IBIS 2009
Oct. 21, 2009
Fukuoka Japan

Learning to Rank Methods

Hang Li
Microsoft Research Asia

Talk Outline

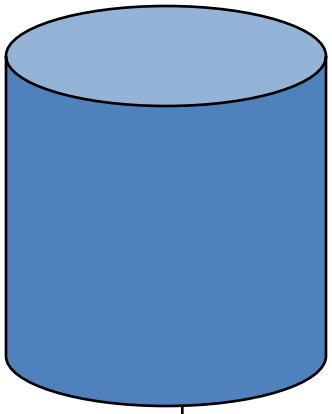
- What is Learning to Rank
- Learning to Rank Methods
  - Ranking SVM
  - IR SVM
  - ListMLE
  - Ada Rank
- Learning to Rank Theory
- Learning to Rank Applications
- Future Directions of Learning to Rank Research

What is Learning to Rank?

Ranking Problem: Example = Document Retrieval

documents

$D = \{d_1, d_2, \cdots, d_N\}$



ranking based on relevance, importance, preference

query

q

$f(q,d)$

ranking of documents

$d_{m+1,1}$ $f(q_{m+1}, d_{m+1,1})$

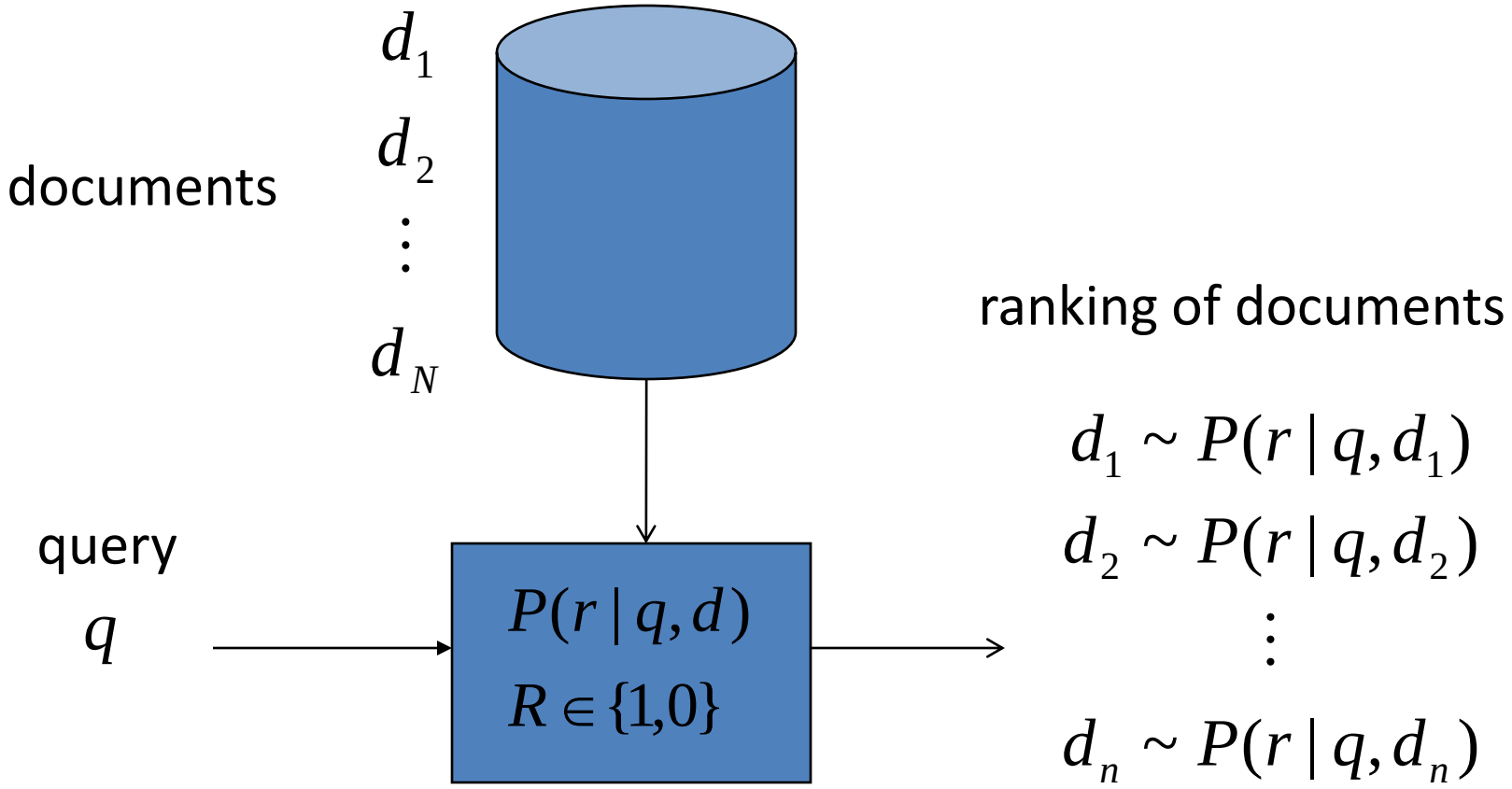
$d_{m+1,2}$ $f(q_{m+1}, d_{m+1,2})$

$\vdots$

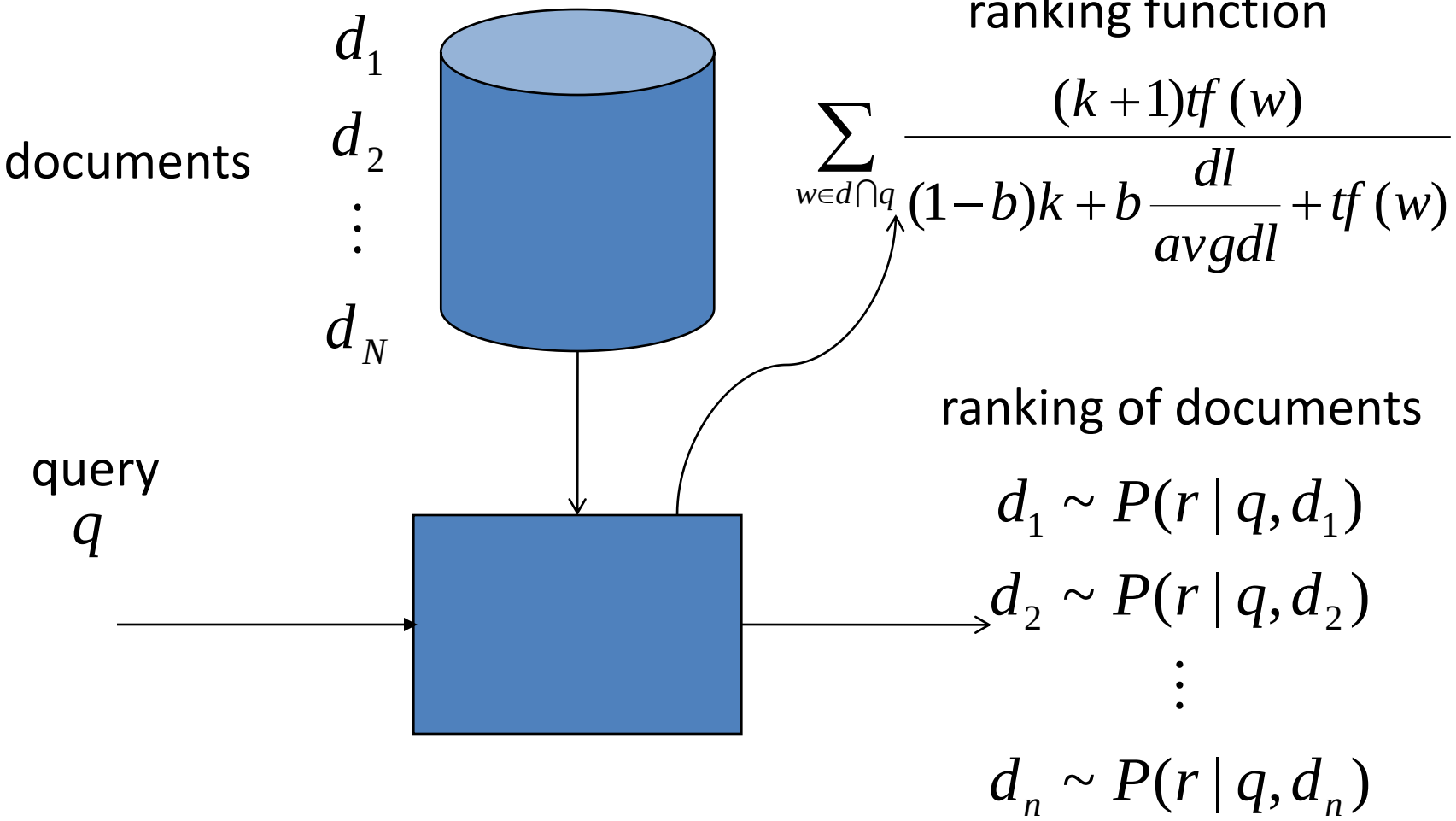
$d_{m+1,n_{m+1}}$ $f(q_{m+1}, d_{m+1,n_{m+1}})$

