Post Processing of Ranking in Search

Jun Xu

Institute of Computing Technology,
Chinese Academy of Sciences

Joint work with Fangzhao Wu, Hang Li, and Xin Jiang
Outline

• Post processing of ranking
• Ranking optimization with constraints
• Summary
One may want to ‘twist’ relevance ranking
Learning Approach to Ranking

- **Documents**: \( \left\{ x_1^{(1)}, 1 \right\}, \ldots, \left\{ x_M^{(N)}, 2 \right\} \)

  - **Training Data**: \( \left\{ x_1^{(1)}, 4 \right\}, \left\{ x_1^{(N)}, 5 \right\}, \ldots \)

  - **Test Data**: \( \left\{ x_1^{(1)}, 2 \right\}, \left\{ x_1^{(N)}, 3 \right\}, \ldots \)

- **Human Labels**: \( \{ \text{queries} \} \)

- **Learning Algorithm**: \( f(x; w) \)

  - **Ranking Model**: \( f(x; w) \)

- **Minimizing Loss**: 

  - **Online Ranking**: 
    - **Scoring & Sorting**

- **Post Processing of Ranking**

  - **Human Knowledge**: 
    - \( \left\{ x_1, f(x_1; w) \right\}, \left\{ x_2, f(x_2; w) \right\}, \ldots, \left\{ x_M, f(x_M; w) \right\} \)
Incorporating Human Knowledge

• Designing features or ranking models
  – Indirect
  – Limited to cross-query knowledge, for generalizing to other queries; It is difficult, costly, or even impossible to implement in features and models to incorporate some types of knowledge
  – Modify both offline and online components

• Post processing of ranking
  – Direct (apply on test queries directly)
  – Can incorporate query/user/context dependent knowledge
  – Only modify online component
Post Processing of Ranking

- **Documents**
  - Training data
  - Test data

- **Human labels**

- **Learning algorithm**
  - Minimizing loss

- **Ranking model**
  - \( f(x; w) \)

- **Online ranking**
  - Scoring & sorting

- **Post processing**
  - Knowledge (rules)
Post Processing Knowledge (Rules)

• Query dependent
  – Specific query type: if the query is a name, promote the corresponding personal homepage
  – Specific query: if the query is “Microsoft”, promote http://www.microsoft.com/ to rank 1

• User dependent (personalization)
  – Query: “Michael Jordan”
    • For basketball fan: promote the Wikipedia entry of the basketball player
    • For CS researcher: promote the Wikipedia entry of the professor at the UC Berkeley
Post Processing Knowledge (cont’)

- Context dependent (session-based)

<table>
<thead>
<tr>
<th>Query 1: “homes for rent in atlanta”</th>
<th>Query 2: “houses for rent in atlanta”</th>
</tr>
</thead>
<tbody>
<tr>
<td>× Atlanta homes for rent - home rentals - houses for ren...</td>
<td>× Atlanta homes for rent - home rentals - houses for ren...</td>
</tr>
<tr>
<td>Rentlist is directory of Atlanta home rentals featuring links to...</td>
<td>Rentlist is directory of Atlanta home rentals featuring links to...</td>
</tr>
<tr>
<td><strong>Homes For Rent, lease in Atlanta suburbs. Can’t sell ...</strong></td>
<td><strong>Homes for Rent in Atlanta, GA</strong></td>
</tr>
<tr>
<td>Atlanta homes for rent, homes for lease in Gwinnett and north...</td>
<td>Houses, Apartments and Homes for Rent in Atlanta, GA Find ...</td>
</tr>
<tr>
<td><strong>Rentals.com - Homes for Rent, Apartments, Houses ...</strong></td>
<td><strong>Atlanta Home Rentals, Homes for Rent in Atlanta ...</strong></td>
</tr>
<tr>
<td>Atlanta Home Rentals; Austin Home Rentals; Charlotte Home...</td>
<td>Atlanta Rentals - Homes for Rent in Atlanta, Apartments, Re...</td>
</tr>
<tr>
<td><a href="http://www.rentals.com">http://www.rentals.com</a></td>
<td><a href="http://www.rentals.com/Georgia/Atlanta">http://www.rentals.com/Georgia/Atlanta</a></td>
</tr>
<tr>
<td>× Atlanta Home Rentals, Homes for Rent in Atlanta ...</td>
<td>× Homes For Rent, lease in Atlanta suburbs. Can’t sell ...</td>
</tr>
<tr>
<td>Atlanta Rentals - Homes for Rent in Atlanta, Apartments, Re...</td>
<td>Atlanta homes for rent, homes for lease in Gwinnett and north...</td>
</tr>
<tr>
<td><a href="http://www.rentals.com/Georgia/Atlanta">http://www.rentals.com/Georgia/Atlanta</a></td>
<td><a href="http://atlantahomesforrenti.com">http://atlantahomesforrenti.com</a></td>
</tr>
<tr>
<td><strong>Homes for Rent in Atlanta, GA</strong></td>
<td><strong>Atlanta Homes for Rent, Rental Properties, Houses for ...</strong></td>
</tr>
<tr>
<td>Houses, Apartments and Homes for Rent in Atlanta, GA Find ...</td>
<td>Search for Homes for Rent in Atlanta, Georgia for free. View li...</td>
</tr>
</tbody>
</table>

