

Learning to Rank:

How Commercial Search Engines use Machine Learning to Rank Search Results?

Yasser Ganjisaffar

PhD Candidate

University of California, Irvine

Joint work with Rich Caruana (Microsoft)

Ranking Problem



ics uci
About 878,000 results (0.17 seconds)

Everything
Images

▶ donald bren school of information and computer sciences @ the ...
Department of Information and Computer Science. Teaching and research information.

Web Images Videos Shopping News Maps More | MSN Hotmail



ics uci



SEARCH HISTORY
Search more to see your

ALL RESULTS

1-10 of 725,000 results · [Advanced](#)

[University of California - Irvine](#)

amazon.com Hello, Cristina Lopes. We have recom...
Cristina's Amazon.com Today's

Shop All Departments Search Computer Programming

Books Advanced Search

Department
Any Department
Books
Computers & Internet
Programming
Software Design, Testing & Engineering (7)
Languages & Tools (10)
Java (3)
Algorithms (2)
Graphics & Multimedia (2)

Format
Any Format
HTML (2)
Printed Books (17)

Binding
Any Binding
Paperback (15)
Hardcover (2)

Shipping Option (What's this?)
Any Shipping Option
Free Super Saver Shipping

Avg. Customer Review
Any Avg. Customer Review
★★★★☆ & Up (13)
★★★★☆ & Up (17)
★★★★☆ & Up (17)
★★★★☆ & Up (17)

Condition
Any Condition
New (19)
Used (17)

Availability
Include Out of Stock

Search Graduate

10 of 395 Open as list

[Allgrads] Board Game + BBQ after the Grad
to: allgrads@ics.uci.edu

Hi all,

In conjunction with this week's grad
a BBQ + Board Game night at the PV C
There are about 40 hot dogs, 20-ish

West Coast Sea Grant Fellowship -- apply by
to: Multiple recipients of list

=====

New 2011-2013 West Coast Sea Grant F
DEADLINE: Apply by October 7, 2010..

Graduate Student Welcome Week Party, 9/
to: Multiple recipients of list

Come celebrate the new academic year
Students (AGS), as well as old and p

Don't miss out on our biggest event.
Event will be held Thursday, 9/23, f

[Allgrads] TA/Reader Application Now Open
to: allgrads@ics.uci.edu

Hi Everybody,

The TA/Reader application for Winter
is now available. The deadline to s
is Friday, November 12 at 4pm. Ther

1. **LOOK INSIDE!**
 Cassandra

2. **Author Pages**

3. **LOOK INSIDE!**



ics uci

Search

Search results for ics uci

About 117 results

Search options

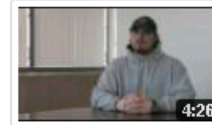


UCI Commencement: Engineering and ICS

June 12, 2009

1 year ago | 160 views

by frannycheska | [engineering](#) | [uc irvine](#)



Student Ambassadors @ UC Irvine

The Bren School of ICS Student Ambassadors program features current students that are the face and voice of the Bren School to prospective ...

1 year ago | 164 views

by UCIBrenICS | [university of cal...](#) | [people to people ...](#)



Project:ICS In-Class Internship Program @ UC Irvine

A brief overview of Project:ICS (Innovative Collaborative Solutions), the Bren School of ICS @ UC Irvine's in-class internship program, that allows ...

2 years ago | 353 views

by UCIBrenICS | [intern](#) | [uc irvine](#)



Kristina Winbladh Profile @ UC Irvine

Kristina Winbladh was a Ph.D. student at UCI's Bren School of Information and Computer Sciences before receiving her Ph.D. and moving on to a ...

5 months ago | 218 views

by UCIBrenICS | [university of cal...](#) | [uc irvine](#)



UCI Fusion: Lucky [Fox] vs Fly Amanita [ICs] 2

UCI Fusion Singles Winner's Semi's

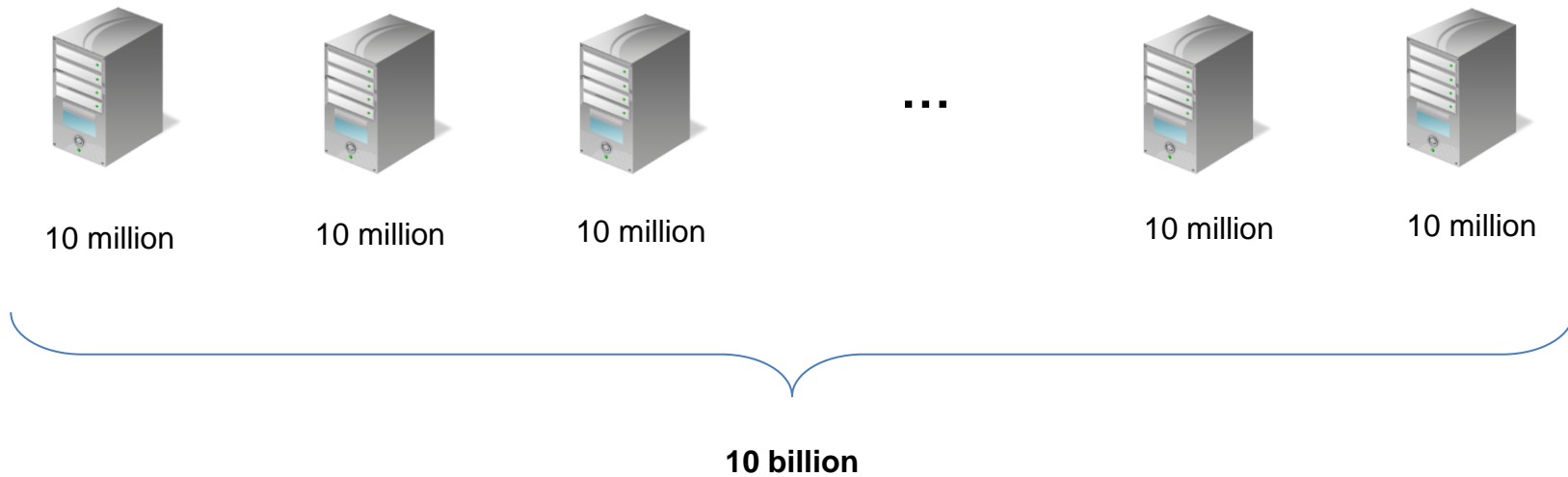
7 months ago | 175 views

by stabbedbyahippie | [fox mccloud](#) | [m2k](#)

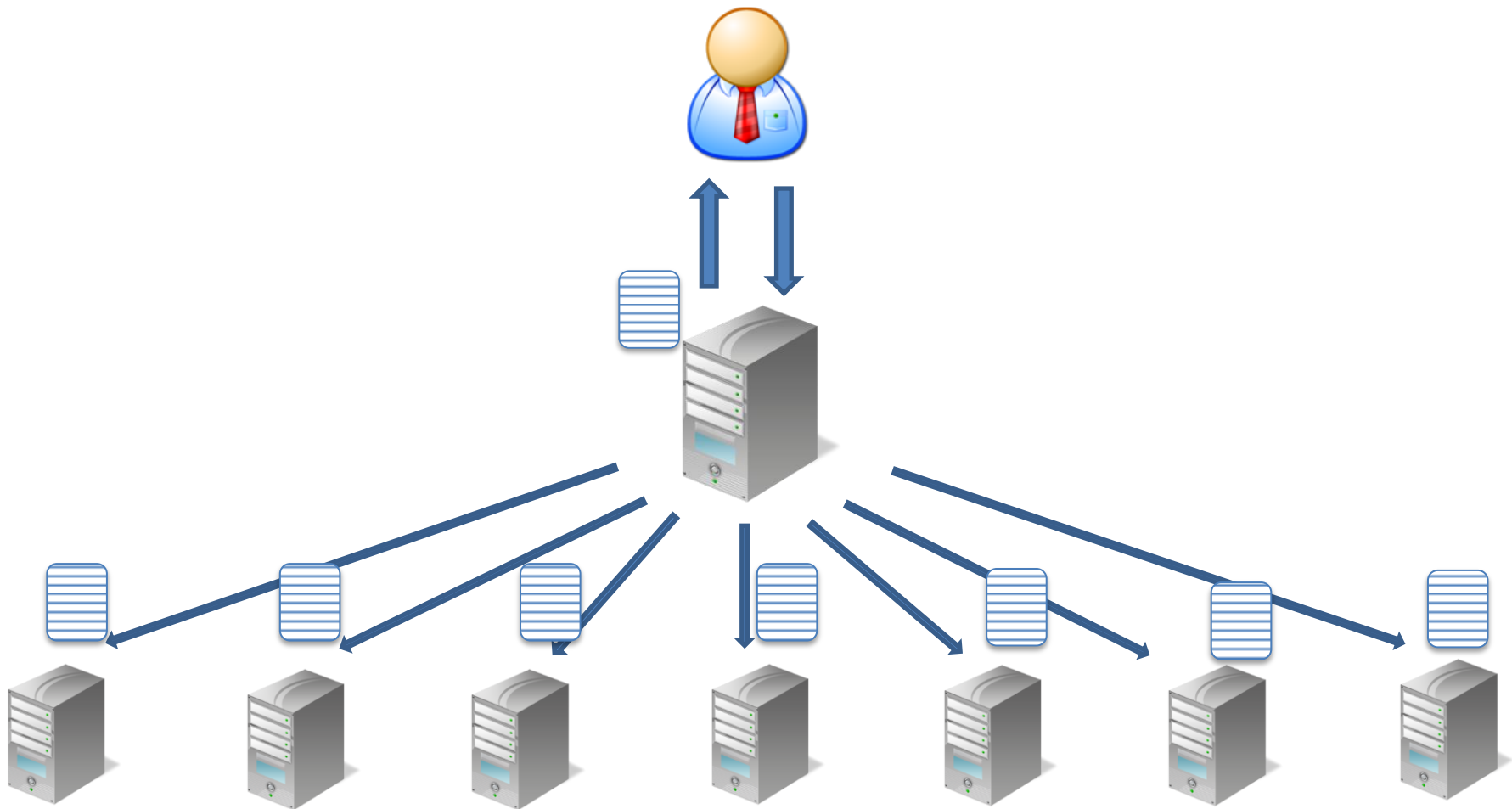
Ranking in Web Scale

Estimated index size:

- Google: 18 billion
- Bing: 10 billion



Ranking in Web Scale



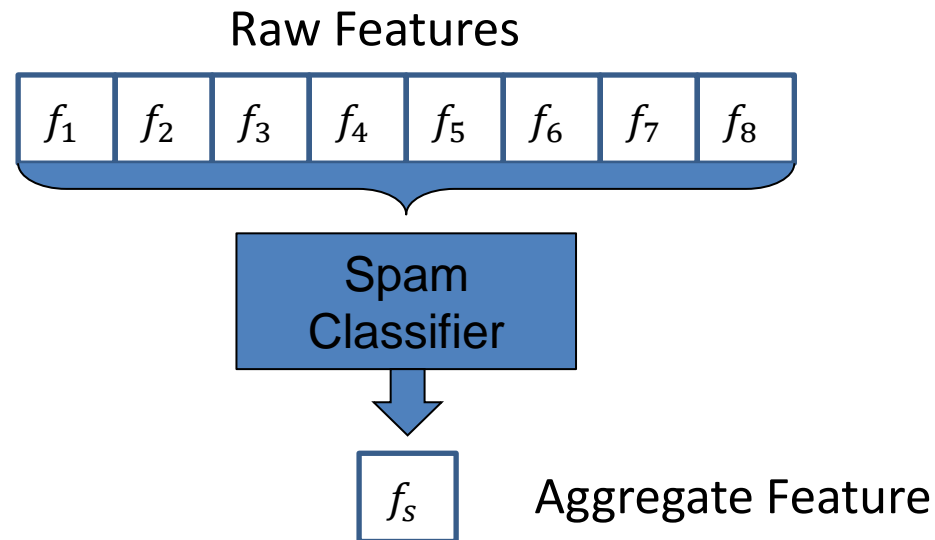
How to Rank Search Results?

