Microsoft Research

Microsoft

Machine Learning for Information Retrieval (ML4IR)

Katja Hofmann

Researcher at Microsoft Research, Cambridge
Twitter: @katjahofmann

Presented at the European Summer School in Information Retrieval (ESSIR) 2015
Thessaloniki, Greece, September 3, 2015

Machine Learning is one of the core tools of IR

in research …

Search Results

Results 1 - 50 of 92

Sort by relevance in condensed form

Result page: **1** 2 next >>

1 Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks
Aliaksei Severyn, Alessandro Moschitti
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

2 LBMCH: Learning Bridging Mapping for Cross-modal Hashing
Yang Wang, Xuemin Lin, Lin Wu, Wenjie Zhang, Qing Zhang
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

3 Learning to Reweight Terms with Distributed Representations
Guoqing Zheng, Jamie Callan
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

4 Predicting Search Intent Based on Pre-Search Context
Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, James Allan
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

5 Semi-supervised Hashing with Semantic Confidence for Large Scale Visual Search
Yingwei Pan, Ting Yao, Houqiang Li, Chong-Wah Ngo, Tao Mei
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

6 Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings
Ivan Vulić, Marie-Francine Moens
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

7 Learning Context-aware Latent Representations for Context-aware Collaborative Filtering
Xin Liu, Wei Wu
August 2015 **SIGIR '15:** Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval

Learning to rank

Learning / mapping document representations

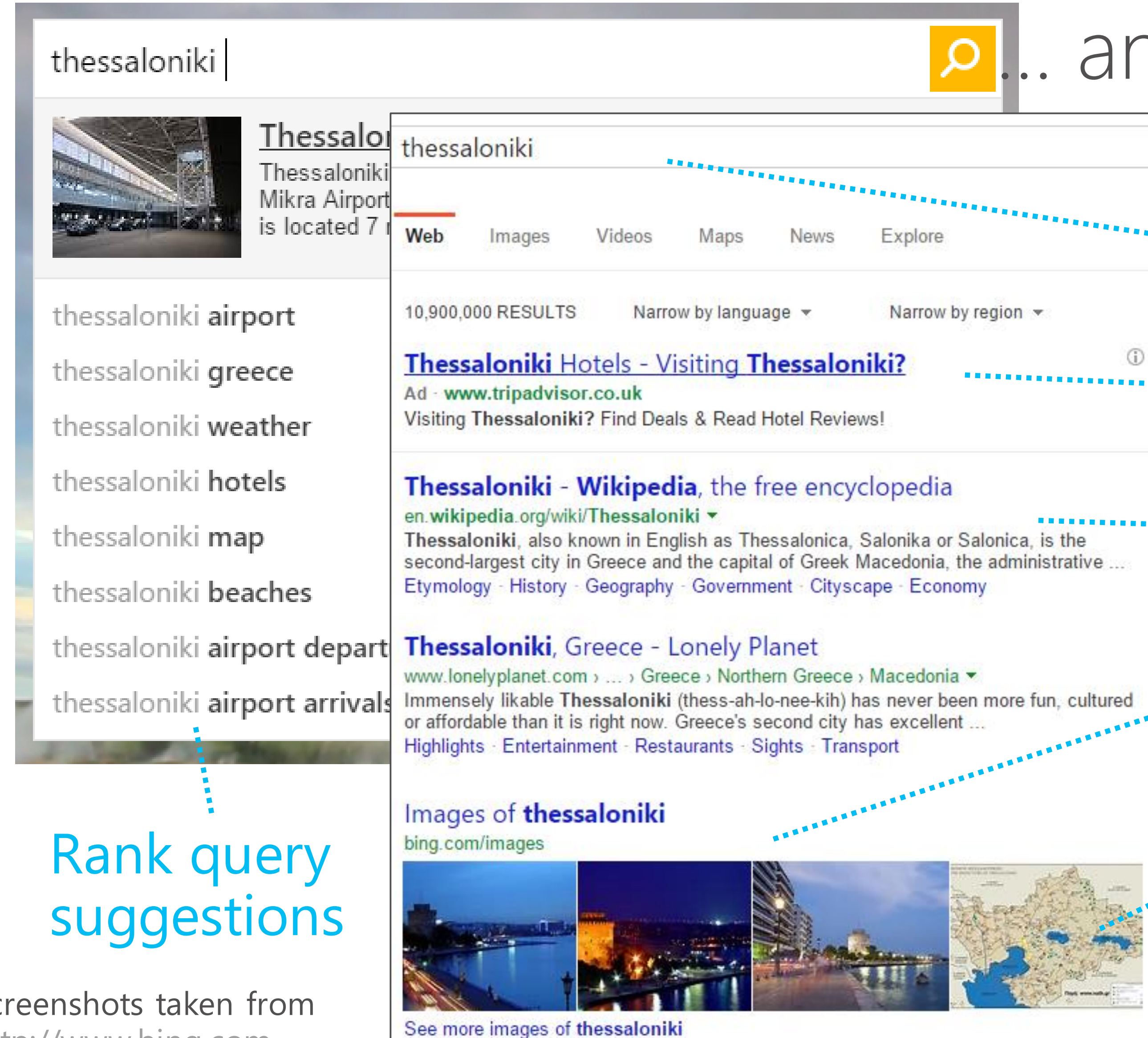
Predicting intent

Recommendation

Figure: top search results for "learn" in the SIGIR 2015 conference proceedings, from http://dl.acm.org

Machine Learning is one of the core tools of IR

… and in practice



Detect and classify named entities

Match ads

Rank search results

Predict query intent

Rank verticals and their content

…

Rank query suggestions

Screenshots taken from http://www.bing.com

Outline

Past

Text classification and
foundations of machine learning

Present

From learning to rank to
learning from user interactions

Future

Trends and directions

Past

Text classification and foundations of machine learning

ML automated text classification

Initially: manual classification to assign documents / books to subject categories

Figure: library of the Rijksmuseum in Amsterdam

Early work on news classification

Classifying News Stories using Memory Based Reasoning

Brij Masand, Gordon Linoff, David Waltz*

Thinking Machines Corporation

245 First Street, Cambridge, Massachusetts, 02142 USA

1 Abstract

We describe a method for classifying news stories using Memory Based Reasoning (MBR) (a *k*-nearest neighbor method), that does not require manual topic definitions. Using an already coded training database of about 50,000 stories from the Dow Jones Press Release News Wire, and SEEKER [Stanfill] (a text retrieval system that supports relevance feedback) as the underlying match engine, codes are assigned to new, unseen stories with a recall of about 80% and precision of about 70%. There are about 350 different codes to be assigned. Using a massively parallel supercomputer, we leverage the information already contained in the thousands of coded stories and are able to code a story in about 2 seconds.[1] Given SEEKER, the text retrieval system, we achieved these results in about two person-months. We believe this approach is effective in reducing the development time to implement classification systems involving

and/or specific permission.
...n Int'l SIGIR '92/Denmark-6/92
ACM 0-89791-524-0/92/0006/0059...$1.50

tasks such as extraction of relational information from text [Young] [Jacobs].

Alternative systems [Biebricher] [Lewis] use statistical approaches such as conditional probabilities on summary representations of the documents. One problem with statistical representations of the training database is the high dimensionality of the training space, generally at least 150k unique single features -- or words. Such a large feature space makes it difficult to compute probabilities involving conjunctions or co-occurrence of features. It also makes the application of neural networks a daunting task. We describe a new approach for classifying news stories using their full text that achieves high recall and at least moderate precision without requiring manual definitions of the various topics, as required by most of the earlier approaches.

Section 3 describes the problem; Section 4, the main results and Section 5 reviews MBR. The classification algorithm and variations of parameters are described in Sections 7 - 9 and we conclude with a discussion of results and future

[Masand et al. '92]

Approach and results

FIGURE 2 Sample News Story and Codes

0023000PR PR 910820
I/AUT I/CPR I/ELQ M/CYC M/IDU M/TEC
R/EU R/FE R/GE R/JA R/MI R/PRM R/TX R/WEU

Suggested Codes: = system suggested

* = editor assigned

* 3991	R/FE	Far East
* 3991	M/IDU	Industrial
* 3991	I/ELQ	Electrical Components & Equipment
* 3067	R/JA	Japan
* 2813	M/TEC	Technology
* 2813	M/CYC	Consumer, Cyclical
* 2813	I/CPR	Computers
* 2813	I/AUT	Automobile Manufacturers
2460	P/MCR	Mainframes
1555	R/CA	California
1495	M/UTI	Utilities
*1285	R/MI	Michigan
*1178	R/PRM	Pacific Rim
*1175	R/EU	Europe

"DAIMLER-BENZ UNIT SIGNS $11,000,000 AGREE-MENT FOR HITATCHI DATA SYSTEMS DISK DRIVES"

SANTA CLARA, Calif.--(BUSINESS WIRE)--Debis Sys-temhaus GmbH, a 100 percent subsidiary of Daim-
...gned a contract to purchase

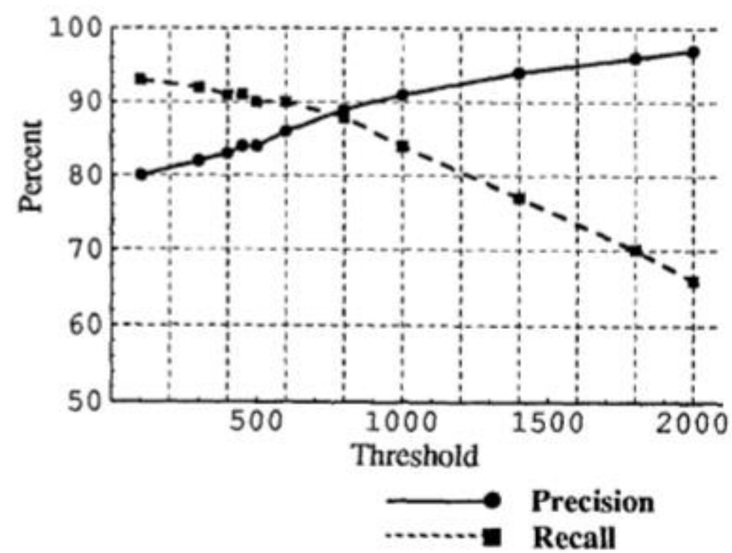
[Masand et al. '92]

Approach: k-nearest neighbor (knn)

Requires document representation (here: single words – stop / frequent words + "capital pairs"), distance metric (here: Euclidean distance), choice of k

Results: > 80 % precision / recall,
huge increase in efficiency (months vs. years)

FIGURE 5 Variation by Threshold for Industry Codes, *k* = 10



An alternative approach

Training Algorithms for Linear Text Classifiers

David D. Lewis Robert E. Schapire
AT&T Laboratories
Murray Hill, NJ 07974 USA
{*lewis, schapire*}*@research.att.com*

James P. Callan Ron Papka
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
{*callan, papka*}*@cs.umass.edu*

Abstract

Systems for text retrieval, routing, categorization and other IR tasks rely heavily on linear classifiers. We propose that two machine learning algorithms, the Widrow-Hoff and EG algorithms, be used in training linear text classifiers. In contrast to most IR methods, theoretical analysis provides performance guarantees and guidance on parameter settings for these algorithms. Experimental data is presented showing Widrow-Hoff and EG to be more effective than the widely used Rocchio algorithm on several categorization and routing tasks.

are better understood from a theoretical standpoint, leading to performance guarantees and guidance on parameter settings. In addition, we show experimentally that Widrow-Hoff and EG are more effective than Rocchio on both routing and categorization tasks.

