CCF YOCSEF Tianjin
July 11, 2015

Learning for Search Result Diversification

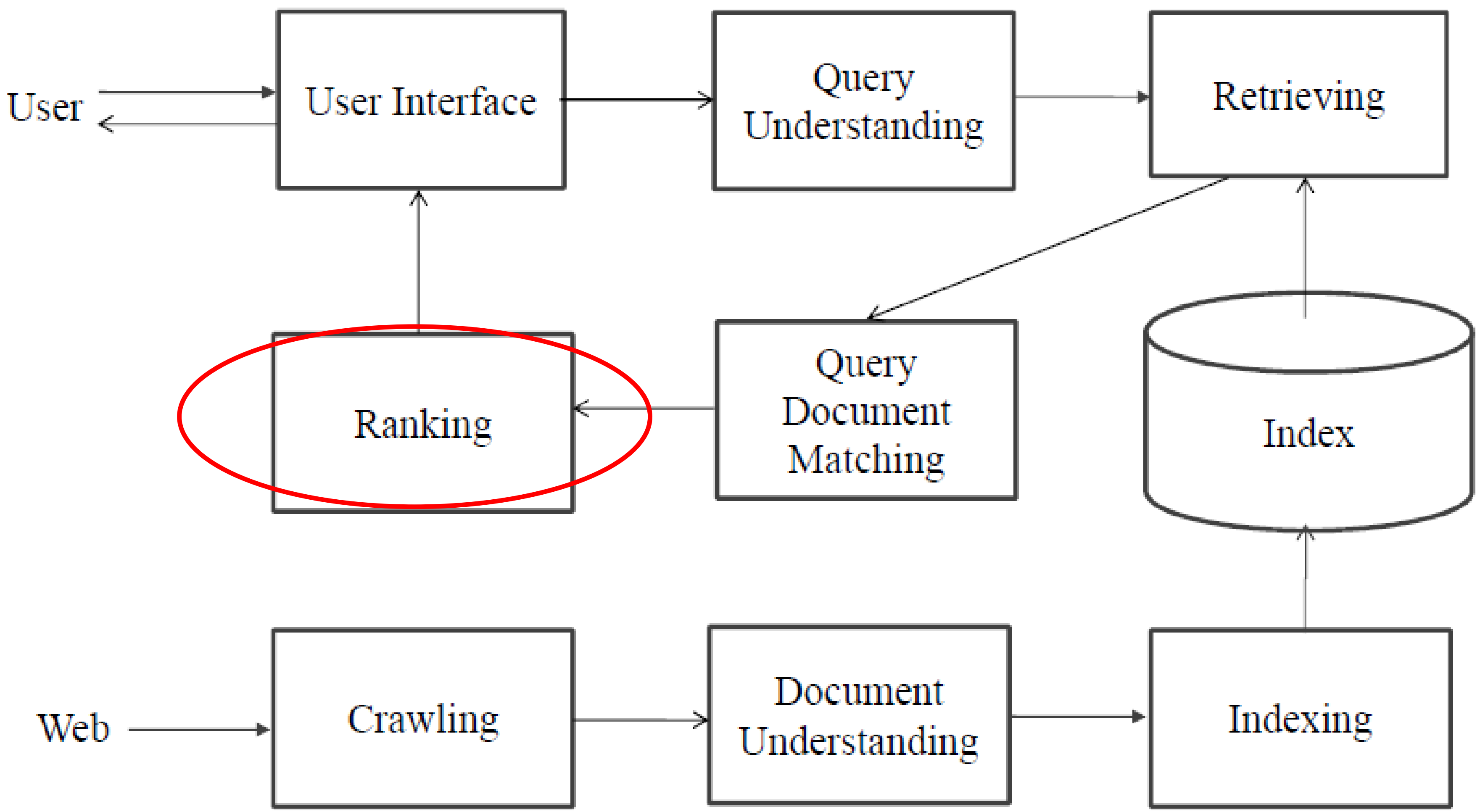
Jun Xu
junxu@ict.ac.cn

Institute of Computing Technology, CAS

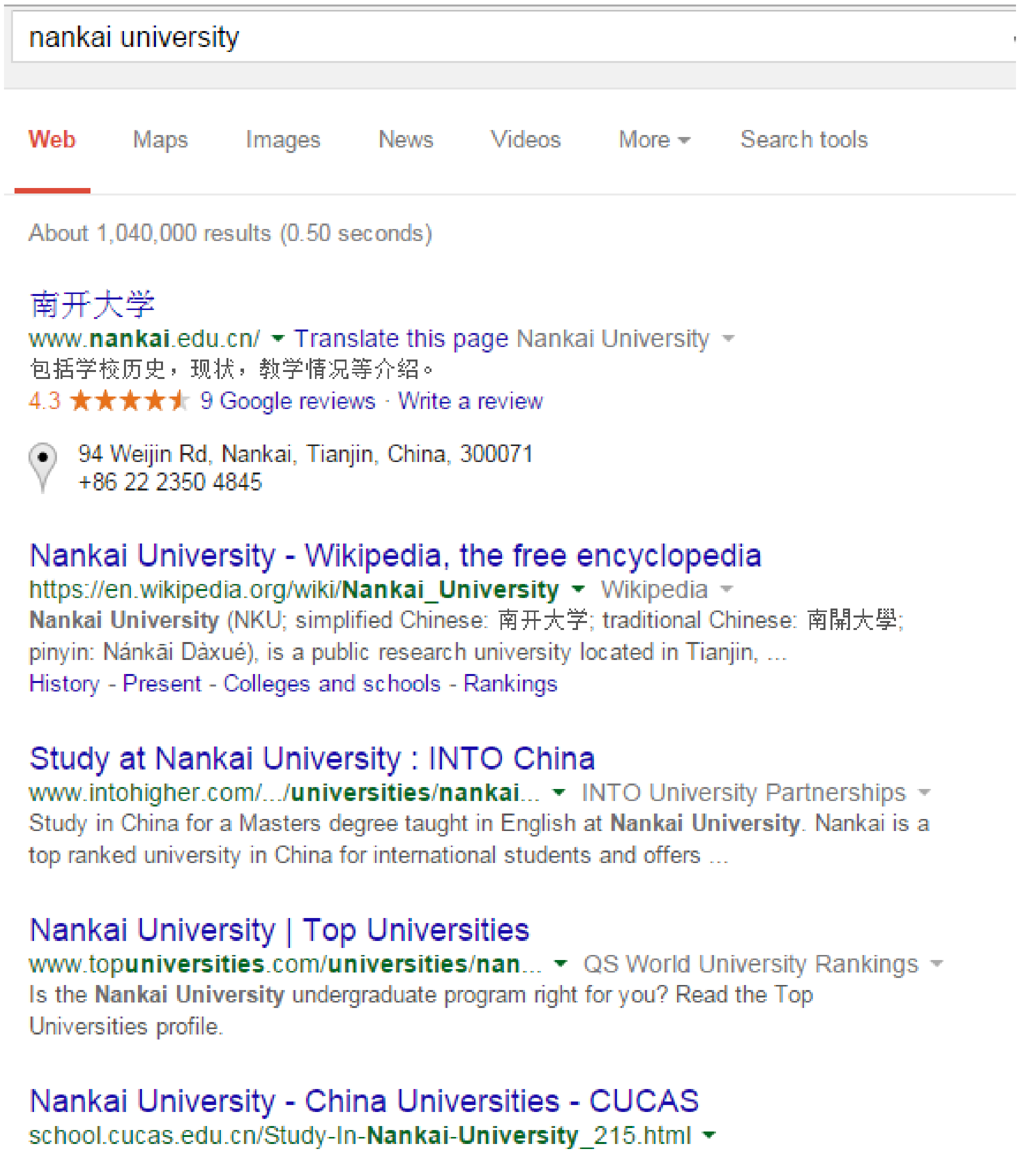
Outline

- Search result diversification
- Learning for search result diversification
- Summary

