Lecture 5: Evaluation

Information Retrieval
Computer Science Tripos Part II

Simone Teufel

Natural Language and Information Processing (NLIP) Group

UNIVERSITY OF CAMBRIDGE

Simone.Teufel@cl.cam.ac.uk

Lent 2014

Overview

1. Recap/Catchup
2. Introduction
3. Unranked evaluation
4. Ranked evaluation
5. Benchmarks
6. Other types of evaluation

Overview

tf-idf

tf-idf

$$w_{t,d} = (1 + \log \mathrm{tf}_{t,d}) \cdot \log \frac{N}{\mathrm{df}_t}$$

Cosine similarity of $\vec{q}$ and $\vec{d}$

$$\cos(\vec{q}, \vec{d}) = \textsc{sim}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

- q_i : tf-idf weight of term i in the query.
- d_i : tf-idf weight of term i in the document.
- $|\vec{q}|$ and $|\vec{d}|$: lengths of $\vec{q}$ and $\vec{d}$.

Components of tf-idf weighting

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \ldots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N - df_t}{df_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t \in d}(tf_{t,d}))}$				

Best known combination of weighting options

Default: no weighting

tf-idf example

- We often use different weightings for queries and documents.
- Notation: ddd.qqq

Example: lnc.ltn

Document:

- $\boxed{\text{l}}$ogarithmic tf
- $\boxed{\text{n}}$o df weighting
- $\boxed{\text{c}}$osine normalization

Query:

- $\boxed{\text{l}}$ogarithmic tf
- $\boxed{\text{t}}$ (means idf)
- $\boxed{\text{n}}$o normalization

tf-idf example: lnc.ltn

Query: "best car insurance". Document: "car insurance auto insurance".

word	query					document				product
	tf-raw	tf-wght	df	idf	weight	tf-raw	tf-wght	weight	n'lized	
auto	0	0	5000	2.3	0	1	1	1	0.52	0
best	1	1	50000	1.3	1.3	0	0	0	0	0
car	1	1	10000	2.0	2.0	1	1	1	0.52	1.04
insurance	1	1	1000	3.0	3.0	2	1.3	1.3	0.68	2.04

Key to columns: tf-raw: raw (unweighted) term frequency, tf-wght: logarithmically weighted term frequency, df: document frequency, idf: inverse document frequency, weight: the final weight of the term in the query or document, n'lized: document weights after cosine normalization, product: the product of final query weight and final document weight

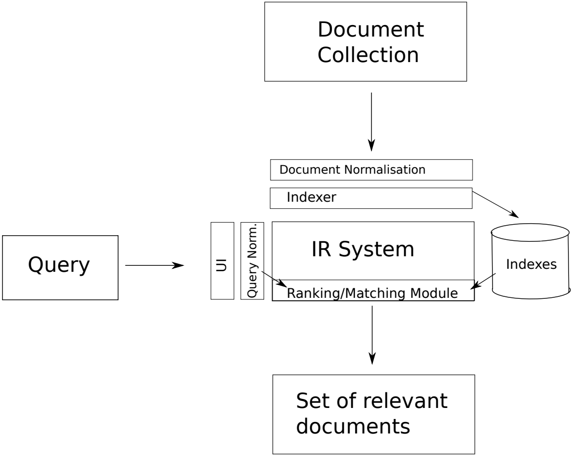
$\sqrt{1^2 + 0^2 + 1^2 + 1.3^2} \approx 1.92$
$1/1.92 \approx 0.52$
$1.3/1.92 \approx 0.68$

Final similarity score between query and document: $\sum_i w_{qi} \cdot w_{di} = 0 + 0 + 1.04 + 2.04 = 3.08$

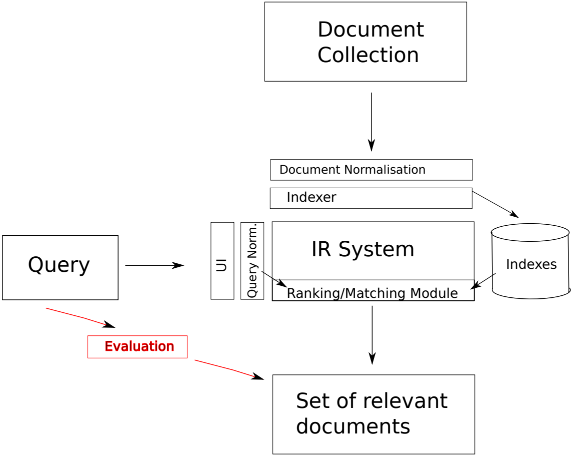
Summary: Ranked retrieval in the vector space model