2 Linear Functions in IR

IR systems often represent texts as *feature vectors*, that is, tuples of values:

$$\mathbf{x} = (x_1, x_2, \ldots, x_d)$$

[Lewis et al. '96]

Approach

Linear, model-based learning

Given training instances of the form $\boldsymbol{x} = (x_1, x_2, \dots x_d)$, learn a linear model:

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} = \sum_{j=1}^{d} w_j x_j$$

Training algorithm uses stochastic gradient descent, using the update rule:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - 2\eta(\boldsymbol{w}_i \cdot \boldsymbol{x}_i - y_i)\boldsymbol{x}_i$$

(*stochastic* because a single sample $\boldsymbol{x}_i$ is used in each update step i;
gradient descent because $2(\boldsymbol{w}_i \cdot \boldsymbol{x}_i - y_i)\boldsymbol{x}_i$ is the gradient $\partial L(\boldsymbol{w})/\partial \boldsymbol{w}_i$;
L is a *loss* or *cost function*, here: $L(\boldsymbol{w}) = \frac{\mathbf{1}}{\mathbf{2}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}}(w_i \cdot x_i - y_i)^{\mathbf{2}}$; η = *learning rate*)

Called "least mean squares" or Widrow-Hoff (WH) learning rule. Compared to Rocchio and exponentiated gradient (EG) learning rules

[Lewis et al. '96]

Results

Data Set: Training	DocRep	Num Classes	Num Features	WH vs. Rocchio: Wins	WH vs. Rocchio: Mean F_1	EG vs. Rocchio: Wins	EG vs. Rocchio: Mean F_1
AP Headline Categorization							
10000	bin	20	40820	16 > 4	(.45 > .33)	16 > 4	(.44 > .33)
10000	*tf* × *idf*	20	40820	14 >? 6	(.48 > .44)	[10 =? 10	(.44 > .44)]
142791	bin	20	40820	15 > 5	(.57 > .40)	18 > 2	(.55 > .40)
142791	*tf* × *idf*	20	40820	13 >? 7	(.58 > .52)	[14 >? 6	(.55 > .52)]
AP Body Categorization							
10000	bin	20	264836	18 > 2	(.48 > .25)	18 > 2	(.52 > .25)
10000	*tf* × *idf*	20	264836	15 > 5	(.60 > .50)	[10 =? 10	(.52 > .50)]
142791	bin	20	264836	18 > 2	(.65 > .33)	18 > 2	(.61 > .33)
142791	*tf* × *idf*	20	264836	16 > 4	(.72 > .63)	[9 <? 11	(.61 < .63)]
OHSUMED Title Categorization (big categories)							
10000	bin	49	64781	48 > 1	(.29 > .15)	47 > 2	(.29 > .15)
10000	*tf* × *idf*	49	64781	41 > 7	(.34 > .29)	[30 >? 19	(.29 > .29)]
183229	bin	49	64781	47 > 2	(.53 > .26)	48 > 1	(.51 > .26)
183299	*tf* × *idf*	49	64781	32 > 17	(.51 > .47)	[35 > 14	(.51 > .47)]
OHSUMED Title Categorization (small categories)							
10000	bin	28	64781	15 > 4	(.04 > .02)	23 > 3	(.03 > .02)
10000	*tf* × *idf*	28	64781	21 > 1	(.06 > .03)	[20 > 7	(.03 > .03)]
183229	bin	28	64781	26 > 0	(.43 > .22)	27 > 0	(.46 > .42)
183299	*tf* × *idf*	28	64781	16 > 10	(.43 > .41)	[21 > 4	(.46 > .41)]
OHSUMED Abstract Categorization (big categories)							
10000	bin	49	135531	45 > 4	(.16 > .07)	49 > 0	(.27 > .07)
10000	*tf* × *idf*	49	135531	45 > 4	(.28 > .18)	[43 > 6	(.27 > .18)]
183229	bin	49	135531	49 > 0	(.51 > .13)	48 > 1	(.50 > .13)
183299	*tf* × *idf*	49	135531	44 > 5	(.55 > .44)	[34 > 15	(.50 > .44)]
OHSUMED Abstract Categorization (small categories)							
10000	bin	28	135531	13 > 1	(.01 > .00)	24 > 1	(.02 > .00)
10000	*tf* × *idf*	28	135531	11 > 1	(.03 > .00)	[24 > 1	(.02 > .00)]
183229	bin	28	135531	22 > 3	(.29 > .10)	26 > 0	(.39 > .10)
183299	*tf* × *idf*	28	135531	15 > 12	(.39 > .33)	[15 > 12	(.39 > .33)]
TREC Document Routing (topics 51-100)							
varies	INQUERY	50	Q+50	34 > 16	(.28 > .22)	42 > 8	(.36 > .22)
varies	INQUERY	50	Q+1000	41 > 9	(.16 > .06)	49 > 1	(.36 > .06)
TREC Document Routing (topics 101-150)							
varies	INQUERY	50	Q+50	13 <? 37	(.23 < .29)	38 > 11	(.35 > .29)
varies	INQUERY	50	Q+1000	36 > 14	(.13 > .07)	48 > 2	(.19 > .07)

Table 4: Pairwise comparisons of WH vs. Rocchio, and EG vs. Rocchio. For each condition we show data set, training set size, document representation, number of classes, and number of features. Wins shows the number of classes for which each algorithm had a higher F_1 value. Rocchio is significantly worse by one-tailed sign test unless a "?" is shown. Mean F_1 is the mean value of F_1 across all classes for each algorithm. Results for EG vs. Rocchio on a *tf* × *idf* representation are bracketed "[]" to indicate that EG was actually run on a binary representation.

Gradient updates for WH and EG theoretically well-motivated (provide performance guarantees)

Outperform Rocchio (relevance feedback) empirically

[Lewis et al. '96]

Comparing linear model and knn

Linear model: strong assumptions on global structure

FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then fit by linear regression. The line is the decision boundary defined by $x^T\hat{\beta} = 0.5$. The orange shaded region denotes that part of input space classified as* ORANGE*, while the blue region is classified as* BLUE.

Image from [Hastie et al. '09]

… but we can fit higher-order polynomials:

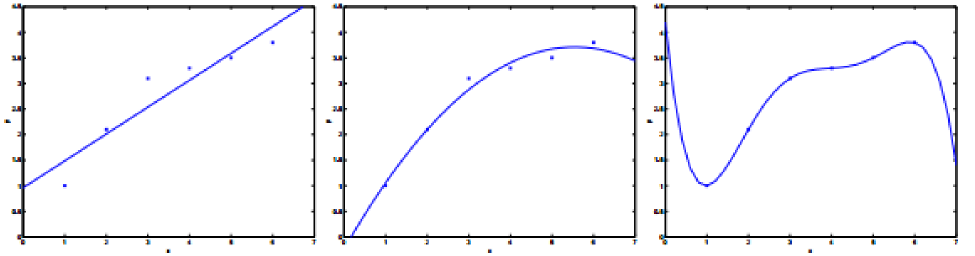


Image from [Ng '15]

Comparing linear model and knn

knn: uses only local structure

… larger k smooth decision boundaries:

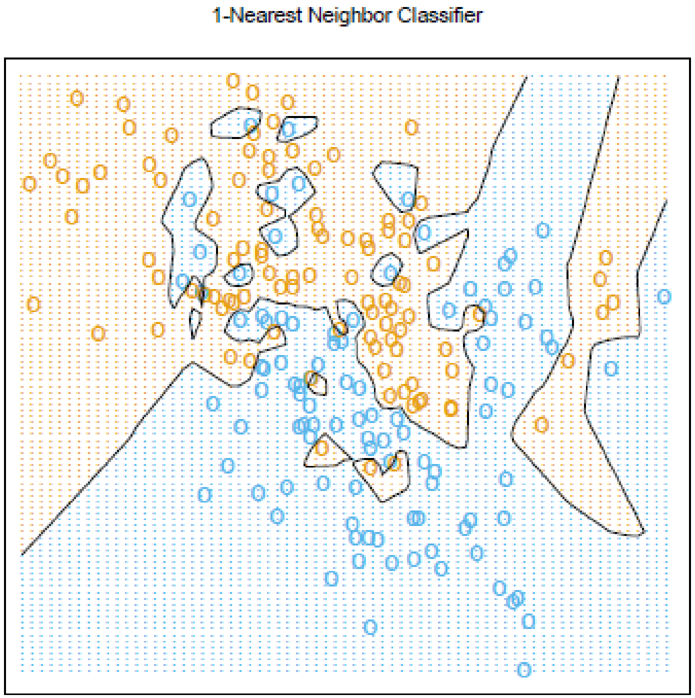


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$), *and then predicted by 1-nearest-neighbor classification.*

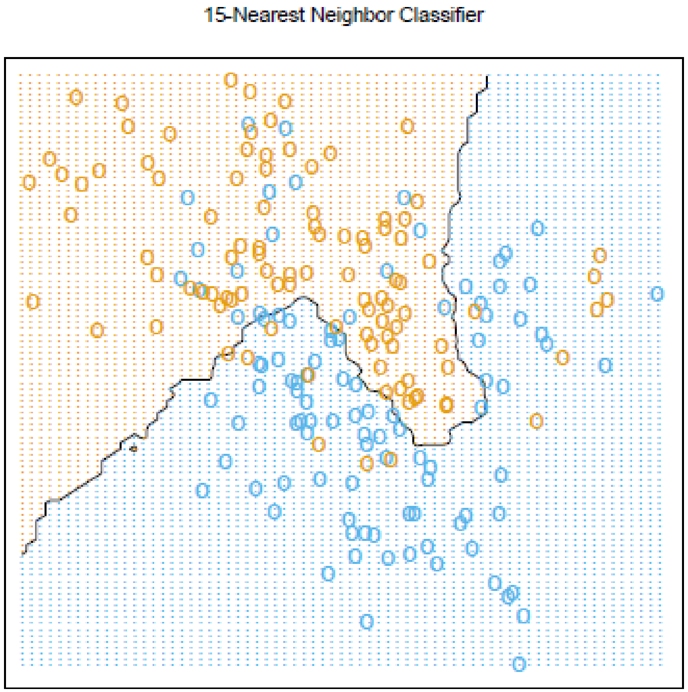


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.*

Images from [Hastie et al. ’09]

How to select the right model?

Goal: generalize to unseen samples

So we should pick a model with low **generalization error** (defined for a given model h):

$$\varepsilon(h) = P_{(x,y)\sim D}(h(x) \neq y)$$

However, training (e.g., empirical risk minimization) minimizes the **training error**:

$$\hat{\varepsilon}(h) = \frac{1}{m}\sum_{i=1}^{m} 1\{h(x_i) \neq y_i\}$$

Learning theory relates training and generalization error, e.g.,:

$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2e^{-2\gamma^2 m}$ (γ is a fixed threshold, m the sample size)

Empirically, the generalization error is estimated using cross validation.

[Ng ‘15]

How to select the right model?

Goal: generalize to unseen samples

So we should pick a model with low **generalization error** (defined for a given model h):

$$\varepsilon(h) = P_{(x,y)\sim D}(h(x) \neq y)$$

Key assumption: training and test data have the same distribution, samples are independently and identically distributed.

However, training (e.g., empi

training error:

$$\hat{\varepsilon}(h) =$$

Learning theory relates train

$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2e^{-2\gamma^2 m}$ (γ is a fixed threshold, m the sample size)

Empirically, the generalization error is estimated using cross validation.

[Ng '15]

Summary

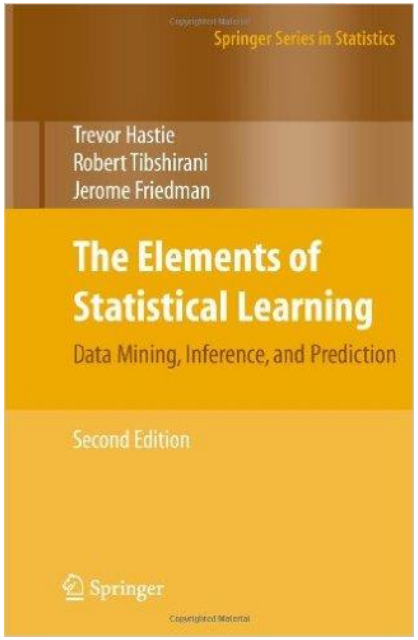
1st main application of ML to IR: automatic text classification

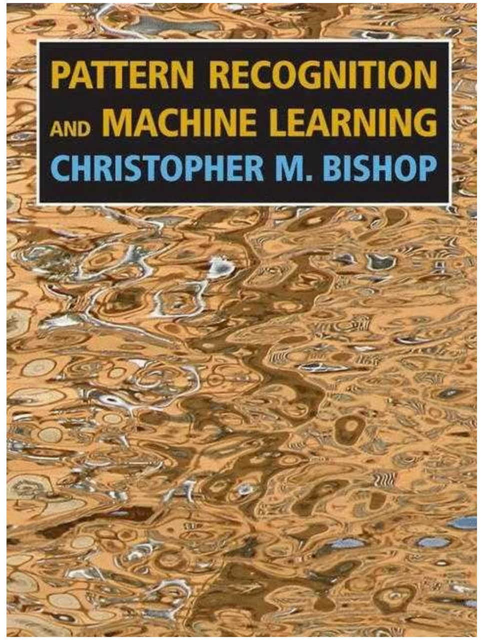
Supervised ML:

- Model-free (knn) and model-based (linear) approaches
- Empirical and generalization error – assumptions and key to generalization

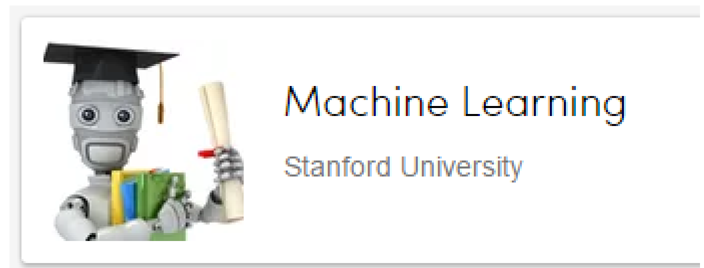
Today: wide range of (supervised) ML approaches are standard tools for IR

Further reading





See [Hastie et al. '09] for a (mostly) frequentist perspective, [Bishop '07] for a (mostly) Bayesian perspective.