Web Search Engine

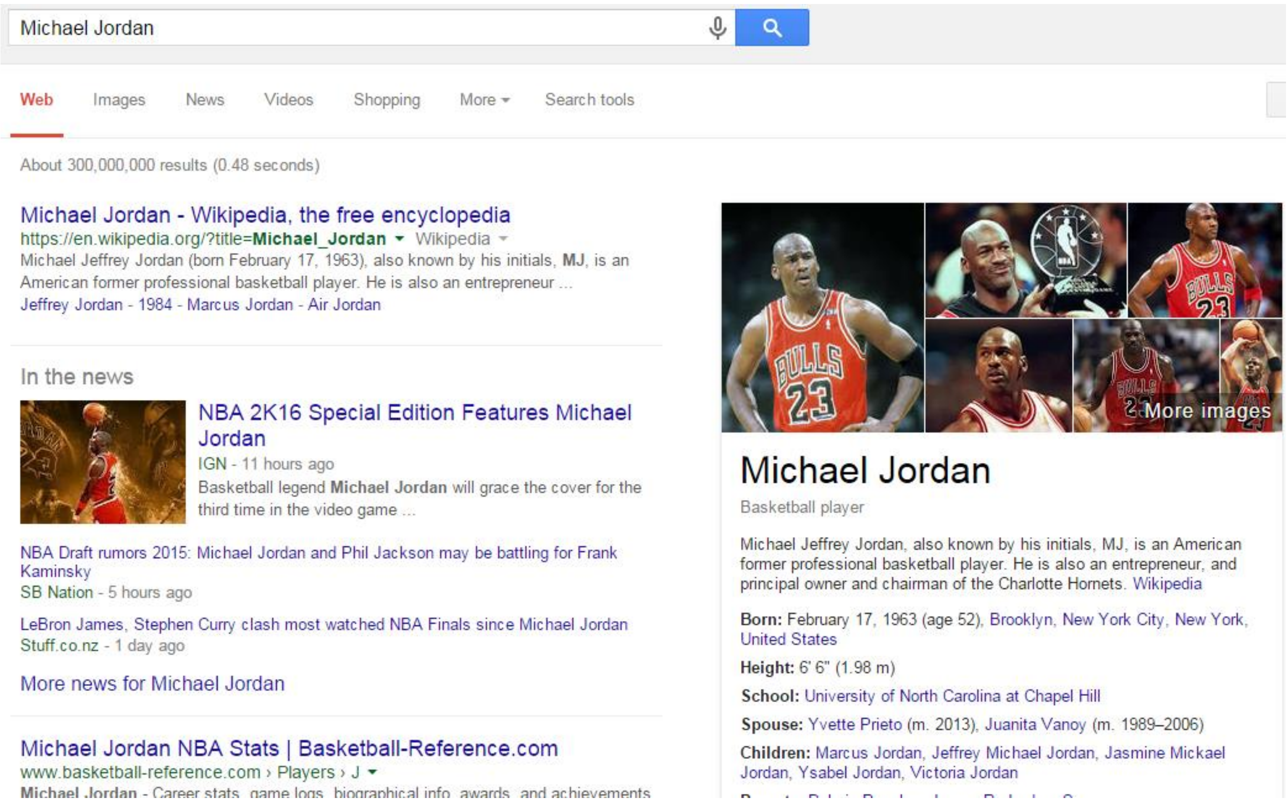


Relevance Ranking is Important



- Criteria: relevance
- Ranking model
  - Heuristic: VSM, BM25, LMIR
  - Learning to rank

Beyond Relevance Ranking



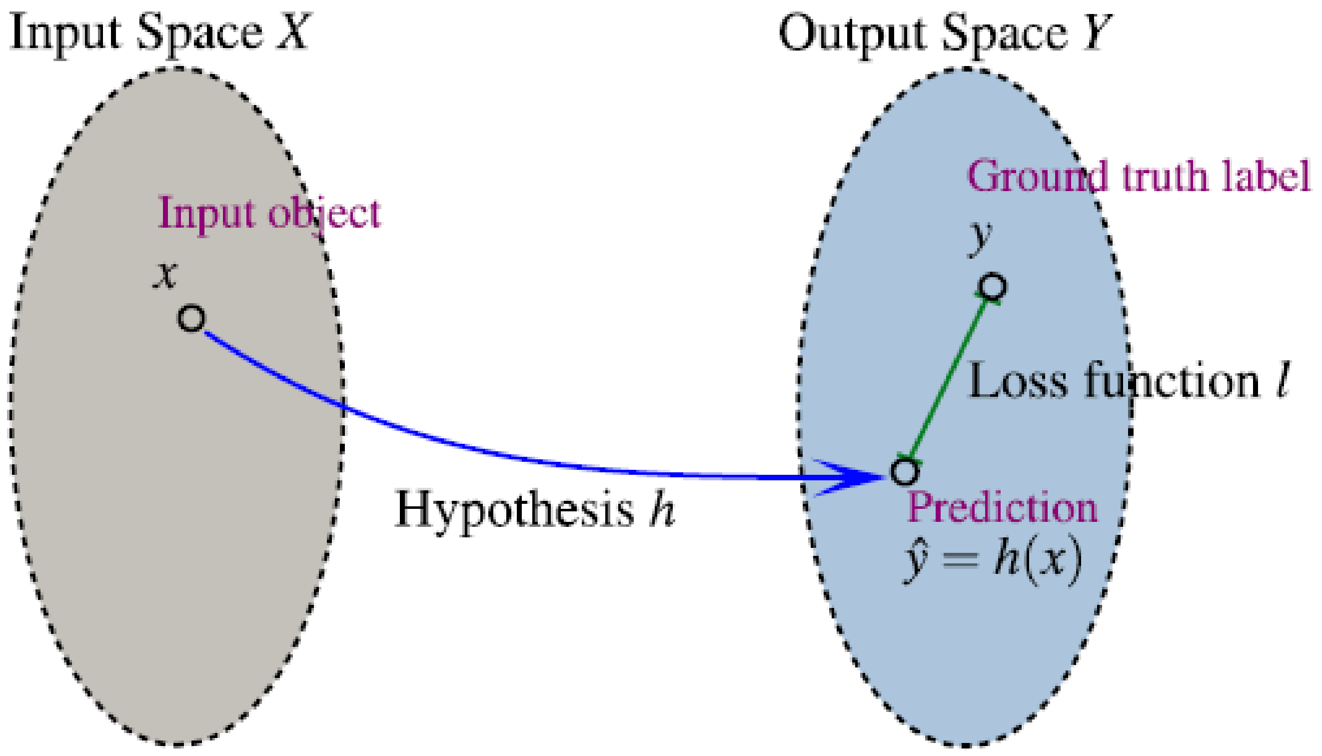
- Freshness
- Response time
- Diversification

Why Diversification?

