Y' - Y|| \)

NOTE: We will only talk about supervised learning.


http://en.wikipedia.org/wiki/Regression_analysis
Learning to Rank

• General solution in optimization framework
  – Input: \( \{(q_i, d_1, y_1), (q_i, d_2, y_2), \ldots, (q_i, d_n, y_n)\} \), where \( d_n \in R^N, y_i \in \{0, \ldots, L\} \)
  – Object: \( O = \{P@k, MAP, NDCG\} \)
  – Output: \( f(q, d) \rightarrow Y \), s.t., \( f = \arg\max_{f' \subset F} O(f'(q, d), Y) \)

<table>
<thead>
<tr>
<th>DocID</th>
<th>BM25</th>
<th>LM</th>
<th>PageRank</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1.6</td>
<td>1.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>0002</td>
<td>2.7</td>
<td>1.9</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>
Challenge: how to optimize?

• Evaluation metric recap
  – Average Precision
    • \( \text{AveP} = \frac{\sum_{k=1}^{n} (P(k) \times \text{rel}(k))}{\text{number of relevant documents}} \)
  – DCG
    • \( DCG_p = \text{rel}_1 + \sum_{i=2}^{p} \frac{\text{rel}_i}{\log_2 i} \)

• Order is essential!
  – \( f \rightarrow \text{order} \rightarrow \text{metric} \)

Not continuous with respect to \( f(X) \)!
Approximating the Objects!

• Pointwise
  – Fit the relevance labels individually
• Pairwise
  – Fit the relative orders
• Listwise
  – Fit the whole order
Pointwise Learning to Rank

• Ideally perfect relevance prediction leads to perfect ranking
  – $f \rightarrow \text{score} \rightarrow \text{order} \rightarrow \text{metric}$

• Reducing ranking problem to
  – Regression
    • $O(f(Q, D), Y) = -\sum_i \|f(q_i, d_i) - y_i\|$
    • Subset Ranking using Regression, D.Cossock and T.Zhang, COLT 2006
  – (multi-)Classification
    • $O(f(Q, D), Y) = \sum_i \delta(f(q_i, d_i) = y_i)$
    • Ranking with Large Margin Principles, A. Shashua and A. Levin, NIPS 2002
Subset Ranking using Regression

D.Cossock and T.Zhang, COLT 2006

• Fit relevance labels via regression

\[ \hat{f} = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left[ \sum_{j=1}^{m} (f(x_{i,j}, S_i) - y_{i,j})^2 \right] \]

• Emphasize more on relevant documents

\[ \sum_{j=1}^{m} w(x_j, S) (f(x_j, S) - y_j)^2 + u \sup_j w'(x_j, S) (f(x_j, S) - \delta(x_j, S))^2 \]

Weights on each document

Most positive document

http://en.wikipedia.org/wiki/Regression_analysis
Ranking with Large Margin Principles
A. Shashua and A. Levin, NIPS 2002

• Goal: correctly placing the documents in the corresponding category and maximize the margin

\[
\begin{align*}
\frac{2}{|w|} & \quad \text{maximize the margin} \\
\text{Y=1} & \quad \text{Y=0} \quad \text{Y=2} \\
(w, b_1) & \quad (w, b_2) \\
\end{align*}
\]

\[
\begin{align*}
w \cdot x^j_i - b_j & \leq -1 + \epsilon^j_i, \\
w \cdot x^{j+1}_i - b_j & \geq 1 - \epsilon^{*j+1}_i, \\
\epsilon^j_i & \geq 0, \epsilon^{*j}_i \geq 0
\end{align*}
\]

Reduce the violations
Maximizing the sum of margins

\[
\begin{align*}
\sum_{j=1}^{k-1} (a_j - b_j) + C \sum_{i} \sum_{j} (\epsilon_i^j + \epsilon_i^{*j+1}) \\
\text{subject to} \\
a_j \leq b_j, \\
b_j \leq a_{j+1}, \quad j = 1, \ldots, k - 2 \\
w \cdot x_i^j \leq a_j + \epsilon_i^j, \quad b_j - \epsilon_i^{*j+1} \leq w \cdot x_i^{j+1} \\
w \cdot w \leq 1, \quad \epsilon_i^j \geq 0, \epsilon_i^{*j+1} \geq 0
\end{align*}
\]
Ranking with Large Margin Principles
A. Shashua and A. Levin, NIPS 2002

• Ranking lost is consistently decreasing with more training data
What did we learn

• Machine learning helps!
  – Derive something optimizable
  – More efficient and guided
There is always a catch

- Cannot directly optimize IR metrics
  - (0 → 1, 2 → 0) worse than (0->-2, 2->4)

