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

Microsoft Research

Georgia Tech College of Computing School of Computer Science

Query Difference

Relational info needs

Navigational

Informational

Transactional

Queries

Subtopic retrieval

Topic distillation

Search intention

Microsoft Research

Georgia Tech College of Computing School of Computer Science

Position-Sensitive Query Difference

Navigational

"WSDM 2010"

1. wsdm2010.org
2. …
3. …
4. …
5. …

…

n. …

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

Informational

"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. …

The ranking model should target at presenting a set of relevant web pages on the top-K positions of returned results.

Transactional

"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

$$L_f = \sum_{q \in Q} L(f)$$

❖Query level loss
❖Having same form among all queries

Diverse ranking objectives implied by different queries

$$L_f = \sum_{q \in Q} L(f; q)$$

❖Query level loss
❖Each query has its own form

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

$$L_f = \sum_{q \in Q} \left(\sum_{i=1}^{m} \underbrace{P(C_i|q)}_{\textit{Query categorization}} \underbrace{L(f; q, C_i)}_{} \right)$$

Query categorization

❖Category level loss
❖Each query category has its own form

Query-Dependent Loss based on Query Taxonomy of Web Search

Navigational → C_N ← Transactional

The loss should focus on the exact relevant document

Informational → C_I

The loss should consider relevant documents which should be ranked in top-K positions

Query-dependent loss function:

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

$$\alpha(q) = P(C_I|q) \qquad \beta(q) = P(C_N|q)$$

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

example-level loss
ranking scores
ground truth
true positions
important positions

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 w.r.t. the ranking parameters, denoted as ω

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

  - 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

Parameters for query categorization

$$\alpha_\gamma(q) = \frac{\exp(\langle \gamma, \mathbf{z}_q \rangle)}{1 + \exp(\langle \gamma, \mathbf{z}_q \rangle)}, \quad \beta_\gamma(q) = \frac{1}{1 + \exp(\langle \gamma, \mathbf{z}_q \rangle)}$$

Features of query

    - …

Learning Methods

- Unified Method:
  - Alternates between minimizing the loss w.r.t. to ω and γ:

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

$$\omega_{k+1} \leftarrow \arg\min_{\omega} \sum_{q \in Q} \alpha_{\gamma_k}(q) L(f_{\omega_k}; q, \mathcal{C}_I) + \beta_{\gamma_k}(q) L(f_{\omega_k}; q, \mathcal{C}_N)$$

$$\gamma_{k+1} \leftarrow \arg\min_{\gamma} \sum_{q \in Q} \alpha_{\gamma_k}(q) L(f_{\omega_{k+1}}; q, \mathcal{C}_I) + \beta_{\gamma_k}(q) L(f_{\omega_{k+1}}; q, \mathcal{C}_N)$$

- We do not need query categorization during testing, thus γ will not be used for ranking during testing -- γ is considered as hidden information in learning

Example Query-Dependent Loss Functions

- RankNet: (pairwise)
  - Original loss function:

desired target values

$$L(o_{ij}) = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log(1 - P_{ij})$$

  - Query-dependent loss function:

$$\underbrace{\boxed{L(o_{ij}, q)}}_{\text{q-d loss}} = \underbrace{\sum_{p(i)=1}^{n_q}}_{\text{sum over rank positions}} \underbrace{P(p(i)|x_i, g(x_i))}_{\text{Probability that } x_i \text{ with label } g(x_i) \text{ is ranked at position } p(i)} \Big(\underbrace{\alpha(q) \cdot \mathbf{1}_{\{p(i) \in \Phi(q, C_I)\}}}_{\text{informational}} + \underbrace{\beta(q) \cdot \mathbf{1}_{\{p(i) \in \Phi(q, C_N)\}}}_{\text{navigational}} \Big) \cdot L(o_{ij}),$$

q-d loss

sum over rank positions

navigational

informational

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

Example Query-Dependent Loss Functions

- ListMLE: (listwise)
  - Original loss function:

$$L(f;q) = \phi(\Pi_f(\mathbf{x}), \mathbf{y}) = -\log \underline{P_{\mathbf{y}}^{k}(\Pi_f(\mathbf{x}))}$$

