

Dynamic Ranked Retrieval

Christina Brandt, Thorsten Joachims,
Yisong Yue, Jacob Bank

Department of Computer Science
Cornell University

Handling Query Ambiguity

Max DCG

svm - Surf Canyon Se... x

file:///Y:/doc/wsdm11-interrank/dynamic_ranking/wsdm

Query: svm Search Results 1 to 10 of 654,000 for svm

[Support vector machine - Wikipedia, the free encyclopedia](#)
Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Since an SVM is a classifier, then given a set of training ...
en.wikipedia.org/wiki/Support_vector_machine

[SVM-Light Support Vector Machine](#)
Overview. SVM light is an implementation of Support Vector Machines (SVMs) in C. The main features of the program are the following: fast optimization algorithm
www.cs.cornell.edu/People/tj/svm_light

[LIBSVM -- A Library for Support Vector Machines](#)
An integrated and easy-to-use tool for support vector classification and regression
www.csie.ntu.edu.tw/~cjlin/libsvm

[SVM-Struct Support Vector Machine for Complex Outputs](#)
Overview. SVM struct is a Support Vector Machine (SVM) algorithm for predicting multivariate or structured outputs. It performs supervised learning by approximating
www.cs.cornell.edu/People/tj/svm_light/svm_struct.html

[SVM-Light FAQ](#)
Overview. SVMlight is an implementation of Vapnik's Support Vector Machine [Vapnik,...]. If the system does not compile properly, check this FAQ.
www.cs.cornell.edu/People/tj/svm_light/faq.html

[SVM and Kernel Methods Matlab toolbox](#)
Overview. Yes, this is another SVM Toolbox but the thing new is that it is fully written in Matlab (even the QP solver).
asi.insa-rouen.fr/~arakotom/toolbox/index.html

Template.docx Show all downloads...



Diversified

svm - Surf Canyon Se... x

file:///Y:/doc/wsdm11-interrank/dynamic_ranking/wsdm

Query: svm Search Results 1 to 10 of 654,000 for svm

[Support vector machine - Wikipedia, the free encyclopedia](#)
Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Since an SVM is a classifier, then given a set of training ...
en.wikipedia.org/wiki/Support_vector_machine

[ServiceMaster -- We Are Home](#)
ServiceMaster Legal Notice | ServiceMaster Privacy Statement © 2001–2010 The ServiceMaster Company. All Rights Reserved.
www.svm.com

[School of Volunteering Management | School of Volunteer Management ...](#)
School of Volunteering Management. The role of the School of Volunteer Management (SVM) is to enhance the development of volunteering in the community by assisting organisations ...
www.svm.edu.au

[Society for Vascular Medicine : About SVM : Home](#)
Mark Your Calendar. SVM 22nd Scientific Sessions June 2-4, 2011 Seaport Hotel Boston, Mass. Thanks to all who made the 2010 SVM Annual Meeting a huge success.
svmb.org

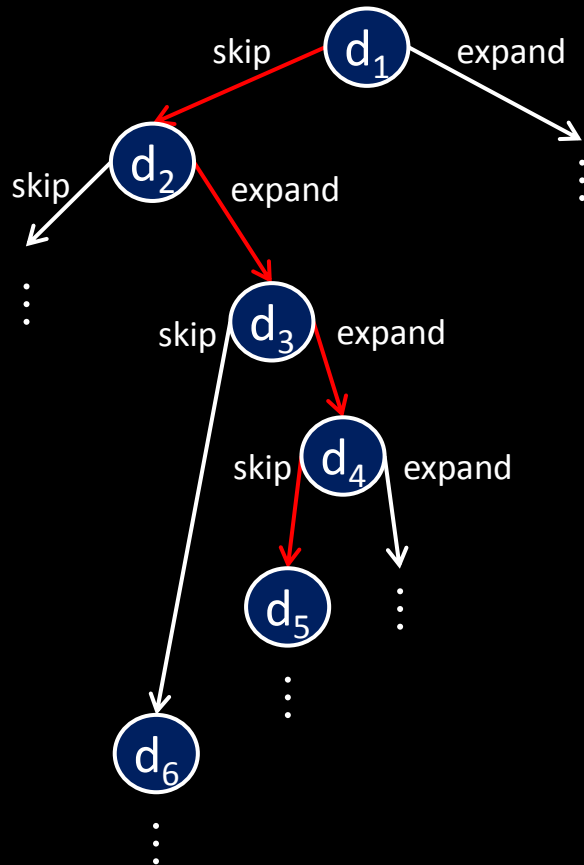
[SVM E-Business Solutions - Online Marketing. Bottom-line Results.](#)
SVM helps B-to-B companies leverage the Web to increase sales, strengthen relationships with customers and measure the return on marketing investments
svmsolutions.com

