Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

Evaluation
16 February 2017

Evaluation

- Critical step for understanding if our system/algorithm/improvement actually does anything net positive
  - Or even if it worsens things (!!!)
- Evaluation framework should be targeted to the application scenario:
  - Typically, different metrics and approaches for ranking, clustering, classification, recommender systems
- Today:
  - Focus on evaluating a search engine

IR is an experimental science

- Formulate a research question: the hypothesis
- Design an experiment to answer the question
- Perform the experiment
  - Compare with a baseline "control"
- Does the experiment answer the question?
  - Are the results significant? Or is it just luck?
- Report the results!

Research questions

- Does stemming improve retrieval performance?
  - Experiment: Build a "stemmed" index and compare against an "unstemmed" baseline
- Does expanding the query with synonyms improve retrieval performance?
  - Experiment: Expand queries with synonyms and compare against baseline unexpanded queries

Research questions

- Does keyword highlighting help users evaluate document relevance?
  - Experiment: Build two different interfaces, one with highlighting, one without; run a user study
- Is letting users weight search terms a good idea?
  - Experiment: Build two different interfaces, one with term weighting, one without: run a user study

The importance of evaluation

- The ability to measure differences underlies experimental science
  - How well do our systems work?
  - Is A better than B?
  - Really?
  - Under what conditions?
- Evaluation drives what to research
  - Identify techniques that work and that don't

What we look for in evaluations ...

- Insightful
- Affordable
- Repeatable
- Explainable

Evaluating a Search Engine

Measures for a search engine

- How fast does it index
  - Number of documents/hour
  - (Average document size)
- How fast does it search
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries

Measures for a search engine

- All of the preceding criteria are *measurable*: we can quantify speed/size; we can make expressiveness precise
- The key measure: user happiness
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
- Need a way of quantifying user happiness

How do you tell if users are happy?

- Search returns products relevant to users
  - How do you assess this at scale?
- Search results get clicked a lot
  - Misleading titles/summaries can cause users to click
- Users buy after using the search engine
  - Or, users spend a lot of $ after using the search engine
- Repeat visitors/buyers
  - Do users leave soon after searching?
  - Do they come back within a week/month/… ?

Happiness: elusive to measure

- Most common proxy: **relevance** of search results
  - But how do you measure relevance?
- Pioneered by Cyril Cleverdon in the Cranfield Experiments

Be careful ...

McNamara's fallacy

The quantifying of success in the war (e.g. in terms of enemy body count) while ignoring other variables.

Beware of measuring what is easy instead of what's important.

McNamara’s fallacy

The first step is to measure whatever can be easily measured. This is OK as far as it goes.

The second step is to disregard that which can't be easily measured or to give it an arbitrary quantitative value. This is artificial and misleading.

The third step is to presume that what can't be measured easily really isn't important. This is blindness.

The fourth step is to say that what can't be easily measured really doesn't exist. This is suicide.

Examples of easy to measure but (possibly) not important

Time spent on website (Objective: MAX)

Split pages (annoying news sites) = $$$

Problem: boneheaded data-driven decisions

Time until purchase (Objective: MIN)

Perhaps miss out on serendipity, exploration of other items (e.g., kill the recommender system so you optimize time to checkout)

