Information Retrieval and Web Search

Cornelia Caragea

Computer Science
University of Illinois at Chicago

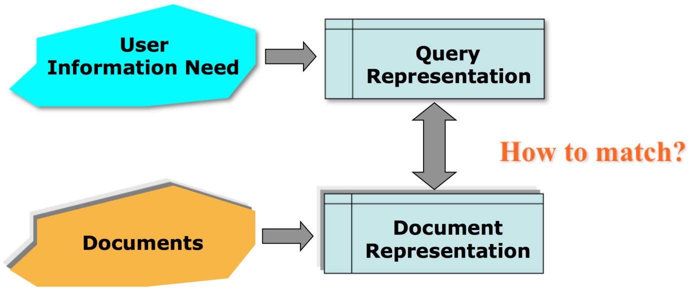
Credits for slides: Hofmann, Mobasher, Mooney, Schutze

Probabilistic Models for IR

Required Reading

- “Information Retrieval” textbook
    - Chapter 11: Probabilistic Information Retrieval

IR Systems



- An IR system tries to determine how well documents satisfy information needs.
- Given a query, an IR system has an uncertain understanding of the information need.
- Given the query and document representations, a system has an uncertain guess of whether a document has content relevant to the information need.

Why Probabilities in IR?

- In traditional IR systems, matching between each document and query is attempted in a semantically imprecise space of index terms.
- Probabilities provide a principled foundation for uncertain reasoning.
- **Can we use probabilities to quantify our uncertainties?**

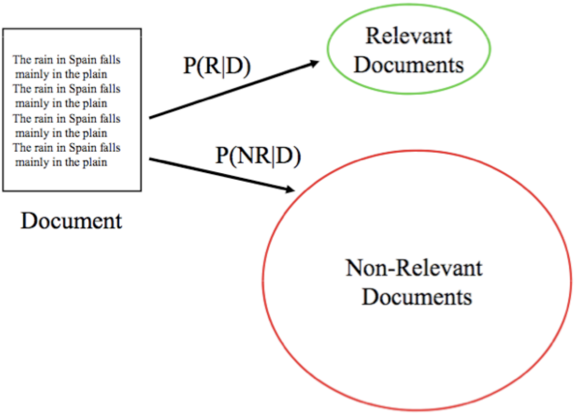
Probabilistic IR Topics

- Classical probabilistic retrieval model
    - Probability ranking principle, Binary Independence Model
- Okapi BM25 weighting scheme
- Bayesian networks for text retrieval
- Probabilistic language model approach to IR
- (Naïve) Bayesian Text Categorization

Basic Probabilistic Retrieval Model

- Retrieval is modeled as a classification process
- Two classes for each query: the relevant and non-relevant documents
- Given a particular document D, calculate the probability of belonging to the relevant class, retrieve if greater than probability of belonging to non-relevant class
  - i.e., retrieve if $P(R|D) > P(NR|D)$
- Different ways of estimating these probabilities lead to different probabilistic models.

Basic Probabilistic Model


The rain in Spain falls mainly in the plain The rain in Spain falls mainly in the plain The rain in Spain falls mainly in the plain The rain in Spain falls mainly in the plain
Document
P(R|D)
P(NR|D)
Relevant Documents
Non-Relevant Documents

- Present documents to a user ranked by their estimated probability of relevance w.r.t. the information need, P(R|D).
    - Basis of Probability Ranking Principle (van Rijsbergen 1979)

The Probability Ranking Principle

“If a reference retrieval system’s response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, 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.”

van Rijsbergen (1979)

The Probability Ranking Principle

- Let d be a document in the collection.
- Let R represent relevance of a document w.r.t. a given (fixed) query and let NR represent non-relevance.
  - $R = \{0, 1\}$
- Need to find $P(R|d)$ - probability that a document d is relevant.

$$P(R|d) = \frac{P(d|R)P(R)}{P(d)} \text{ and } P(NR|d) = \frac{P(d|NR)P(NR)}{P(d)}$$

  - $P(R), P(NR)$ - prior probability of retrieving a (non)relevant document
  - $P(R|d) + P(NR|d) = 1$
  - $P(d|R), P(d|NR)$ - probability that if a relevant (non-relevant) document is retrieved, it is d.

