A General Magnitude-Preserving Boosting Algorithm for Search Ranking


Microsoft Research Asia, *Tsinghua University

2009.11
Motivation

* Traditional learning to rank algorithms
  * Pairwise:
    * Ranking SVM, RankBoost, RankNet

* Deficiency:
  * e.g. Mis-ranking two documents with ratings 5 and 1 gets the same loss as mis-ranking two documents with ratings 5 and 4
Challenges for incorporating magnitude information in pairwise learning to rank

- More complex than classification:
  - Deals with the order of document list

- More complex than regression:
  - The prediction is carried out for each single document, not document pairs
Our Contribution

Basis
• Apply *magnitude-preserving labels* for pairwise learning

Algorithm
• Leverage Boosting for learning to rank: MPBoost

Theory
• Prove the convergence property on ranking accuracy
Formalization for Learning to Rank

* Input: Query set $Q$ and $\{(x_{qi}, r_{qi})\}_{i=1}^{n_q}$, for each $q \in Q$

* Learning:
  * Generate pairwise learning instances:
    $$S = \bigcup_q \{(x_{qi}, x_{qj}), y_{qij}) \mid r_{qi} \neq r_{qj}\}$$
  * Learning to generate ranking function $F(x)$

* Prediction:
  * For a retrieved document list $\{x_{qi}\}_{i=1}^{n_q}$, rank the documents from the largest $F(x_{qi})$ to the smallest one.
Pairwise Approach

* Traditional pairwise labels

\[
y_{ij} = \begin{cases} 
1, & r_i \text{ is preferenced to } r_j \\
-1, & r_j \text{ is preferenced to } r_i 
\end{cases}
\]
Magnitude Preserving Labels:

- Directed Distance Function (DDF)
  - Preserve preference relationship:
    \[ \text{sgn}(\text{dist}(r_i, r_j)) = \text{sgn}(r_i - r_j) \]
  - Preserve magnitude information
    \[ |r_i - r_j| \geq |r_i' - r_j'| \implies |\text{dist}(r_i, r_j)| \geq |\text{dist}(r_i', r_j')| \]
Directed Distance Function

Figure 1. Three ratings $r_a, r_b, r_c$, where the directed distances $\text{dist}(r_c, r_a)$ and $\text{dist}(r_a, r_b)$ are marked. The exact values of the distances can vary, but it should follow that:

1. $|\text{dist}(r_c, r_a)| > |\text{dist}(r_a, r_b)|$, to preserve magnitude of rating differences.

2. $\text{dist}(r_c, r_a) < 0$ and $\text{dist}(r_a, r_b) > 0$, to present the advantage of placing documents with higher ratings in front.
Directed Distance Function (DDF)

* The form of DDF can vary
* We investigate three kinds of DDF:
  * Linear Directed Distance (LDD)
    \[ \text{dist}(r_i, r_j) = \alpha(r_i - r_j) \]
  * Logarithmic Directed Distance (LOGDD)
    \[ \text{dist}(r_i, r_j) = \text{sgn}(r_i - r_j) \log(1 + \lambda |r_i - r_j|) \]
  * Logistic Directed Distance (LOGITDD)
    \[ \text{dist}(r_i, r_j) = \frac{1}{1 + e^{-\beta|r_i - r_j|}} \]
Directed Distance Function (DDF)

Figure 2. Curves of LDD, LOGDD and LOGITDD under different values of rating differences. The parameters are $\alpha = 0.2$, $\lambda = 3$ and $\beta = 0.5$. 
We propose the Magnitude-Preserving Boosting algorithm (MPBoost)
MPBoost Algorithm

* Loss $J$ for a ranking function $F$:

$$J(F) = \sum e^{-\text{dist}(r_i, r_j)(F(x_i) - F(x_j))}$$

* Optimization:
  * GentleBoost (Friedman et al.)
  * Stage-wise gradient descent
  * Require $|\text{dist}(r_i, r_j)| \approx 1$ ($r_i \neq r_j$)
MPBoost Algorithm

* **Input:** Query set $Q$ and $\{(x_{qi}, r_{qi})\}_{i=1}^{n_q}$, for each $q \in Q$
* **Output:** ranking function $F(x)$

1: Generate $S = \bigcup_q \{(x_{qi}, x_{qj}), \text{dist}(r_{qi}, r_{qj})\} | r_{qi} \neq r_{qj}$

2: Generate index set $I = \{(i, j) | ((x_i, x_j), \text{dist}(r_i, r_j)) \in S\}$