- Represent the query as a weighted tf-idf vector
- Represent each document as a weighted tf-idf vector
- Compute the cosine similarity between the query vector and each document vector
- Rank documents with respect to the query
- Return the top K (e.g., $K = 10$) to the user

Today


Document Collection
Document Normalisation
Indexer
Query
UI
Query Norm.
IR System
Ranking/Matching Module
Indexes
Set of relevant documents

Today



Today: how good are the returned documents?

Overview

1. Recap/Catchup
2. Introduction
3. Unranked evaluation
4. Ranked evaluation
5. Benchmarks
6. Other types of evaluation

Measures for a search engine

- How fast does it index?
  - e.g., number of bytes per hour
- How fast does it search?
  - e.g., latency as a function of queries per second
- What is the cost per query?
  - in dollars

Measures for a search engine

- All of the preceding criteria are measurable: we can quantify speed / size / money
- However, the key measure for a search engine is user happiness.
- What is user happiness?
- Factors include:
  - Speed of response
  - Size of index
  - Uncluttered UI
  - Most important: relevance
  - (actually, maybe even more important: it's free)
- Note that none of these is sufficient: blindingly fast, but useless answers won't make a user happy.

Who is the user?

- Who is the user we are trying to make happy?
- Web search engine: searcher. Success: Searcher finds what she was looking for. Measure: rate of return to this search engine
- Web search engine: advertiser. Success: Searcher clicks on ad. Measure: clickthrough rate
- Ecommerce: buyer. Success: Buyer buys something. Measures: time to purchase, fraction of "conversions" of searchers to buyers
- Ecommerce: seller. Success: Seller sells something. Measure: profit per item sold
- Enterprise: CEO. Success: Employees are more productive (because of effective search). Measure: profit of the company

Most common definition of user happiness: Relevance

- User happiness is equated with the relevance of search results to the query.
- But how do you measure relevance?
- Standard methodology in information retrieval consists of three elements.
  - A benchmark document collection
  - A benchmark suite of queries
  - An assessment of the relevance of each query-document pair

Relevance: query vs. information need

- Relevance to what? The query?

Information need i

"I am looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine."

- translated into:

Query q

[red wine white wine heart attack]

- So what about the following document:

Document d'

At the heart of his speech was an attack on the wine industry lobby for downplaying the role of red and white wine in drunk driving.

- d' is an excellent match for query q . . .
- d' is not relevant to the information need i.

Relevance: query vs. information need

- User happiness can only be measured by relevance to an information need, not by relevance to queries.
- Sloppy terminology here and elsewhere in the literature: we talk about query–document relevance judgments even though we mean information-need–document relevance judgments.

Overview

Precision and recall

- Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

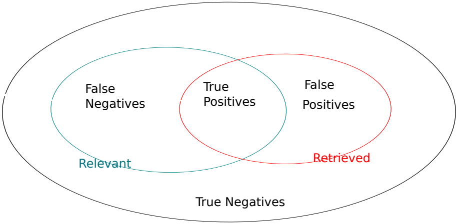
- Recall (R) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

THE TRUTH

WHAT THE SYSTEM THINKS

	Relevant	Nonrelevant
Retrieved	true positives (TP)	false positives (FP)
Not retrieved	false negatives (FN)	true negatives (TN)



$$P = TP/(TP + FP)$$
$$R = TP/(TP + FN)$$

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.

A combined measure: F

- F allows us to trade off precision against recall.

