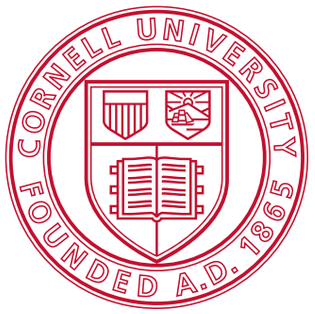


Cornell University
Computer Science

Learning to Rank (part 2)

NESCAI 2008 Tutorial

Filip Radlinski
Cornell University

Recap of Part 1

1. Learning to rank is widely used for information retrieval, and by web search engines.

2. There are many measures for performance evaluation. Two of the most common are MAP and NDCG.

3. You can use machine learning to optimize performance for these measures.

4. There are many algorithms for learning to rank out there.

Plan for Part 2

- Training Data: Thinking about the people in the loop.
  - The making of standard training datasets.
  - Using user behavior to collect training data.
  - Evaluating for user behavioral metrics.

- Diversity: Modeling dependencies between the labels.
  - Performance isn't just the sum of the parts.
  - Diversifying rankings we've already learned.
  - Learning to rank with diversity in mind.

- Conclusions

Getting the training data

- Everything Yisong said is all great, but where does the training data come from?
- This is something Machine Learning people rarely think about: We just download the data. To learn generalizable ranking functions, we must.
- When learning to rank, data usually comes from one of two places: "expert" human judges, or users:
  - experts: TREC data & Web search data.
  - users: Logs of user behavior.

(1) old-style TREC data

Topic Number	503
Title	Vikings in Scotland?
Description	What hard evidence proves that the Vikings visited or lived in Scotland?
Narrative	A document that merely states that the vikings visited or lived in Scotland is not relevant. A relevant document must mention the source of the information, such as relics, sagas, runes or other records of those times.

- Evaluated systems each contribute top N results.
- Human judges rate documents as:
  not relevant, *relevant* or *highly relevant*.
- Topics, documents and scores are made public.

(1) old-style TREC data

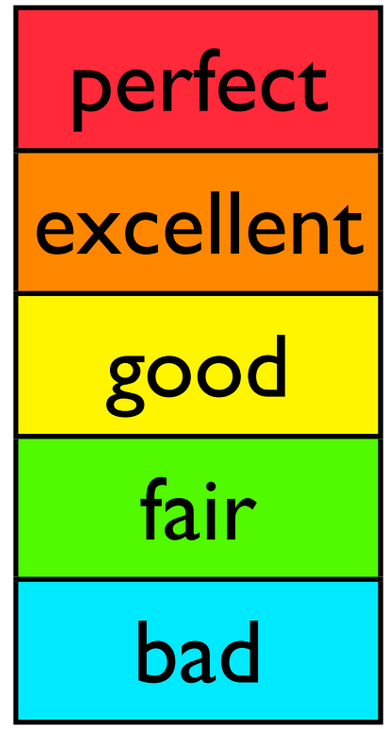
- Pros:
  - Get exhaustively labeled (query, document) pairs.
  - We know the ground truth, definitively.
  - Static, reproducible and understandable labels.
  - The data is easy to download and just use.
- Cons:
  - Queries not designed to reflect real needs.
  - Detail in queries not representative.
  - Coarse-grained relevance judgments.

(2) Web search data

- TREC anticipates user needs. Search engines really want to taylor performance to real queries.
- Judge are given real web queries:
  - Often just 1, 2 or maybe 3 words long.
- Typically documents are judged on a four to six point ordinal scale:
  - bad, fair, good, excellent, perfect. [Carterette & Jones, ECIR '08]
- Judge are given real web documents:
  - The content is clearly key.
  - Presentation plays a role too.
  - The URL / authority is important too.

Pretending to be a judge

- First query: “DCG”:
  - Disability Consulting Group?
  - Department of Charitable Gaming?
  - Discounted Cumulative Gain?
  - Acronym finder!
- Second query: “support vector machine”
  - Tutorials about SVMs.
  - Downloadable SVM implementations.
  - Theoretical proofs showing why SVMs work.
  - Variants like Ranking SVMs.



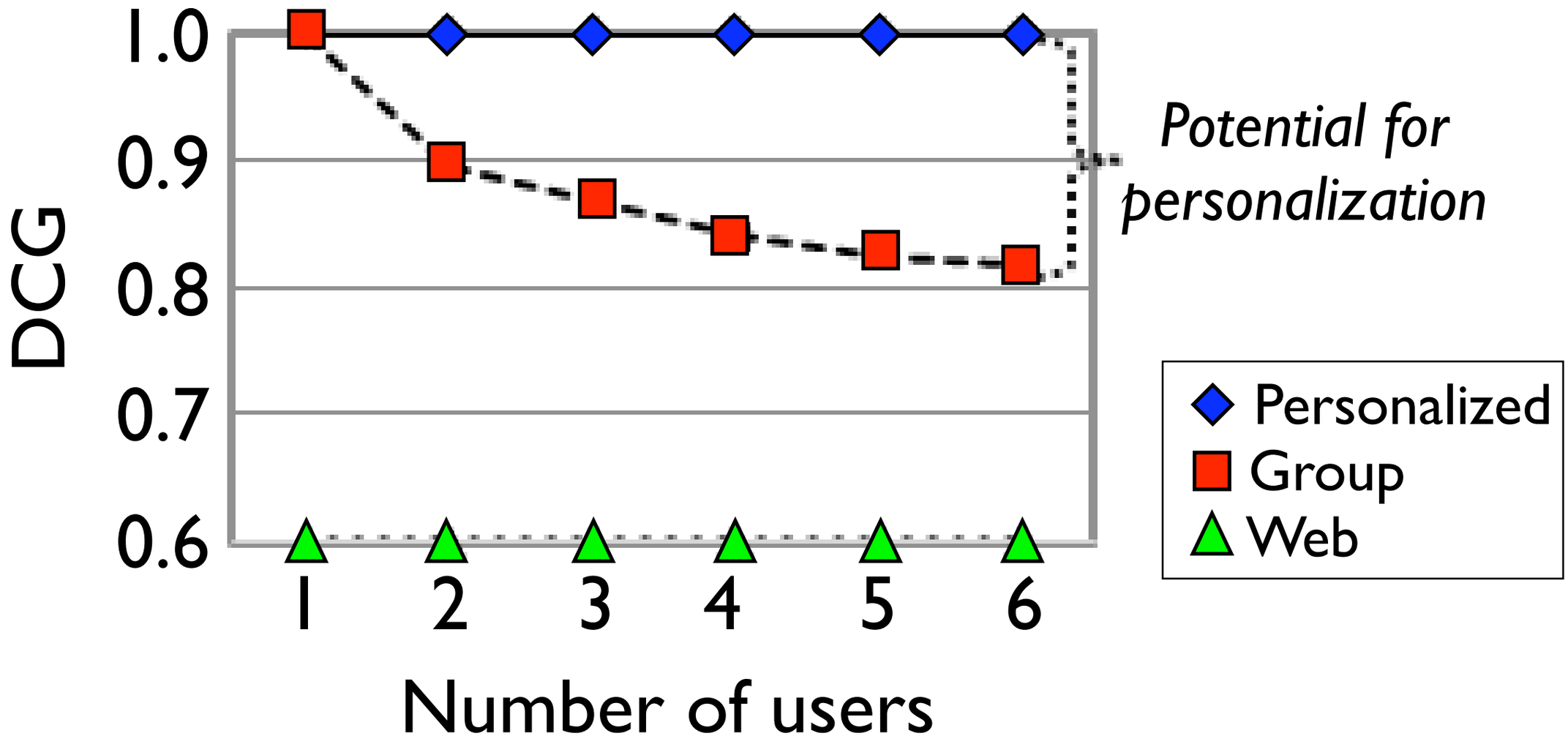
Web search data: pros & cons

- We can't make exhaustive judgments.
  - Which web queries do we make them for?
  - Which docs do we judge? (top or lower ones?)
- We need to constantly update judgments.
- How do we know the actual intent?
  - Queries are too short to be unambiguous.
  - Studying distribution of meanings impractical.
- How should we trade off content, source, presentation?

+ The data reflects what users really search for.

A con of (1) and (2)

- Different people may have different things in mind for the same query. [Teevan et al, SIGIR '07]
- Why should they all get the same search results?



A con of (1) and (2)

- Different people may have different things in mind for the same query. [Teevan et al, SIGIR '07]
- Why should they all get the same search results?
- To fix this, we need to do one of two things:

1. Encode user profile in ranking function. For example, rerank specifically to each user. [Teevan et al, SIGIR '05]
   - Q: What features are best?

2. Get relevance judgments from users, so that we can learn a ranking for specific subgroups of users. [Radlinski & Joachims, KDD '05]
   - Q: Can we cluster users? Share knowledge?