- Position of documents are ignored
  - Penalty on documents at higher positions should be larger

- Favor the queries with more documents
Pairwise Learning to Rank

• Ideally perfect partial order leads to perfect ranking
  – \( f \rightarrow \text{partial order} \rightarrow \text{order} \rightarrow \text{metric} \)

• Ordinal regression
  – \( O(f(Q,D), Y) = \sum_{i \neq j} \delta(y_i > y_j)\delta(f(q_i, d_i) > f(q_i, d_i)) \)
    • Relative ordering between different documents is significant
    • E.g., \((0 \rightarrow -2, 2 \rightarrow 4)\) is better than \((0 \rightarrow 1, 2 \rightarrow 0)\)
  – Large body of work
Optimizing Search Engines using Clickthrough Data
Thorsten Joachims, KDD’02

• Minimizing the number of mis-ordered pairs

\[ y_1 > y_2, y_2 > y_3, y_1 > y_4 \]

\[ f(q, d) = w^T X_{q,d} \]

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

subject to:

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

...  

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

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

Keep the relative orders

RankingSVM
Optimizing Search Engines using Clickthrough Data
Thorsten Joachims, KDD’02

• How to use it?
  – \( f \rightarrow \text{score} \rightarrow \text{order} \)
• What did it learn from the data?
  – Linear correlations

<table>
<thead>
<tr>
<th>weight</th>
<th>feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.60</td>
<td>query_abstract_cosine</td>
</tr>
<tr>
<td>0.48</td>
<td>top10_google</td>
</tr>
<tr>
<td>0.24</td>
<td>query_url_cosine</td>
</tr>
<tr>
<td>0.24</td>
<td>top1count_1</td>
</tr>
<tr>
<td>0.24</td>
<td>top10_msnsearch</td>
</tr>
<tr>
<td>0.22</td>
<td>host_citeseer</td>
</tr>
<tr>
<td>0.21</td>
<td>domain_nec</td>
</tr>
<tr>
<td>0.19</td>
<td>top10count_3</td>
</tr>
<tr>
<td>0.17</td>
<td>top1_google</td>
</tr>
<tr>
<td>0.17</td>
<td>country_de</td>
</tr>
<tr>
<td>0.16</td>
<td>abstract_contains_home</td>
</tr>
<tr>
<td>0.16</td>
<td>top1_hotbot</td>
</tr>
<tr>
<td>0.14</td>
<td>domain_name_in_query</td>
</tr>
<tr>
<td>-0.13</td>
<td>domain_tu-bs</td>
</tr>
<tr>
<td>-0.15</td>
<td>country_fi</td>
</tr>
<tr>
<td>-0.16</td>
<td>top50count_4</td>
</tr>
<tr>
<td>-0.17</td>
<td>url_length</td>
</tr>
<tr>
<td>-0.32</td>
<td>top10count_0</td>
</tr>
<tr>
<td>-0.38</td>
<td>top1count_0</td>
</tr>
</tbody>
</table>

Positive correlated features

Negative correlated features
Optimizing Search Engines using Clickthrough Data
Thorsten Joachims, KDD’02

• How good is it?
  – Test on real system
An Efficient Boosting Algorithm for Combining Preferences

- Smooth the loss on mis-ordered pairs

\[- \sum_{y_i > y_j} Pr(d_i, d_j) \exp[f(q, d_j) - f(q, d_i)]\]
An Efficient Boosting Algorithm for Combining Preferences

• RankBoost: optimize via boosting
  – Vote by a committee

Updating $Pr(d_i, d_j)$

Credibility of each committee member (ranking feature)

$$Y_M(x) = \text{sign} \left( \sum_{m} \alpha_m y_m(x) \right)$$
An Efficient Boosting Algorithm for Combining Preferences

• How good is it?
A Regression Framework for Learning Ranking Functions Using Relative Relevance Judgments
Zheng et al. SIRIG’07

• Non-linear ensemble of features
  – Object: \( \sum_{y_i > y_j} \left( \max\{0, f(q, d_j) - f(q, d_i)\} \right)^2 \)
  – Gradient descent boosting tree
    • Boosting tree
      – Using regression tree to minimize the residuals
      – \( r^t(q, d, y) = O^t(q, d, y) - f^{(t-1)}(q, d, y) \)

BM25 > 0.5
- True
- False

LM > 0.1
- True
- False

PageRank > 0.3
- True
- False

- \( r = 1.0 \)
- \( r = 0.7 \)
- \( r = 0.4 \)
- \( r = 0.1 \)
A Regression Framework for Learning Ranking Functions Using Relative Relevance Judgments

Zheng et al. SIRIG’07

- Non-linear v.s. linear
  - Comparing with RankingSVM
Where do we get the relative orders

• Human annotations
  – Small scale, expensive to acquire

• Clickthroughs
  – Large amount, easy to acquire
Accurately Interpreting Clickthrough Data as Implicit Feedback
Thorsten Joachims, et al., SIGIR’05

• Position bias

Table 2: Percentage of times the user viewed an abstract at a particular rank before he clicked on a link at a particular rank.

