

Modern Information Retrieval

Probabilistic Information Retrieval¹

Hamid Beigy

Sharif university of technology

November 13, 2022



¹Some slides have been adapted from slides of Manning, Yannakoudakis, and Schütze.



1. Introduction
2. Relevance Feedback
3. Probabilistic Approach to IR
4. Probability Theory
5. Probability Ranking Principle
6. Appraisal and Extensions
7. References

Introduction



1. An **information need** may be expressed using different keywords (**synonymy**) such as **aircraft** vs **airplane**.
2. The **same word** can have different meanings (**polysemy**).
3. Vocabulary of searcher may not match that of the documents.
4. Solutions: **refining queries manually** or **expanding queries automatically**
5. **Relevance feedback** and **query expansion** aim to overcome the problem of **synonymy**.

Relevance Feedback



1. The user issues a (short, simple) query.
2. The search engine returns a set of documents.
3. User marks some docs as relevant, some as non-relevant.
4. Search engine computes a new representation of the information need (should be better than the initial query).
5. Search engine runs new query and returns new results.
6. New results have (hopefully) better recall.



1. Manual thesaurus (maintained by editors, e.g., PubMed)
2. Automatically derived thesaurus (e.g., based on co-occurrence statistics)
3. Query-equivalence based on query log mining (common on the web as in the “palm” example)

Probabilistic Approach to IR



1. Given a user information need (**query**) and a collection of documents (**document**), a system must determine how well the documents satisfy the query
2. An IR system has an **uncertain understanding** of the user query, and makes an **uncertain guess** of whether a document satisfies the query
3. Probability theory provides a principled foundation for such **reasoning under uncertainty**
4. Probabilistic models exploit this foundation to estimate how likely it is that a document is relevant to a query



- ▶ Vector space model: rank documents according to similarity to query.
- ▶ The notion of similarity does not translate directly into an assessment of **is the document a good document to give to the user or not?**
- ▶ The most similar document can be highly relevant or completely nonrelevant.
- ▶ Probability theory is arguably a cleaner formalization of what we really want an IR system to do: **give relevant documents to the user.**

Probability Theory



1. For events A and B
 - ▶ Joint probability $P(A \cap B)$ of both events occurring
 - ▶ Conditional probability $P(A|B)$ of event A occurring given that event B has occurred
2. **Chain rule** gives fundamental relationship between joint and conditional probabilities:

$$P(AB) = P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

3. Similarly for the complement of an event $P(\bar{A})$:

$$P(\bar{A}B) = P(B|\bar{A})P(\bar{A})$$

4. **Partition rule**: if B can be divided into an exhaustive set of disjoint subcases, then $P(B)$ is the sum of the probabilities of the subcases. A special case of this rule gives:

$$P(B) = P(AB) + P(\bar{A}B)$$



1. **Bayes' Rule** for inverting conditional probabilities:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \left[\frac{P(B|A)}{\sum_{X \in \{A, \bar{A}\}} P(B|X)P(X)} \right] P(A)$$

Can be thought of as a way of updating probabilities:

- ▶ Start off with **prior probability** $P(A)$ (initial estimate of how likely event A is in the absence of any other information)
 - ▶ Derive a **posterior probability** $P(A|B)$ after having seen the evidence B , based on the likelihood of B occurring in the two cases that A does or does not hold
2. **Odds** of an event provide a kind of multiplier for how probabilities change:

$$\text{Odds: } O(A) = \frac{P(A)}{P(\bar{A})} = \frac{P(A)}{1 - P(A)}$$

Probability Ranking Principle



1. Ranked retrieval setup: given a collection of documents, the user issues a query, and an ordered list of documents is returned
2. Assume binary notion of relevance: $R_{d,q}$ is a random variable, such that
 - ▶ $R_{d,q} = 1$ if document d is relevant w.r.t query q
 - ▶ $R_{d,q} = 0$ otherwise
3. Probabilistic ranking orders documents decreasingly by their estimated probability of relevance w.r.t. query: $P(R = 1|d, q)$
4. Assume that the relevance of each document is independent of the relevance of other documents



1. If the retrieved documents (w.r.t a query) are ranked decreasingly on their probability of relevance, then the effectiveness of the system will be the best that is obtainable
2. If [the IR] system's response to each [query] is a ranking of the documents [...] in order of decreasing probability of relevance to the [query], where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data