On coursera: https://www.coursera.org/learn/machine-learning, also see the related course on http://cs229.stanford.edu

See: http://scikit-learn.org

Present

From learning to rank to learning from user interactions

Part 1: learning to rank

Motivating learning to rank

INCORPORATING DIFFERENT SEARCH MODELS
INTO ONE DOCUMENT RETRIEVAL SYSTEM

W. Bruce Croft
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

Abstract

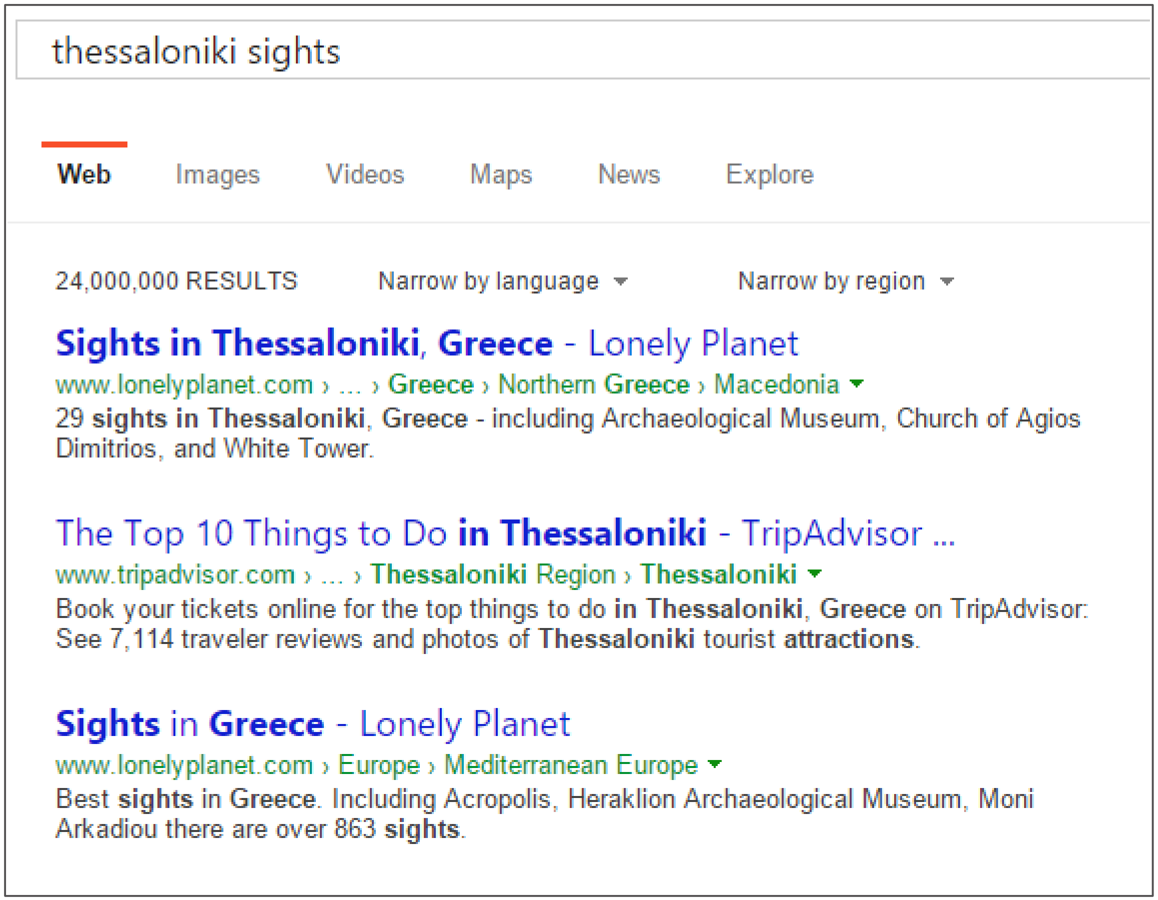
Many effective search strategies derived from different models are available for document retrieval systems. However, it does not appear that there is a single most effective strategy. Instead, different strategies perform optimally under different conditions. This paper outlines the design of an adaptive document retrieval system that chooses the best search strategy for a particular situation and user. In order to be able to support a variety of search strategies, a general network representation of the documents and terms in the database is proposed. This network representation leads to efficient methods of generating and using document and term classifications.

One of the most desirable features of an adaptive system would be the ability to learn from experience. A method of incorporating this learning ability into the system is described. The adaptive control strategy for choosing search strategies enables the system to base its actions on a number of factors, including a model of the current user.

Finally, some ideas for a flexible interface for casual users are suggested. Part of this interface is the heuristic search, which is used when searches based on formal models have failed. The heuristic search provides a browsing capability for the user.

[Croft '81]

Learning to rank's success story in web search

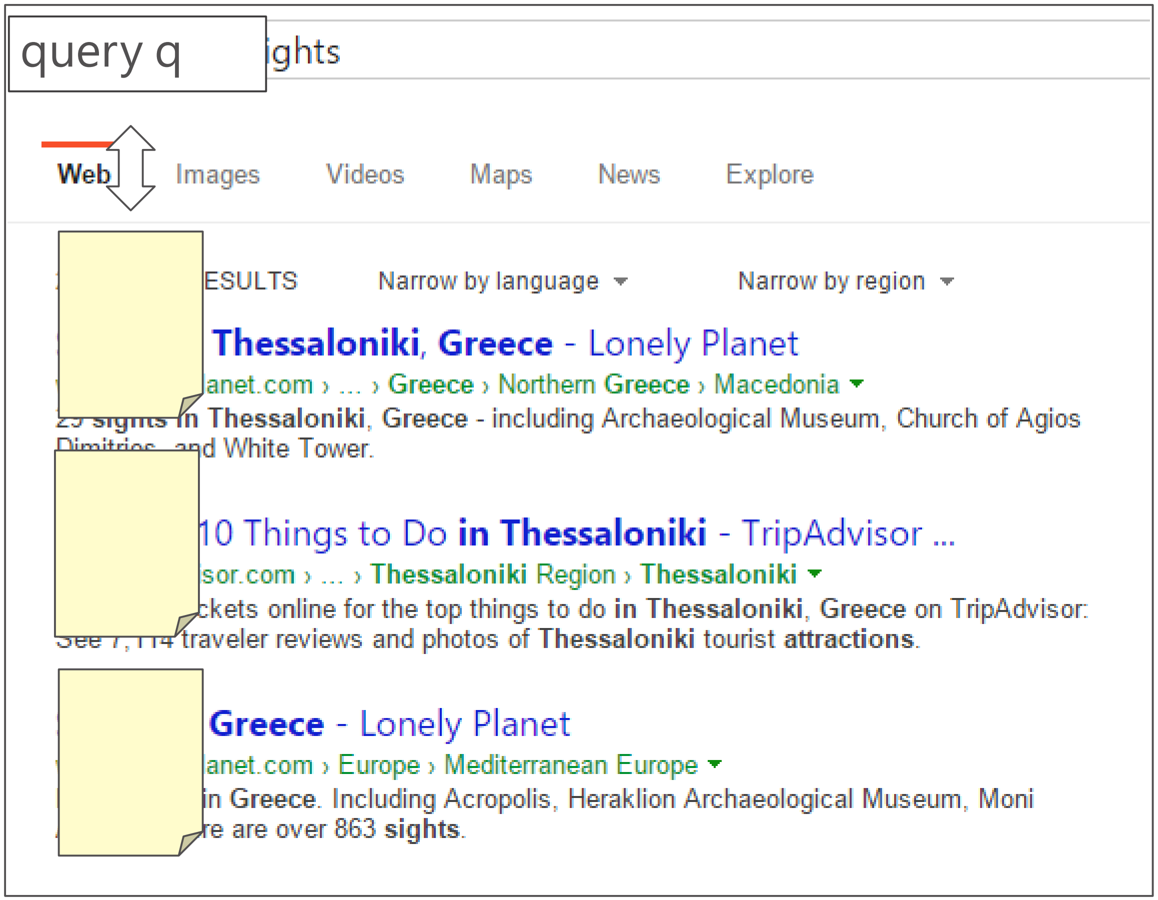


Hundreds of ranking signals: BM25, topic models, title / body / heading matches, phrase models, PageRank, HITS, date of publication / recent update, author, location …

Different combinations for different types of queries: head / body / tail, spiking queries, queries related to news, artists, TV shows, science, homepage finding (navigation), long-term information needs …

Impossible to tune models by hand for each query type – ML to the rescue …

Learning to rank's success story in web search

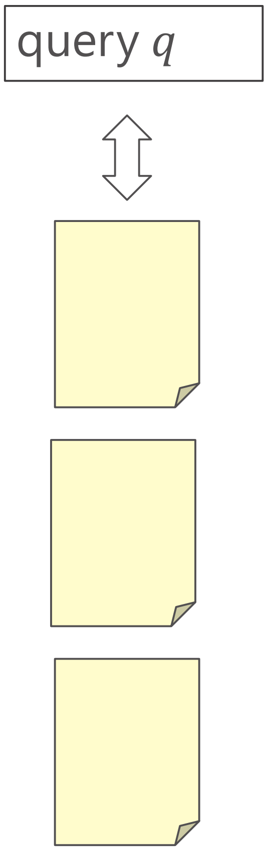


Hundreds of ranking signals: BM25, topic models, title / body / heading matches, phrase models, PageRank, HITS, date of publication / recent update, author, location …

Different combinations for different types of queries: head / body / tail, spiking queries, queries related to news, artists, TV shows, science, homepage finding (navigation), long-term information needs …

Impossible to tune models by hand for each query type – ML to the rescue …

Feature representation



feature representation

$$\boldsymbol{X}_q = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix}$$

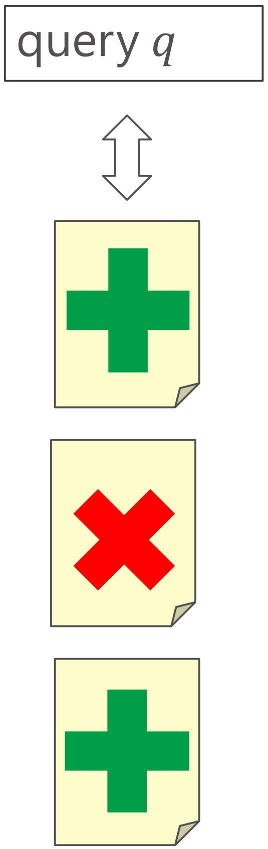
document i
X
ranking feature j

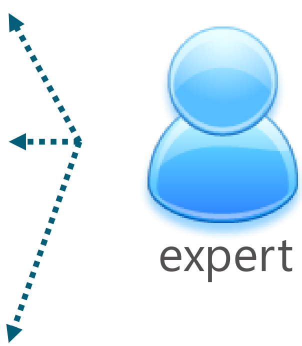
ranking

$\pi(\boldsymbol{X}_q)$

Goal:
find $\pi \in \Pi$ to optimize "user satisfaction"

Feature representation, expert labels





feature representation

$$\boldsymbol{X}_q = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix}$$

document i
X
ranking feature j

ranking

$$\pi(\boldsymbol{X}_q)$$

Goal:
find $\pi \in \Pi$ to optimize "user satisfaction"

Feature representation, expert labels

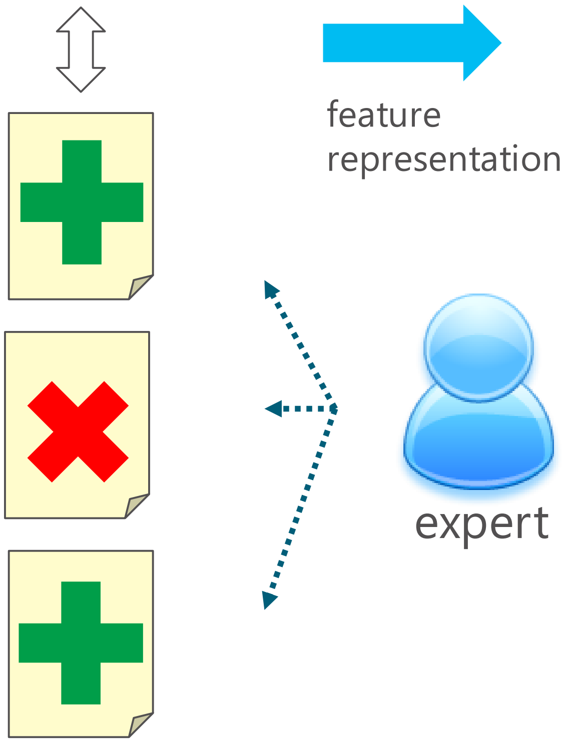
query q

feature representation

$$\boldsymbol{X}_q = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix}$$

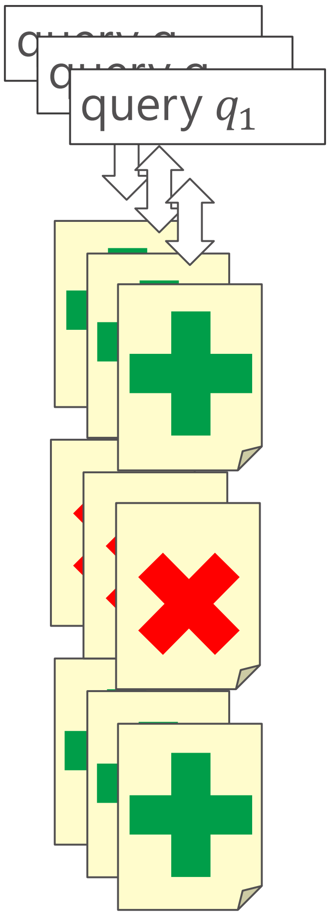
ranking

$$\pi(\boldsymbol{X}_q)$$



“TREC Assessors at work” (2001) – from http://www.itl.nist.gov/iad/photos/assessors.html

Feature representation, expert labels



training / test data for supervised learning

feature representation

$$\boldsymbol{X}_q = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix}$$

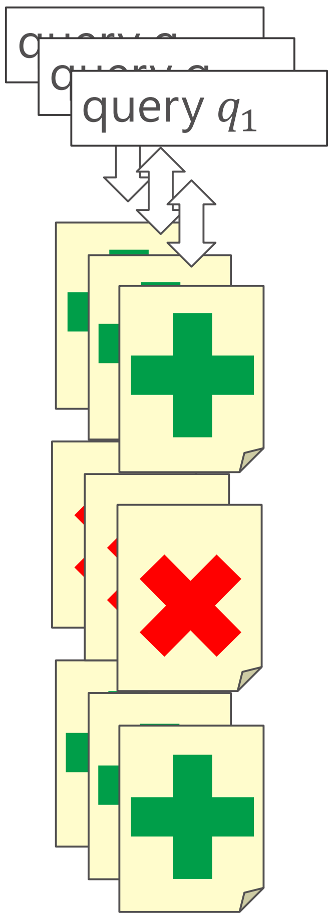
document i
X
ranking feature j

ranking

$$\pi(\boldsymbol{X}_q)$$

Goal:
find $\pi \in \Pi$ to optimize "user satisfaction"

Feature representation, expert labels



training / test data for supervised learning

feature representation

$$\boldsymbol{X}_q = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix}$$

document i
X
ranking feature j

ranking

$\pi(\boldsymbol{X}_q)$

Goal:
find $\pi \in \Pi$ to optimize "user satisfaction"

expert labels

$$\boldsymbol{Y}_q = \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix}$$

What loss function to optimize?

