Classification

Jordan Boyd-Graber
University of Maryland
RANKING
Roadmap

- Combining rankings: taking advantage of multiple weak rankers
- Maximum margin ranking: support vector machines
- Reduction to classification: optimizing
Ranking

Given input $x_1 \ldots x_r$, return permutation of $[r]$. Permutation often parameterized by vector of scalars $y_1 \ldots y_r$.

- Web search (Google used > 200 features)
- Movie rankings
- Dating
What’s the goal? Loss functions . . .

- Kendall-$\tau$

\[
L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j 1 \left[ \text{sign}(y_i' - y_j') \neq (y_i - y_j) \right]
\]  

(1)
What’s the goal? Loss functions ...

- Kendall-\(\tau\)

\[
L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j 1[\text{sign}(y'_i - y'_j) \neq (y_i - y_j)]
\]  

- Normalized Discounted Cumulative Gain:

\[
D(i) = \frac{1}{\lg(r - i + 2)} \text{ if } i \in \{r - k + 1, \ldots, r\}
\]  

\[
G(y', y) = \sum_i D(\pi(y')_i) y_i
\]

Discount function: focus on top \(k\) elements in list
What’s the goal? Loss functions . . .

- Kendall-\(\tau\)

\[
L(y', y) = \frac{2}{r(r-1)} \sum_{i} \sum_{j} 1 \left[ \text{sign}(y'_i - y'_j) \neq (y_i - y_j) \right]
\]  \hspace{1cm} (1)

- Normalized Discounted Cumulative Gain:

\[
D(i) = \frac{1}{\lg(r - i + 2)} \text{ if } i \in \{r - k + 1, \ldots, r\}
\]  \hspace{1cm} (2)

\[
G(y', y) = \sum_{i} D(\pi(y')_i) y_i
\]  \hspace{1cm} (3)

\[
G(y', y) = \sum_{i} D(\pi(y')_i) y_i
\]  \hspace{1cm} (4)

Gain function: weight examples based on whether they are in important part of list, defined by permutation \(\pi\). For example, for \(r = 5\), the vector \(y = (2, 1, 6, 1, 0.5)\) induces the permutation \(\pi(y) = (4, 3, 5, 1, 2)\)
What’s the goal? Loss functions . . .

- Kendall-\(\tau\)

\[
L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j 1[\text{sign}(y'_i - y'_j) \neq (y_i - y_j)]
\]

- Normalized Discounted Cumulative Gain:

\[
D(i) = \frac{1}{\lg(r - i + 2)} \text{ if } i \in \{r - k + 1, \ldots, r\}
\]

\[
G(y', y) = \sum_i D(\pi(y')_i)y_i
\]

\[
L(y', y) = \sum_i \frac{1}{G(y, y)} \sum_i (D(\pi(y)_i) - D(\pi(y')_i))y_i
\]
Examples as feature vectors

Every example has a feature vector $f(x)$

<table>
<thead>
<tr>
<th>example</th>
<th>docID</th>
<th>query</th>
<th>cosine score</th>
<th>$\omega$</th>
<th>judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi_1$</td>
<td>37</td>
<td>linux operating system</td>
<td>0.032</td>
<td>3</td>
<td>relevant</td>
</tr>
<tr>
<td>$\Phi_2$</td>
<td>37</td>
<td>penguin logo</td>
<td>0.02</td>
<td>4</td>
<td>nonrelevant</td>
</tr>
<tr>
<td>$\Phi_3$</td>
<td>238</td>
<td>operating system</td>
<td>0.043</td>
<td>2</td>
<td>relevant</td>
</tr>
<tr>
<td>$\Phi_4$</td>
<td>238</td>
<td>runtime environment</td>
<td>0.004</td>
<td>2</td>
<td>nonrelevant</td>
</tr>
<tr>
<td>$\Phi_5$</td>
<td>1741</td>
<td>kernel layer</td>
<td>0.022</td>
<td>3</td>
<td>relevant</td>
</tr>
<tr>
<td>$\Phi_6$</td>
<td>2094</td>
<td>device driver</td>
<td>0.03</td>
<td>2</td>
<td>relevant</td>
</tr>
<tr>
<td>$\Phi_7$</td>
<td>3191</td>
<td>device driver</td>
<td>0.027</td>
<td>5</td>
<td>nonrelevant</td>
</tr>
</tbody>
</table>
Turning features to rank

- Have a series of “levels” or ranks $y = 1 \ldots$
- We want to find a function to separate examples

$$f(x) \equiv \langle w \cdot \phi(x) \rangle$$  \hspace{2cm} (6)
Maximizing the margin
Recap

- Ranking is an important problem
- Different objective function
- Implementation similar to regression