- Looking at features of each document:
 - How many times keywords appear in title/body/anchors/url?
 - PageRank of url
 - Is there a phrase match?
 - Is this document clicked for same query by other users?
 - ...



Ranking in Commercial Search Engines

- Google: about 200 aggregate features
- Bing: about 1000 raw features
- Yahoo!: at least 600 raw features



Ranking in Commercial Search Engines



- Google
 - Hand-tuned algorithms
- Bing, Yahoo!, Yandex, ...
 - Machine Learning

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

$\text{IDF}(q_1)$	$\text{IDF}(q_2)$	$\text{IDF}(q_3)$
$f(q_1, D)$	$f(q_2, D)$	$f(q_3, D)$
$ D $	avgdl	

Evaluation Metrics

NDCG@k

$$NDCG@k \propto \sum_{j=1}^k \frac{2^{r_j} - 1}{\log_2(1 + j)}$$

MAP

$$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q}$$

Discontinuous and **nondifferentiable** functions

Variance in Measurements

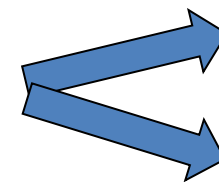
NDCG@10 = 0.6412

STDEV = 0.01



New NDCG@10 = 0.6452

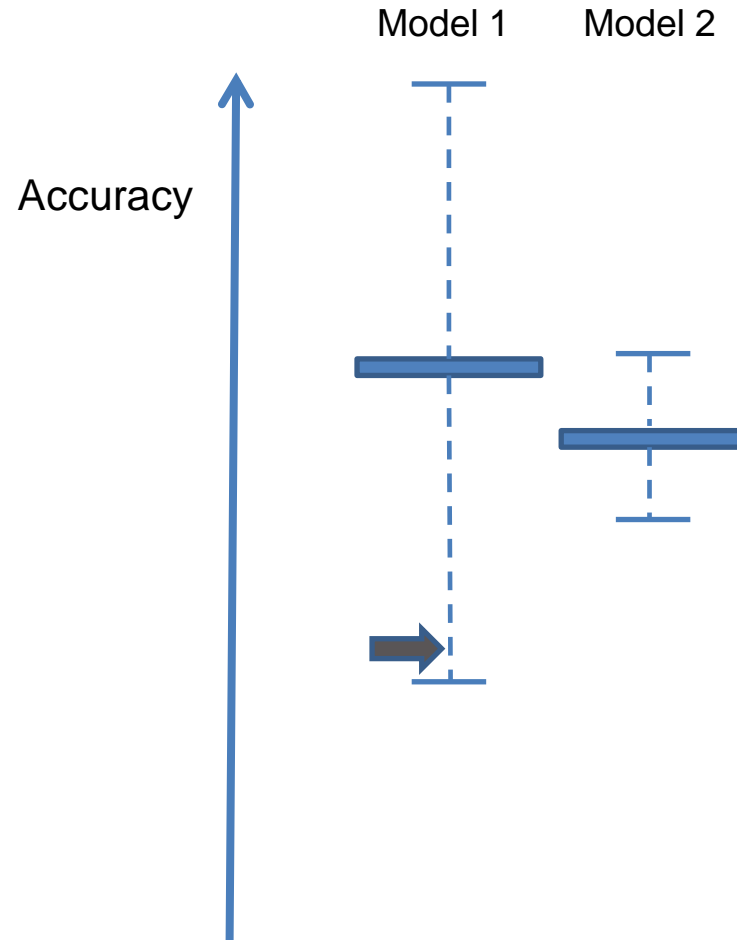
STDEV = 0.01



0.6552

0.6352

Variance in Measurements



Click-through Data: Implicit Feedback



Assuming that user has checked results from top to bottom:

2 is more relevant than 1

5 is more relevant than 1, 3, 4

7 is more relevant than 1, 3, 4, 6

(2,1)	(5,1)	(5,3)	(5,4)
(7,1)	(7,3)	(7,4)	(7,6)

Collecting Training Data

Query	URL	Label
ics uci	www.ics.uci.edu	Perfect
ics uci	vision.ics.uci.edu	Fair
ics uci	www.cs.uci.edu	Good
ics uci	www.ietf.org/rfc/rfc2396.txt	Bad
bren hall uci	en.wikipedia.org/wiki/Donald_Bren	Bad
bren hall uci	http://www.ics.uci.edu/about/brenhall/	Excellent

- 4 Perfect
- 3 Excellent
- 2 Good
- 1 Fair
- 0 Bad

Is Ranking a Classification Problem?

- Perfect classification leads to perfect ordering
- A completely incorrect classification might still result in perfect ordering

Document	True Class	Predicted Class
d_1	4	3
d_2	2	1
d_3	1	0

- Order between class labels
- Rows are related to each other through queries

Approaches to Learning-to-rank

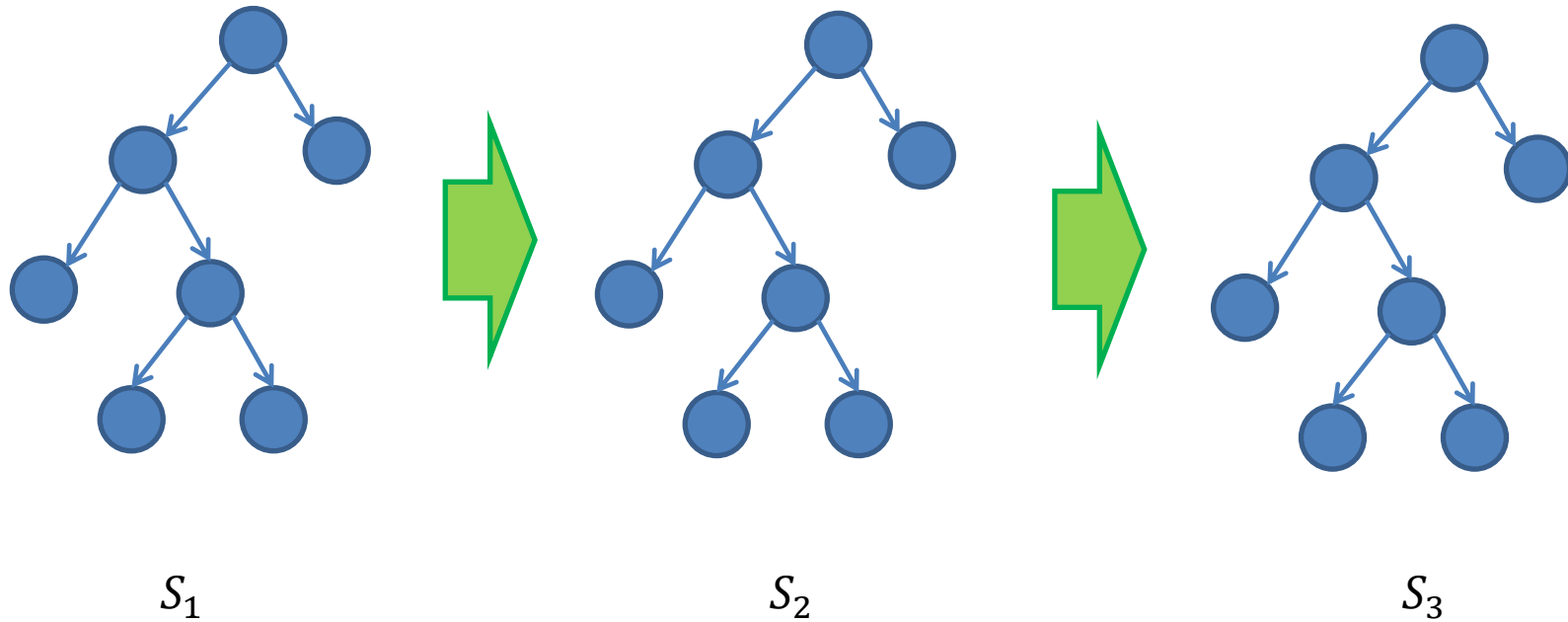
- **Pointwise**
 - Using existing learning methods for ranking: Regression and Classification
 - Documents and their ground truth labels are considered i.i.d. random variables.
- **Pairwise**
 - Ranking problem is reduced to classification problem on document pairs.
 - Document pairs are not independent (violates the basic assumption of classification)
- **Listwise**
 - Direct optimization of IR evaluation measures
 - Continuous and differentiable approximations of IR evaluation metrics
 - Continuous and differentiable bound of the IR metric
 - Using optimization frameworks that allow optimizing complex objectives

Yahoo! Learning to Rank Challenge:

May 2010

Team	Approach
C. Burges et al. [Microsoft Research]	Weighted Average of several Ensembles of Gradient Boosting
E Gottschalk et al. [Activision Blizzard & Data Mining Solutions]	Weighted Average of several Ensembles of Gradient Boosting and Random Forests
D. Pavlov et al. [Yandex Labs]	Ensembles of Gradient Boosting

Gradient Boosted Trees



$$S = S_1 + S_2 + S_3$$

Gradient Boosted Trees: Binary Classification

Labels: $y_i \in \{\pm 1\}$

$F(x)$: the model score for sample $x \in R^n$

Notations:

$$P_+ = P(y = +1|x)$$

$$\bar{P}_+(x_i) = \begin{cases} 1 & y_i = +1 \\ 0 & \text{otherwise} \end{cases}$$