Proposed approaches: regression (squared error), classification (0-1 or hinge loss), ordinal regression, pairwise loss functions …

problem: mismatch with IR metrics!

Breakthrough: optimizing IR metrics using gradient descent

RankNet, LambdaRank, LambdaMART = family of learning to rank approaches that directly optimize smooth approximations of complex cost functions (e.g., NDCG, MAP, ERR)

Learning to Rank using Gradient Descent

Chris Burges CBURGES@MICROSOFT.COM
Tal Shaked* TAL.SHAKED@GMAIL.COM
Erin Renshaw ERINREN@MICROSOFT.COM
Microsoft Research, One Micr

Ari Lazier
Matt Deeds
Nicole Hamilton
Greg Hullender
Microsoft, One Microsoft Way

Abstra

We investigate using grad
ods for learning ranking
pose a simple probabilisti
we introduce RankNet, an
these ideas using a neural
the underlying ranking fu
test results on toy data a
commercial internet searc

From RankNet to LambdaRank to LambdaMART: An Overview

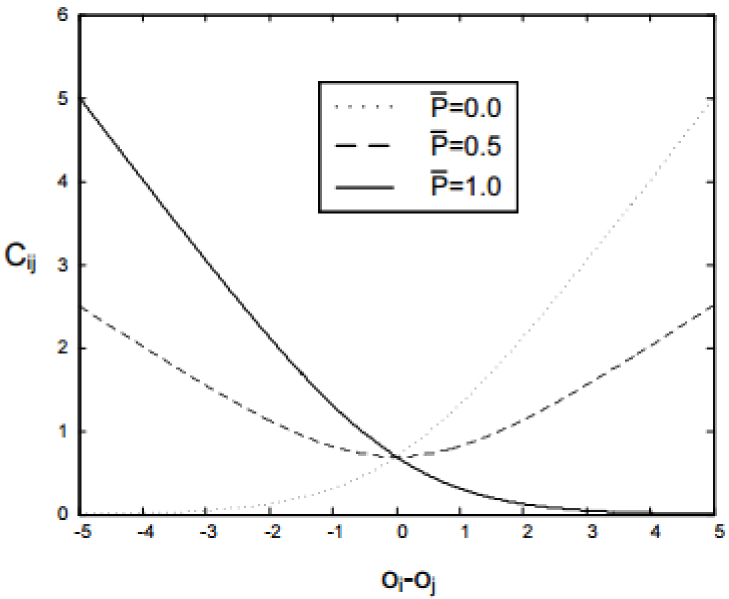
Christopher J.C. Burges
Microsoft Research Technical Report MSR-TR-2010-82

Abstract
LambdaMART is the boosted tree version of LambdaRank, which is based on RankNet. RankNet, LambdaRank, and LambdaMART have proven to be very successful algorithms for solving real world ranking problems: for example an ensemble of LambdaMART rankers won Track 1 of the 2010 Yahoo! Learning To Rank Challenge. The details of these algorithms are spread across several papers and reports, and so here we give a self-contained, detailed and complete description of them.

(target evaluation measure) that is not necessarily the one used to train the system. Typical target measures used in IR (see [8] for a detailed list) depend only on the sorted list and the relevance levels of the listed items. These measures are generally either flat everywhere or non-differentiable with respect to the model parameters; hence they are difficult to optimize directly. One way to address this issue is to find a close smooth approximation to the target measure, and optimize it via gradient descent. However, this is quite challenging due to the sort component of the target ranking measures. LambdaRank [1] tackles the problem by defining a smooth approximation to the *gradient* of the target cost instead of searching for a smooth approximation to the target cost itself. The basic idea of LambdaRank is to specify rules determining how the rank order of documents should change. These rules are incorporated into a λ-gradient that defines the gradient of an implicit cost function only at the points of interest [1]. LambdaRank was originally proposed for NDCG (Normalized Discounted Cumulative Gain), but

[Burges et al. ‘05, Donmez et al. ‘09, Burges ‘10]

From RankNet cost to LambdaRank gradients



Cost function C_{ij} for 3 ground truth values and varying score difference $o_{ij} = s_i$-s_j.

RankNet cost (w.r.t. changes in score difference o_{ij}):

$$C_{ij} \equiv -\bar{P}_{ij} o_{ij} + \log(1 + e^{\bar{P}_{ij} o_{ij}})$$

with ground truth

$$\bar{P}_{ij} = \begin{cases} +1 & if \;\; l_i > l_j \\ -1 & if \;\; l_j > l_i \end{cases}$$

and derivative

$$\frac{\partial C_{ij}}{\partial o_{ij}} = -\frac{\bar{P}_{ij}}{1+e^{\bar{P}_{ij} o_{ij}}}.$$

λ-gradients approximate the gradient of the target cost.

Here: scale the RankNet gradient with the change in target metric due to document swaps. E.g., for NDCG:

$$\lambda_{ij} \equiv \bar{P}_{ij} \left| \Delta NDCG \frac{\partial C_{ij}}{\partial o_{ij}} \right| =$$

$$\bar{P}_{ij} \left| \frac{1}{Z} (2^{l_i} - 2^{l_j}) \left(\frac{1}{\log(1+r_i)} - \frac{1}{\log(1+r_j)} \right) \left(\frac{1}{1+e^{\bar{P}_{ij} o_{ij}}} \right) \right|.$$

[Burges et al. '05, Donmez et al. '09, Burges '10]

With NDCG defined as:

$$\mathrm{NDCG@L} = \frac{1}{Z} \sum_{r=1}^{L} \frac{2^{l(r)} - 1}{\log(1 + r)}$$

Results

Excellent performance in Yahoo! learning to rank challenge

Yahoo! Learning to Rank

Olivier Chapelle*
Yi Chang
Yahoo! Labs
Sunnyvale, CA

Abstra

Learning to rank for information retrieval has g
but there is a lack for large real-world datasets t
publicly release two datasets used internally at Y
function. To promote these datasets and foster th
to rank algorithms, we organized the Yahoo! Le
This paper provides an overview and an analysi
description of the released datasets.

Table 7: Winners of the challenge along with the ERR score of their primary submission.

	Track 1	
1	C. Burges, K. Svore, O. Dekel, Q. Wu, P. Bennett, A. Pastusiak and J. Platt (Microsoft Research)	0.46861
2	E. Gottschalk (Activision Blizzard) and D. Vogel (Data Mining Solutions)	0.46786
3	M. Parakhin (Microsoft) – *Prize declined*	0.46695
4	D. Pavlov and C. Brunk (Yandex Labs)	0.46678
5	D. Sorokina (Yandex Labs)	0.46616
	Track 2	
1	I. Kuralenok (Yandex)	0.46348
2	P. Li (Cornell University)	0.46317
3	D. Pavlov and C. Brunk (Yandex Labs)	0.46311
4	P. Geurts (University of Liège)	0.46169

Used an ensemble of LambdaRank and LambdaMART rankers

+ enormous practical impact on web search engines!

[Chapelle & Chang '11]

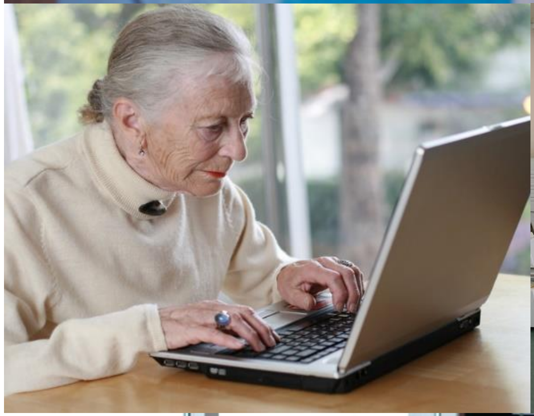
Present

From learning to rank to learning from user interactions

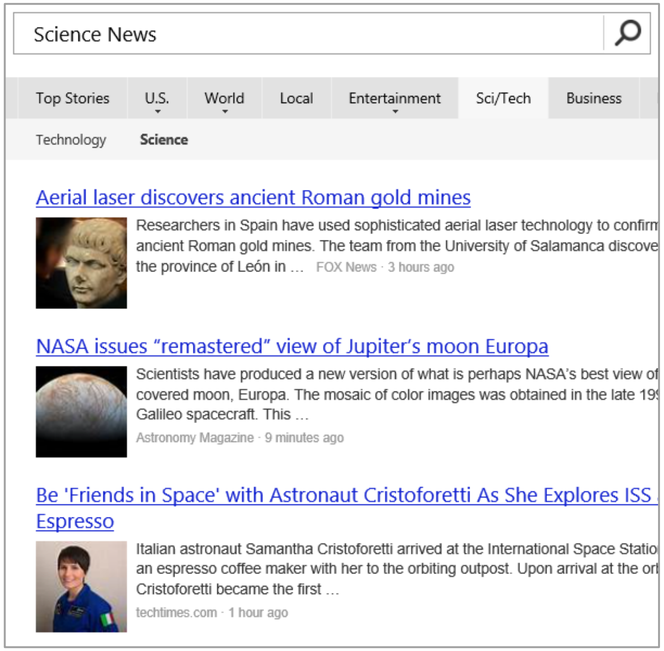
Part 2: learning from user interactions

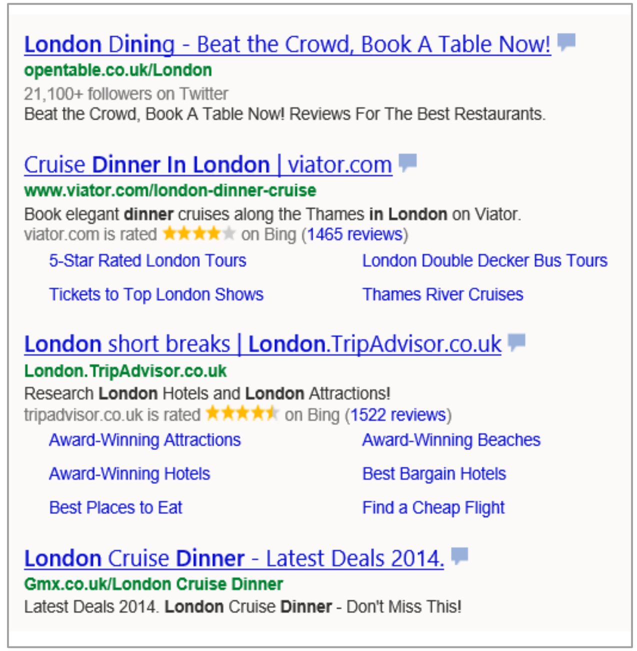
Are we there yet?

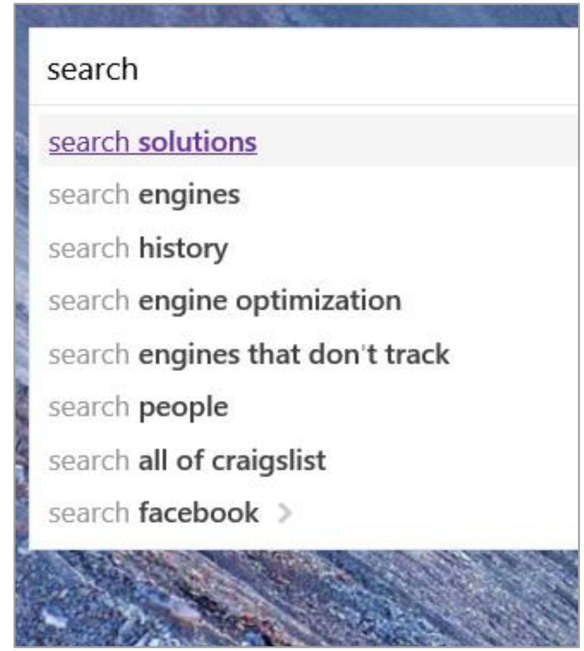
What would it take to actually learn from and for users?



Example applications







News recommendation

Find best news articles to match the users' current interest; optimize, e.g., click-through rate.

