Listwise Approach to Learning to Rank – Theory and Algorithm

Fen Xia*, Tie-Yan Liu
Jue Wang, Wensheng Zhang and Hang Li

Microsoft Research Asia
Chinese Academy of Sciences
Learning to Rank for Information Retrieval

Queries:
\[ q^{(1)}, q^{(m)} \]

Documents:
\[ d_1^{(1)}, 4, d_1^{(m)}, 5, d_2^{(1)}, 2, d_2^{(m)}, 3, \ldots, d_n^{(1)}, 1, d_n^{(m)}, 2 \]

Training Data
\[ q \rightarrow \{ (d_1, ?), (d_2, ?), \ldots, (d_n, ?) \} \]

Test data
\[ \{ d_{i1}, f(q, d_{i1}, w) \}, d_{i2}, f(q, d_{i2}, w), \ldots, d_{in}, f(q, d_{in}, w) \]
State-of-the-art Approaches

• Pointwise: (Ordinal) regression / classification
  – Pranking, MCRank, etc.

• Pairwise: Preference learning
  – Ranking SVM, RankBoost, RankNet, etc.

• Listwise: Taking the entire set of documents associated with a query as the learning instance.
  – Direct optimization of IR measure
    • AdaRank, SVM-MAP, SoftRank, LambdaRank, etc.

  – Listwise loss minimization
    • RankCosine, ListNet, etc.
Motivations

• The listwise approach captures the ranking problem in a conceptually more natural way and performs better than other approaches on many benchmark datasets.

• However, the listwise approach lacks of theoretical analysis.
  – Existing work focuses more on algorithm and experiments, than theoretical analysis.
  – While many existing theoretical results on regression and classification can be applied to the pointwise and pairwise approaches, the theoretical study on the listwise approach is not sufficient.
Our Work

• Take listwise loss minimization as an example, to perform theoretical analysis on the listwise approach.
  – Give a formal definition of the listwise approach.
  – Conduct theoretical analysis on listwise ranking algorithms in terms of their loss functions.
  – Propose a novel listwise ranking method with good loss function.
  – Validate the correctness of the theoretical findings through experiments.
Listwise Ranking

- Input space: $X$
  - Elements in $X$ are sets of objects to be ranked
- Output space: $Y$
  - Elements in $Y$ are permutations of objects
- Joint probability distribution: $P_{XY}$
- Hypothesis space: $H$
  - $h \in H : X \rightarrow Y$

\[
R(h) = \int_{X \times Y} l(h(x), y) dP(x, y) \quad R_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h(x^{(i)}), y^{(i)}).
\]
True Loss in Listwise Ranking

• To analyze the theoretical properties of listwise approach, the “true” loss of ranking is to be defined.
  – The true loss describes the difference between a given ranked list (permutation) and the ground truth ranked list (permutation).
• Ideally, the “true” loss should be cost-sensitive, but for simplicity, we start with the investigation of the “0-1” loss.
  – \( l(h(x), y) = \begin{cases} 
  1, & \text{if } h(x) \neq y \\
  0, & \text{if } h(x) = y,
\end{cases} \)
Surrogate Loss in Listwise Ranking

• Widely-used ranking function
  \( h(x^{(i)}) = \text{sort}(g(x_1^{(i)}), \ldots, g(x_{n_i}^{(i)})) \).

• Corresponding empirical risk
  \( R_S(g) = \frac{1}{m} \sum_{i=1}^{m} l(\text{sort}(g(x_1^{(i)}), \ldots, g(x_{n_i}^{(i)})), y^{(i)}) \)

• Challenges
  – Due to the sorting function and the 0-1 loss, the empirical loss is non-differentiable w.r.t. \( g(x) \).
  – To tackle the problem, a surrogate loss is used.

  \( R_S^\phi(g) = \frac{1}{m} \sum_{i=1}^{m} \phi(g(x^{(i)}), y^{(i)}) \)
Surrogate Listwise Loss Minimization

- RankCosine and ListNet can be well fitted into the framework of surrogate loss minimization.
  - Cosine Loss (RankCosine, IPM 2007)
    \[ \phi(g(x), y) = \frac{1}{2} \left( 1 - \frac{\psi_y(x)^T g(x)}{\|\psi_y(x)\| \|g(x)\|} \right). \]
  - Cross Entropy Loss (ListNet, ICML 2007)
    \[ \phi(g(x), y) = D(P(\pi|x; \psi_y) \| P(\pi|x; g)) \]
- A new loss function
  - Likelihood loss (ListMLE, proposed in this paper)
    \[ \phi(g(x), y) = -\log P(y|x; g) \quad P(y|x; g) = \prod_{i=1}^{n} \frac{\exp(g(x_y(i)))}{\sum_{k=i}^{n} \exp(g(x_y(k)))} \]
Analysis on Surrogate Loss

- Continuity, differentiability and convexity
- Computational efficiency
- Statistical consistency
- Soundness

These properties have been well studied in classification, but not sufficiently in ranking.
## Continuity, Differentiability, Convexity, Efficiency

<table>
<thead>
<tr>
<th>Loss</th>
<th>Continuity</th>
<th>Differentiability</th>
<th>Convexity</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosine Loss (RankCosine)</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>O(n)</td>
</tr>
<tr>
<td>Cross-entropy loss (ListNet)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(n·n!)</td>
</tr>
<tr>
<td>Likelihood loss (ListMLE)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(n)</td>
</tr>
</tbody>
</table>
When minimizing the expected surrogate loss $R_\phi(g)$ is equivalent to minimizing the expected 0-1 loss $R(h)$ (which solution is Bayes ranker $y_*$), we say the surrogate loss function is consistent.

