Information Retrieval Methods
- Probabilistic retrieval

Pirjo Moen
Autumn 2010, Period II
Outline

- Previous lecture revisited
- Why probabilistic information retrieval?
- Review of basic probability theory
- Classical probabilistic model in IR
  - Probability ranking principle
  - Binary independence model
  - Extensions of the basic model
- Language model approach in IR (next lecture)
Previous lecture revisited

- Improving results of retrieval
  - Local methods
    - Relevance feedback
    - Pseudo relevance feedback
  - Global methods
    - Query expansion
    - Thesauri
    - Automatic thesaurus generation
Why probabilistic information retrieval?

• Relevance feedback
  • User's feedback on relevant and irrelevant documents
  • Re-weighting in a vector space
• Possible to estimate a probability of term $t_k$ appearing in a relevant document
  • Let $D_r$ be the set of known relevant documents and $D_{rk}$ its subset that contain $t_k$; similarly, $D_{nr}$ and $D_{n rk}$ for irrelevant documents
  • $p(t_k|R) = \frac{|D_{rk}|}{|D_r|}$
  • $p(t_k|NR) = \frac{|D_{n rk}|}{|D_{nr}|}$

=> Build a probabilistic classifier for documents
Why probabilistic inform. retrieval? (2)

- Probabilities provide a principled foundation for uncertain reasoning
- Can we use probabilities to quantify our uncertainties?
Probabilistic IR models

- Probabilistic methods for information retrieval
  - One of the oldest but also one of the currently hottest topics in information retrieval
  - Traditionally neat ideas, that have never won on performance
- Different approaches
  - Classical probabilistic retrieval model
    - Probability ranking principle
  - Extensions of the classical model
    - Okabi BM25 weighting scheme
    - Bayesian networks for text retrieval
  - Language models to information retrieval
    - An important emphasis in recent work
Document ranking problem

- A collection of documents
- User issues a query
- A list of documents needs to be returned
- Binary notion of relevance
- Ranking method is the core of an IR system:
  - In what order do we present documents to the user?
- Idea:
  - Rank by (decreasing) probability of relevance of the document w.r.t. information need
  - \( p(\text{relevant} \mid \text{document}_i, \text{query}) \)
Basic probability theory

- **Joint probability** $p(A, B)$ of both events occurring
- **Conditional probability** $p(A|B)$ of event A occurring given that event B has occurred
- **Chain rule** gives fundamental relationship between joint and conditional probabilities:
  - $p(A,B) = p(A \cap B) = p(A|B)p(B) = p(B|A)p(A)$
- Similarly for the **complement** of an event $p(\overline{A})$:
  - $p(\overline{A},B) = p(B|\overline{A})p(\overline{A})$
- **Partition rule**:
  - if B can be divided into an exhaustive set of disjoint sub-cases, then $p(B)$ is the sum of the probabilities of the sub-cases
  - A special case of this rule gives: $p(B) = p(A,B) + p(\overline{A},B)$
Basic probability theory (2)

- Bayes’ rule

\[ p(a, b) = p(a \cap b) = p(a \mid b) p(b) = p(b \mid a) p(a) \]
\[ p(\overline{a} \mid b) p(b) = p(b \mid \overline{a}) p(\overline{a}) \]
\[ p(a \mid b) = \frac{p(b \mid a) p(a)}{p(b)} = \frac{p(b \mid a) p(a)}{\sum_{x=a, \overline{a}} p(b \mid x) p(x)} \]

where \( a \) and \( b \) are events

- Odds of an event

\[ O(a) = \frac{p(a)}{p(\overline{a})} = \frac{p(a)}{1 - p(a)} \]
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.”

S. Robertson, W.S. Cooper, M.E. Maron (1960s/1970s); van Rijsbergen (1979:113); Manning & Schütze (1999:538)
Probability ranking principle (2)

• Assumptions
  • x - a document in the collection
  • R - relevance of a document w.r.t. given (fixed) query
  • NR - non-relevance of a document w.r.t. given (fixed) query
  • $p(R)$, $p(NR)$ - a prior probability of retrieving a relevant or a non-relevant document, respectively
  • $p(x|R)$, $p(x|NR)$ - a probability that if a relevant or non-relevant document is retrieved, respectively, it is x

• Need to find $p(R|x)$ - a probability that a document x is relevant

\[
p(R|x) = \frac{p(x|R)p(R)}{p(x)} \quad \quad \quad p(NR|x) = \frac{p(x|NR)p(NR)}{p(x)}
\]

\[
p(R|x) + p(NR|x) = 1
\]
Probability ranking principle (3)