1. Traditionally used with the PRP
2. Assumptions:
 - ▶ **Binary(equivalent to Boolean)**: documents and queries represented as binary term incidence vectors
 - ▶ E.g., document d represented by vector $\vec{x} = (x_1, \dots, x_M)$, where $x_t = 1$ if term t occurs in d and $x_t = 0$ otherwise
 - ▶ Different documents may have the same vector representation
 - ▶ **Independence**: no association between terms (not true, but practically works - **naive** assumption of Naive Bayes models)



	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	1	1	0	0	0	1	
BRUTUS	1	1	0	1	0	0	
CAESAR	1	1	0	1	1	1	
CALPURNIA	0	1	0	0	0	0	
CLEOPATRA	1	0	0	0	0	0	
MERCY	1	0	1	1	1	1	
WORSER	1	0	1	1	1	0	
...							

Each document is represented as a **binary vector** $\in \{0, 1\}^{|V|}$.



1. To make a probabilistic retrieval strategy precise, **need to estimate how terms in documents contribute to relevance**
2. Find measurable statistics (term frequency, document frequency, document length) that affect judgments about document relevance
3. Combine these statistics to estimate the probability $P(R|d, q)$ of document relevance
4. How exactly we can do this?



1. $P(R|d, q)$ is modeled using term incidence vectors as $P(R|\vec{x}, \vec{q})$

$$P(R = 1|\vec{x}, \vec{q}) = \frac{P(\vec{x}|R = 1, \vec{q})P(R = 1|\vec{q})}{P(\vec{x}|\vec{q})}$$

$$P(R = 0|\vec{x}, \vec{q}) = \frac{P(\vec{x}|R = 0, \vec{q})P(R = 0|\vec{q})}{P(\vec{x}|\vec{q})}$$

- ▶ $P(\vec{x}|R = 1, \vec{q})$ and $P(\vec{x}|R = 0, \vec{q})$: probability that if a relevant or nonrelevant document is retrieved, then that document's representation is \vec{x}
- ▶ Use statistics about the document collection to estimate these probabilities



1. $P(R|d, q)$ is modeled using term incidence vectors as $P(R|\vec{x}, \vec{q})$

$$P(R = 1|\vec{x}, \vec{q}) = \frac{P(\vec{x}|R = 1, \vec{q})P(R = 1|\vec{q})}{P(\vec{x}|\vec{q})}$$

$$P(R = 0|\vec{x}, \vec{q}) = \frac{P(\vec{x}|R = 0, \vec{q})P(R = 0|\vec{q})}{P(\vec{x}|\vec{q})}$$

- ▶ $P(R = 1|\vec{q})$ and $P(R = 0|\vec{q})$: prior probability of retrieving a relevant or nonrelevant document for a query \vec{q}
- ▶ Estimate $P(R = 1|\vec{q})$ and $P(R = 0|\vec{q})$ from percentage of relevant documents in the collection
- ▶ Since a document is either relevant or nonrelevant to a query, we must have that:

$$P(R = 1|\vec{x}, \vec{q}) + P(R = 0|\vec{x}, \vec{q}) = 1$$



1. Given a query q , ranking documents by $P(R = 1|d, q)$ is modeled under BIM as ranking them by $P(R = 1|\vec{x}, \vec{q})$
2. Rank documents by their odds of relevance (gives same ranking)

$$\begin{aligned} O(R|\vec{x}, \vec{q}) &= \frac{P(R = 1|\vec{x}, \vec{q})}{P(R = 0|\vec{x}, \vec{q})} = \frac{\frac{P(R=1|\vec{q})P(\vec{x}|R=1,\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(R=0|\vec{q})P(\vec{x}|R=0,\vec{q})}{P(\vec{x}|\vec{q})}} \\ &= \frac{P(R = 1|\vec{q})}{P(R = 0|\vec{q})} \cdot \frac{P(\vec{x}|R = 1, \vec{q})}{P(\vec{x}|R = 0, \vec{q})} \end{aligned}$$

3. $\frac{P(R=1|\vec{q})}{P(R=0|\vec{q})}$ is a constant for a given query - can be ignored



