



# Learning to Rank: from Pairwise Approach to Listwise Approach

Tie-Yan Liu

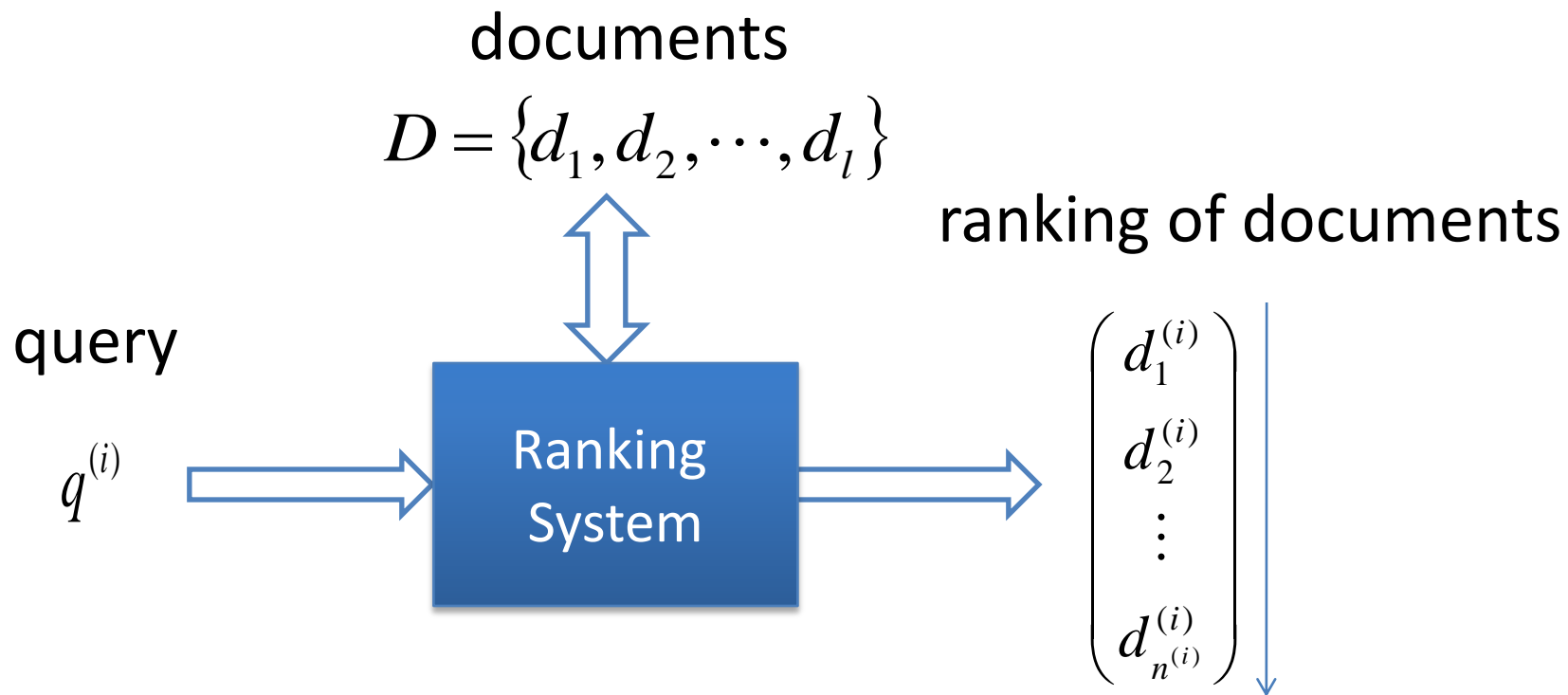
Lead Researcher, Microsoft Research Asia

<http://research.microsoft.com/users/tyliu/>

# Talk Outline

- Introduction to Learning to Rank
- Previous work: Pairwise Approach
- Our proposal: Listwise Approach
  - ListNet
  - Relational Ranking
- Summary

# Ranking Problem: Example = Information Retrieval



*Ranking is also important in NLP applications, such as first-pass attachment disambiguation, and reranking alternative parse trees generated for the same sentence by a statistical parser, etc.*