[SVM \(company\) - Wikipedia, the free encyclopedia](#)
SVM LP began as Stored Value Marketing in 1997. SVM is based in the Des Plaines, Illinois and provider of gasoline, restaurant, and other retail gift cards (also known as ...
[en.wikipedia.org/wiki/SVM_\(company\)](http://en.wikipedia.org/wiki/SVM_(company))

Template.docx Show all downloads...



Dynamic Ranking



svm - Surf Canyon Search

file:///C:/Users/tj/Documents/changed/dynamic_rankin

Query: svm Search Results 1 to 10 of 654,000 for svm

[ServiceMaster -- We Are Home](#)

ServiceMaster Privacy Notice | ServiceMaster Privacy Statement © 2001–2010 The ServiceMaster Company. All Rights Reserved.
www.svm.com

[Support vector machine - Wikipedia, the free encyclopedia](#)

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Since an SVM is a classifier, then given a set of training ...
en.wikipedia.org/wiki/Support_vector_machine

[SVM-Light Support Vector Machine](#)

Overview. SVM-Light is an implementation of Support Vector Machines (SVMs) in C. The main features of the program are the following: fast optimization algorithm
www.cs.cornell.edu/people/tj/svm_light

[LIBSVM -- A Library for Support Vector Machines](#)

An integrated software package for support vector classification and regression
www.csie.ntu.edu.tw/~cjlin/libsvm

[SVM-Struct Support Vector Machine for Complex Outputs](#)

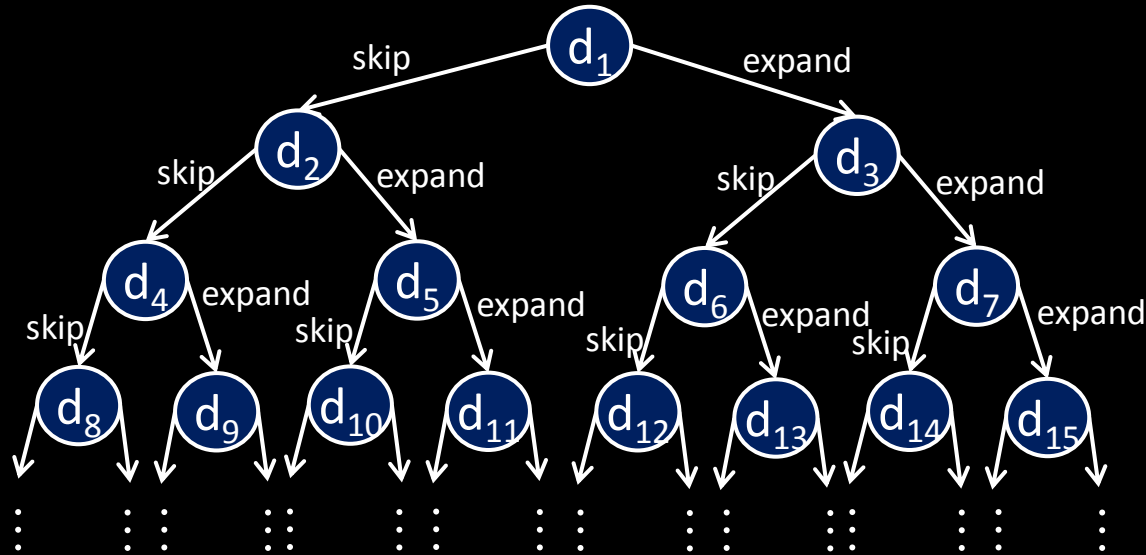
Overview. SVM-Struct is a Support Vector Machine (SVM) algorithm for predicting multivariate or structured outputs. It performs supervised learning by approximating
www.cs.cornell.edu/people/tj/svm_light/svm_struct.html

[\(more results\)](#)

[SVM Tutorials](#)

Support vector machines tutorials ... SVM Tutorials Best Tutorials. Introductory: HEARST, Marti A., et al., 1998.