**Theorem.** Let $\phi_y(g)$ be an order sensitive loss function on $\Omega \subset \mathbb{R}^n$. \(\forall n\) objects, if its permutation probability space is order preserving with respect to $n-1$ objective pairs $(j_1, j_2), (j_2, j_3), \ldots, (j_{n-1}, j_n)$. Then the loss $\phi_y(g)$ is consistent.

The perfect ranking of an object is inherently determined by its own. Minimum $\phi_y(g)$ is achieved when sorting $g(x)$ results in the same permutation with a given $y$. 
Statistical Consistency (3)

- It can be proven
  - Cosine Loss is statistically consistent.
  - Cross entropy loss is statistically consistent.
  - Likelihood loss is statistically consistent.
  - For detailed proof, please refer to the paper.
Soundness

• Cosine loss is not very sound
  – Suppose we have two documents $D_2 \succ D_1$. 

\[ g_1 = g_2 \]

Incorrect Ranking  Correct ranking
Soundness (2)

• Cross entropy loss is not very sound
  – Suppose we have two documents $D_2 \triangleright D_1$. 

Incorrect Ranking  Correct ranking
Soundness (3)

- Likelihood loss is sounder
  - Suppose we have two documents \( D_2 \succ D_1 \).

- Correct ranking vs Incorrect Ranking
Discussions

• All three losses can be minimized using common optimization technologies. (continuity and differentiability)
• When the number of training samples is very large, the model learning can be effective. (consistency)
• The cross entropy loss and the cosine loss are both sensitive to the mapping function. (soundness)
• The cost of minimizing the cross entropy loss is high. (complexity)
• The cosine loss is sensitive to the initial setting of its minimization. (convexity)
• The likelihood loss is the best among the three losses.
Experimental Verification

• Synthetic data
  – Different mapping function (log, sqrt, linear, quadratic, and exp)
  – Different initial setting of the gradient descent algorithm (report the mean and var of 50 runs)

• Real data
  – OHSUMED dataset in the LETOR benchmark
## Experimental Results on Synthetic Data

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListMLE</td>
<td>0.92 ± 0.011</td>
<td>0.999 ± 0.002</td>
</tr>
<tr>
<td>ListNet-log</td>
<td>0.905 ± 0.010</td>
<td>0.999 ± 0.002</td>
</tr>
<tr>
<td>ListNet-sqrt</td>
<td>0.917 ± 0.009</td>
<td>0.999 ± 0.002</td>
</tr>
<tr>
<td>ListNet-l</td>
<td>0.767 ± 0.021</td>
<td>0.995 ± 0.003</td>
</tr>
<tr>
<td>ListNet-q</td>
<td>0.868 ± 0.028</td>
<td>0.999 ± 0.002</td>
</tr>
<tr>
<td>ListNet-exp</td>
<td>0.832 ± 0.074</td>
<td>0.997 ± 0.004</td>
</tr>
<tr>
<td>RankCosine-log</td>
<td>0.180 ± 0.217</td>
<td>0.948 ± 0.034</td>
</tr>
<tr>
<td>RankCosine-sqrt</td>
<td>0.080 ± 0.159</td>
<td>0.886 ± 0.056</td>
</tr>
<tr>
<td>RankCosine-l</td>
<td>0.917 ± 0.112</td>
<td>0.999 ± 0.002</td>
</tr>
<tr>
<td>RankCosine-q</td>
<td>0.102 ± 0.161</td>
<td>0.890 ± 0.060</td>
</tr>
<tr>
<td>RankCosine-exp</td>
<td>0.047 ± 0.163</td>
<td>0.746 ± 0.136</td>
</tr>
</tbody>
</table>
Experimental Results on OHSUMED

![Graph comparing MAP and NDCG@K for ListMLE, ListNet, and RankCosine](image)

- **ListMLE**
- **ListNet**
- **RankCosine**

- **Metric**: MAP and NDCG@K (K = 1 to 10)

7/8/2008

Tie-Yan Liu @ ICML 2008
Conclusion and Future Work

• Study has been made on the listwise approach to learning to rank.
  • Likelihood loss seems to be the best listwise loss functions under investigation, according to both theoretical and empirical studies.

• Furthermore
  • In addition to consistency, rate of convergence and generalization ability should also be studied.
  • In real ranking problems, the true loss should be cost-sensitive (e.g. NDCG in Information Retrieval).
Thanks!

tyliu@microsoft.com
http://research.microsoft.com/users/tyliu/