Happiness: elusive to measure

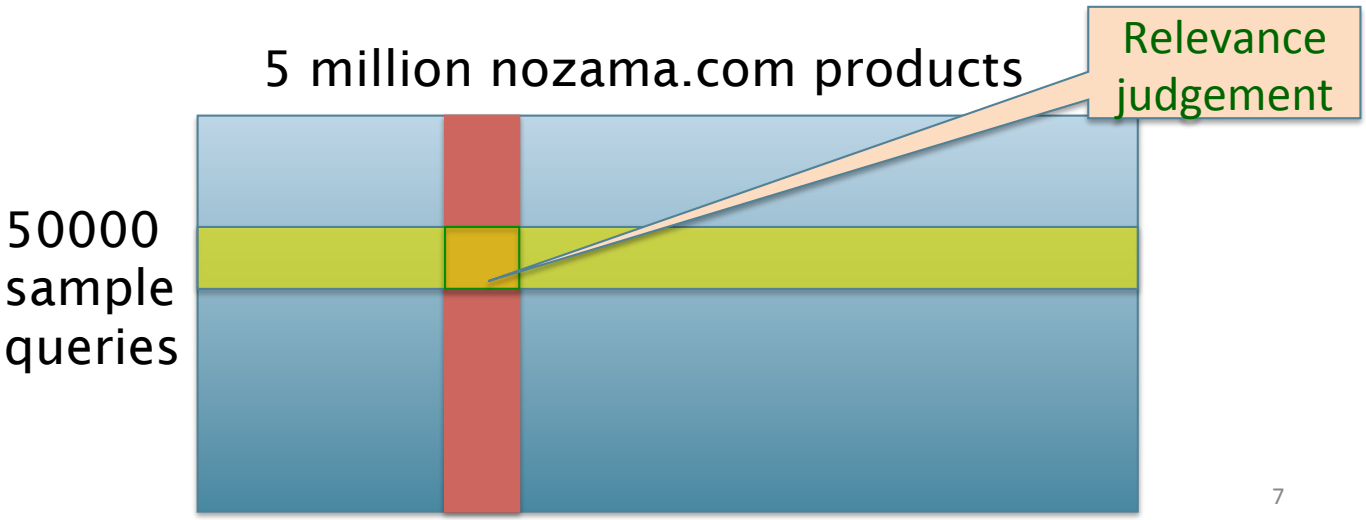
- Most common proxy: **relevance** of search results
  - But how do you measure relevance?
- Pioneered by Cyril Cleverdon in the Cranfield Experiments

Measuring relevance

- Three elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - A binary assessment of either Relevant or Nonrelevant for each query and each document

Let's measure the quality of our new search algorithm

- Benchmark documents – (could be products, web pages, etc.)
- Benchmark query suite – more on this
- Judgments of document relevance for each query



Relevance judgments

- Binary (relevant vs. non-relevant) in the simplest case, more nuanced (0, 1, 2, 3 …) in others
- What are some issues already?
- 5 million times 50K takes us into the range of a quarter trillion judgments
  - If each judgment took a human 2.5 seconds, we'd still need 1011 seconds, or nearly $300 million if you pay people $10 per hour to assess
  - 10K new products per day

Crowdsource relevance judgments?

- Present query-document pairs to low-cost labor on online crowd-sourcing platforms
  - Hope that this is cheaper than hiring qualified assessors
- Lots of literature on using crowd-sourcing for such tasks
- Main takeaway – you get some signal, but the variance in the resulting judgments is very high

What else?

- Still need test queries
  - Must be germane to docs available
  - Must be representative of actual user needs
  - Random query terms from the documents generally not a good idea
  - Sample from query logs if available
- Classically (non-Web)
  - Low query rates – not enough query logs
  - Experts hand-craft "user needs"

Some public test collections

TABLE 4.3 Common Test Corpora

Collection	*NDocs*	*NQrys*	*Size (MB)*	*Term/Doc*	*Q-D RelAss*
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000