(3) Ask users for labeled data

- We can get training data directly from users.
  - *Explicitly* ask users what the labels are.
- Pros:
  - Capture users' intent exactly.
  - Data reflects real needs users actually had.
  - Don't need to hire judges.
- Cons:
  - In some settings, it's just annoying and few people bother.
  - If ranking for web search, spammers have a real incentive to provide noise.

(4) Get data implicitly from users

- "*Implicitly*" means we just record behavior and draw inferences.
- What can we record like this?
  - Queries: queries sent to the search engine.
  - Clicks on results: what was clicked.
  - Dwell time: When clicks & queries happened.
  - Browser buttons, printing, bookmarking, mousing: User interactions with web browser.
  - Reformulations: The whole sequence of user actions.

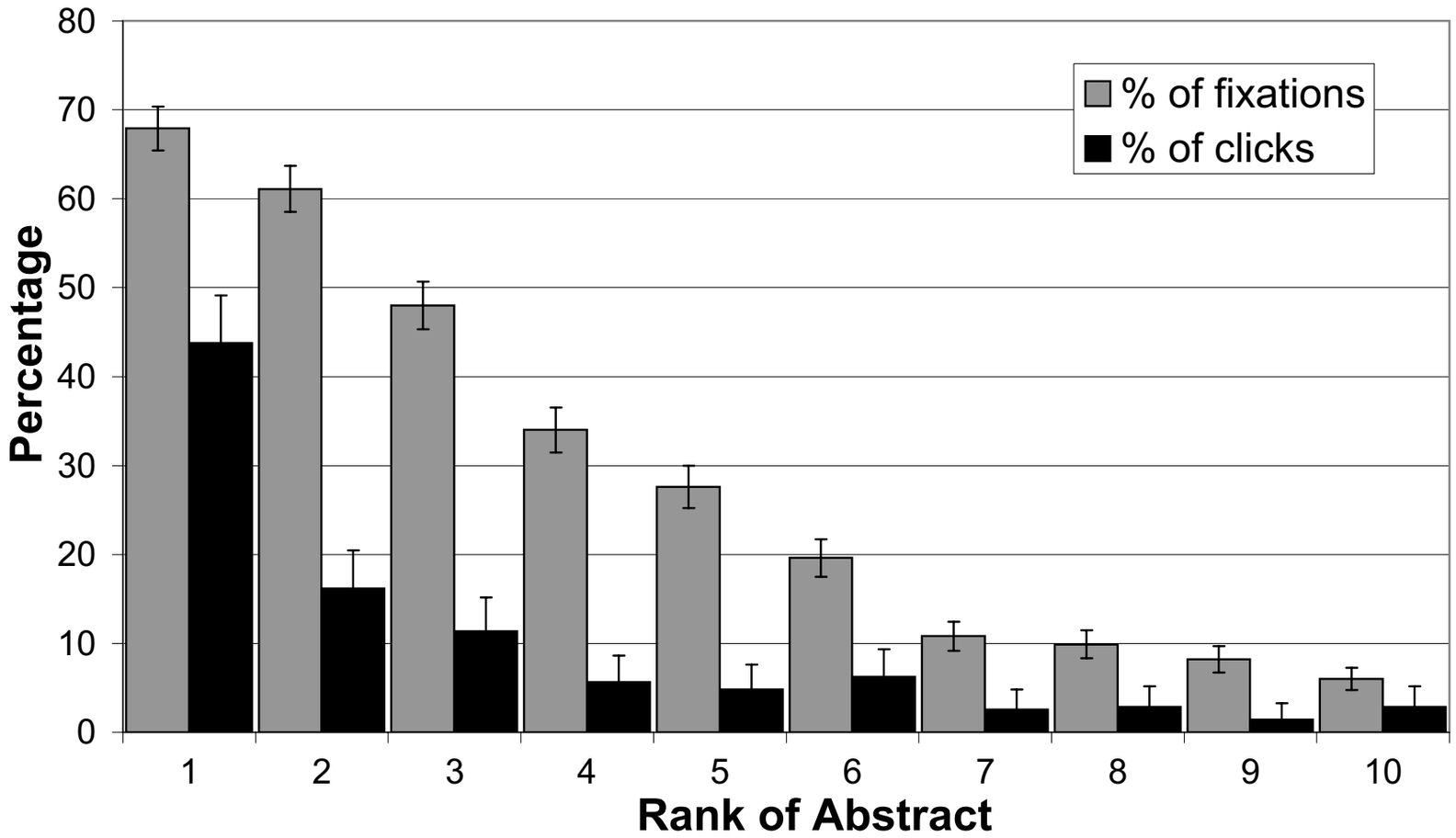
[Kelly & Teevan, SIGIR Forum '03; Fox et al, TOIS '05; White et al, SIGIR '05]

Implicit feedback: pros & cons

- Pros:
  - Easy to get lots and lots of data (try building your own search proxy!)
  - Reflects real user needs & natural interactions.
  - Up to date with users, queries and documents.
- Cons:
  - Some measures need instrumented browsers.
  - Not repeatable - your users, queries change.
  - Data can't be shared due to privacy concerns.
  - We need to understand users' decision process to make valid inferences.
  - Spammers & crawlers can pollute it easily.

How do users search?

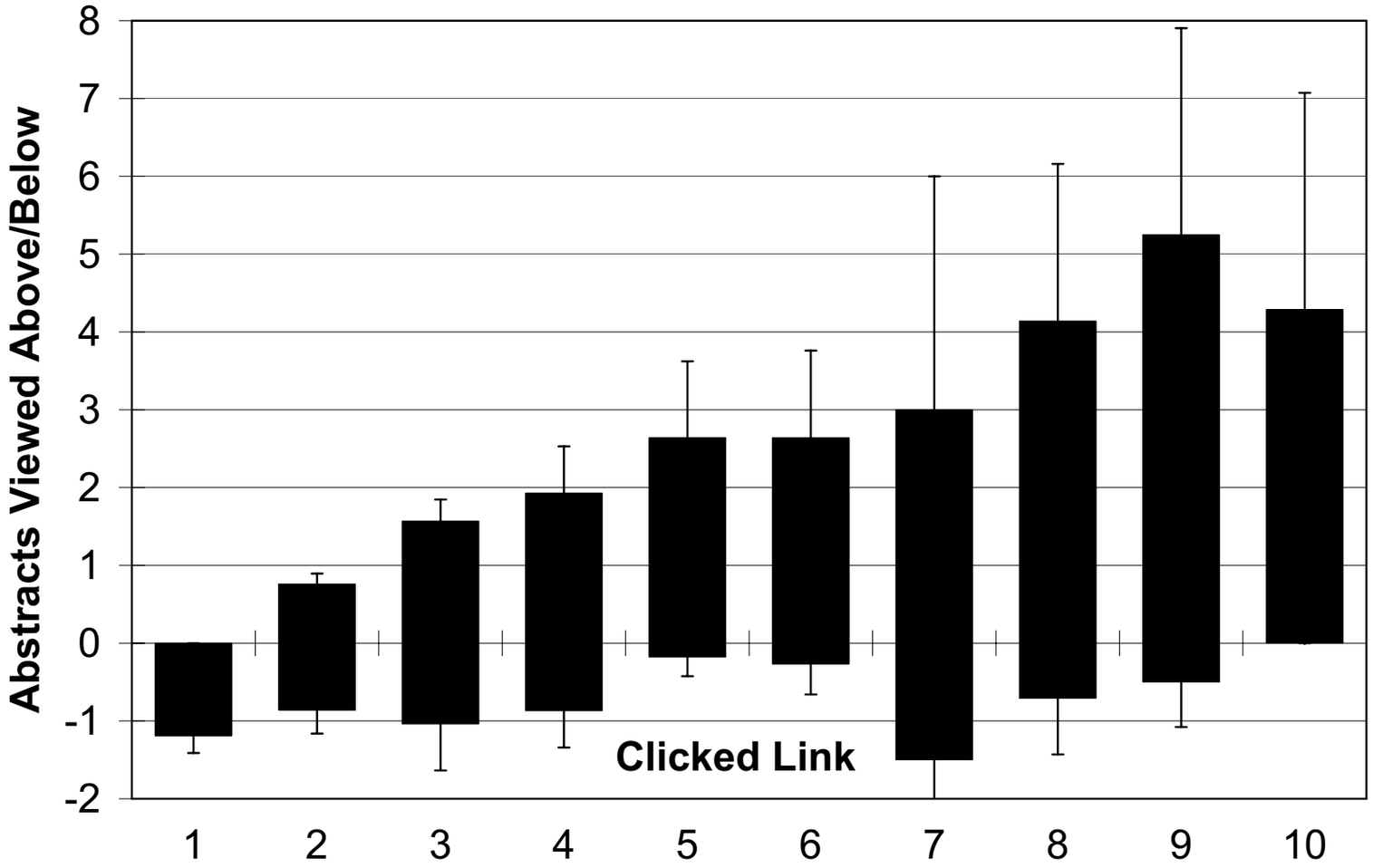
- If we want to use implicit feedback, lets first understand how users behave when searching.



