IRRS: Information Retrieval and Recommender Systems

FIB, Master in Data Science

Slides by Marta Arias, José Luis Balcázar,
Ramon Ferrer-i-Cancho, Ricard Gavaldá
Department of Computer Science, UPC

Fall 2022

http://www.cs.upc.edu/~ir-miri

4. Evaluation, Relevance Feedback and LSI

Evaluation of Information Retrieval Usage, I

What are we exactly to do?

In the Boolean model, the specification is unambiguous:

We know what we are to do:

> Retrieve and provide to the user
>
> all those documents
>
> that satisfy the query.

But, is this what the user really wants?

> Sorry, but usually… no.

Evaluation of Information Retrieval Usage, II

Then, what exactly are we to optimize?

Notation:

$\mathcal{D}$: set of all our documents on which the user asks one query;

$\mathcal{A}$: answer set: documents that the system retrieves as answer;

$\mathcal{R}$: relevant documents: those that the user actually wishes to see as answer.

(But no one knows this set, not even the user!)

Unreachable goal: $\mathcal{A} = \mathcal{R}$, that is:

- $Pr(d \in \mathcal{A} | d \in \mathcal{R}) = 1$ and
- $Pr(d \in \mathcal{R} | d \in \mathcal{A}) = 1$.

The Recall and Precision measures

Let's settle for:

- high recall, $\frac{|\mathcal{R} \cap \mathcal{A}|}{|\mathcal{R}|}$:

  $Pr(d \in \mathcal{A} | d \in \mathcal{R})$ not too much below 1,

- high precision, $\frac{|\mathcal{R} \cap \mathcal{A}|}{|\mathcal{A}|}$:

  $Pr(d \in \mathcal{R} | d \in \mathcal{A})$ not too much below 1.

Difficult balance. More later.

Recall and Precision, II

Example: test for tuberculosis (TB)

- 1000 people, out of which 50 have TB
- test is positive on 40 people, of which 35 *really* have TB

Recall

% of true TB that test positive = 35 / 50 = 70 %

Precision

% of positives that really have TB = 35 / 40 = 87.5 %

- Large recall: few sick people go away undetected
- Large precision: few people are scared unnecessarily (few *false alarms*)

Recall and Precision, III. Confusion matrix

Equivalent definition

Confusion matrix

		Answered	
		relevant	not relevant
Reality	relevant	tp	fn
	not relevant	fp	tn

- $|\mathcal{R}| = tp + fn$
- $|\mathcal{A}| = tp + fp$
- $|\mathcal{R} \cap \mathcal{A}| = tp$

- Recall = $\frac{|\mathcal{R}\cap\mathcal{A}|}{|\mathcal{R}|} = \frac{tp}{tp+fn}$
- Precision = $\frac{|\mathcal{R}\cap\mathcal{A}|}{|\mathcal{A}|} = \frac{tp}{tp+fp}$

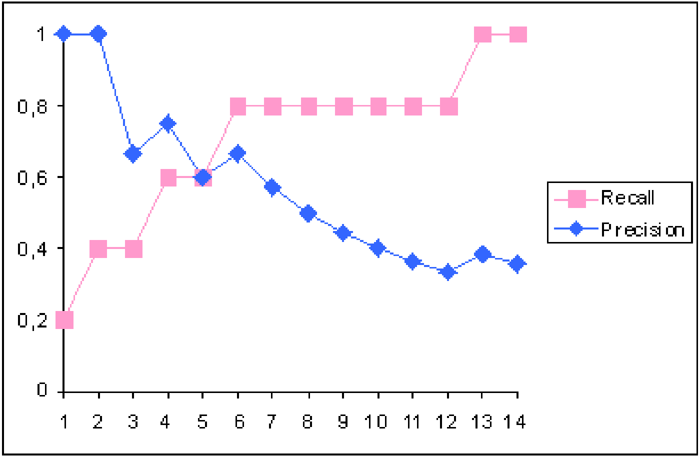
How many documents to show?

We rank all documents according to some measure.
How many should we show?

- Users won't read too large answers.
- Long answers are likely to exhibit low precision.
- Short answers are likely to exhibit low recall.

We analyze precision and recall as functions of the number of documents k provided as answer.

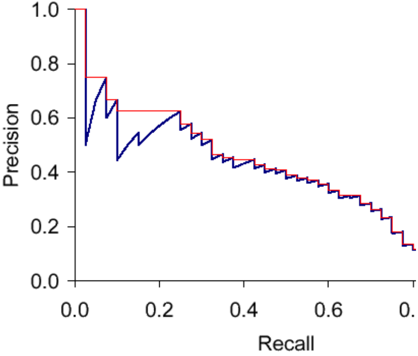
Rank-recall and rank-precision plots



(Source: Prof. J. J. Paijmans, Tilburg)

A single “precision and recall” curve

x-axis for recall, and y-axis for precision.
(Similar to, and related to, the ROC curve in predictive models.)



(Source: Stanford NLP group)
Often: Plot 11 points of interpolated precision, at 0 %, 10 %, 20 %, . . . , 100 % recall

Other measures of effectiveness

- AUC: Area under the curve of the plots above, relative to best possible
- F-measure: $\dfrac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}$
  - Harmonic mean. Closer to min of both than arithmetic mean
- α-F-measure: $\dfrac{2}{\frac{\alpha}{\text{recall}} + \frac{1-\alpha}{\text{precision}}}$

Other measures of effectiveness, II

Take into account *the documents previously known to the user*.

