Probabilistic IR topics

- Classical probabilistic retrieval model
  - Probability ranking principle, etc.
  - Binary independence model (≈ Naïve Bayes)
  - (Okapi) BM25

- Bayesian networks for text retrieval

- Language model approach to IR
  - An important emphasis in recent work

- *Probabilistic methods have been a recurring theme in Information Retrieval*
  - Traditionally: neat ideas, but didn’t win on performance
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 the 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**
  - \( P(R=1|\text{document}_i, \text{query}) \)

Recall a few probability basics

- For events A and B:
- Bayes’ Rule
  \[
  p(A, B) = p(A \mid B)p(B) = p(B \mid A)p(A)
  \]
  \[
  p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)} = \frac{p(B \mid A)p(A)}{\sum_{X=A,\overline{A}} p(B \mid X)p(X)}
  \]
  **Posterior**
  **Prior**
- Odds:
  \[
  O(A) = \frac{p(A)}{p(A)} = \frac{p(A)}{1 \cdot p(A)}
  \]
The Probability Ranking Principle (PRP)

“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.”


Probability Ranking Principle (PRP)

Let $x$ represent a document in the collection. Let $R$ represent relevance of a document w.r.t. given (fixed) query and let $R=1$ represent relevant and $R=0$ not relevant.

Need to find $p(R=1| x)$ - probability that a document $x$ is relevant.

$$p(R=1| x) = \frac{p(x | R=1)p(R=1)}{p(x)}$$

$p(R=1), p(R=0)$ - prior probability of retrieving a relevant or non-relevant document $p(x| R=1)$, $p(x| R=0)$ - probability that if a relevant (not 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
- PRP in action: Rank all documents by \( p(R=1|x) \)
- 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]
- How do we compute all those probabilities?
  - Do not know exact probabilities, have to use estimates

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**
Binary Independence Model

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

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
- Use odds and Bayes’ Rule:
  $$O(R|q,\bar{x}) = \frac{p(R=1|q)p(\bar{x}|R=1,q)}{p(R=0|q)p(\bar{x}|R=0,q)} = \frac{p(R=1|q)}{p(R=0|q)} \frac{p(\bar{x}|R=1,q)}{p(\bar{x}|R=0,q)}$$
Binary Independence Model

\[
O(R | q, \bar{x}) = \frac{p(R=1 | q, \bar{x})}{p(R=0 | q, \bar{x})} = \frac{p(R=1 | q)}{p(R=0 | q)} \frac{p(\bar{x} | R=1, q)}{p(\bar{x} | R=0, q)}
\]

* Using Independence Assumption:

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

\[
op(R | q, \bar{x}) = op(R | q) \times \prod_{i=1}^{n} p(x_i | R=1, q)
\]

\[
op(R | q, \bar{x}) = op(R | q) \times \prod_{i=1}^{n} p(x_i | R=1, q)
\]

* Since \( x_i \) is either 0 or 1:

\[
op(R | q, \bar{x}) = op(R | q) \times \prod_{x_i=1} p(x_i =1 | R=1, q) \times \prod_{x_i=0} p(x_i =0 | R=1, q) \times \prod_{x_i=1} p(x_i =1 | R=0, q) \times \prod_{x_i=0} p(x_i =0 | R=0, q)
\]

* Let \( p_i = p(x_i =1 | R=1, q) \); \( r_i = p(x_i =1 | R=0, q) \);

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

\[
op(R | q, \bar{x}) = op(R | q) \times \prod_{x_i=1} \frac{p_i}{r_i} \times \prod_{x_i=0} \frac{1}{r_i}
\]
**Model Parameters**

<table>
<thead>
<tr>
<th></th>
<th>document</th>
<th>relevant (R=1)</th>
<th>not relevant (R=0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>term present</td>
<td>(x_i = 1)</td>
<td>(p_i)</td>
<td>(r_i)</td>
</tr>
<tr>
<td>term absent</td>
<td>(x_i = 0)</td>
<td>((1 - p_i))</td>
<td>((1 - r_i))</td>
</tr>
</tbody>
</table>

**Binary Independence Model**

\[
O(R \mid q, \bar{x}) = O(R \mid q) \times \frac{p_i}{r_i} \times \frac{1}{1 - p_i} \times \frac{1}{1 - r_i}
\]

- **All matching terms**
- **Non-matching query terms**

\[
O(R \mid q, \bar{x}) = O(R \mid q) \times \frac{p_i}{r_i} \times \frac{1}{1 - p_i} \times \frac{1}{1 - r_i}
\]

- **All matching terms**
- **All query terms**
**Binary Independence Model**

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

**Retrieval Status Value:**

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

All boils down to computing RSV.

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

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

The \(c_i\) are log odds ratios
They function as the term weights in this model

**So, how do we compute \(c_i\)'s from our data?**
Binary Independence Model

- Estimating RSV coefficients in theory
- 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.

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

$$\log \frac{1}{r_i} = \log \frac{N}{n} \frac{n}{S+s} \quad \log \frac{N}{n} \frac{n}{s} \quad \log \frac{N}{n} = IDF!$$
Estimation – key challenge

- $p_i$ (probability of occurrence in relevant documents) cannot be approximated as easily
- $p_i$ can be estimated in various ways:
  - from relevant documents if known some
    - Relevance weighting can be used in a feedback loop
  - constant (Croft and Harper combination match) – then just get idf weighting of terms (with $p_i=0.5$)
    \[
    RSV = \log \frac{N}{\sum_{x_i=q_i=1} n_i}
    \]
    - proportional to prob. of occurrence in collection
      - Greiff (SIGIR 1998) argues for $1/3 + 2/3 \frac{df}{N}$