- They look at only a few results per query.
- They click mostly at the top.

[Joachims, Granka et al, SIGIR '05]

How do users search?

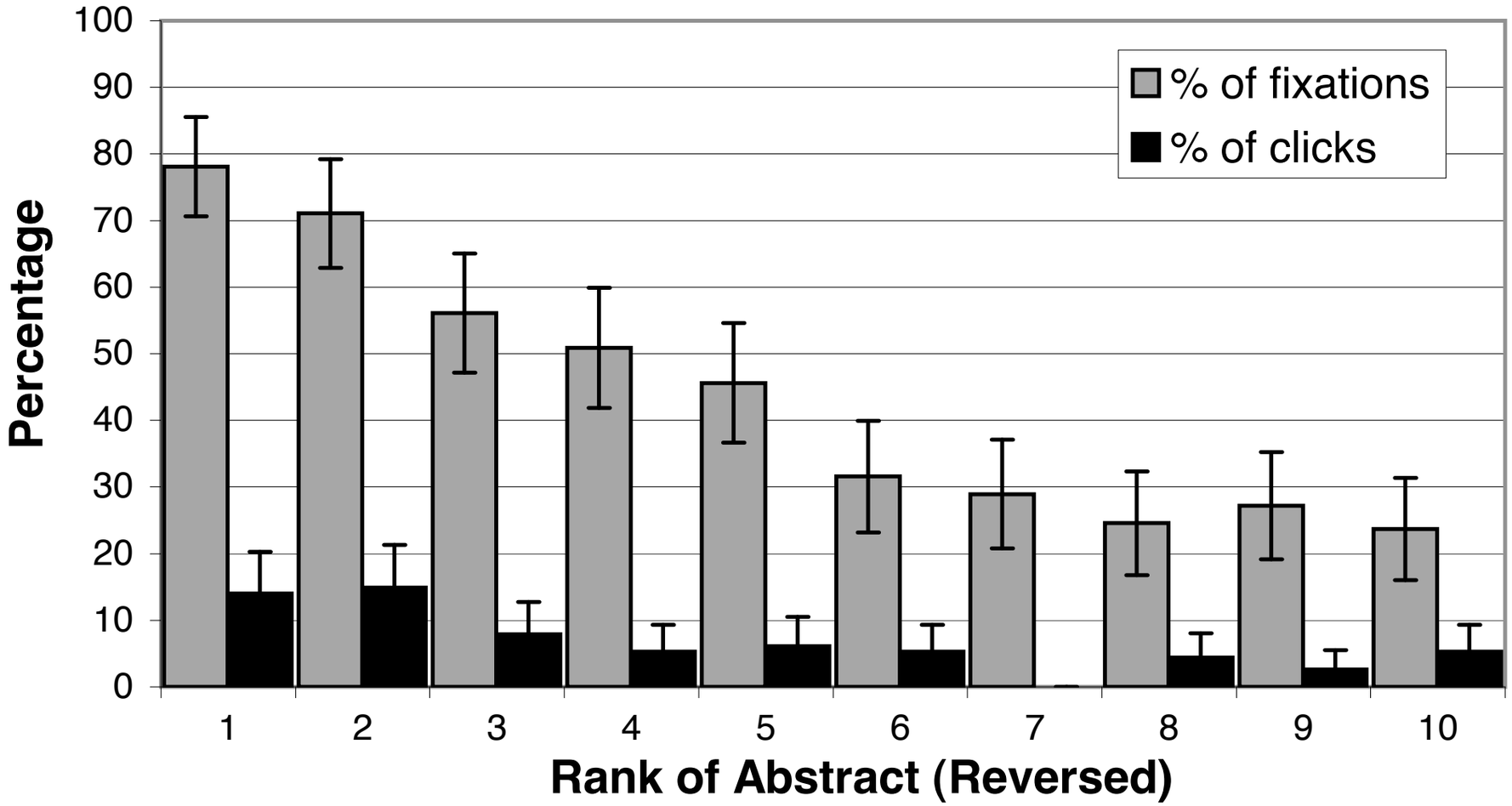


- They look at results mostly in order.
- They look at most of the results before a click
- ... and usually one result after.

[Joachims, Granka et al, SIGIR '05]

How do users search?

- Say you present the top 10 results *reversed*.
- Users still click more on the top results, even though those results are less relevant.



[Joachims, Granka et al, SIGIR '05]

What *do* clicks mean anyway?

- Two ways to interpret user behavior:

Absolute

Rank	Website
1	SVM-Light Support Vector Machine
2	Kernel Machines
3	SVM Application List
4	SVM - L'essentiel de la micro et ...
5	Support Vector Machines - The Book
6	Support Vector Machine

Relevant(d_1)

NotRelevant(d_2)

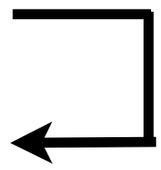
NotRelevant(d_3)

NotRelevant(d_4)

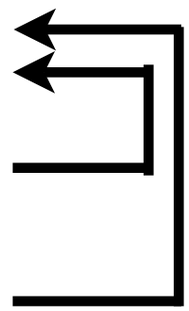
Relevant(d_5)

Relevant(d_6)

Preferences



$d_1 > d_2$

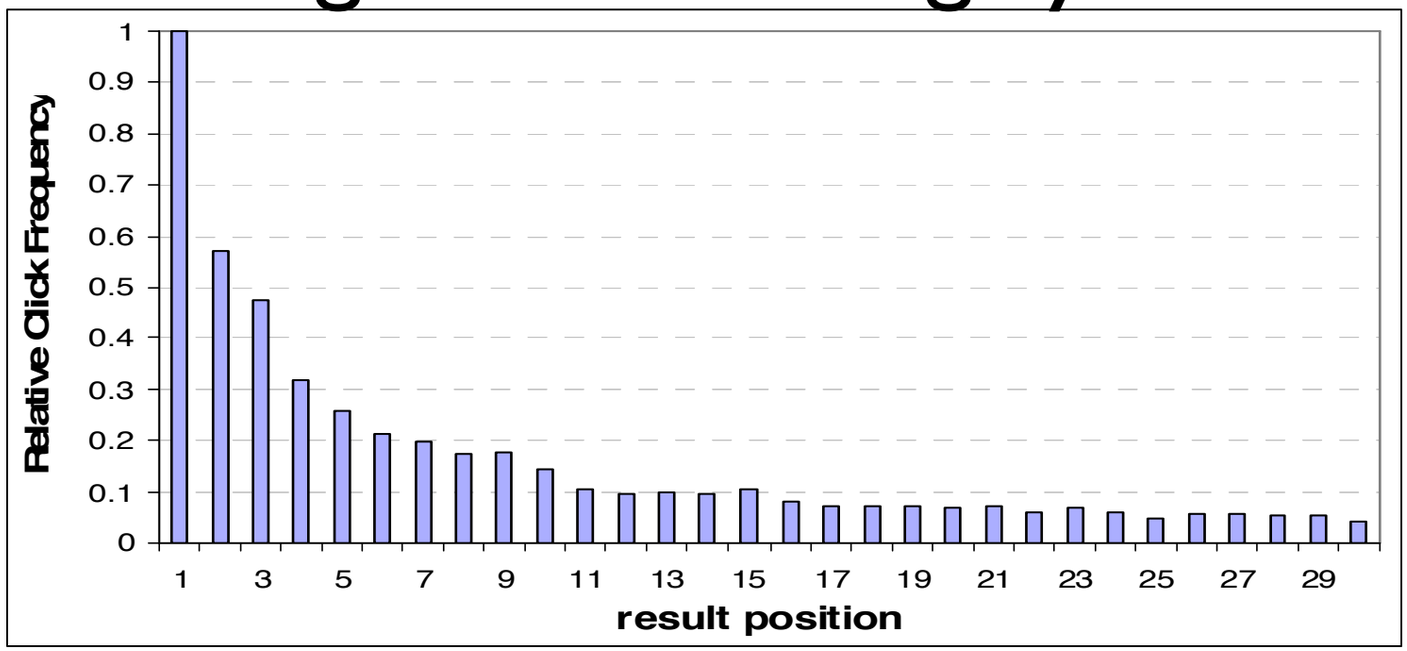


$d_5 > d_4$

$d_6 > d_4$

Option 1: Absolute data

- Clickthrough rates are highly skewed:



- But subtracting it out, we see a signal:

[Agichtein et al, SIGIR '06]

Option 1: Absolute data

- Clicking behavior can be modeled with a joint model: P(s,u,q,d)
  - **s**: Was the document selected?
  - **u**: Document URL
  - **q**: The query
  - **d**: Distance to previous selection
- Rewrite as P(s,a,c | u,q,d) = P(c | d) P(a | u,q)
  - **a**: Attractivity (relevance)
  - **c**: Consideration (did user look at it?)
- Model **a** and **c** using Bernoulli random variable, with a Beta prior, learning the parameters.