Traditional Approach to Search Ranking

Probabilistic Model

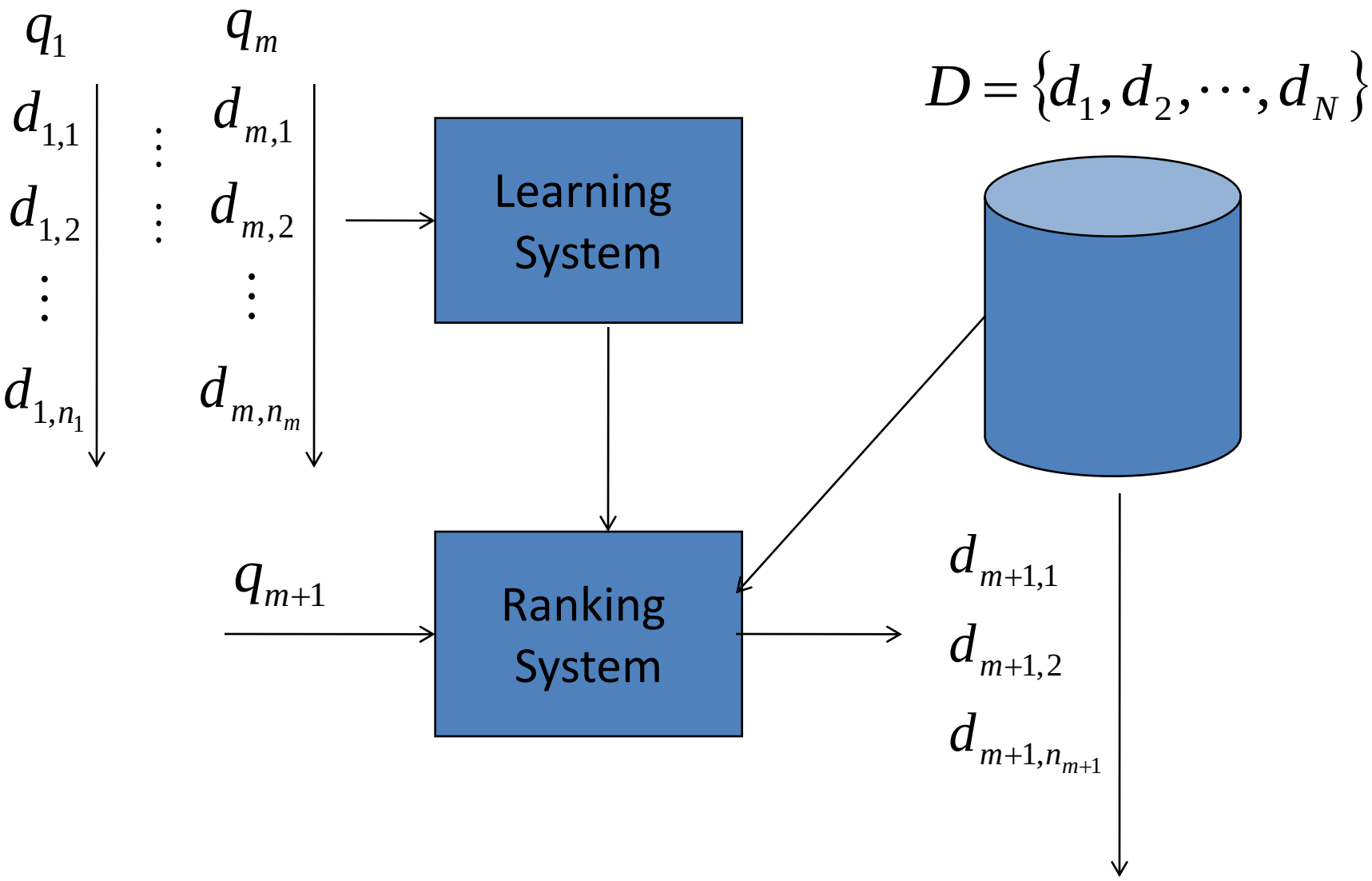


BM25

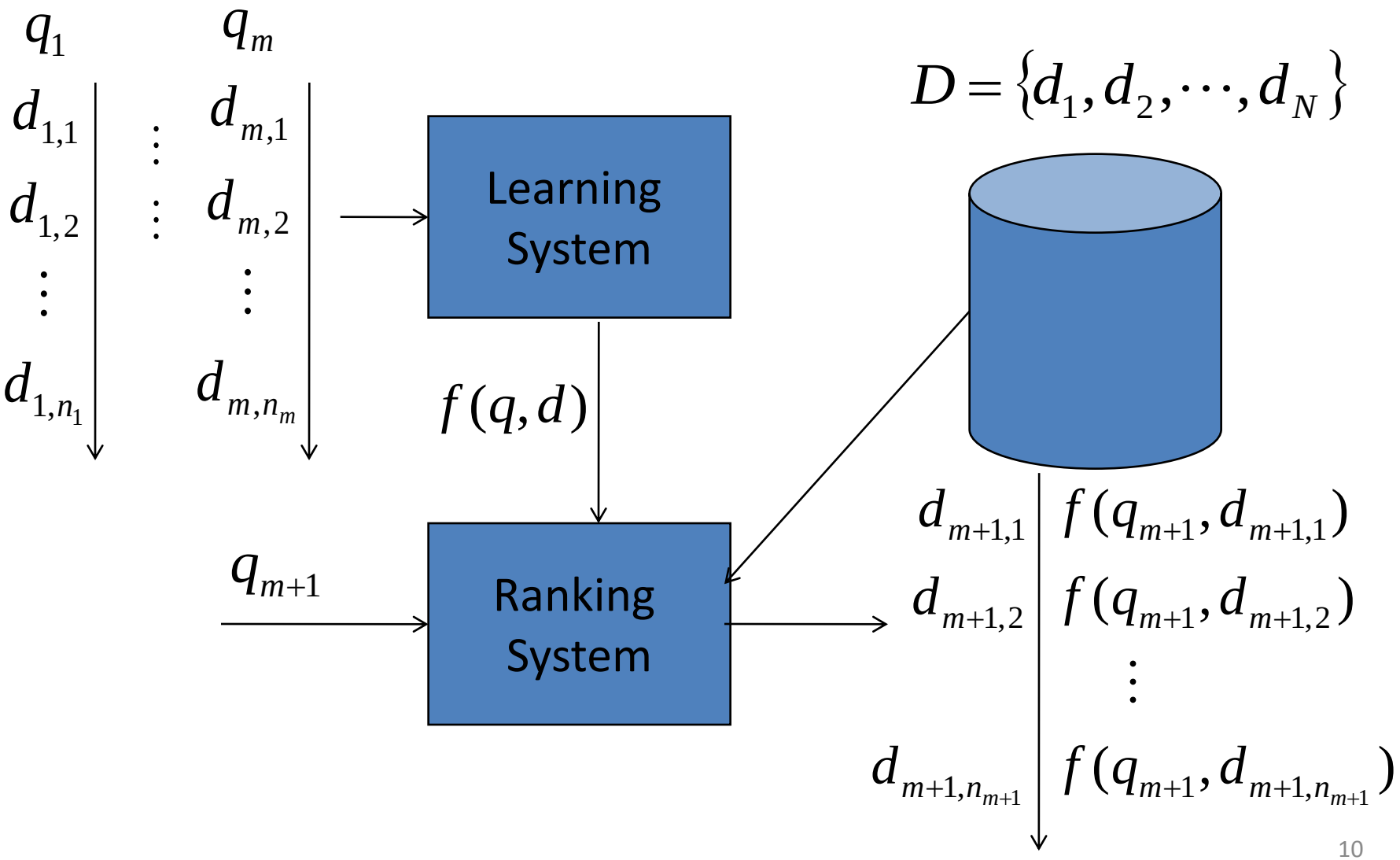

documents
d_1
d_2
$\vdots$
d_N
ranking function
$\sum_{w \in d \cap q} \frac{(k+1)tf(w)}{(1-b)k + b\frac{dl}{avgdl} + tf(w)}$
query
q
ranking of documents
$d_1 \sim P(r \mid q, d_1)$
$d_2 \sim P(r \mid q, d_2)$
$\vdots$
$d_n \sim P(r \mid q, d_n)$