1. It is at this point that we make the **Naive Bayes conditional independence assumption** that the presence or absence of a word in a document is independent of the presence or absence of any other word (given the query):

$$\frac{P(\vec{x}|R = 1, \vec{q})}{P(\vec{x}|R = 0, \vec{q})} = \prod_{t=1}^M \frac{P(x_t|R = 1, \vec{q})}{P(x_t|R = 0, \vec{q})}$$

2. So we obtain

$$O(R|\vec{x}, \vec{q}) = O(R|\vec{q}) \cdot \prod_{t=1}^M \frac{P(x_t|R = 1, \vec{q})}{P(x_t|R = 0, \vec{q})}$$



1. Since each x_t is either 0 or 1, we can separate the terms:

$$O(R|\vec{x}, \vec{q}) = O(R|\vec{q}) \times \prod_{t:x_t=1} \frac{P(x_t = 1|R = 1, \vec{q})}{P(x_t = 1|R = 0, \vec{q})} \times \prod_{t:x_t=0} \frac{P(x_t = 0|R = 1, \vec{q})}{P(x_t = 0|R = 0, \vec{q})}$$



1. Let $p_t = P(x_t = 1 | R = 1, \vec{q})$ be the probability of a term appearing in relevant document
2. Let $u_t = P(x_t = 1 | R = 0, \vec{q})$ be the probability of a term appearing in a nonrelevant document
3. Can be displayed as contingency table:

	document	relevant ($R = 1$)	nonrelevant ($R = 0$)
Term present	$x_t = 1$	p_t	u_t
Term absent	$x_t = 0$	$1 - p_t$	$1 - u_t$



1. Additional simplifying assumption: terms not occurring in the query are equally likely to occur in relevant and nonrelevant documents

If $q_t = 0$, then $p_t = u_t$

2. Now we need only to consider terms in the products that appear in the query:

$$O(R|\vec{x}, \vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=0, q_t=1} \frac{1-p_t}{1-u_t}$$

3. The left product is over query terms found in the document and the right product is over query terms not found in the document



1. Including the query terms found in the document into the right product, but simultaneously dividing by them in the left product, gives:

$$\begin{aligned} O(R|\vec{x}, \vec{q}) &= O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=q_t=1} \frac{1-u_t}{1-p_t} \cdot \frac{1-p_t}{1-u_t} \cdot \prod_{\substack{t:x_t=0 \\ q_t=1}} \frac{1-p_t}{1-u_t} \\ &= O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} \cdot \prod_{t:q_t=1} \frac{1-p_t}{1-u_t} \end{aligned}$$

2. The left product is still over query terms found in the document, but the right product is now over all query terms, hence constant for a particular query and can be ignored.
3. The only quantity that needs to be estimated to rank documents w.r.t a query is the left product
4. Hence the **Retrieval Status Value** (RSV) in this model:

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$



1. Equivalent: rank documents using the **log odds ratios** for the terms in the query c_t :

$$c_t = \log \frac{p_t(1 - u_t)}{u_t(1 - p_t)} = \log \frac{p_t}{(1 - p_t)} - \log \frac{u_t}{1 - u_t}$$

2. The **odds ratio** is the ratio of two odds:
 - ▶ the odds of the term appearing if the document is relevant ($p_t/(1 - p_t)$), and
 - ▶ the odds of the term appearing if the document is nonrelevant ($u_t/(1 - u_t)$)
3. $c_t = 0$: term has equal odds of appearing in relevant and nonrelevant docs
4. c_t **positive**: higher odds to appear in relevant documents
5. c_t **negative**: higher odds to appear in nonrelevant documents



1. $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t}$ functions as a term weight.
2. Retrieval status value for document d : $RSV_d = \sum_{x_t=q_t=1} c_t$.
3. So BIM and vector space model are identical on an operational level except that the term weights are different.
4. In particular: we can use the same data structures (inverted index etc) for the two models.