[Dupret et al, QLA@WWW '07]

Option 2: Preferences

- Like in part 1, a preference is a relative statement about how two documents relate:

$$\{x_i \succ x_j\} \text{ where } x_i, x_j \in X$$

- Unlike "expert" data, the judgments are possibly not to be transitive or consistent!
- Are people really making preferences?
  - Intuitively, a purchase is not saying that you've found the perfect item; just the best item available.
  - Asking expert judges for preferences gives more reliable data than asking for absolute judgments and inferring preferences. [Carterette et al, ECIR '08]

What preferences could we get?

- There are many possible strategies for interpreting clicks as pairwise preferences:

	Website	
1	SVM-Light Support Vector Machine	Click
2	Kernel Machines	
3	SVM Application List	
4	SVM - L'essentiel de la micro et ...	
5	Support Vector Machines - The Book	Click
6	Support Vector Machine	Click

Strategy	**Validity**
Inter-Judge Agreement	*86.4*
Click > Skip Above	78.2 ± 5.6
Last Click > Skip Above	80.9 ± 5.1
Click > Earlier Click	64.3 ± 15.4
Click > Skip Previous	80.7 ± 9.6
Click > No Click Next	67.4 ± 8.2

[Joachims et al, SIGIR '05]

Using Reformulations

- There is extra information in query reformulations.

Rank	Website
1	Ithaca Tompkins Regional Airport - Home
2	Ithaca Tompkins Regional Airport - Schedule
3	Ithaca Airport, Greece: Travel information

"ithaca airport"

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	Cheap Flights Ithaca - Discount Airfares
3	Ithaca Tompkins Regional Airport - The ...

"ithaca airline"

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	14850 Today: News for Ithaca, New York
3	14850 Today: Independence Air to serve ...

"new ithaca airline"

[Radlinski & Joachims, KDD '05]

Using Reformulations

Rank	Website
1	Ithaca Tompkins Regional Airport - Home
2	Ithaca Tompkins Regional Airport - Schedule
3	Ithaca Airport, Greece: Travel information

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	Cheap Flights Ithaca - Discount Airfares
3	Ithaca Tompkins Regional Airport - The ...

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	14850 Today: News for Ithaca, New York
3	14850 Today: Independence Air to serve ...

- Same idea as before: Click > Skip Above

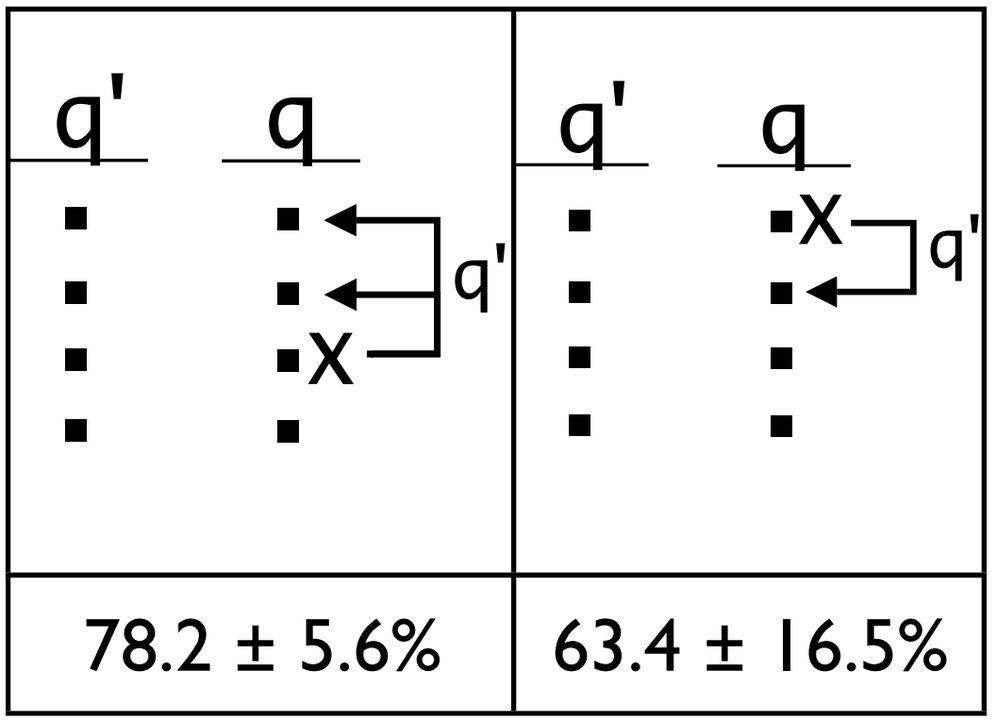
[Radlinski & Joachims, KDD '05]

Using Reformulations

Rank	Website
1	Ithaca Tompkins Regional Airport - Home
2	Ithaca Tompkins Regional Airport - Schedule
3	Ithaca Airport, Greece: Travel information

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	Cheap Flights Ithaca - Discount Airfares
3	Ithaca Tompkins Regional Airport - The ...

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	14850 Today: News for Ithaca, New York
3	14850 Today: Independence Air to serve ...



78.2 ± 5.6%	63.4 ± 16.5%

Inter-Judge Agreement: 86.4%

Baseline: 50.0%

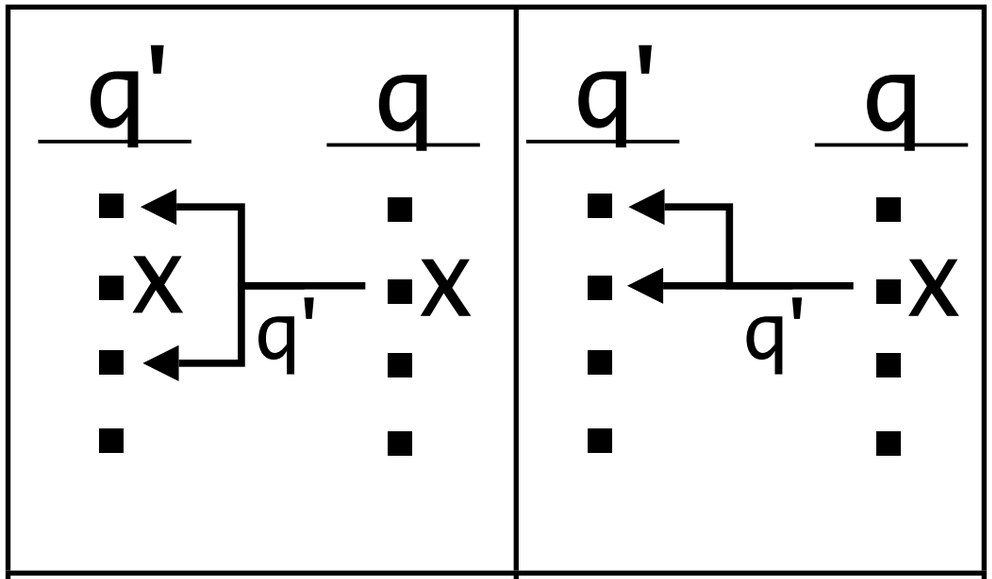
[Radlinski & Joachims, KDD '05]

Using Reformulations

Rank	Website
1	Ithaca Tompkins Regional Airport - Home
2	Ithaca Tompkins Regional Airport - Schedule
3	Ithaca Airport, Greece: Travel information

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	Cheap Flights Ithaca - Discount Airfares
3	Ithaca Tompkins Regional Airport - The ...

Rank	Website
1	Cornell Remote Sensing | Airline Service
2	14850 Today: News for Ithaca, New York
3	14850 Today: Independence Air to serve ...



68.0 ± 8.4%	84.5 ± 6.1%

Inter-Judge Agreement: 86.4%

Baseline: 50.0%

[Radlinski & Joachims, KDD '05]

Learning with pairwise prefs

- These preferences can be used to train any of the pairwise algorithms Yisong talked about.
- But it is very biased training data:
  - *Reversing* the ranking satisfies most constraints!