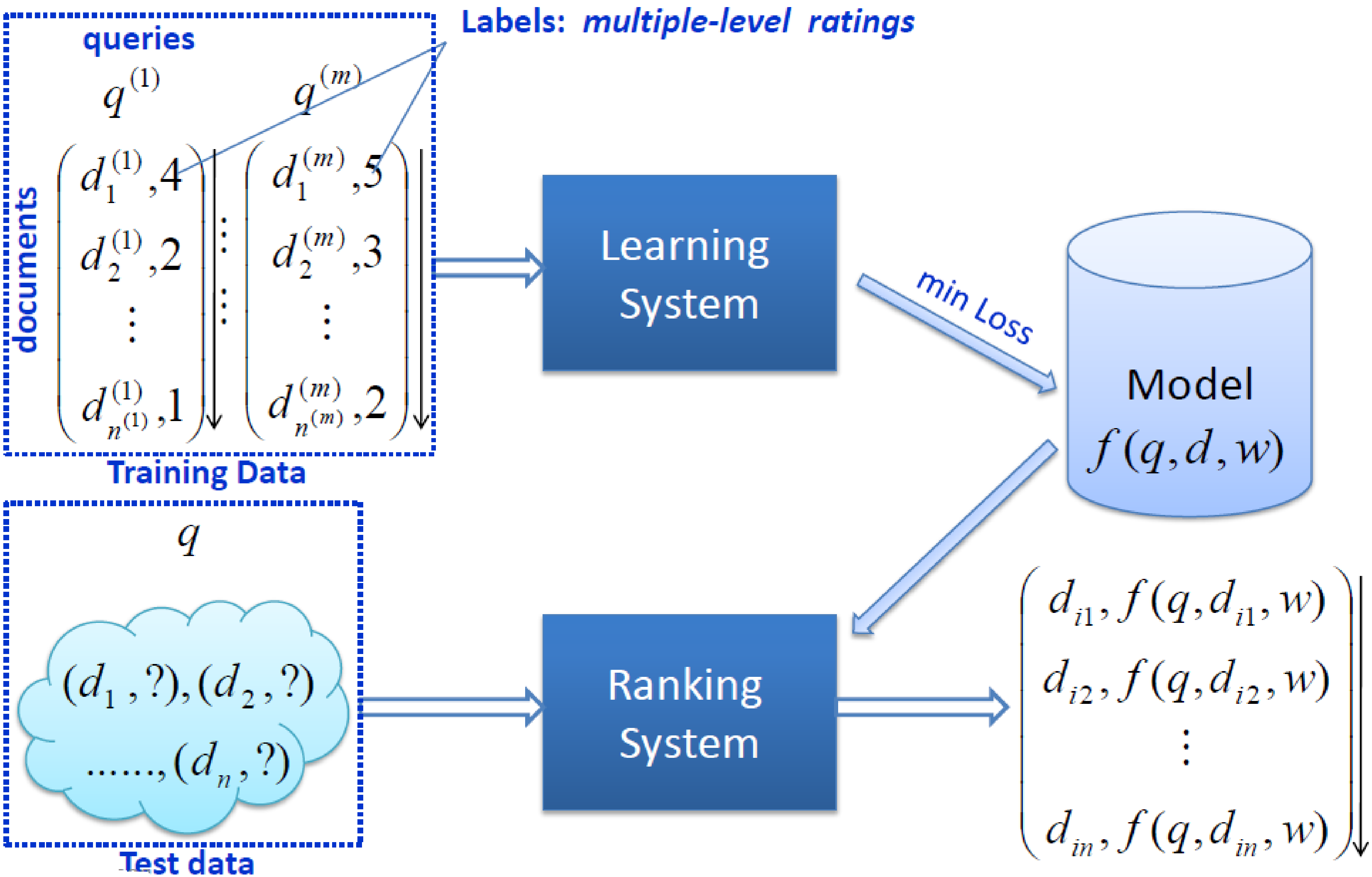
Outline

- Search result diversification
- Learning for search result diversification
  - Ranking model
  - Loss function and optimization
  - Experimental results
- Summary

Framework of Machine Learning



Bring Machine Learning to Ranking



Ranking Model for Relevance Learning to Rank

- Independent scoring function, e.g., $f(q,d) = \langle \vec{w}, \phi(q,d) \rangle$
- Features $\phi(q,d)$

Table 1: Relevance Features for learning on ClueWeb09-B collection [21, 19].

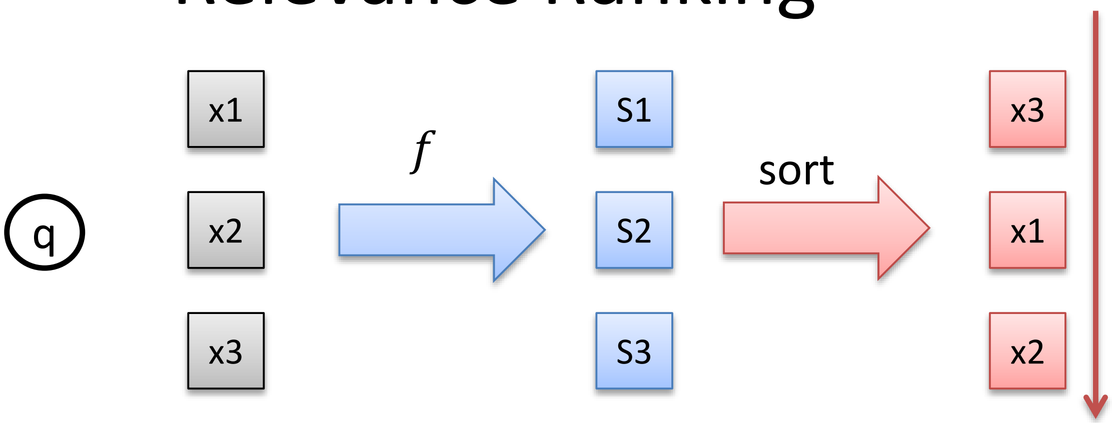
Category	Feature Description	Total
Q-D	TF-IDF	5
Q-D	BM25	5
Q-D	QL.DIR	5
Q-D	MRF	10
D	PageRank	1
D	#Inlinks	1
D	#Outlinks	1