<table>
<thead>
<tr>
<th>Viewed Rank</th>
<th>Clicked Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>90.6%</td>
</tr>
<tr>
<td>2</td>
<td>56.8%</td>
</tr>
<tr>
<td>3</td>
<td>30.2%</td>
</tr>
<tr>
<td>4</td>
<td>17.3%</td>
</tr>
<tr>
<td>5</td>
<td>8.6%</td>
</tr>
<tr>
<td>6</td>
<td>4.3%</td>
</tr>
</tbody>
</table>
Accurately Interpreting Clickthrough Data as Implicit Feedback
Thorsten Joachims, et al., SIGIR’05

• Controlled experiment
  – Over trust of the top ranked positions

<table>
<thead>
<tr>
<th>“normal”</th>
<th>(l^-_1, l^-_2)</th>
<th>(l^+_1, l^-_2)</th>
<th>(l^-_1, l^+_2)</th>
<th>(l^+_1, l^+_2)</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{rel}(l_1) &gt; \text{rel}(l_2))</td>
<td>15</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>36</td>
</tr>
<tr>
<td>(\text{rel}(l_1) &lt; \text{rel}(l_2))</td>
<td>11</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>(\text{rel}(l_1) = \text{rel}(l_2))</td>
<td>19</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>total</td>
<td>45</td>
<td>33</td>
<td>4</td>
<td>3</td>
<td>85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>“swapped”</th>
<th>(l^-_1, l^-_2)</th>
<th>(l^+_1, l^-_2)</th>
<th>(l^-_1, l^+_2)</th>
<th>(l^+_1, l^+_2)</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{rel}(l_1) &gt; \text{rel}(l_2))</td>
<td>11</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>28</td>
</tr>
<tr>
<td>(\text{rel}(l_1) &lt; \text{rel}(l_2))</td>
<td>17</td>
<td>10</td>
<td>7</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>(\text{rel}(l_1) = \text{rel}(l_2))</td>
<td>36</td>
<td>11</td>
<td>3</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>total</td>
<td>64</td>
<td>36</td>
<td>11</td>
<td>3</td>
<td>114</td>
</tr>
</tbody>
</table>
Accurately Interpreting Clickthrough Data as Implicit Feedback
Thorsten Joachims, et al., SIGIR’05

• Pairwise preference matters
  – Click: examined and clicked document
  – Skip: examined but non-clicked document

<table>
<thead>
<tr>
<th>Explicit Feedback Data Strategy</th>
<th>Phase I “normal”</th>
<th>“normal”</th>
<th>Phase II “swapped”</th>
<th>“reversed”</th>
<th>all</th>
<th>Pages Phase II all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-Judge Agreement</td>
<td>89.5</td>
<td>N/A</td>
<td>83.0 ± 6.7</td>
<td>83.1 ± 4.4</td>
<td>82.5</td>
<td>78.2 ± 5.6</td>
</tr>
<tr>
<td>Click &gt; Skip Above</td>
<td>80.8 ± 3.6</td>
<td>88.0 ± 9.5</td>
<td>79.6 ± 8.9</td>
<td>83.1 ± 4.4</td>
<td>82.5</td>
<td>78.2 ± 5.6</td>
</tr>
<tr>
<td>Last Click &gt; Skip Above</td>
<td>83.1 ± 3.8</td>
<td>89.7 ± 9.8</td>
<td>77.9 ± 9.9</td>
<td>83.8 ± 4.6</td>
<td>81.6</td>
<td>80.7 ± 9.6</td>
</tr>
<tr>
<td>Click &gt; Earlier Click</td>
<td>67.2 ± 12.3</td>
<td>75.0 ± 25.8</td>
<td>36.8 ± 22.9</td>
<td>79.5 ± 15.4</td>
<td>70.4</td>
<td>66.6 ± 8.2</td>
</tr>
<tr>
<td>Click &gt; Skip Previous</td>
<td>82.3 ± 7.3</td>
<td>88.9 ± 24.1</td>
<td>80.0 ± 18.0</td>
<td>79.5 ± 15.4</td>
<td>70.4</td>
<td>66.6 ± 8.2</td>
</tr>
<tr>
<td>Click &gt; No Click Next</td>
<td>84.1 ± 4.9</td>
<td>75.6 ± 14.5</td>
<td>66.7 ± 13.1</td>
<td>70.0 ± 15.7</td>
<td>70.4</td>
<td>66.6 ± 8.2</td>
</tr>
</tbody>
</table>