Strategy	**Validity**
Inter-Judge Agreement	*86.4*
Click > Skip Above	78.2 ± 5.6
Last Click > Skip Above	80.9 ± 5.1
Click > Earlier Click	64.3 ± 15.4
Click > Skip Previous	80.7 ± 9.6
Click > No Click Next	67.4 ± 8.2

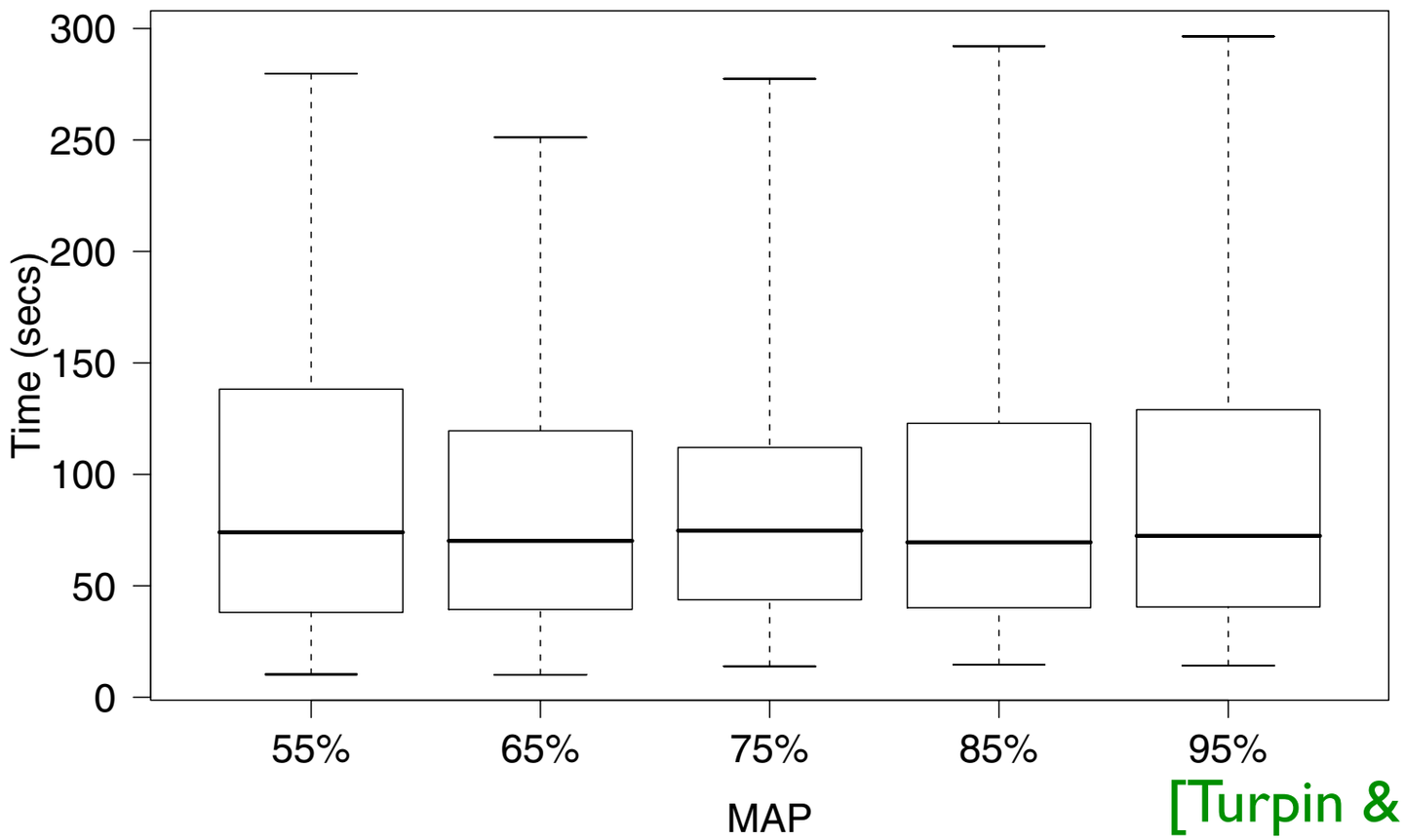
Learning with pairwise prefs

- These preferences can be used to train any of the pairwise algorithms Yisong talked about.
- But it is very biased training data:
  - *Reversing* the ranking satisfies most constraints!
- We need to introduce additional constraints to the learning problem:
  - Suppose we're learning a combination of a underlying ranking functions.
  - *Data properties* mean we must require that, e.g. all learned weights be positive.

[Radlinski & Joachims, KDD '05]

Evaluating performance

- If we had absolute relevance judgments and could measure MAP performance, it doesn't necessarily correlate well with user experience.
- Given rankings with different MAPs, users took the same time to find the first relevant document.



[Turpin & Scholer, SIGIR '06]

Evaluating performance

- If we had absolute relevance judgments and could measure MAP performance, it doesn't necessarily correlate well with user experience.
- Given rankings with different MAPs, users took the same time to find the first relevant document.
- But we don't have absolute judgments.

How good is my ranking?

- What can we measure based on user behavior?
  - Mean rank of clicks?

Function	bad	medium	good
Mean Rank	6.26 ± 1.14	6.18 ± 1.33	6.04 ± 0.92

[Boyan et al, IBIS@AAAI '96]

  - Reformulation rate?

Function	bad	medium	good
Mean Rank	0.24 ± 0.02	0.25 ± 0.02	0.25 ± 0.02

[Radlinski et al, in preparation]

  - Time to first click?

Function	bad	medium	good
Mean Rank	33.0 ± 4.0	30.7 ± 3.4	30.0 ± 4.0

[Radlinski et al, in preparation]

Evaluating with Preferences

Ranking Function 1

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsv
8. Royal Holloway Support Vector Machine
   http://svm.dvc.rhbnc.ac.uk/

Ranking Function 2

1. An Introduction to Support Vector Machines
   http://www.support-vector.net/
2. Kernel Machines
   http://svm.first.gmd.de/
3. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
4. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
5. Support Vector Machine
   http://jbolivar.freeservers.com/
6. Lagrangian Support Vector Machine Home Page
   …u/dmi/lsvm/
7. …ort Vector Machine
   …c.uk/
8. …ector Machine
   …rsten/svm_light/

Combined Ranking

1. Kernel Machines
   http://svm.first.gmd.de/
2. An Introduction to Support Vector Machines
   http://www.support-vector.net/
3. Support Vector Machine
   http://jbolivar.freeservers.com/
4. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
7. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
8. Lagrangian Support Vector Machine Home Page
   http://www.cs.wisc.edu/dmi/lsvm/

[Joachims, KDD '02]

Evaluating with Preferences

Ranking Function 1

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsv
8. Royal Holloway Support Vector Machine
   http://svm.dvc.rhbnc.ac.uk/

Ranking Function 2

1. An Introduction to Support Vector Machines
   http://www.support-vector.net/
2. Kernel Machines
   http://svm.first.gmd.de/
3. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
4. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
5. Support Vector Machine
   http://jbolivar.freeservers.com/
6. Lagrangian Support Vector Machine Home Page
   u/dmi/lsvm/
7. ort Vector Machine
   c.uk/
8. ector Machine
   rsten/svm_light/

Combined Ranking

1. Kernel Machines
   http://svm.first.gmd.de/
2. An Introduction to Support Vector Machines
   http://www.support-vector.net/
3. Support Vector Machine
   http://jbolivar.freeservers.com/
4. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
7. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
8. Lagrangian Support Vector Machine Home Page
   http://www.cs.wisc.edu/dmi/lsvm/

[Joachims, KDD '02]

Evaluating with Preferences

Ranking Function 1

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsv
8. Royal Holloway Support Vector Machine
   http://svm.dvc.rhbnc.ac.uk/

Click

Click

Ranking Function 2

1. An Introduction to Support Vector Machines
   http://www.support-vector.net/
2. Kernel Machines
   http://svm.first.gmd.de/
3. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
4. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
5. Support Vector Machine
   http://jbolivar.freeservers.com/
6. Lagrangian Support Vector Machine Home Page
   /dmi/lsvm/
7. Vector Machine
   .uk/
8. ector Machine
   rsten/svm_light/

Click

Combined Ranking

1. Kernel Machines
   http://svm.first.gmd.de/
2. An Introduction to Support Vector Machines
   http://www.support-vector.net/
3. Support Vector Machine
   http://jbolivar.freeservers.com/
4. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm_light/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
7. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
8. Lagrangian Support Vector Machine Home Page
   http://www.cs.wisc.edu/dmi/lsvm/

Click

Click

[Joachims, KDD '02]

A problem with the picture so far...

- Say we have a relevance score for every user, query and document.

	d1	d2	d3	d4	d5	d6	d7	d8	d9
User 1	✓				✓		✓	✓	
User 2		✓			✓		✓	✓	
User 3		✓	✓	✓		✓			
User 4	✓				✓		✓	✓	
User 5	✓		✓		✓			✓	
User 6			✓			✓			
User 7	✓				✓		✓	✓	
User 8				✓					✓

- What is the right ranking to show?
- When different meanings are possible, we probably want to hedge our bets.
- How can we learn that ranking function?
  - Most methods assumed independent doc relevances.