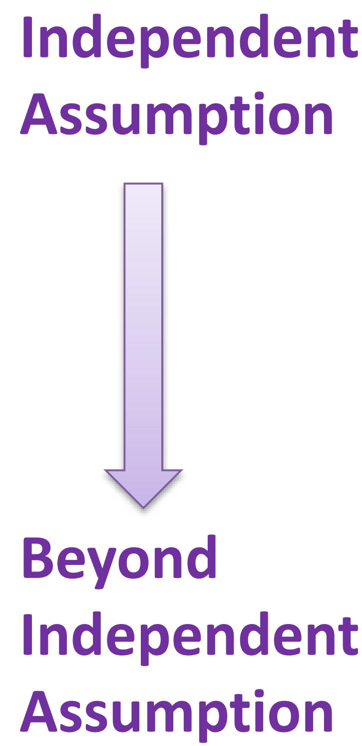
- Ranking: scoring and sorting ascendingly

Query	Documents	score
q	d1	$f(q,d1)$
	d2	$f(q,d2)$
	d3	$f(q,d3)$

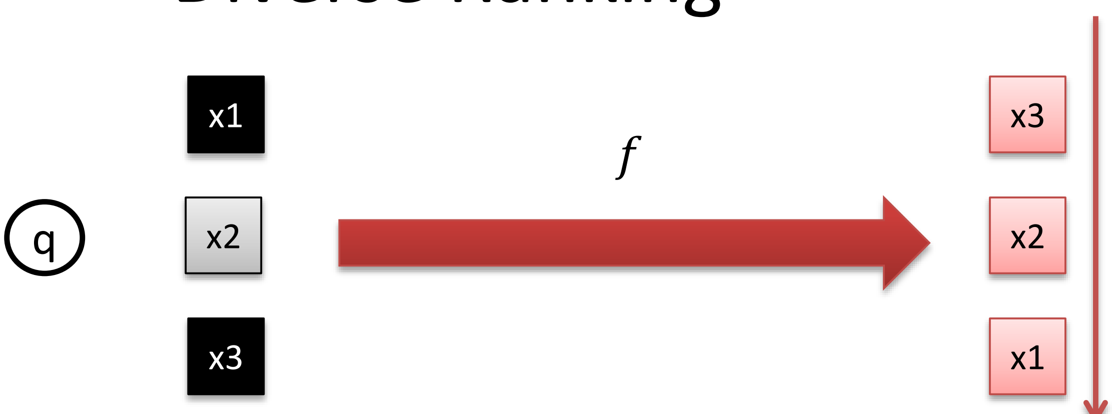
Beyond Relevance Ranking

- Relevance Ranking





- Diverse Ranking



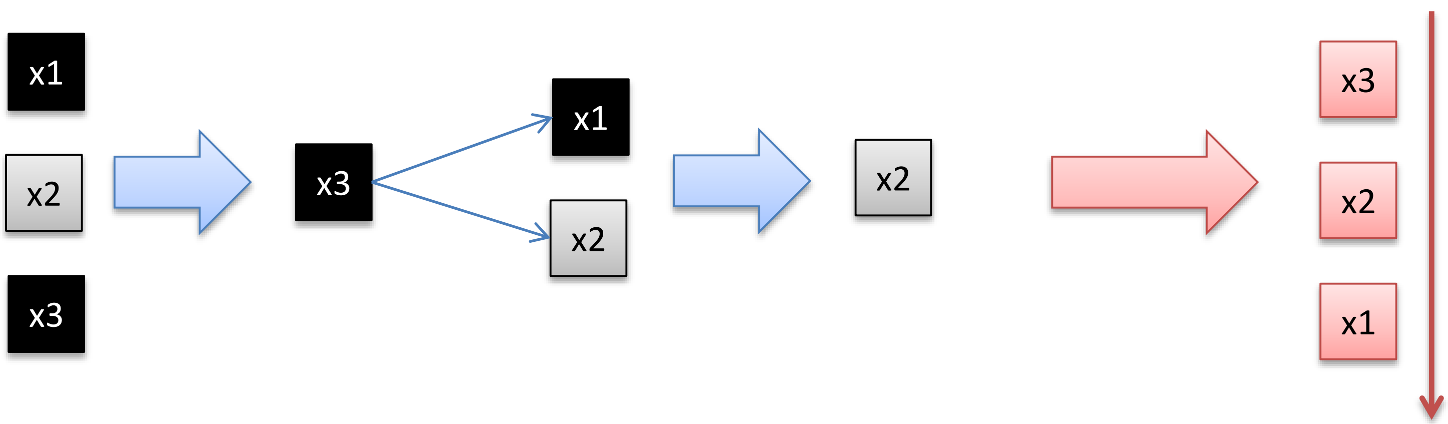
Maximal Marginal Relevance

- A greedy approach to search result diversification
  - Scoring function

$$MMR \stackrel{\text{def}}{=} \operatorname*{Arg\,max}_{D_i \in R \setminus S} \left[\lambda\, Sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j) \right]$$

Relevance Relation

  - Ranking: sequential selection procedure



Relational Learning to Rank (R-LTR) Model

- Scoring function for selecting a document

$$f_S(x_i, R_i) = \boxed{\omega_r^T \mathbf{x}_i} + \boxed{\omega_d^T h_S(R_i)}, \forall x_i \in X \setminus S$$

Selected documents

Content-based score

Relation-based score

- Ranking with sequential document selection

$$\mathbf{f}(X, R) = (f_{S_\emptyset}, f_{S_1}, \cdots, f_{S_{n-1}})$$

Relevance Features $\mathbf{x}_i$

- $f_S(x_i, R_i) = \langle w_r, \mathbf{x}_i \rangle + \langle w_d, h_S(R_i) \rangle, \forall x_i \in X \backslash S$
- Adopting features used in relevance learning to rank
  - Query-Document features
  - Document features

Relational (Diversity) Features

- $f_S(x_i, R_i) = \langle w_r, x_i \rangle + \langle w_d, h_S(R_i) \rangle, \forall x_i \in X \backslash S$
  - R: the relation cube, R_{ijk} is the k^{th} feature describing relation of document i and document j
  - R_i: the matrix describing the relationship of document i and other documents

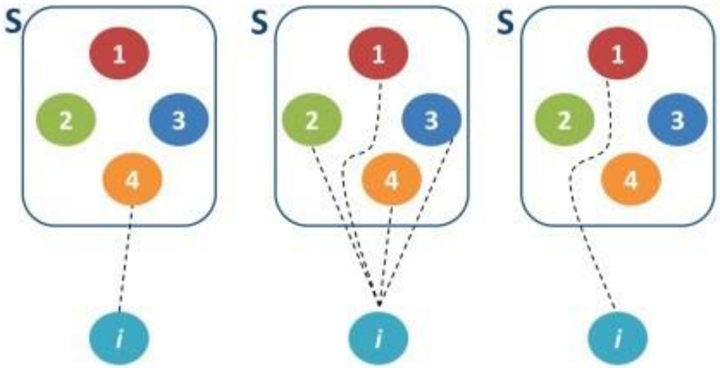
	feature 1	……	Feature k	…	
d1					
……					
dj			R_{ijk}		
….					