Plackeet-Luce model as top-k surrogate loss

    - $\mathbf{x}$: the list of documents
    - $\mathbf{y}$: the true permutation of document under q
    - $\Pi_f(\mathbf{x})$: the permutation ordered by ranking function f

  - Query-dependent loss function:

$$L(f;q) = \underline{-\alpha_q \log P_{\mathbf{y}}^{k_I}(\Pi_f(\mathbf{x}))} - \underline{\beta_q \log P_{\mathbf{y}}^{k_N}(\Pi_f(\mathbf{x}))}$$

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

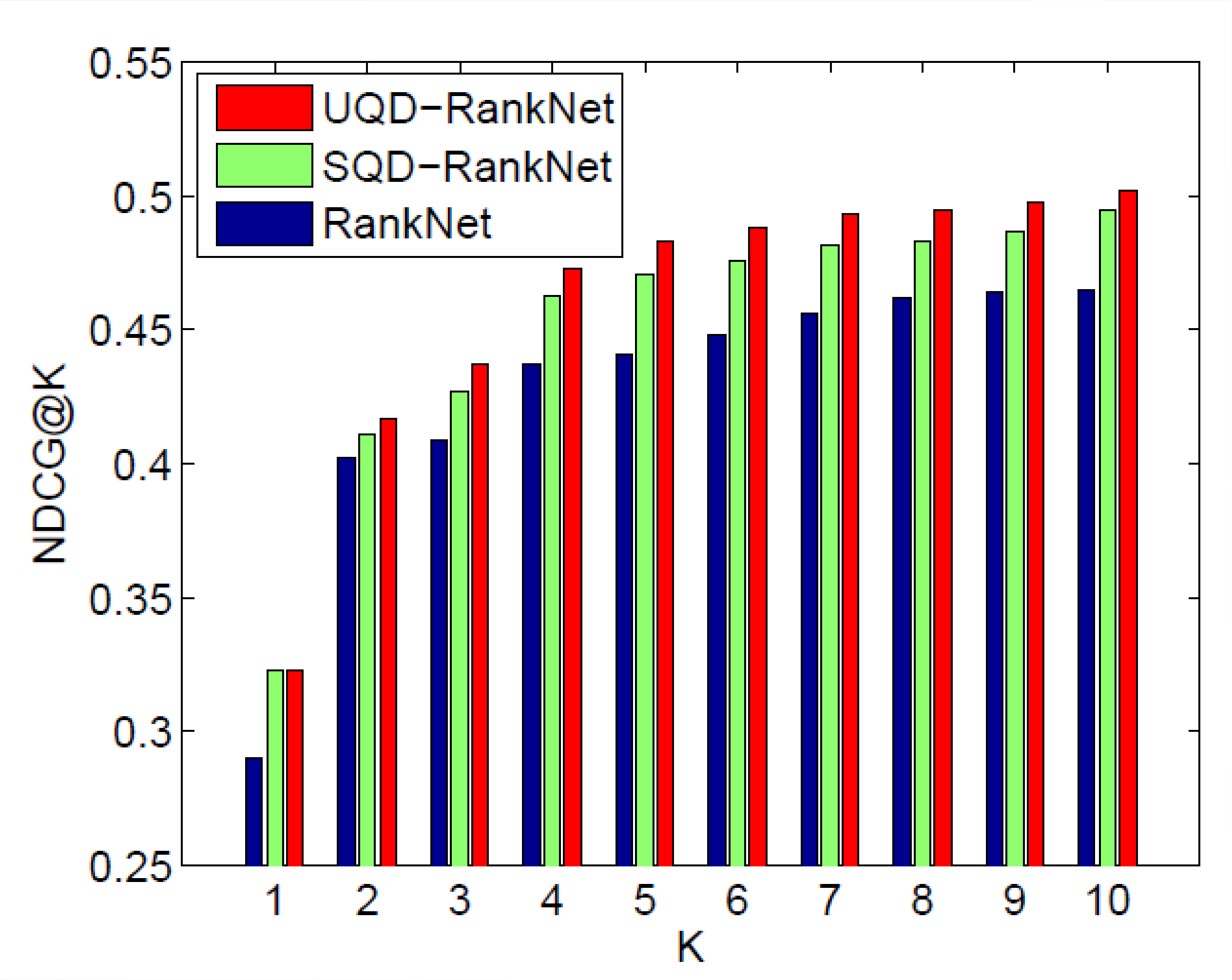