Diversity

- Say for a (user, query), we have a very relevant doc.
  - All very similar docs are also probably relevant.
  - But they are also probably *useless* given the first.
  - Yet standard data would tell us to rank them highly.
  - The usefulness *d2* depends on what *d1* was!

- At the same time, many queries have a number of different possible meanings.
- Often, users can be happy with just one or two relevant documents (e.g. *find a definition*).
- We'd ideally like to capture all the topics.

Diversity: What is it?

- "All the topics" isn't well defined.
- Sometimes, like *flash* or *leopard*, there are very distinct topics.
- But consider *support vector machine*. Different "topics" could include:
  - Tutorials about SVMs.
  - Downloadable SVM implementations.
  - Variants like Ranking SVMs.
  - Theoretical proofs showing why SVMs work.
- How do we draw the line between topics?

Performance isn't always a sum

- From a mathematical perspective, NDCG is a linear performance measure:

$$NDCG(d_1, ..., d_n) = \frac{1}{N} \sum_{i=1}^{n} \frac{2^{r_i} - 1}{\log(i+1)}$$

- r_i is implicitly the *expectation* of the relevance of d_i across all of our users.
- We're summing the contribution of each document.
- The same applies to Precision, Recall, MAP, ...
- Rather, an alternative view is that we should be taking the expectation of the max relevance.

A Heuristic Approach

- Suppose we have a relevance function $rel(q, d)$ and similarity function $sim(d_1, d_2)$
- Maximal Marginal Relevance (MMR) suggests we greedily pick documents in order, maximizing

$$\lambda rel(d, q) - (1 - \lambda) \max_{d' \in D} sim(d, d')$$

  - We first pick the most relevant document
  - We pick the next one to have highest *marginal relevance* given what we've already picked
  - Loop and repeat
- This diversifies an existing ranking!

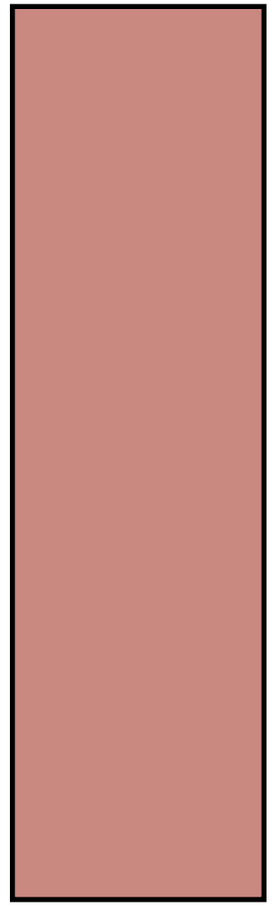
[Carbonell & Goldstein, SIGIR '98]

MMR Example

- Similarities

	d2	d3	d4
d1	0.8	0.3	0.2
d2	-	0.4	0.3
d3	-	-	0.7
d4	-	-	-

Results



- Relevances

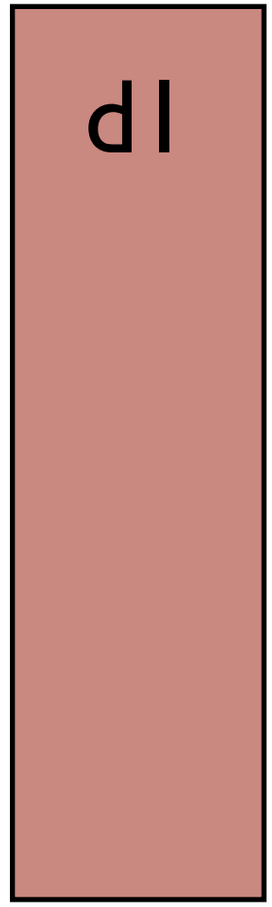
Document	d1	d2	d3	d4
Relevance	0.9	0.7	0.5	0.4
MR (λ=0.7)	0.63	0.49	0.35	0.28

MMR Example

- Similarities

	d2	d3	d4
d1	0.8	0.3	0.2
d2	-	0.4	0.3
d3	-	-	0.7
d4	-	-	-

Results



- Relevances

Document	d1	d2	d3	d4
Relevance	0.9	0.7	0.5	0.4
MR (λ=0.7)	✓	0.25	0.26	0.22

MMR Example

- Similarities

	d2	d3	d4
d1	0.8	0.3	0.2
d2	-	0.4	0.3
d3	-	-	0.7
d4	-	-	-

Results

d1

d3

- Relevances

Document	d1	d2	d3	d4
Relevance	0.9	0.7	0.5	0.4
MR (λ=0.7)	✓	0.25	✓	0.07

MMR Example

- Similarities

	d2	d3	d4
d1	0.8	0.3	0.2
d2	-	0.4	0.3
d3	-	-	0.7
d4	-	-	-

Results

d1

d3

d2

- Relevances

Document	d1	d2	d3	d4
Relevance	0.9	0.7	0.5	0.4
MR (λ=0.7)	✓	✓	✓	0.07

A Document Modeling Approach

- Build a probabilistic relevance model; Pick document that maximizes:

$$P(d_1\ relevant \mid query)$$

- But if that document wasn't relevant, pick the next document to maximize:

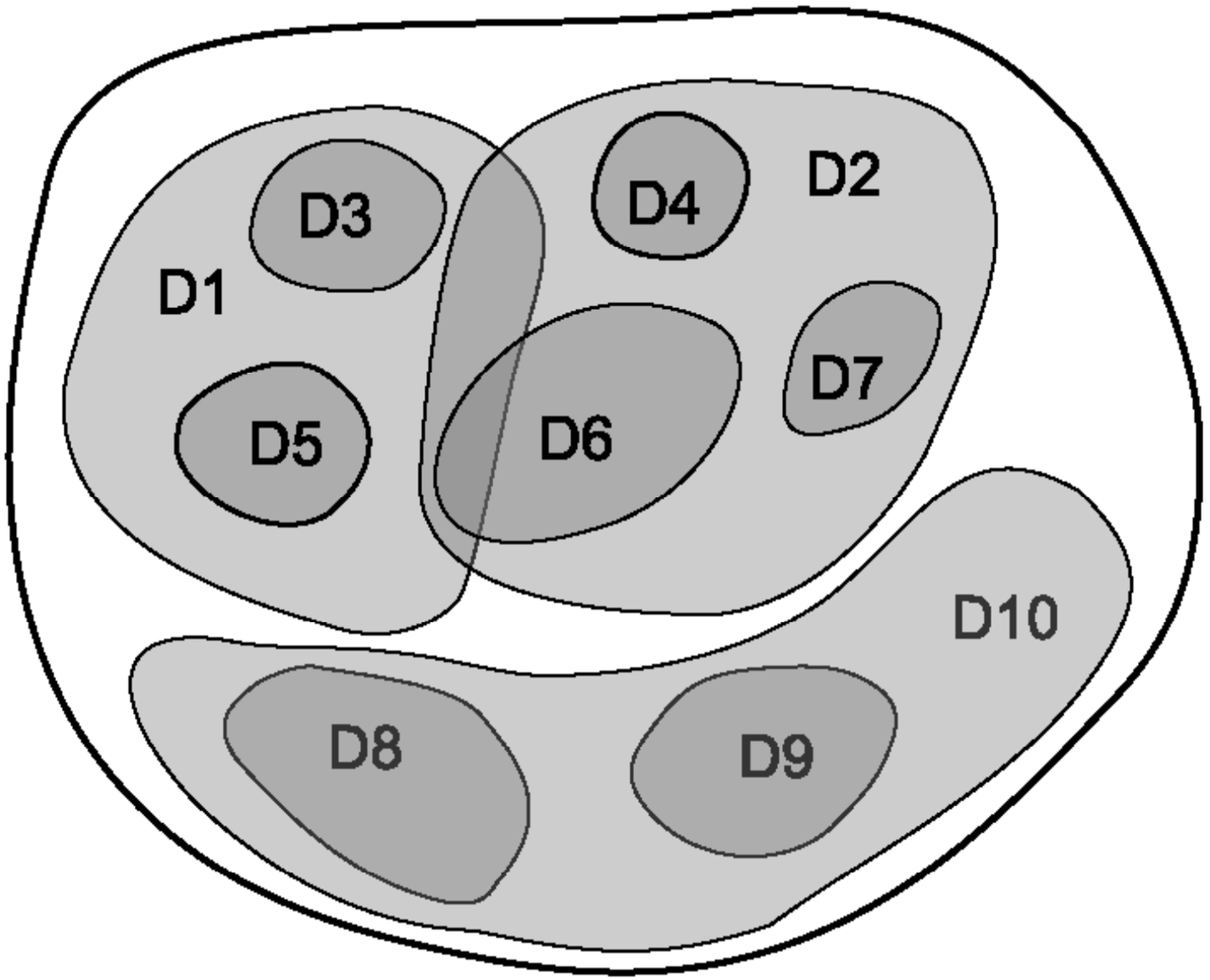
$$P(d_2\ relevant \mid query \wedge d_1\ not\ relevant)$$

- And keep going all the way down the ranking.
- But we need to be able to estimate this probability of relevance. If we are to learn it we need absolute judgments.