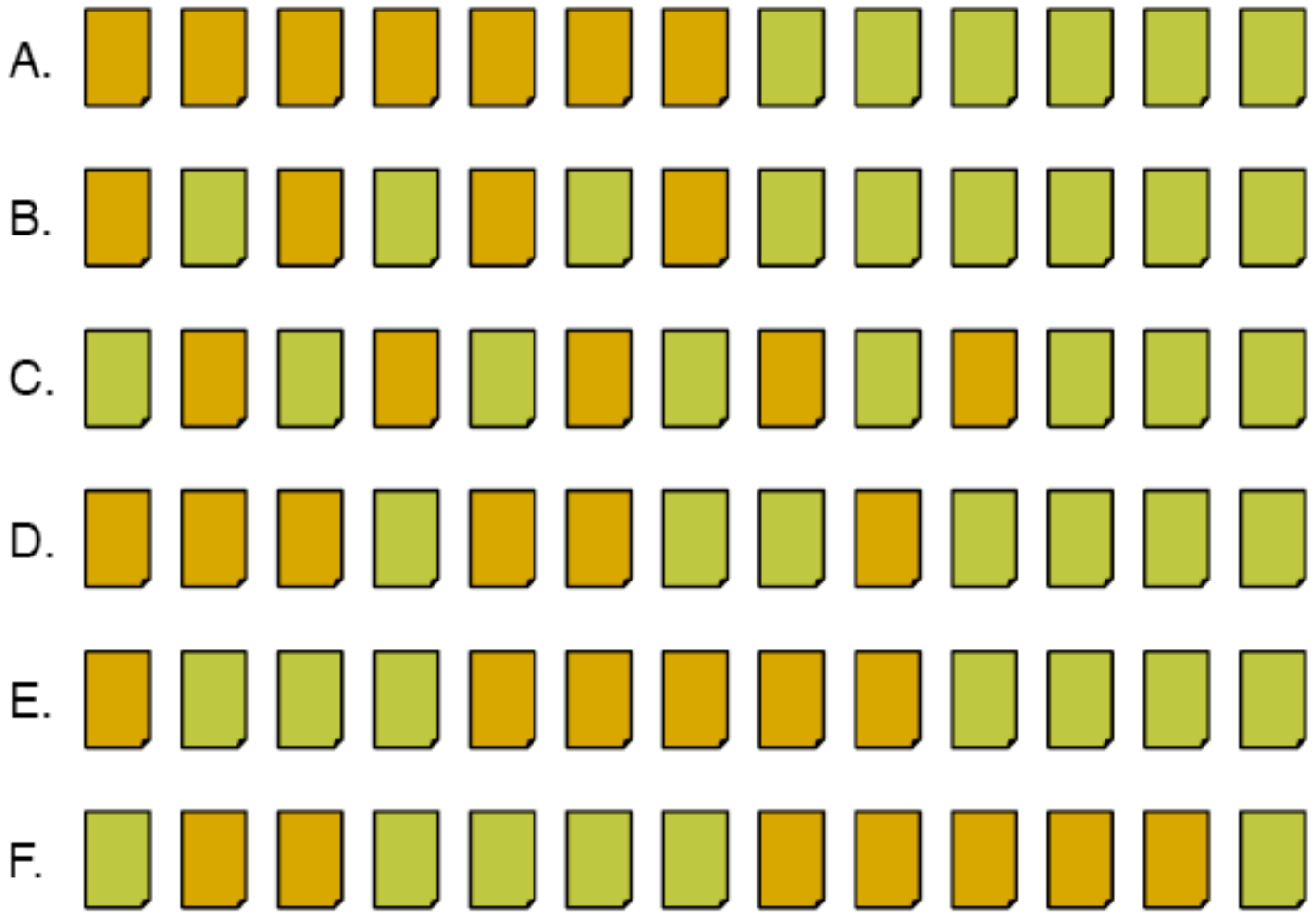
Now we have the basics of a benchmark

- Let's review some evaluation measures
  - Precision
  - Recall
  - F
  - Precision @ k
  - NDCG