Definition of $h_S(R_i)$

vector

matrix

- $f_S(x_i, R_i) = \langle w_r, x_i \rangle + \langle w_d, h_S(R_i) \rangle, \forall x_i \in X \backslash S$
  - Minimal: $h_S(R_i) = \left\{ \min_{x_j \in S} R_{ij1}, \ldots, \min_{x_j \in S} R_{ijK} \right\}$
  - Mean: $h_S(R_i) = \left\{ \frac{1}{|S|} \sum_{x_j \in S} R_{ij1}, \ldots, \frac{1}{|S|} \sum_{x_j \in S} R_{ijK} \right\}$
  - Maximal: $h_S(R_i) = \left\{ \max_{x_j \in S} R_{ij1}, \ldots, \max_{x_j \in S} R_{ijK} \right\}$



(a) Minimal (b) Average (c) Maximal

Diversity Features

- Feature vector $[R_{ij1}, R_{ij2}, \cdots, R_{ijK}]$
  - Subtopic Diversity: based on PLSA $R_{ij1} = \sqrt{\sum_{k=1}^{m} (p(z_k|x_i) - p(z_k|x_j))^2}$
  - Text diversity: $R_{ij2} = 1 - \frac{\mathbf{d}_i \cdot \mathbf{d}_j}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|}$
  - Title diversity
  - Anchor diversity
  - Link-based diversity
  - URL-based diversity
  - ODP-based diversity

Relevance Ranking Model vs. Relational Ranking Model

Relevance ranking model

- A single scoring function
- Assigning document scores independently
- Sorting document ascendingly
- Relevance features

Relational ranking model

- N-1 scoring functions (sharing parameters) for ranking N documents
- Document relations are taken into consideration
- Sequential document selection
- Relevance features + diversity features (d-d)

Outline

- Search result diversification
- Learning for search result diversification
  - Ranking model
  - Loss function and optimization
  - Experimental results
- Summary

Relevance Learning to Rank

$$x \rightarrow y$$

Pointwise Methods

- **Regression，Order Regression**
- **OC SVM, McRank**

$$(x_1, x_2) \rightarrow y$$

Pairwise Methods

- **Pairwise classification**
- **RankSVM，RankBoost，RankNet，GBRank**

$$(x_1, x_2, \cdots, x_n) \rightarrow \vec{y}$$

Listwise Methods

- **Listwise ranking**
- **ListMLE，ListNet，RankCosine，StructureSVM，SoftRank，AdaRank**

Conventional Learning to Rank (cont')

- Pointwise

$$\sum_{q}\sum_{d\in\mathbf{d}} l(f(q,d),y)+\Omega(f)$$

- Pairwise

$$\sum_{q}\sum_{d\succ_q d'} l(f(q,d),y,f(q,d'),y')+\Omega(f)$$

- Listwise

$$\sum_{(q,\mathbf{d})} l(\mathbf{F}(q,\mathbf{d}),\mathbf{y})+\Omega(\mathbf{F})$$

Permutation over $\mathbf{d}$

Loss Functions for Search Result Diversification

- Generative approach [Zhu et al., SIGIR ‘14]
  - Modeling the generation of the result list in a sequential way
  - Using the Plackett-Luce model
  - Optimize with MLE
- Discriminative approach [Xia et al., SIGIR ‘15]
  - Maximizing margins between “positive” and “negative” rankings
  - Optimize with Perceptron (other methods can also be used)

Generative Approach [Zhu et al., SIGIR '14]

Initialize $S_0 \leftarrow \emptyset, \mathbf{y}^{(t)} = (1, \cdots, n_t)$
for $k = 1, ..., n_t$ **do**
  $\text{bestDoc} \leftarrow \text{argmax}_{x \in X_t} f_{S_{k-1}}(x, R)$
  $S_k \leftarrow S_{k-1} \cup \text{bestDoc}$
  $y^{(t)}(k) \leftarrow$ the *index* of bestDoc
end for
return $\mathbf{y}^{(t)} = (y^{(t)}(1), \cdots, y^{(t)}(n_t))$

Process of sequential document selection

Modeling the generation of the result list in a sequential way

Loss Function:
Negative log likelihood of generation probability

$$L(\mathbf{f}(X, R), \mathbf{y}) = -\log P(\mathbf{y}|X)$$

$$P(\mathbf{y}|X) = P\left(x_{y(1)}, x_{y(2)}, \cdots, x_{y(n)}|X\right)$$
$$= P\left(x_{y(1)}|X\right)P\left(x_{y(2)}|X\backslash S_1\right) \cdots P\left(x_{y(2)}|X\backslash S_{n-1}\right)$$

Loss Function of R-LTR

- Plackett-Luce Model

$$\mathbf{P}(\pi \mid \boldsymbol{v}) = \prod_{i=1}^{M} \frac{v_{\pi(i)}}{v_{\pi(i)} + v_{\pi(i+1)} + \ldots + v_{\pi(M)}}$$

- Detailed definition

$$P(x_{y(1)} \mid X) = \frac{\exp\{f_\phi(x_{y(1)})\}}{\sum_{k=1}^{n}\exp\{f_\phi(x_{y(k)})\}}, \quad P(x_{y(j)} \mid X \setminus S_{j-1}) = \frac{\exp\{f_{S_{j-1}}(x_{y(j)}, R_{y(j)})\}}{\sum_{k=j}^{n}\exp\{f_{S_{k-1}}(x_{y(k)}, R_{y(k)})\}}.$$

- maximize the sum of the likelihood function

$$-\sum_{i=1}^{N}\sum_{j=1}^{n_i}\log\left\{\frac{\exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\sum_{k=j}^{n_i}\exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}}\right\}$$

Optimization

- Stochastic gradient ascent

$$\Delta\omega_r^{(i)} = \sum_{j=1}^{n_i} \left\{ \frac{\sum_{k=j}^{n_i} \mathbf{x}_{y(k)}^{(i)} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}}{\sum_{k=j}^{n_i} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}} - \frac{\mathbf{x}_{y(j)}^{(i)} \exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}} \right\},$$

$$\Delta\omega_d^{(i)} = \sum_{j=1}^{n_i} \left\{ \frac{\sum_{k=j}^{n_i} h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)}) \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}}{\sum_{k=j}^{n_i} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}} - \frac{h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)}) \exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}} \right\}.$$

R-LTR Algorithm

Algorithm 2 Optimization Algorithm

Input: training data $\{(X^{(i)}, R^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$,
parameter: learning rate η, tolerance rate ϵ

Output: model vector: ω_r, ω_d

1: Initialize parameter value ω_r, ω_d
2: **repeat**
3: Shuffle the training data
4: **for** $i = 1, ..., N$ **do**
5: Compute gradient $\Delta\omega_r{}^{(i)}$ and $\Delta\omega_d{}^{(i)}$
6: Update model: $\omega_r = \omega_r - \eta \times \Delta\omega_r{}^{(i)}$,
$\omega_d = \omega_d - \eta \times \Delta\omega_d{}^{(i)}$
7: **end for**
8: Calculate likelihood loss on the training set
9: **until** the change of likelihood loss is below ϵ

Can R-LTR be Further Improved?