Cross-entropy loss function:

$$L(y, F) = -\bar{P}_+ \log P_+ - (1 - \bar{P}_+) \log(1 - P_+)$$

$$P_+ \equiv \frac{1}{1 + e^{-F(x)}}$$



$$L(y, F) = \log(1 + e^{-yF(x)})$$

Gradient Boosted Trees: Binary Classification

Goal is to minimize: $L(y, F) = \log(1 + e^{-yF(x)})$

$$F_m(x_i) = F_{m-1}(x_i) + \gamma_{im}$$

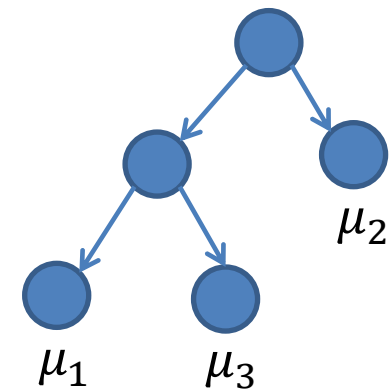
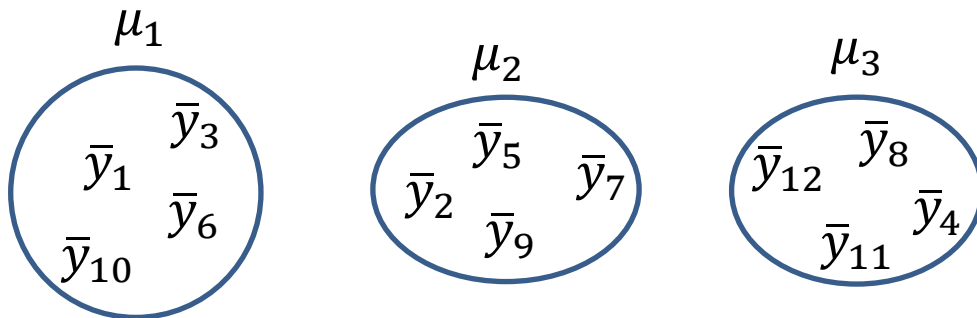
Newton-Raphson method for minimization:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

$$\gamma_{im} = -\frac{f'}{f''} = \frac{\bar{y}_i}{|\bar{y}_i|(1 - |\bar{y}_i|)} \quad \bar{y}_i \equiv -\frac{\partial L}{\partial F} = \frac{y_i}{1 + e^{y_i F}}$$

Gradient Boosted Trees: Binary Classification

$$\gamma_{im} = -\frac{f'}{f''} = \frac{\bar{y}_i}{|\bar{y}_i|(1 - |\bar{y}_i|)}$$



$$\gamma_{jm} = -\frac{f'}{f''} = \frac{\sum_{x_i \in R_{jm}} \bar{y}_i}{\sum_{x_i \in R_{jm}} |\bar{y}_i|(1 - |\bar{y}_i|)}$$

Regression Trees



(features₁, y₁)

(features₂, y₂)

(features₃, y₃)

(features₄, y₄)

(features₅, y₅)

$$Error = \sum_i (y_i - \mu)^2$$

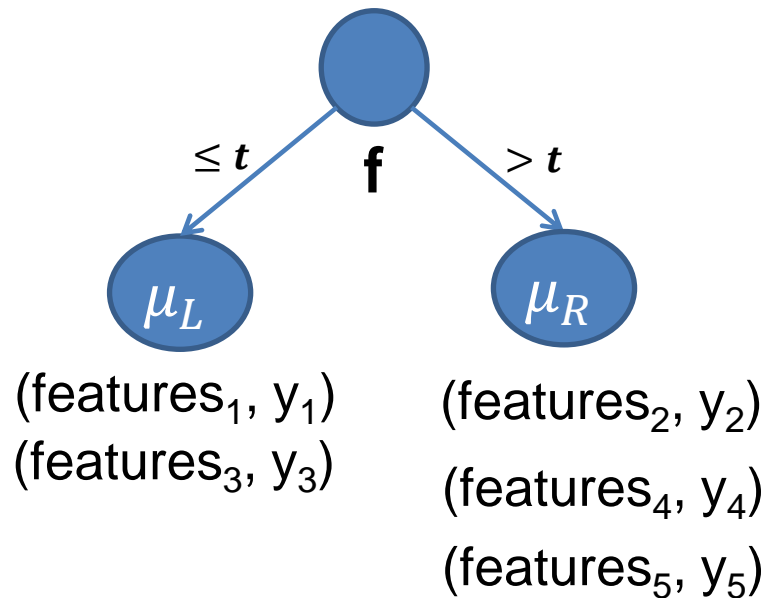
Initially, all samples are at the root of the tree

The output value on the root node would be the

Average of targets.

Regression Trees

What is the best feature and threshold for splitting the samples?



$$S = \sum_i (y_i - \mu)^2 \quad \longrightarrow \quad S_j = \sum_{i \in L} (y_i - \mu_L)^2 + \sum_{i \in R} (y_i - \mu_R)^2$$

$$\text{Split gain} = S - S_j$$

Regression Trees

- We need to compute gain of **EVERY** feature and threshold combination and then pick the best one.
- How can we do it fast?

Split Gain

$$\begin{aligned} S_j &= \sum_{i \in L} (y_i^2 + \mu_L^2 - 2y_i \mu_L) + \sum_{i \in R} (y_i^2 + \mu_R^2 - 2y_i \mu_R) \\ &= \left(\sum_i y_i^2 \right) + \|L\| \mu_L^2 + \|R\| \mu_R^2 - 2\mu_L \sum_{i \in L} y_i - 2\mu_R \sum_{i \in R} y_i \\ &= \left(\sum_i y_i^2 \right) - \|L\| \mu_L^2 - \|R\| \mu_R^2 = \left(\sum_i y_i^2 \right) - \left(\frac{\text{sum}_L^2}{\|L\|} + \frac{\text{sum}_R^2}{\|R\|} \right) \end{aligned}$$

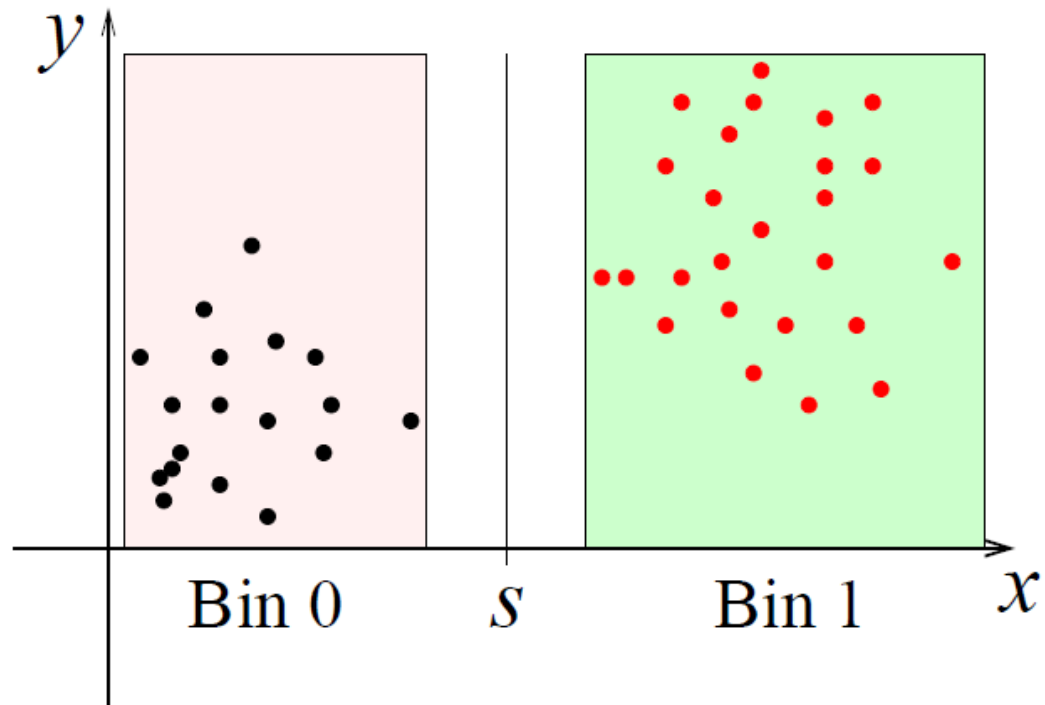
$$\begin{aligned} S &= \sum_i (y_i - \mu)^2 = \left(\sum_i y_i^2 \right) + \|total\| \mu^2 - 2\mu \sum_i y_i = \left(\sum_i y_i^2 \right) - \|total\| \mu^2 \\ &= \left(\sum_i y_i^2 \right) - \frac{\text{sum}^2}{\|total\|} \end{aligned}$$

Fixed for all
feature/thresholds

$$\begin{aligned} \text{Split gain} &= S - S_j \\ &= \left(\frac{\text{sum}_L^2}{\|L\|} + \frac{\text{sum}_R^2}{\|R\|} \right) - \frac{\text{sum}^2}{\|total\|} \end{aligned}$$

Thresholds

Which thresholds should we examine?