- Simple case:
  - No selection costs or other utility concerns that would differentially weight errors or actions

- Bayes’ optimal decision rule
  - $x$ is relevant iff $p(R|x) > p(NR|x)$
  - Minimize risk of loss

- Probability ranking principle (PRP) in action:
  - Rank all documents by decreasing $p(R|x)$

- Theorem:
  - Using the PRP is optimal, in that it minimizes the loss (Bayes risk) under 1/0 loss
  - Provable if all probabilities correct
Probability ranking principle (4)

- More complex case: retrieval costs
  - Let $d$ be a document
  - $C$ – a cost of retrieval of relevant document
  - $C'$ - a cost of retrieval of non-relevant document
- Probability ranking principle now
  \[
  C \cdot p(R|d) + C' \cdot (1 - p(R|d)) \leq C \cdot p(R|d') + C' \cdot (1 - p(R|d'))
  \]
  for all $d'$ not yet retrieved, then $d$ is the next document to be retrieved
- No further consideration of loss/utility from now on
Probability retrieval strategy

- How do we compute all those probabilities?
  - Do not know exact probabilities => have to use estimates
  - Binary Independence Model (BIM) 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 or near duplicates
  - Boolean model of relevance
    - Document either relevant or non-relevant
  - One has a single step information need
    - Seeing a range of results might let user refine query
Probabilistic retrieval strategy (2)

- Estimate how terms contribute to relevance
  - Occurring vs. not occurring in documents
  - How do things like tf, df, and document length influence your judgments about document relevance?
    - One answer is the Okapi formulae

- Combine to find a probability of document relevance

- Order documents by decreasing probability of relevance
Probabilistic retrieval strategy (3)

**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*
Binary independence model

- Traditionally used in conjunction with PRP
- “Binary” = Boolean
  - Documents are represented as binary incidence vectors of terms
    \[ \mathbf{x} = (x_1, \ldots, x_n) \]
    where \( x_i = 1 \) iff term \( i \) is present in document \( x \)
  - Different documents can be modelled as same vector
- “Independence”
  - Terms occur in documents independently
  - No associations between terms are recognized
Binary independence model (2)

- 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 a binary term incidence vector representing \( d \)
- Interested only in ranking => use odds and Bayes’ rule:

\[
O(R|q,\tilde{x}) = \frac{p(R|q,\tilde{x})}{p(NR|q,\tilde{x})} = \frac{\frac{p(R|q)p(\tilde{x}|R,q)}{p(\tilde{x}|q)}}{\frac{p(NR|q)p(\tilde{x}|NR,q)}{p(\tilde{x}|q)}}
\]
Binary independence model (3)

- Rule re-organized:

\[
O(R|q, \hat{x}) = \frac{p(R|q, \hat{x})}{p(NR|q, \hat{x})} = \frac{p(R|q)}{p(NR|q)} \cdot \frac{p(\hat{x}|R, q)}{p(\hat{x}|NR, q)}
\]

- Using naïve Bayesian independence assumption:

\[
\frac{p(\hat{x}|R, q)}{p(\hat{x}|NR, q)} = \prod_{i=1}^{n} \frac{p(x_i|R, q)}{p(x_i|NR, q)}
\]

Constant for a given query

Needs estimation
Binary independence model (4)

- So

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

- Since \( x_i \) is either 0 or 1, can be re-organized

\[ O(R|q,d) = O(R|q) \cdot \prod_{x_i=1} \frac{p(x_i=1|R,q)}{p(x_i=1|NR,q)} \cdot \prod_{x_i=0} \frac{p(x_i=0|R,q)}{p(x_i=0|NR,q)} \]

- Let: \( p_i = p(x_i=1|R,q) \); \( r_i = p(x_i=1|NR,q) \);

- Assume, for all terms not occurring in the query (i.e., \( q_i=0 \)), \( p_i = r_i \)

  - Can be changed (e.g. in relevance feedback)
Binary independence model (5)

\[ O(R \mid q, x) = O(R \mid q) \cdot \prod_{x_i=q_i=1} p_i \cdot \prod_{x_i=0\atop q_i=1} \frac{1-p_i}{1-r_i} \]

All matching terms

\[ = O(R \mid q) \cdot \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \cdot \prod_{q_i=1} \frac{1-p_i}{1-r_i} \]

All matching terms

Non-matching query terms

All query terms
Binary independence model (6)