- R-LTR only utilizes “positive” rankings
  - Discriminative learning is effective in many machine learning tasks
- Not all “negative rankings” are equally negative
  - Can be measured with evaluation measures
- There exists a number of diversity evaluation measures such as α-NDCG, ERR-IA etc.

Directly optimizing performance measures

Discriminative Approach [Xia et al., SIGIR ’15]

$$\min_{\omega_r,\omega_d} \sum_{n=1}^{N} L\left(\hat{\mathbf{y}}^{(n)}, J^{(n)}\right)$$

$\hat{\mathbf{y}}^{(n)}$: predicted ranking
$J^{(n)}$: ground truth

$$\sum_{n=1}^{N} \left(1 - E(X^{(n)}, \hat{\mathbf{y}}^{(n)}, J^{(n)})\right)$$

E: evaluation measure

Upper bounded

$$\sum_{n=1}^{N} \max_{\substack{\mathbf{y}^+ \in \mathcal{Y}^{+(n)};\\ \mathbf{y}^- \in \mathcal{Y}^{-(n)}}} \left(E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)})\right)$$
$$\left[\!\left[F(\mathbf{y}^+, X^{(n)}, R^{(n)}) \le F(\mathbf{y}^-, X^{(n)}, R^{(n)}) \right]\!\right]$$

$\mathcal{Y}^{+(n)}$: positive rankings
$\mathcal{Y}^{-(n)}$: negative rankings
F: ranking model

$$\hat{\mathbf{y}}^{(n)} = \arg\max_{\mathbf{y} \in \mathcal{Y}^{(n)}} F(X^{(n)}, R^{(n)}, \mathbf{y})$$

Upper bounded if $E \in [0,1]$

$$\sum_{n=1}^{N} \sum_{\mathbf{y}^+;\mathbf{y}^-} \left[\!\left[F(X^{(n)}, R^{(n)}, \mathbf{y}^+) - F(X^{(n)}, R^{(n)}, \mathbf{y}^-) \le E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)}) \right]\!\right].$$

$$\begin{aligned}
F(X, R, \mathbf{y}) &= \Pr(\mathbf{y}|X, R) \\
&= \Pr(\mathbf{x}_{y(1)} \cdots \mathbf{x}_{y(M)}|X, R) \\
&= \prod_{r=1}^{M-1} \Pr(\mathbf{x}_{y(r)}|X, S_{r-1}, R) \\
&= \prod_{r=1}^{M-1} \frac{\exp\{f_{S_{r-1}}(\mathbf{x}_i, R_{y(r)})\}}{\sum_{k=r}^{M} \exp\{f_{S_{r-1}}(\mathbf{x}_i, R_{y(k)})\}}
\end{aligned}$$

Optimization with Perceptron (PAMM)

Algorithm 2 The PAMM Algorithm

Input: training data $\{(X^{(n)}, R^{(n)}, J^{(n)})\}_{n=1}^{N}$, learning rate η, diversity evaluation measure E, number of positive rankings per query τ^+, number of negative rankings per query τ^-.

Output: model parameters (ω_r, ω_d)

for $n = 1$ **to** N **do**
  $PR^{(n)} \leftarrow$ PositiveRankings$(X^{(n)}, J^{(n)}, E, \tau^+)$ {Algorithm 3}
  $NR^{(n)} \leftarrow$ NegativeRankings$(X^{(n)}, J^{(n)}, E, \tau^-)$ {Algorithm 4}
end for
initialize $\{\omega_r, \omega_d\} \leftarrow$ random values in $[0, 1]$
repeat
  for $n = 1$ **to** N **do**
    for all $\{\mathbf{y}^+, \mathbf{y}^-\} \in PR^{(n)} \times NR^{(n)}$ **do**
      $\Delta F \leftarrow F(X^{(n)}, R^{(n)}, \mathbf{y}^+) - F(X^{(n)}, R^{(n)}, \mathbf{y}^-)$ {$F(X, R, \mathbf{y})$ is defined in Equation (8)}
      if $\Delta F \leq E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)})$ **then**
        calculate $\nabla\omega_r^{(n)}$ and $\nabla\omega_d^{(n)}$ {Equation (10) and Equation (11)}
        $(\omega_r, \omega_d) \leftarrow (\omega_r, \omega_d) + \eta \times (\nabla\omega_r^{(n)}, \nabla\omega_d^{(n)})$
      end if
    end for
  end for
until convergence
return (ω_r, ω_d)

Margin between "positive" and "negative" rankings

Advantages of PAMM

- Online ranking
  - Meets the MMR criteria
- Offline learning
  - Can directly optimizing any diversity evaluation measures
  - Discriminative approach: ability to use both positive rankings and negative rankings
  - Maximizing the margins between positive rankings and negative rankings

Outline

- Search result diversification
- Learning for search result diversification
  - Ranking model
  - Loss function and optimization
  - Experimental results
- Summary

Datasets

- TREC datasets: WT2009, WT2010, and WT2011
- Query

```
▼<topic number="1" type="faceted">
   <query>obama family tree</query>
  ▼<description>
     Find information on President Barack Obama's family history, including genealogy, national origins,
   </description>
  ▼<subtopic number="1" type="nav">
     Find the TIME magazine photo essay "Barack Obama's Family Tree".
   </subtopic>
  ▼<subtopic number="2" type="inf">
     Where did Barack Obama's parents and grandparents come from?
   </subtopic>
  ▼<subtopic number="3" type="inf">
     Find biographical information on Barack Obama's mother.
   </subtopic>
 </topic>
```

- Label

	Subtopic 1	Subtopic 2	Subtopic 3
Document 1	0	1	0
Document 2	1	1	0
Document 3	0	0	0
……			

Evaluation Measures

- α-NDCG

$$\alpha\text{-NDCG@}k = \frac{\sum_{r=1}^{k} NG(r)/\log(r+1)}{\sum_{r=1}^{k} NG^{*}(r)/\log(r+1)},$$

$$NG(r) = \sum_{s} J(y(r), s)(1-\alpha)^{C_s(r-1)}$$

$$C_s(r-1) = \sum_{k=1}^{r-1} J(y(k), s)$$

- ERR-IA

$$\text{ERR-IA@}k = \sum_{s} \Pr(s|q)\text{ERR@}k(s),$$

where $\text{ERR@}k(s)$ is the expected reciprocal rank score at k in terms of subtopic s.

Experimental Results

Table 5: Performance comparison of all methods in official TREC diversity measures for WT2009.

