The 4th China-Australia
Database Workshop
Melbourne, Australia
Oct. 19, 2015

Learning to Rank Revisited:
Our Progresses in New Algorithms and Tasks

Jun Xu

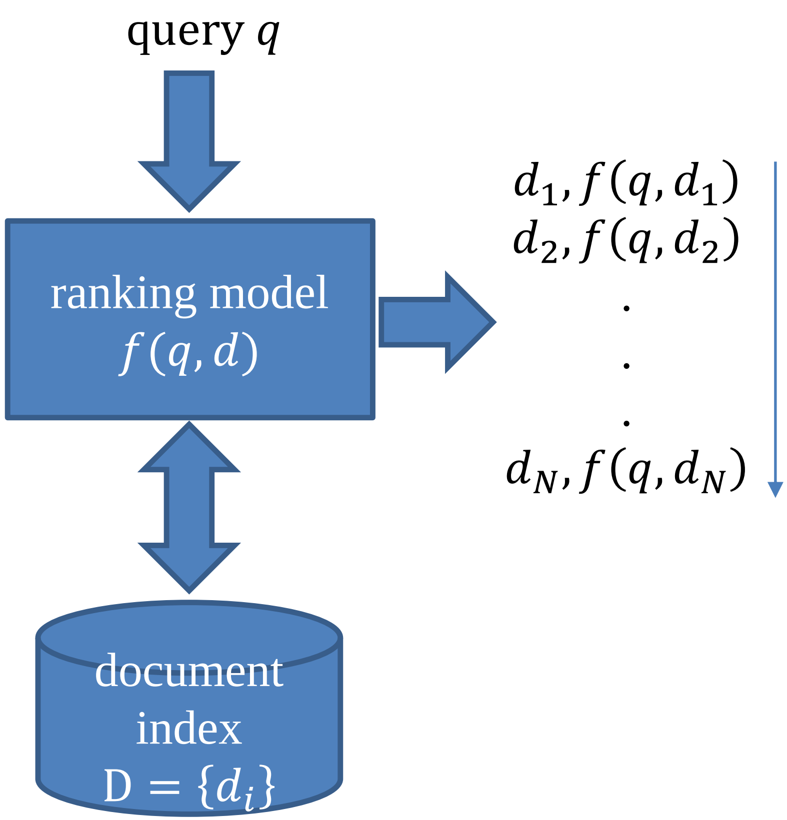
Institute of Computing Technology,

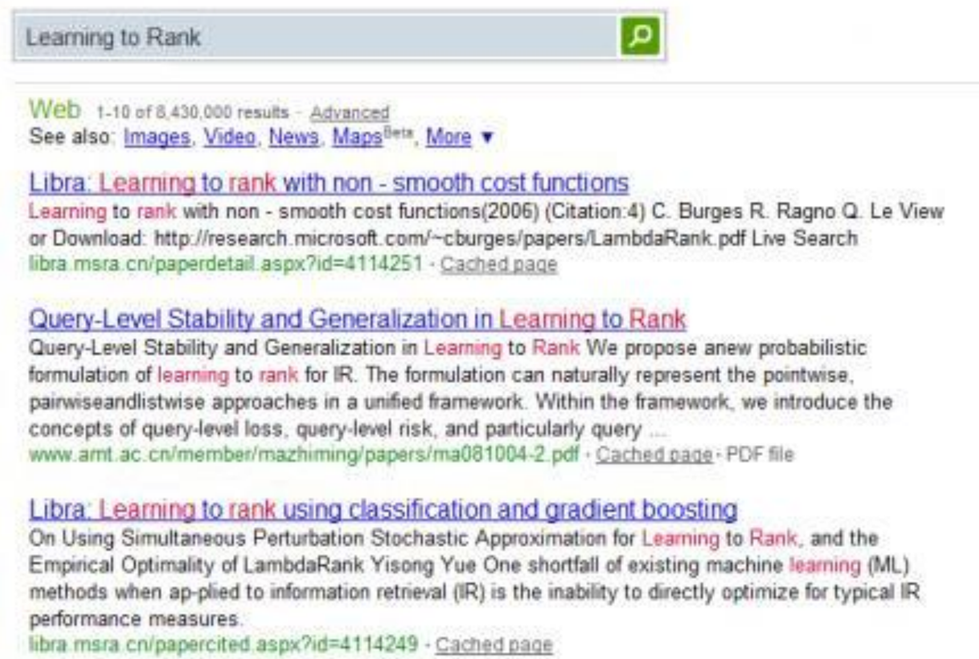
Chinese Academy of Sciences

Outline

- Learning to rank
- Our progresses
  - Improving existing algorithms
  - Adventure with new ranking tasks
- Summary