# IR Evaluation Measures

- Precision at position  $n$

$$P @ n = \frac{\#\{\text{relevant documents in top } n \text{ results}\}}{n}$$

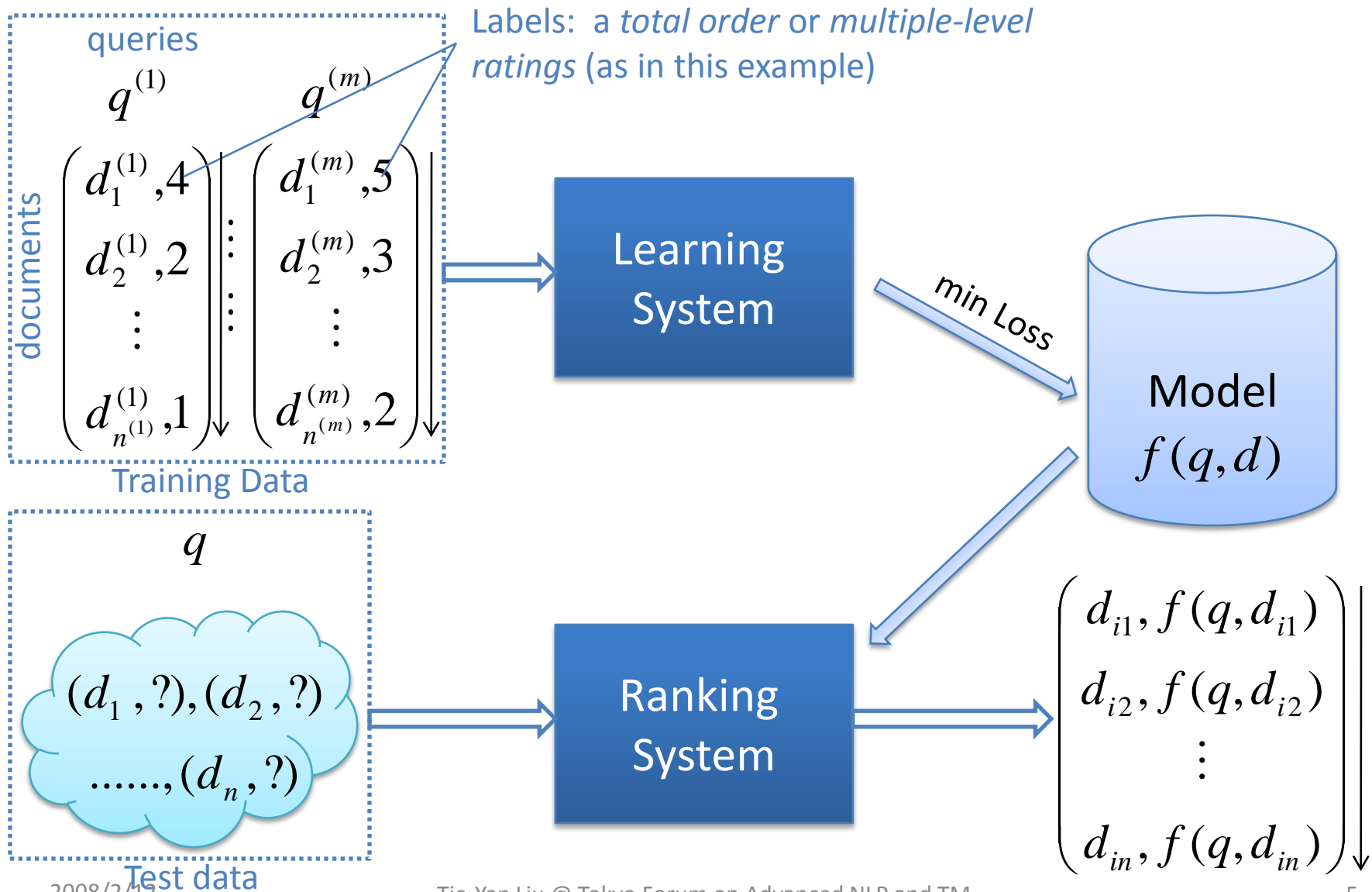
- Average precision

$$AP = \frac{\sum P @ n \cdot I\{\text{document } n \text{ is relevant}\}}{\#\{\text{relevant documents}\}}$$

- MAP: averaged over all queries in the test set

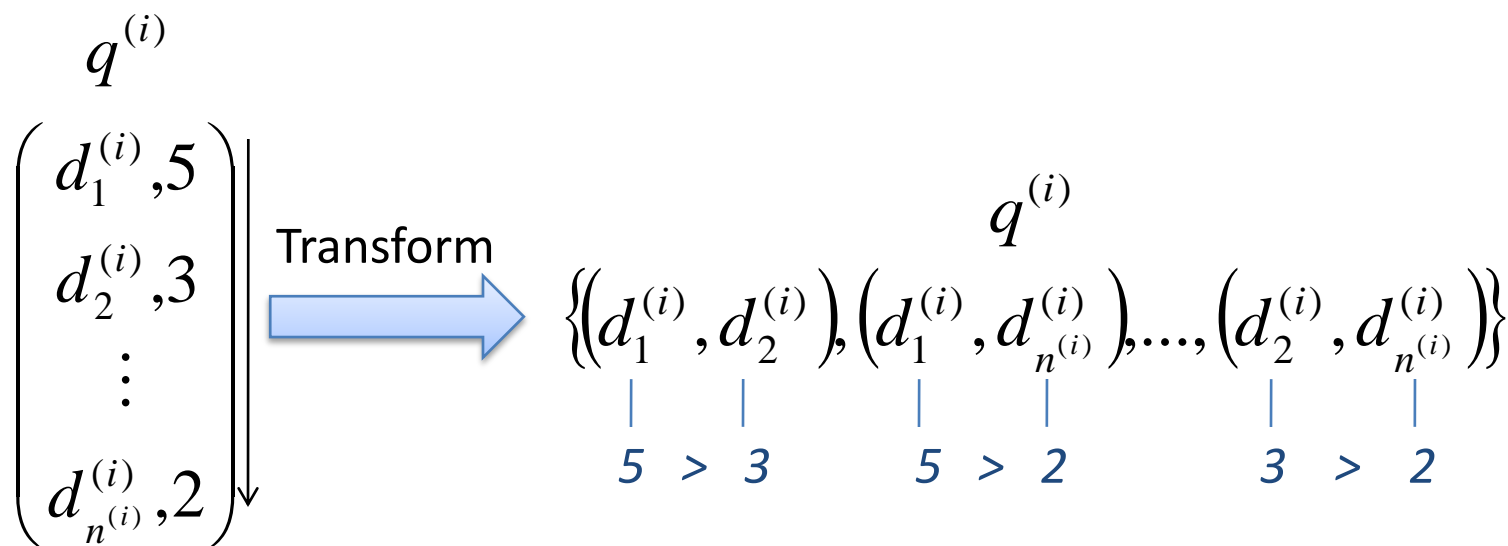
- NDCG at position  $n$ :  $N @ n = Z_n \sum_{j=1}^n (2^{r(j)} - 1) / \log(1 + j)$