Dynamic Ranking: Model

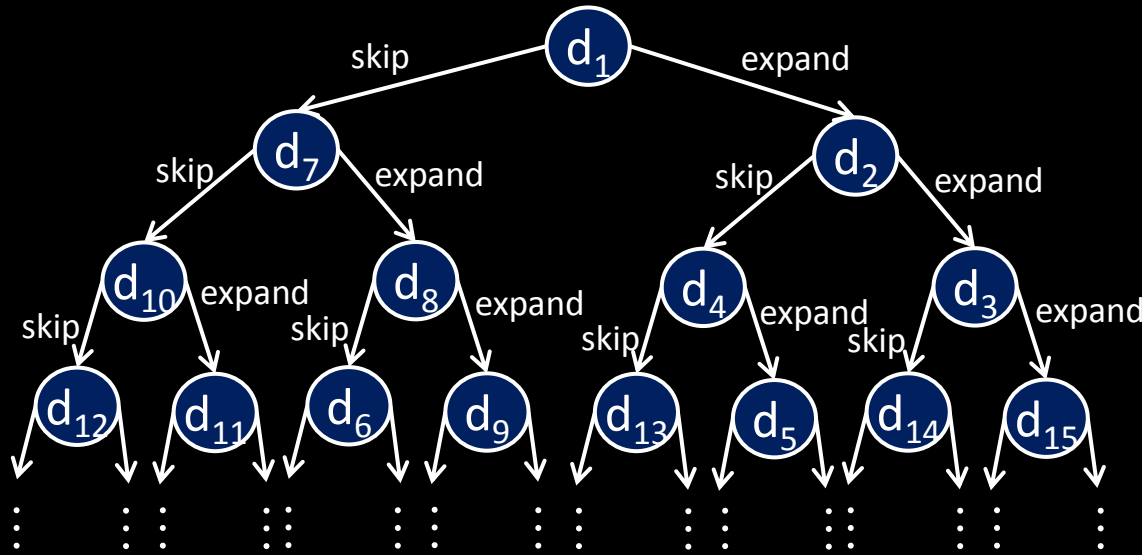


- Users: $u_1 \dots u_k$ with $P(u)$
- User behavior: $\pi: u \in \Psi \rightarrow \sigma$
 \rightarrow paths: $\sigma_1 \dots \sigma_k$
- Utility of σ for user u : $U(u, \sigma)$
- Overall utility: $U(\Psi) = \sum_u P(u) U(u, \pi(u, \Psi))$

Your favorite IR measure

“Intent-Aware” Measure [Agrawal et al. 2009]

Example



U(u,d)	u ₁	u ₂	u ₃	u ₄	u ₅
d ₁	1	1			
d ₂	1				
d ₃	1				
d ₄		1			
d ₅		1			
d ₆			1		
d ₇			1	1	
d ₈				1	
d ₉				1	
d ₁₀					1
d ₁₁					1
d ₁₂					

Measure: $U(u, \sigma) = \sum_i \frac{1}{\log(i+1)} U(u, \sigma[i])$

User behavior π : Expand if doc is relevant, else skip.

- **User 1**: $U(u_1, \sigma) = 2.13$
- **User 2**: $U(u_2, \sigma) = 1.93$
- **User 3**: $U(u_3, \sigma) = 1.06$
- **User 4**: $U(u_4, \sigma) = 1.56$
- **User 5**: $U(u_5, \sigma) = 0.93$