Method	ERR-IA@20	α-NDCG@20
QL	0.164	0.269
ListMLE	0.191(+16.46%)	0.307(+14.13%)
MMR	0.202(+23.17%)	0.308(+14.50%)
xQuAD	0.232(+41.46%)	0.344(+27.88%)
PM-2	0.229(+39.63%)	0.337(+25.28%)
SVM-DIV	0.241(+46.95%)	0.353(+31.23%)
StructSVM(α-NDCG)	0.260(+58.54%)	0.377(+40.15%)
StructSVM(ERR-IA)	0.261(+59.15%)	0.373(+38.66%)
R-LTR	0.271(+65.24%)	0.396(+47.21%)
PAMM(α-NDCG)	0.284(+73.17%)	**0.427**(+58.74%)
PAMM(ERR-IA)	**0.294**(+79.26%)	0.422(+56.88%)

Table 6: Performance comparison of all methods in official TREC diversity measures for WT2010.

Method	ERR-IA@20	α-NDCG@20
QL	0.198	0.302
ListMLE	0.244(+23.23%)	0.376(+24.50%)
MMR	0.274(+38.38%)	0.404(+33.77%)
xQuAD	0.328(+65.66%)	0.445(+47.35%)
PM-2	0.330(+66.67%)	0.448(+48.34%)
SVM-DIV	0.333(+68.18%)	0.459(+51.99%)
StructSVM(α-NDCG)	0.352(+77.78%)	0.476(+57.62%)
StructSVM(ERR-IA)	0.355(+79.29%)	0.472(+56.29%)
R-LTR	0.365(+84.34%)	0.492(+62.91%)
PAMM(α-NDCG)	0.380(+91.92%)	**0.524**(+73.51%)
PAMM(ERR-IA)	**0.387**(+95.45%)	0.511(+69.21%)

Table 7: Performance comparison of all methods in official TREC diversity measures for WT2011.

Method	ERR-IA@20	α-NDCG@20
QL	0.352	0.453
ListMLE	0.417(+18.47%)	0.517(+14.13%)
MMR	0.428(+21.59%)	0.530(+17.00%)
xQuAD	0.475(+34.94%)	0.565(+24.72%)
PM-2	0.487(+38.35%)	0.579(+27.81%)
SVM-DIV	0.490(+39.20%)	0.591(+30.46%)
StructSVM(α-NDCG)	0.512(+45.45%)	0.617(+36.20%)
StructSVM(ERR-IA)	0.513(+45.74%)	0.613(+35.32%)
R-LTR	0.539(+53.13%)	0.630(+39.07%)
PAMM(α-NDCG)	0.541(+53.70%)	**0.643**(+41.94%)
PAMM(ERR-IA)	**0.548**(+55.68%)	0.637(+40.62%)

- R-LTR outperforms other baselines
- PAMM performs better than R-LTR
- PAMM can directly optimize the evaluation measure

Effect of MMR

ranking positions

		1	2	3	4	5	α-NDCG@5
StructSVM		2, 4	1, 4	2	1, 3	4	0.788
PAMM intermediate rankings	f_{S_0}	**2, 4**	2	4	1, 3	1, 4	0.744
	f_{S_1}	2, 4	**1, 3**	2	4	1, 4	0.803
	f_{S_2}	2, 4	1, 3	**1, 4**	4	2	0.812
	f_{S_3}	2, 4	1, 3	1, 4	**2**	4	0.815

Figure 1: Example rankings from WT2009. Each shaded block represents a document and the number(s) in the block represent the subtopic(s) covered by the document.

Ability to Improve Evaluation Measures

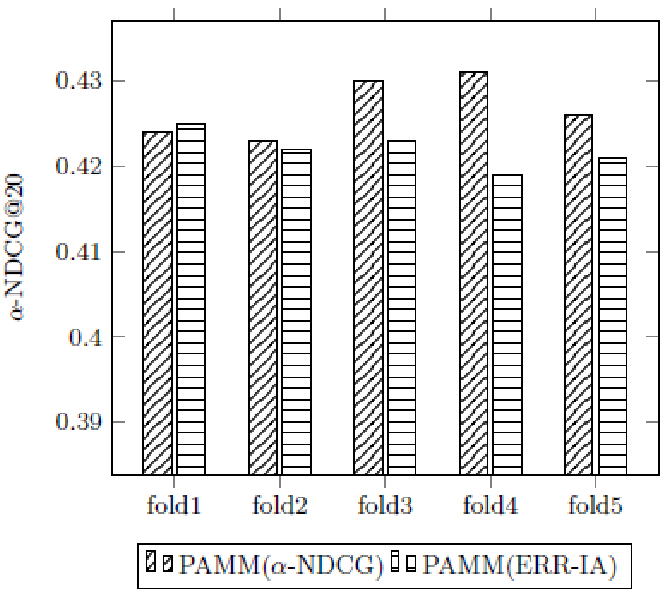


Figure 2: Performance in terms of α-NDCG@20 when model is trained with α-NDCG@20 or ERR-IA@20.

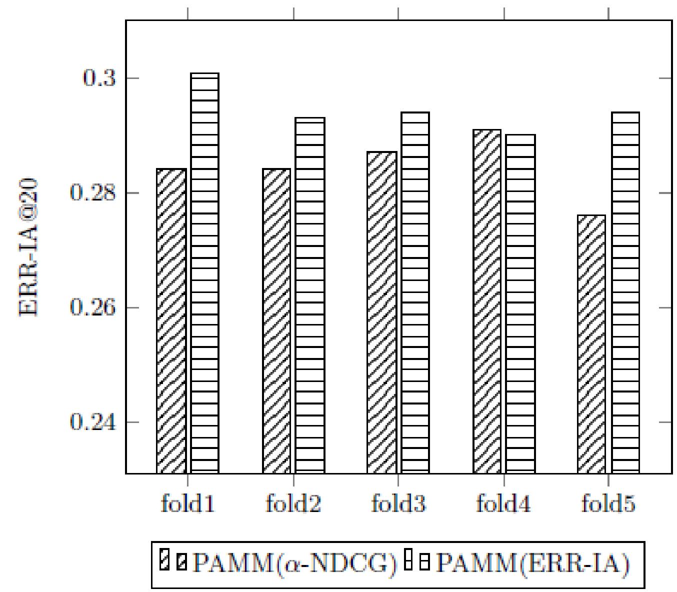


Figure 3: Performance in terms of ERR-IA@20 when model is trained with α-NDCG@20 or ERR-IA@20.

