IntervalRank: Isotonic Regression with Listwise and Pairwise Constraints

Taesup Moon
(Joint work with Alex Smola, Yi Chang, and Zhaohui Zheng)

Yahoo! Labs

WSDM 2010
February 5
1 Backgrounds

2 IntervalRank
   - Loss function via isotonic regression
   - Efficient implementation

3 Experimental results
1 Backgrounds

2 IntervalRank
   • Loss function via isotonic regression
   • Efficient implementation

3 Experimental results
A supervised machine learning framework for ranking

- relevance labeled data

\[ \{(x_i, y_i)\}_{i=1}^n \]

- a *loss function*

\[ L\left(\{(y_i, f(x_i))\}_{i=1}^n\right) \]

- train a ranking function \( f \) via *optimizing* the loss
  - e.g., functional gradient descent
Various loss functions have been proposed

*pointwise* loss functions
- treat each example individually
- e.g., regression, etc.
Various loss functions have been proposed

**pointwise** loss functions
- treat each example individually
- e.g., regression, etc.

**pairwise** loss functions
- focus on relative orderings of pairs
- e.g., GBRank, RankNet, RankBoost, etc.
Various loss functions have been proposed

**pointwise** loss functions
- treat each example individually
- e.g., regression, etc.

**pairwise** loss functions
- focus on relative orderings of pairs
- e.g., GBRank, RankNet, RankBoost, etc.

**listwise** loss functions
- treat the whole list jointly
- e.g., LambdaRank, SoftRank, SmoothDCG, etc.
Does one approach dominate others?

Common wisdom: pointwise $\preceq$ pairwise $\preceq$ listwise

- But, listwise loss functions also have some caveats

"initial scores for training examples"
Does one approach dominate others?

Common wisdom: pointwise $\preceq$ pairwise $\preceq$ listwise

- But, listwise loss functions also have some caveats

"initial scores for training examples"

"possible scores from ranking function trained from the listwise loss"
Does one approach dominate others?

Common wisdom: pointwise $\leq$ pairwise $\leq$ listwise

- But, listwise loss functions also have some caveats

"initial scores for training examples"

"possible scores from ranking function trained from the listwise loss"

"may not assign similar scores to similarly relevant documents"
Can we mix the different approaches?

Ideally, we would like to train a function to

$$\text{Bad} \quad \text{Fair} \quad \text{Good} \quad \text{Excellent} \quad \text{Perfect}$$

ranking scores
Can we mix the different approaches?

Ideally, we would like to train a function to

- separate documents with different relevance
- cluster documents with similar relevance
1 Backgrounds

2IntervalRank
   • Loss function via isotonic regression
   • Efficient implementation

3 Experimental results
Our approach

1. Define a loss function via *isotonic regression*
2. Reformulate the problem to *efficiently* find the gradient
Loss function via isotonic regression

Loss = *minimum total efforts* to make scores satisfy the constraints

\[ \Delta_{i,j} = y_i - y_j \]

\[ \xi_g \leq f(x_i) \leq \xi_{g+1} \]

\[ | \xi_g | \leq | \xi_{g+1} | \leq | \xi_g | \]
total effort = \( \delta_1^2 + \delta_2^2 + \delta_3^2 + \delta_4^2 + \delta_5^2 \)
total effort = $\delta_1^2 + \delta_2^2 + \delta_3^2 + \delta_4^2$
Loss function via isotonic regression

Loss = "minimum" total effort

\[ L(\{(y_i, f(x_i))\}_{i=1}^{n}) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \text{ where } \delta \in \mathbb{R}^n \text{ satisfies} \]

\[ f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \text{ for all } (i, j) \in \{\text{ordered pairs}\} \]

\[ |f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{gi} \text{ for all } (i, j) \in \{\text{tied pairs}\} \]
Loss function via isotonic regression

Loss = “minimum” total effort

\[
L(\{(y_i, f(x_i))\}_{i=1}^{n}) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \quad \text{where } \delta \in \mathbb{R}^n \text{ satisfies}
\]

\[
f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \text{ for all } (i, j) \in \{\text{ordered pairs}\}
\]

\[
|f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{gi} \text{ for all } (i, j) \in \{\text{tied pairs}\}
\]

- first proposed by [Zheng et.al 2008]
Loss function via isotonic regression

Loss = "minimum" total effort

\[ L(\{(y_i, f(x_i))\}_{i=1}^n) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \quad \text{where } \delta \in \mathbb{R}^n \text{ satisfies} \]

\[ f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \text{ for all } (i, j) \in \{\text{ordered pairs}\} \]

\[ |f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{gi} \text{ for all } (i, j) \in \{\text{tied pairs}\} \]