- ***Experiments and Discussions***

Microsoft Research Georgia Tech College of Computing School of Computer Science

Experimental Setup

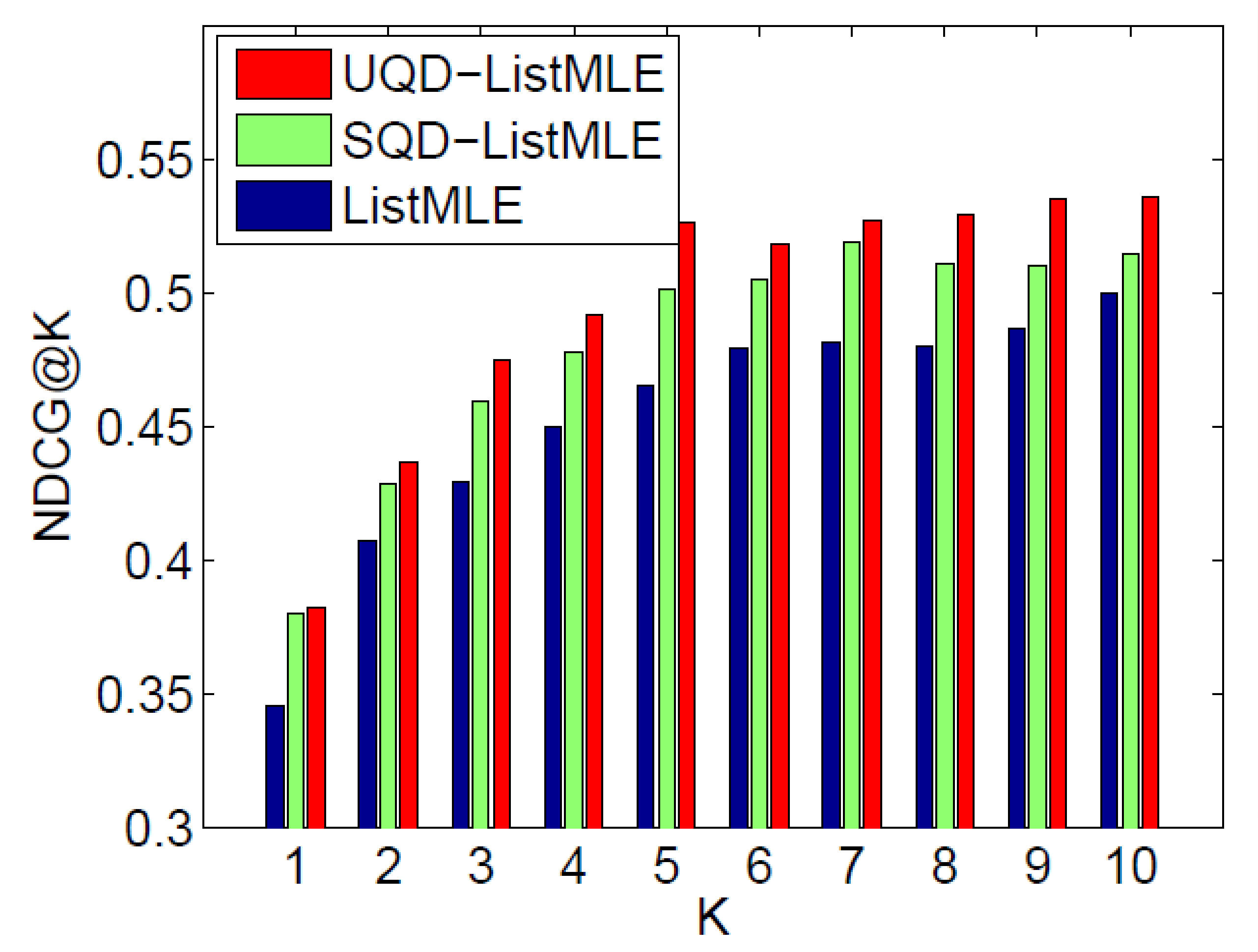
- Dataset: LETOR 3.0:
  - TREC2003
    - 300 navigational queries, 50 informational queries
  - TREC2004
    - 150 navigational queries, 75 informational queries
  - 64 features for ranking
  - To define query features:
    - Use a reference model (BM25) to find top-50 ranked documents, and take the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