Click > Skip
What did we learn

• Predicting relative order
  – Getting closer to the nature of ranking

• Promising performance in practice
  – Pairwise preferences from click-throughs
Listwise Learning to Rank

• Can we directly optimize the ranking?
  – $f \rightarrow \text{order} \rightarrow \text{metric}$

• Tackle the challenge
  – Optimization without gradient
From RankNet to LambdaRank to LambdaMART: An Overview
Christopher J.C. Burges, 2010

• Minimizing mis-ordered pair => maximizing IR metrics?

Mis-ordered pairs: 6
AP: $\frac{5}{8}$
DCG: 1.333

Mis-ordered pairs: 4
AP: $\frac{5}{12}$
DCG: 0.931

Position is crucial!
From RankNet to LambdaRank to LambdaMART: An Overview
Christopher J.C. Burges, 2010

- Weight the mis-ordered pairs?
  - Some pairs are more important to be placed in the right order
  - Inject into object function

\[ y_i > y_j \Omega d_i, d_j \exp -\Delta \Phi q n, d_i, d_j \]

- Inject into gradient

\[ \lambda_{ij} = \frac{\partial O_{approx}}{\partial \Delta O} \] Gradient with respect to approximated object, i.e., exponential loss on mis-ordered pairs

Change in original object, e.g., NDCG, if we switch the documents i and j, leaving the other documents unchanged

Depend on the ranking of document i, j in the whole list

Gradient of the objective function in the 3D space.
From RankNet to LambdaRank to LambdaMART: An Overview
Christopher J.C. Burges, 2010

- Lambda functions
  - Gradient?
    - Yes, it meets the sufficient and necessary condition of being partial derivative
  - Lead to optimal solution of original problem?
    - Empirically
## From RankNet to LambdaRank to LambdaMART: An Overview

**Christopher J.C. Burges, 2010**

### Evolution

<table>
<thead>
<tr>
<th></th>
<th>RankNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td>Cross entropy over the pairs</td>
</tr>
<tr>
<td><strong>Gradient (λ function)</strong></td>
<td>Gradient of cross entropy</td>
</tr>
<tr>
<td><strong>Optimization method</strong></td>
<td>neural network</td>
</tr>
</tbody>
</table>

- As we discussed in RankBoost
- Optimize solely by gradient
- Non-linear combination
From RankNet to LambdaRank to LambdaMART: An Overview
Christopher J.C. Burges, 2010

• A Lambda tree

```xml
<tree id="8" weight="0.1">  
  <split>  
    <feature> 811 </feature>  
    <threshold> 5.0 </threshold>  
    <split pos="left">  
      <feature> 33 </feature>  
      <threshold> 20.0 </threshold>  
      <split pos="left">  
        <feature> 589 </feature>  
        <threshold> 43493.125 </threshold>  
        <split pos="left">  
          <feature> 1094 </feature>  
          <threshold> 302.73438 </threshold>  
          <split pos="left">  
            <feature> 108 </feature>  
            <threshold> 9881.824 </threshold>  
            <split pos="left">  
              <output> -0.66917753 </output>  
            </split>  
          </split>  
        </split>  
      </split>  
    </split>  
  </split>  
</tree>
```
AdaRank: a boosting algorithm for information retrieval

Jun Xu & Hang Li, SIGIR’07

- Loss defined by IR metrics
  \[ \sum_{q \in Q} Pr(q) \exp[-O(q)] \]
  
- Optimizing by boosting

Target metrics: MAP, NDCG, MRR

Credibility of each committee member (ranking feature)

\[ Y_M(x) = \text{sig}\left( \sum_{m} a_m y_m(x) \right) \]

from Pattern Recognition and Machine Learning, P658
A Support Vector Machine for Optimizing Average Precision
Yisong Yue, et al., SIGIR’07

RankingSVM
• Minimizing the pairwise loss

\[
\begin{align*}
\text{minimize:} & \quad V(\vec{w}, \vec{\xi}) = \frac{1}{2} \vec{w} \cdot \vec{w} + C \sum \xi_{i,j,k} \\
\text{subject to:} & \quad \forall (d_i, d_j) \in r_1^* : \vec{w} \Phi(q_1, d_i) \geq \vec{w} \Phi(q_1, d_j) + 1 - \xi_{i,j,1} \\
& \quad \forall (d_i, d_j) \in r_n^* : \vec{w} \Phi(q_n, d_i) \geq \vec{w} \Phi(q_n, d_j) + 1 - \xi_{i,j,n} \\
& \quad \forall i \forall j \forall k : \xi_{i,j,k} \geq 0
\end{align*}
\]