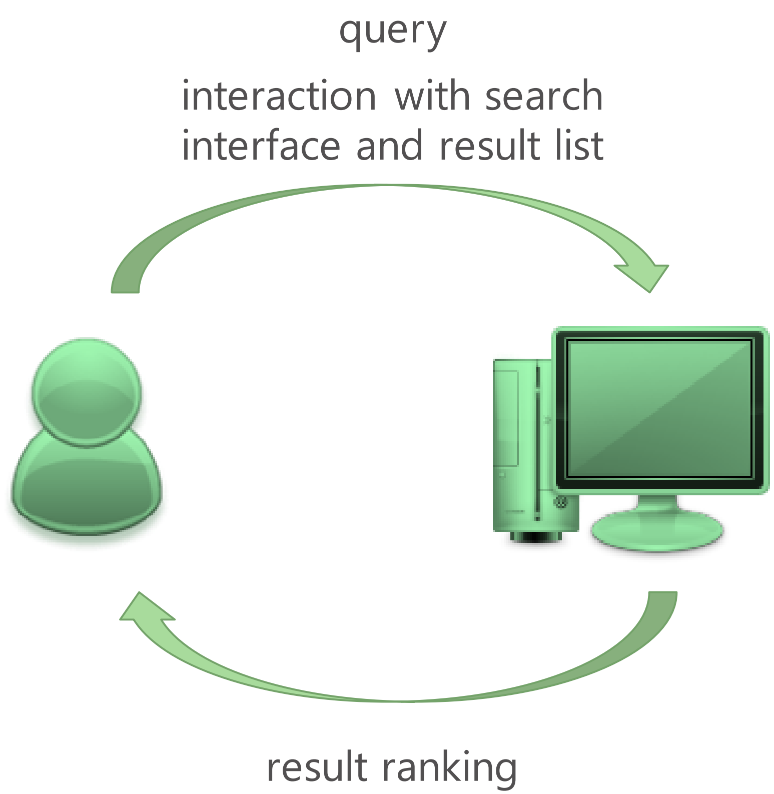
Online advertising

Tune ad display parameters (e.g., mainline reserve) to optimize revenue.

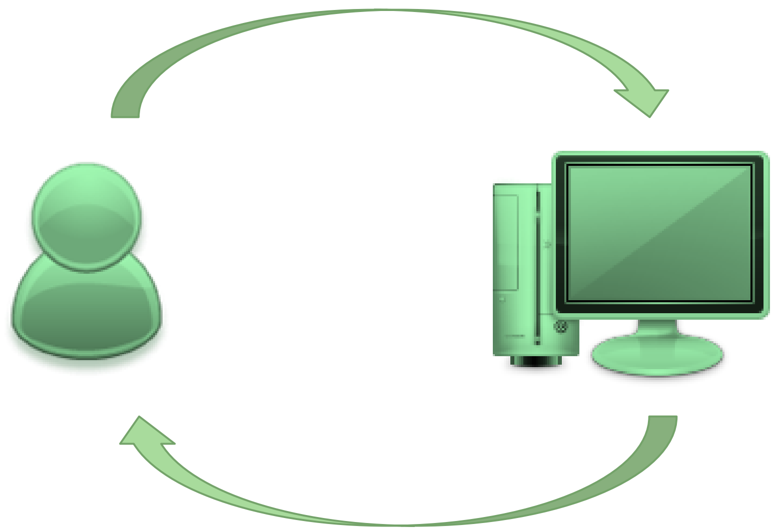
Web search

Adapt the search interface and results to individual users and their context; optimize search satisfaction.

Learning from user interactions



Mapping to reinforcement / bandit learning



Goal: take actions that maximize discounted cumulative reward $C = \sum_{t=0}^{\infty} \gamma^{t-1} r_t(a_t)$

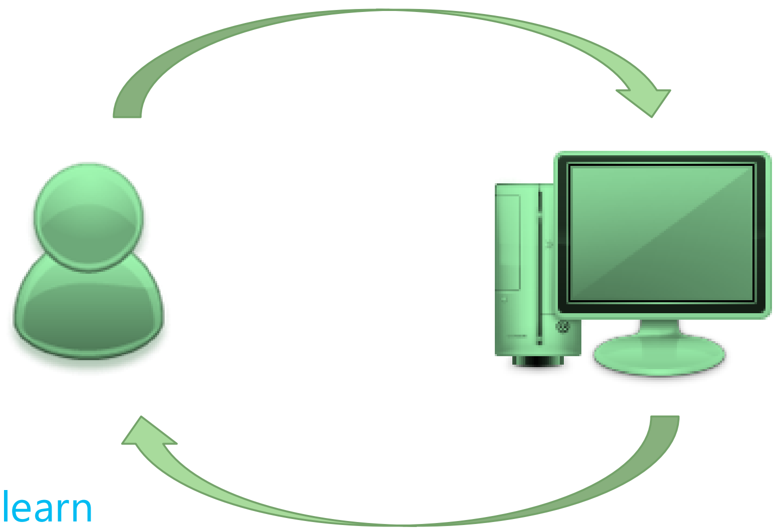
γ - discount factor

Bandit / RL challenges

Need to generalize across context.

query = *context* x_t

interaction with search interface and result list $\approx$ reward r_t



environment

generates context $x_t \sim X$

agent

acts according to policy $\pi(x)$

Partial feedback (need to learn through trial and error, no i.i.d. labeled examples) –exploration-exploitation trade-off.

result ranking = *action* $a_t = \pi_t(x_t)$

Goal: take actions that maximize discounted cumulative reward $C = \sum_{t=0}^{\infty} \gamma^{t-1} r_t(a_t)$

γ - discount factor

IR challenges

query $=$ *context* x_t

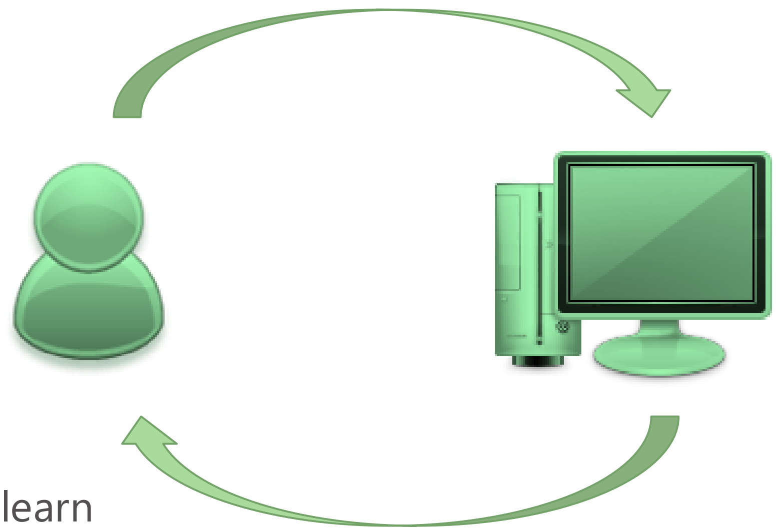
Need to generalize across context.

interaction with search interface and result list $\approx$ reward r_t

Need to infer feedback for learning from noisy, biased user interactions (relative feedback).

environment

generates context $x_t \sim X$



agent

acts according to policy $\pi(x)$

Partial feedback (need to learn through trial and error, no i.i.d. labeled examples) –exploration-exploitation trade-off.

result ranking $=$ *action* $a_t = \pi_t(x_t)$

Need to deal with large action spaces.

Goal: take actions that maximize discounted cumulative reward $C = \sum_{t=0}^{\infty} \gamma^{t-1} r_t(a_t)$

γ - discount factor

Approaches for learning from user interactions

Approach	Explore-exploit	Generalize across context	Relative feedback	Large action spaces
K-armed bandits	+	-	-	+/-
Dueling K-armed bandits	+	-	+	+/-
Online learning to rank	(implicit)	(only linear)	+	+
Contextual bandits	+	+	-	-
Contextual dueling bandits	+	+	+	-

Approaches for learning from user interactions – this lecture

Approach	Explore-exploit	Generalize across context	Relative feedback	Large action spaces
K-armed bandits	+	-	-	+/-
Dueling K-armed bandits	+	-	+	+/-
Online learning to rank	(implicit)	(only linear)	+	+
Contextual bandits	+	+	-	-
Contextual dueling bandits	+	+	+	-

K-armed bandits

[Sutton & Barto '98]

Balance exploration and exploitation, e.g.,:

ε-greedy: $\pi_t(x_t) = \begin{cases} \underset{a \in A}{\operatorname{argmax}}\, \hat{r}_t(a) & w.prob.\, 1-\varepsilon \\ rand(a) & w.prob.\, \varepsilon \end{cases}$

UCB (upper confidence bound): $\pi_t(x_t) = \underset{a \in A}{\operatorname{argmax}}\, \hat{r}_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}$

[Auer et al. '02a]

Thompson sampling

Maintain distribution $P(r|a)$. At time t sample from this distribution, and take the optimal action according to the sample; update P.

[Thompson '33, Chapelle & Li '11, Russo & Van Roy '14]

Exponential weights (EXP3)

Maintain a distribution over actions, p_t, sample from it to take actions. Updates: compute estimated cumulative reward $\widehat{R_{i,t}} = \widehat{R_{i,t-1}} + \frac{r_{i,t}}{p_{i,t}}$ ($a_{i,t}$ is the action taken in round t). Update $p_{i,t+1} = \frac{e^{-\eta \widehat{R_{i,t}}}}{\sum_{k=1}^{K} e^{-\eta \widehat{R_{k,t}}}}$.

[Auer et al. '02b]

Easy to implement and flexible, nice theoretical characteristics.

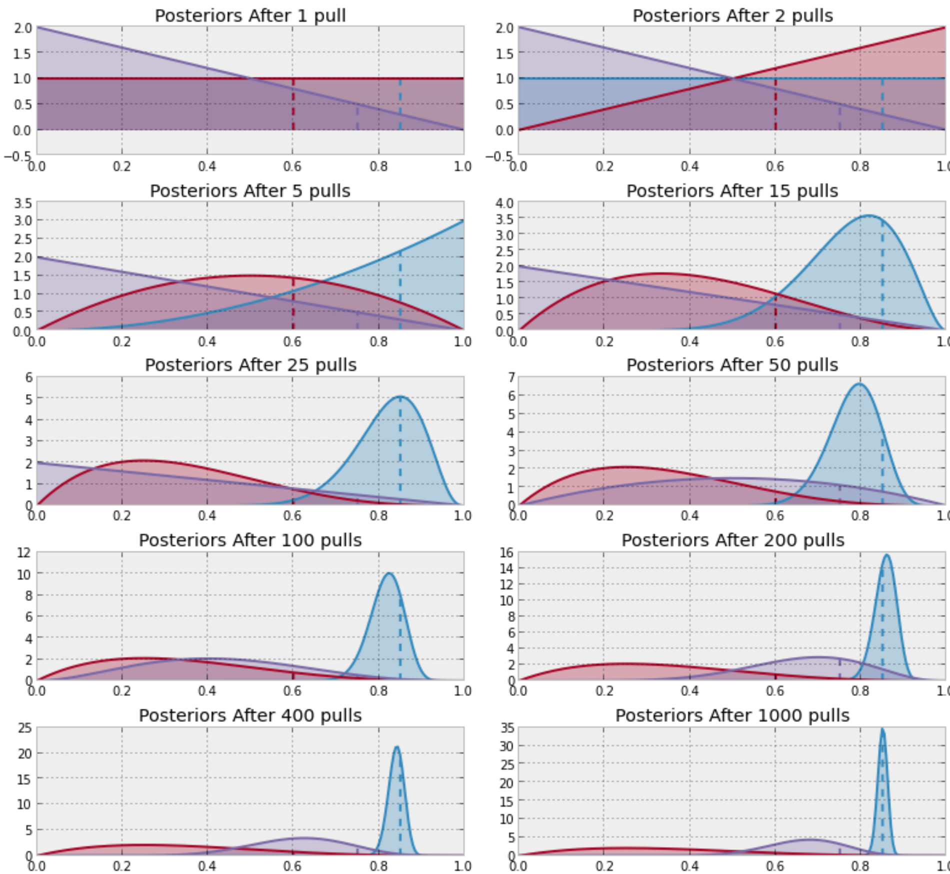
Performance guarantees in adversarial settings!

Example: Thompson Sampling

Balances exploration-exploitation using a simple principle:

1. maintain reward distribution (e.g., per action)
2. when taking an action, sample from that distribution and act optimally according to the sample
3. use observed reward to update reward distribution

Example: 3-armed bandit



[Thompson '33, Chapelle & Li '11, Russo & Van Roy '14]

For IR:
need to address context and relative feedback

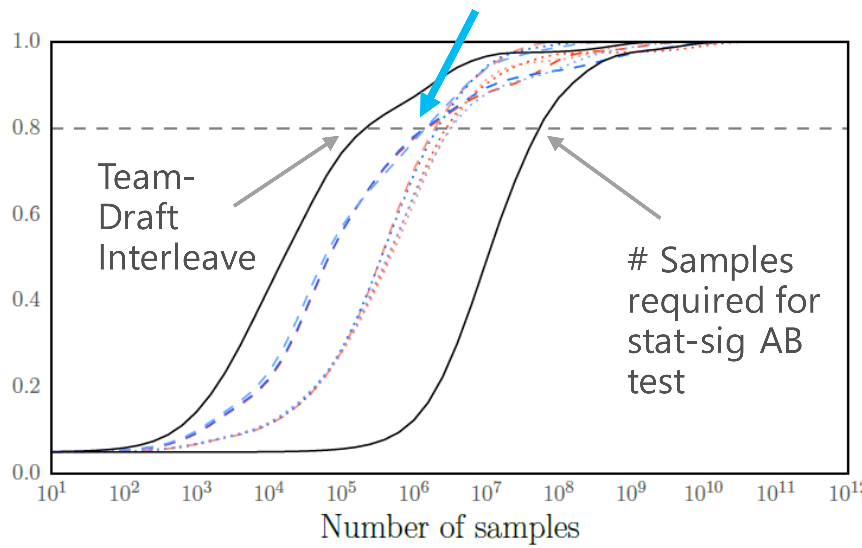
1 - Query-document pairs are represented as feature vectors

$$X_q = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix}$$

document i
X
ranking feature j

2 - Relative feedback (interleaved comparison is more reliable than absolute metrics):

Best trade-off between **high agreement** with long-term AB metrics (85%) and **sample efficiency.**



Power as a function of sample size, assuming the target test is a two-sided t-test with p = 0.05.