The Probability Ranking Principle (PRP)

- Simple case: no selection costs or other utility concerns that would differentially weight errors
  - You lose a point for either returning a non-relevant document or failing to return a relevant document - **1/0 loss**.
- Bayes' Optimal Decision Rule
  - d is relevant iff $P(R|d) > P(NR|d)$
  - [Return documents that are more likely relevant than non-relevant.]

Probability Ranking Principle - With Costs

- Assuming retrieval costs:
  - Let d be a document
  - C_1 - cost of not retrieving a relevant document
  - C_0 - cost of retrieving a non-relevant document
- Probability Ranking Principle: if for a specific document d and for all documents d' not yet retrieved, the following holds:

$$C_0 \cdot P(NR|d) - C_1 \cdot P(R|d) \leq C_0 \cdot P(NR|d') - C_1 \cdot P(R|d'),$$

then d **is the next document to be retrieved.**

Binary Independence Retrieval

- How do we compute all those probabilities?
  - Do not know exact probabilities, have to use estimates
  - Binary Independence Retrieval (BIR) - the simplest model
- Questionable assumptions
  - "Relevance" of each document is independent of relevance of other documents
    - Especially harmful in practice, if a system is allowed to return duplicate or near duplicate documents.
  - The user has a single step information need
    - Seeing a range of results might let user refine query
  - BIR assumptions
    - Documents and queries are represented as binary term incidence vectors (0/1).
    - Terms in a document are independent (naive assumption of the Naïve Bayes model).

Binary Independence Model

- Traditionally used in conjunction with PRP
- "Binary" = Boolean: documents are represented as binary incidence vectors of terms:
  - $\vec{x} = (x_1, \cdots, x_n)$
  - $x_i = 1$ iff term i is present in document d having representation x.
- "Independence": terms occur in documents independently
  - Different documents can be modeled as the same vector

Probabilistic Retrieval Strategy

To make a probabilistic retrieval strategy precise, we need to:

- Estimate how terms in documents contribute to relevance
    - How do things such as term frequency, document frequency, document length, and other statistics influence judgments about document relevance?
    - How terms can be reasonably combined to find document relevance probability?
- Order documents by decreasing estimated probability of relevance.

Binary Independence Model

- Queries: binary term incidence vectors
- Given query q,
  - For each document d, need to compute $P(R|q,d)$.
  - Replace with computing $P(R|\vec{q},\vec{x})$ where $\vec{x}$ is binary term incidence vector representing d.
  - **Recall: we are interested in the ranking of documents.**
- We will use Bayes Rule and the ratio of probability of relevance to probability of non-relevance, aka odds of relevance:
  - The odds of relevance is monotonic with the probability of relevance.
  - Easier to compute and gives the same ordering of documents.

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

Binary Independence Model

$$O(R|\vec{q},\vec{x}) = \frac{P(R|\vec{q},\vec{x})}{P(NR|\vec{q},\vec{x})} = \frac{P(R|\vec{q})}{P(NR|\vec{q})} \cdot \frac{P(\vec{x}|R,\vec{q})}{P(\vec{x}|NR,\vec{q})}$$

- $\frac{P(R|\vec{q})}{P(NR|\vec{q})} = O(R|\vec{q})$ - constant for a given query
- $\frac{P(\vec{x}|R,\vec{q})}{P(\vec{x}|NR,\vec{q})}$ - needs estimation

Binary Independence Model

$$O(R|\vec{q},\vec{x}) = \frac{P(R|\vec{q},\vec{x})}{P(NR|\vec{q},\vec{x})} = \frac{P(R|\vec{q})}{P(NR|\vec{q})} \cdot \frac{P(\vec{x}|R,\vec{q})}{P(\vec{x}|NR,\vec{q})}$$

- $\frac{P(R|\vec{q})}{P(NR|\vec{q})} = O(R|\vec{q})$ - constant for a given query
- $\frac{P(\vec{x}|R,\vec{q})}{P(\vec{x}|NR,\vec{q})}$ - needs estimation
- Using **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):

$$\frac{P(\vec{x}|R,\vec{q})}{P(\vec{x}|NR,\vec{q})} = \prod_{i=1}^{n} \frac{P(x_i|R,\vec{q})}{P(x_i|NR,\vec{q})}$$

So:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{i=1}^{n} \frac{P(x_i|R,\vec{q})}{P(x_i|NR,\vec{q})}$$

Binary Independence Model

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{i=1}^{n} \frac{P(x_i|R,\vec{q})}{P(x_i|NR,\vec{q})}$$

- Since x_i is either 0 or 1:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{x_i=1} \frac{P(x_i=1|R,\vec{q})}{P(x_i=1|NR,\vec{q})} \cdot \prod_{x_i=0} \frac{P(x_i=0|R,\vec{q})}{P(x_i=0|NR,\vec{q})}$$

- Denote:

$$p_i = P(x_i=1|R,\vec{q}), u_i = P(x_i=1|NR,\vec{q})$$

Binary Independence Model

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{i=1}^{n} \frac{P(x_i|R,\vec{q})}{P(x_i|NR,\vec{q})}$$

- Since x_i is either 0 or 1:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{x_i=1} \frac{P(x_i=1|R,\vec{q})}{P(x_i=1|NR,\vec{q})} \cdot \prod_{x_i=0} \frac{P(x_i=0|R,\vec{q})}{P(x_i=0|NR,\vec{q})}$$

- Denote:

$$p_i = P(x_i=1|R,\vec{q}), u_i = P(x_i=1|NR,\vec{q})$$

- Assume, for all terms not occurring in the query ($q_i = 0$):

$$p_i = u_i$$

That is, terms not occurring in the query are equally likely to occur in relevant and non-relevant documents.

Binary Independence Model

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{\boxed{x_i = q_i = 1}} \frac{p_i}{u_i} \cdot \prod_{\boxed{x_i = 0,\, q_i = 1}} \frac{1-p_i}{1-u_i}$$

All matching terms

Non-matching query terms

Binary Independence Model

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{\underbrace{\boxed{x_i = q_i = 1}}_{\text{All matching terms}}} \frac{p_i}{u_i} \cdot \prod_{\underbrace{\boxed{x_i = 0, q_i = 1}}_{\text{Non-matching query terms}}} \frac{1-p_i}{1-u_i}$$

$$= O(R|\vec{q}) \cdot \prod_{\underbrace{\boxed{x_i = q_i = 1}}_{\text{All matching terms}}} \frac{p_i(1-u_i)}{u_i(1-p_i)} \cdot \prod_{\underbrace{\boxed{q_i = 1}}_{\text{All query terms}}} \frac{1-p_i}{1-u_i}$$

- Left product - over query terms found in the document.
- Right product - over all query terms - does not depend on the document - constant for a particular query.

Binary Independence Model

Constant for each query

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{\boxed{x_i = q_i = 1}} \frac{p_i(1-u_i)}{u_i(1-p_i)} \cdot \prod_{\boxed{q_i = 1}} \frac{1-p_i}{1-u_i}$$

Only quantity to be estimated for rankings

- Rank documents by the log of the above quantity, called **Retrieval Status Value (RSV):**

$$\text{RSV} = \log \prod_{\boxed{x_i = q_i = 1}} \frac{p_i(1-u_i)}{u_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

Binary Independence Model

- All boils down to computing RSV.

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-u_i)}{u_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

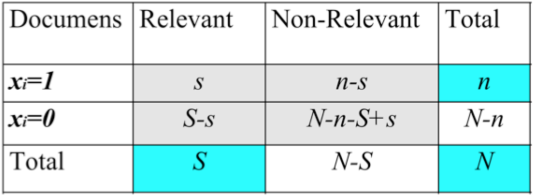
$$RSV = \sum_{x_i=q_i=1} c_i; \text{ where } c_i = \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

$$c_i = \log \frac{p_i}{(1-p_i)} + \log \frac{1-u_i}{u_i}$$

- So, how do we compute c_i's from our data?