$U(\Psi) = 1.52$

Best static ranking σ^* : $U(\sigma^*) = 0.84$

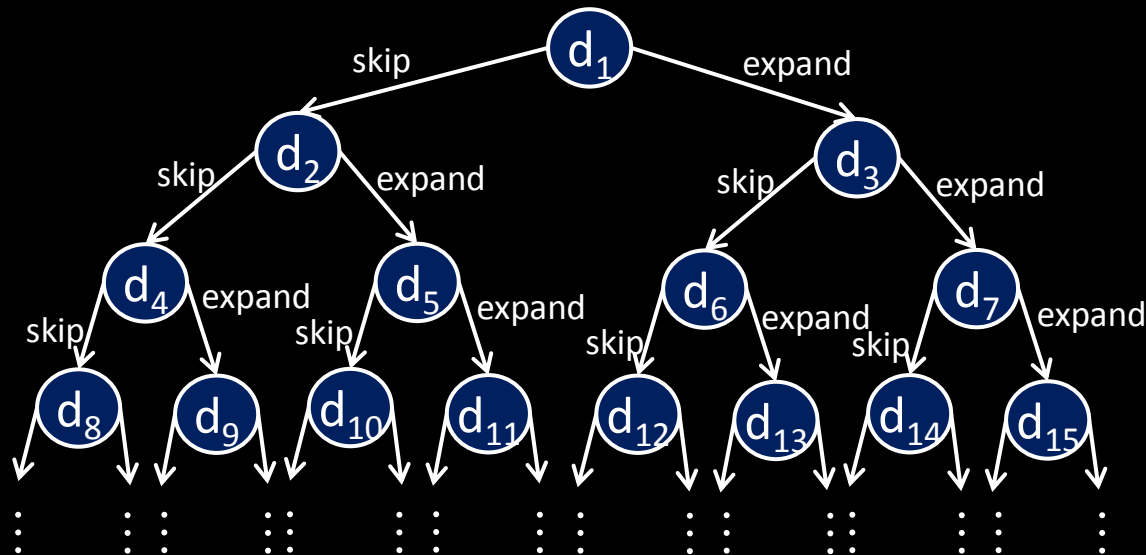
Assumption

- *Definition: Modular Utility Functions*

A utility function $U(u, \sigma)$ is modular, if

$$U(u, \sigma) = \sum_i \gamma[i] U(u, \sigma[i])$$

with $\gamma[i] \geq 0$ and decreasing, and $U(u, \sigma[i]) \geq 0$.
(e.g. DCG, nDCG, Prec@k)



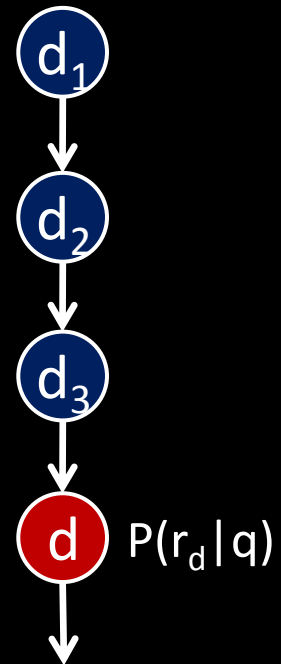
Algorithm for Static Ranking

- *Algorithm 3: StatMyopic($q, D, P(u)$)*
 - sort documents by expected doc utility
 - $P(r_d | q) = \sum_u P(r_d | q, u) P(u)$

- *Theorem: Optimality of StaticMyopic*

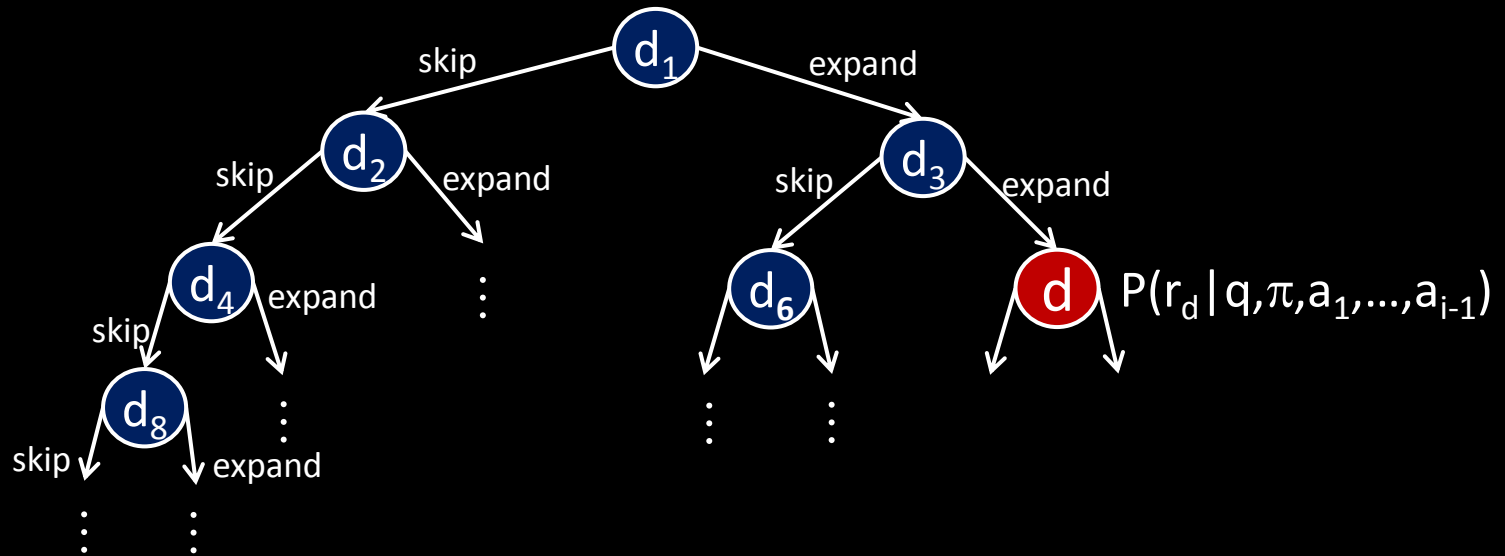
For modular utility functions, StatMyopic is optimal for constructing static rankings.

