Ranking with Query-Dependent Loss for Web Search

Jiang Bian\(^1\), Tie-Yan Liu\(^2\), Tao Qin\(^2\), Hongyuan Zha\(^1\)

Georgia Institute of Technology\(^1\)
Microsoft Research Asia\(^2\)
Outline

• **Motivation**

• Incorporating Query Difference into Ranking
  – Position-sensitive query-dependent loss function
  – Learning methods
  – Example query-dependent loss functions
    • RankNet
    • ListMLE

• Experiments and Discussions
Query Difference

Navigational

Informational

Transactional

Search intention

Queries

Relational info needs

Subtopic retrieval

Topic distillation
Position-Sensitive Query Difference

The ranking model should aim to rank the exact web page on the top position of the result list.

For "WSDM 2010":
1. wsdm2010.org
2. ...
3. ...
4. ...
5. ...
... n. ...

For "New York City":
1. www.nyc.gov
2. en.wikipedia.org/wiki/New_York_City
3. www.nyctourist.com
4. www.nycgo.com
5. ...
... n. ...

For "winamp download":
1. www.winamp.com/media-player
2. ...
3. ...
4. ...
5. ...
... n. ...

This kind of position-sensitive query difference requires different objectives (loss function) for the ranking model.
Incorporate Query Difference into Ranking

• We propose to incorporate query difference into ranking by introducing position-sensitive query-dependent loss functions in the learning process.

• Previous Work:
  – Key idea: employ different ranking functions for different classes/clusters of queries
  – Query type classification for web document retrieval (Kang et al. SIGIR2003)
  – Query-dependent ranking using k-nearest neighbor (Geng et al. SIGIR2008)
  – Incorporating query difference for learning retrieval functions in information retrieval (Zha et al. CIKM2006)

• We propose to learn one ranking function based on query-dependent loss function
Outline

• Motivation

• Incorporating Query Difference into Ranking
  – Position-sensitive query-dependent loss function
  – Learning methods
  – Example query-dependent loss functions
    • RankNet
    • ListMLE

• Experiments and Discussions
Incorporating Query Difference into Ranking: Query-Dependent Loss Function

Query level loss
- Having same form among all queries

Diverse ranking objectives implied by different queries

Query level loss
- Each query has its own form

Difficult and expensive in practice to define individual objective for each query

Category level loss
- Each query category has its own form

Query categorization
Query-Dependent Loss based on Query Taxonomy of Web Search

Navigational $\rightarrow$ $C_N$
Transactional $\rightarrow$ $C_I$

The loss should focus on the exact relevant document
The loss should consider relevant documents which should be ranked in top-K positions

Query-dependent loss function:

$$L(f; q) = \alpha(q)L(f; q, C_I) + \beta(q)L(f; q, C_N)$$

$$L(f; q, C) = \sum_{x \in X_q} \ell(f(x), g(x), p(x); \Phi(q, C))$$

The example-level loss $l$ contribute to the whole loss if the true rank position $p(x)$ of the example $x$ is included in $\Phi(q, C)$.

The actual value of example-level loss is defined by $f(x)$ and $g(x)$.
Learning Methods

• Basic method:
  - To minimize the query-dependent loss function $L_f \text{ w.r.t.}$ the ranking parameters, denoted as $\omega$

\[
L_f = \sum_{q \in Q} \alpha(q) L(f_\omega; q, C_I) + \beta(q) L(f_\omega; q, C_N)
\]

- First, obtain pre-defined categorization for each query
  • Navigational: $\alpha(q) = 0, \beta(q) = 1$.
  • Informational: $\alpha(q) = 1, \beta(q) = 0$.

- Then, learn the parameters of ranking functions using traditional optimization methods
  • Gradient descent
Learning Methods

• Query categorization may not be available
• Even the existing query categorization may not be best for ranking

• Unified Method:
  – We propose to learn the ranking function jointly with query categorization
    • Consider query categorization is defined by a set of query features

\[
\alpha_\gamma(q) = \frac{\exp(\langle \gamma, z_q \rangle)}{1 + \exp(\langle \gamma, z_q \rangle)}, \quad \beta_\gamma(q) = \frac{1}{1 + \exp(\langle \gamma, z_q \rangle)}
\]

Parameters for query categorization

Features of query

• ...
Learning Methods

• Unified Method:
  – Alternates between minimizing the loss w.r.t. to $\omega$ and $\gamma$:

  $\textbf{while } (L_f(\omega_k, \gamma_k) - L_f(\omega_{k+1}, \gamma_{k+1})) > \epsilon \textbf{ do}$