[Schuth et al. 2015]

Online learning to rank

Reusing Historical Interaction Data for Faster Online Learning to Rank for IR

Katja Hofmann k.hofmann@uva.nl
Anne Schuth a.g.schuth@uva.nl
Shimon Whiteson s.a.whiteson@uva.nl
Maarten de Rijke derijke@uva.nl
ISLA, University of Amsterdam

ABSTRACT

Online learning to rank for information retrieval (IR) holds promise for allowing the development of "self-learning" search engines that can automatically adjust to their users. With the large amount of e.g., click data that can be collected in web search settings, such techniques could enable highly scalable ranking optimization. However, feedback obtained from user interactions is noisy, and developing approaches that can learn from this feedback quickly and reliably is a major challenge.

In this paper we investigate whether and how previously collected (historical) interaction data can be used to speed up learning in online learning to rank for IR. We devise the first two methods that can utilize historical data (1) to make feedback available during learning more reliable and (2) to preselect candidate ranking functions to be evaluated in interactions with users of the retrieval system. We evaluate both approaches on 9 learning to rank data sets and find that historical data can speed up learning, leading to substantially and significantly higher online performance. In particular, our preselection method proves highly effective at compensating for noise in user feedback. Our results show that historical data can be used to make online learning to rank for IR much more effective than previously possible, especially when feedback is noisy.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

Keywords

Information retrieval, Interleaved comparisons, Learning to Rank

1. INTRODUCTION

In recent years, learning to rank methods have become popular in information retrieval (IR) as a means of tuning retrieval systems to the requirements of specific search environments, groups of users, or individual users [18]. However, most current approaches work *offline*, meaning that manually annotated data needs to be collected beforehand, and that, once deployed, the system cannot continue to adjust to user needs, unless it is retrained with additional data.

An alternative setting is *online* learning to rank, where the system learns directly from interactions with its users [10]. These approaches are typically based on reinforcement learning (RL) techniques [25], meaning that the system tries out new ranking functions (also called rankers), and learns from feedback inferred from users' interactions with the presented rankings. In contrast to offline learning to rank approaches, online approaches do not require any initial training material, but rather automatically improve rankers while they are being used.

A main challenge that online learning to rank for IR approaches have to address is to learn as quickly as possible from the limited quality and quantity of feedback that can be inferred from user interactions. Learning speed is particularly important in terms of the number of user interactions. The better the system's performance is after a smaller number of interactions, the more likely users are to be satisfied with the system. Also, the more effective an online learning to rank algorithm is, the more feasible it is to adapt to smaller groups of users, or even individual users.[1] Furthermore, user feedback is limited because the learning algorithm should be invisible to system users, i.e., feedback is inferred from natural (noisy) user interactions.

In this paper, we address the following question: *Can previously observed (historical) interaction data be reused to speed up online learning to rank?* Current online learning to rank approaches for IR utilize each observed data sample (consisting of a query, the displayed results, and observed user clicks on the result list) only once. This was necessary because, until recently, it was not clear how feedback from previous user interactions (that were collected with different rankers) could be reused. However, a recently developed probabilistic method for inferring relative feedback [9] has been shown to allow data re-use [11] for ranker evaluation. It was found to be effective for making ranker comparisons more reliable, especially when large amounts of historical data were available. Here, we investigate whether and how this evaluation method can be integrated with online learning to rank approaches, and whether and in what way these additional (historical, and possibly noisier or biased) evaluations can lead to faster learning.

Specifically, we make the following contributions.

- We propose the first two approaches for reusing historical data in online learning to rank for IR: *reliable historical comparisons* (RHC), which uses historical data directly to make feedback more reliable, and *candidate preselection* (CPS), which uses historical data to preselect candidate rankers.
- In extensive experiments using 9 learning to rank data sets, we

[1] In this paper we focus on the effectiveness of the learning algorithm, and assume a system is used by a group of users with similar search behavior and preferences. The question of how to form user groups to which to adapt is orthogonal to this work.

WSDM'13, February 4–8, 2013, Rome, Italy.

[Hofmann et al. '13b]

3 insights allow effective learning from relative, listwise feedback:

1 – Comparison

Interleaved comparison methods [Joachims et al. '05, Radlinski et al. '08], infer listwise relative feedback.

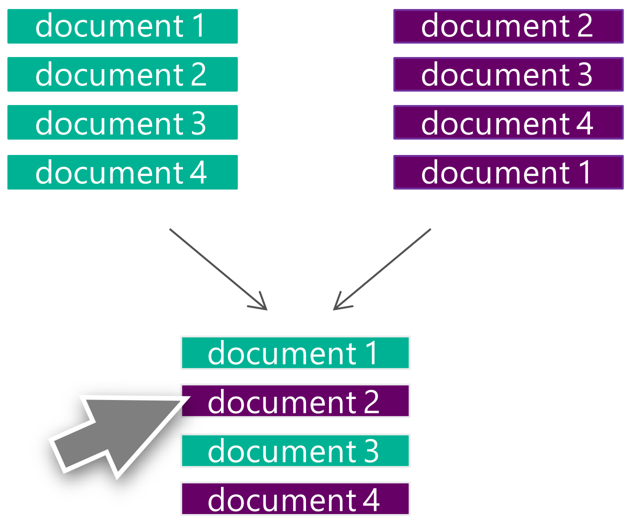
2 – Learning

Dueling bandit gradient descent (DBGD) [Yue & Joachims '09] optimizes a weight vector for weighted-linear combinations of ranking features.

3 – Data reuse

Use probabilistic interleave [Hofmann et al. '13a] and importance sampling to compare candidate rankers on historical data and focus exploration [Hofmann et al. '13b], approach called candidate pre-selection (CPS).

Online learning to rank



Interleaved comparisons:

1) Generate interleaved (combined) ranking
2) Observe user clicks
3) Credit clicks to original rankers to infer outcome $o \in \{-1, 0, +1\}$

3 insights allow effective learning from relative, listwise feedback:

1 – Comparison

Interleaved comparison methods [Joachims et al. '05, Radlinski et al. '08], infer listwise relative feedback.

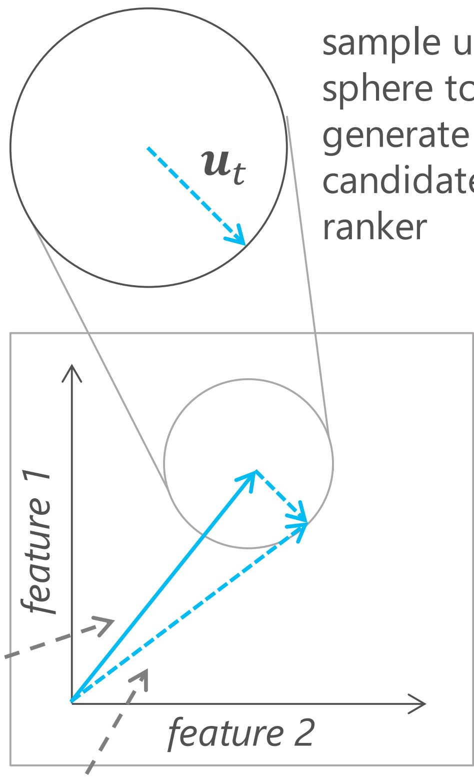
2 – Learning

Dueling bandit gradient descent (DBGD) [Yue & Joachims '09] optimizes a weight vector for weighted-linear combinations of ranking features.

3 – Data reuse

Use probabilistic interleave [Hofmann et al. '13a] and importance sampling to compare candidate rankers on historical data and focus exploration [Hofmann et al. '13b], approach called candidate pre-selection (CPS).

Online learning to rank



sample unit sphere to generate candidate ranker

current best weight vector $\boldsymbol{w}_t$

generate candidate vector
$\boldsymbol{w}'_t = \boldsymbol{w}_t + \delta \boldsymbol{u}_t$

compare $\boldsymbol{w}_t$ and $\boldsymbol{w}'_t$ using interleaving; if $\boldsymbol{w}'_t$ wins, update:
$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha \boldsymbol{u}_t$

3 insights allow effective learning from relative, listwise feedback:

1 – Comparison

Interleaved comparison methods [Joachims et al. '05, Radlinski et al. '08], infer listwise relative feedback.

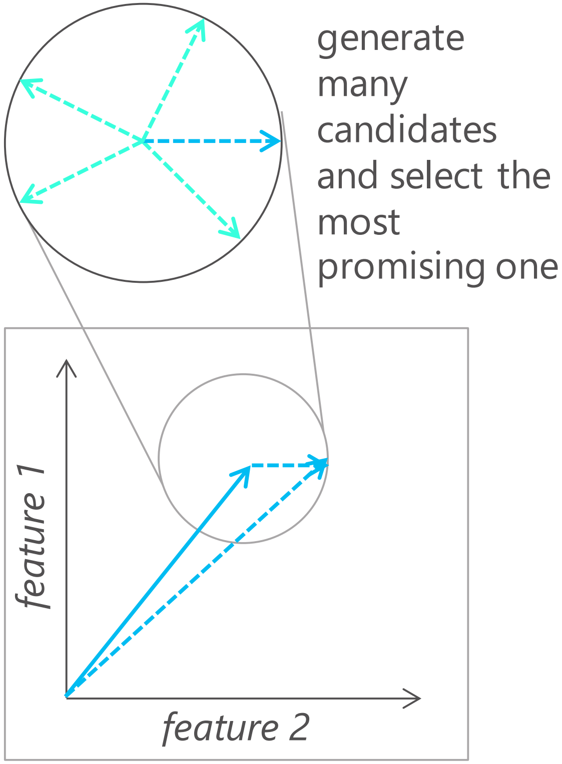
2 – Learning

Dueling bandit gradient descent (DBGD) [Yue & Joachims '09] optimizes a weight vector for weighted-linear combinations of ranking features.

3 – Data reuse

Use probabilistic interleave [Hofmann et al. '13a] and importance sampling to compare candidate rankers on historical data and focus exploration [Hofmann et al. '13b], approach called candidate pre-selection (CPS).

Online learning to rank



Generate several candidate rankers, and select the best one by running a tournament on historical data

Use probabilistic interleave and importance sampling for ranker comparisons during the tournament

3 insights allow effective learning from relative, listwise feedback:

1 – Comparison

Interleaved comparison methods [Joachims et al. '05, Radlinski et al. '08], infer listwise relative feedback.

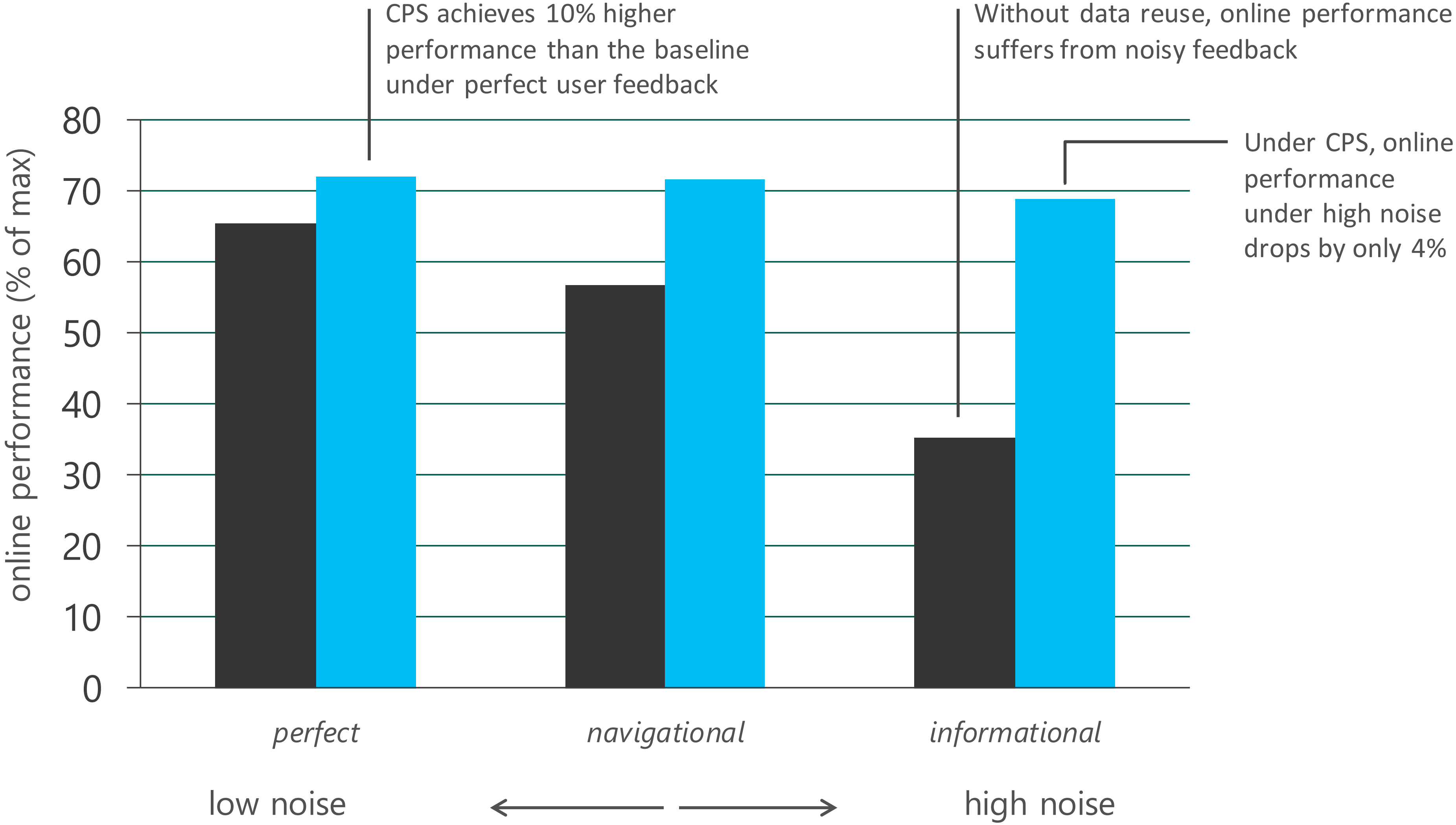
2 – Learning

Dueling bandit gradient descent (DBGD) [Yue & Joachims '09] optimizes a weight vector for weighted-linear combinations of ranking features.