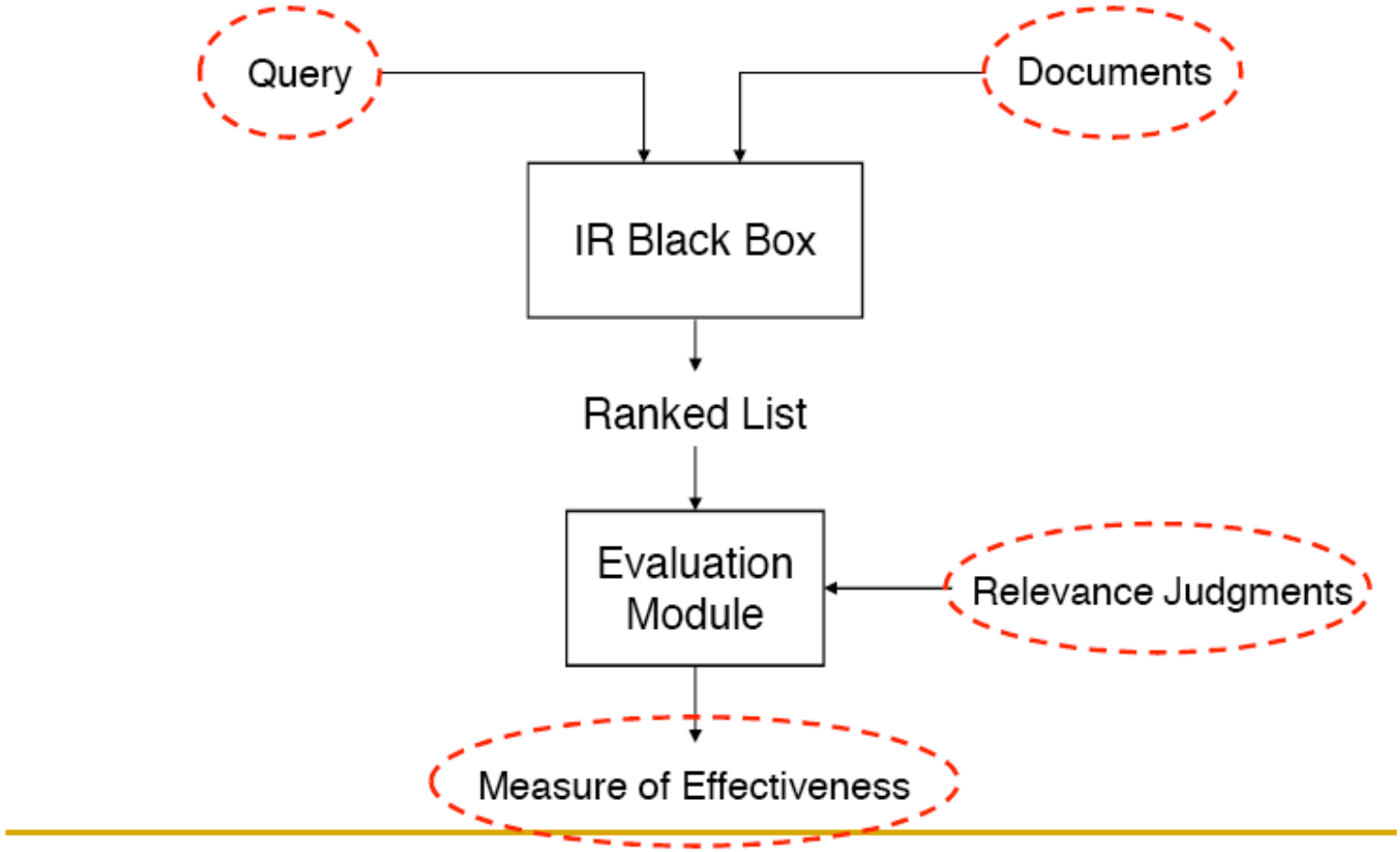
Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., Information need: *: My swimming pool bottom is becoming black and needs to be cleaned.*
- Query: ***pool cleaner***
- You evaluate whether the doc addresses the information need, not whether it has those words

Which is the best rank order?