# Learning to Rank



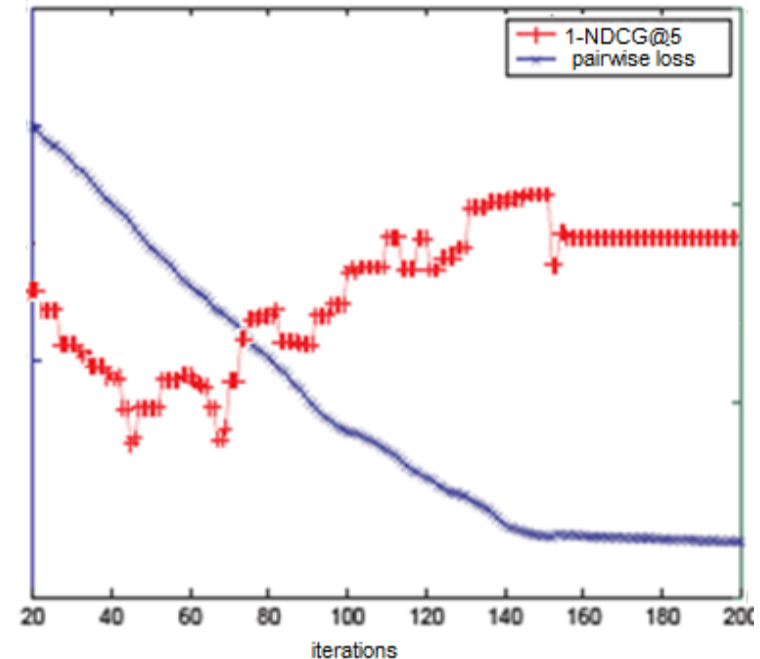
# Previous Work: Pairwise Approach

- Transforming ranked list into document pairs
- Formalizing ranking as classification on document pairs
- Ranking SVM, RankNet, RankBoost



# Problems with Pairwise Approach

- Loss function is suboptimal
  - Not to optimize evaluation measures (e.g. NDCG and MAP)
  - Not to consider position in ranking and number of documents per query
- Ranking function cannot represent relational information
- Generalization theory is limited



Pairwise loss vs. (1-NDCG@5)  
TREC Dataset

# Our Proposal: Listwise Approach

- Listwise Loss Function
  - Using permutation probability distribution to represent a ranked list
  - Using KL-divergence between permutation probability distributions to define loss function
- Learning to rank relational objects
  - Embed object relationship in the ranking function
  - Formalize ranking function as a new optimization problem.
- Query-level Generalization Analysis

# ListNet: A Listwise Loss Function for Learning to Rank

(ICML, 2007)

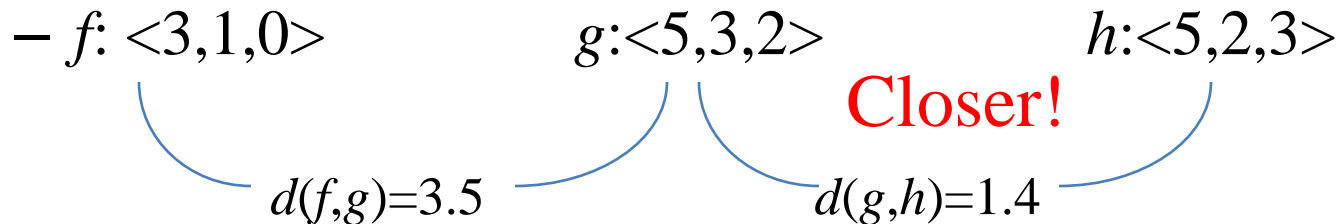
# Distance between Ranked Lists

- A Simple Example:

- function  $f$ :  $f(A)=3, f(B)=1, f(C)=0$       ABC
- function  $h$ :  $h(A)=5, h(B)=2, h(C)=3$       ACB
- ground truth  $g$ :  $g(A)=5, g(B)=3, g(C)=2$       ABC

- Question: which function is closer to ground truth?

- Euclidian distance between score vectors?

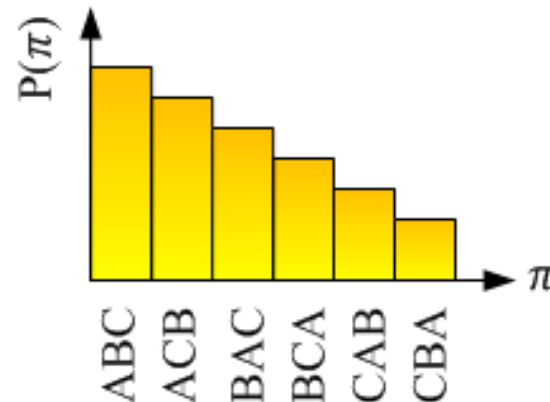
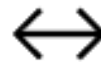


- However, ranked lists given by  $f$  and  $g$  are exactly the same!