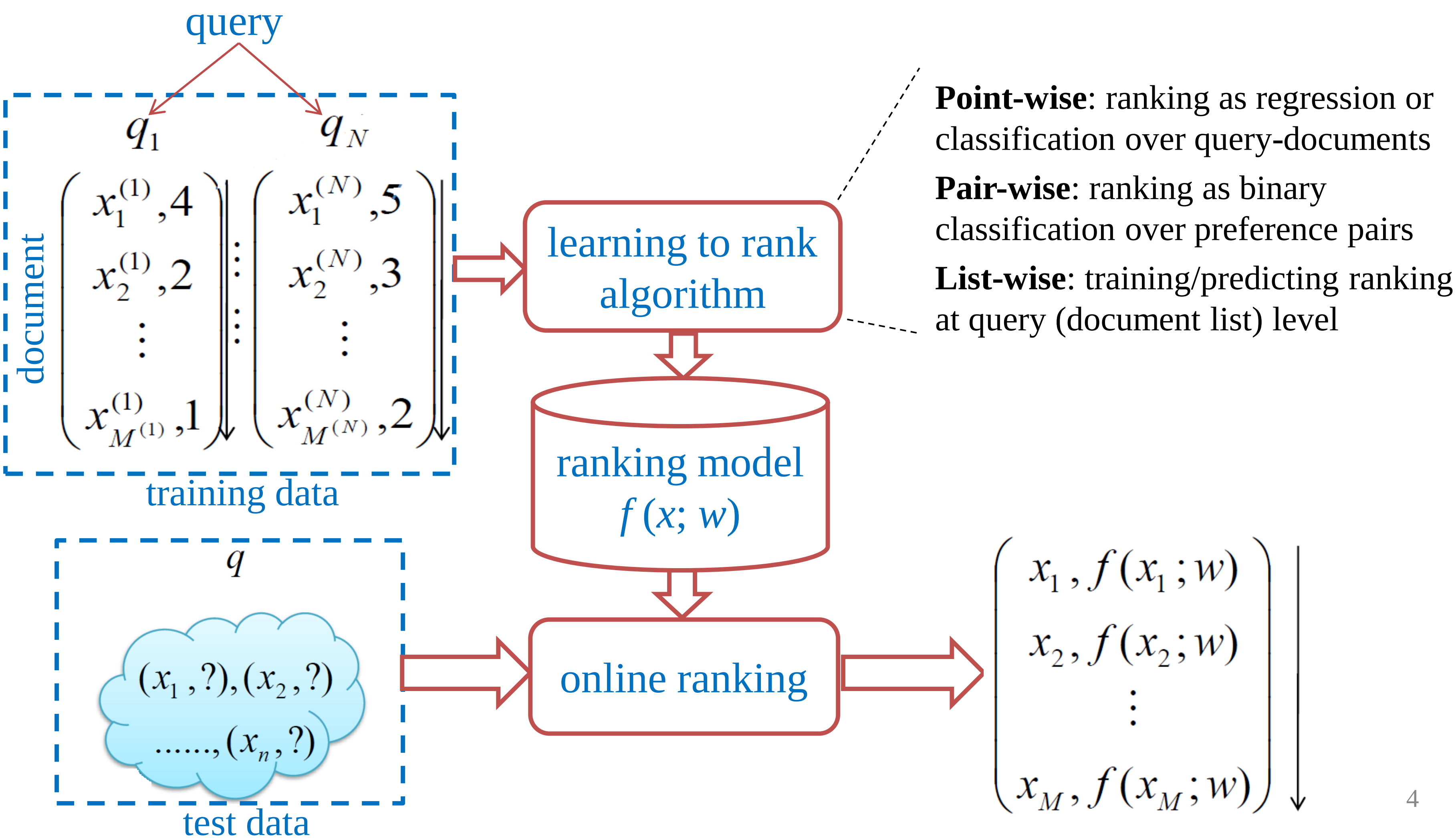
Ranking in Information Retrieval





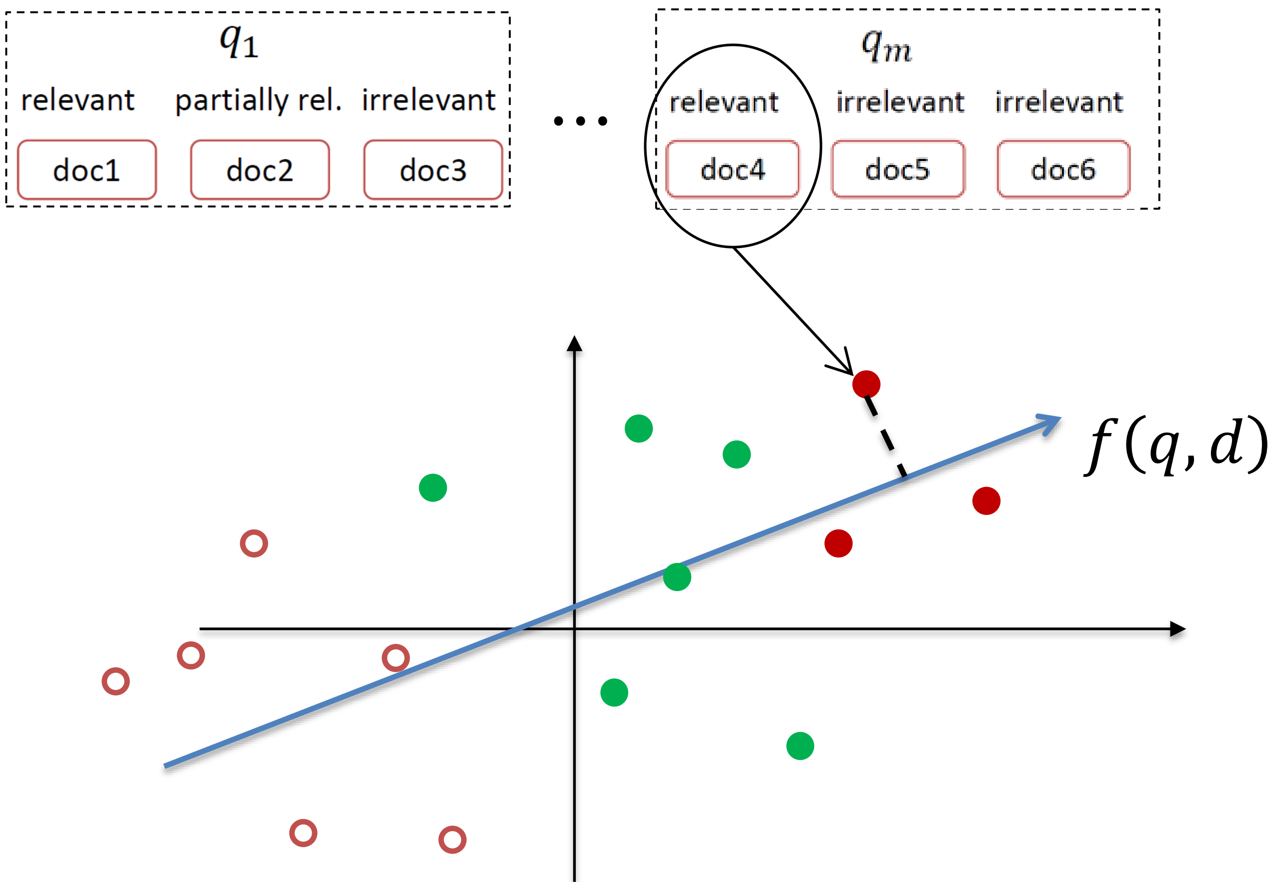
Learning to Rank for Information Retrieval

- Machine learning algorithms for relevance ranking



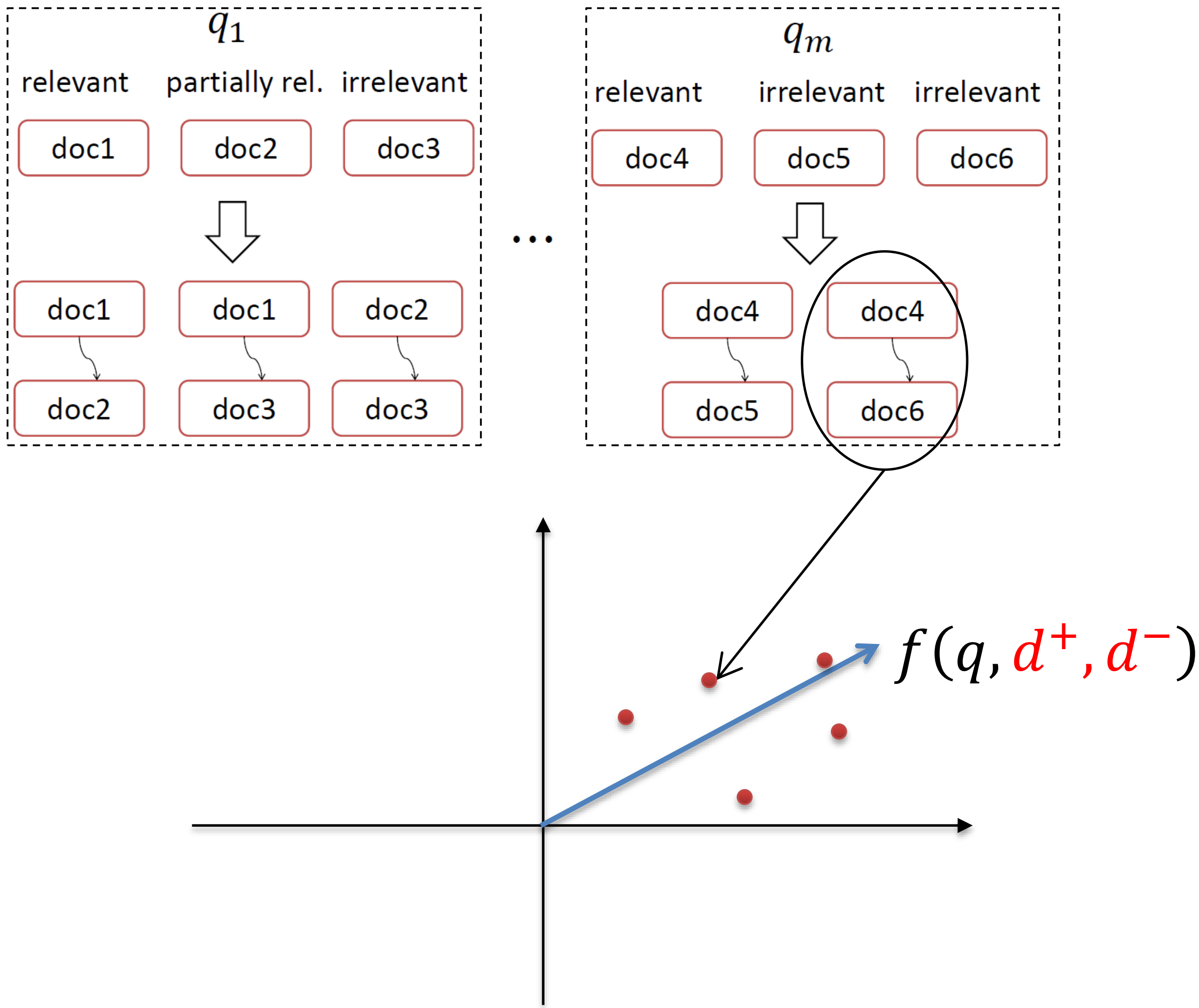
Pointwise Learning to Rank

- Ranking → classification/regression over query-document pairs [R. Nallapati, SIGIR ’04]