Learning to Rank: New Approach to Search Ranking

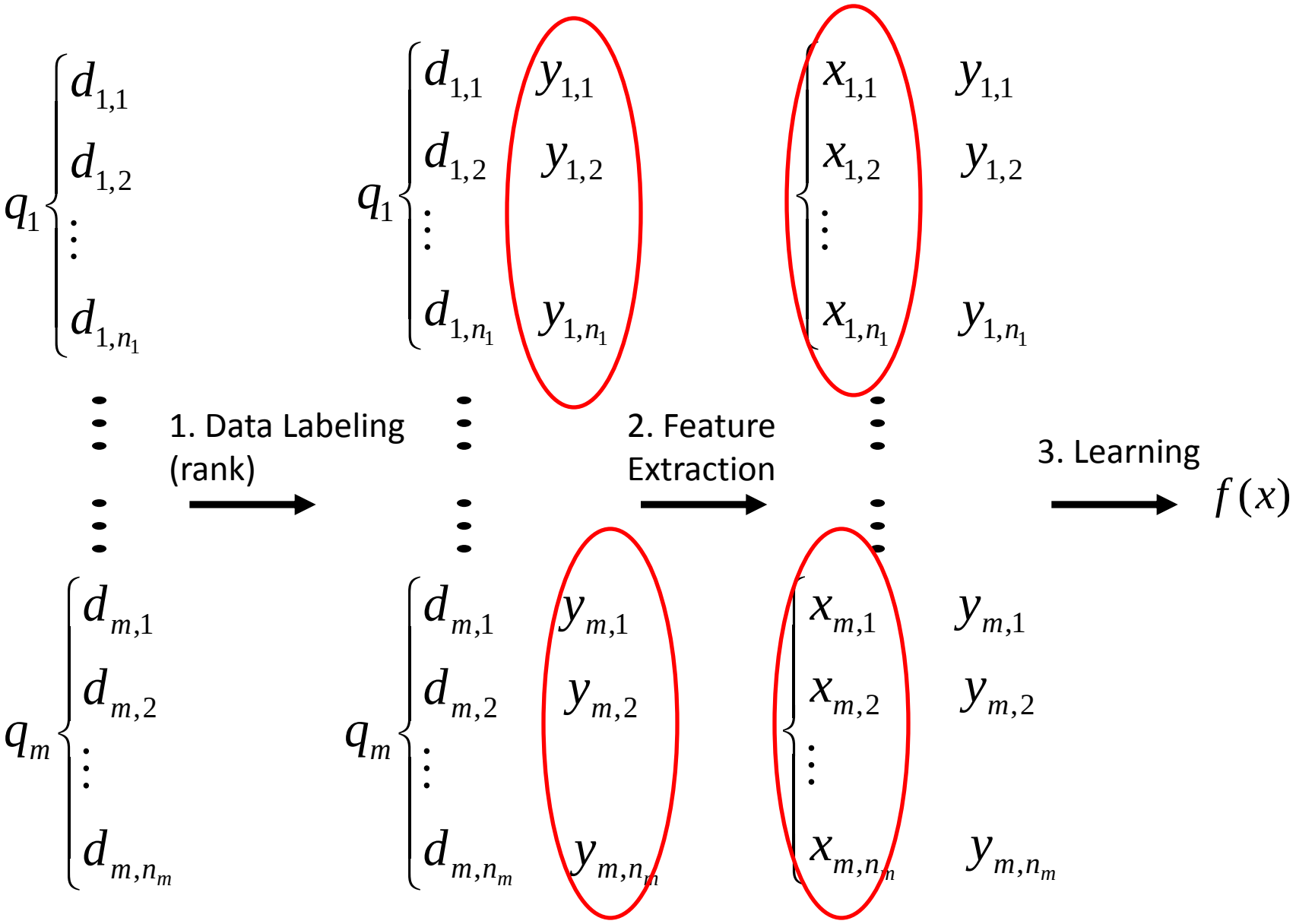
Learning to Rank



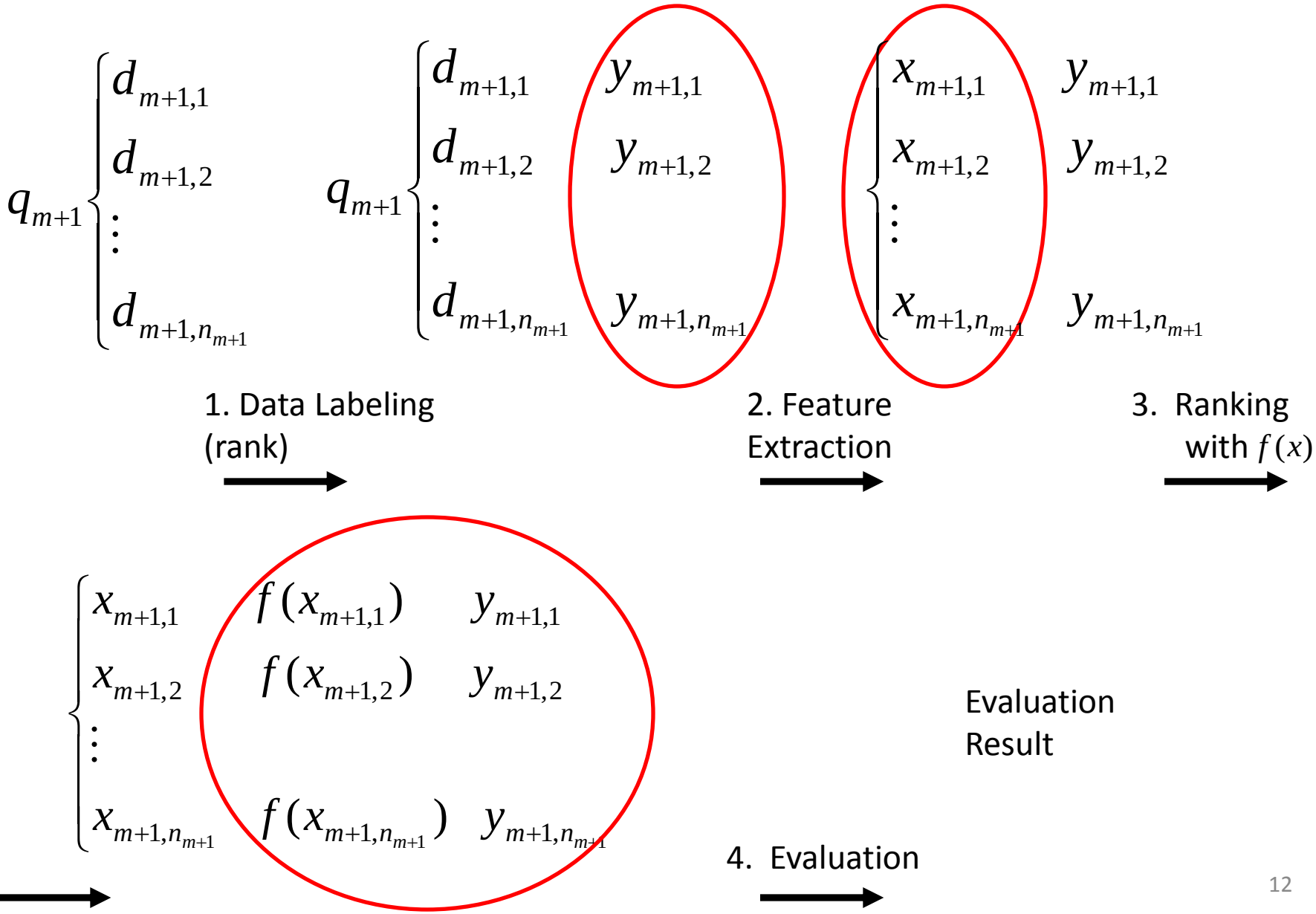
Learning to Rank



Training Process



Testing Process



Notes

- Features are functions of query and document
- Query and associated documents form a group
- Groups are i.i.d. data
- Feature vectors within group are not i.i.d. data
- Ranking model is function of features
- Several data labeling methods (here labeling of rank as example)

Recent Trends on Learning to Rank

- Successfully applied to search
- *Hot topic* in Information Retrieval and Machine Learning