Max: 32,767 bins \longrightarrow 2 byte integers

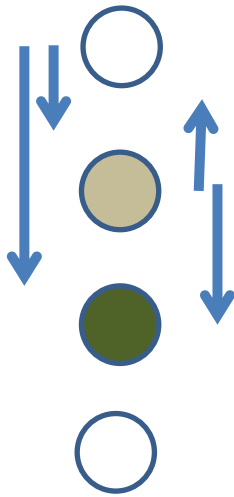
Ranking as Binary Classification

$(d_i, d_j): +1$

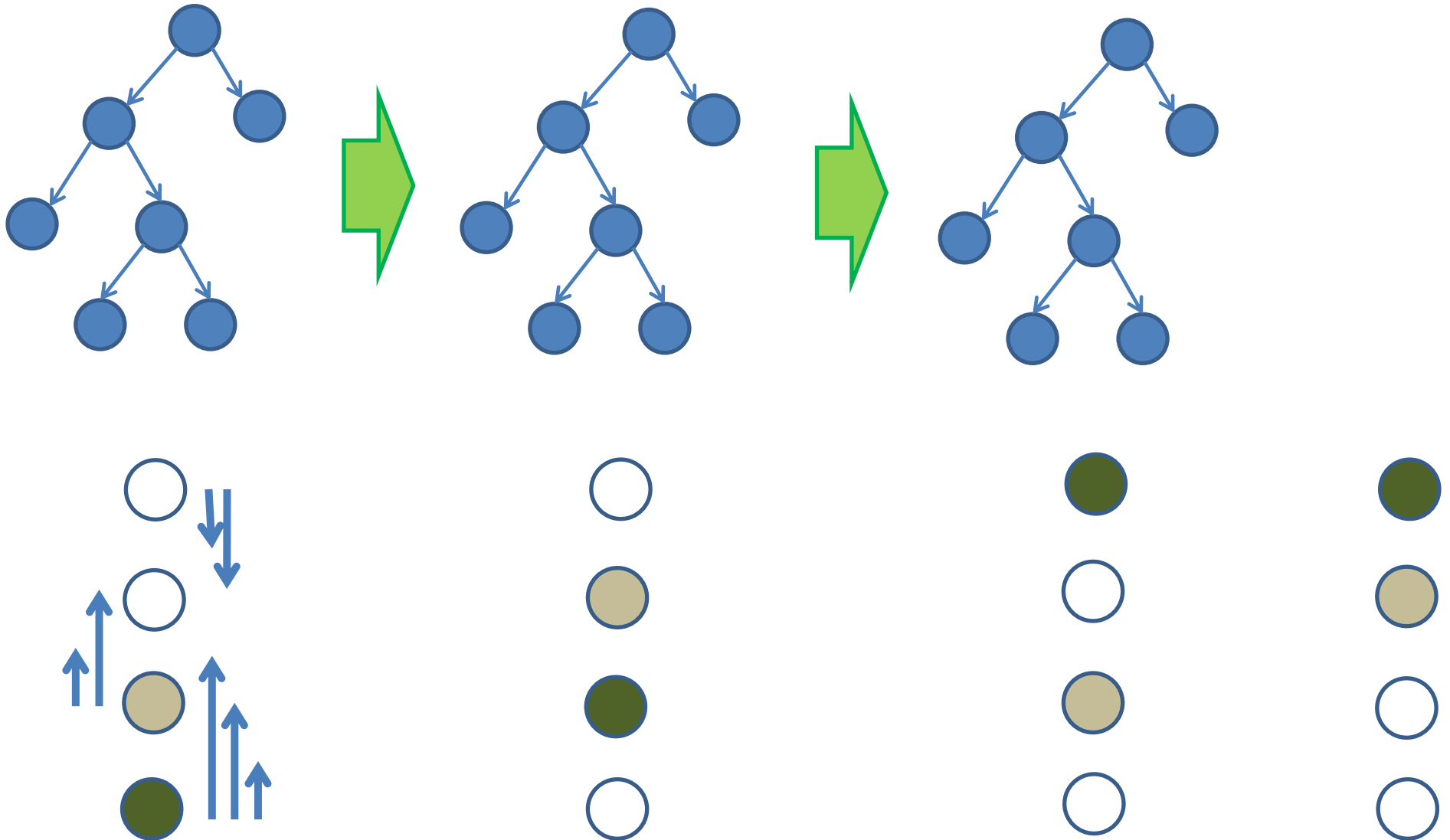
d_i should be ranked **higher** than d_j

$(d_i, d_j): -1$

d_i should be ranked **lower** than d_j



Gradient Boosted Trees



Gradient Boosted Trees

Binary Classification

$$P_+ \equiv \frac{1}{1 + e^{-F(x)}}$$

$$L(y, F) = \log(1 + e^{-yF(x)})$$

$$\frac{\partial L}{\partial F} = \frac{-y_i}{1 + e^{y_i F}}$$

Ranking

$$P_{ij} \equiv \frac{1}{1 + e^{-(F(x_i) - F(x_j))}}$$

$$L(y, F) = \log(1 + e^{-y(F(x_i) - F(x_j))})$$

$$\lambda_{ij} \equiv \frac{\partial L}{\partial F} = \frac{-y_i}{1 + e^{(F(x_i) - F(x_j))}} |\Delta_{NDCG}|$$

LambdaMART

PART 2

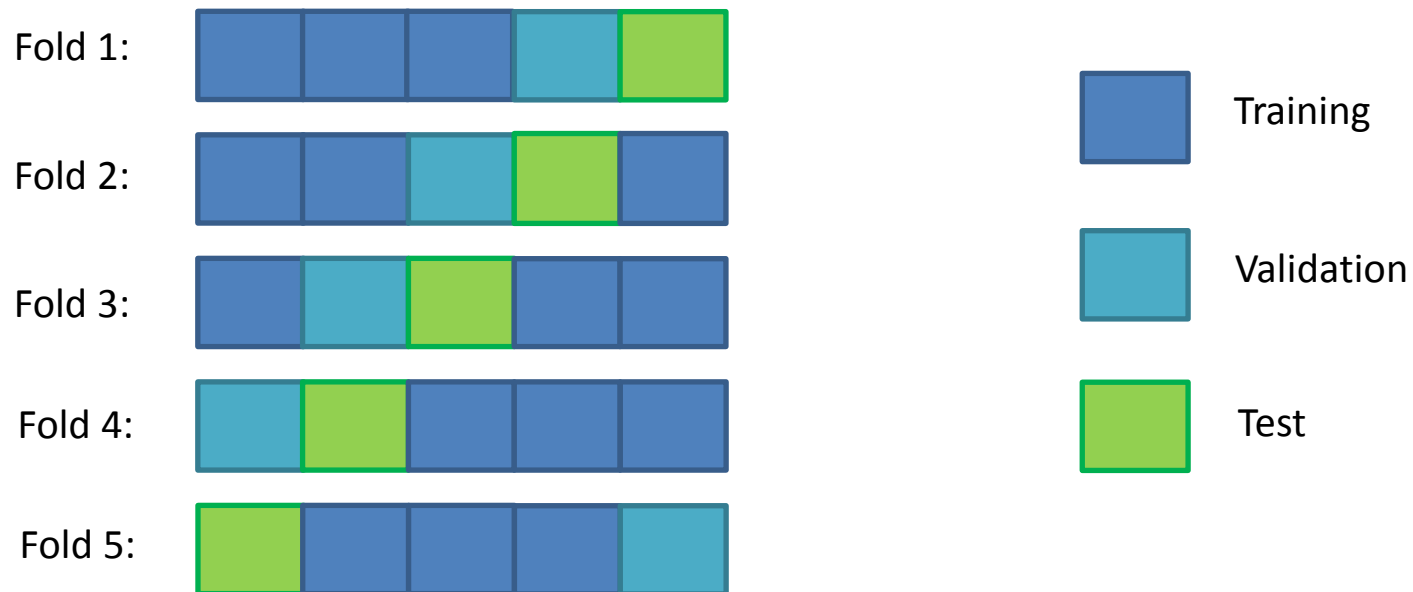
On Improving LambdaMART

- Parameter Tuning
- Improving Accuracy and Reducing Variance
- Compressing the Final Model

Data sets

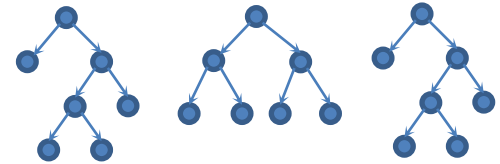
Data set	Queries	Avg. Docs Per Query	Features	Labels
TD2004	75	989	64	{0, 1}
MQ2007	1,692	41	46	{0, 1, 2}
MSLR-WEB10K	10,000	120	136	{0, 1, 2, 3, 4}
Yahoo LTRC	29,921	24	519	{0, 1, 2, 3, 4}

$$10,000 \times 120 \times \$0.20 = \$240,000$$



LambdaMART Parameters

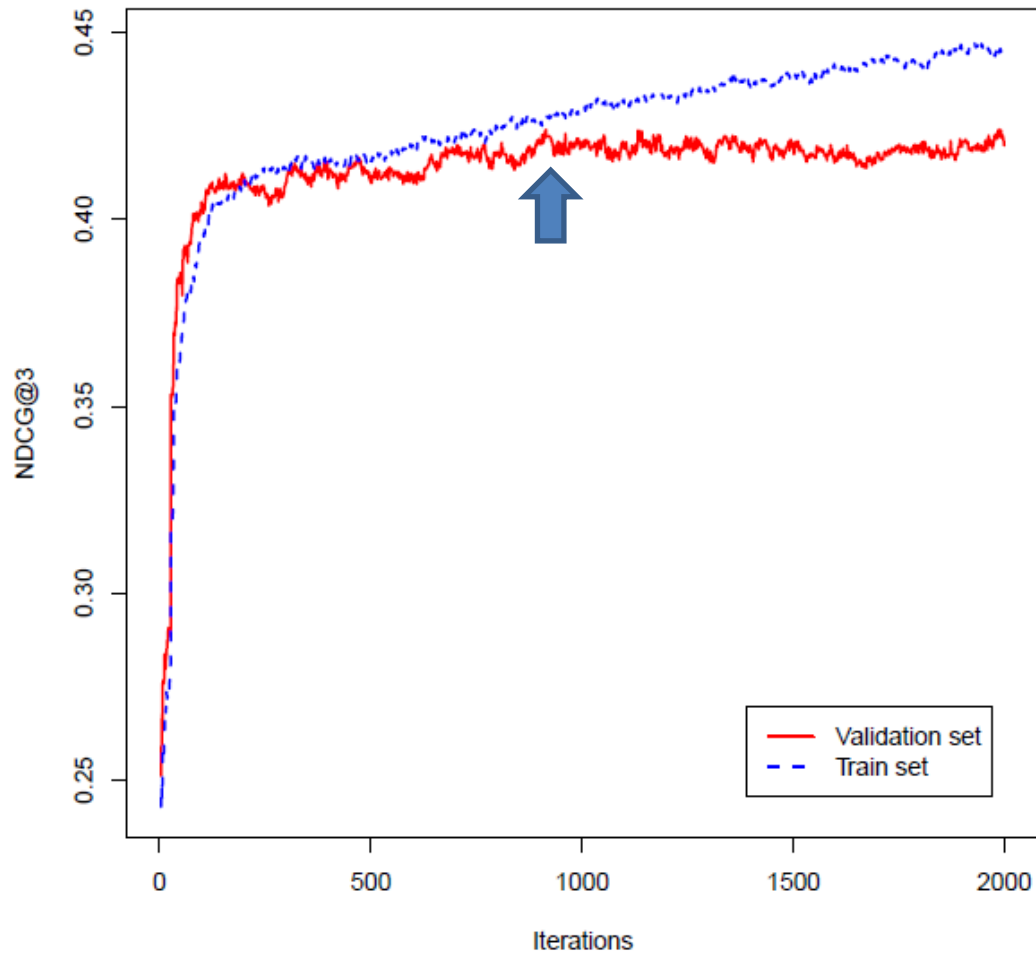
- Maximum Number of Leaves
- Min observations per leaf
- Learning rate