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

- $\alpha \in [0,1]$ and thus $\beta^2 \in [0,\infty]$
- Most frequently used: balanced F with $\beta = 1$ or $\alpha = 0.5$
  - This is the harmonic mean of P and R: $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$

Example for precision, recall, F1

	relevant	not relevant	
retrieved	20	40	60
not retrieved	60	1,000,000	1,000,060
	80	1,000,040	1,000,120

- $P = 20/(20 + 40) = 1/3$
- $R = 20/(20 + 60) = 1/4$
- $F_1 = 2\frac{1}{\frac{1}{\frac{1}{3}} + \frac{1}{\frac{1}{4}}} = 2/7$

Accuracy

- Why do we use complex measures like precision, recall, and F?
- Why not something simple like accuracy?
- Accuracy is the fraction of decisions (relevant/nonrelevant) that are correct.
- In terms of the contingency table above, accuracy $= (TP + TN)/(TP + FP + FN + TN)$.

Thought experiment

- Compute precision, recall and F_1 for this result set:

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

- The snoogle search engine below always returns 0 results ("0 matching results found"), regardless of the query.

- Snoogle demonstrates that accuracy is not a useful measure in IR.

Why accuracy is a useless measure in IR

- Simple trick to maximize accuracy in IR: always say no and return nothing
- You then get 99.99% accuracy on most queries.
- Searchers on the web (and in IR in general) want to find something and have a certain tolerance for junk.
- It's better to return some bad hits as long as you return something.
- $\rightarrow$ We use precision, recall, and F for evaluation, not accuracy.

Recall-criticality and precision-criticality

- Inverse relationship between precision and recall forces general systems to go for compromise between them
- But some tasks particularly need good precision whereas others need good recall:

	Precision-critical task	Recall-critical task
Time	matters	matters less
Tolerance to cases of overlooked information	a lot	none
Information Redundancy	There may be many equally good answers	Information is typically found in only one document
Examples	web search	legal search, patent search

- We should always average over a large set of queries.
  - There is no such thing as a "typical" or "representative" query.
- We need relevance judgments for information-need-document pairs – but they are expensive to produce.
- For alternatives to using precision/recall and having to produce relevance judgments – see end of this lecture.

Overview

Moving from unranked to ranked evaluation

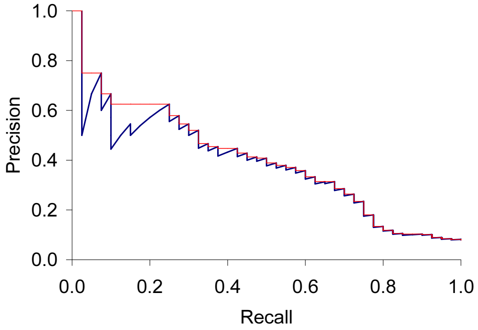
- Precision/recall/F are measures for unranked sets.
- We can easily turn set measures into measures of ranked lists.
- Just compute the set measure for each "prefix": the top 1, top 2, top 3, top 4 etc results
- This is called Precision/Recall at Rank
- Rank statistics give some indication of how quickly user will find relevant documents from ranked list

Precision/Recall @ Rank

Rank	Doc
1	d_{12}
2	d_{123}
3	d_4
4	d_{57}
5	d_{157}
6	d_{222}
7	d_{24}
8	d_{26}
9	d_{77}
10	d_{90}

- Blue documents are relevant
- P@n: P@3=0.33, P@5=0.2, P@8=0.25
- R@n: R@3=0.33, R@5=0.33, R@8=0.66

A precision-recall curve



- Each point corresponds to a result for the top k ranked hits ($k = 1, 2, 3, 4, \ldots$)
- Interpolation (in red): Take maximum of all future points
- Rationale for interpolation: The user is willing to look at more stuff if both precision and recall get better.

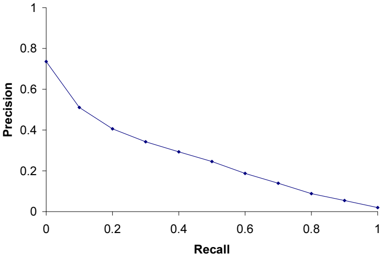
Another idea: Precision at Recall r

Rank	S1	S2
1	X	
2		X
3	X	
4		
5		X
6	X	X
7		X
8		X
9	X	
10	X	

$\rightarrow$

	S1	S2
p @ r 0.2	1.0	0.5
p @ r 0.4	0.67	0.4
p @ r 0.6	0.5	0.5
p @ r 0.8	0.44	0.57
p @ r 1.0	0.5	0.63

Averaged 11-point precision/recall graph



- Compute interpolated precision at recall levels 0.0, 0.1, 0.2, ...
- Do this for each of the queries in the evaluation benchmark
- Average over queries
- The curve is typical of performance levels at TREC (more later).