# Representation of Ranked List

- Question:
  - How to represent a ranked list?
- Our proposal:
  - Ranked list  $\leftrightarrow$  Permutation probability distribution

$f: f(A)=3, f(B)=1, f(C)=0;$   
Ranking by  $f$ : ABC



# Permutation Probability

- Probability of permutation  $\pi$  is defined as (Luce Model)

$$P_s(\pi) = \prod_{j=1}^n \frac{\exp(s_{\pi(j)})}{\sum_{k=j}^n \exp(s_{\pi(k)})}$$

- Example:

$$P_f(\text{ABC}) = \frac{\exp(f(A))}{\exp(f(A)) + \exp(f(B)) + \exp(f(A))} \cdot \frac{\exp(f(B))}{\exp(f(B)) + \exp(f(C))} \cdot 1$$

**P(A ranked No.1)**

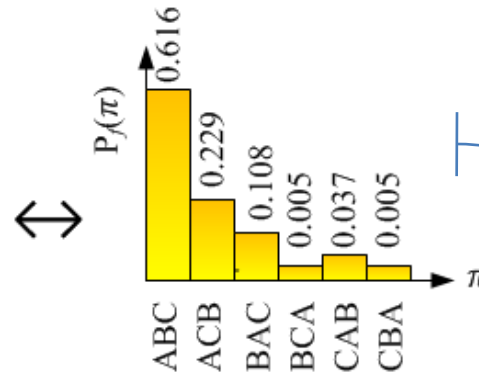
**P(B ranked No.2 | A ranked No.1)**  
**= P(B ranked No.1)/(1- P(A ranked No.1))**

**P(C ranked No.3 | A ranked No.1, B ranked No.2)**

# Distance between Ranked Lists

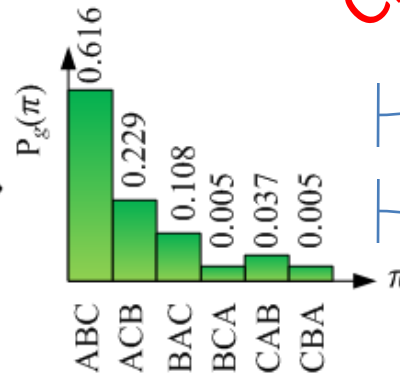
$\varphi = \exp$

$f: f(A) = 3, f(B) = 1, f(C) = 0;$   
 Ranking by  $f$ : ABC



Using *KL-divergence* to measure difference between distributions

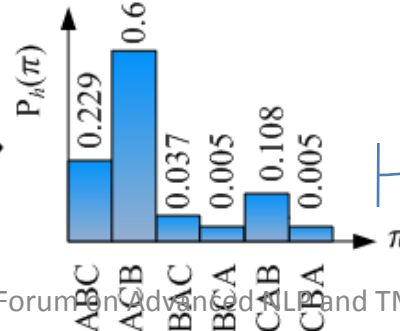
$g: g(A) = 5, g(B) = 3, g(C) = 2;$   
 Ranking by  $g$ : ABC



**Closer!**

$d(f, g) = 0$

$h: h(A) = 5, h(B) = 2, h(C) = 3;$   
 Ranking by  $h$ : ACB



$d(g, h) = 0.5$

# Permutation Probability Loss

- Formulation

$$L(w) \propto - \sum_{q \in Q} \sum_{G(j_1, \dots, j_k)} \left( \prod_{t=1}^n \frac{\exp(s_{y(j_t)})}{\sum_{u=t}^n \exp(s_{y(j_u)})} \right) \log \left( \prod_{t=1}^n \frac{\exp(w \cdot X_{y(j_t)})}{\sum_{u=t}^n \exp(w \cdot X_{y(j_u)})} \right)$$

- Properties

- Continuous and Differentiable

- Convex

- Log of a convex function is still convex, and the set of convex functions is closed under addition.

- Consistent

- In the large sample limit, minimizing the proposed listwise loss can achieve the optimal Bayes error rate with respect to 0-1 loss.

# Top-k Probability

- Correctly ranking top- $k$  documents is more critical
- Computation of Permutation Probability is intractable
- Top- $k$  Probability

– Defining Top- $k$  subgroup  $G(j_1, \dots, j_k)$  containing all permutations whose top- $k$  documents are  $j_1, \dots, j_k$

$$-P_s(G(j_1, \dots, j_k)) = \prod_{t=1}^k \frac{\exp(s_{\pi(j_t)})}{\sum_{u=t}^n \exp(s_{\pi(j_u)})}$$