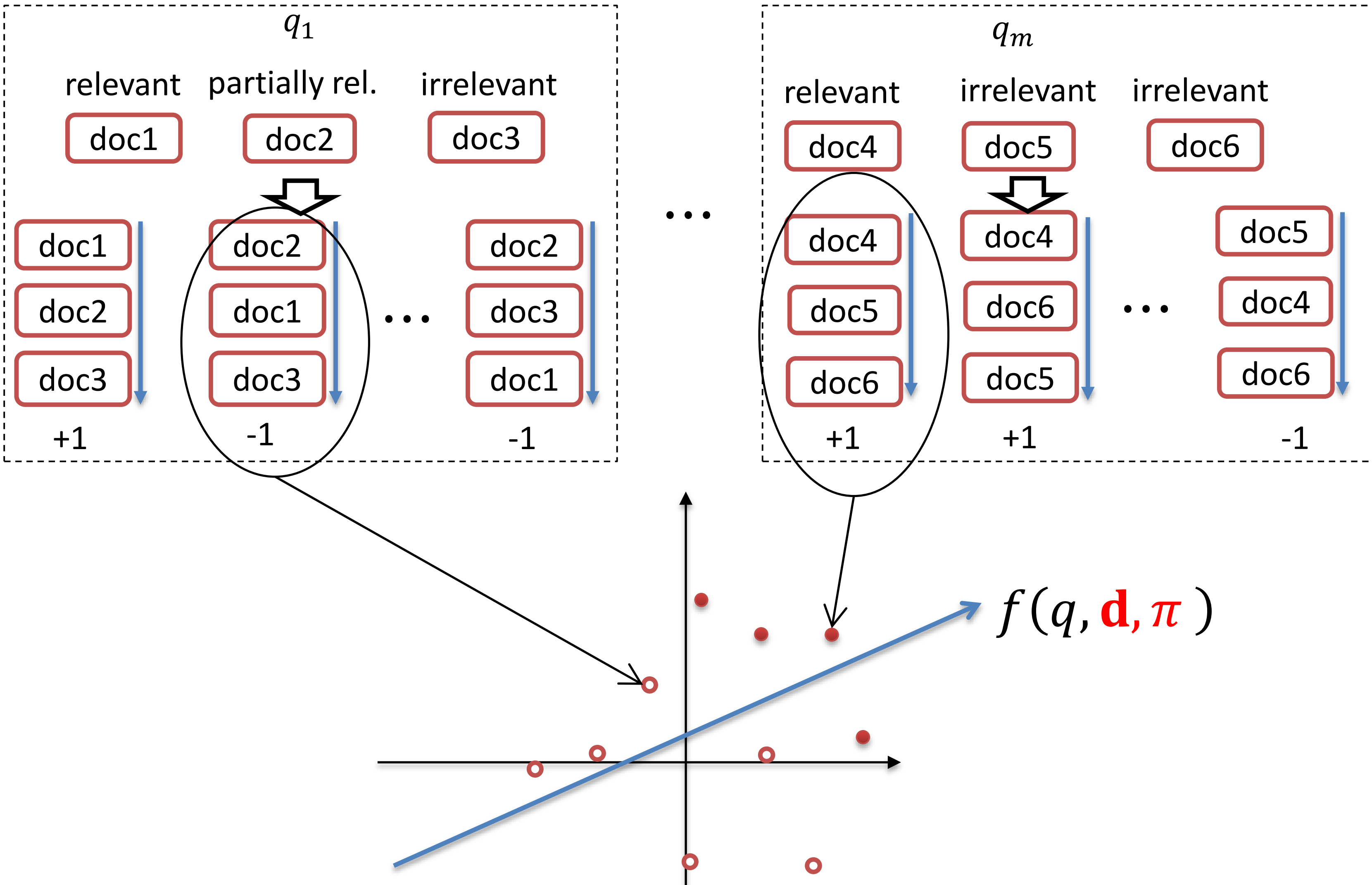
Pairwise Learning to Rank

- Ranking → binary classification over document preference pairs
  [Joachims, KDD '02; Freund et al., JMLR '03; Cao et al., SIGIR ' 06]