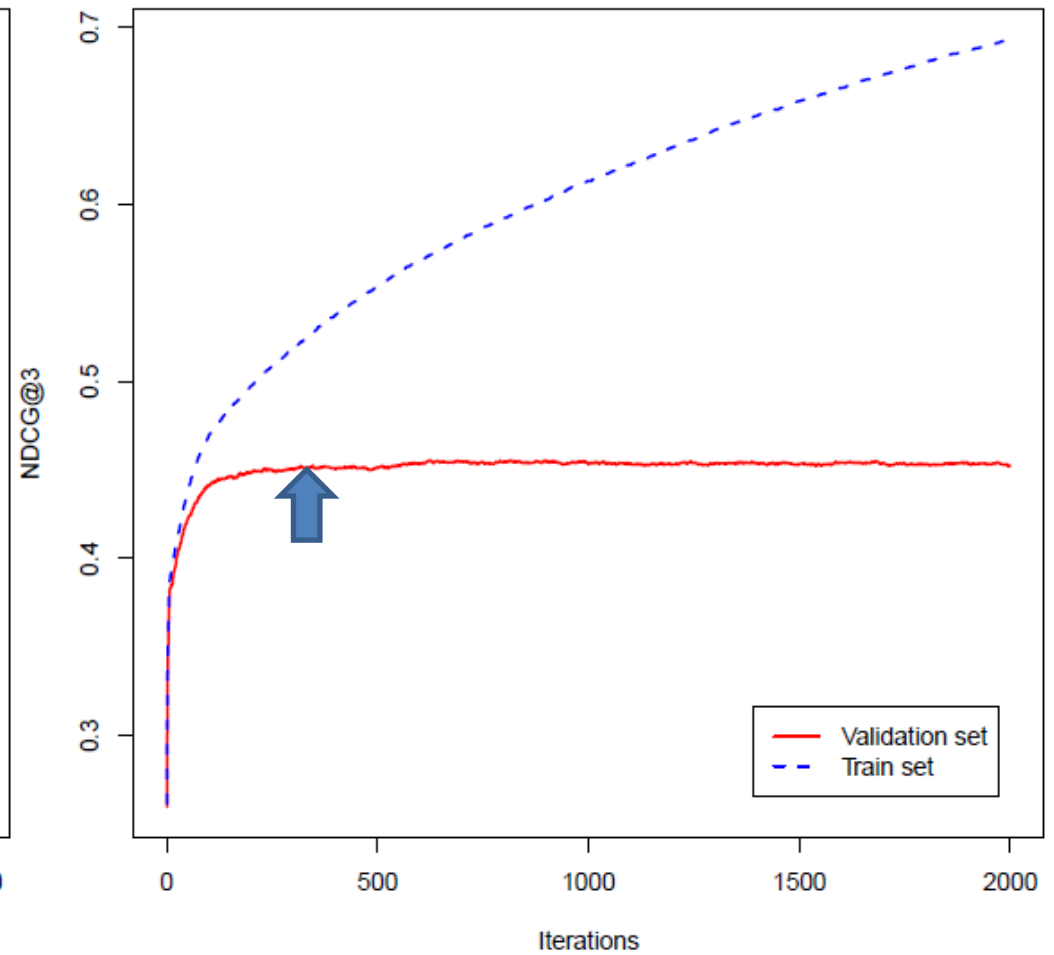
- Randomization:
 - Feature Sampling
 - Subsampling

Number of Trees

MQ2007 data set



MSLR-WEB10K data set



Parameter tuning for LambdaMART

(a) TD2004 and MQ2007 data sets

Parameter	Values
Max Number of Leaves	2, 4, 7, 10, 15, 20, 25
Min Percentage of Obs. per Leaf	0.12, 0.25, 0.50
Learning rate	0.05, 0.1, 0.2, 0.3
Sub-sampling rate	0.3, 0.5, 1.0
Feature Sampling rate	0.1, 0.3, 0.5, 1.0

(b) MSLR-WEB10K data set

Parameter	Values
Max Number of Leaves	10, 40, 70
Min Percentage of Obs. per Leaf	0.12, 0.25, 0.50
Learning rate	0.05, 0.1, 0.2
Sub-sampling rate	0.5, 1.0
Feature Sampling rate	0.3, 0.5, 1.0

$$1,008 \times 5 \times 3 = 15,120$$

$$162 \times 5 \times 3 = 2,430$$

Total Number of Experiments: $2 \times 15,120 + 2,430 = \mathbf{32,670}$

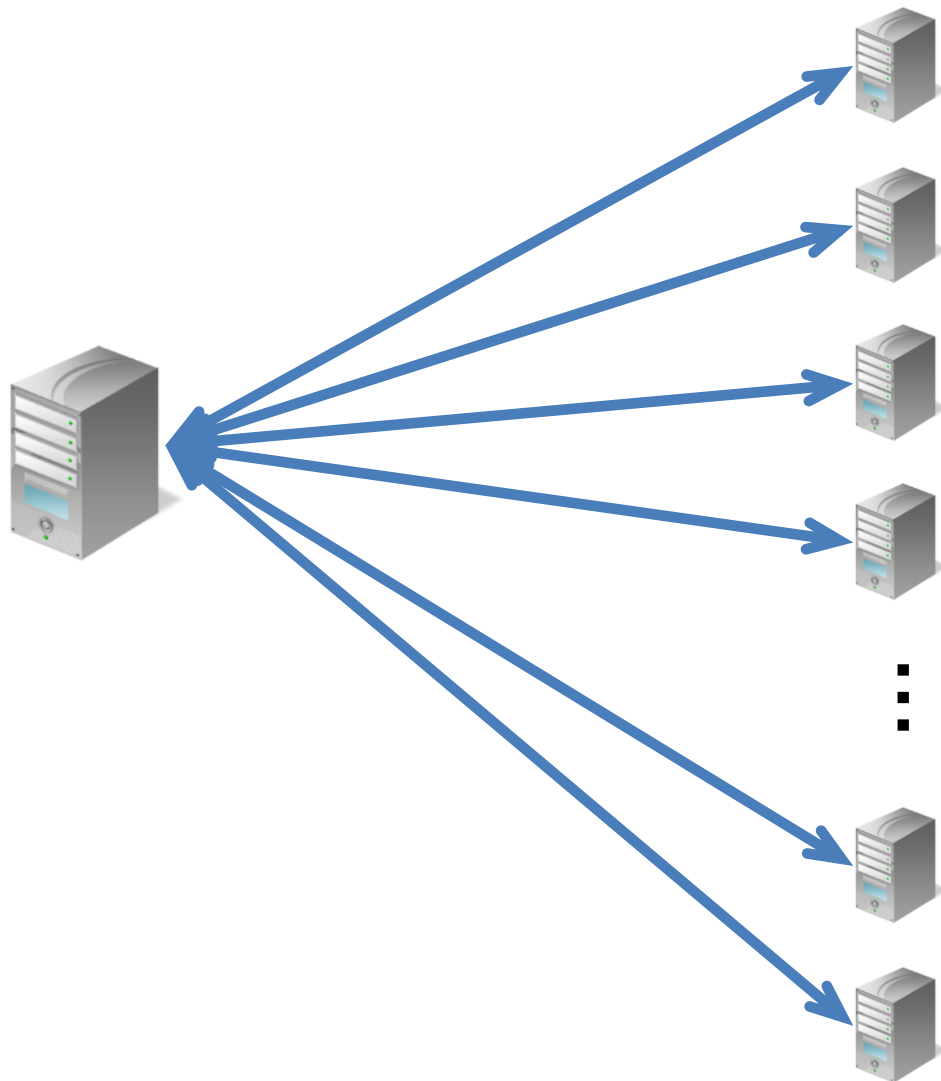
Trees: **8 million**

2 weeks for one run!

Splits: **192 million**

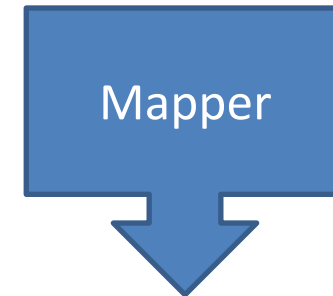
Splits/Threshold: **73 trillion**

Parameter Tuning on a MapReduce Cluster



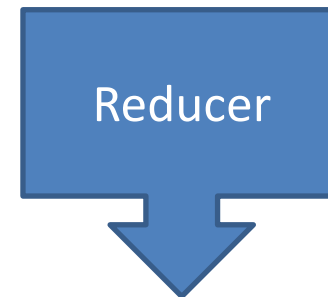
Config:1, Fold:4: Seed: 2
Config:1, Fold:4: Seed: 3

...



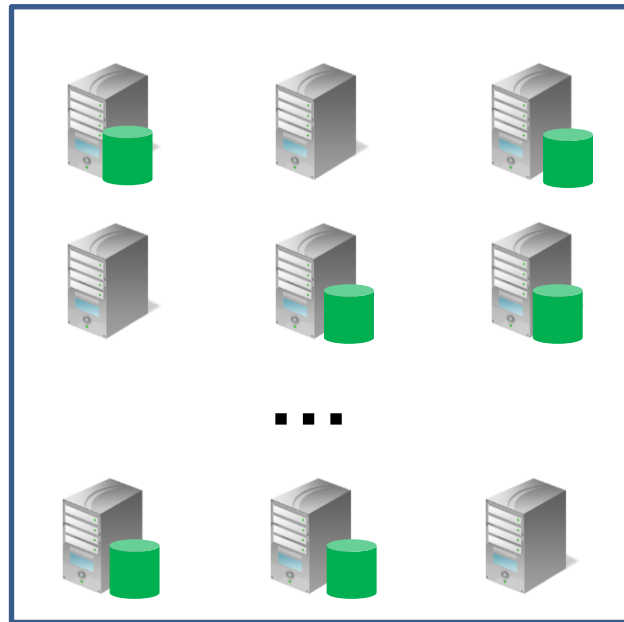
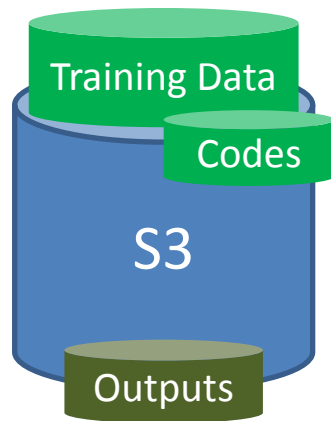
Config:1, Valid: 0.4567, Test: 0.4324
Config:1, Valid: 0.4534, Test: 0.4311

...