... but it is **not** optimal for general utility functions, for example Average Precision.



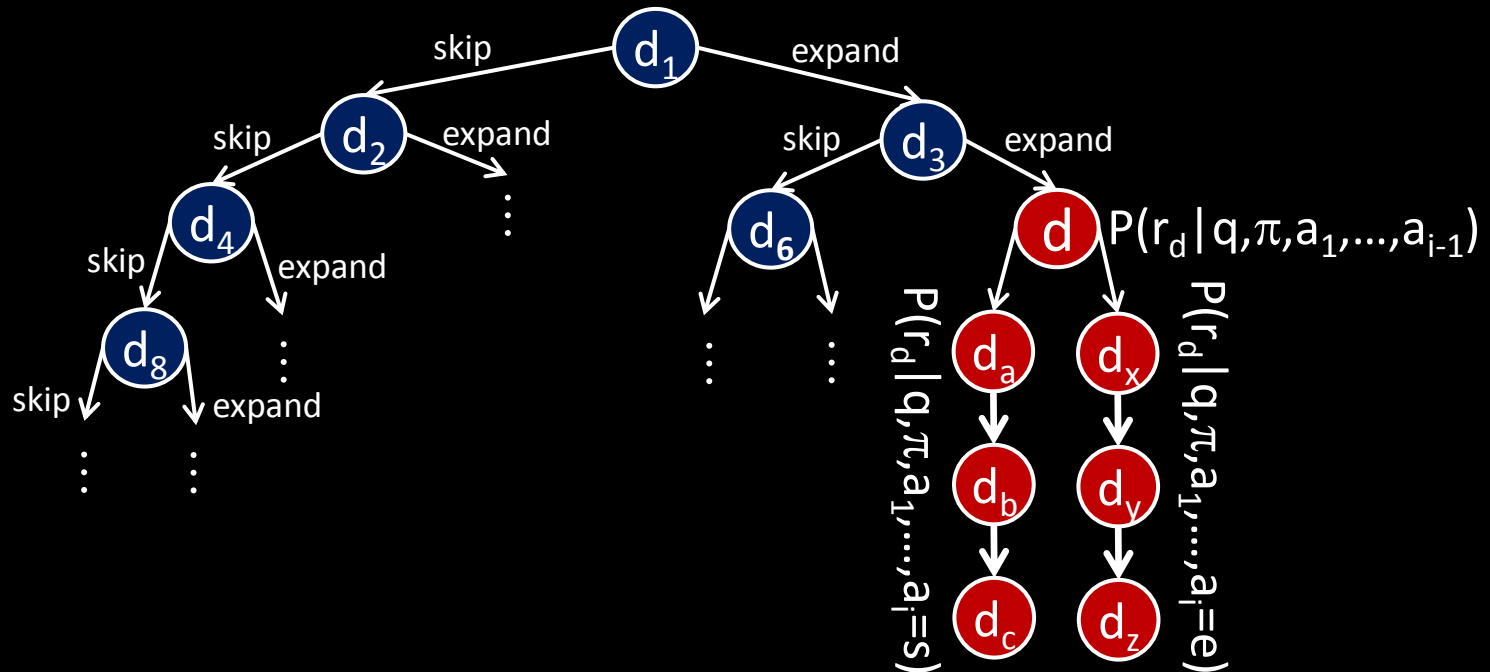
Algorithms for Dynamic Ranking

- *Algorithm 1: DynMyopic($q, D, \pi, P(u)$)*
 - Select d to maximize utility for user distribution that is conditioned on past actions
 - recurse on left and right subtree with user distribution updated according to π