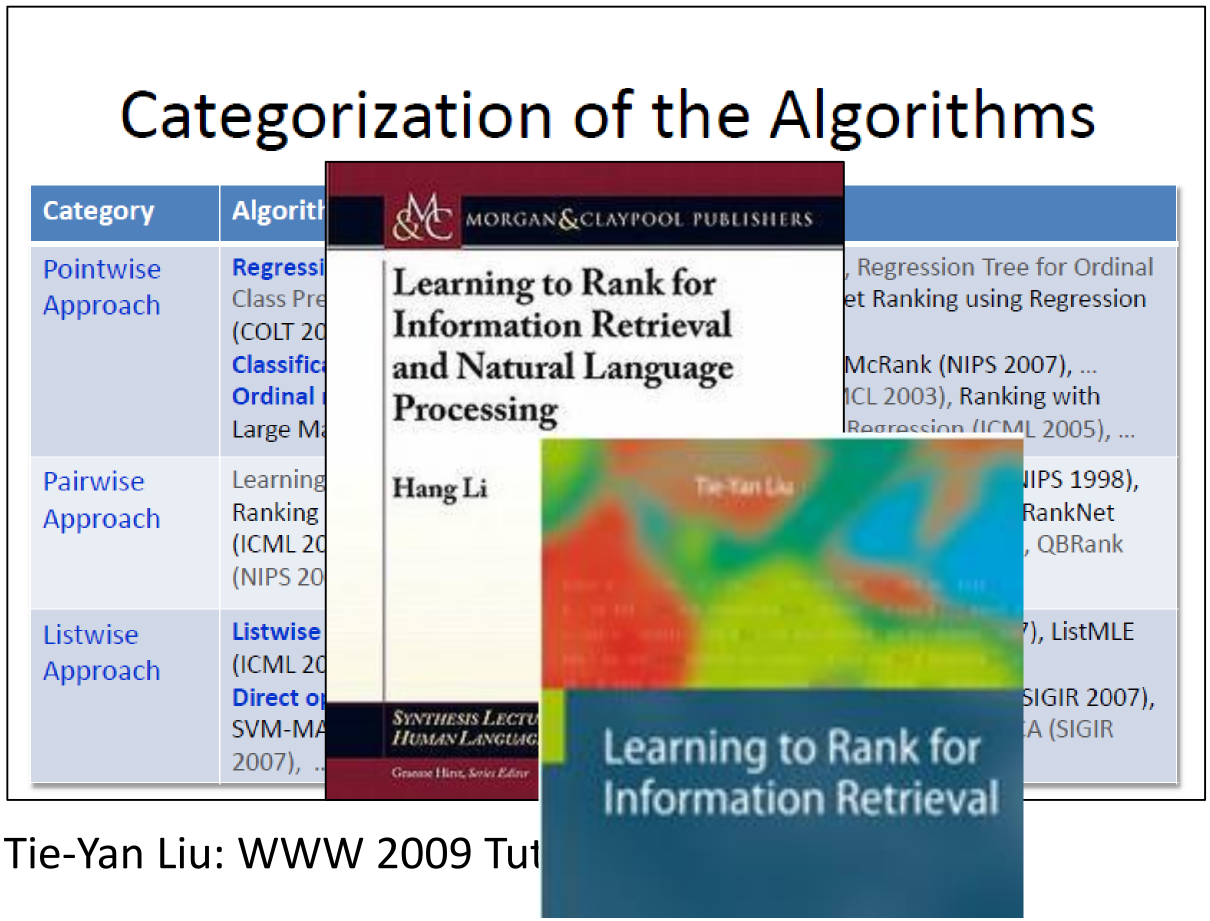
Listwise Learning to Rank

- Ranking → query (document list) level ranking prediction



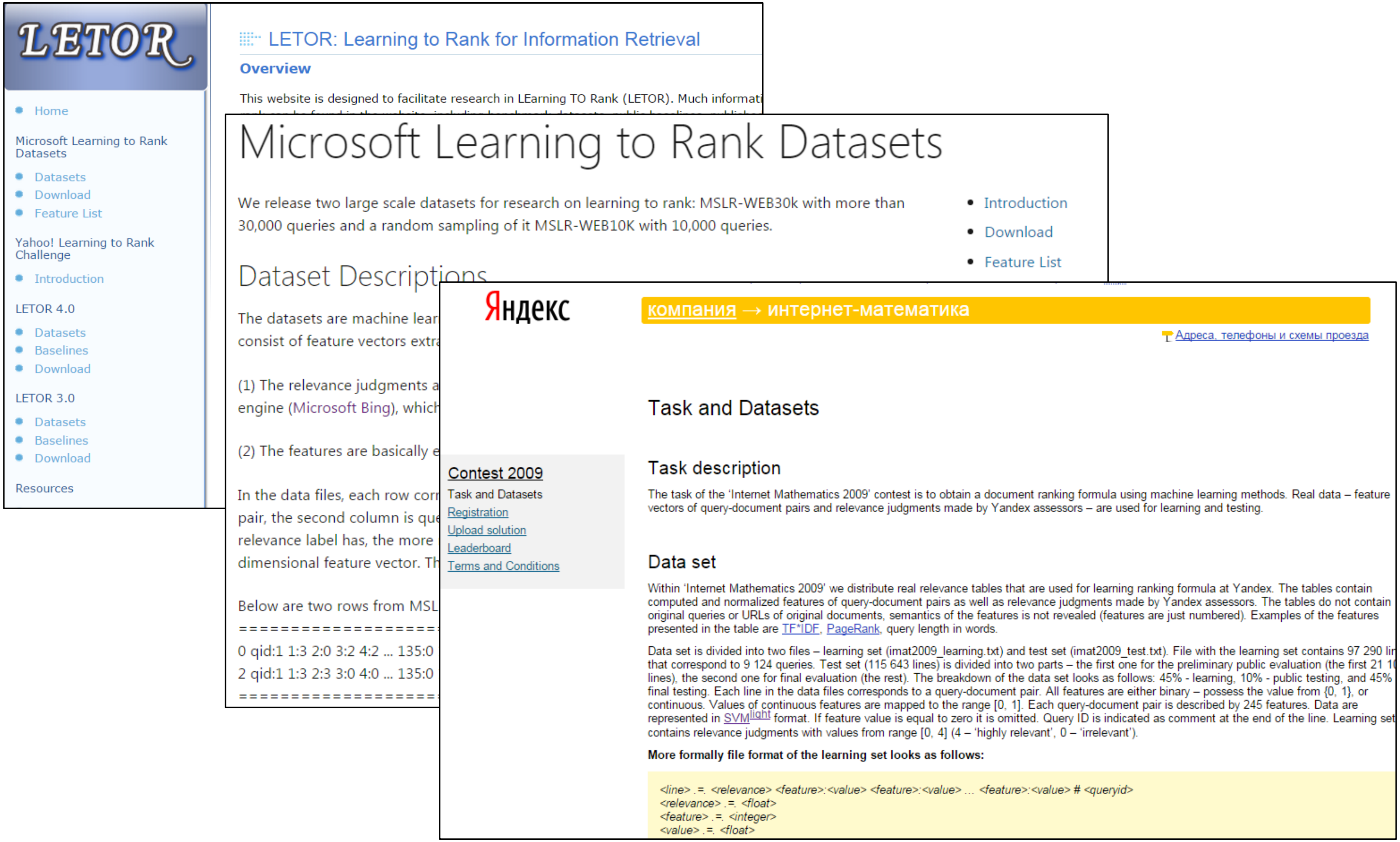
A lot of work

- Publications



Tie-Yan Liu: WWW 2009 Tut

Benchmarks



LETOR
Home
Microsoft Learning to Rank Datasets
Datasets
Download
Feature List
Yahoo! Learning to Rank Challenge
Introduction
LETOR 4.0
Datasets
Baselines
Download
LETOR 3.0
Datasets
Baselines
Download
Resources
LETOR: Learning to Rank for Information Retrieval
Overview
This website is designed to facilitate research in LEarning TO Rank (LETOR). Much informati
Microsoft Learning to Rank Datasets
We release two large scale datasets for research on learning to rank: MSLR-WEB30k with more than 30,000 queries and a random sampling of it MSLR-WEB10K with 10,000 queries.
Introduction
Download
Feature List
Dataset Descriptions
The datasets are machine lear
consist of feature vectors extr
(1) The relevance judgments a
engine (Microsoft Bing), which
(2) The features are basically e
In the data files, each row cor
pair, the second column is que
relevance label has, the more
dimensional feature vector. Th
Below are two rows from MSL
0 qid:1 1:3 2:0 3:2 4:2 ... 135:0
2 qid:1 1:3 2:3 3:0 4:0 ... 135:0
Яндекс
компания → интернет-математика
Адреса, телефоны и схемы проезда
Task and Datasets
Contest 2009
Task and Datasets
Registration
Upload solution
Leaderboard
Terms and Conditions
Task description
The task of the 'Internet Mathematics 2009' contest is to obtain a document ranking formula using machine learning methods. Real data – feature vectors of query-document pairs and relevance judgments made by Yandex assessors – are used for learning and testing.
Data set
Within 'Internet Mathematics 2009' we distribute real relevance tables that are used for learning ranking formula at Yandex. The tables contain computed and normalized features of query-document pairs as well as relevance judgments made by Yandex assessors. The tables do not contain original queries or URLs of original documents, semantics of the features is not revealed (features are just numbered). Examples of the features presented in the table are TF*IDF, PageRank, query length in words.
Data set is divided into two files – learning set (imat2009_learning.txt) and test set (imat2009_test.txt). File with the learning set contains 97 290 li
that correspond to 9 124 queries. Test set (115 643 lines) is divided into two parts – the first one for the preliminary public evaluation (the first 21 1
lines), the second one for final evaluation (the rest). The breakdown of the data set looks as follows: 45% - learning, 10% - public testing, and 45%
final testing. Each line in the data files corresponds to a query-document pair. All features are either binary – possess the value from {0, 1}, or continuous. Values of continuous features are mapped to the range [0, 1]. Each query-document pair is described by 245 features. Data are represented in SVMlight format. If feature value is equal to zero it is omitted. Query ID is indicated as comment at the end of the line. Learning set contains relevance judgments with values from range [0, 4] (4 – 'highly relevant', 0 – 'irrelevant').
More formally file format of the learning set looks as follows:
<line> .=. <relevance> <feature>:<value> <feature>:<value> ... <feature>:<value> # <queryid>
<relevance> .=. <float>
<feature> .=. <integer>
<value> .=. <float>

Workshops and Tutorials

Tools

CORNELL

SVM^{rank}

Support Vector Machine for Ranking

Author: Thorsten Joachims <thorsten@joachims.org>
Cornell University
Department of Computer Science

Overview

AdaRank

The basic idea of AdaRank is con
queries and linearly combining t
minimizes a loss function directly
found in the paper "AdaRank: A

Details

Type	Download
File Name	AdaRank.zip
Version	1.0
Date Published	11 April 2011
Download Size	0.89 MB

RankLib

News (08/12/2013): RankLib is now a part of The Lemur Project, which develops search engines, browser toolbars, text ar
support research and development of information retrieval and text mining software, including Indri and Galago search engi

I am still managing RankLib and will continue to do so. And I now have a proper channel for bug reports, feature requests a
license is still BSD, as most (if not all) of the softwares in The Lemur Project.

Please visit the new home of RankLib for more details.