From Xiang et al., SIGIR’ 10
Post Processing Knowledge (cont’)

- Document (website) dependent

  - Example rule: if webpage from one site is ranked at top, webpages from the other site will be demoted

http://www.baike.com/wiki/中科院计算技术研究所

http://baike.baidu.com/view/730187.htm
Heuristic Approaches

• Widely used in real search systems, however
  – Rules may be ambiguous, e.g., the document should be ranked at top three positions, no specific position is decided
  – Rules might be contradictory, e.g., two rules want to rank different documents to top 1
  – Different orders of applications of rules might yield different ranking results. The later one has higher priority
  – Hard to balance between application of rules and preservation of the original ranking list
  – Hard to manage the old/new rules
• Difficult to formalize in a theoretically sound, effective, and efficient way
Outline

• Motivation of post processing of ranking
• Ranking optimization with constraints (Wu et al., CIKM 2014)
• Conclusion
Main Idea

• Traditional approaches
  – Mainly based on heuristic rules
  – No principled approach

• Our work
  – Formalizes as a constrained optimization problem
    • Constraints: post-processing rules
    • Object function: tradeoff between original ranking and rules
  – Implementation with Bradley-Terry model
Covered Constraints

• Top-k constraint
  – A document should be at top k positions

• Not-top-k constraint
  – A document cannot be at top k positions
Related Work

• Post ranking with heuristics
  – **Result diversification**: re-ranking after a ranking list based on relevance is created [Dou et al., ‘11; Vee et al., ‘08]
  – **Personalized search**: client side re-ranking based on user interest [Radlinski & Dumais, ‘06; Sugiyama et al., ‘04; Teevan et al., ‘05]
  – **Context aware ranking**: demoting clicked URL in current search result, if it was clicked in the previous search in the same session [Xiang et al., ‘10]

• Probabilistic models for ranking
  – Plackett-Luce model: stage-wise generative model [Luce ‘75]
  – Mallows model: distance based [Mallows ‘57]
  – Bradely-Terry model: pairwise comparisons [Bradley & Terry, ‘52]
Ranking Optimization with Constraints

- **Constraints**: rules for post ranking. \( C = \{ c_i(\cdot) \}, c_i: \Omega_N \to \{0, 1\} \)
- **Objective function**: trade-off between adherence to the *original ranking list* and satisfaction of the *constraints*
Probabilistic Approach

• Introducing probabilistic ranking model $M$ and

$$\pi = \arg \max_{\pi} P(\tau|M)$$

$$\min_{M} L(\sigma, M) + \lambda \cdot R(C, M)$$

• Define

  - $L(\sigma, M) = -\log P(\sigma|M)$
  - $R(C, M) = -\log P(C|M)$

• Two steps

  - Estimating $M$

    $$\min_{M} -\log P(\sigma|M) - \lambda \cdot \log P(C|M)$$

  - Getting optimal ranking list

    $$\pi^* = \arg \max_{\pi \in \Omega_N} P(\pi|M)$$
Using Bradley-Terry Model

- Represents distribution of permutation by making pairwise comparisons
  \[ p_{ij} = P\{(i, j)\} = \frac{\theta_i}{\theta_i + \theta_j} \]