Algorithms for Dynamic Ranking

- *Algorithm 2: DynLookahead($q, D, \pi, P(u)$)*
 - Select d to maximize conditional utility plus utility of static ranking for left and right subtree
 - recurse on left and right subtree with user distribution updated according to π



Adaptivity Gain

- *Definition: Adaptivity Gain*

The adaptivity gain of a dynamic ranking algorithm A is

$$U(A) - U(\text{StatMyopic}).$$

- *Theorem: Non-Negative Adaptivity Gain*

For modular utility functions and any (known) user behavior policy π , both DynMyopic and DynLookahead have a non-negative adaptivity gain.

... it is possible to construct examples where adaptivity gain is 0, but where the best possible adaptivity gain > 0 .

Related Work

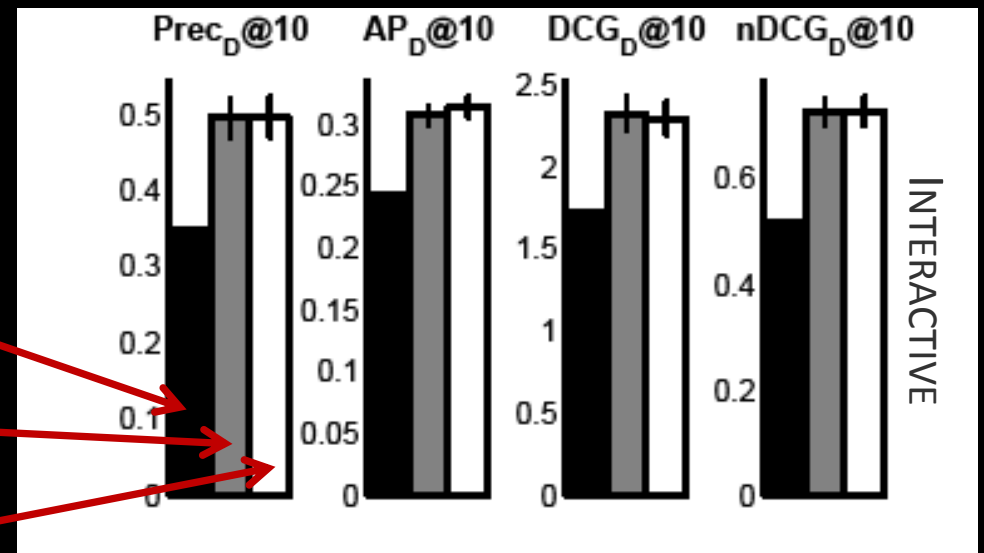
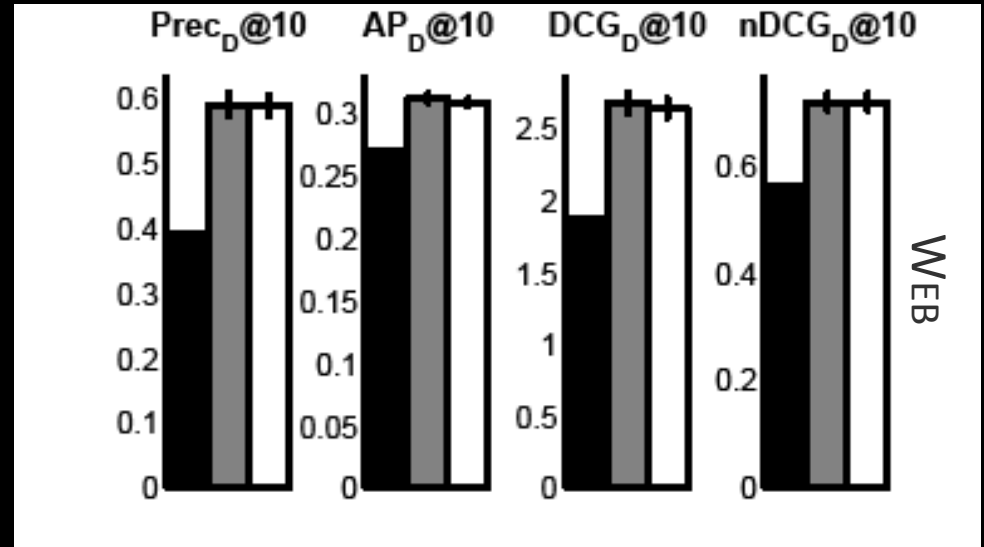
- Interactive Information Retrieval
 - Interface like SurfCanyon.com [Cramer et al., 2009]
 - Simple interaction, like menu browsing
 - Decision-theoretic model [Fuhr, 2008]
- Relevance Feedback
 - Use feedback to refine query, see [Ruthven & Lalmas, 2003]
 - Integrating feedback and result presentation; users “keep state”
 - Exploration, not just exploitation
- Diversified Retrieval
 - Novelty vs. relevance [Carbonell & Goldstein, 1998] [Zhai et al., 2003]
 - Left branch of DynMyopic similar to [Chen & Karger, 2006]