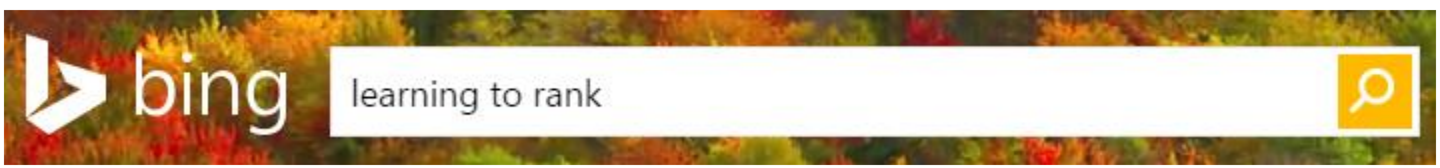
Overview

RankLib is a library of learning to rank algorithms. Currently eight popular algorithms have been implemented:

- **MART** (Multiple Additive Regression Trees, a.k.a. Gradient boosted regression tree) [6]
- **RankNet** [1]
- **RankBoost** [2]
- **AdaRank** [3]

Adopted by Commercial Search Engines

- A number of commercial search engines used learning to rank as their core ranking models
  - LambdaMART



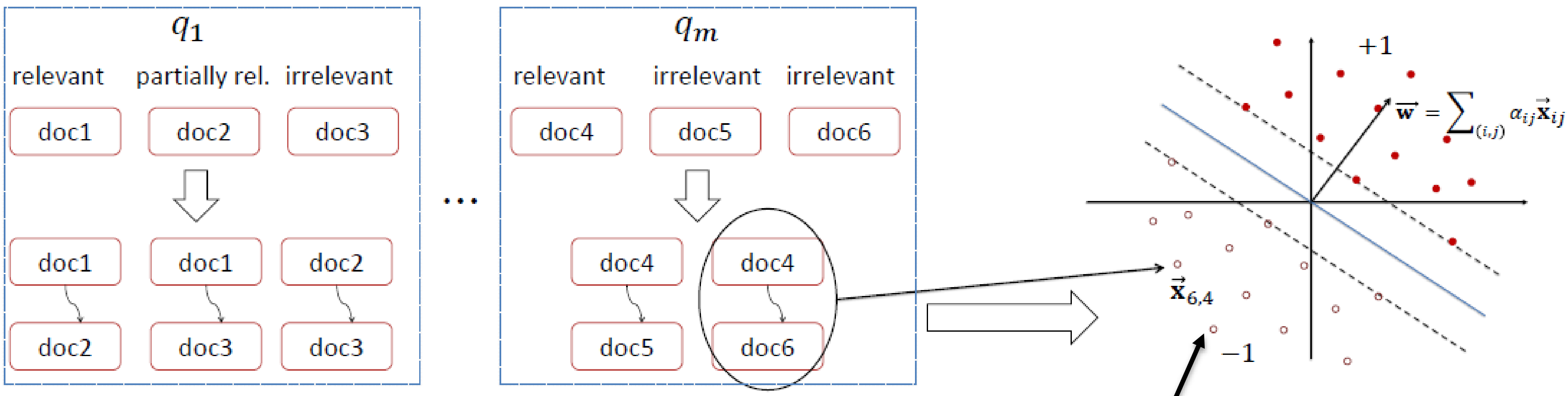
Enough?

- Not yet!
- Existing algorithms are not perfect (from both practical and theoretical views)
  - Violate machine learning assumptions (for making the formulation feasible)
- Few algorithms for ranking tasks other than relevance ranking
  - Search result diversification
  - Incorporating human knowledge
  - ...

Outline

- Learning to rank
- Our progresses
  - Improving existing algorithms
  - Adventure with new ranking tasks
- Summary

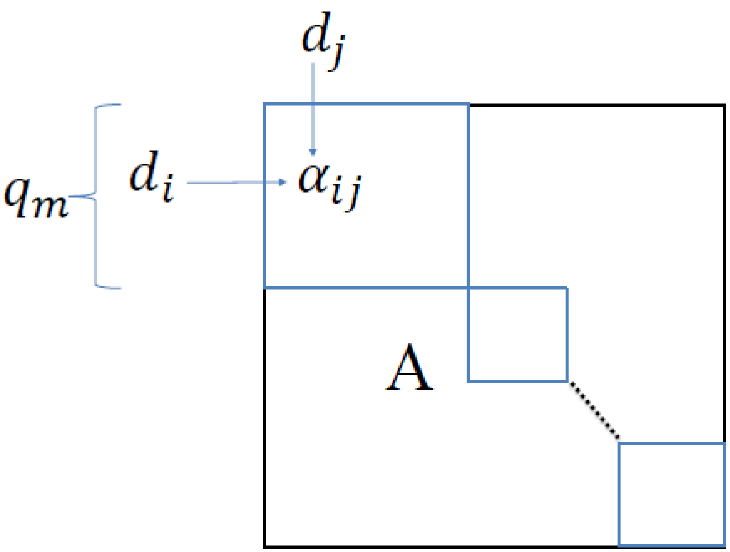
Ranking SVM [Joachims, KDD ‘02]



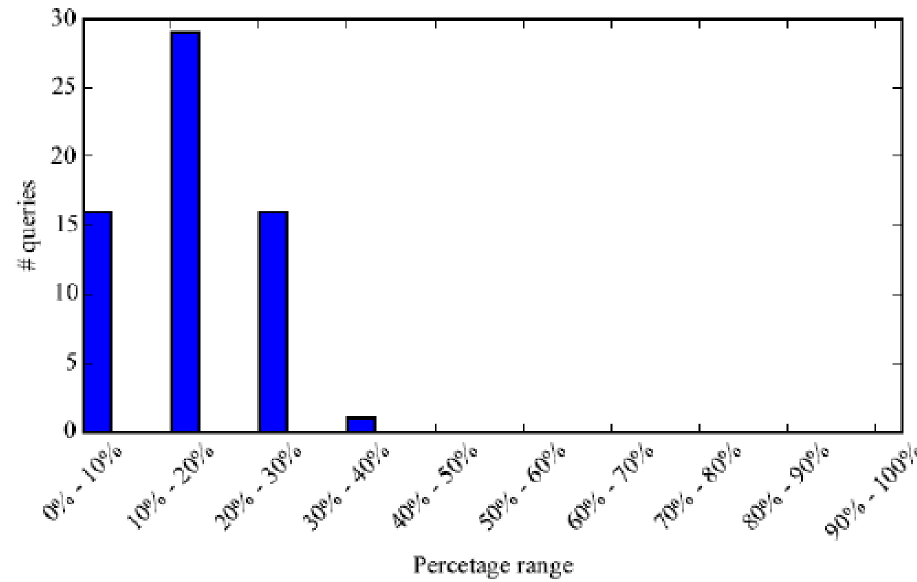
Ranking model: $f(q,d) = \langle \mathbf{w}, \vec{\phi}(q,d) \rangle = \left\langle \sum_{(i,j)} \alpha_{ij} \cdot \mathbf{x}_{ij}, \vec{\phi}(q,d) \right\rangle$

Motivation: There exist significant interactions among the training pairs, e.g., (doc1, doc2) and (doc1, doc3) share doc1. Whether there also exist interactions among model parameters? How to utilize the interactions if the answer is yes?