Results on ListMLE



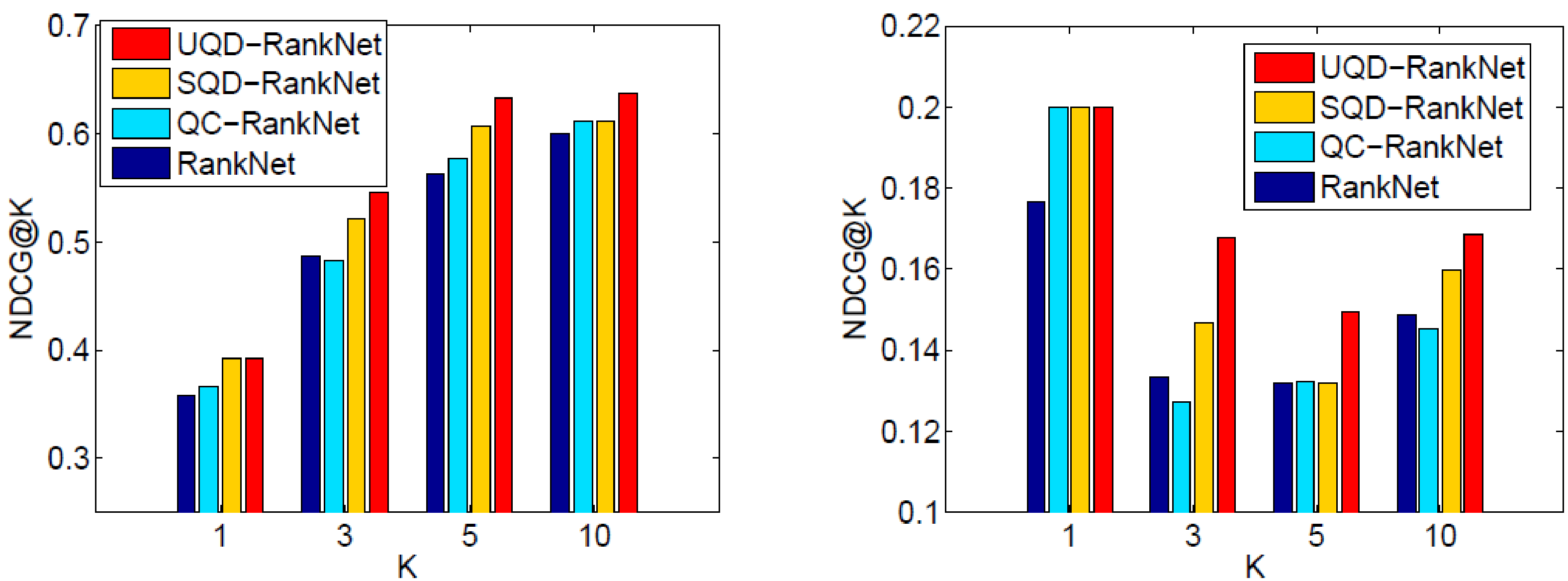
TREC2004

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



(a) Navigational queries (b) Informational queries

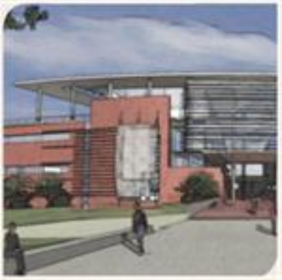
approach

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

Microsoft Research
Georgia Tech College of Computing
School of Computer Science