The document ranking problem

- We have a collection of documents
- User issues a query
- A list of documents needs to be returned
- **Ranking method is core of an IR system:**
  - **In what order do we present documents to the user?**
  - We want the "best" document to be first, second best second, etc….
- **Idea: Rank by probability of relevance of the document w.r.t. information need**
  - P(relevant|document$_i$, query)

Recall a few probability basics

Bayes' Rule: For events *a* and *b,*

$$p(a,b) = p(a \cap b) = p(a \mid b)\,p(b) = p(b \mid a)\,p(a)$$

$$p(\bar{a} \mid b)\,p(b) = p(b \mid \bar{a})\,p(\bar{a})$$

$$p(a \mid b) = \frac{p(b \mid a)\,p(a)}{p(b)} = \frac{p(b \mid a)\,p(a)}{\sum_{x=a,\bar{a}} p(b \mid x)\,p(x)}$$

Prior

Posterior

Odds: $$O(a) = \frac{p(a)}{p(\bar{a})} = \frac{p(a)}{1 - p(a)}$$

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."

[1960s/1970s] S. Robertson, W.S. Cooper, M.E. Maron;
van Rijsbergen (1979:113); Manning & Schütze (1999:538)

Probability Ranking Principle

Let *x* be a document in the collection.
Let *R* represent **relevance** of a document w.r.t. given (fixed) query and let *NR* represent **non-relevance**.

R={0,1} vs. NR/R

Need to find p(*R/x)* - probability that a document *x* is **relevant.**

$$p(R \mid x) = \frac{p(x \mid R)\, p(R)}{p(x)}$$

$$p(NR \mid x) = \frac{p(x \mid NR)\, p(NR)}{p(x)}$$

p(R),p(NR) - prior probability of retrieving a (non) relevant document

$$p(R \mid x) + p(NR \mid x) = 1$$

p(x|R), p(x|NR) - probability that if a relevant (non-relevant) document is retrieved, it is *x*.

Probability Ranking Principle (PRP)

- Simple case: no selection costs or other utility concerns that would differentially weight errors
- ***Bayes' Optimal Decision Rule***
  - ***x*** is **relevant** iff $p(R|x) > p(NR|x)$
- PRP in action: Rank all documents by $p(R|x)$
- Theorem:
  - Using the PRP is optimal, in that it minimizes the loss (Bayes risk) under 1/0 loss
  - Provable if all probabilities correct, etc. [e.g., Ripley 1996]

Probability Ranking Principle

- More complex case: retrieval costs.
  - Let d be a document
  - C - cost of retrieval of relevant document
  - C' - cost of retrieval of non-relevant document
- Probability Ranking Principle: if

$$C \cdot p(R|d) + C' \cdot (1 - p(R|d)) \le C \cdot p(R|d') + C' \cdot (1 - p(R|d'))$$

for all d' *not yet retrieved*, then d **is the next document to be retrieved**

- **We won't further consider loss/utility from now on**

Probability Ranking Principle

- How do we compute all those probabilities?
  - Do not know exact probabilities, have to use estimates
  - Binary Independence Retrieval (BIR) is the simplest model
- Questionable assumptions
  - "Relevance" of each document is independent of relevance of other documents.
    - Really, it's bad to keep on returning **duplicates**
  - Boolean model of relevance
  - That one has a single step information need
    - Seeing a range of results might let user refine query

Probabilistic Retrieval Strategy

- Estimate how terms contribute to relevance
  - How do things like tf, df, and length influence your judgments about document relevance?
- Combine to find document relevance probability
- Order documents by decreasing probability

Probabilistic Ranking

Basic concept:

"For a given query, if we know some documents that are relevant, terms that occur in those documents should be given greater weighting in searching for other relevant documents.

By making assumptions about the distribution of terms and applying Bayes Theorem, it is possible to derive weights theoretically."

Van Rijsbergen

PM are based on the hypothesis that the *distribution* of term in relevant document is *different* from the one in irrelevant documents

Then,

- A greater importance should be given to terms that occur in many relevant documents and are absent in many irrelevant documents
- A smaller importance should be given to terms that occur in many irrelevant documents and are absent in many relevant documents

Binary Independence Model
Robertson & Spark Jones (1976)

- Traditionally used in conjunction with PRP
- **"Binary" = Boolean**: documents are represented as binary incidence vectors of terms (cf. lecture 1):
  - $\vec{x} = (x_1, \ldots, x_n)$
  - $x_i = 1$ iff term *i* is present in document *x*.
- **"Independence"**: terms occur in documents independently
- Different documents can be modeled as same vector
- Bernoulli Naive Bayes model (cf. text categorization!)