  - Over 100 publications at SIGIR, ICML, NIPS, etc
  - 2 sessions at SIGIR every year
  - 3 SIGIR workshops
  - Special issue at Information Retrieval Journal
  - LETOR benchmark dataset, over 1,000 downloads

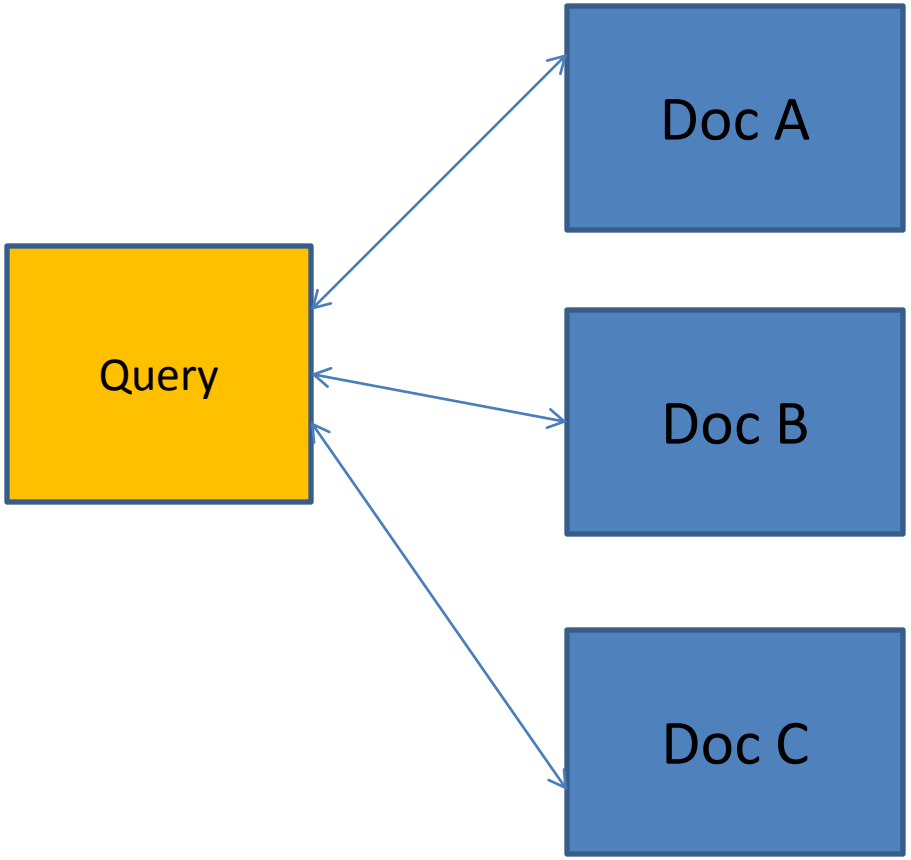
    http://research.microsoft.com/en-us/um/beijing/projects/letor/index.html

Issues in Learning to Rank

- Data Labeling
- Feature Extraction
- Evaluation Measure
- Learning Method (Model, Loss Function, Algorithm)

Data Labeling Problem

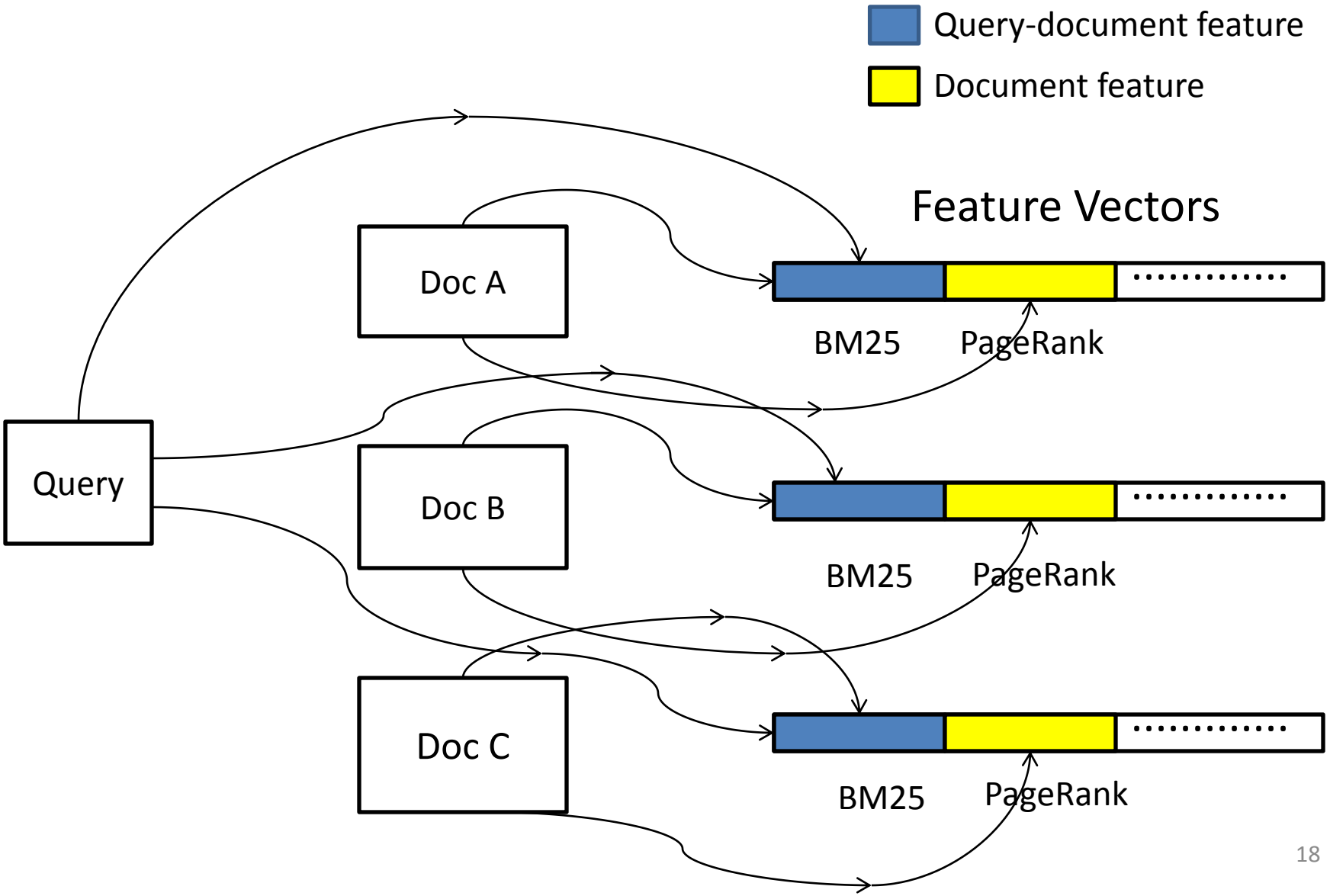
- E.g., relevance of documents w.r.t. query