  $\omega_{k+1} \leftarrow \arg\min_{\omega} \sum_{q \in Q} \alpha_{\gamma_k}(q)L(f_{\omega_k}; q, C_I)$
  $\quad + \beta_{\gamma_k}(q)L(f_{\omega_k}; q, C_N)$

  $\gamma_{k+1} \leftarrow \arg\min_{\gamma} \sum_{q \in Q} \alpha_{\gamma_k}(q)L(f_{\omega_{k+1}}; q, C_I)$
  $\quad + \beta_{\gamma_k}(q)L(f_{\omega_{k+1}}; q, C_N)$

• We do not need query categorization during testing, thus $\gamma$ will not be used for ranking during testing -- $\gamma$ is considered as hidden information in learning
Example Query-Dependent Loss Functions

- RankNet: (pairwise)
  - Original loss function:
    \[
    L(o_{ij}) = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log(1 - P_{ij})
    \]

- Query-dependent loss function:
  \[
  L(o_{ij}, q) = \sum_{p(i)=1}^{n_q} P(p(i)|x_i, g(x_i))(\alpha(q) \cdot 1_{\{p(i)\in\Phi(q,c_I)\}} + \beta(q) \cdot 1_{\{p(i)\in\Phi(q,c_N)\}}) \cdot L(o_{ij}),
  \]

Probability that \(x_i\) with label \(g(x_i)\) is ranked at position \(p(i)\)

q-d loss

sum over rank positions

navigational

informational

desired target values
Example Query-Dependent Loss Functions

- **ListMLE**: (listwise)
  - Original loss function:
    \[ L(f; q) = \phi(\Pi_f(x), y) = -\log P^k_y(\Pi_f(x)) \]
    - \( x \): the list of documents
    - \( y \): the true permutation of document under \( q \)
    - \( \Pi_f(x) \): the permutation ordered by ranking function \( f \)

  - Query-dependent loss function:
    \[ L(f; q) = -\alpha_q \log P^{k_I}_y(\Pi_f(x)) - \beta_q \log P^{k_N}_y(\Pi_f(x)) \]

  Plackeet-Luce model as top-\( k \) surrogate loss

  Navigational: top-\( k_N \) surrogate likelihood loss
  Informational: top-\( k_I \) surrogate likelihood loss
Outline

• Motivation

• Incorporating Query Difference into Ranking
  – Position-sensitive query-dependent loss function
  – Learning methods
  – Example query-dependent loss functions
    • RankNet
    • ListMLE

• Experiments and Discussions
Experimental Setup

• Dataset: LETOR 3.0:
  – TREC2003
    • 300 navigational queries, 50 informational queries
  – TREC2004
    • 150 navigational queries, 75 informational queries
  – 64 features for ranking
• To define query features:
  • Use a reference model (BM25) to find top-50 ranked documents, and take the mean of the features values of the 50 documents as the features of the query

• Compared methods:
  – Ranking algorithms using original loss function (RankNet, ListMLE)
  – Ranking algorithms using query-dependent loss function with pre-defined query categorization (SQD-RankNet, SQD-ListMLE)
  – Ranking algorithms using query-dependent loss function without pre-defined query categorization (UQD-RankNet, UQD-ListMLE)

• 5-fold cross validation
Results on RankNet

TREC2003

NDG@K

K

1 2 3 4 5 6 7 8 9 10

UQD-RankNet
SQD-RankNet
RankNet
Results on ListMLE

TREC2004
Discussions (1)

• Query-specific categories (features) is not available at testing time:
  – They can be viewed as extra tasks for the learner
  – Query-specific categories (features) of training data are transferred into other common features as training signals
  – The extra training signals serve as a query-specific inductive bias for ranking
Discussions (2)

- Query-dependent loss function vs. query-dependent ranking function
  - Query-dependent loss function contains more information for ranking than the loss for query-dependent ranking function.
  - Many queries can fit into more than one category.
  - There exists a number of documents which are not critical for ranking in the training set, but are very difficult to rank, such that, they may have much influence on the training process and attenuate the effect from important documents.
  - Query-dependent ranking function uses only a part of training dataset to learn the ranking model for each category.
  - Shorter training and testing time of query-dependent loss approach

![Graphs showing NDCG@K for different K values for navigational and informational queries.](image)
Summary

• Proposed to incorporate query difference into ranking by introducing query-dependent loss functions

• Introduced a new methods for learning the ranking function jointly with learning query categorization

• Exploited the position-sensitive query-dependent loss function on a popular query categorization scheme of Web search and applied it to two specific ranking algorithms, RankNet and ListMLE
Thanks!

Jiang Bian
Georgia Tech
jbian@cc.gatech.edu