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|q,x)*** where ***x*** is binary term incidence vector representing ***d*** Interested only in ranking
- Will use odds and Bayes' Rule:

$$O(R \mid q, \vec{x}) = \frac{p(R \mid q, \vec{x})}{p(NR \mid q, \vec{x})} = \frac{\dfrac{p(R \mid q)\, p(\vec{x} \mid R, q)}{p(\vec{x} \mid q)}}{\dfrac{p(NR \mid q)\, p(\vec{x} \mid NR, q)}{p(\vec{x} \mid q)}}$$

Binary Independence Model

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

Constant for a given query

Needs estimation

- Using **Independence** Assumption:

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

- So: $O(R \mid q, d) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$

Binary Independence Model

$$O(R \mid q, d) = O(R \mid q) \cdot \prod_{i=1}^{n} \frac{p(x_i \mid R, q)}{p(x_i \mid NR, q)}$$

- Since x_i is either 0 or 1:

$$O(R \mid q, d) = O(R \mid q) \cdot \prod_{x_i = 1} \frac{p(x_i = 1 \mid R, q)}{p(x_i = 1 \mid NR, q)} \cdot \prod_{x_i = 0} \frac{p(x_i = 0 \mid R, q)}{p(x_i = 0 \mid NR, q)}$$

- Let $p_i = p(x_i = 1 \mid R, q); \quad r_i = p(x_i = 1 \mid NR, q);$

- Assume, for all terms not occurring in the query ($q_i = 0$) $\quad p_i = r_i$

This can be changed (e.g., in relevance feedback)

Then...

Binary Independence Model

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

All matching terms

Non-matching query terms

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

All matching terms

All query terms

Binary Independence Model

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

Constant for each query

Only quantity to be estimated for rankings

- Retrieval Status Value:

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

Binary Independence Model

- All boils down to computing RSV.

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

$$RSV = \sum_{x_i = q_i = 1} c_i; \qquad c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

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:

Documents	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: $p_i \approx \frac{s}{S} \qquad r_i \approx \frac{(n-s)}{(N-S)}$

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

For now, assume no zero terms.

Estimation – key challenge

- If non-relevant documents are approximated by the whole collection, then r_i (prob. of occurrence in non-relevant documents for query) *is* n/N and
  - $\log (1- r_i)/r_i = \log (N- n)/n \approx \log N/n$ = IDF!
- p_i (probability of occurrence in relevant documents) can be estimated in various ways:
  - from relevant documents if know some
    - Relevance weighting can be used in feedback loop
  - constant (Croft and Harper combination match) – then just get idf weighting of terms
  - proportional to prob. of occurrence in collection
    - more accurately, to log of this (Greiff, SIGIR 1998)

Iteratively estimating p_i

1. Assume that p_i constant over all x_i in query
   - p_i = 0.5 (even odds) for any given doc
2. Determine guess of relevant document set:
   - V is fixed size set of highest ranked documents on this model (note: now a bit like tf.idf!)
3. We need to improve our guesses for p_i and r_i, so
   - Use distribution of x_i in docs in V. Let V_i be set of documents containing x_i
     - $p_i = |V_i| / |V|$
   - Assume if not retrieved then not relevant
     - $r_i = (n_i - |V_i|) / (N - |V|)$
4. Go to 2. until converges then return ranking

Advantges and Disadvantages

- Advantage
  - Documents are ranked in decreasing order of probability of being relevant
- Disadvantages
  - The need to guess the initial separation of documents into relevant and irrelevant
  - It does not take into account the frequency with which a term occurs inside a document
  - The adoption of independence of index terms

PM: Other directions

- Just one of many types of "Naïve Bayes" IR models. Important research directions are:
  - Introducing non-binary document weights
  - Introducing document length normalization
  - Relaxing the independence assumption

Bayesian Networks
Jensen and Jensen [2001]

- Probabilistic Graphical Model for information retrieval

- Use of directed graph to describe dependencies between variables (e.g. terms)

- Algorithms for propagating probabilities (infering) by using the Bayes rule