Low Rank Structure in Model Parameters



$$A(i,j) = \begin{cases} \alpha_{ij} & (i,j) \in P \\ 0 & \text{otherwise.} \end{cases}$$

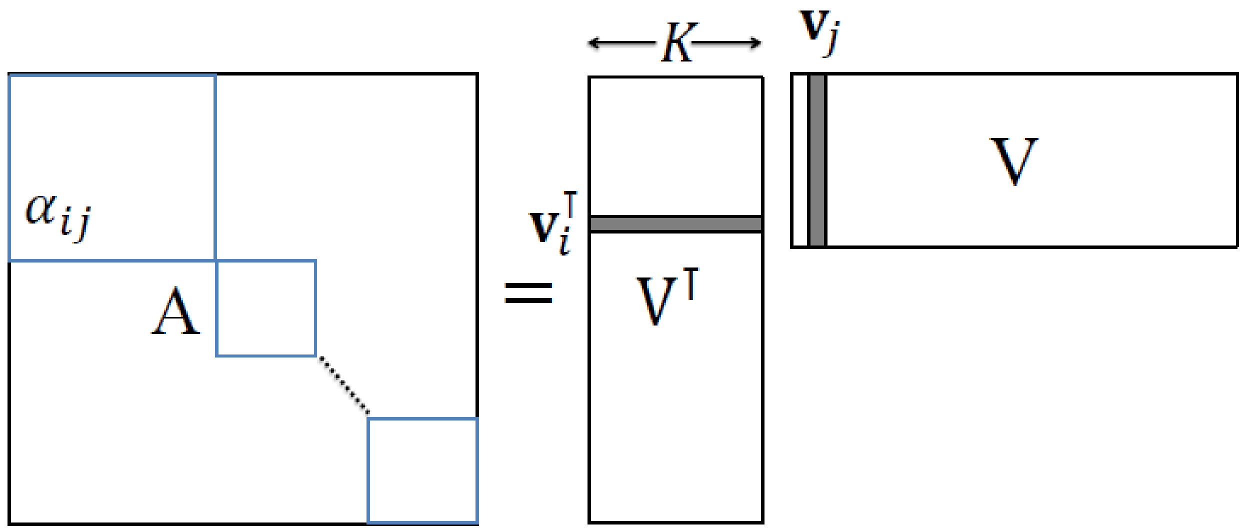


(a) Capturing 90% energy

- A: doc-doc matrix, $O(N^2)$ parameters
  - Block diagonal, each block corresponds to a query

Factorized Ranking SVM [Zhang et al., CIKM '15]

- Directly modeling the low rank structure $\alpha_{ij} = \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$



- V: doc-latent matrix, $O(KN)$ parameters
- K: number of latent dimensions

$$\mathbf{w} = \sum_{(i,j)} \alpha_{ij} \cdot \mathbf{x}_{ij} = \sum_{(i,j)} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \cdot \mathbf{x}_{ij}$$

Loss Functions

- Ranking SVM loss function

$$\min_{\boldsymbol{w}\in\mathcal{R}^n} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{(i,j)\in P}[1-\langle \boldsymbol{w}, \boldsymbol{x}_i - \boldsymbol{x}_j\rangle]_+$$

- Factorized Ranking SVM loss function

$$\min_{\boldsymbol{v}_1,\cdots,\boldsymbol{v}_N} \frac{1}{2}\left\|\underbrace{\sum_{(i,j)\in P}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle(\boldsymbol{x}_i - \boldsymbol{x}_j)}_{\mathbf{w}}\right\|^2 + C\sum_{(k,l)\in P}\left[1-\left\langle \underbrace{\sum_{(i,j)\in P}\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle(\boldsymbol{x}_i - \boldsymbol{x}_j)}_{\mathbf{w}}, \boldsymbol{x}_k - \boldsymbol{x}_l\right\rangle\right]_+$$

new parameters

Experiments

- Based on Letor datasets
- Outperformed all baselines including Ranking SVM
- More improvements can be achieved on datasets with denser preference pairs (OHSUMED)

Results on OHSUMED (dense preference pairs)

	MAP	NDCG@1	NDCG@3	NDCG@5
RSVM	0.4427	0.5289	0.4553	0.4392
RankNet	0.404	0.4007	0.3616	0.3388
ListNet	0.4443	0.5134	0.4664	0.4530
Fac-RSVM	**0.4463**	**0.5507**	**0.4798**	**0.4546**

Results on MQ2008 (sparse preference pairs)

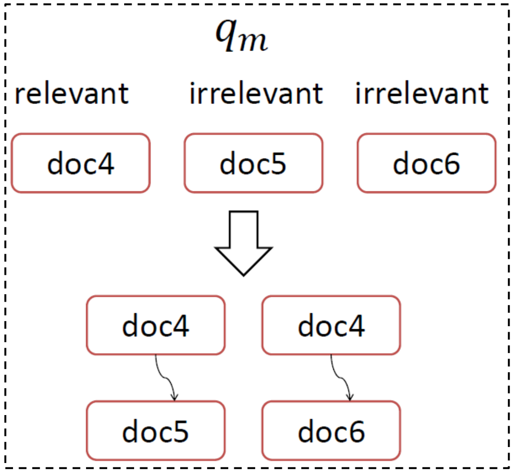
	MAP	NDCG@1	NDCG@3	NDCG@5
RSVM	0.4713	**0.3686**	0.4277	0.4730
RankNet	0.4522	0.3414	0.3991	0.4500
ListNet	0.4415	0.3244	0.3916	0.4396
Fac-RSVM	**0.4714**	0.3660	**0.4289**	**0.4731**

Summary

- There exists interactions over the training pairs in pairwise learning to rank
- The interactions lead to low rank structure among the Lagrange multipliers
- Explicitly model the low rank structure (Factorized Ranking SVM)
  - Improve ranking accuracies
  - Reduce the number of parameters $O(N^2) \to O(KN)$

Discussion

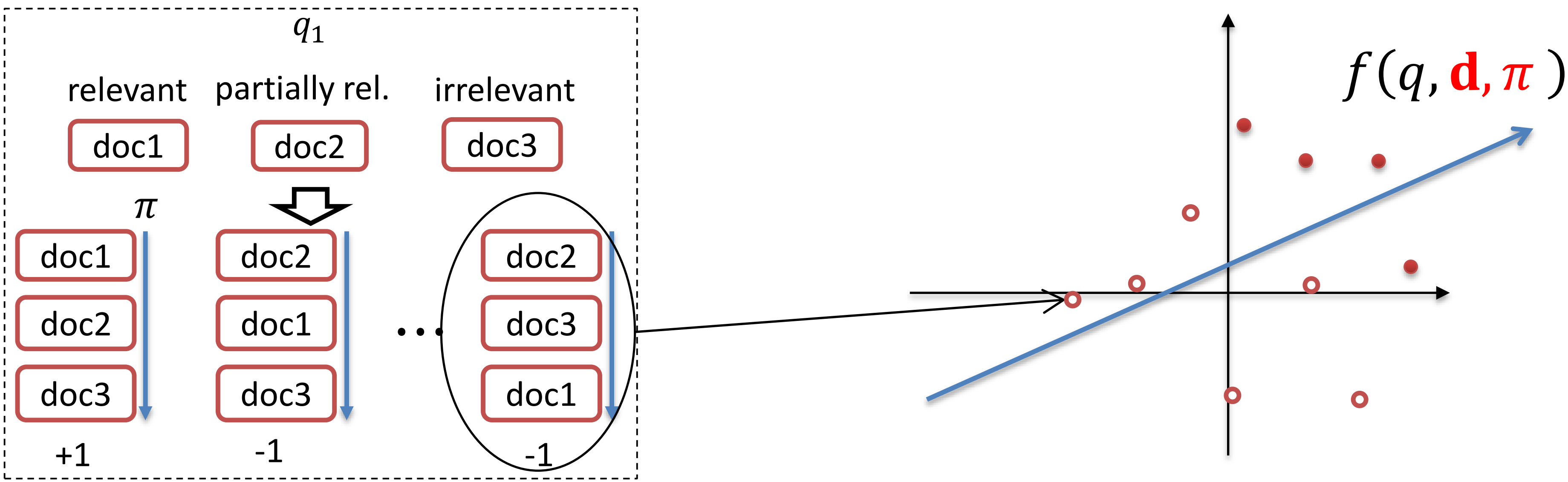
- Parameter interactions exist in a lot of learning to rank algorithms
  - Violate I.I.D. assumption to make formalization and optimization feasible
- Other Pairwise learning to rank algorithms



Pair (doc4, doc5) and (doc4, doc6) share one document doc4.

Discussion

- Listwise learning to rank
  - Generate “positive” and “negative” rankings as training data. The training instances have interactions.