Averaged 11-point precision more formally

$$P_{11\_pt} = \frac{1}{11}\sum_{j=0}^{10}\frac{1}{N}\sum_{i=1}^{N}\tilde{P}_i(r_j)$$

with $\tilde{P}_i(r_j)$ the precision at the jth recall point in the ith query (out of N)

- Define 11 standard recall points $r_j = \frac{j}{10}$: $r_0 = 0$, $r_1 = 0.1$ ... $r_{10} = 1$
- To get $\tilde{P}_i(r_j)$, we can use $P_i(R = r_j)$ directly if a new relevant document is retrieved exacty at r_j
- Interpolation for cases where there is no exact measurement at r_j:

$$\tilde{P}_i(r_j) = \begin{cases} max(r_j \leq r < r_{j+1}) P_i(R = r) & \text{if } P_i(R = r) \text{ exists} \\ \tilde{P}_i(r_{j+1}) & \text{otherwise} \end{cases}$$

- Note that $P_i(R = 1)$ can always be measured.
- Worked avg-11-pt prec example for supervisions at end of slides.

Mean Average Precision (MAP)

- Also called "average precision at seen relevant documents"
- Determine precision at each point when a new relevant document gets retrieved
- Use P=0 for each relevant document that was not retrieved
- Determine average for each query, then average over queries

$$MAP = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{Q_j} \sum_{i=1}^{Q_j} P(doc_i)$$

with:

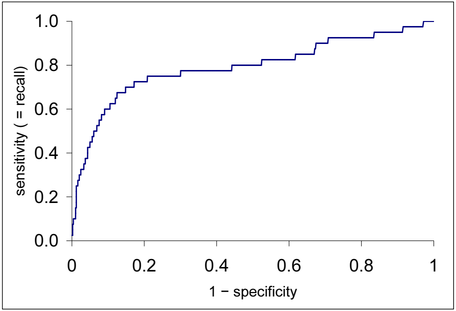
Q_j	number of relevant documents for query j
N	number of queries
$P(doc_i)$	precision at ith relevant document

Mean Average Precision: example ($MAP = \frac{0.564+0.623}{2} = 0.594$)

Query 1		
Rank		$P(doc_i)$
1	X	1.00
2		
3	X	0.67
4		
5		
6	X	0.50
7		
8		
9		
10	X	0.40
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	X	0.25
AVG:		0.564

Query 2		
Rank		$P(doc_i)$
1	X	1.00
2		
3	X	0.67
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15	X	0.2
AVG:		0.623

ROC curve (Receiver Operating Characteristic)



- x-axis: FPR (false positive rate): FP/total actual negatives;
- y-axis: TPR (true positive rate): TP/total actual positives, (also called sensitivity) $\equiv$ recall
- FPR = fall-out = 1 - specificity (TNR; true negative rate)
- But we are only interested in the small area in the lower left corner (blown up by prec-recall graph)

Variance of measures like precision/recall

- For a test collection, it is usual that a system does badly on some information needs (e.g., $P = 0.2$ at $R = 0.1$) and really well on others (e.g., $P = 0.95$ at $R = 0.1$).
- Indeed, it is usually the case that the variance of the same system across queries is much greater than the variance of different systems on the same query.
- That is, there are easy information needs and hard ones.

Overview

What we need for a benchmark

- A collection of documents
  - Documents must be representative of the documents we expect to see in reality.
- A collection of information needs
  - ... which we will often incorrectly refer to as queries
  - Information needs must be representative of the information needs we expect to see in reality.
- Human relevance assessments
  - We need to hire/pay "judges" or assessors to do this.
  - Expensive, time-consuming
  - Judges must be representative of the users we expect to see in reality.

First standard relevance benchmark: Cranfield

- Pioneering: first testbed allowing precise quantitative measures of information retrieval effectiveness
- Late 1950s, UK
- 1398 abstracts of aerodynamics journal articles, a set of 225 queries, exhaustive relevance judgments of all query-document-pairs
- Too small, too untypical for serious IR evaluation today

Second-generation relevance benchmark: TREC

- TREC = Text Retrieval Conference (TREC)
- Organized by the U.S. National Institute of Standards and Technology (NIST)
- TREC is actually a set of several different relevance benchmarks.
- Best known: TREC Ad Hoc, used for first 8 TREC evaluations between 1992 and 1999
- 1.89 million documents, mainly newswire articles, 450 information needs
- No exhaustive relevance judgments – too expensive
- Rather, NIST assessors' relevance judgments are available only for the documents that were among the top k returned for some system which was entered in the TREC evaluation for which the information need was developed.