3: Initialize $w_{ij}^{(1)} = \frac{1}{|I|}$ for $\forall (i, j) \in I$

4: for $t = 1...T$ do

5: Fit the weak ranker $f_t$, such that: $f_t = \arg\min J_{wse}(f)$

6: Update: $w_{ij}^{(t+1)} \leftarrow w_{ij}^{(t)} e^{-\text{dist}(r_i, r_j)(f_t(x_i) - f_t(x_j))} / Z_t$

    where $Z_t = \sum_I w_{ij}^{(t)} e^{-\text{dist}(r_i, r_j)(f_t(x_i) - f_t(x_j))}$

7: end for

8: Output the final ranking function $F(x) = \sum_{t=1}^{T} f_t(x)$
Theorem 1. The normalized ranking loss (mis-ordering) of $F$ is bounded:

$$\sum_{(i, j) \in I | r_i > r_j} w_{ij}^{(1)} \left[ [F(x_i) \leq F(x_j)] \right] + \sum_{(i, j) \in I | r_i < r_j} w_{ij}^{(1)} \left[ [F(x_i) \geq F(x_j)] \right] \leq \prod_{t=1}^{T} Z_t$$

where $[\pi]$ is defined to be 1 if the condition $\pi$ holds and 0 otherwise.

Proof: Omitted
Datasets

- OHSUMED dataset (From LETOR 3.0)
- Web-1 dataset (From Yandex competition)
- Web-2 dataset (From a commercial English search engine)
- 3+1+1 five-fold cross validation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Queries</th>
<th>Documents</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHSUMED</td>
<td>106</td>
<td>16140</td>
<td>{0,1,2}</td>
</tr>
<tr>
<td>Web-1</td>
<td>9124</td>
<td>97290</td>
<td>[0,4]</td>
</tr>
<tr>
<td>Web-2</td>
<td>467</td>
<td>50000</td>
<td>{0,1,2,3,4}</td>
</tr>
</tbody>
</table>
Experiment Methodology

* Baseline:
  * RankBoost, ListNet, AdaRank-NDCG
  * MPBoost.BINARY

* Proposed method:
  * MPBoost.LDD
  * MPBoost.LOGDD
  * MPBoost.LOGITDD
Experimental Results on OHSUMED

- MPBoost.* outperforms pairwise baselines
- MPBoost.BINARY outperforms RankBoost

Figure 3. Ranking accuracies on OHSUMED dataset
Experimental Results on Web-1

Figure 4. Ranking accuracies on Web-1 dataset
Experimental Results on Web-2

Figure 5. Ranking accuracies on Web-2 dataset.
Discussion

- The magnitude-preserving versions of MPBoost outperform baselines by 0.16% to 2.2% over average NDCG

- MPBoost.LOGDD and MPBoost.LOGITDD outperform MPBoost.LDD

- Linear directed distance can hardly guarantee that for all pairs $|\text{dist}(r_i, r_j)| \approx 1$ ($r_i \neq r_j$)
Overfitting Issues

Figure 6. Average NDCG@5 on test set over 5 folds in Web-2 Dataset. For MPBoost.LDD, MPBoost.LOGDD and MPBoost.LOGITDD, the parameters $\alpha$, $\lambda$ and $\beta$ are set as the one achieving the best performance on validation set.
Conclusion

- Magnitude-preserving labels can effectively improve ranking accuracy
- Directed Distance Function can have various forms
- MPBoost inherits theoretical properties from RankBoost
Q&A

Thanks!
Related Work

* Qin et al.
  * Based on RankSVM
  * Multiple Hyperplane Ranker (MHS)
  * Complex when the number of ratings are large

* Cortes et al.
  * Based on RankSVM
  * Incorporate magnitude differences
  * Limited due to $\sigma$-admissibility of cost function
# MPBoost & RankBoost

<table>
<thead>
<tr>
<th></th>
<th>MPBoost</th>
<th>RankBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based On</td>
<td>GentleBoost</td>
<td>AdaBoost</td>
</tr>
<tr>
<td>Loss Function</td>
<td>$\sum e^{-\text{dist}(r_i, r_j)(F(x_i) - F(x_j))}$</td>
<td>$\sum_{x_0, x_i} D(x_0, x_i) \left[ [H(x_i) \leq H(x_0)] \right]$</td>
</tr>
<tr>
<td>Weak Learner’s criteria</td>
<td>$\sum w_{ij} \left[ \text{dist}(r_i, r_j) - (f(x_i) - f(x_j)) \right]^2$</td>
<td>$\sum_{x_0, x_i} D(x_0, x_i) (h(x_i) - h(x_0))$</td>
</tr>
<tr>
<td>Combination</td>
<td>$F(x) = \sum_{t=1}^{T} f_t(x)$</td>
<td>$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$</td>
</tr>
</tbody>
</table>