Config:1, Avg Valid: 0.4544, Avg Test: 0.4318

Amazon MapReduce Cluster



40 xlarge machines (15GB RAM)

Best Configs

(c) TD2004 data set

Validation NDCG@3	Max Leaves	Min Obs. Per Leaf	Learning Rate	Sub-sampling	Feature Sampling
0.5120	15	0.12	0.05	0.5	0.1
0.5110	15	0.12	0.1	0.5	0.5
0.5107	25	0.50	0.1	0.5	0.5
0.5082	15	0.12	0.05	0.5	0.1
0.5057	20	0.50	0.1	0.5	1.0

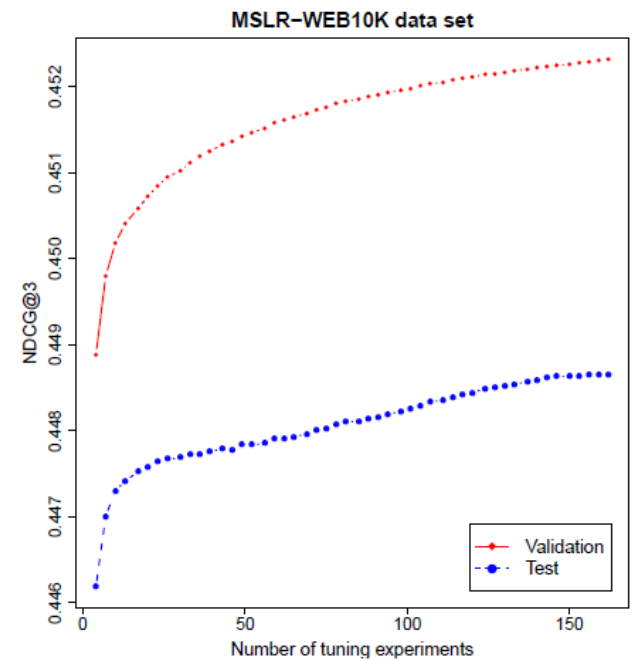
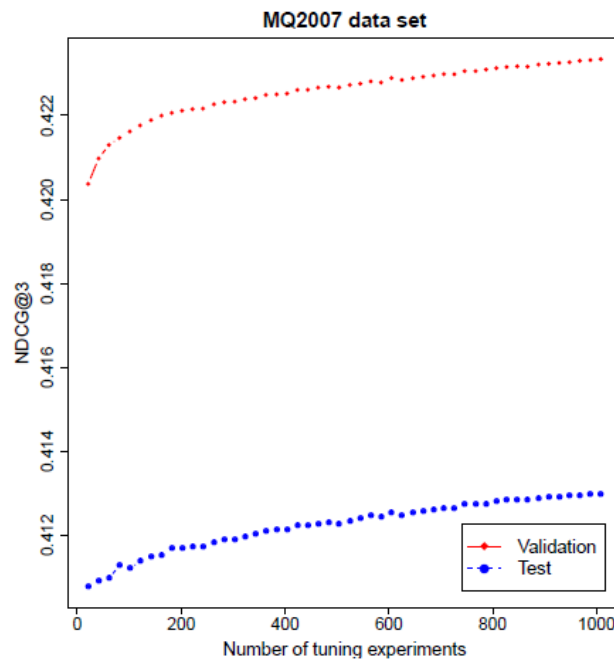
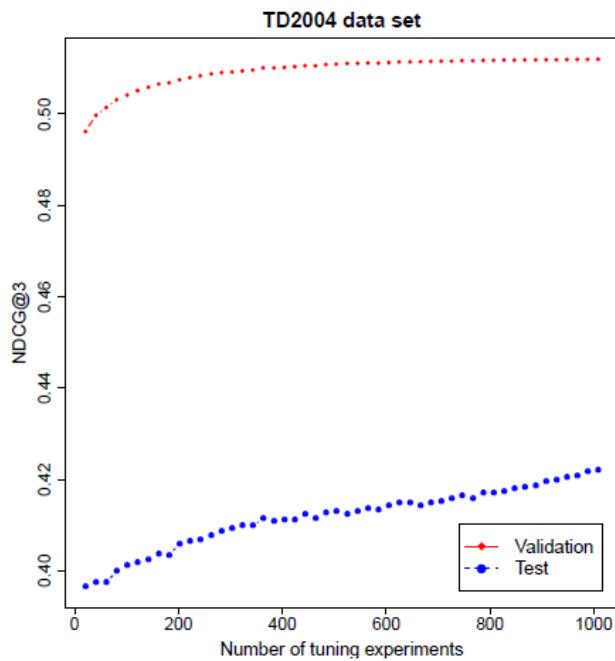
(d) MQ2007 data set

Validation NDCG@3	Max Leaves	Min Obs. Per Leaf	Learning Rate	Sub-sampling	Feature Sampling
0.4249	10	0.12	0.05	0.5	0.5
0.4246	20	0.5	0.05	0.3	0.5
0.4244	7	0.5	0.1	0.5	0.3
0.4242	4	0.5	0.1	1.0	0.1
0.4240	4	0.25	0.05	0.3	0.1

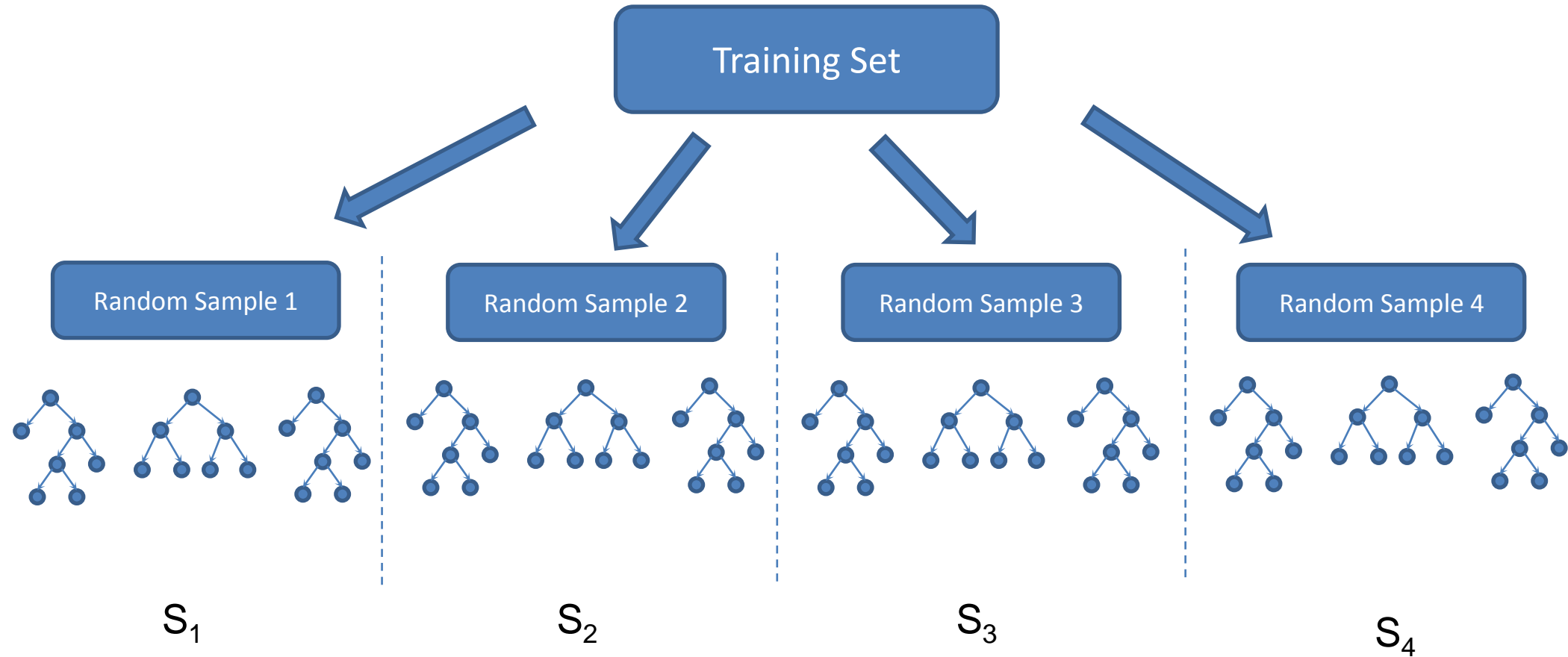
(e) MSLR-WEB10K data set

Validation NDCG@3	Max Leaves	Min Obs. Per Leaf	Learning Rate	Sub-sampling	Feature Sampling
0.4523	70	0.25	0.05	1.0	0.3
0.4516	70	0.25	0.05	1.0	0.5
0.4514	70	0.25	0.05	1.0	1.0
0.4508	40	0.25	0.05	1.0	0.3
0.4501	40	0.25	0.05	0.5	0.3

Did we need this many experiments?



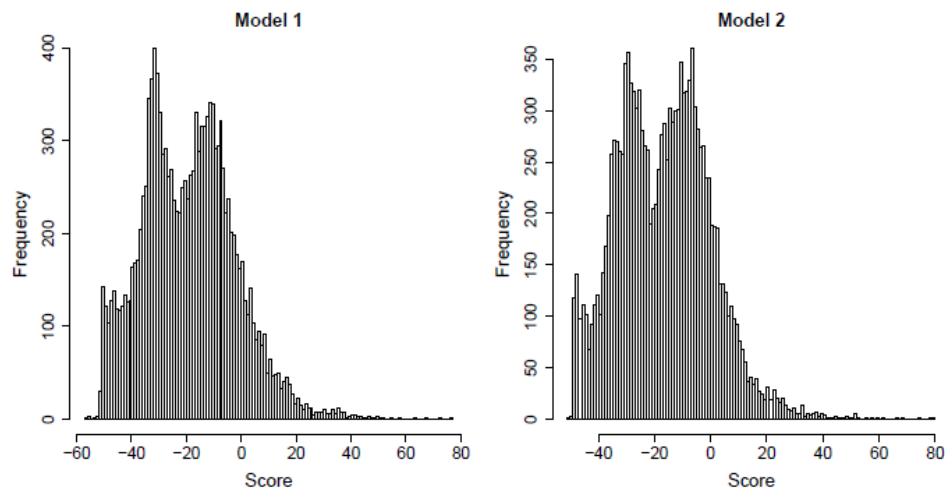
Bagging



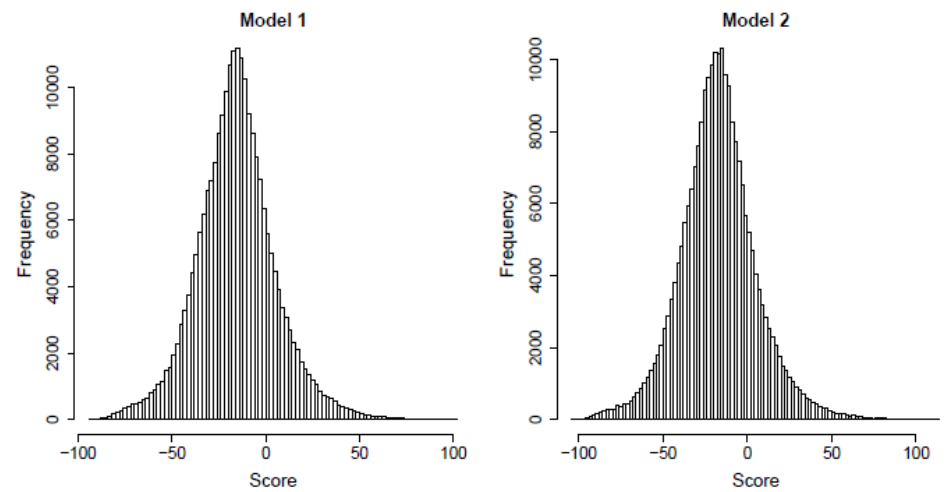
$$S = \frac{S_1 + S_2 + S_3 + S_4}{4}$$

Training time increase?