Data Labeling Methods

- Labeling of Ranks
  - Multiple levels (e.g., relevant, partially relevant, irrelevant)
  - Widely used in IR
- Labeling of Ordered Pairs
  - Ordered pairs between documents (e.g. A>B, B>C)
  - Implicit relevance judgment: derived from click-through data
- Creation of List
  - List (or permutation) of documents is given
  - Ideal but difficult to implement

Feature Extraction



Example Features

- Relevance: BM25
- Relevance: proximity
- Relevance: query exactly occurs in document
- Importance: PageRank

Evaluation Measures

- Important to rank top results correctly
- Measures
  - NDCG (Normalized Discounted Cumulative Gain)
  - MAP (Mean Average Precision)
  - MRR (Mean Reciprocal Rank)
  - WTA (Winners Take All)
  - Kendall's Tau

NDCG

- Evaluating ranking using labeled ranks
- NDCG at position *j*

$$\frac{1}{n_j}\sum_{i=1}^{j}(2^{r(i)}-1)/\log(1+i)$$

NDCG (cont')

- Example: perfect ranking
  - (3, 3, 2, 2, 1, 1, 1) rank *r=3,2,1*
  - (7, 7, 3, 3, 1, 1, 1) gain $2^{r(j)}-1$
  - (1, 0.63, 0.5, 0.43, 0.39, 0.36, 0.33) position discount $1/\log(1+j)$
  - (7, 18.11, 24.11, …) DCG

$$\sum_{i=1}^{j}(2^{r(i)}-1)/\log(1+i)$$

  - (1/7, 1/18.11, 1/24.11, …) normalizing factor n_j
  - (1, 1,1,1,1,1,1) NDCG for perfect ranking

NDCG (cont’)

- Example: imperfect ranking
  - (2, 3, 2, 3, 1, 1, 1)
  - (3, 7, 3, 7, 1, 1, 1) Gain
  - (1, 0.63, 0.5, 0.43, 0.39, 0.36, 0.33) Position discount
  - (3, 14.11, 20.11, …) DCG
  - (0.43, 0.78, 0.83, ….) NDCG

- Imperfect ranking decreases NDCG

Relations with Other Learning Tasks

- No need to predict category
  vs Classification
- No need to predict value of $f(q,d)$
  vs Regression
- Relative ranking order is more important
  vs Ordinal regression
- *Learning to rank can be approximated by classification, regression, ordinal regression*

Ordinal Regression (Ordinal Classification)

- Categories are ordered
  - 5, 4, 3, 2, 1
  - e.g., rating restaurants
- Prediction
  - Map to ordered categories

Learning to Rank Methods

Learning to Rank Methods

- Pointwise Approach
  - Subset Ranking [Cossock and Zhang, 2006]: Regression
  - SVM [Nallapati, 2004]: Binary Classification Using SVM
  - McRank [Li et al 2007]: Multi-Class Classification Using Boosting Tree
  - Prank [Crammer and Singer 2002]: Ordinal Regression Using Perceptron
  - Large Margin [Shashua & Levin 2002]: Ordinal Regression Using SVM

Learning to Rank Methods

- Pairwise Approach
  - Ranking SVM: Pairwise Classification Using SVM
  - RankBoost [Freund et al 2003]: Pairwise Classification Using Boosting
  - RankNet [Burges et al 2005]: Pairwise Classification Using Neural Net
  - Frank [Tsai et al 2007]: Pairwise Classification Using Fidelity Loss and Neural Net
  - GBRank [Zheng et al 2007]: Pairwise Regression Using Boosting Tree
  - IR SVM [Cao et al 2006]: Cost-sensitive Pairwise Classification Using SVM
  - Multiple SVMs [Qin et al 2007]: Multiple SVMs

Learning to Rank Methods

- Listwise Approach
  - ListNet [Cao et al 2007]: Probabilistic Ranking Model
  - ListMLE [Xia et al 2008]: Probabilistic Ranking Model
  - AdaRank [Xu and Li 2007]: Direct Optimization of Evaluation Measure
  - SVM Map [Yue et al 2007]: Direct Optimization of Evaluation Measure
  - PermuRank [Xu et al 2008]: Direct Optimization of Evaluation Measure
  - Soft Rank [Taylor et al 2008]: Approximation of Evaluation Measure
  - Lambda Rank [Burges et al 2007]: Using Implicit Loss Function

Learning to Rank Methods

- Other Methods
  - K-Nearest Neighbor Ranker [Geng et al 2008]
  - Semi-Supervised Learning [Jin et al 2008]

Evaluation Results

- Pairwise approach and listwise approach perform better than pointwise approach
- Listwise approach performs better than pairwise approach in most cases
- Listwise approach
  - ListMLE, ListNet, AdaRank, PermuRank, SVM-MAP
- Pairwise approach
  - Ranking SVM, RankNet, RankBoost
- Pointwise approach
  - Linear Regression