3 – Data reuse

Use probabilistic interleave [Hofmann et al. '13a] and importance sampling to compare candidate rankers on historical data and focus exploration [Hofmann et al. '13b], approach called candidate pre-selection (CPS).

Results: candidate pre-selection (CPS)



[Hofmann et al. '13b]

Moving beyond linear models: contextual dueling bandits

Learn from partial relative feedback in the form of pairwise comparisons; use supervised ML for generalization.

P	1	2	3	4
1	0	0.1	-0.2	-0.9
2	-0.1	0	0.4	0
3	0.2	-0.4	0	1.0
4	0.9	0	-1.0	0

Example preference matrix for a 4-armed dueling bandit problem.

JMLR: Workshop and Conference Proceedings vol 40:1–25, 2015

Contextual Dueling Bandits

Miroslav Dudík MDUDIK@MICROSOFT.COM
Microsoft Research, New York, NY USA
Katja Hofmann KATJA.HOFMANN@MICROSOFT.COM
Microsoft Research, Cambridge, United Kingdom
Robert E. Schapire* SCHAPIRE@MICROSOFT.COM
Microsoft Research, New York, NY USA
Aleksandrs Slivkins SLIVKINS@MICROSOFT.COM
Microsoft Research, New York, NY USA
Masrour Zoghi† M.ZOGHI@UVA.NL
University of Amsterdam, Amsterdam, The Netherlands

Abstract

We consider the problem of learning to choose actions using contextual information when provided with limited feedback in the form of relative pairwise comparisons. We study this problem in the dueling-bandits framework of Yue et al. (2009), which we extend to incorporate context. Roughly, the learner's goal is to find the best policy, or way of behaving, in some space of policies, although "best" is not always so clearly defined. Here, we propose a new and natural solution concept, rooted in game theory, called a *von Neumann winner*, a randomized policy that beats or ties every other policy. We show that this notion overcomes important limitations of existing solutions, particularly the Condorcet winner which has typically been used in the past, but which requires strong and often unrealistic assumptions. We then present three efficient algorithms for online learning in our setting, and for approximating a von Neumann winner from batch-like data. The first of these algorithms achieves particularly low regret, even when data is adversarial, although its time and space requirements are linear in the size of the policy space. The other two algorithms require time and space only logarithmic in the size of the policy space when provided access to an oracle for solving classification problems on the space.
Keywords: contextual dueling bandits, online learning, bandit algorithms, game theory.

1. Introduction

We study how to learn to act based on contextual information when provided only with partial, relative feedback. This problem naturally arises in information retrieval (IR) and recommender systems, where the user feedback is considerably more reliable when interpreted as relative comparisons rather than absolute labels (Radlinski et al., 2008). For instance, in web search, for a particular query, the IR system may have several candidate rankings of documents that could be presented, with the best option being dependent upon the specific user. By presenting a mix or interleaving of two of the candidate rankings and observing the user's response (Chapelle et al., 2012; Hofmann et al., 2013), it is possible for such a system to get feedback about user preferences.

* On leave from Princeton University.
† Part of this research was conducted during an internship with Microsoft Research.

[Dudik et al. '15]

Contextual Dueling Bandits

Learn from partial relative feedback in the form of pairwise comparisons

In each round:

select two actions $a, b \in A = \{1 \ldots K\}$
observe outcome o of a stochastic duel:

$$o = \begin{cases} +1 \text{ if } a \text{ beats } b \\ -1 \text{ if } b \text{ beats } a \end{cases}$$

Goal: learn to select the best action, given context.

[Dudik et al. '15]

Contextual Dueling Bandits

Learn from partial relative feedback in the form of pairwise comparisons, given context.

In each round:

environment chooses context $x \in X$

select two actions $a, b \in A = \{1 \dots K\}$

observe outcome o of a stochastic duel:

$$o = \begin{cases} +1 \text{ if } a \text{ beats } b \\ -1 \text{ if } b \text{ beats } a \end{cases}$$

Goal: learn to select the best action, given context.

[Dudik et al. '15]

Challenges & Contributions

How to define the “best” action?

Introduce new solution concept: the von Neumann winner (simple, natural, guaranteed to exist).

How to use context effectively?

Extend the von Neumann winner to the contextual dueling bandit setting.

Present learning algorithms for computing, approximating, or performing as well as the “best” solution.

[Dudik et al. ’15]

Solution Concept

Previous solutions

Make assumptions about preference structure.

Example: transitivity

$P(a, b) > 0 \wedge P(b, c) > 0 \Leftrightarrow P(a, c) > 0$

Example: Condorcet assumption

$\exists a \in A \quad \forall b \in A \quad P(a, b) \geq 0$

All are frequently violated in practice.

Define $P(a, b) = E$ [outcome of duel between a and b]
"a beats b" when $P(a, b) > 0$; "a ties b" when $P(a, b) = 0$

[Dudik et al. '15]

Solution Concept

Previous solutions

Make assumptions about preference structure.

Example: transitivity

$P(a,b) > 0 \wedge P(b,c) > 0 \Leftrightarrow P(a,c) > 0$

Example: Condorcet assumption

$\exists a \in A \quad \forall b \in A \quad P(a,b) \geq 0$

All are frequently violated in practice.

Contextual dueling bandits

Idea: Sample actions from a distribution w.

Goal: find w such that:

$$\forall b : E\,[outcome] = \sum_a w(a)P(a,b) \geq 0$$

When this holds, w is called the von Neumann winner (guaranteed to exist).

Define $P(a,b) = E$ [outcome of duel between a and b]
"a beats b" when $P(a,b) > 0$; "a ties b" when $P(a,b) = 0$

[Dudik et al. '15]

Contextual Setting

Goal: find the "best" policy $\pi \in \Pi$

Extend von Neumann winner to contextual dueling bandits:

Randomized policy (distribution over Π) that beats or ties every other policy ρ - guaranteed to exist

[Dudik et al. '15]

Contextual Setting

Goal: find the "best" policy $\pi \in \Pi$

Extend von Neumann winner to contextual dueling bandits:

Randomized policy (distribution over Π) that beats or ties every other policy ρ - guaranteed to exist

Challenges

Π typically huge! Need to solve a $|\Pi| \times |\Pi|$ game matrix M.

Even representing von Neumann winner seems to require $O(|\Pi|)$ space

Wanted: time, space, data in $poly(\log|\Pi|)$

[Dudik et al. '15]

Contextual Setting

Goal: find the "best" policy $\pi \in \Pi$

Extend von Neumann winner to contextual dueling bandits:

Randomized policy (distribution over Π) that beats or ties every other policy ρ - guaranteed to exist

Challenges

Π typically huge! Need to solve a $|\Pi| \times |\Pi|$ game matrix M.

Even representing von Neumann winner seems to require $O(|\Pi|)$ space

Wanted: time, space, data in $poly(\log|\Pi|)$

Algorithms

Full-explore-exploit setting

Optimal regret but $O(|\Pi|)$ time and space.

Explore-first setting

Suboptimal regret/approximation bound, $poly(\log|\Pi|)$ time and space.

[Dudik et al. '15]

Contextual Setting

Goal: find the "best" policy $\pi \in \Pi$

Extend von Neumann winner to contextual dueling bandits:

Randomized policy (distribution over Π) that beats or ties every other policy ρ - guaranteed to exist

Challenges

Π typically huge! Need to solve a $|\Pi| \times |\Pi|$ game matrix M.

Even representing von Neumann winner seems to require $O(|\Pi|)$ space

Wanted: time, space, data in $poly(\log|\Pi|)$

Algorithms

Full-explore-exploit setting

Optimal regret but $O(|\Pi|)$ time and space.

Explore-first setting

Suboptimal regret/approximation bound, $poly(\log|\Pi|)$ time and space.

Open problem: optimal-regret or optimal-approximation algorithm with $poly(\log|\Pi|)$ time and space!

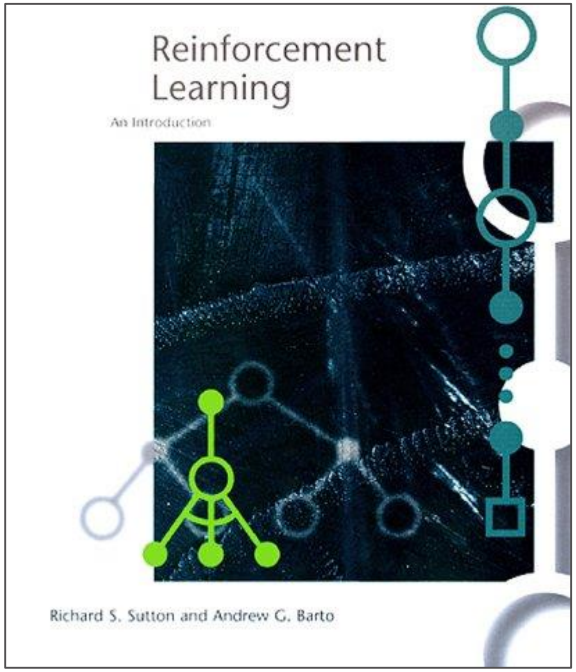
[Dudik et al. '15]

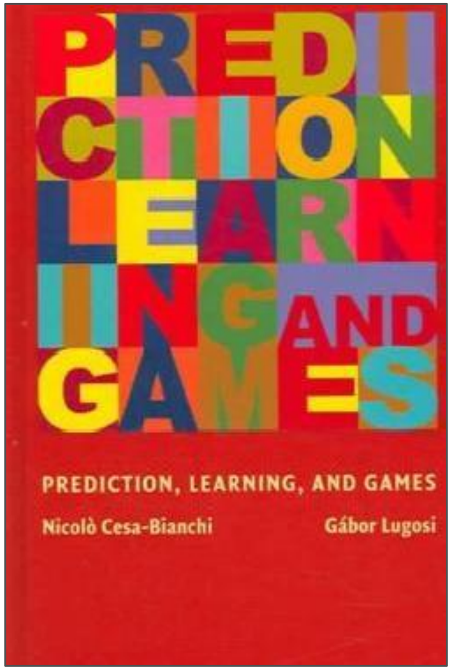
Approaches for learning from user interactions – this lecture

Approach	Explore-exploit	Generalize across context	Relative feedback	Large action spaces
K-armed bandits	+	-	-	+/-
Dueling K-armed bandits	+	-	+	+/-
Online learning to rank	(implicit)	(only linear)	+	+
Contextual bandits	+	+	-	-
Contextual dueling bandits	+	+	+	-

Further reading

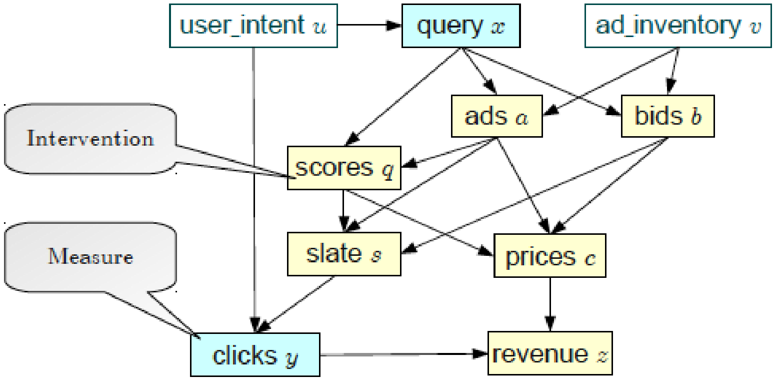
Background in reinforcement learning [Sutton & Barto '98] online learning theory, contextual bandits [Cesa-Bianchi & Lugosi '06]





K-armed dueling bandits [Zoghi et al. '15]
Bandit overview and theory [Bubeck & Cesa-Bianchi]

