Learning to Top-K Search Using Pairwise Comparisons

Brian Eriksson
brian.eriksson@technicolor.com

Learning to Rank Problem Setup

Goal: Find the Top-K preferred items

Potential Applications:
- Image-Based Search
- Recommender Systems

How? We focus on pairwise comparisons:

Which item is more preferred?

\[ c_{i,j} = \begin{cases} 1 & \pi_j < \pi_i \\ 0 & \text{otherwise} \end{cases} \]

Pairwise comparison matrix, C

Limitations:
1. Inconsistency: Not all the pairwise comparisons will be correct.
   Noise Model: \[ P(c_{i,j} \neq 1 | \pi_j < \pi_i) \leq q \] Where: \( q > 0 \)

2. Incompleteness: Not all O(N^2) pairwise comparisons will be available.
   - User query limitations – e.g., paper reviews are time consuming
   - Computation time limitations – e.g., SIFT Feature Comparisons

Top-K Search using Randomly Chosen Comparisons

Consider the noise-free case (q=0) where the pairwise comparisons are observed at random with

\[ P(c_{i,j} \text{ is observed}) = p \]

Where: \( p > 0 \)

We construct the directed Sampling Comparison Graph, where each item is a vertex and there is an edge between two vertices if we observe the pairwise comparison (i.e., \( c_{i,j} = 1 \)) if we observe \( c_{j,i} = 0 \)

Complete Comparison Matrix

Incomplete Comparison Matrix

Example: Top-3 Search using an incomplete comparison matrix

Is node 5 in the top-3?

No, there exists a path of length 4.

Possibly, as there exists no path of length 3.

Theorem 1: Given N items with unknown underlying ordering, \( \pi \), and i.i.d. pairwise comparison observation probability, p, then with probability \( \geq 1 - 4 \alpha \) (where \( \alpha > 0 \)), the PathRank algorithm will only return items from the top-O(\( \log(N) \)) items, using only O(N \( \log(N) \)) adaptively-chosen pairwise comparisons.

Top-K Search using Adaptively Chosen Comparisons

Now consider potentially erroneous pairwise comparisons that can be adaptively queried.

Revised Observed Comparisons

Observed Comparisons

Known results: In the noise-free case (q=0), the Top-K search problem can be solved using a bisection search.

Unsorted rest of items

Sorted top-(K+1) items

Then, the Top-K items can be found exactly using \( \leq N \log(k+1) \) pairwise comparisons.

But, comparison errors defeat this methodology.

Experiments:

The iterative adaptive procedure is run until a subset of <50 top-ranked items are returned.

<table>
<thead>
<tr>
<th>Number of items</th>
<th>Prob. of Incorrect Comparisons (( \alpha ))</th>
<th>Fraction of Total Comparisons Used</th>
<th>Lowest Ranked Item Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>0.10</td>
<td>3.66 ( \times 10^2 )</td>
<td>36.31</td>
</tr>
<tr>
<td>1,000,000</td>
<td>0.10</td>
<td>5.53 ( \times 10^4 )</td>
<td>36.14</td>
</tr>
<tr>
<td>10,000</td>
<td>0.40</td>
<td>3.69 ( \times 10^2 )</td>
<td>117.21</td>
</tr>
<tr>
<td>1,000,000</td>
<td>0.40</td>
<td>5.56 ( \times 10^4 )</td>
<td>101.26</td>
</tr>
</tbody>
</table>

Results averaged across 100 realizations