- Probability of a permutation
  \[
  P(\sigma|M) \propto \prod_{(i,j):\sigma(i) < \sigma(j)} p_{ij} = \prod_{(i,j):\sigma(i) < \sigma(j)} \frac{\theta_i}{\theta_i + \theta_j}
  \]

- Probability of a constraint set
  \[
  P(C|M) \propto \prod_{c \in C} \prod_{(i,j) \in P^c} p_{ij} = \prod_{c \in C} \prod_{(i,j) \in P^c} \frac{\theta_i}{\theta_i + \theta_j}
  \]

\(P^c\): set of preference pairs derived from constraint \(c\)

- top-k constraint: \(P^c = \{(i, j) | j: \sigma(j) > k\}\)
- not-top-k constraint: \(P^c = \{(j, i) | j: \sigma(j) \leq k\}\)
**Objective Function**

\[
\min_M M - \log P(\sigma|M) - \lambda \cdot \log P(C|M)
\]

\[
P(\sigma|M) \propto \prod_{(i,j): \sigma(i) < \sigma(j)} \frac{\theta_i}{\theta_i + \theta_j'}
\]

\[
P(C|M) \propto \prod_{c \in C} \prod_{(i,j) \in P^c} \frac{\theta_i}{\theta_i + \theta_j}
\]

\[
\min_{\Theta} f(\Theta) = -\sum_{(i,j): \sigma(i) < \sigma(j)} \log \frac{\theta_i}{\theta_i + \theta_j} - \sum_{c \in C} \left( \rho^c \cdot \sum_{(i,j) \in P^c} \log \frac{\theta_i}{\theta_i + \theta_j} \right)
\]

subject to \( \forall i : \theta_i > 0, \sum_{i=1}^N \theta_i = 1, \)

\[
\theta_i = \exp\{s_i\}, s_i \in R
\]

\[
\min_S f(S) = \sum_{(i,j): \sigma(i) < \sigma(j)} (\log(e^{s_i} + e^{s_j}) - s_i) + \sum_{c \in C} \left( \rho^c \cdot \sum_{(i,j) \in P^c} (\log(e^{s_i} + e^{s_j}) - s_i) \right)
\]

**Theorem 4.1.** \( f(S) \) is a convex function.
Optimizing with Gradient Descent

\[
\frac{df}{ds_i} = \left( \sum_{j: \sigma(j) < \sigma(i)} \frac{e^{s_i}}{e^{s_i} + e^{s_j}} - \sum_{j: \sigma(i) < \sigma(j)} \frac{e^{s_j}}{e^{s_i} + e^{s_j}} \right) + \sum_{c \in C} \rho^c \left( \sum_{j: (j, i) \in \mathcal{P}_c} \frac{e^{s_i}}{e^{s_i} + e^{s_j}} - \sum_{j: (i, j) \in \mathcal{P}_c} \frac{e^{s_j}}{e^{s_i} + e^{s_j}} \right)
\]

\[
S^{(t)} = S^{(t-1)} - \gamma^{(t)} \frac{\partial f}{\partial s} \bigg|_{s=s^{t-1}}
\]

- Intuitive explanation: given a pair \((i, j)\), \(i\) be pushed upward and \(j\) be pushed downward with identical force strengths

- Demote \(i\)
- Promote \(i\)
Algorithm 1 Ranking Optimization Algorithm

Require: Initial ranking $\sigma$, constraints $C$, and shrinkage rate $0 < \alpha < 1$

1: $S^{(0)} \leftarrow$ random values
2: $t \leftarrow 1$
3: repeat
4: $\nabla S = \frac{\partial f}{\partial S} |_{S=S^{(t-1)}} \{\text{Equation (6)}\}$
5: $\gamma \leftarrow 1$
   {search optimal step size using backtracking}
6: while $f(S^{(t-1)} - \gamma \nabla S) > f(S^{(t-1)}) - \frac{\gamma}{2} \| \nabla S \|^{2}$ do
7: $\gamma \leftarrow \alpha \gamma$
8: end while
9: $S^{(t)} \leftarrow S^{(t-1)} - \gamma \nabla S \{\text{Equation (7)}\}$
10: $t \leftarrow t + 1$
11: until convergence
12: return $\Theta = \{\frac{e^{s_1}}{Z}, \ldots, \frac{e^{s_N}}{Z}\}$, where $Z = \sum_{n=1}^{N} e^{s_n}$