Complementary view: counterfactual analysis and reasoning [Bottou et al. '13]



Future

Trends and directions

Supervised ML for IR

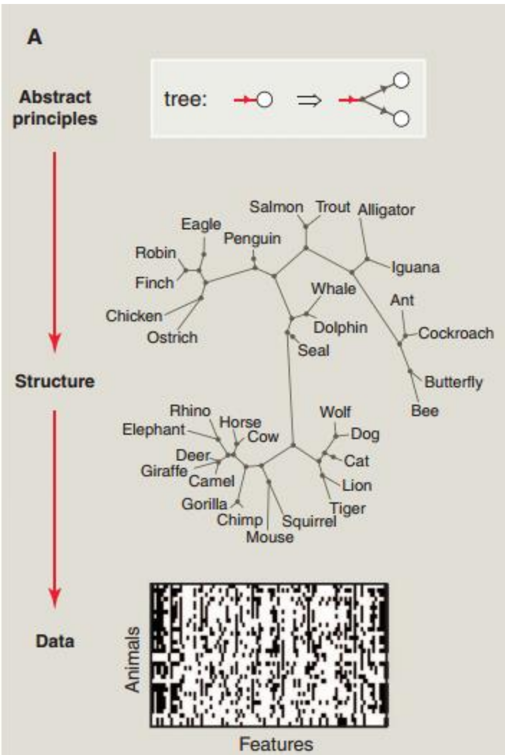


Image from [Tenenbaum et al. '11]

Structure learning: how to represent / learn about rankings, or more complex composite actions? [Tenenbaum et al. '11]

Representation learning for documents, users, information needs? [Mikolov et al. '13, Huang et al. '13]

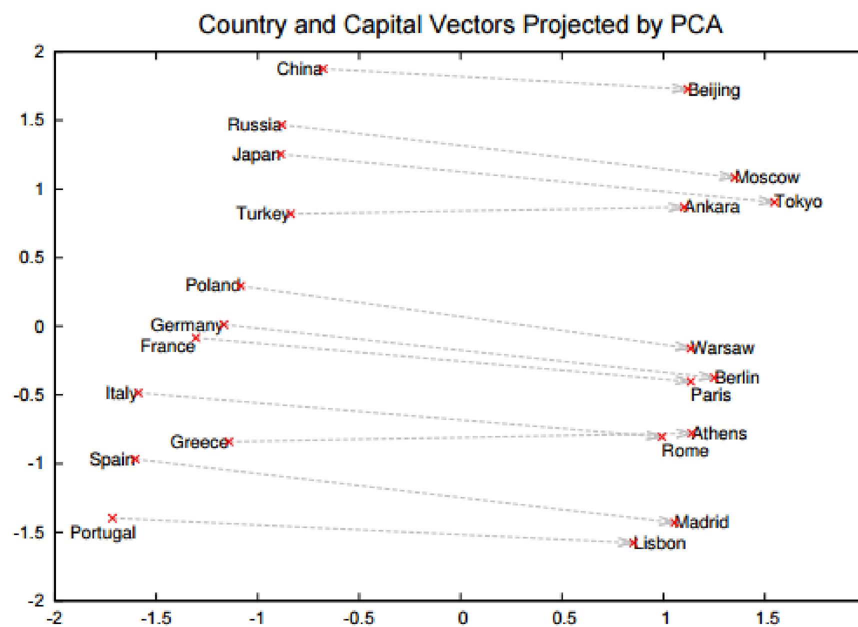


Image from [Mikolov et al. '13]

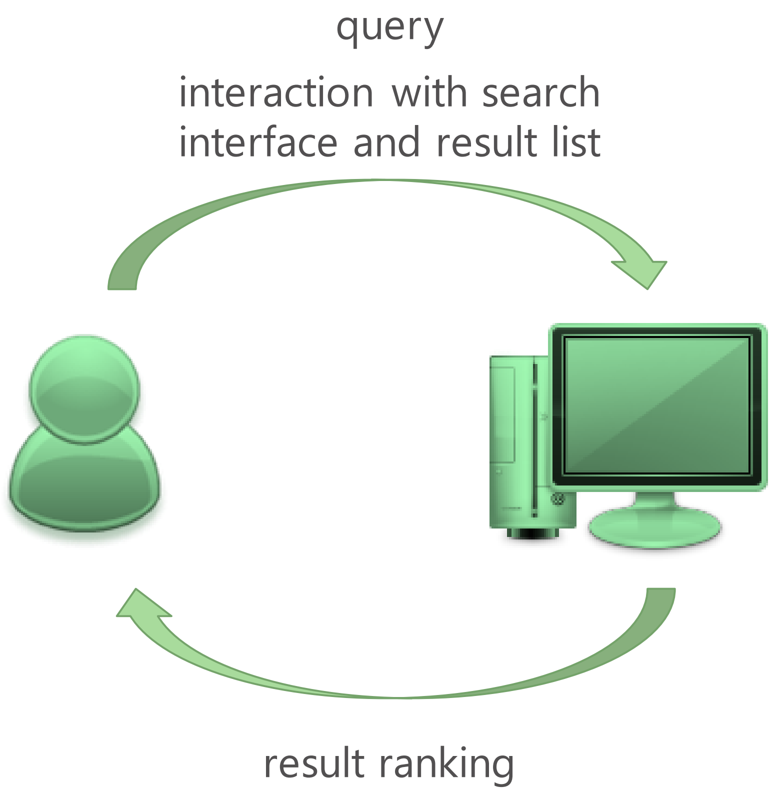
Interactive learning for IR

Efficient approaches to generalizing across context in large action spaces

Modeling long-term dependencies [Yang et al. '14, Sloan & Wang '12]

Challenges in evaluation

[Balog et al. '14]



Interactive learning for IR

Efficient approaches to generalizing across context in large action spaces

Modeling long-term dependencies

Challenges in evaluation

Try online learning

Try open-source code samples; Living labs challenge allows experimentation with interactive learning and evaluation methods

Code: https://bitbucket.org/ilps/lerot

Challenge: http://living-labs.net/challenge/

References

References and further reading

[Auer et al. '02a] P. Auer, N. Cesa-Bianchi, P. Fischer: *Finite-time analysis of the Multiarmed Bandit Problem*. Machine Learning 47, 2002a.

[Auer et al. '02b] P. Auer, N. Cesa-Bianchi, Y. Freund, & R. E. Schapire: *The nonstochastic Multiarmed Bandit Problem*. SIAM Journal of Computing 32(1) 2002b.

[Balog et al. '14] K. Balog, L. Kelly, & A. Schuth: *Head First: Living Labs for Ad-hoc Search Evaluation*. CIKM 2014.

[Bishop '07] C. M. Bishop: *Pattern Recognition and Machine Learning*. Springer 2007.

[Bottou et. al '13] L. Bottou, J. Peters, J. Quiñonero-Candela, D.X. Charles, D.M. Chickering, E. Portugaly, D. Ray, P. Simard, E. Snelson: *Counterfactual reasoning and learning systems: the example of computational advertising* (Journal of Machine Learning Research 14 (1), 2013).

[Bubeck & Cesa-Bianchi '12] S. Bubeck & N. Cesa-Bianchi: *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*. Foundations and Trends in Machine Learning 5(1), 2012.

[Burges et al. '05] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazler, M. Deeds, N. Hamilton, G. Hullender: *Learning to Rank using Gradient Descent*. ICML, 2005.

[Burges '10] C. J. C. Burges: *From RankNet to LambdaRank to LambdaMART: An Overview*. Microsoft Research Technical Report MSR-TR-2010-82, 2010.

[Cesa-Bianchi & Lugosi '06] N. Cesa-Bianchi & G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[Chapelle & Chang '11] O. Chapelle & Y. Chang: *Yahoo! Learning to Rank Challenge Overview*. Proc. Yahoo! Learning to Rank Challenge, 2011.

[Chapelle, Joachims & Radlinski '12] O. Chapelle, T. Joachims, F. Radlinski, & Y. Yue: *Large Scale Validation and Analysis of Interleaved Search Evaluation* (ACM Transactions on Information Systems 30(1): 6, 2012).

[Chapelle & Li '11] O. Chapelle & L. Li: *An Empirical Evaluation of Thompson Sampling*. NIPS, 2011.

[Chen & Hofmann '15] Y. Chen & K. Hofmann: *Online Learning to Rank: Absolute vs. Relative*. WWW 2015.

[Croft '81] W. B. Croft: *Incorporating Different Search Models into One Document Retrieval System*. SIGIR 1981..

[Donmez et al. '09] P. Donmez, K. M. Svore, C. J. C. Burges: *On the Local Optimality of LambdaRank*. SIGIR 2009..

[Dudik et al. '15] M. Dudík, K. Hofmann, R. E. Schapire, A. Slivkins, & M. Zoghi, *Contextual Dueling Bandits* (COLT 2015).

References and further reading

[Hastie et al. '09] T. Hastie, R. Tibshirani, J. Friedman: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd Edition. Springer 2009.

[Hofmann et al. '13a] K. Hofmann, S. Whiteson, & M. de Rijke: *Fidelity, Soundness, and Efficiency of Interleaved Comparison Methods* (ACM Transactions on Information Systems 31(4): 17, 2013).

[Hofmann et al. '13b] K. Hofmann, A. Schuth, S. Whiteson, & M. de Rijke: *Reusing Historical Interaction Data for Faster Online Learning to Rank for IR* (WSDM 2013).

[Hofmann et al. '13c] K. Hofmann, S. Whiteson, & M. de Rijke: *Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval* (Information Retrieval 16, 2013).

[Huang et al. '13] P. Huang, X. H, J. Gao, L. Deng, A. Acero, L. Heck: *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*,. CIKM 2013.

[Lewis et al. '96] D.D. Lewis, R.E. Schapire, J.P. Callan, R. Papka: *Training Algorithms for Linear Text Classifiers* (SIGIR 1996).

[Li, Chu, Langford & Wang '11] L. Li, W. Chu, J. Langford, & X. Wang: *Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms* (WWW, 2014).

[Li, Chu, Langford, Moon & Wang '12] L. Li, W. Chu, J. Langford, T. Moon, & X. Wang: *An Unbiased Offline Evaluation of Contextual Bandit Algorithms based on Generalized Linear Models*, ICML-2011 Workshop on Online Trading of Exploration and Exploitation.

[Masand et al. '92] B. Masand, G. Linoff, D. Waltz: *Classifying News Stories using Memory Based Reasoning* (SIGIR 1992).

[Mikolov et al. '13] T. Mikolov, K. Chen, G. Corrado, J. Dean: *Efficient Estimation of Word Representations in Vector Space* (arXiv 2013).

[Ng '15] A. Ng: CS229 Lecture notes: Part VI: Learning Theory, retrieved from http://cs229.stanford.edu/notes/cs229-notes4.pdf on September 3, 2015.

[Precup et al. '00] D. Precup, R. S. Sutton, S. Singh: *Eligibility Traces for Off-Policy Policy Evaluation* (ICML 2000).

[Radlinski et al. '08] F. Radlinski, M. Kurup, & T. Joachims: *How does clickthrough data reflect retrieval quality?* (CIKM 2008).

[Russo & Van Roy '14] D. Russo & B. Van Roy: *An Information-Theoretic Analysis of Thompson Sampling*. JMLR pre-print (to appear).

References and further reading

[Russo & Van Roy '14] D. Russo & B. Van Roy: *An Information-Theoretic Analysis of Thompson Sampling*. JMLR pre-print (to appear).

[Sloan & Wang '12] M. Sloan, J. Wang: *Dynamical Information Retrieval Modelling: A Portfolio-Armed Bandit Machine Approach*. WWW' 2012.

[Schuth et al. '15] A. Schuth, K. Hofmann, & F. Radlinski: *Predicting Search Satisfaction Metrics with Interleaved Comparisons*. SIGIR 2015.

[Sutton & Barto '98] R.S. Sutton & A.G. Barto: *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Tenenbaum et al. '11] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, N. D. Goodman: *How to Grow a Mind: Statistics, Structure, and Abstraction*. Science 331 (1279) 2011.

[Thompson '33] W. R. Thompson: *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*. Biometrika, 25(3–4):285–294, 1933.

[Yang et al. '14] H. Yang, M. Sloan, J. Wang: *Dyanmic Information Retrieval Modeling*. SIGIR 2014..

[Yue & Joachims '09] Y. Yue & T. Joachims: *Interactively optimizing information retrieval system as a dueling bandits problem* (ICML 2009).

Related Tutorials

[WWW 2008, SIGIR 2008, & WWW 2009] T. Liu: Tutorial on Learning to Rank for Information Retrieval – http://research.Microsoft.com/en-us/um/Beijing/projects/letor/tutorial.aspx

[SIGIR 2008] R Jin & Y. Zhang: Machine Learning for Information Retrieval - http://www.cse.msu.edu/~rongjin/sigir08-ml-tutorial.pdf

[RUSSIR 2014] K. Hofmann: Online Experimentation for Information Retrieval - http://1drv.ms/1yYLNlp

[SIGIR 2014 & WSDM 2015] G. H. Yang, M. Sloan, J. Wang: Dynamic Information Retrieval Modeling – http://www.dynamic-ir-modelling.org

[WSDM 2015] L. Li: Offline Evaluation and Optimization for Interactive Systems: A Practical Guide - http://research.microsoft.com/pubs/240388/tutorial.pdf