Loss defined on the number of mis-ordered document pairs

SVM-MAP
• Minimizing the structural loss

\[
\begin{align*}
\min_{\vec{w}, \xi \geq 0} & \quad \frac{1}{2} \|\vec{w}\|^2 + \frac{C}{n} \sum \xi_i \\
\text{s.t.} & \quad \forall i, \forall y \in Y \setminus y_i : \\
& \quad \vec{w}^T \Psi(x_i, y_i) \geq \vec{w}^T \Psi(x_i, y) + \Delta(y_i, y) - \xi_i
\end{align*}
\]

MAP difference

Loss defined on the quality of the whole list of ordered documents
A Support Vector Machine for Optimizing Average Precision
Yisong Yue, et al., SIGIR’07

• Max margin principle
  – Push the ground-truth far away from any mistakes you might make
  – Finding the most violated constraints
• Finding the most violated constraints
  – MAP is invariant to permutation of (ir)relevant documents
  – Maximize MAP over a series of swaps between relevant and irrelevant documents
    \[
    \arg\max_{y \in \mathcal{Y}} \Delta(y_i, y) + \mathbf{w}^T \Psi(x_i, y)
    \]
A Support Vector Machine for Optimizing Average Precision

Yisong Yue, et al., SIGIR’07

• Experiment results

<table>
<thead>
<tr>
<th>Model</th>
<th>TREC 9</th>
<th></th>
<th>TREC 10</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>W/L</td>
<td>MAP</td>
<td>W/L</td>
</tr>
<tr>
<td>SVM^Δ_{map}</td>
<td>0.290</td>
<td>–</td>
<td>0.287</td>
<td>–</td>
</tr>
<tr>
<td>SVM^Δ_{roc}</td>
<td>0.282</td>
<td>29/21</td>
<td>0.278</td>
<td>35/15 **</td>
</tr>
<tr>
<td>SVM_{acc}</td>
<td>0.213</td>
<td>49/1 **</td>
<td>0.222</td>
<td>49/1 **</td>
</tr>
<tr>
<td>SVM_{acc2}</td>
<td>0.270</td>
<td>34/16 **</td>
<td>0.261</td>
<td>42/8 **</td>
</tr>
<tr>
<td>SVM_{acc3}</td>
<td>0.133</td>
<td>50/0 **</td>
<td>0.182</td>
<td>46/4 **</td>
</tr>
<tr>
<td>SVM_{acc4}</td>
<td>0.233</td>
<td>47/3 **</td>
<td>0.238</td>
<td>46/4 **</td>
</tr>
</tbody>
</table>
Other listwise solutions

• Soften the metrics to make them differentiable
  – Michael Taylor et al., SoftRank: optimizing non-smooth rank metrics, WSDM'08

• Minimize a loss function defined on permutations
  – Zhe Cao et al., Learning to rank: from pairwise approach to listwise approach, ICML'07
What did we learn

• Taking a list of documents as a whole
  – Positions are visible for the learning algorithm
  – Directly optimizing the target metric

• Limitation
  – The search space is huge!
Summary

• Learning to rank
  – Automatic combination of ranking features for optimizing IR evaluation metrics

• Approaches
  – Pointwise
    • Fit the relevance labels individually
  – Pairwise
    • Fit the relative orders
  – Listwise
    • Fit the whole order
Experimental Comparisons