Ranking SVM

Transforming Ranking to Pairwise Classification

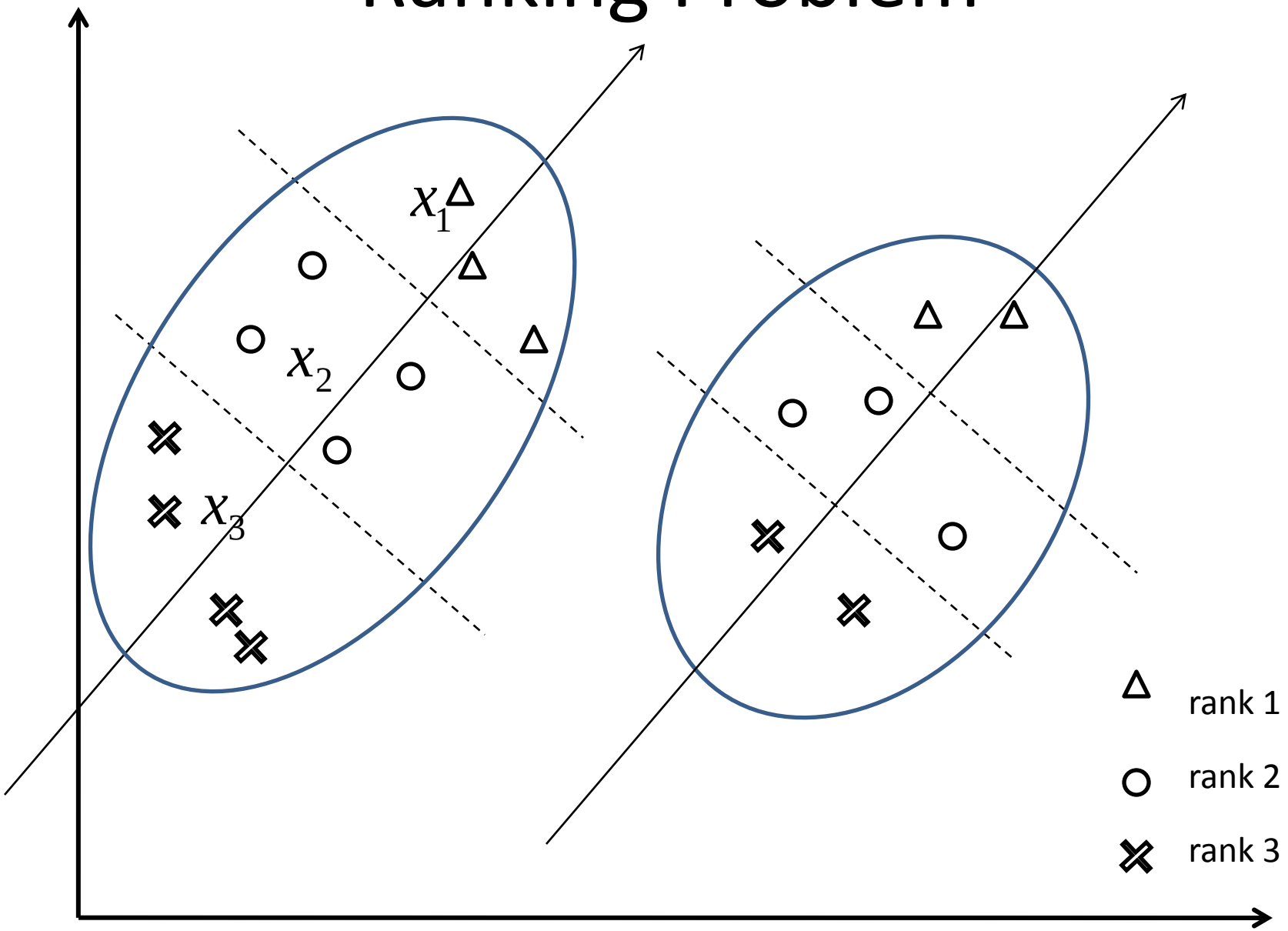
- Input space: *X*
- Ranking function $f : X \to R$
- Ranking: $x_i \succ x_j \;\Leftrightarrow\; f(x_i; w) > f(x_j; w)$
- Linear ranking function: $f(x; w) = \langle w, x \rangle$

$$\langle w, x_i - x_j \rangle > 0 \quad \Leftrightarrow \quad f(x_i; w) > f(x_j; w)$$

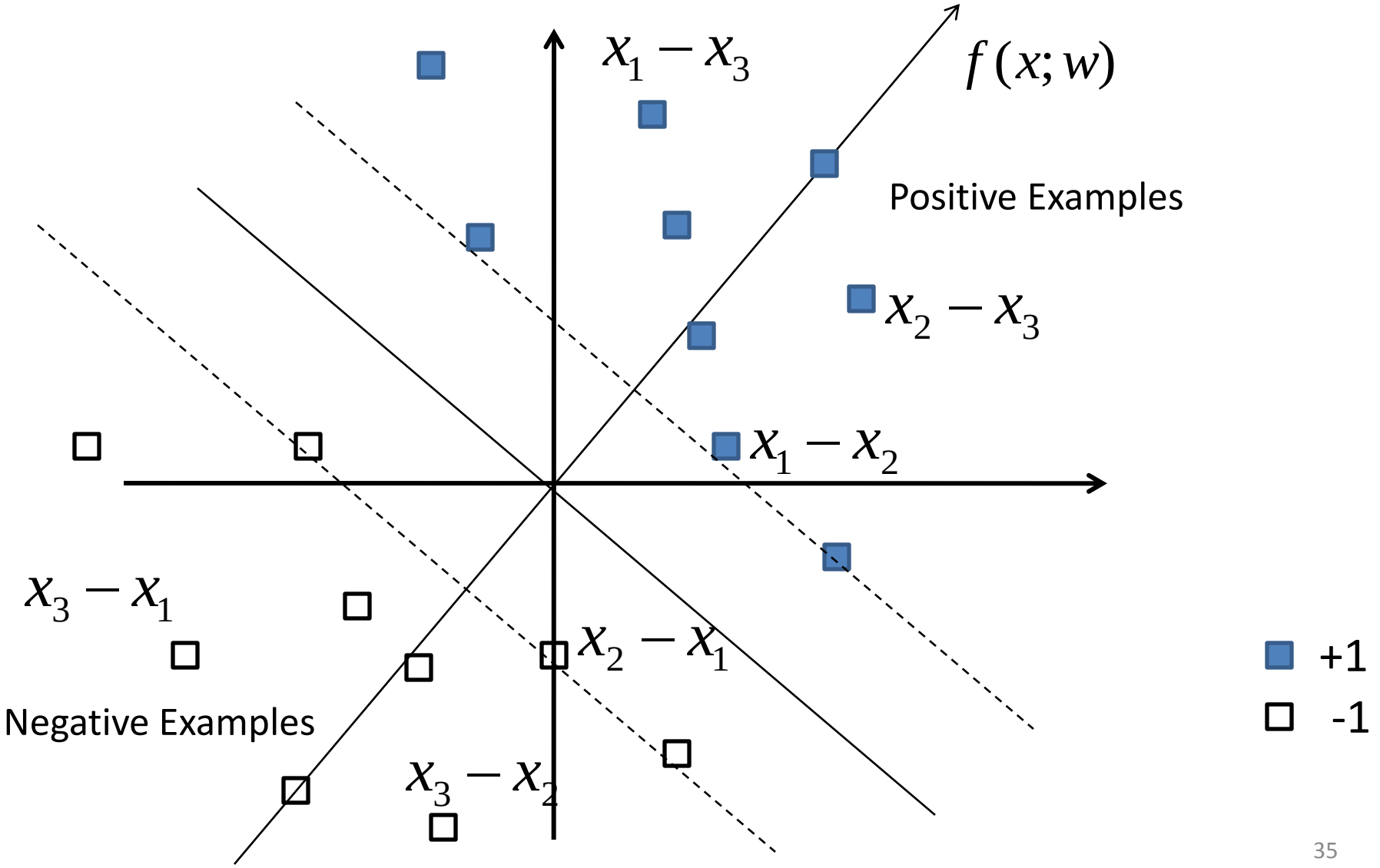
- Transforming to pairwise classification:

$$(x_i - x_j, z),\ z = \begin{cases} +1 & x_i \succ x_j \\ -1 & x_j \succ x_i \end{cases}$$

Ranking Problem



Transformed Pairwise Classification Problem



Ranking SVM

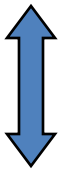
- Pairwise classification on differences of feature vectors
- Corresponding positive and negative examples
- Negative examples are redundant and can be discarded
- Hyper plane passes the origin
- Soft Margin and Kernel can be used
- *Ranking SVM* = pairwise classification SVM