Distribution of Scores



(a) MQ2007 data set

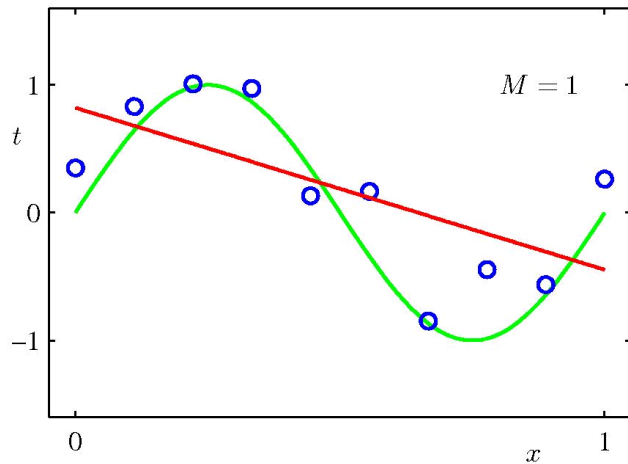


(b) MSLR-WEB10K data set

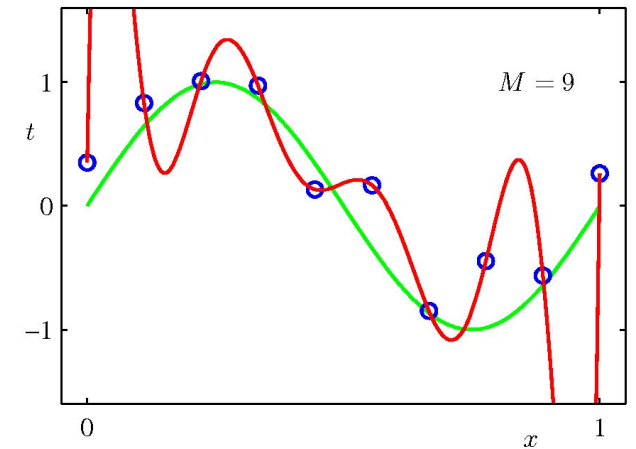
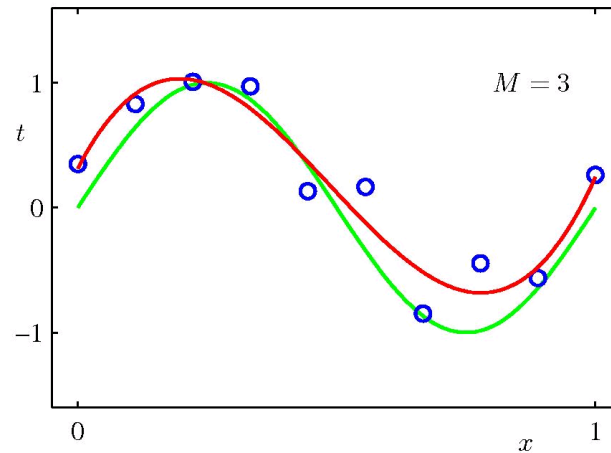
Can we also increase accuracy?