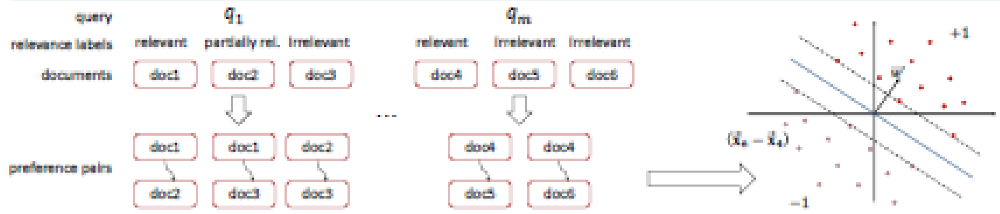


Training lists from one query are based on the same set of the documents.

Modeling Parameter Interactions in Ranking SVM

Yaogong Zhang[1], Jun Xu[2], Yanyan Lan[2], Jiafeng Guo[2], Maoqiang Xie[1], Yalou Huang[1], Xueqi Cheng[2]
[1]College of Computer and Control Engineering, Nankai University
[2]Institute of Computing Technology, Chinese Academy of Sciences

1. Pairwise learning to rank: ranking as binary classification over preference pairs



Motivation: two preference pairs could share one document, e.g., pairs (doc1, doc2) and (doc1, doc3) share doc1. There exist significant interactions over the training pairs.

2. Parameter interactions in Ranking SVM

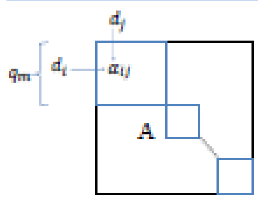
Ranking SVM

Prime
$$\min_{\mathbf{w}\in R^n} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{(i,j)\in P}\left[1-\langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j\rangle\right]_+$$

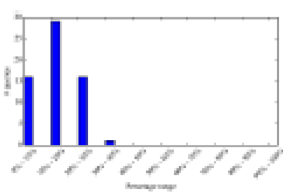
α_{ij} corresponds to preference pair (i,j)

Dual
$$\min_{\alpha} \frac{1}{2}\alpha^T \mathrm{M}\alpha - \boldsymbol{e}^T\alpha$$
$$s.t.\quad 0 \le \alpha_{ij} \le C, \forall (i,j)\in P$$

Low rank structure among Lagrange multipliers α_{ij}



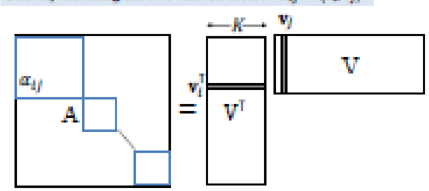
$$A(i,j) = \begin{cases} \alpha_{ij} & (i,j)\in P \\ 0 & \text{otherwise.}\end{cases}$$



(a) Capturing 90% energy

3. Factorized Ranking SVM

Directly modeling the low rank structure: $\alpha_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j\rangle$



New loss function

new parameters

$$\min_{\mathbf{v}_1,\cdots,\mathbf{v}_N} \frac{1}{2}\left\|\sum_{(i,j)\in P}\langle \mathbf{v}_i, \mathbf{v}_j\rangle(\mathbf{x}_i - \mathbf{x}_j)\right\|^2 +$$
$$C\sum_{(k,l)\in P}\left[1-\left\langle \sum_{(i,j)\in P}\langle \mathbf{v}_i, \mathbf{v}_j\rangle(\mathbf{x}_i - \mathbf{x}_j), \mathbf{x}_k - \mathbf{x}_l\right\rangle\right]_+$$

Number of parameters: $O(N^2) \to O(KN)$

4. Experiments

Results on OHSUMED (dense preference pairs)

	MAP	NDCG@1	NDCG@3	NDCG@5
RSVM	0.4427	0.5289	0.4553	0.4392
RankNet	0.404	0.4007	0.3616	0.3388
ListNet	0.4443	0.5134	0.4664	0.4530
Fac-RSVM	0.4463	0.5507	0.4798	0.4546

Results on MQ2008 (sparse preference pairs)

	MAP	NDCG@1	NDCG@3	NDCG@5
RSVM	0.4713	0.3686	0.4277	0.4730
RankNet	0.4522	0.3414	0.3991	0.4500
ListNet	0.4415	0.3244	0.3916	0.4396
Fac-RSVM	0.4714	0.3660	0.4289	0.4731

1) Factorized Ranking SVM outperformed all baselines including Ranking SVM.
2) More improvements can be achieved on datasets with denser preference pairs.

5. Conclusion

1) There exists a low-rank structure among the Lagrange multipliers of Ranking SVM.
2) Factorized Ranking SVM decomposes each Lagrange multiplier as a dot product of two low-dimensional vectors.
3) Factorized Ranking SVM decreases space complexities from $O(N^2)$ to $O(KN)$.
4) Experimental results showed that Factorized Ranking SVM outperformed all baselines.

Outline

- Learning to rank
- Our progresses
  - Improving existing algorithms
  - Adventure with new ranking tasks
- Summary

Existing Work Focuses on Relevance Ranking

Categorization of the Algorithms

Category	Algorithms
Pointwise Approach	**Regression**: Least Square Retrieval Function (TOIS 1989), Regression Tree for Ordinal Class Prediction (Fundamenta Informaticae, 2000), Subset Ranking using Regression (COLT 2006), … **Classification**: Discriminative model for IR (SIGIR 2004), McRank (NIPS 2007), … **Ordinal regression**: Pranking (NIPS 2002), OAP-BPM (EMCL 2003), Ranking with Large Margin Principles (NIPS 2002), Constraint Ordinal Regression (ICML 2005), …
Pairwise Approach	Learning to Retrieve Information (SCC 1995), Learning to Order Things (NIPS 1998), Ranking SVM (ICANN 1999), RankBoost (JMLR 2003), LDM (SIGIR 2005), RankNet (ICML 2005), Frank (SIGIR 2007), MHR(SIGIR 2007), GBRank (SIGIR 2007), QBRank (NIPS 2007), MPRank (ICML 2007), IRSVM (SIGIR 2006), …
Listwise Approach	**Listwise loss minimization**: RankCosine (IP&M 2008), ListNet (ICML 2007), ListMLE (ICML 2008), … **Direct optimization of IR measure**: LambdaRank (NIPS 2006), AdaRank (SIGIR 2007), SVM-MAP (SIGIR 2007), SoftRank (LR4IR 2007), GPRank (LR4IR 2007), CCA (SIGIR 2007), …

- A single scoring function for all queries, documents, and ranking positions
- Score for one document is independent of other documents
- Scores independent of ranking positions

Search Result Diversification

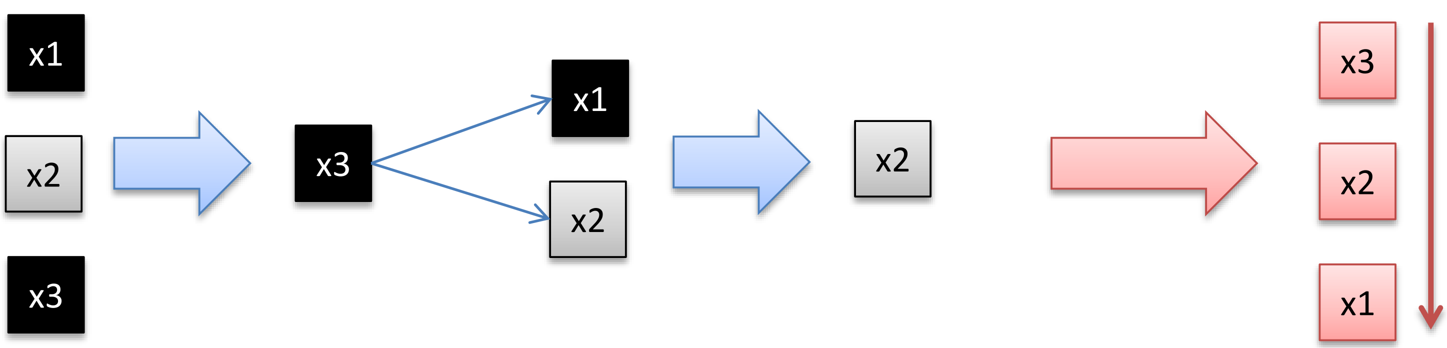
MMR for Search Result Diversification

- Maximal Marginal Relevance (MMR) [Carbonell & Goldstein, SIGIR '98]
  - Scoring: query-doc relevance + doc-doc similarity