– Time complexity of computation : from  $n!$  to  $n!/(n-k)!$

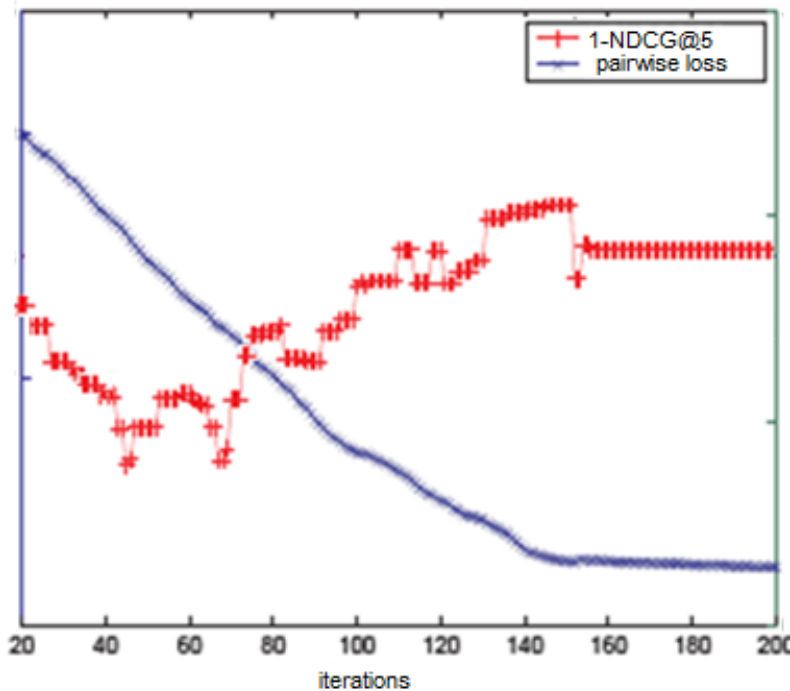
# ListNet Method

- Loss function = KL-divergence between two Top- $k$  probability distributions

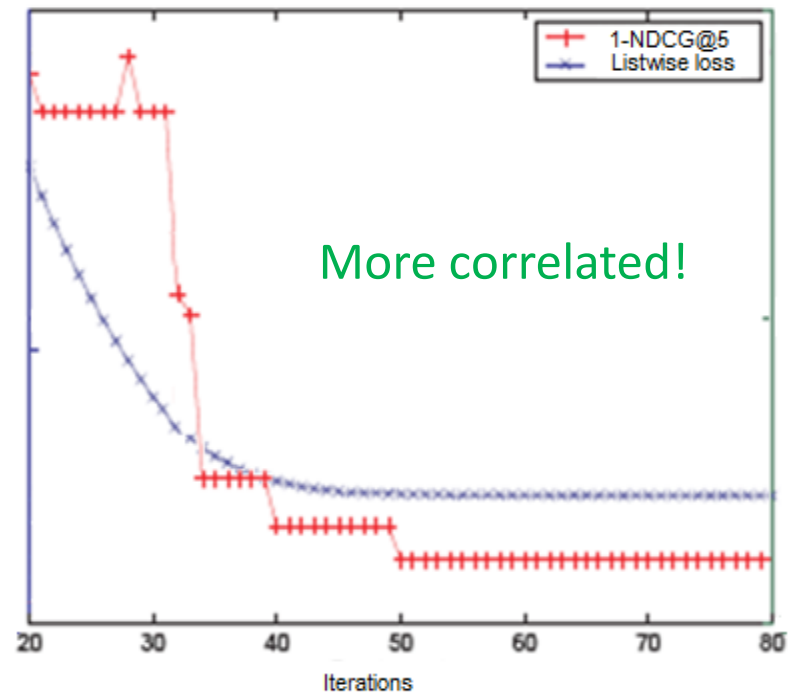
$$L(w) \propto - \sum_{q \in Q} \sum_{G(j_1, \dots, j_k)} \left( \prod_{t=1}^k \frac{\exp(s_{y(j_t)})}{\sum_{u=t}^n \exp(s_{y(j_u)})} \right) \log \left( \prod_{t=1}^k \frac{\exp(w \cdot X_{y(j_t)})}{\sum_{u=t}^n \exp(w \cdot X_{y(j_u)})} \right)$$