• Ranking performance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N@1</th>
<th>N@3</th>
<th>N@10</th>
<th>P@1</th>
<th>P@3</th>
<th>P@10</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>0.320</td>
<td>0.307</td>
<td>0.326</td>
<td>0.320</td>
<td>0.260</td>
<td>0.178</td>
<td>0.241</td>
</tr>
<tr>
<td>RankSVM</td>
<td>0.320</td>
<td>0.344</td>
<td>0.346</td>
<td>0.320</td>
<td>0.293</td>
<td>0.188</td>
<td>0.263</td>
</tr>
<tr>
<td>RankBoost</td>
<td>0.280</td>
<td>0.325</td>
<td>0.312</td>
<td>0.280</td>
<td>0.280</td>
<td>0.170</td>
<td>0.227</td>
</tr>
<tr>
<td>FRank</td>
<td>0.300</td>
<td>0.267</td>
<td>0.269</td>
<td>0.300</td>
<td>0.233</td>
<td>0.152</td>
<td>0.203</td>
</tr>
<tr>
<td>ListNet</td>
<td>0.400</td>
<td>0.337</td>
<td>0.348</td>
<td>0.400</td>
<td>0.293</td>
<td>0.200</td>
<td>0.275</td>
</tr>
<tr>
<td>AdaRank</td>
<td>0.260</td>
<td>0.307</td>
<td>0.306</td>
<td>0.260</td>
<td>0.260</td>
<td>0.158</td>
<td>0.228</td>
</tr>
<tr>
<td>SVM&lt;sup&gt;map&lt;/sup&gt;</td>
<td>0.320</td>
<td>0.320</td>
<td>0.328</td>
<td>0.320</td>
<td>0.253</td>
<td>0.170</td>
<td>0.245</td>
</tr>
</tbody>
</table>
Experimental Comparisons

- Winning count
  - Over seven different data sets

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N@1</th>
<th>N@3</th>
<th>N@10</th>
<th>P@1</th>
<th>P@3</th>
<th>P@10</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>RankSVM</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>RankBoost</td>
<td>18</td>
<td>22</td>
<td>22</td>
<td>17</td>
<td>22</td>
<td>23</td>
<td>19</td>
</tr>
<tr>
<td>FRank</td>
<td>18</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>17</td>
<td>23</td>
<td>15</td>
</tr>
<tr>
<td>ListNet</td>
<td>29</td>
<td>31</td>
<td>33</td>
<td>30</td>
<td>32</td>
<td>35</td>
<td>33</td>
</tr>
<tr>
<td>AdaRank</td>
<td>26</td>
<td>25</td>
<td>26</td>
<td>23</td>
<td>22</td>
<td>16</td>
<td>27</td>
</tr>
<tr>
<td>SVM$^{map}$</td>
<td>23</td>
<td>24</td>
<td>22</td>
<td>25</td>
<td>20</td>
<td>17</td>
<td>25</td>
</tr>
</tbody>
</table>
Experimental Comparisons

• My experiments
  – 1.2k queries, 45.5K documents with 1890 features
  – 800 queries for training, 400 queries for testing

<table>
<thead>
<tr>
<th>Model</th>
<th>MAP</th>
<th>P@1</th>
<th>ERR</th>
<th>MRR</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListNET</td>
<td>0.2863</td>
<td>0.2074</td>
<td>0.1661</td>
<td>0.3714</td>
<td>0.2949</td>
</tr>
<tr>
<td>LambdaMART</td>
<td>0.4644</td>
<td>0.4630</td>
<td>0.2654</td>
<td>0.6105</td>
<td>0.5236</td>
</tr>
<tr>
<td>RankNET</td>
<td>0.3005</td>
<td>0.2222</td>
<td>0.1873</td>
<td>0.3816</td>
<td>0.3386</td>
</tr>
<tr>
<td>RankBoost</td>
<td>0.4548</td>
<td>0.4370</td>
<td>0.2463</td>
<td>0.5829</td>
<td>0.4866</td>
</tr>
<tr>
<td>RankingSVM</td>
<td>0.3507</td>
<td>0.2370</td>
<td>0.1895</td>
<td>0.4154</td>
<td>0.3585</td>
</tr>
<tr>
<td>AdaRank</td>
<td>0.4321</td>
<td>0.4111</td>
<td>0.2307</td>
<td>0.5482</td>
<td>0.4421</td>
</tr>
<tr>
<td>pLogistic</td>
<td>0.4519</td>
<td>0.3926</td>
<td>0.2489</td>
<td>0.5535</td>
<td>0.4945</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.4348</td>
<td>0.3778</td>
<td>0.2410</td>
<td>0.5526</td>
<td>0.4762</td>
</tr>
</tbody>
</table>
Analysis of the Approaches

• What are they really optimizing?
  – Relation with IR metrics
Pointwise Approaches

• Regression based

\[
1 - NDCG(f) \leq \frac{1}{Z_m} \left( \frac{2 \sum_{j=1}^{m} \eta_j^e}{\sum_{j=1}^{m} (f(x_j) - y_j)^\beta} \right)^{1/\beta}
\]

Discount coefficients in DCG
Regression loss

• Classification based

\[
1 - NDCG(f) \leq \frac{15}{Z_m} \sqrt{2 \left( \sum_{j=1}^{m} \eta_j^2 - m \prod_{j=1}^{m} \eta_j^m \right)^2} \cdot \sum_{j=1}^{m} I_{\{y_j \neq f(x_j)\}}
\]