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

High Bias
Low Variance

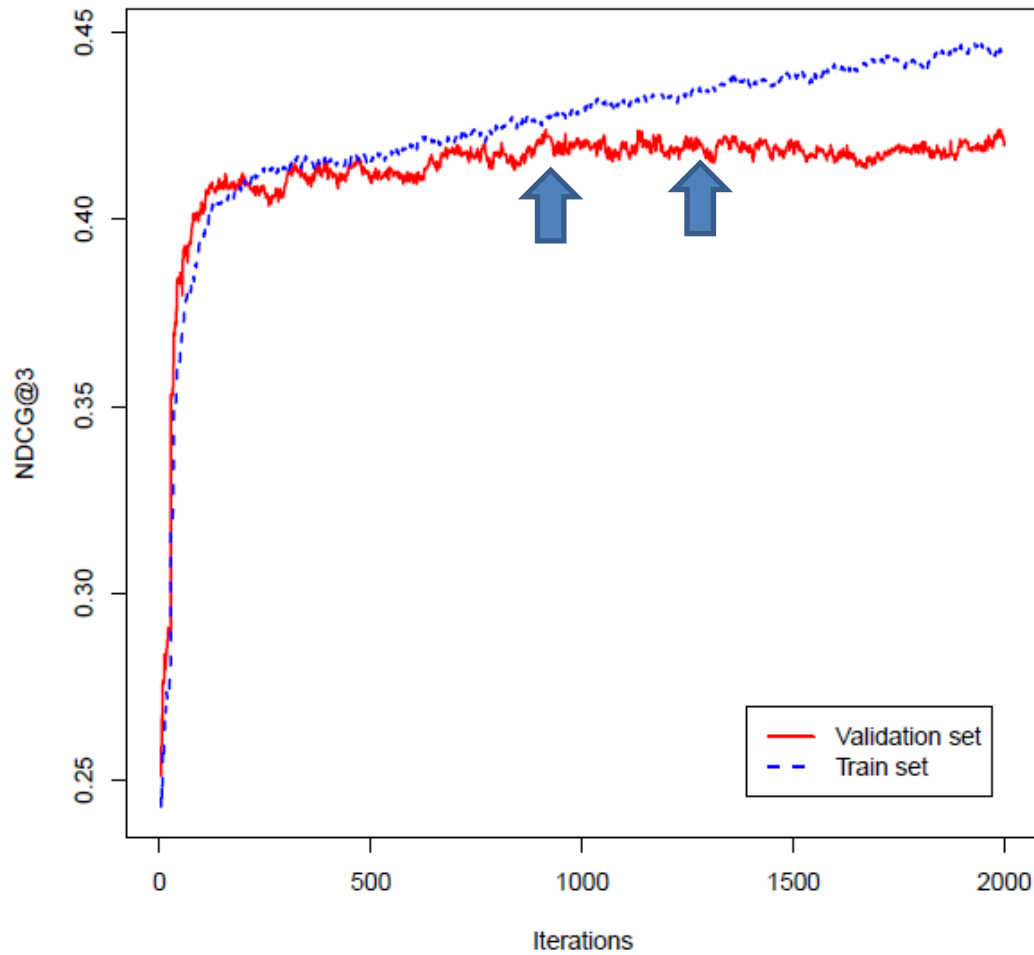


Low Bias
High Variance

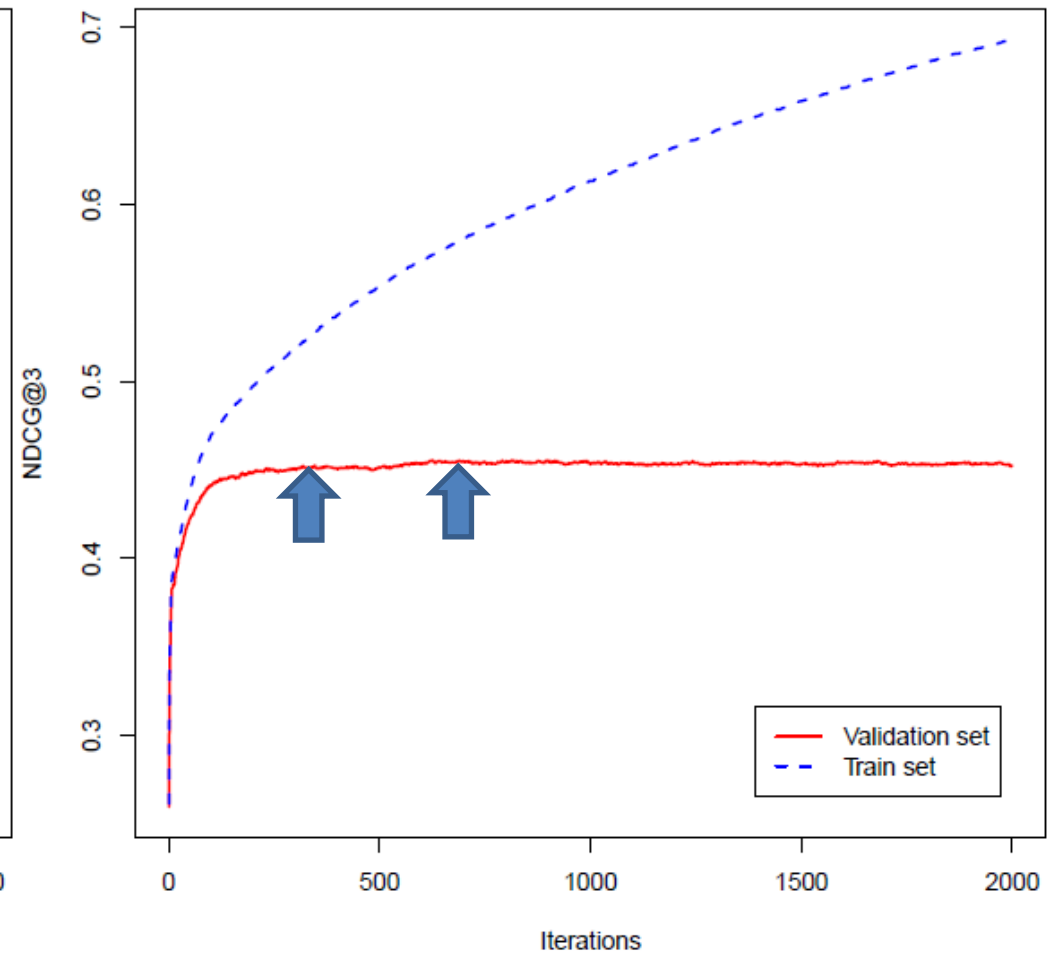


Overfitting Tolerance

MQ2007 data set

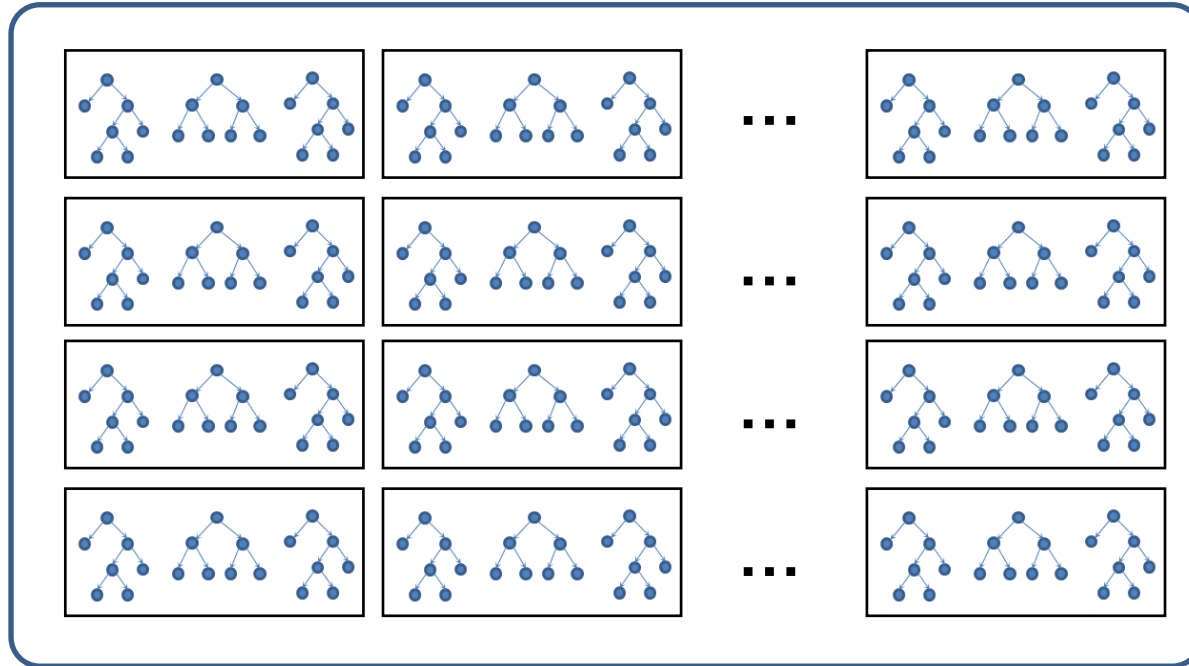


MSLR-WEB10K data set



Bagging on MapReduce Cluster

Model Pool



TD2004 and MQ2007

1000 x 5 x 2

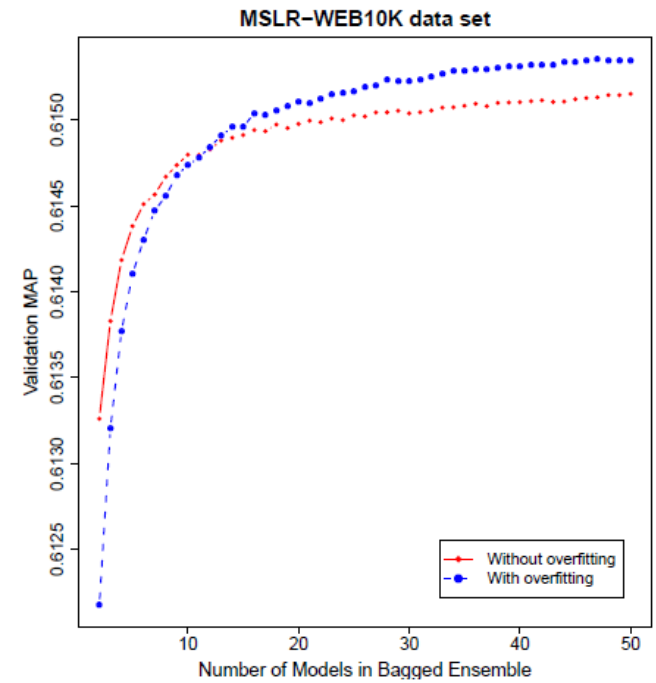
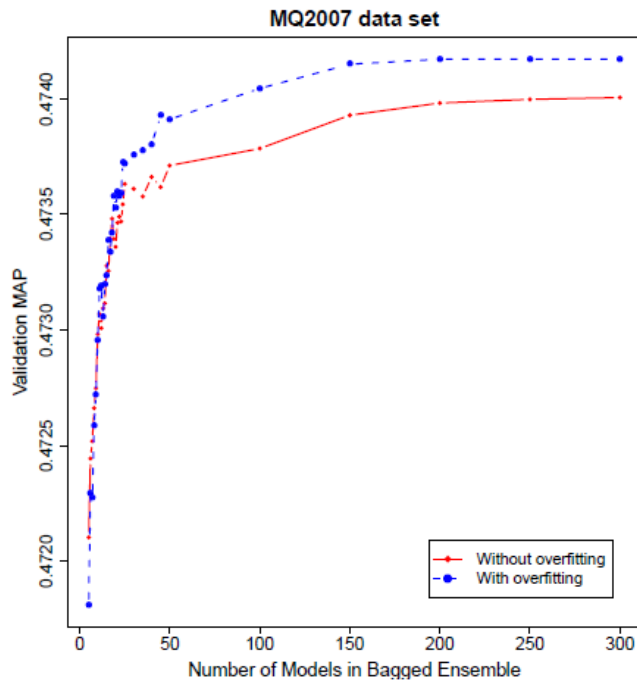
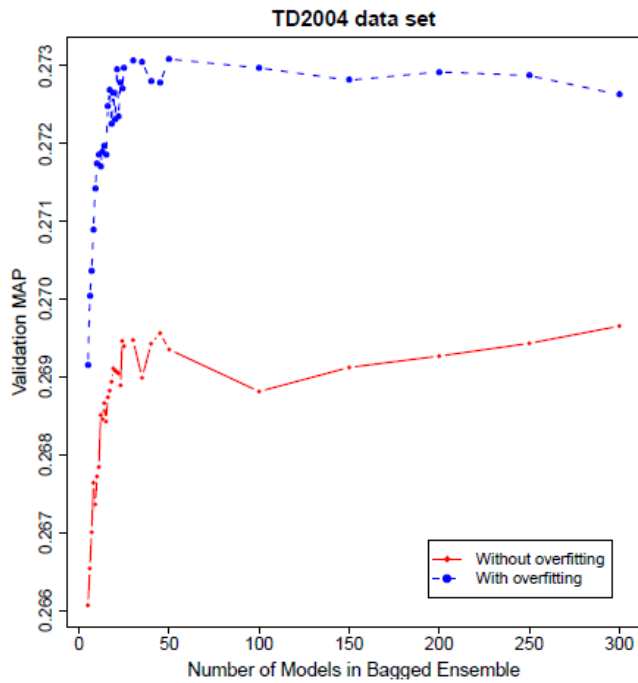
MSLR-WEB10K

50 x 5 x 2

20,500 Models need to be trained and evaluated.

18 hours on Amazon cluster

Overfitting tolerance



Evaluation Results: TD2004

	NDCG@1	NDCG@3	NDCG@5	MAP
SVMmap [34]	0.2933	0.3035	0.3007	0.2049
RankSVM-Struct [20]	0.3467	0.3371	0.3192	0.2196
ListNet [9]	0.3600	0.3573	0.3325	0.2231
SmoothRank [11]	0.4000	0.3832	0.3555	0.2326
RankSVM [17]	0.4133	0.3467	0.3240	0.2237
AdaRank-MAP [33]	0.4133	0.3757	0.3602	0.2189
AdaRank-NDCG [33]	0.4267	0.3688	0.3514	0.1936
BoltzRank [30]	0.4767	0.3902	0.3635	0.2390
FRank [28]	0.4933	0.3875	0.3629	0.2388
RankBoost [13]	0.5067	0.4295	0.3878	0.2614
BagBoo [23]	0.5067	0.4080	0.3898	0.2499
LambdaMART	0.4267	0.3584	0.3266	0.2378
LambdaMART with randomization	0.4560	0.4033	0.3722	0.2513
BL-MART without overfitting	0.4947	0.4217	0.3886	0.2649
BL-MART with overfitting	0.4947	0.4270	0.3948	0.2684

Evaluation Results: MQ2007

	NDCG@1	NDCG@3	Mean NDCG	MAP
RankSVM-Struct [20]	0.4096	0.4063	0.4966	0.4645
ListNet [9]	0.4002	0.4091	0.4988	0.4652
AdaRank-MAP [33]	0.3821	0.3984	0.4891	0.4577
AdaRank-NDCG [33]	0.3876	0.4044	0.4914	0.4602
RankBoost [13]	0.4134	0.4072	0.5003	0.4662
CRR [25]	–	–	0.5000	0.4660
BagBoo [23]	0.4071	0.4176	–	0.4676
LambdaMART	0.4147	0.4119	0.5011	0.4660
LambdaMART with randomization	0.4137	0.4157	0.5035	0.4684
BL-MART without overfitting	0.4197	0.4217	0.5079	0.4726
BL-MART with overfitting	0.4200	0.4224	0.5093	0.4731

Evaluation Results: MSLR-WEB10K

	NDCG@1	NDCG@3	Mean NDCG	MAP
LambdaMART	0.4580	0.4467	0.5693	0.3670
LambdaMART with randomization	0.4628	0.4487	0.5706	0.3684
BL-MART without overfitting	0.4640	0.4514	0.5720	0.3696
BL-MART with overfitting	0.4642	0.4516	0.5729	0.3705

Variance Reduction

10 random samples from Fold1 of MSLR-WEB10K

	NDCG@1		NDCG@3		Mean NDCG		MAP	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
LambdaMART	0.4484	18×10^{-6}	0.4395	9.1×10^{-6}	0.5640	1.2×10^{-6}	0.3657	0.9×10^{-6}
LambdaMART with randomization	0.4492	22×10^{-6}	0.4421	5.4×10^{-6}	0.5647	1.4×10^{-6}	0.3665	1.3×10^{-6}
BL-MART without overfitting	0.4516	10×10^{-6}	0.4468	7.8×10^{-6}	0.5675	1.5×10^{-6}	0.3690	0.9×10^{-6}
BL-MART with overfitting	0.4528	7×10^{-6}	0.4471	4.4×10^{-6}	0.5686	0.8×10^{-6}	0.3703	0.5×10^{-6}

- NDCG@1: -67.3%
- NDCG@3: -18.8%
- Mean NDCG: -40.2%
- MAP: -57.1%

Thank You