Learning of Ranking SVM

$$\min_{w,\xi} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} \xi_i$$

$$z_i \left\langle w, x_i^{(1)} - x_i^{(2)} \right\rangle \geq 1 - \xi_i \quad i = 1, \cdots, l$$

$$\xi_i \geq 0$$



$$\min_{w} \sum_{i=1}^{l} \left[1 - z_i \left\langle w, x_i^{(1)} - x_i^{(2)} \right\rangle \right]_+ + \lambda \| w \|^2$$

$$[s]_+ = \max(0, s) \quad \lambda = \frac{1}{2C}$$

IR SVM

Problems with Ranking SVM

- Not sufficient emphasis on correct ranking on top

  r: relevant, p: partially relevant, i: irrelevant

  ranking 1: p r p i i i i

  ranking 2: r p i p i i i

  ranking 2 should be better than ranking 1

  Ranking SVM views them as the same

- Numbers of pairs vary according to queries

  q1: r p p i i i i

  q2: r r p p p i i i i i

  q1 pairs: 2*(r, p) + 4*(r, i) + 8*(p, i) = 14

  q2 pairs: 6*(r, p) + 10*(r, i) + 15*(p, i) = 31

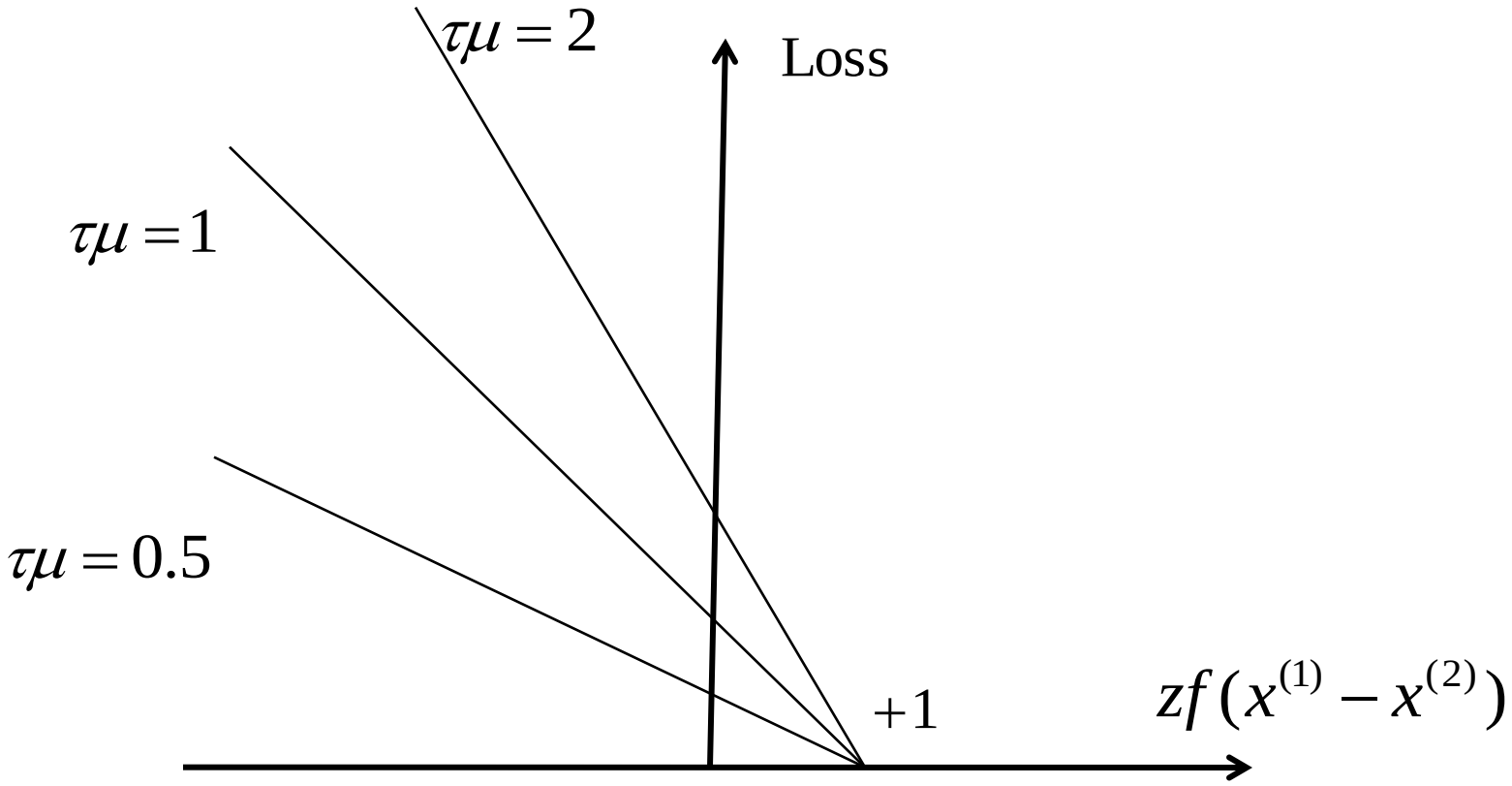
  Ranking SVM is biased toward q2

IR SVM

- Solving the two problems of Ranking SVM
- Higher weight on important rank pairs $\tau_{k(i)}$
- Normalization weight on pairs in query $\mu_{q(i)}$
- IR SVM = Ranking SVM using modified hinge loss

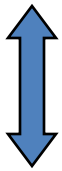
Modified Hinge Loss function

$$\min_{w} \sum_{i=1}^{l} \tau_{k(i)} \mu_{q(i)} \left[1 - z_i \left\langle w, x_i^{(1)} - x_i^{(2)} \right\rangle \right]_+ + \lambda \| w \|^2$$



Learning of IR SVM

$$\min_{w} \sum_{i=1}^{l} \tau_{k(i)} \mu_{q(i)} \left[1 - z_i \left\langle w, x_i^{(1)} - x_i^{(2)} \right\rangle \right]_+ + \lambda \| w \|^2$$



$$\min_{w,\xi} \frac{1}{2} \| w \|^2 + \sum_{i=1}^{l} C_i \xi_i$$

$$z_i \left\langle w, x_i^{(1)} - x_i^{(2)} \right\rangle \geq 1 - \xi_i \quad i = 1, \cdots, l$$

$$\xi_i \geq 0$$

$$C_i = \frac{\tau_{k(i)} \mu_{q(i)}}{2\lambda}$$

ListMLE

Plackett-Luce Model (Permutation Probability)

- Probability of permutation π is defined as

$$P(\pi) = \prod_{i=1}^{n} \frac{s_{\pi(i)}}{\sum_{j=i}^{n} s_{\pi(j)}}$$

- Example:

$$P(\mathrm{ABC}) = \frac{s_A}{s_A + s_B + s_C} \cdot \frac{s_B}{s_B + s_C} \cdot \frac{s_C}{s_C}$$

P(A ranked No.1)

P(B ranked No.2 | A ranked No.1)

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

Properties of Plackett-Luce Model

- Objects: ABC
- Scores: $s_{\mathrm{A}} = 5, s_{\mathrm{B}} = 3, s_{\mathrm{C}} = 1$
- Property 1: *P*(ABC) is largest, *P*(CBA) is smallest
- Property 2: swap B and C in ABC, *P*(ABC) > *P*(ACB)

Plackett-Luce Model (Top-k Probability)

- Computation of permutation probabilities is intractable
- Top-k probability
  - Defining Top-k subgroup $G(o_1 \ldots o_k)$ containing all permutations whose top-k objects are $o_1, \ldots, o_k$
  - $P\big(G\big(o_1 \cdots o_k\big)\big) = \prod_{i=1}^{k} \frac{s_{o_i}}{\sum_{j=i}^{n} s_{o_j}}$
  - Time complexity of computation : from $n!$ to $n!/(n-k)!$
- Example: $P\big(G(\mathrm{A})\big) = \frac{s_A}{s_A + s_B + s_C}$