<num> Number: 508
<title> hair loss is a symptom of what diseases
<desc> Description:
Find diseases for which hair loss is a symptom.
<narr> Narrative:
A document is relevant if it positively connects the loss of head hair in humans with a specific disease. In this context, “thinning hair” and “hair loss” are synonymous. Loss of body and/or facial hair is irrelevant, as is hair loss caused by drug therapy.

Humans decide which document–query pairs are relevant.

Example of more recent benchmark: ClueWeb09

- 1 billion web pages
- 25 terabytes (compressed: 5 terabyte)
- Collected January/February 2009
- 10 languages
- Unique URLs: 4,780,950,903 (325 GB uncompressed, 105 GB compressed)
- Total Outlinks: 7,944,351,835 (71 GB uncompressed, 24 GB compressed)

Interjudge agreement at TREC

information need	number of docs judged	disagreements
51	211	6
62	400	157
67	400	68
95	400	110
127	400	106

Impact of interjudge disagreement

- Judges disagree a lot. Does that mean that the results of information retrieval experiments are meaningless?
- No.
- Large impact on absolute performance numbers
- Virtually no impact on ranking of systems
- Supposes we want to know if algorithm A is better than algorithm B
- An information retrieval experiment will give us a reliable answer to this question ...
- ... even if there is a lot of disagreement between judges.

Overview

Evaluation at large search engines

- Recall is difficult to measure on the web
- Search engines often use precision at top k, e.g., $k = 10$ ...
- ... or use measures that reward you more for getting rank 1 right than for getting rank 10 right.
- Search engines also use non-relevance-based measures.
  - Example 1: clickthrough on first result
  - Not very reliable if you look at a single clickthrough (you may realize after clicking that the summary was misleading and the document is nonrelevant) ...
  - ... but pretty reliable in the aggregate.
  - Example 2: A/B testing

A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an "automatic" measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most

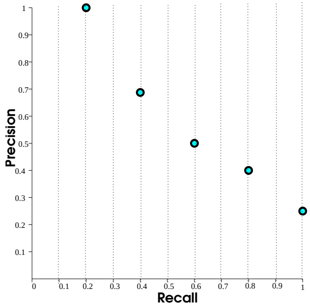
Take-away today

- Focused on evaluation for ad-hoc retrieval
  - Precision, Recall, F-measure
  - More complex measures for ranked retrieval
  - other issues arise when evaluating different tracks, e.g. QA, although typically still use P/R-based measures
- Evaluation for interactive tasks is more involved
- Significance testing is an issue
  - could a good result have occurred by chance?
  - is the result robust across different document sets?
  - slowly becoming more common
  - underlying population distributions unknown, so apply non-parametric tests such as the sign test

Reading

- MRS, Chapter 8

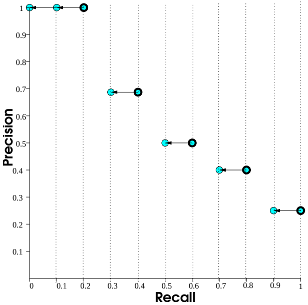
Worked Example avg-11-pt prec: Query 1, measured data points



- Blue for Query 1
- Bold Circles measured

Query 1				
Rank		R	P	
1	X	0.2	1.00	$\tilde{P}_1(r_2) = 1.00$
2				
3	X	0.4	0.67	$\tilde{P}_1(r_4) = 0.67$
4				
5				
6	X	0.6	0.50	$\tilde{P}_1(r_6) = 0.50$
7				
8				
9				
10	X	0.8	0.40	$\tilde{P}_1(r_8) = 0.40$
11				
12				
13				
14				
15				
16				
17				
18				
19				
20	X	1.0	0.25	$\tilde{P}_1(r_{10}) = 0.25$