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

- Constant for each query
- Only quantity to be estimated for rankings

· Retrieval Status Value:

\[ RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \]
Retrieval status value

- Everything comes down to computing RSV

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

- Let's reformulate RSV

\[ RSV = \sum_{x_i=q_i=1} c_i ; \quad c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \]

- So, how to compute \( c_i \)'s from the data?
Estimating RSV coefficients

- For each term i look at this table of document counts:

<table>
<thead>
<tr>
<th>Documents</th>
<th>Relevant</th>
<th>Non- Relevant</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_i = 1$</td>
<td>$s$</td>
<td>$n - s$</td>
<td>$n$</td>
</tr>
<tr>
<td>$X_i = 0$</td>
<td>$S - s$</td>
<td>$N - n - S + s$</td>
<td>$N - n$</td>
</tr>
<tr>
<td>Total</td>
<td>$S$</td>
<td>$N - S$</td>
<td>$N$</td>
</tr>
</tbody>
</table>

- Estimates:

$$p_i \approx \frac{s}{S} \quad r_i \approx \frac{(n - s)}{(N - S)}$$

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

For now, assume no zero terms.
Estimating RSV coefficients (2)

• If the proportion of relevant documents is low
  • Statistics of non-relevant documents approximated by the whole collection
  • \( r_i \) is \( n/N \)
  • \( \log \left( \frac{1 - r_i}{r_i} \right) = \log \left( \frac{N - n}{n} \right) \approx \log \frac{N}{n} = \text{IDF!} \)

• Probability \( p_i \) can be estimated in various ways:
  • Constant: just get idf weighting of terms
  • Proportional to probability of occurrence in collection
    • more accurately, to log of this
  • From relevant documents, if know some
    • Relevance weighting used in feedback loop
Estimating RSV coefficients (3)

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

• Using probabilistic relevance feedback iteratively

1. Guess a preliminary probabilistic description of R and use it to retrieve a first set of documents V
2. Interact with the user to refine the description:
   • learn some definite members of R and NR
3. Re-estimate \( p_i \) and \( r_i \) on the basis of these

OR

4. Combine new information with original guess (use Bayesian prior):

\[
p_i^{(2)} = \frac{|V_i| + \kappa p_i^{(1)}}{|V| + \kappa}
\]

\( \kappa \) is prior weight

• Repeat and generate a succession of approximations to R until satisfied
PRP and BIM

- Getting reasonable approximations of probabilities is possible
- Requires restrictive assumptions:
  - Term independence
  - Terms not in query don’t affect the outcome
  - Boolean representation of documents/queries/relevance
  - Document relevance values are independent
- Some of these assumptions can be removed
- Problem:
  - Require partial relevance information
    OR
  - Can only derive somewhat inferior term weights
Good and Bad News

- Standard Vector Space Model
  - Empirical for the most part; success measured by results
  - Few properties provable

- Probabilistic Model Advantages
  - Based on a firm theoretical foundation
  - Theoretically justified optimal ranking scheme

- Probabilistic Model Disadvantages
  - Making the initial guess to get the (first) set of results
  - 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
Extensions of the basic prob. model

- **BIM**
  - Originally designed for short catalog records of fairly consistent length
  - For modern full-text search collections require sensitivity to term frequency and document length

- **Okabi BM25**
  - Non-binary model
  - Uses BM25 weighting scheme (also called Okabi weighting after the system in which it was implemented)
  - Sensitive for term frequencies and document length (as required)
  - Different variants
  - From 1994 until today, one of the most widely used and robust retrieval models
Extensions of the basic prob. model (2)

- Bayesian network models for information retrieval
  - Bayesian networks
    • A directed acyclic graph of nodes (events or variables; in this case Boolean values) and links (dependencies between nodes)
    • Model causal relations between events
  - Model documents in a document network (large)
    • Compute once for each document collection
  - Model information need in a query network (small)
    • Compute one for every query
  - Flexible ways of combining term weights
  - Not much follow-on work, but new Bayesian network technology yet to be applied
Idea on Bayesian Networks for IR

Document Network:
- $d_i$ - documents
- $t_i$ - document representations
- $r_i$ - "concepts"

Query Network:
- $c_i$ - query concepts
- $q_i$ - high-level concepts
- $l$ - goal node
In the following lectures

- Next lecture
  - Language models for information retrieval
- After that
  - Web search
  - XML retrieval
Resources for this lecture

- Introduction to Information Retrieval, chapter 11
- Lecture slides of Christopher Manning, Prabhakar Raghavan and Hinrich Schütze – Thank you!