Discount coefficients in DCG
Classification loss
Pointwise Approach

- Although it seems the loss functions can bound \((1 - \text{NDCG})\), the constants before the losses seem too large.

\[
\begin{align*}
    x_i, y_i &\to Z_m \approx 21.4 & x_i, f(x_i) \\
    \left( \begin{array}{c} x_1, 4 \\ x_2, 3 \\ x_3, 2 \\ x_4, 1 \end{array} \right) &\to DCG(f) \approx 21.4 & \left( \begin{array}{c} x_1, 3 \\ x_2, 2 \\ x_3, 1 \\ x_4, 0 \end{array} \right) \\
    |1 - \text{NDCG}(f)| = 0
\end{align*}
\]

\[
\frac{15}{Z_m} \sqrt{2 \left( \sum_{j=1}^{m} \left( \frac{1}{\log(j+1)} \right)^2 - m \sum_{j=1}^{m} \left( \frac{1}{\log(j+1)} \right)^2 \right)} \cdot \sum_{j=1}^{m} I_{\{y_j \neq f(x_j)\}} \approx 1.15 > 1
\]
Pairwise Approach

• Unified loss vs. (1-NDCG)  
  Discount coefficients in DCG
  
  When \( \beta_t = \frac{G(t)\eta(t)}{Z_m} \), \( \hat{L}(f) \) is a tight bound of (1-NDCG).

• Surrogate function of Unified loss
  
  – After introducing weights \( \beta_t \), loss functions in Ranking SVM, RankBoost, RankNet are Cost-sensitive Pairwise Comparison surrogate functions, and thus are consistent with and are upper bounds of the unified loss.
  
  – Consequently, they also upper bound (1-NDCG).
Listwise Approaches

• No general analysis
  – Method dependent
  – Directness and consistency
Connection with Traditional IR

• People have foreseen this topic long time ago
  – Nicely fit in the risk minimization framework
Applying Bayesian Decision Theory

\[ (D^*, \pi^*) = \arg \min_{D, \pi} \int_{\Theta} L(D, \pi, \theta) p(\theta | q, U, C, S) d\theta \]

RISK MINIMIZATION

Bayes risk for choice \((D, \pi)\)

Loss

Choice: \((D_1, \pi_1)\)

\[ \theta_1 \]

Choice: \((D_2, \pi_2)\)

\[ \theta_2 \]

Choice: \((D_n, \pi_n)\)

\[ \theta_n \]

query \(q\)  
user \(U\)

doc set \(C\)

source \(S\)

Metric to be optimized
Available ranking features

loss
hidden
observed
Traditional Solution

- Set-based models (choose D)
- Ranking models (choose $\pi$)
  - Independent loss
    - Relevance-based loss
    - Distance-based loss
  - Dependent loss
    - MMR loss
    - MDR loss

Boolean model

Pointwise

Probabilistic relevance model
Generative Relevance Theory

Vector-space Model
Two-stage LM
KL-divergence model

Subtopic retrieval model

Pairwise/Listwise

Unsupervised!
Traditional Notion of Relevance

- Relevance
  - $\Delta(\text{Rep}(q), \text{Rep}(d))$
  - Similarity
  - $P(r=1|q,d) \quad r \in \{0,1\}$
    - Probability of Relevance
    - $P(d \rightarrow q)$ or $P(q \rightarrow d)$
      - Probabilistic inference
    - Different inference system
      - Inference network model (Turtle & Croft, 91)
      - Prob. concept space model (Wong & Yao, 95)
    - Div. from Randomness (Amati & Rijsbergen 02)
    - Relevance constraints [Fang et al. 04]
  - Different rep & similarity
    - Vector space model (Salton et al., 75)
    - Prob. distr. model (Wong & Yao, 89)
    - Regression Model (Fuhr 89)
    - Learn. To Rank (Joachims 02, Berges et al. 05)
    - Generative Model
      - Doc generation
      - Query generation
      - Classical prob. Model (Robertson & Sparck Jones, 76)
      - LM approach (Ponte & Croft, 98) (Lafferty & Zhai, 01a)
      - Inference network model (Turtle & Croft, 91)

Broader Notion of Relevance

• Traditional view
  – Content-driven
    • Vector space model
    • Probability relevance model
    • Language model

• Modern view
  – Anything related to the quality of the document
    • Clicks/views
    • Link structure
    • Visual structure
    • Social network
    • ....