- Five r_js ($r_2, r_4, r_6, r_8, r_{10}$) coincide directly with datapoint



- Bold circles measured
- **thin circles interpolated**

Query 1			
Rank		R	P
1	X	.20	1.00
2			
3	X	.40	.67
4			
5			
6	X	.60	.50
7			
8			
9			
10	X	.80	.40
11			
12			
13			
14			
15			
16			
17			
18			
19			
20	X	1.00	.25

$\tilde{P}_1(r_0) = 1.00$

$\tilde{P}_1(r_1) = 1.00$

$\tilde{P}_1(r_2) = 1.00$

$\tilde{P}_1(r_3) = .67$

$\tilde{P}_1(r_4) = .67$

$\tilde{P}_1(r_5) = .50$

$\tilde{P}_1(r_6) = .50$

$\tilde{P}_1(r_7) = .40$

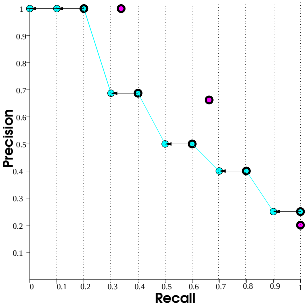
$\tilde{P}_1(r_8) = .40$

$\tilde{P}_1(r_9) = .25$

$\tilde{P}_1(r_{10}) = .25$

- The six other r_js (r_0, r_1, r_3, r_5, r_7, r_9) are interpolated.

Worked Example avg-11-pt prec: Query 2, measured data points



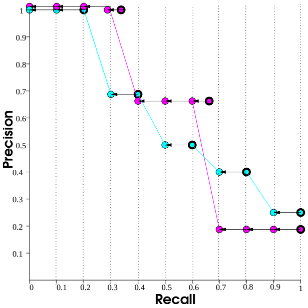
- Blue: Query 1; **Red: Query 2**
- **Bold circles measured**; thin circles interpol.

Query 2			
Rank	Relev.	R	P
1	X	.33	1.00
2			
3	X	.67	.67
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15	X	1.0	.2

$\tilde{P}_2(r_{10}) = .20$

- Only r_{10} coincides with a measured data point

Worked Example avg-11-pt prec: Query 2, interpolation



- Blue: Query 1; **Red: Query 2**
- Bold circles measured; **thin circles interpol.**

Query 2			
Rank	Relev.	R	P
1	X	.33	1.00
2			
3	X	.67	.67
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15	X	1.0	.2

$\tilde{P}_2(r_{10}) = .20$

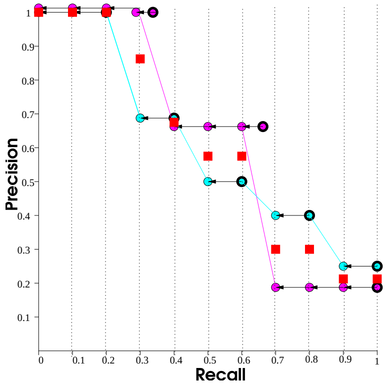
$\tilde{P}_2(r_0) = 1.00$
$\tilde{P}_2(r_1) = 1.00$
$\tilde{P}_2(r_2) = 1.00$
$\tilde{P}_2(r_3) = 1.00$

$\tilde{P}_2(r_4) = .67$
$\tilde{P}_2(r_5) = .67$
$\tilde{P}_2(r_6) = .67$

$\tilde{P}_2(r_7) = .20$
$\tilde{P}_2(r_8) = .20$
$\tilde{P}_2(r_9) = .20$

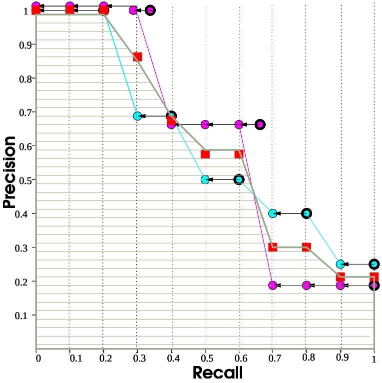
- 10 of the r_js are interpolated

Worked Example avg-11-pt prec: averaging



- Now average at each p_j
- over N (number of queries)
- $\rightarrow$ 11 averages

Worked Example avg-11-pt prec: area/result



- End result:
- 11 point average precision
- Approximation of area under prec. recall curve