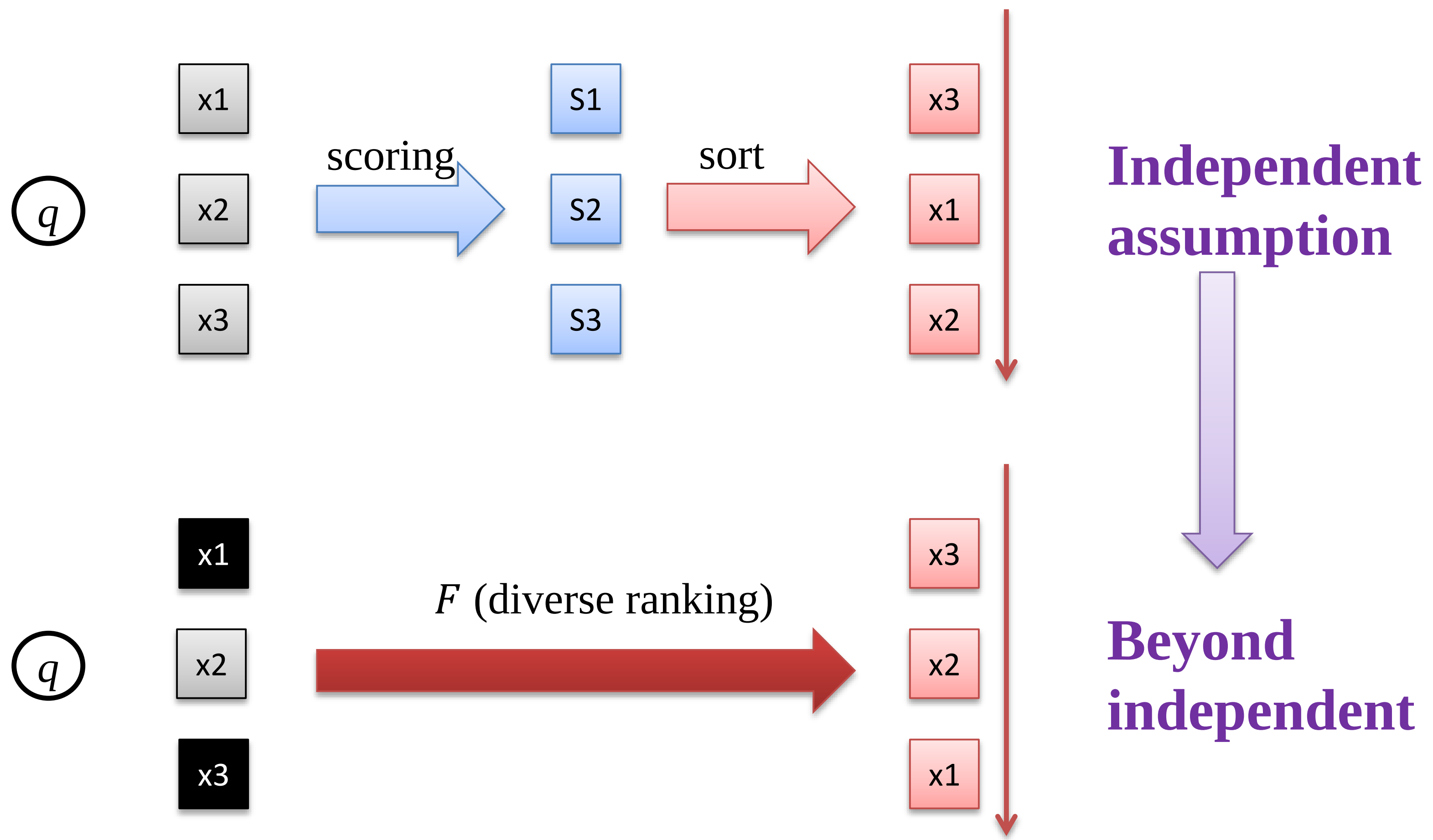
$$MMR \stackrel{\text{def}}{=} \operatorname{Arg}\max_{D_i \in R \setminus S} \left[\lambda Sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j)\right]$$

relevance similarity

  - Ranking: sequential document selection



Beyond Independent Assumption



Learning to Rank Model for Diversification

- Scoring function: relevance + similarity

$$f_S(\mathbf{x}_i, R_i) = \underbrace{\omega_r^T \mathbf{x}_i}_{\text{relevance}} + \underbrace{\omega_d^T h_S(R_i)}_{\text{similarity}}, \forall \mathbf{x}_i \in X \backslash S$$

  – Parameters to learn: $(\boldsymbol{\omega}_r, \boldsymbol{\omega}_d)$

- Ranking: sequential document selection
  – Scoring function for position *n* depends on the documents selected for the previous *n*-1 positions

Learning the Scoring Function

- Generative approach (R-LTR) [Zhu et al., SIGIR ‘14]
  - Simulating the process of sequential document selection with Plackett-Luce model
  - Optimizing with MLE

- Discriminative approach (PAMM) [Xia et al., SIGIR ‘15]
  - Maximizing the margin between “positive” and “negative” rankings
  - Directly optimizes (any) diverse ranking measures
  - Optimizing with structured Perceptron

Experimental Results

	WT2009		WT2010		WT2011	
Method	ERR-IA@20	α-NDCG@20	ERR-IA@20	α-NDCG@20	ERR-IA@20	α-NDCG@20
QL	0.164	0.269	0.198	0.302	0.352	0.453
ListMLE	0.191	0.307	0.244	0.376	0.417	0.517
MMR	0.202	0.308	0.274	0.404	0.428	0.530
xQuAD	0.232	0.344	0.328	0.445	0.475	0.565
PM-2	0.229	0.337	0.330	0.448	0.487	0.579
SVM-DIV	0.241	0.353	0.333	0.459	0.490	0.591
StructSVM(ERR-IA)	0.261	0.373	0.355	0.472	0.513	0.613
StructSVM(α-NDCG)	0.260	0.377	0.352	0.476	0.512	0.617
R-LTR	0.271	0.396	0.365	0.492	0.539	0.630
PAMM(ERR-IA)	**0.294**	0.422	**0.387**	0.511	**0.548**	0.637
PAMM(α-NDCG)	0.284	**0.427**	0.380	**0.524**	0.541	**0.643**

- PAMM and R-LTR significantly outperforms the baselines, including the non-learning models and relevance learning to rank models
- PAMM can improve the performance w.r.t. a measure by directly optimizing the measure in training phase

Next Step

- MMR is not the only criterion for search result diversification
- Diversity features are hard to define
  - Relationship between one document and a set of selected documents
  - Can the model automatically learn diversity features from existing document representations?
  - Preliminary experiments showed it does work!

Outline

- Learning to rank
- Our advances
  - Improving existing algorithms
  - Adventure with new applications
- Summary

Summary

- A lot of work on learning to rank
- However, we still have a long way to go
  - Existing algorithms are not perfect
  - New ranking tasks are waiting for solutions

Acknowledgement

Yanyan Lan

Jiafeng Guo

Ph.D. students:

Long Xia (ICT, CAS)

Yaogong Zhang (Nankai University)

References

- Yaogong Zhang, Jun Xu, Yanyan Lan, Jiafeng Guo, Maoqiang Xie, Yalou Huang, and Xueqi Cheng. Modeling Parameter Interactions in Ranking SVM. Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM '15).
- T. Joachims, Optimizing Search Engines Using Clickthrough Data, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.
- I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun. Support Vector Machine Learning for Interdependent and Structured Output Spaces. International Conference on Machine Learning (ICML), 2004.
- Jaime Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. SIGIR 1998.
- Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng and Shuzi Niu, Learning for Search Result Diversification. Proceedings of the 37th Annual ACM SIGIR Conference, GoldCoast, Australia, 2014. (SIGIR 2014).
- Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. Proceedings of the 38th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '15).

Thanks!

www.bigdatalab.ac.cn/~junxu

junxu@ict.ac.cn