- Coverage:

  |relevant & known & retrieved| / |relevant & known|

- Novelty:

  |relevant & retrieved & UNknown| / |relevant & retrieved|

Relevance Feedback, I

Going beyond what the user asked for

The user relevance cycle:

1. Get a query q
2. Retrieve relevant documents for q
3. Show top k to user
4. Ask user to mark them as relevant / irrelevant
5. Use answers to refine q
6. If desired, go to 2

Relevance Feedback, II

How to create the new query?

Vector model: queries and documents are vectors
Given a query q, and a set of documents, split into relevant R and nonrelevant NR sets, build a new query q':

Rocchio's Rule:

$$q' = \alpha \cdot q + \beta \cdot \frac{1}{|R|} \cdot \sum_{d \in R} d - \gamma \cdot \frac{1}{|NR|} \cdot \sum_{d \in NR} d$$

- All vectors q and d's must be normalized (e.g., unit length).
- Weights α, β, γ, scalars, with $\alpha > \beta > \gamma \geq 0$; often $\gamma = 0$.
  - α: degree of trust on the original user's query,
  - β: weight of positive information (terms that do not appear on the query but do appear in relevant documents),
  - γ: weight of negative information.

Relevance Feedback, III

In practice, often:

- good improvement of the recall for first round,
- marginal for second round,
- almost none beyond.

In web search, precision matters much more than recall, so the extra computation time and user patience may not be productive.

Relevance Feedback, IV

. . . as Query Expansion

It is a form of Query Expansion:

The new query has non-zero weights on words
that were not in the original query

Pseudorelevance feedback

Do not ask anything from the user!

- User patience is precious resource. They'll just walk away.
- Assume you did great in answering the query!
- That is, top-k documents in the answer are all relevant
- No interaction with user
- But don't forget that the search will feel slower.
- Stop, at the latest, when you get the same top k documents.

Pseudorelevance feedback, II

Alternative sources of feedback / query refinement:

- Links clicked / not clicked on.
- Think time / time spent looking at item.
- User's previous history.
- Other users' preferences!
- Co-occurring words: Add words that often occur with words in the query - for query expansion.

Latent Semantic Indexing, I

Alternative to vector model using *dimensionality reduction*

Idea:

- Suppose that documents are about a (relatively small) number of concepts
- Compute similarity of each document to each concept
- Given query q, return docs about the same concepts as q

Latent Semantic Indexing, II

SVD theorem

Singular Value Decomposition (SVD) theorem from linear algebra makes this *formal*:

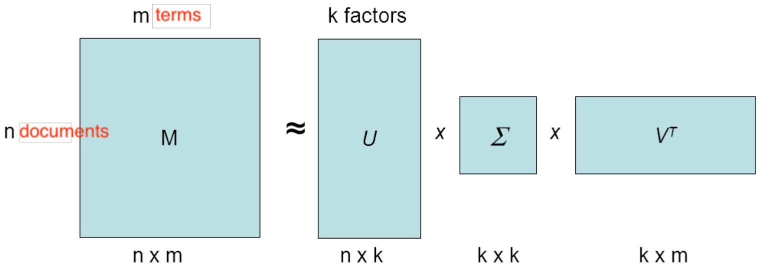
Theorem: Every $n \times m$ matrix M of rank K can be decomposed as $M = U \Sigma V^T$ where

- U is $n \times K$ and orthonormal
- V is $m \times K$ and normal
- Σ is $K \times K$ and diagonal

Furthermore, if we keep the $k < K$ highest values of Σ and zero the rest, we obtain the best approximation of M with a matrix of rank k

Latent Semantic Indexing, III

Interpretation



- There are k latent factors – "topics" or "concepts"
- U tells how much each user is affected by a factor
  - *document to concept* similarities
- V tells how much each item is related to a factor
  - *term to concept* similarities
- Σ tells the weight of each different factor
  - *strength* of each concept

Latent Semantic Indexing, IV

Computing similarity

For document-term matrix M, let m_{ij} be the weight of term t_j for document d_i (e.g. in tf-idf scheme). Then:

$$\begin{aligned}
sim(d_i, q) &= \sum_j m_{ij} \times q_j \\
&= \sum_j (U\Sigma V^T)_{ij} \times q_j \\
&= \sum_j (\sum_k (U\Sigma)_{ik} (V^T)_{kj}) \times q_j \\
&= \sum_{k,j} ((U\Sigma)_{ik} (V^T)_{kj}\ q_j) \\
&= \sum_k [(U\Sigma)_{ik} \times \sum_j ((V^T)_{kj}\ q_j)]
\end{aligned}$$

Which can be interpreted as the sum over all concepts k of product of similarity of d_i to concept k and similarity of query to concept k

Latent Semantic Indexing, V

- Can be seen as query expansion: Answer may contain documents using terms related to query words (synonims, or part of the same expression)
- LSI tends to increase recall at the expense of precision
- Feasible for small to mid-size collections