- first proposed by [Zheng et.al 2008]
  - pairwise constraints and listwise objective
Loss = “minimum” total effort

\[ L(\{(y_i, f(x_i))\}_{i=1}^{n}) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \quad \text{where } \delta \in \mathbb{R}^n \text{ satisfies} \]

\[ f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \quad \text{for all } (i, j) \in \{\text{ordered pairs}\} \]

\[ |f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{g_i} \quad \text{for all } (i, j) \in \{\text{tied pairs}\} \]

- first proposed by [Zheng et al. 2008]
  - pairwise constraints and listwise objective
  - obtain the optimum \( \delta^* \) and use it as a functional gradient for \( f(x_i) \)
Loss function via isotonic regression

Loss = “minimum” total effort

\[
L(\{(y_i, f(x_i))\}_{i=1}^n) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \quad \text{where } \delta \in \mathbb{R}^n \text{ satisfies}
\]

\[
f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \quad \text{for all } (i, j) \in \{\text{ordered pairs}\}
\]

\[
|f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{gi} \quad \text{for all } (i, j) \in \{\text{tied pairs}\}
\]

- first proposed by [Zheng et al. 2008]
  - pairwise constraints and listwise objective
  - obtain the optimum \(\delta^*\) and use it as a functional gradient for \(f(x_i)\)

- problems:
Loss function via isotonic regression

Loss = “minimum” total effort

\[ L(\{(y_i, f(x_i))\}_{i=1}^{n}) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \quad \text{where } \delta \in \mathbb{R}^n \text{ satisfies} \]

\[ f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \quad \text{for all } (i, j) \in \{\text{ordered pairs}\} \]
\[ |f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{gi} \quad \text{for all } (i, j) \in \{\text{tied pairs}\} \]

- first proposed by [Zheng et.al 2008]
  - pairwise constraints and listwise objective
  - obtain the optimum \( \delta^* \) and use it as a functional gradient for \( f(x_i) \)

- problems:
  - no formal proof for functional gradient
Loss function via isotonic regression

Loss = “minimum” total effort

\[ L(\{(y_i, f(x_i))\}_{i=1}^n) = \min_{\delta \in \mathbb{R}^n} \|\delta\|_2^2, \quad \text{where } \delta \in \mathbb{R}^n \text{ satisfies} \]

\[ f(x_i) + \delta_i - f(x_j) - \delta_j \geq \Delta_{ij} \text{ for all } (i, j) \in \{\text{ordered pairs}\} \]

\[ |f(x_i) + \delta_i - f(x_j) - \delta_j| \leq \xi_{gi} \text{ for all } (i, j) \in \{\text{tied pairs}\} \]

- first proposed by [Zheng et.al 2008]
  - pairwise constraints and listwise objective
  - obtain the optimum \( \delta^* \) and use it as a functional gradient for \( f(x_i) \)
- problems:
  - no formal proof for functional gradient
  - not practical - quadratic program (QP) with \( O(n^3) \) complexity
The optimum $\delta^*$ is the functional gradient

We prove that

$$\delta^*_i = \frac{\partial L(\{(y_i, f(x_i))\}_{i=1}^n)}{\partial f(x_i)}$$

- Lemma 2 in the paper
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  - $\delta$ satisfying constraints with equality is enough

\[
\Delta_{ij} = y_i - y_j
\]

\[
\delta_2 \quad \xi_g \quad \delta_4 \quad \xi_{g+1}
\]

\[
\ell_g, u_g
\]

\[
\ell^*, u^*
\]

\[
\max \left\{ \ell^* - f(x_i), \min \left\{ f(x_i) - u^*, 0 \right\} \right\}
\]
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  - $\delta$ satisfying constraints with equality is enough

$$\xi_g + 1 \xi_g + \Delta_{ij} = y_i - y_j$$

$$\delta_2$$

$$\delta_4$$

$$f(x_i)$$
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  1. $\delta$ satisfying constraints with equality is enough
  2. relevance grade interval $\{[l_g, u_g]\}$ can be found first, then obtain $\delta$

$$f(x_i) = \delta^* = \max\{l^*_g - f(x_i), \min\{f(x_i) - u^*_g, 0\}\}$$
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  1. $\delta$ satisfying constraints with equality is enough
  2. relevance grade interval $\{[\ell_g, u_g]\}$ can be found first, then obtain $\delta$
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  1. $\delta$ satisfying constraints with equality is enough
  2. relevance grade interval $\{[\ell_g, u_g]\}$ can be found first, then obtain $\delta$
We can reduce the number of variables