Automatic evaluation model



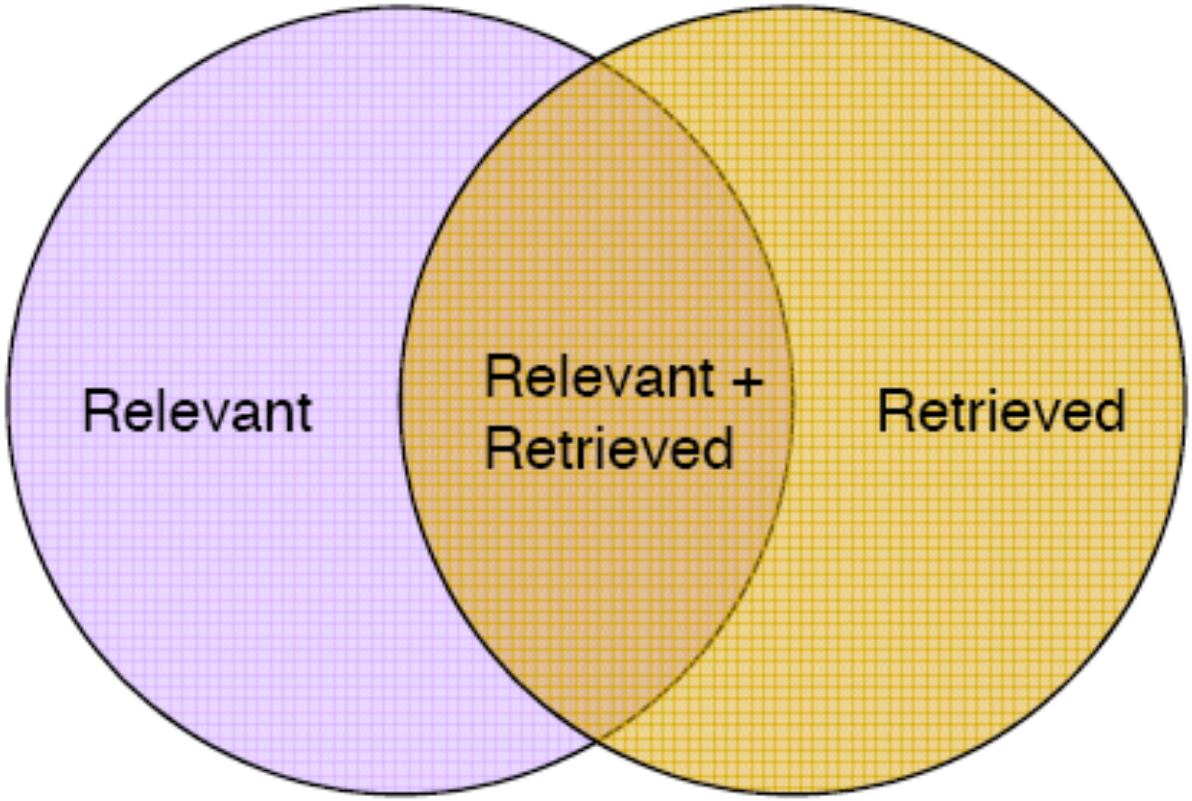
Unranked Evaluation

Unranked retrieval evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

	Relevant	Not Relevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)


Relevant
Relevant +
Retrieved
Retrieved
Not Relevant + Not Retrieved

Precision/Recall: Things to watch out for

- Should average over large number of queries
  - 100s to 1000s
- Assessments have to be binary
  - more on this later
- Heavily skewed by corpus/authorship
  - Results may not translate from one domain to another

Precision/Recall tradeoff

- You can increase recall by returning more docs.
- Recall is a non-decreasing function of the number of docs retrieved.
- A system that returns all docs has 100% recall!
- The converse is also true (usually): It's easy to get high precision for very low recall.
- Assume the document with the largest score is relevant. How can we maximize precision?

A combined measure: *F*

- Combined measure that assesses this tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = ½$
- Harmonic mean is a conservative average

F-measure details

$\beta^2 = \frac{1-\alpha}{\alpha}$

Harmonic mean: $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$

$F_1 = \frac{2PR}{P+R}$

F-measure example

	relevant	not relevant
retrieved	18	2
not retrieved	82	1,000,000,000

Precision?
Recall?
F?

F-measure example

	relevant	not relevant
retrieved	18	2
not retrieved	82	1,000,000,000

- precision = 18/(18+2) = 0.9
- recall = 18/(18+82) = 0.18
- F = 2PR/(P+R) = 2 * 0.9 * 0.18 / (0.9+0.18) = 0.3
- Note: F is a lot lower than AVG(P,Q) = 0.54
- Number of true negatives is not factored in: same F for 1000 true negatives

Ranked evaluation

Precision @ k

- Ranked version of precision
- Only evaluate precision for the top-k results
  - The denominator is always k
- Example: if 3 out of 5 results are relevant, then precision@5 is 0.6

DCG

- Normalized Discounted Cumulative Gain
- Sensitive to the position of the highest rated page
- Log-discounting of results
- Normalized for different lengths lists