Query-Document specific
Unsupervised

Query, Document, Query-Document specific
Supervised
Broader Notion of Relevance

Documents
Query
BM25
Language Model
Cosine

Click/View
Linkage structure
Visual Structure

Social network

Query relation

Query

Likes

Query

Language Model
Cosine
BM25

Documents

Query relation

Query

Language Model
Cosine
BM25

Documents

Query relation

Query

Language Model
Cosine
BM25

Documents
Future

- Tighter bounds
- Faster solution
- Larger scale
- Wider application scenario
Resources

• Books

• Helpful pages

• Packages

• Data sets
References

References

Thank you!

• Q&A
Recap of last lecture

• Goal
  – Design the ranking module for Bing.com
Basic Search Engine Architecture

Learning to Rank

• Given: (query, document) pairs represented by a set of relevance estimators, a.k.a., features

<table>
<thead>
<tr>
<th>QueryID</th>
<th>DocID</th>
<th>BM25</th>
<th>LM</th>
<th>PageRank</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0001</td>
<td>1.6</td>
<td>1.1</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>0002</td>
<td>2.7</td>
<td>1.9</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>

• Needed: a way of combining the estimators
  \[ f(q, \{d\}_{i=1}^D) \rightarrow \text{ordered } \{d\}_{i=1}^D \]

• Criterion: optimize IR metrics
  \[ P@k, \text{MAP, NDCG, etc.}\]
Challenge: how to optimize?

• Order is essential!
  \[ f \rightarrow \text{order} \rightarrow \text{metric} \]

• Evaluation metrics are not continuous and not differentiable
Approximating the Objects!

• Pointwise
  – Fit the relevance labels individually

• Pairwise
  – Fit the relative orders

• Listwise
  – Fit the whole order
Pointwise Learning to Rank

• Ideally perfect relevance prediction leads to perfect ranking
  – $f \rightarrow \text{score} \rightarrow \text{order} \rightarrow \text{metric}$

• Reduce ranking problem to
  – Regression
  – Classification

http://en.wikipedia.org/wiki/Classification
http://en.wikipedia.org/wiki/Regression_analysis
Deficiency

- Cannot directly optimize IR metrics
  - \((0 \rightarrow 1, 2 \rightarrow 0)\) worse than \((0 \rightarrow -2, 2 \rightarrow 4)\)

- Position of documents are ignored
  - Penalty on documents at higher positions should be larger

- Favor the queries with more documents
Pairwise Learning to Rank

• Ideally perfect partial order leads to perfect ranking
  – \( f \rightarrow \text{partial order} \rightarrow \text{order} \rightarrow \text{metric} \)

• Ordinal regression
  – \( O(f(Q,D),Y) = \sum_{i \neq j} \delta(y_i > y_j) \delta(f(q_i,d_i) < f(q_i,d_i)) \)
    • Relative ordering between different documents is significant
• Minimizing the number of mis-ordered pairs

\[ y_1 > y_2, y_2 > y_3, y_1 > y_4 \]

minimize:

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

subject to:

\[ \forall (q,d) \quad f(q,d) = w^T X_{q,d} \]

linear combination of features

\[ f(q,d) = w^T X_{q,d} \]

<table>
<thead>
<tr>
<th>QueryID</th>
<th>DocID</th>
<th>BM25</th>
<th>PageRank</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0001</td>
<td>1.6</td>
<td>0.9</td>
<td>4</td>
</tr>
<tr>
<td>0001</td>
<td>0002</td>
<td>2.7</td>
<td>0.2</td>
<td>2</td>
</tr>
<tr>
<td>0001</td>
<td>0003</td>
<td>1.3</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>0001</td>
<td>0004</td>
<td>1.2</td>
<td>0.7</td>
<td>0</td>
</tr>
</tbody>
</table>
Minimizing the number of mis-ordered pairs

\[
\begin{align*}
\text{minimize:} & \quad V(\vec{w}, \vec{\xi}) = \frac{1}{2} \vec{w} \cdot \vec{w} + C \sum \xi_{i,j,k} \\
\text{subject to:} & \quad \forall (d_i, d_j) \in r_1^* : \vec{w} \Phi(q_1, d_i) \geq \vec{w} \Phi(q_1, d_j) + 1 - \xi_{i,j,1} \\
& \quad \cdots \\
& \quad \forall (d_i, d_j) \in r_n^* : \vec{w} \Phi(q_n, d_i) \geq \vec{w} \Phi(q_n, d_j) + 1 - \xi_{i,j,n} \\
& \quad \forall i \forall j \forall k : \xi_{i,j,k} \geq 0 \\
& \quad w \Delta \Phi(q_n, d_i, d_j) \geq 1 - \xi_{i,j,n}
\end{align*}
\]
General Idea of Pairwise Learning to Rank

• For any pair of $y_i > y_j$