1. For each term t in a query, estimate c_t in the whole collection using a contingency table of counts of documents in the collection, where df_t is the number of documents that contain term t :

	documents	relevant	nonrelevant	Total
Term present	$x_t = 1$	s	$df_t - s$	df_t
Term absent	$x_t = 0$	$S - s$	$(N - df_t) - (S - s)$	$N - df_t$
Total		S	$N - S$	N

$$p_t = s/S$$

$$u_t = (df_t - s)/(N - S)$$

$$c_t = K(N, df_t, S, s) = \log \frac{s/(S - s)}{(df_t - s)/((N - df_t) - (S - s))}$$



1. If any of the counts is a zero, then the term weight is not well-defined.
2. Maximum likelihood estimates do not work for rare events.
3. To avoid zeros: **add 0.5 to each count**
4. For example, use $S - s + 0.5$ in formula for $S - s$



1. Assuming that relevant documents are a very small percentage of the collection, approximate statistics for nonrelevant documents by statistics from the whole collection
2. Hence, u_t (the probability of term occurrence in nonrelevant documents for a query) is df_t/N and

$$\log[(1 - u_t)/u_t] = \log[(N - df_t)/df_t] \approx \log N/df_t$$

3. This should look familiar: **idf**.
4. The above approximation cannot easily be extended to relevant documents. (**why?** check it.)



1. Statistics of relevant documents (p_t) in relevance feedback can be estimated using maximum likelihood estimation or add 0.5.
 - ▶ Use the frequency of term occurrence in known relevant documents.
2. This is the basis of probabilistic approaches to relevance feedback weighting in a feedback loop

Appraisal and Extensions



1. Among the oldest formal models in IR
 - ▶ Maron & Kuhns, 1960: Since an IR system cannot predict with certainty which document is relevant, we should deal with probabilities
2. Assumptions for getting reasonable approximations of the needed probabilities (in the BIM):
 - ▶ Boolean representation of documents/queries/relevance
 - ▶ Term independence
 - ▶ Out-of-query terms do not affect retrieval
 - ▶ Document relevance values are independent



- ▶ They are not that different.
- ▶ In either case you build an information retrieval scheme in the exact same way.
- ▶ For probabilistic IR, at the end, you score queries not by cosine similarity and tf-idf in a vector space, but by a slightly different formula motivated by probability theory.
- ▶ How to add term frequency and length normalization to the probabilistic model.



1. Okapi BM25 is a probabilistic model that incorporates term frequency (i.e., it's nonbinary) and length normalization.
2. BIM was originally designed for short catalog records of fairly consistent length, and it works reasonably in these contexts
3. For modern full-text search collections, a model should pay attention to term frequency and document length
4. BestMatch25 (a.k.a [BM25](#) or [Okapi](#)) is sensitive to these quantities
5. BM25 is one of the most widely used and robust retrieval models



1. The simplest score for document d is just idf weighting of the query terms present in the document:

$$RSV_d = \sum_{t \in q} \log \frac{N}{df_t}$$



1. Improve idf term $[\log N/df]$ by factoring in term frequency and document length.

$$RSV_d = \sum_{t \in q} \log \left[\frac{N}{df_t} \right] \cdot \frac{(k_1 + 1)tf_{td}}{k_1((1 - b) + b \times (L_d/L_{ave})) + tf_{td}}$$

2. tf_{td} : term frequency in document d
3. L_d (L_{ave}): length of document d (average document length in the whole collection)
4. k_1 : tuning parameter controlling the document term frequency scaling
5. b : tuning parameter controlling the scaling by document length



1. For long queries, use similar weighting for query terms

$$RSV_d = \sum_{t \in q} \left[\log \frac{N}{df_t} \right] \cdot \frac{(k_1 + 1)tf_{td}}{k_1((1 - b) + b \times (L_d/L_{ave})) + tf_{td}} \cdot \frac{(k_3 + 1)tf_{tq}}{k_3 + tf_{tq}}$$

2. tf_{tq} : term frequency in the query q
3. k_3 : tuning parameter controlling term frequency scaling of the query
4. No length normalization of queries (because retrieval is being done with respect to a single fixed query)
5. The above tuning parameters should ideally be set to optimize performance on a development test collection. In the absence of such optimization, experiments have shown reasonable values are to set k_1 and k_3 to a value between 1.2 and 2 and $b = 0.75$



1. I want something basic and simple: use vector space with tf-idf weighting.
2. I want to use a state-of-the-art ranking model with excellent performance: use language models or BM25 with **tuned parameters**

References



1. Chapter 11 of [Information Retrieval Book](#)²

²Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.



Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