ListMLE

- Parameterized Plackett-Luce Model

$$s = \exp(f(x;w))$$

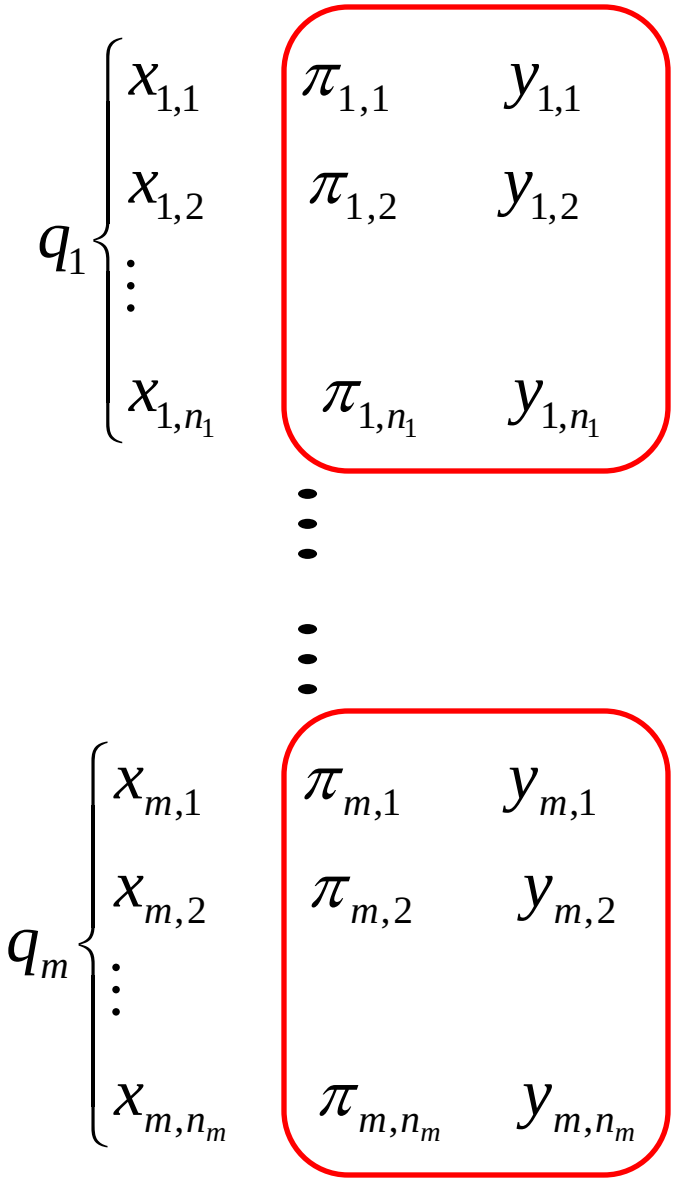
$$P(G(x_1 \cdots x_k)) = \prod_{i=1}^{k} \frac{s_{x_i}}{\sum_{j=i}^{n} s_{x_j}}$$

- Maximum Likelihood Estimation

$$L(w) = -\sum_{q \in Q} \log \left(\prod_{i=1}^{k} \frac{\exp(f(x_i;w))}{\sum_{j=i}^{n} \exp(f(x_j;w))} \right)$$

AdaRank

Listwise Loss



$$\max_{f \in \mathcal{F}} \sum_{i=1}^{m} E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i)$$

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (1 - E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i))$$

AdaRank

- Optimizing exponential loss function
- Algorithm: AdaBoost-like algorithm for ranking

Loss Function of AdaRank

$$\max_{f \in \mathcal{F}} \sum_{i=1}^{m} E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i)$$

Any evaluation measure taking value between [-1,+1]

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (1 - E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i))$$

$$e^{-x} \geq 1 - x$$

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} \exp\{-E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i)\}$$

$$f(\vec{x}) = \sum_{t=1}^{T} \alpha_t h_t(\vec{x})$$

$$\min_{h_t \in \mathcal{H}, \alpha_t \in \Re^+} L(h_t, \alpha_t) = \sum_{i=1}^{m} \exp\{-E(\pi(q_i, \mathbf{d}_i, f_{t-1} + \alpha_t h_t), \mathbf{y}_i)\}$$

AdaRank Algorithm

Input: $S = \{(q_i, \mathbf{d}_i, \mathbf{y}_i)\}_{i=1}^{m}$, and parameters E and T
Initialize $P_1(i) = 1/m$.
For $t = 1, \cdots, T$

- Create weak ranker h_t with weighted distribution $\boldsymbol{P}_t$ on training data S.
- Choose α_t

$$\alpha_t = \frac{1}{2} \cdot \ln \frac{\sum_{i=1}^{m} P_t(i)\{1 + E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i)\}}{\sum_{i=1}^{m} P_t(i)\{1 - E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i)\}}.$$

- Create f_t

$$f_t(\vec{x}) = \sum_{k=1}^{t} \alpha_k h_k(\vec{x}).$$

- Update $\boldsymbol{P}_{t+1}$

$$P_{t+1}(i) = \frac{\exp\{-E(\pi(q_i, \mathbf{d}_i, f_t), \mathbf{y}_i)\}}{\sum_{j=1}^{m} \exp\{-E(\pi(q_j, \mathbf{d}_j, f_t), \mathbf{y}_j)\}}.$$

End For
Output ranking model: $f(\vec{x}) = f_T(\vec{x})$.

Theoretical Results on AdaRank

- Training error will be continuously reduced during learning phase.

THEOREM 1. *The following bound holds on the ranking accuracy of the AdaRank algorithm on training data:*

$$\frac{1}{m}\sum_{i=1}^{m} E(\pi(q_i, \mathbf{d}_i, f_T), \mathbf{y}_i) \geq 1 - \prod_{t=1}^{T} e^{-\delta^t_{min}} \sqrt{1-\varphi(t)^2},$$

where $\varphi(t) = \sum_{i=1}^{m} P_t(i) E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i)$, $\delta^t_{min} = \min_{i=1,\cdots,m} \delta^t_i$, *and*

$$\delta^t_i = E(\pi(q_i, \mathbf{d}_i, f_{t-1} + \alpha_t h_t), \mathbf{y}_i) - E(\pi(q_i, \mathbf{d}_i, f_{t-1}), \mathbf{y}_i) - \alpha_t E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i),$$

for all $i = 1, 2, \cdots, m$ *and* $t = 1, 2, \cdots, T$.

Learning to Rank Theory

Learning to Rank Theory

- Pairwise Approach
  - Generalization Analysis [Lan et al 2008]
- Listwise Approach
  - Generalization Analysis [Lan et al 2009]
  - Consistency Analysis [Xia et al 2008]

Learning to Rank Applications

Learning to Rank Applications

- Search [Burges et al 2005]
- Collaborative Filtering [Freund et al 2003]
- Key Phrase Extraction [Jiang et al 2009]

Collaborative Filtering

	Item1	**Item2**	**Item3**	**...**	
User1	5	4			
User2	1		2		2
...		?	?	?	
UserM	4	3			

Future Directions of Learning to Rank Research

New Issues to be Further Studied

- Learning from implicit data
  - Automatically generate labeled data from implicit feedback
- Model (feature) learning
  - Automatically learn features such as BM25
- Global ranking
  - Using features of current document as well as relations with other documents

New Issues to be Further Studied (cont')

- Query-dependent ranking
  - Creating different ranking models for different queries (in search)
- New applications
  - Machine Translation

Takeaway Message

- Learning to Rank = Machine Learning Task
- Different from classification, regression, ordinal regression
- Learning to Rank has been successfully applied to search
- Existing approaches: pointwise, pairwise, listwise
- Many open problems

Contact: hangli@microsoft.com