DCG

- Popular measure for evaluating web search and related tasks
- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant documents
  - the lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

DCG

- Uses graded relevance as a measure of usefulness, or gain, from examining a document
- Gain is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks
- Typical discount is 1/log (rank)
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3

DCG

- What if relevance judgments are in a scale of [0,r]? r>2
- Cumulative Gain (CG) at rank n
  - Let the ratings of the n documents be r1, r2, …rn (in ranked order)
  - CG = r1+r2+…rn
- Discounted Cumulative Gain (DCG) at rank n
  - DCG = r1 + r2/$\log_2 2$ + r3/$\log_2 3$ + … rn/$\log_2 n$
  - We may use any base for the logarithm

DCG

- DCG is the total gain accumulated at a particular rank p:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

- Alternative formulation:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1+i)}$$

- used by some web search companies
- emphasis on retrieving highly relevant documents

DCG Example

- 10 ranked documents judged on 0-3 relevance scale:
  - 3, 2, 3, 0, 0, 1, 2, 2, 3, 0
- discounted gain:
  - 3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0
  - = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0
- DCG:
  - 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Summarize a ranking: NDCG

- Normalized Discounted Cumulative Gain (NDCG) at rank n
  - Normalize DCG at rank n by the DCG value at rank n of the ideal ranking
  - The ideal ranking would first return the documents with the highest relevance level, then the next highest relevance level, etc
- Normalization useful for contrasting queries with varying numbers of relevant results
- NDCG is now quite popular in evaluating Web search

NDCG: Example

i	Ground Truth		Ranking Function$_1$		Ranking Function$_2$	
	Document Order	r_i	Document Order	r_i	Document Order	r_i
1	d4	2	d3	2	d3	2
2	d3	2	d4	2	d2	1
3	d2	1	d2	1	d4	2
4	d1	0	d1	0	d1	0
	$NDCG_{GT}$=1.00		$NDCG_{RF1}$=1.00		$NDCG_{RF2}$=0.9203	

$$DCG_{GT} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF1} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF2} = 2 + \left(\frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.2619$$

$$MaxDCG = DCG_{GT} = 4.6309$$

Human judgments are:

- Expensive
- Inconsistent
  - Between raters
  - Over time
- Decay in value as documents/query mix evolves
- Not always representative of "real users"
  - Rating vis-à-vis query, vs underlying need
- So – what alternatives do we have?

Using User Clicks

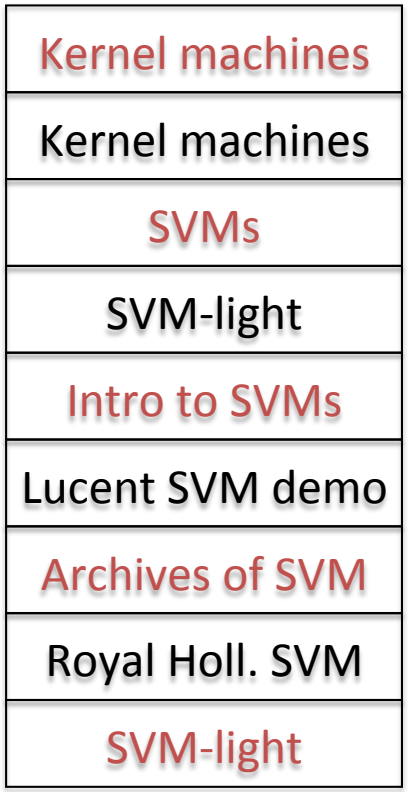
Comparing two rankings via clicks

Query: [support vector machines]