- Model = Neural Network
- Algorithm = Stochastic Gradient Descent

# Experimental Results



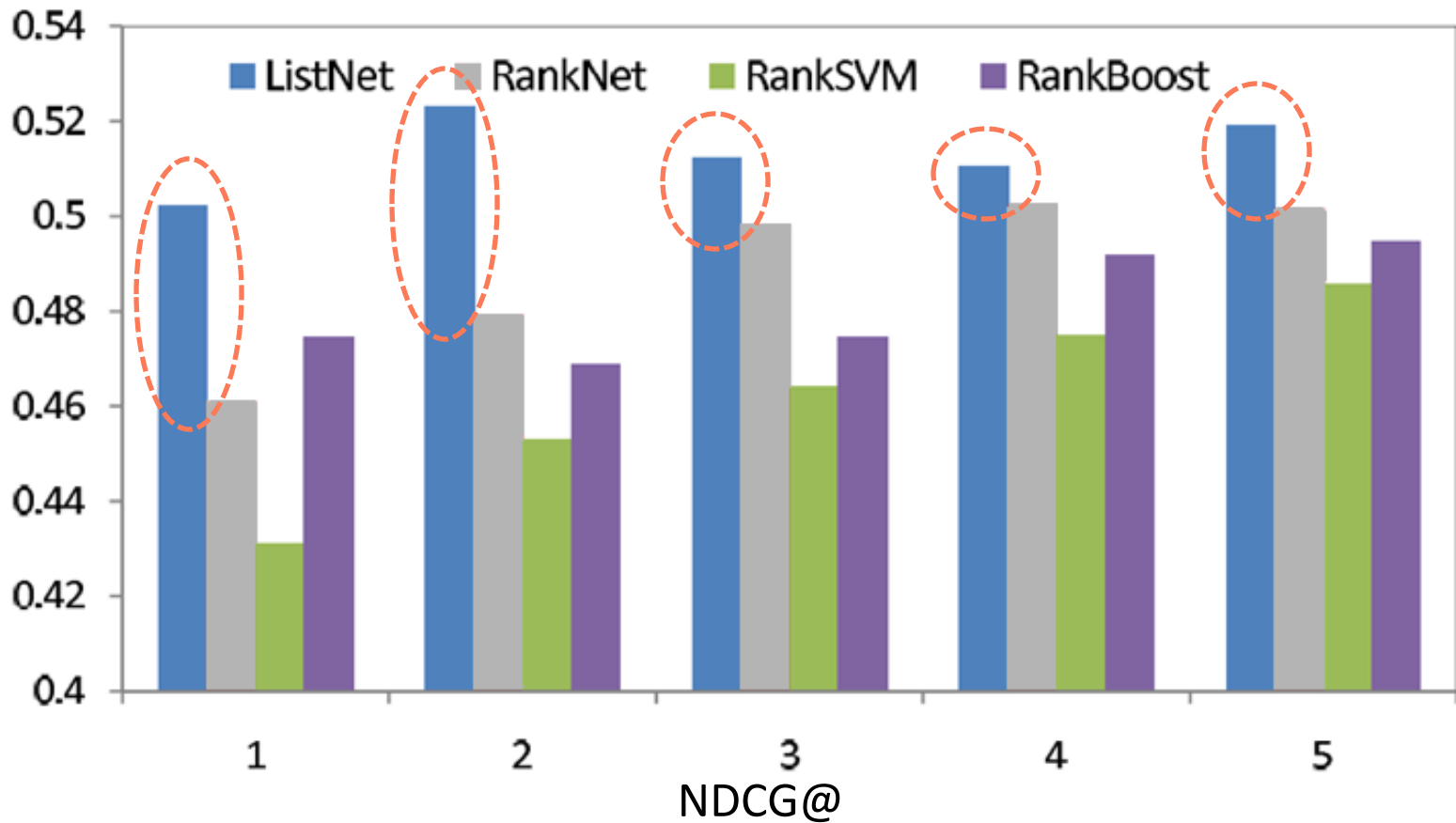
Pairwise (RankNet)



Listwise (ListNet)

Training Performance on TREC Dataset

# Experimental Results



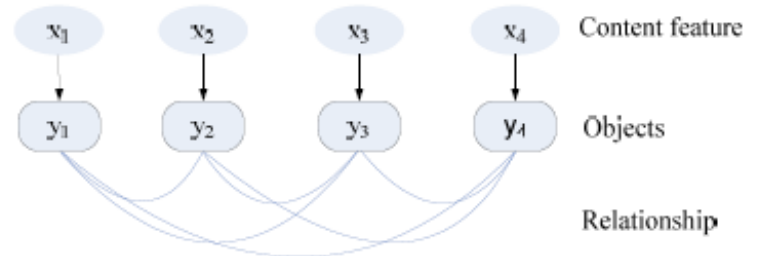
Testing Performance on TREC Dataset

# Learning to Rank Relational Objects using a Listwise Ranking Function

(WWW, 2008)

# Motivation

- Traditional ranking function models **independent relevance**
  - Working on features of single documents
- Many applications in IR are beyond independent relevance
  - Subtopic retrieval
  - Representative page retrieval
  - Pseudo relevance feedback
  - Topic distillation



# Relational Ranking Model

- Generic formulation

$$y = f(X, R) = f(h(X; \omega), R).$$
$$\min_{\omega} \sum_{i=1}^N L(f(h(X_i; \omega), R_i), y_i)$$

- Practical formulation

$$f(h(X; \omega), R) = \arg \min_z \{l_1(h(X; \omega), z) + \beta l_2(R, z)\}$$

$$l_1(h(X; \omega), z) = \|h(X; \omega) - z\|^2$$

Application-dependent

# Pseudo Relevance Feedback

- A very common type of relationship between objects is similarity.
  - Similarity relationship can be represented in an undirected graph.
  - Such kind of graph is widely used in the learning tasks of clustering and semi-supervised learning.

$$l_2(R, z) = 1/2 \sum_i \sum_j R_{i,j} (z_i - z_j)^2$$

$$f(h(X; \omega), R) = \arg \min_z \{ \|X\omega - z\|^2 + \beta/2 \sum_i \sum_j R_{i,j} (z_i - z_j)^2 \}$$

# Topic Distillation

- Topic distillation prefers parent objects to be ranked before child objects.
  - Preference relationship can be represented by directed graph.

$$R_{i,j} = \begin{cases} 1 & \text{if instance } i \text{ is the parent of } j, \\ 0 & \text{other.} \end{cases}$$

$$l_2(R, z) = \sum_i \sum_j R_{i,j} \exp(z_j - z_i)$$

$$f(h(X; \omega), R) = \arg \min_z \{ \|X\omega - z\|^2 + \beta \sum_i \sum_j R_{i,j} \exp(z_j - z_i) \}$$

# Learning of Relational Ranking Model

- Substitute the ranking model into conventional optimization problems for model learning

$$\min_{\omega} \sum_{i=1}^N L(f(h(X_i; \omega), R_i), y_i)$$

$$s.t. \quad f(h(X; \omega), R) = \arg \min_z \{l_1(h(X; \omega), z) + \beta l_2(R, z)\}$$