- original QP has \( n \) variables and \( O(n^2) \) constraints
- observation:
  1. \( \delta \) satisfying constraints with equality is enough
  2. relevance grade interval \( \{[\ell_g, u_g]\} \) can be found first, then obtain \( \delta \)

\[ \Delta_{g, g+1} \]

\[ \delta_3 = u_g - f(x_4) \]
We can reduce the number of variables

- original QP has \( n \) variables and \( O(n^2) \) constraints
- observation:
  1. \( \delta \) satisfying constraints with equality is enough
  2. relevance grade interval \( \{[\ell_g, u_g]\} \) can be found first, then obtain \( \delta \)

\[
\delta_1 = 0
\]
IntervalRank

Efficient implementation

We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  1. $\delta$ satisfying constraints with equality is enough
  2. relevance grade interval $\{[\ell_g, u_g]\}$ can be found first, then obtain $\delta$

\[
\delta_3 = u_g - f(x_i)
\]
\[
\delta_4 = \ell_{g+1} - f(x_4)
\]

$\Delta_{g,g+1}$

$\xi_g$

$f(x_i)$

$\ell_g$

$u_g$

$\ell_{g+1}$

$u_{g+1}$

Taesup Moon (Yahoo! Labs)

WSDM 2010

Feb 5, 2010 13 / 21
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  1. $\delta$ satisfying constraints with equality is enough
  2. relevance grade interval $\{[\ell_g, u_g]\}$ can be found first, then obtain $\delta$

  \[
  \Delta_{g,g+1} = \xi_{g+1} - \xi_g
  \]

- finding \textit{minimum efforts} $\delta^*$ can be obtained from the \textit{optimum intervals} $\{[\ell^*_g, u^*_g]\}$ that lead to them
We can reduce the number of variables

- original QP has $n$ variables and $O(n^2)$ constraints
- observation:
  1. $\delta$ satisfying constraints with equality is enough
  2. relevance grade interval $\{[\ell_g, u_g]\}$ can be found first, then obtain $\delta$

finding minimum efforts $\delta^*$ can be obtained from the optimum intervals $\{[\ell^*_g, u^*_g]\}$ that lead to them

$$\delta^*_i = \min\{f(x_i) - u^*_g, 0\}$$
We can reduce the number of variables

- original QP has \( n \) variables and \( O(n^2) \) constraints
- observation:
  1. \( \delta \) satisfying constraints with equality is enough
  2. relevance grade interval \([\ell_g, u_g]\) can be found first, then obtain \( \delta \)

\[
\begin{align*}
\Delta_{g,g+1} &= \xi_g + 1 \\
\xi_g &= \ell_g \\
\ell_g &= u_g \\
u_g &= \xi_{g+1} \\
\xi_{g+1} &= u_{g+1} \\
\ell_{g+1} &= \Delta_{g,g+1} \\
u_{g+1} &= \ell_{g+1} - f(x_i) \\
\delta_4^* &= \ell_{g+1} - f(x_4) \\
\end{align*}
\]

- finding minimum efforts \( \delta^* \) can be obtained from the optimum intervals \([\ell^*_g, u^*_g]\) that lead to them
  - \( \delta^*_i = \max \{\ell^*_g - f(x_i), \min\{f(x_i) - u^*_g, 0\}\} \)
IntervalRank

Efficient implementation

Equivalent problem does not depend on $n$

Loss function

$$L\left(\{(y_i, f(x_i))\}_{i=1}^{n}\right) = \min_{\{[\ell_g, u_g]\}} \sum_{g \in G} \sum_{i \in S_g} \left[ (\ell_g - f(x_i))^2 + (f(x_i) - u_g)^2 \right],$$

where $\{[\ell_g, u_g]\}$ satisfy

- $\ell_g \leq u_g \leq \ell_g + \xi_g$, for all $g \in \{\text{relevance grades}\}$
- $\ell_{g+1} - u_g \geq \Delta_{g+1,g}$, for all $g \in \{\text{relevance grades}\}$

- problem reduced to $O(1)$ variables and $O(1)$ constraints
- no longer a QP, but we can still solve this efficiently
We can solve with $O(n \log n)$ complexity

Apply techniques from convex optimization

- log-barrier method
  - remove inequality constraints via log-barriers
- L-BFGS or conjugate gradient (CG) method
  - need to compute the objective and the gradient for each $\{\ell_g, u_g\}$
  - sorting of sums of $f(x_i)$ and $f(x_i)^2$ will do (details in the paper)
Summary of the algorithm

1. Find the intervals that lead to the “minimum efforts”.

2. Regress on the intervals to find $\{\delta^*_i\}_{i=1}^n$ and do functional gradient descent.

3. Also add (pointwise) regression loss:

$$L(\{(y_i, f(x_i))\}_{i=1}^n) = \frac{1}{2} \|\delta^*\|_2^2 + \frac{\lambda}{2} \sum_{i=1}^n (y_i - f(x_i))^2$$

- gives absolute score information.
1. Backgrounds

2. IntervalRank
   - Loss function via isotonic regression
   - Efficient implementation

3. Experimental results
LETOR 3.0 OHSUMED data

Data
- 106 queries, 16,140 query-document pairs
- 5-fold cross validation with $\frac{3}{5}$ for training, $\frac{1}{5}$ for validation, $\frac{1}{5}$ for test