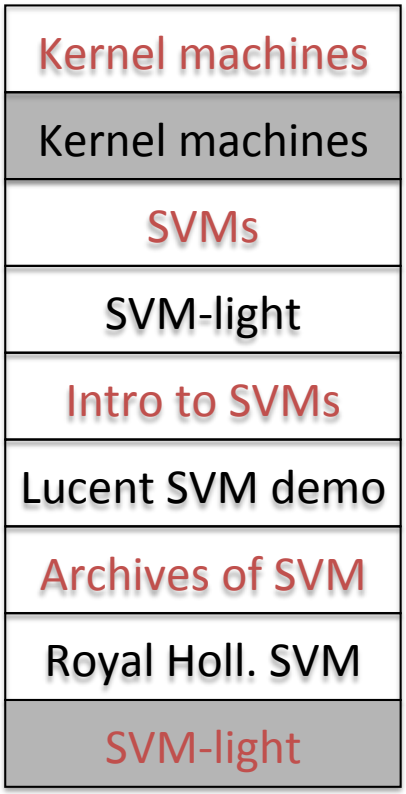
Ranking A	Ranking B
Kernel machines	Kernel machines
SVM-light	SVMs
Lucent SVM demo	Intro to SVMs
Royal Holl. SVM	Archives of SVM
SVM software	SVM-light
SVM tutorial	SVM software