Experiments

- Data:
 - TREC 18 Web Track (WEB)
 - 50 queries
 - 2-8 profiles (i.e. user types) for each query.
 - $P(u)$ uniform.
 - Binary relevance judgments
 - TREC 6-8 Interactive Track (INTERACTIVE)
 - 20 queries
 - 7-56 profiles (i.e. user types) for each query.
 - $P(u)$ proportional to number of relevant documents in profile.
 - Binary relevance judgments

How Large is the Adaptivity Gain?

- Relevance profiles
 - known to the algorithm
- User Behavior
 - π_{det} : users expand relevant docs, skip non-relevant docs.



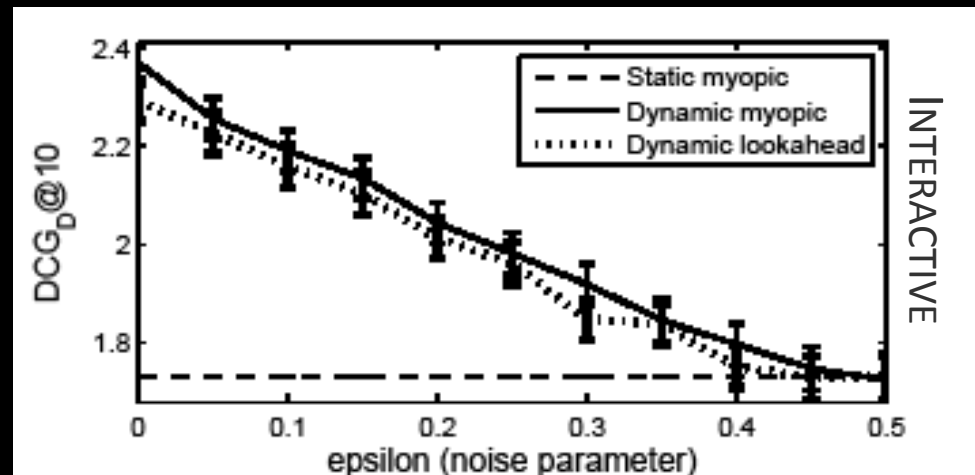
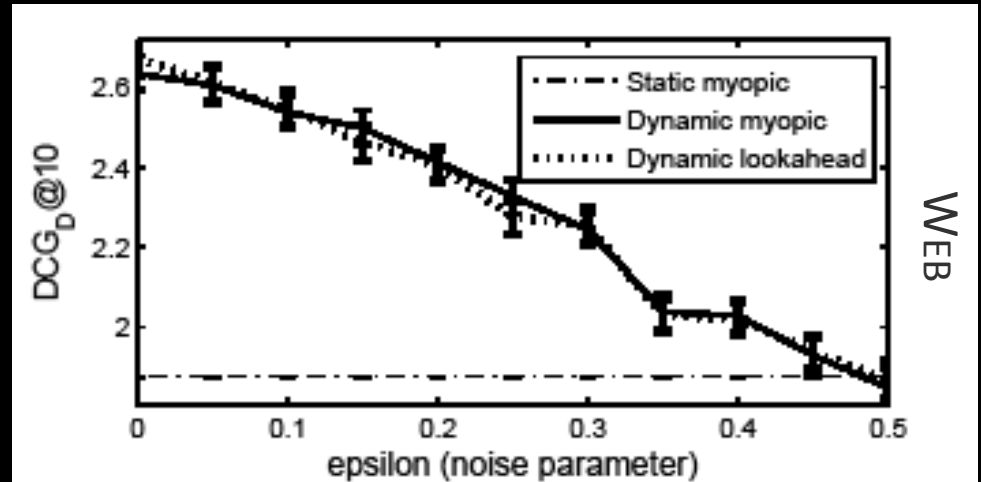
StatMyopic