Effects of Positive and Negative Rankings

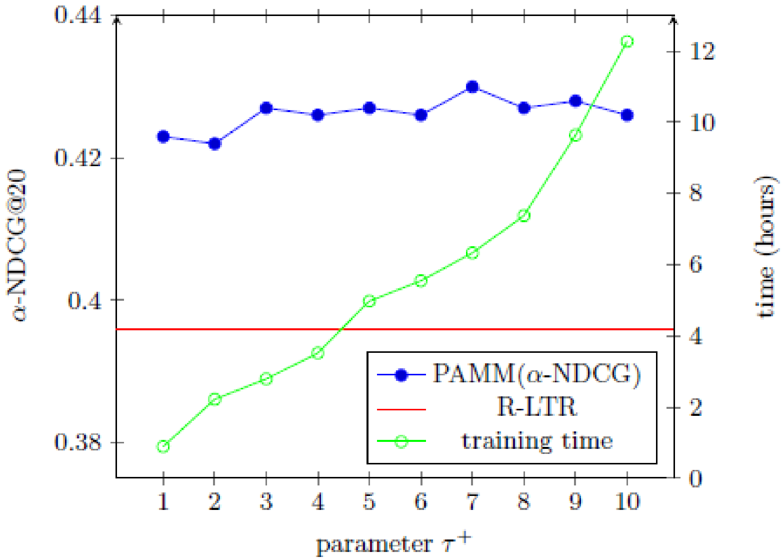


Figure 4: Ranking accuracies and training time w.r.t. τ^{+}.

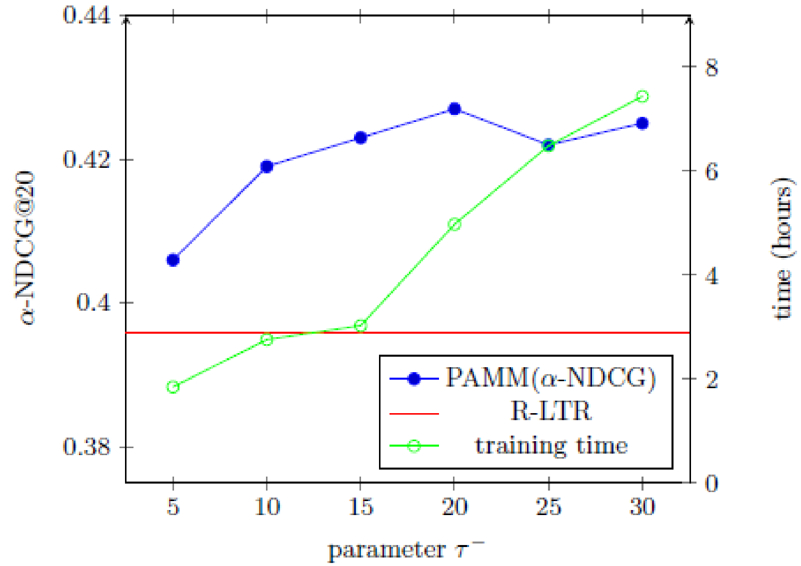


Figure 5: Ranking accuracies and training time w.r.t τ^{-}.

Convergence of PAMM

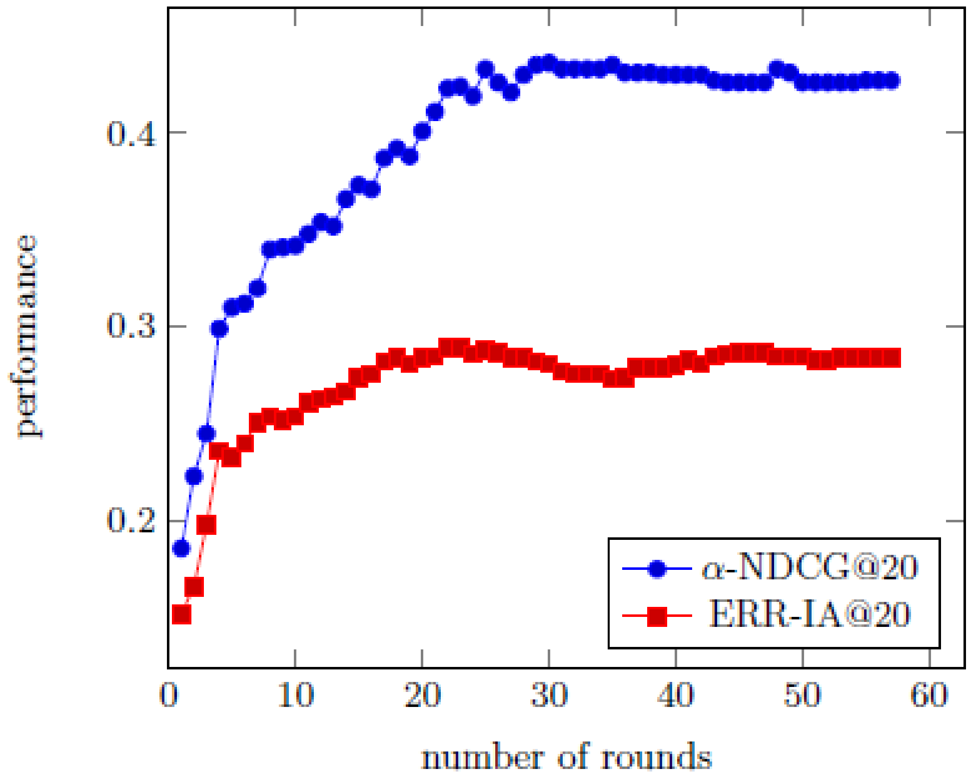


Figure 7: Learning curve of PAMM(α-NDCG).

Outline

- Search result diversification
- Learning for search result diversification
- Summary

Summary

- New learning to rank models for search result diversification
  - Model: following the MMR criteria
  - Generative learning [Zhu et al., SIGIR ’14]
    - Modeling generation process with Luce model
    - Optimizing with MLE
  - Discriminative learning [Xia et al., SIGIR ’15]
    - Directly optimizing evaluation measures
    - Utilizing both positive rankings and negative rankings
    - Optimizing with structured Perceptron

Future Directions

- Learning to rank is not hot in recent years
  - However, a lot of issues not addressed
- New applications: beyond independent relevance
  - Diversification
  - Whole page relevance
  - Topic distillation
- Addressing issues in existing (famous) algorithms
  - E.g., IID assumption in pairwise ranking algorithms such as Ranking SVM does not hold in real world data [Zhang et al., CIKM ’15]
- New modeling and optimization tools
  - Deep neural networks?
  - ADMM for large scale learning?

References

- R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. WSDM 2009.
- Jaime Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. SIGIR 1998.
- O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. CIKM 2009.
- C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Buttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. SIGIR 2008.
- T.-Y. Liu. Learning to Rank for Information Retrieval. Springer, 2011.
- **Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. SIGIR 2015**.
- C. X. Zhai, W. W. Cohen, and J. Laerty. Beyond independent relevance: methods and evaluation metrics for subtopic SIGIR, 2003.
- Yaogong Zhang, Jun Xu, Yanyan Lan, Jiafeng Guo, Maoqiang Xie, Yalou Huang, and Xueqi Cheng. Modeling Parameter Interactions in Ranking SVM. CIKM 2015, short paper.
- **Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. Learning for Search Result Diversification. SIGIR 2014.**

Thanks! Q&A

Homepage:
www.bigdatalab.ac.cn/~junxu