Functional gradient boosting trees
- *slight variation*: added slack variables for the constraints
- parameters: 125 trees, 20 nodes per tree, shrinkage, $\lambda_1, \lambda_2, \lambda$
## Experimental results

**LETOR 3.0 OHSUMED data**

### NDCG@k results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>N@1</th>
<th>N@2</th>
<th>N@3</th>
<th>N@4</th>
<th>N@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RankBoost</td>
<td>0.4632</td>
<td>0.4504</td>
<td>0.4555</td>
<td>0.4543</td>
<td>0.4494</td>
</tr>
<tr>
<td>RankSVM</td>
<td>0.4958</td>
<td>0.4331</td>
<td>0.4207</td>
<td>0.4240</td>
<td>0.4164</td>
</tr>
<tr>
<td>FRank</td>
<td>0.5300</td>
<td>0.5008</td>
<td>0.4812</td>
<td>0.4694</td>
<td>0.4588</td>
</tr>
<tr>
<td>ListNet</td>
<td>0.5326</td>
<td>0.4810</td>
<td>0.4732</td>
<td>0.4561</td>
<td>0.4432</td>
</tr>
<tr>
<td>AdaRank.MAP</td>
<td>0.5388</td>
<td>0.4789</td>
<td>0.4682</td>
<td>0.4721</td>
<td>0.4613</td>
</tr>
<tr>
<td>AdaRank.NDCG</td>
<td>0.5330</td>
<td>0.4922</td>
<td>0.4790</td>
<td>0.4688</td>
<td>0.4673</td>
</tr>
<tr>
<td><strong>IntervalRank</strong></td>
<td><strong>0.5628</strong></td>
<td><strong>0.5448</strong></td>
<td><strong>0.4900</strong></td>
<td><strong>0.4703</strong></td>
<td><strong>0.4609</strong></td>
</tr>
</tbody>
</table>
### Experimental results

**LETOR 3.0 OHSUMED data**

#### Precision@k results

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>P@1</th>
<th>P@2</th>
<th>P@3</th>
<th>P@4</th>
<th>P@5</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RankBoost</td>
<td>0.5576</td>
<td>0.5481</td>
<td>0.5609</td>
<td>0.5580</td>
<td>0.5447</td>
<td>0.4411</td>
</tr>
<tr>
<td>RankSVM</td>
<td>0.5974</td>
<td>0.5494</td>
<td>0.5427</td>
<td>0.5443</td>
<td>0.5319</td>
<td>0.4334</td>
</tr>
<tr>
<td>FRank</td>
<td>0.6429</td>
<td>0.6195</td>
<td>0.5925</td>
<td>0.5840</td>
<td>0.5638</td>
<td>0.4439</td>
</tr>
<tr>
<td>ListNet</td>
<td>0.6524</td>
<td>0.6093</td>
<td>0.6016</td>
<td>0.5745</td>
<td>0.5502</td>
<td>0.4457</td>
</tr>
<tr>
<td>AdaRank.MAP</td>
<td>0.6338</td>
<td>0.5959</td>
<td>0.5895</td>
<td>0.5887</td>
<td>0.5674</td>
<td>0.4487</td>
</tr>
<tr>
<td>AdaRank.NDCG</td>
<td>0.6719</td>
<td>0.6236</td>
<td>0.5984</td>
<td>0.5838</td>
<td>0.5767</td>
<td>0.4498</td>
</tr>
<tr>
<td>IntervalRank</td>
<td><strong>0.6892</strong></td>
<td><strong>0.6522</strong></td>
<td>0.5768</td>
<td>0.5556</td>
<td>0.5488</td>
<td>0.4466</td>
</tr>
</tbody>
</table>
Commercial search engine data

Data
- training set: 8,180 queries, 341,300 query-document pairs
- test set: 916 queries, 32,008 query-document pairs
- 5 grade relevance judgments: \( \mathcal{G} = \{ \text{Perfect, Excellent, Good, Fair, Bad} \} \)

Functional gradient boosting trees
- slight variation: added slack variables for the constraints
- parameters: 600 trees, 20 nodes per tree, shrinkage, \( \lambda_1, \lambda_2, \lambda \)

Comparing schemes
- GBDT (pointwise), GBRank (pairwise), ListMLE (listwise)
- with or without additional regression term
- The running time of IntervalRank was in the same range with others.
- NDCG@1
Experimental results

Commercial search engine data

- running time of IntervalRank was in the same range with others
- NDCG@5
Experimental results

Commercial search engine data

- running time of IntervalRank was in the same range with others
- NDCG@5

we gain up to 1% over other methods!