DynMyopic

DynLookahead

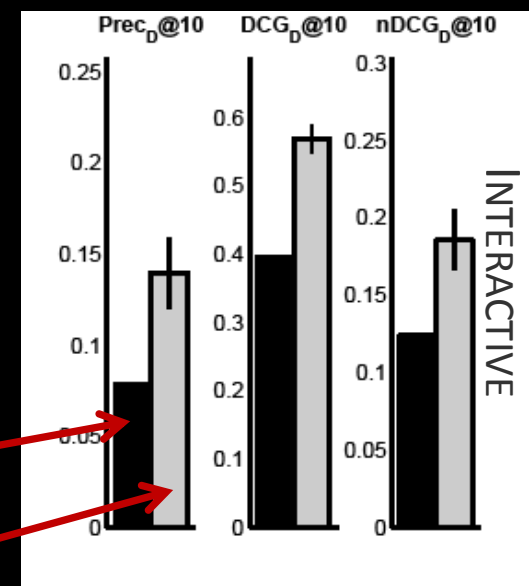
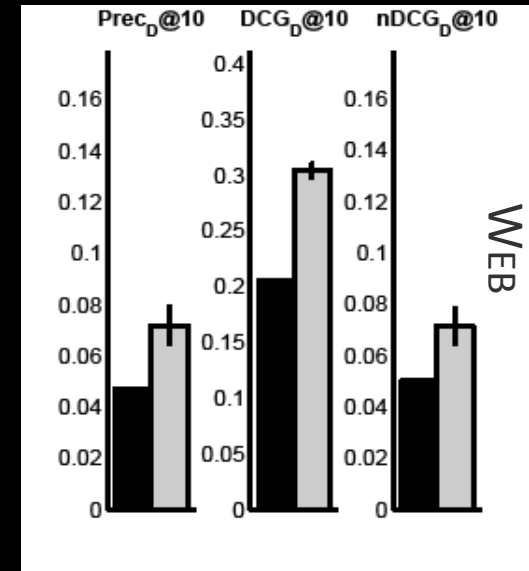
Noisy User Behavior

- Relevance profiles
 - known to the algorithm
- User Behavior
 - π_ε : users
 - skip relevant documents with probability ε ,
 - exp non-relevant documents with probability ε .



Learning the Relevance Function

- Relevance profiles
 - Learned $P(\text{rel} | d, q, \pi, a_1 \dots a_i)$
 - Logistic regression
 - Features relating doc and query
 - Features relating doc and expanded
 - Features relating doc and skipped
 - Training data from simulated users
 - π_{det} : deterministic users
 - Training examples: $((q, d, a_1 \dots a_i) \rightarrow \text{rel})$
 - Sampled $a_1 \dots a_i$ “appropriately”
- Leave-one-query-out cross validation



StatMyopic

DynMyopic

Conclusions

- Simple interactive ranked retrieval model
 - Combines diversity and recall
 - Evolution, not revolution of user interface; menu browsing
- Theoretical model
 - Provides evaluation measures
 - Guides algorithm design
- Algorithms
 - Two efficient algorithms with non-negative adaptivity gain
 - <http://dynamicranking.joachims.org>
- Directions for Research
 - Approximation algorithms for constructing dynamic ranking trees
 - Using click data for learning $P(r_d | q, \pi, a_1, \dots, a_{i-1})$
 - More sophisticated user behavior policies π
 - More actions than skip and expand
 - Usability (layout, clicking vs. mousing)

Directions for Research

- Approximation algorithms for constructing dynamic ranking trees
 - Approximation guarantees
 - Partial-information online learning problem
- Learning dynamic ranking models from real data
 - Using observational click data for learning $P(r_d | q, \pi, a_1, \dots, a_{i-1})$
- More sophisticated user behavior policies π
 - Asymmetric costs for skip and expand / limited depth
 - Verify policy against real user behavior
 - Policies that are not static throughout search
- More actions than skip and expand
 - Backing out of tree
 - Reformulation recommendations
- Usability
 - Result layout
 - Clicking vs. mousing