- Conventional Ranking SVM is a special case of relational ranking

$$\min_{\omega, \xi_{i,j}} \frac{1}{2} \omega^T \omega + c \mathbf{1}^T [\mathbf{1} - C f(X; \omega)]_+$$

$$s.t. \quad f(X; \omega) = \arg \min_z \|X\omega - z\|^2$$

# Challenges

- The optimization problem for relational ranking is a constrained optimization problem, and the constraint itself is also an optimization problem.
- No existing method can be directly applied to the problem.
- We propose solving the problem in two steps.
  - Solve the optimization problem in constraint and obtain an explicit form for the constraint
  - Solve the original optimization problem with explicit constraint

# Explicit Model for Pseudo Relevance Feedback

$$\begin{aligned}l(z) &= \|X\omega - z\|^2 + \frac{\beta}{2} \sum_i \sum_j R_{i,j} (z_i - z_j)^2 \\ &= \|X\omega - z\|^2 + \beta z^T (D - R)z.\end{aligned}$$

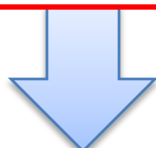


$$\frac{\partial l(z)}{\partial z} = 2(z - X\omega) + 2\beta(D - R)z = 0.$$

$$f(X, G; \omega) = (I + \beta(D - R))^{-1} X\omega$$

# Explicit Model for Topic Distillation

$$l(z) = \|X\omega - z\|^2 + \beta \sum_i \sum_j R_{i,j} \exp(z_j - z_i)$$



$$\exp(z_j - z_i) \approx 1 + (z_j - z_i) + \frac{1}{2}(z_j - z_i)^2.$$

$$\begin{aligned} l(z) &\approx \|X\omega - z\|^2 + \beta \sum_i \sum_j R_{i,j} \left\{ 1 + (z_j - z_i) + \frac{1}{2}(z_j - z_i)^2 \right\} \\ &= \|X\omega - z\|^2 + \beta(g_0 + g_1^T z + z^T (D - R)z) \end{aligned}$$



$$\frac{\partial l(z)}{\partial z} = 2(z - X\omega) + \beta g_1 + \beta(2D - R - R^T)z = 0$$

$$f(X, R; \omega) = (2I + \beta(2D - R - R^T))^{-1}(2X\omega - \beta g_1)$$

# Relational Ranking SVM

## Ranking SVM

$$\begin{aligned} & \min_{\omega, \xi_q} \frac{1}{2} \omega^T \omega + c \sum_q \mathbf{1}_q^T \xi_q \\ \text{s.t. } & C_q f(X_q; \omega) \geq \mathbf{1}_q - \xi_q, f(X_q; \omega) = X_q \omega, \xi \geq 0 \end{aligned}$$

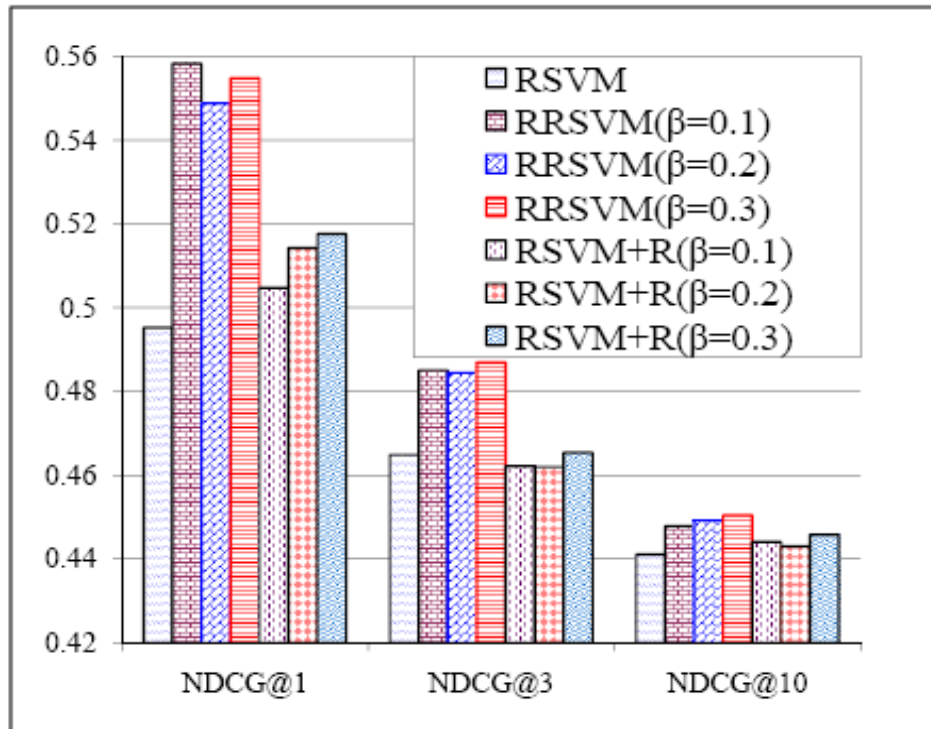
## Relational Ranking SVM for Pseudo Relevance Feedback

$$\begin{aligned} & \min_{\omega, \xi_q} \frac{1}{2} \omega^T \omega + c \sum_q \mathbf{1}_q^T \xi_q \\ \text{s.t. } & C_q f(X_q, R_q; \omega) \geq \mathbf{1}_q - \xi_q, \xi_q \geq 0 \\ & f(X_q, R_q; \omega) = (I - \beta(D_q - R_q))^{-1} X_q \omega \end{aligned}$$