Binary Independence Model

- Estimating RSV coefficients.
- For each term i look at this table of document counts:



Documens	Relevant	Non-Relevant	Total
x_i=1	s	n-s	n
x_i=0	S-s	N-n-S+s	N-n
Total	S	N-S	N

- Estimates: $p_i \approx \frac{s}{S}$, $u_i \approx \frac{n-s}{N-S}$

$$c_i \approx \log \frac{s/(S-s)}{(n-s)/(N-n-S+s)}$$

- For now, assume no probability of zeroes (such as if every or no relevant document has a particular term).

Estimation - Key Challenge - u_i

- Under the assumption that relevant documents are a very small percentage of the collection, it is plausible to approximate statistics for non-relevant documents by statistics from the whole collection.
- If non-relevant documents are approximated by the whole collection, then u_i (prob. of occurrence in non-relevant documents for query) is n/N (where $n = df_i$) and $\log(1 - u_i)/u_i = \log(N - n)/n \approx \log N/n = IDF$

Estimation - Key Challenge - p_i

- p_i (probability of occurrence in relevant documents) can be estimated in various ways:
    - Can be a constant (Croft and Harper, 1979) - each term has even odds of appearing in a relevant document, e.g., $p_i = 0.5$ - then just get idf weighting of terms
        - The document ranking is determined simply by which query terms occur in documents scaled by their idf weighting.
    - Proportional to the probability of occurrence in collection
        - More accurate (Greiff, SIGIR 1998)

Iteratively Estimating p_i

1. Assume that p_i constant over all x_i in query
   - $p_i = 0.5$ (each term has even odds of appearing in a relevant document).
2. Determine a guess for the size of the relevant document set:
   - V - fixed size set of highest ranked documents on this model.
3. We need to improve our guesses for p_i and u_i:
   - Use the distribution of x_i in documents in V. Let V_i be set of documents containing x_i.

$$p_i = \frac{|V_i|}{|V|}$$

   - Assume that documents that are not retrieved are nonrelevant:

$$u_i = \frac{n_i - |V_i|}{N - |V|}$$

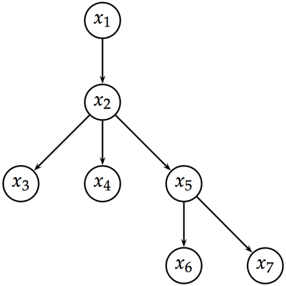
4. Go to 2. until converges, then return ranking

PRP and BIR

- Getting reasonable approximations of probabilities is possible.
- Requires restrictive assumptions:
    - **Term independence**
    - **Terms not in the query do not affect the outcome**
    - **Boolean representation of documents/queries/relevance**
    - **Document relevance values are independent**
- Some of these assumptions can be removed

Removing Term Independence

- In general, index terms are not independent
    - Term pairs such as *Hong* and *Kong*
    - Dependencies can be complex
- van Rijsbergen (1979) proposed a model which allowed a tree structure of term dependencies
- Each term can be directly dependent on only one other term, giving a tree structure of dependencies.



A tree of dependencies between terms.

- x_i depends on x_k if there is an arrow $x_k \rightarrow x_i$.

Okapi BM25: a non-binary model

- BIM - originally designed for short catalog records and abstracts of fairly consistent length
- For modern full-text search collections, term frequency and document length are important - Okapi BM25
- Retrieval Status Value:

$$RSV^{(d)} = \sum_{i \in q} \log \frac{N}{df_i} \cdot \frac{(k_1 + 1)tf_i^{(d)}}{k_1((1-b) + b \cdot (L_d/L_{avg})) + tf_i^{(d)}}$$

  - $tf_i^{(d)}$ - frequency of term i in document d.
  - L_d and L_{avg} - length of document d and the average document length for the whole collection
  - k_1 - positive tuning parameter that calibrates the document term frequency scaling.
  - b is another tuning parameter $(0 \leq b \leq 1)$ which determines the scaling by document length

Good and Bad News

- Standard Vector Space Model
    - Empirical for the most part; success measured by results
- Probabilistic Model
    - Advantages
        - Based on a firm theoretical foundation
        - Theoretically justified optimal ranking scheme
    - Disadvantages
        - Making the initial guess to get V
        - Binary word-in-doc weights (not using term frequencies)
        - Independence of terms (can be alleviated)
        - Amount of computation
        - Has never worked convincingly better in practice, but still an active research area