[Chen & Karger, SIGIR '07]

Set Cover

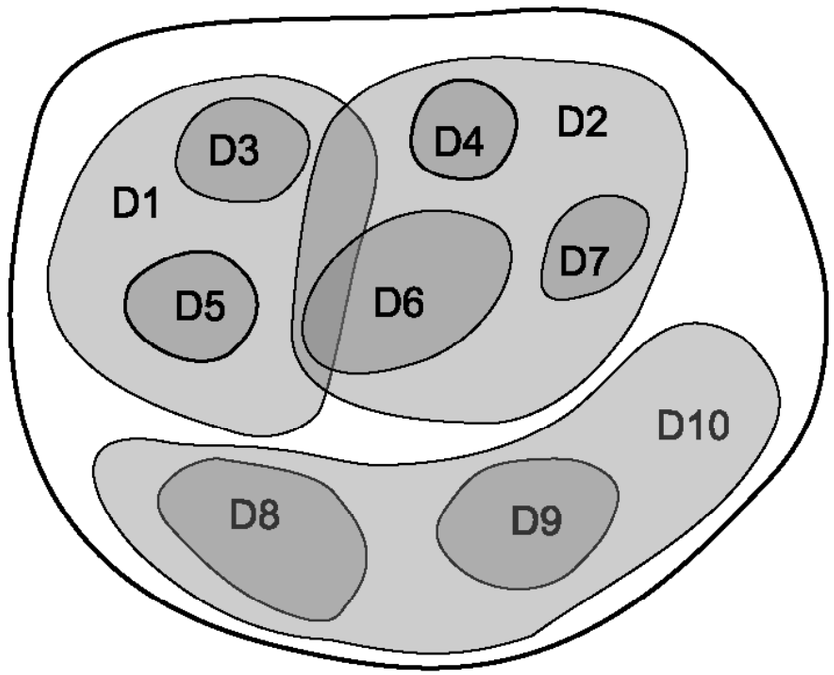
- Another approach to obtaining diverse rankings is to "cover" different topics explicitly.



Set Cover

- Another approach to obtaining diverse rankings is to "cover" different topics explicitly.
- This is the *maximum cover problem*: Its NP hard!
- If we know how much each document adds, we could greedily pick documents, one at a time.

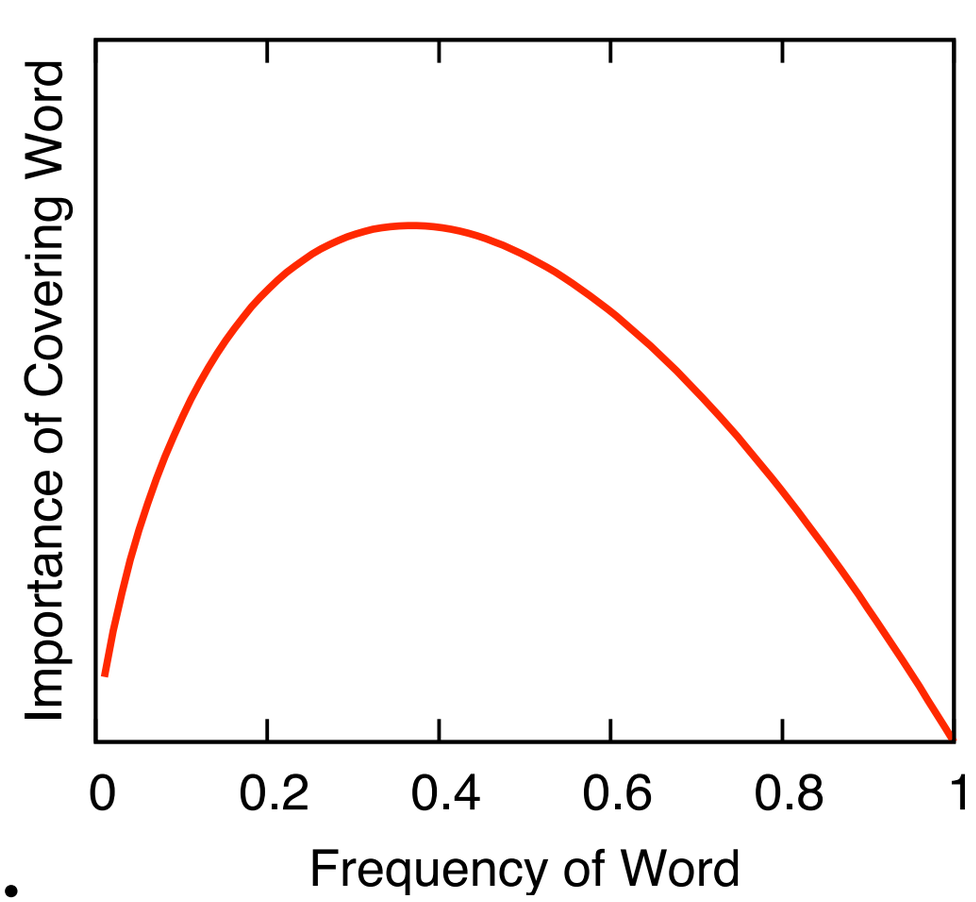
- Greedy gives a $(1 - 1/e)$ approximation (in the worst case). You can't do better in polynomial time.



[Nemhauser et al, MP '78; Khuller et al, IPL '97]

Greedily covering words

- Suppose we have a set of candidate documents.
- We want a subset that covers as many available topics as possible.
- If we don't know the topics, can use words as a proxy.
  - Some words are more important than others.
  - Words can be covered to different degrees.
  - Score of a subset is the sum of the words covered.



[Swaminathan et al '08]

Learning from Diverse Lists

- Why limit ourselves to a static word weighting?
  - Some words are better than others at covering new topics.
  - You might need the same word a few times before a topic is really covered.
- We can build features that describe how important a word is: *Occurs n times; In the title*.
- Can learn how to weight each importance feature to get the best diversity.
  - For training, we need documents with labeled topics!

[Yue & Joachims, ICML '08]

Measuring Diversity with Clicks

- Clicks were useful for relevance judgments, can we use them to optimize for *diverse* rankings?
  - Skipping over a result suggested it was less relevant than something clicked later.
  - If users skipped over everything, the top results were usually particularly bad.
- "No clicks" is usually called *abandonment*.
  - Pros: Easy to observe.
  - Cons: Only negative feedback. Assumes abandonment is bad. What if the answer was in the abstract?

Measuring Diversity with Clicks

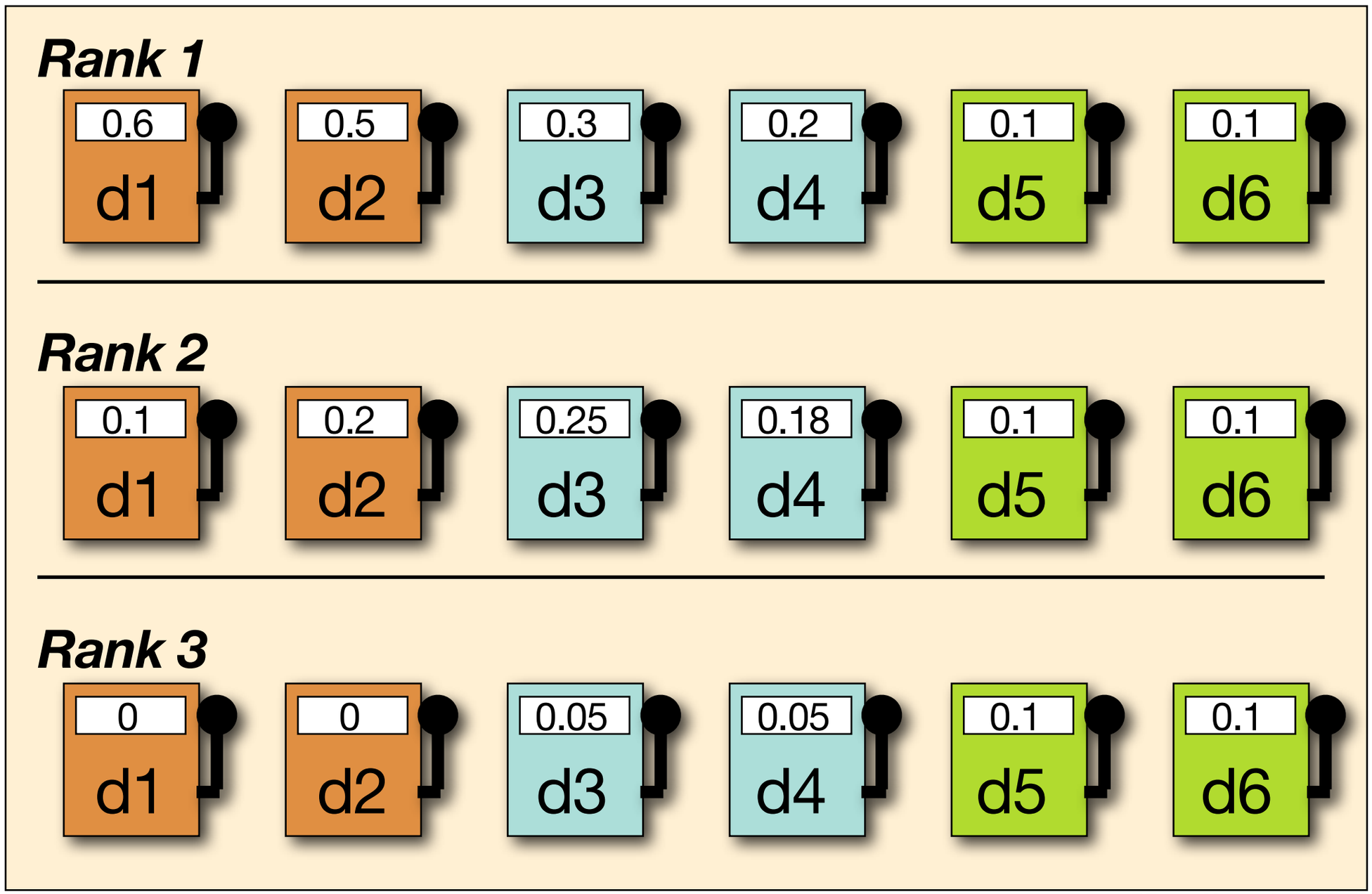
- Suppose we have a binary relevance model:

	d1	d2	d3	d4	d5	d6	d7	d8	d9
User 1	✓				✓		✓	✓	
User 2		✓			✓		✓	✓	
User 3		✓	✓	✓		✓			
User 4	✓				✓		✓	✓	
User 5	✓		✓		✓			✓	
User 6			✓			✓			
User 7	✓				✓		✓	✓	
User 8				✓					✓

- Assume users presented with results click on first relevant document.
- With full information, we can use the greedy algorithm. Can we get close without it?

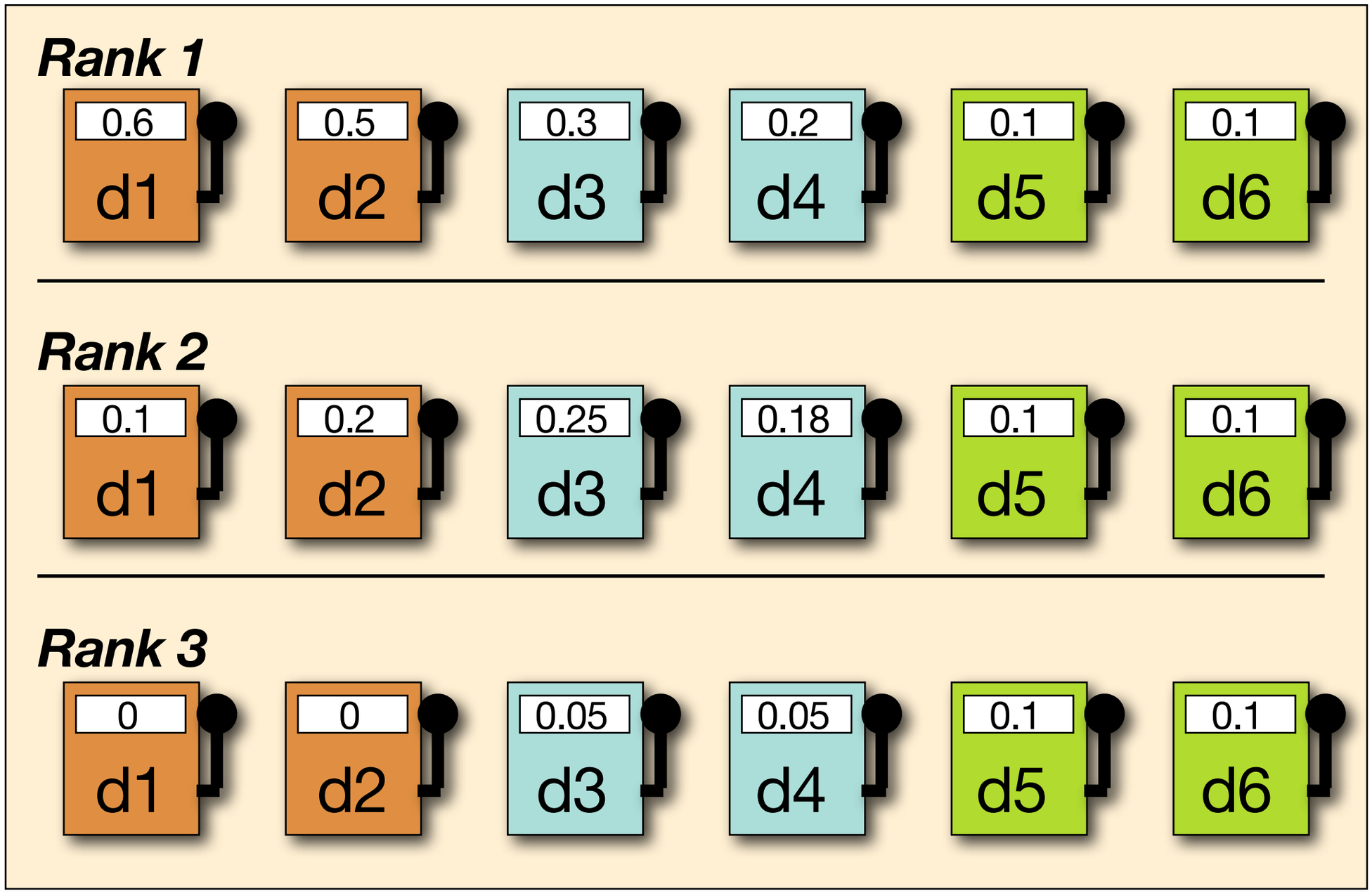
Minimizing Abandonment

- We can learn the value of each document at each rank.
- At rank 1, we'll learn the most popularly relevant document.
- At rank 2, we'll learn the best next one, given the first, etc...



[Radlinski et al, ICML '08]

Minimizing Abandonment



- The payoff we get (non-abandonment) is

$$(1 - 1/e)OPT - sublinear(T)$$

- But we have to learn a huge number of parameters...

[Radlinski et al, ICML '08]

Diversity: Questions

- How do we measure diversity of rankings produced by a ranking function?
- What are the right diversity sensitive loss metrics for machine learning algorithms to optimize?
- Where do we obtain appropriate training data labeled for diversity? What question do you ask human judges?
- Or, what is the best way to interpret clicks as measures of diversity?
- When learning to produce diverse rankings, what features best capture diversity?

Learning to Rank with a Twist

- We've focussed on the task:
  - Given a query, find relevant documents.
- But there is another increasingly studied setting:
  - Given a query, find documents that will make us lots and lots and lots of money.
- This is of course online advertising.
  - Setting 1: rank given a query.
  - Setting 2: rank given a document.
- Evaluation is sort of obvious: The number of dollars your algorithm makes.

Advertising at 10,000 ft

- Lets consider the query setting:
  - We have a query. We want to make money.
  - Each ad has some relevance to the query. Actually, what really matters is *P(click)*
  - Each ad also pays us up to *$x* per click.
- Simplest way: Rank ads by $relevance \times price$
  - If the price is high enough, it dominates.
- Attempt 2: Rank ads by $CTR \times price$
  - How do you know the CTR ahead of time?
  - How do you know when all the ads are terrible?
- How do you charge to get the best bids?

[Edelman, Ostrovsky & Schwarz, AER '07]

Areas for Research

- We want to globally optimize ranking metrics ...
  ... but they are non-convex and non-smooth.
- We want diversity in search results ...
  ... but we don't know how to really optimize it.
- A good performance metric is user satisfaction ...
  ... but we don't know how to measure it.
- Training from user clicks is very appealing ...
  ... but what happens when spammers start clicking?
- The data isn't iid...
  ... but it is highly structured.
- Lots of very interesting open problems!

Some of the data out there

- TREC (many tracks, many scenarios)
- LETOR (some processed TREC & OHSUMED)
- Netflix prize dataset
- (AOL query dataset)
- (build your own search engine proxy)
- (build your own search engine: e.g. Osmot)
- Microsoft, Yahoo!, Google have lots of data...