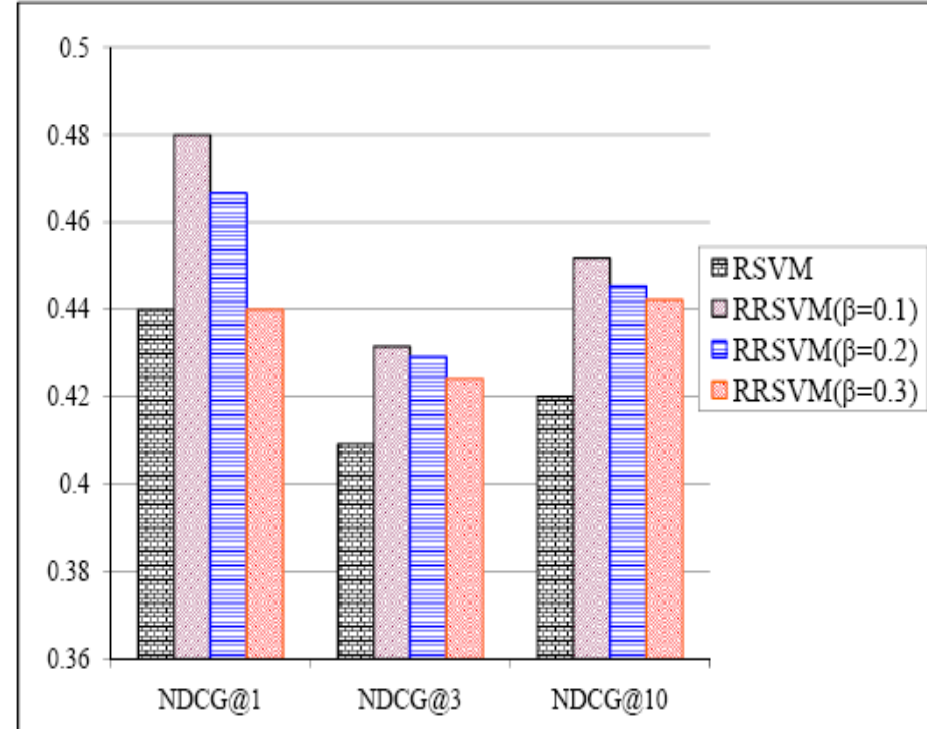
## Relational Ranking SVM for Topic Distillation

$$\begin{aligned} & \min_{\omega, \xi_q} \frac{1}{2} \omega^T \omega + c \sum_q \mathbf{1}_q^T \xi_q \\ \text{s.t. } & C_q f(X_q, R_q; \omega) \geq \mathbf{1}_q - \xi_q, \xi_q \geq 0 \\ & f(X_q, R_q; \omega) = (2I + \beta(2D_q - R_q - R_q^T))^{-1} (2X_q \omega - \beta g_{q,1}) \end{aligned}$$

# Experimental Results



Pseudo Relevance Feedback on OHSUMED



Topic Distillation on TD 2004

# Summary

- Listwise approach is more effective than pairwise approach
- Future Work
  - Combination of listwise loss function and listwise ranking function
  - Investigation of generalization ability of listwise approach

# References

- T. Qin, **T.-Y. Liu**, et al. Learning to Rank Relational Objects, *WWW* 2008.
- X. Geng, **T.-Y. Liu**, et al. Feature Selection for Ranking, *SIGIR* 2007.
- M. Tsai, **T.-Y. Liu**, et al. FRank: A Ranking Method with Fidelity Loss, *SIGIR* 2007.
- T. Qin, **T.-Y. Liu**, et al. Ranking with Multiple Hyperplanes, *SIGIR* 2007.
- J. Xu, H. Li, AdaRank: A Boosting Approach to Information Retrieval, *SIGIR* 2007.
- **T.-Y. Liu**, T. Qin, et al. LETOR: Benchmark dataset for research on learning to rank for information retrieval, *LR4IR 2007*, in conjunction with *SIGIR* 2007.
- T. Joachims, H. Li, **T.-Y. Liu**, C. Zhai, SIGIR Workshop Report: Learning to Rank for Information Retrieval (LR4IR 2007), *SIGIR Forum*, 2007.
- **T.-Y. Liu**, J. Xu, et al. Learning to Rank for Information Retrieval, *TCACS*, 2007.
- Z. Cao, T. Qin, **T.-Y. Liu**, et al. Learning to Rank: From Pairwise Approach to Listwise Approach. *ICML* 2007.
- Y. Liu, **T.-Y. Liu**, et al. Supervised Rank Aggregation, *WWW* 2007.
- T. Qin, **T.-Y. Liu**, et al. Query-level Loss Function for Information Retrieval. *IP&M* 2007.
- Y. Cao, Jun Xu, **T.-Y. Liu**, et al. Adapting Ranking SVM to Document Retrieval, *SIGIR* 2006.

# Thanks!

Benchmark Dataset for Learning to Rank:  
<http://research.microsoft.com/users/LETOR/>