**Theorem 4.2.** Algorithm 1 converges in finite steps and the convergence rate is $O\left(\frac{1}{\epsilon}\right)$, where $\epsilon > 0$ is the tolerance.
Experimental Settings

• Datasets

<table>
<thead>
<tr>
<th>dataset</th>
<th># queries</th>
<th>#documents</th>
<th>#relevance levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQ2007</td>
<td>1692</td>
<td>69623</td>
<td>3</td>
</tr>
<tr>
<td>MQ2008</td>
<td>784</td>
<td>15211</td>
<td>3</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>106</td>
<td>16140</td>
<td>3</td>
</tr>
<tr>
<td>.Gov</td>
<td>50</td>
<td>49058</td>
<td>2</td>
</tr>
<tr>
<td>Enterprise</td>
<td>183</td>
<td>5464</td>
<td>3</td>
</tr>
</tbody>
</table>

• Basic ranking model: LambdaMART

• Constraints construction
  – Top-k constraint (k = 1, 3, 5): for each query, sort documents according to labels and randomly select one document from top k positions
  – Not-top-k constraint (k = 5, 10): for each query, sort documents according to labels and randomly select one document from the positions after k positions
Experimental Settings

• Baselines
  – Radical
    • Top-k constraint → top one position
    • Not-top-k constraint → bottom position
  – Moderate
    • Top-k constraint → middle of the top \( k \) positions
    • Not-top-k constraint → middle of the remaining list after \( k \)
  – Conservative
    • Top-k constraint → the position of \( k \)
    • Not-top-k constraint → the position of \( k + 1 \).
  – Proportional
    • Top-k constraint → the position of \( k \times \frac{p_{o}s}{N} \)
    • Not-top-k constraint → the position of \( k + p_{o}s \left( 1 - \frac{k}{N} \right) \)
Experimental Results

(a) top-3, not-top-5, $\rho^t=100$, $\rho^n=10$

(b) top-3, not-top-10, $\rho^t=100$, $\rho^n=10$

(c) top-5, not-top-10, $\rho^t=100$, $\rho^n=10$

MQ2007

(a) top-3, not-top-5, $\rho^t=10$, $\rho^n=0$

(b) top-3, not-top-10, $\rho^t=10$, $\rho^n=10$

(c) top-5, not-top-10, $\rho^t=10$, $\rho^n=10$

MQ2008
Average Running Time per Query

Table 2: Average time (in milliseconds) of ranking optimization in setting of (top-5, not-top-10).

<table>
<thead>
<tr>
<th></th>
<th>MQ2008</th>
<th>MQ2007</th>
<th>OHSUMED</th>
<th>.Gov</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>4.24</td>
<td>6.85</td>
<td>134.53</td>
<td>70.06</td>
<td>6.45</td>
</tr>
</tbody>
</table>

- Tested on a Laptop PC with 2.4GHZ CPU and 4GB memory
- For most queries, the algorithm converges within 10 iterations
- Ranking optimization can be performed online
Case Study 1: How Ranking Optimization Works

• Example ranking from MQ2008

- RankOpt promoted the relevant document and demoted the not relevant documents
- RankOpt outperformed baselines of Moderate, Conservative, and Proportional, when constraints are correct
Case Study 2: How Ranking Optimization Works

- Example ranking from MQ2008

  - RankOpt outperformed Radical method, if constraints contain noise
  - RankOpt made good trade-off between constraints and original ranking
Discussion: Constraint Types

- Top-$k$ and not-top-$k$ constraints individually improved the ranking performances
- Performances be further improved when both are used
- RankOpt can leverage multiple types of constraints
Outline

• Motivation of post processing of ranking
• Ranking optimization with constraints
• Summary
Summary

• Post-processing of ranking is important for search
• Heuristic approaches have limitations
• Our preliminary work makes use of Bradley-Terry model for handling the top-k and not-top-k rules
• Next step
  – Defining and incorporating other types of constraints into the framework, especially the constraints on search result diversification
References

Thank you!