QuickRank: a C++ Suite of Learning to Rank Algorithms

Gabriele Capannini, Domenico Dato, Claudio Lucchese, Monica Mori, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Salvatore Orlando
Introduction

• **Learning to Rank**: machine learning techniques for ranking Web documents
  - *relevance* estimation in response to a given query
  - huge collections of annotated query-documents examples

• **Aim**: to *learn* “the best” ranking function from examples to be exploited in a ranking architecture

• **State of the art**: additive ensembles of tree-based rankers [1]

Machine-learned Ranking Architectures
Machine-learned Ranking Architectures

• candidate retrieval:
  • BM25 or a first “light” machine-learned ranker:
    • recall of positive examples
    • fast but less effective

• candidate re-ranking:
  • top-K documents
  • precision!
  • thousands of trees
Features

- **Query**: query length, frequency, category, etc.
- **Document**: document length, category, n. links, etc.
- **Statistical**: # of query terms in doc, # docs containing terms, document's length.
- **Proximity**: word-wise distance between query terms.
- **Link**: Hub, Authority, PageRank, etc.
- **Spam**: link and content spam features.
- **Click**: # of click (as a measure of importance of a page);
- **Demographics**: gender, age, location, etc.
- **Session**: last issued queries, last clicked documents, click rate, etc.
QuickRank

- A suite for **efficient** and **effective** Learning to Rank
- Three tree-based Learning to Rank algorithms:
  - Gradient-Boosted Regression Trees (GBRT) [1]
  - LambdaMART (LMART) [2]
  - Oblivious-LambdaMART (OLMART) [3]

Why QuickRank?

• learning tree-based rankers is **expensive**
  
  • **learning time**: (tens of) thousands of trees
    
    • iterative process, one tree per iteration
    
    • for each node in the tree:
      
      • find best feature/value for splitting
      
      • available implementations: RankLib, JForest are slow!

• **scoring time**: (tens of) thousands of trees
QuickRank

• QuickRank allows:
  • to learn ranking models from huge training datasets
  • to easily develop new Learning to Rank algorithms
  • to fairly test and compare the efficiency and effectiveness of the learnt ranking model
QuickRank

- QuickRank is:
  - written in C++, uses OpenMP
  - designed to be **flexible** and **extensible**
    - GBRT, LMART, OLMART
    - MAP, DCG, NDCG
  - released under RPL v1.5 licence
  - suitable for research purposes
Experiments

- **Dataset**: Yahoo! Learning to Rank challenge (set 1)
  - 19,944 queries for training
  - 2,994 queries for validation
  - 6,983 queries for testing
  - 700 features per query/document pair
  - 473,134 training samples in total

http://learningtorankchallenge.yahoo.com
Experiments

• Analysis of the **learning time**
  
  • LMART, 1,000 trees
  
  • 16 leaves per tree
  
  • NDCG@10

• Platform:
  
  • 2 AMD Opteron™ 6276 (32 cores in total)
  
  • 128 GiB RAM
  
  • Ubuntu 14.04 LTS, GCC 4.9.2
Experiments

<table>
<thead>
<tr>
<th># Threads</th>
<th>100%</th>
<th>50%</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>363</td>
<td>192</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>114</td>
<td>63</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>71</td>
<td>42</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>51</td>
<td>31</td>
<td>19</td>
</tr>
<tr>
<td>32</td>
<td>41</td>
<td>25</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Threads</th>
<th>50%</th>
<th>75%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>192</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>63</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>31</td>
<td>19</td>
</tr>
<tr>
<td>32</td>
<td>25</td>
<td>16</td>
</tr>
</tbody>
</table>

- 41 minutes to learn a LMART with 1,000 trees
- RankLib [1] v2.2 takes 2.4 hours on the same platform

Conclusions

• **QuickRank** is a parallel C++ suite of Learning to Rank algorithms
  
  • efficient, flexible and easy to extend
  
  • suitable for research and industry purposes

• **Coming soon:**

  • QuickScorer: a Fast Algorithm to Rank Documents with Additive Ensembles of Regression Trees (to appear @ ACM SIGIR 2015)

  • QuickRank on the Cloud: Amazon EC2
Thank You!

quickrank@isti.cnr.it
http://quickrank.isti.cnr.it/

Questions?