Interleave the two rankings

This interleaving starts with B

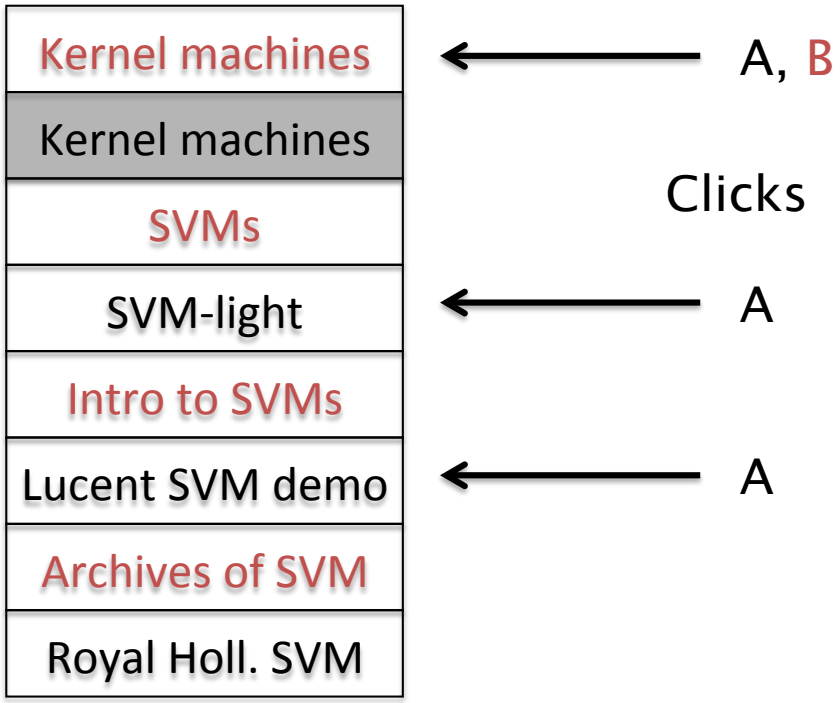


Remove duplicates



Count user clicks

Ranking A: 3
Ranking B: 1

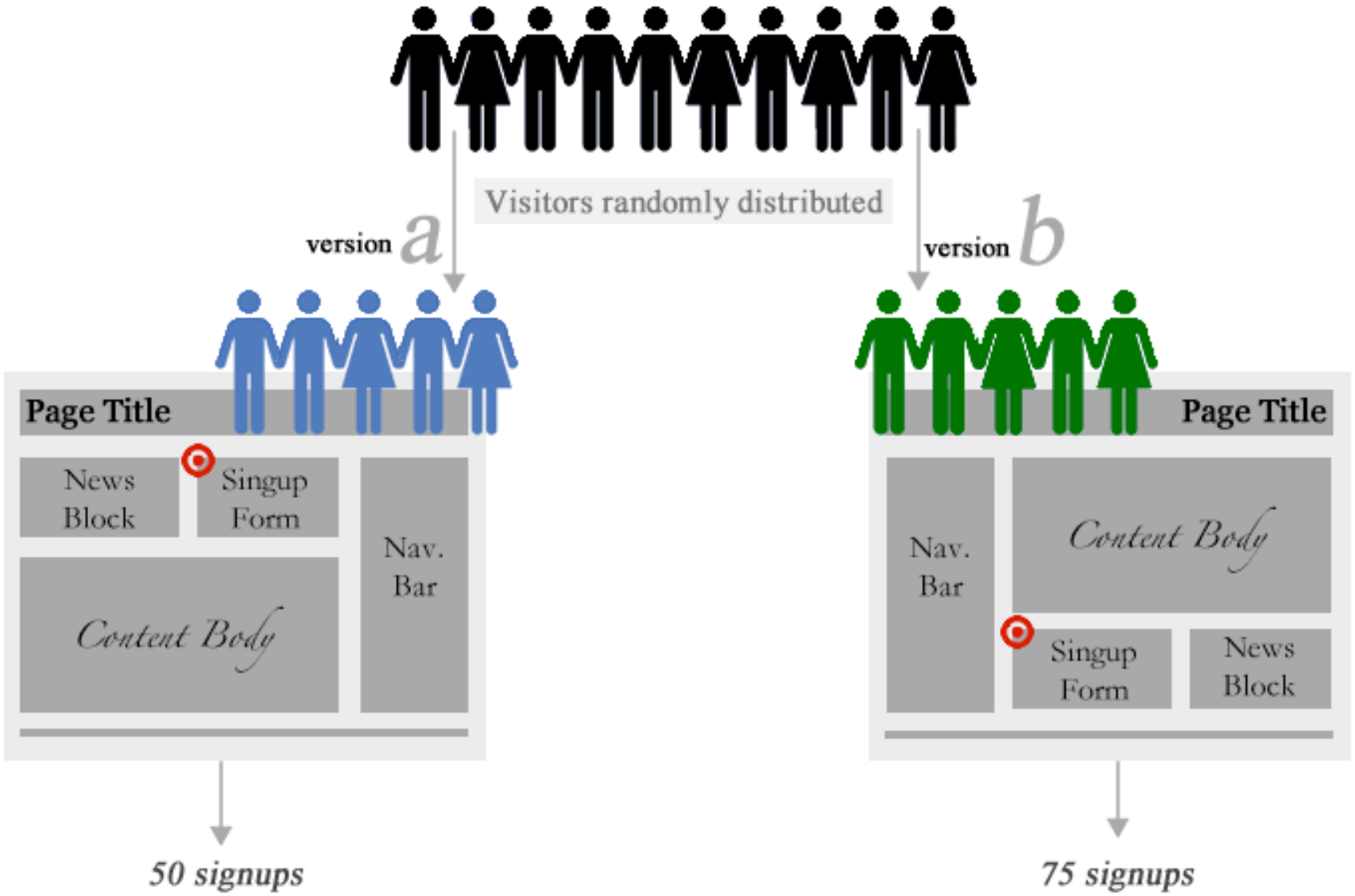


Interleaved ranking

- Present interleaved ranking to users
  - Start randomly with ranking A or ranking B to evens out presentation bias
- Count clicks on results from A versus results from B
- Better ranking will (on average) get more clicks

A/B Testing: Randomized Controlled Experiments

http://www.wired.com/business/2012/04/ff_abtesting/all/



Version B is better than version A

Why run experiments?

- Gathers data on impact of changes
  - How do users behave differently, if at all?
- Data-driven decisions:
  - UI

Hotels.com Official Site
www.**hotels**.com
Hotels.com Low Rates Guaranteed! Call a **Hotel** Expert. 1-866-925-0513

Hotels.com Official Site
www.**hotels**.com
Hotels.com Low Rates Guaranteed! Call a **Hotel** Expert. 1-866-925-0513

Hotels.com Official Site
www.**hotels**.com
Hotels.com Low Rates Guaranteed! Call a **Hotel** Expert. 1-866-925-0513

Why run experiments?

- Gathers data on impact of changes
  - How do users behave differently, if at all?
  - Test everything!
- Data-driven decisions
  - UI

Why run experiments?

- Gathers data on impact of changes
  - How do users behave differently, if at all?
- Data-driven decisions
  - UI
  - Algorithms, e.g. CTR prediction
    - How many passes over the data
    - Date range
    - Different machine learning algorithms

Google