Recuperação de Informação B

Cap. 02: Modeling (Probabilistic Model)
2.5.4, 2.5.5

September 06, 1999
Probabilistic Model

- Objective: to capture the IR problem using a probabilistic framework
- Given a user query, there is an *ideal* answer set
- Querying as specification of the properties of this ideal answer set (clustering)
- But, what are these properties?
- Guess at the beginning what they could be (i.e., guess initial description of ideal answer set)
- Improve by iteration
Probabilistic Model

- An initial set of documents is retrieved somehow.
- User inspects these docs looking for the relevant ones (in truth, only top 10-20 need to be inspected).
- IR system uses this information to refine description of ideal answer set.
- By repeating this process, it is expected that the description of the ideal answer set will improve.
- Have always in mind the need to guess at the very beginning the description of the ideal answer set.
- Description of ideal answer set is modeled in probabilistic terms.
Probabilistic Ranking Principle

- Given a user query $q$ and a document $d_j$, the probabilistic model tries to estimate the probability that the user will find the document $d_j$ interesting (i.e., relevant). The model assumes that this probability of relevance depends on the query and the document representations only. Ideal answer set is referred to as $R$ and should maximize the probability of relevance. Documents in the set $R$ are predicted to be relevant.

- But,
  - how to compute probabilities?
  - what is the sample space?
The Ranking

- Probabilistic ranking computed as:
  - \( \text{sim}(q, dj) = \frac{P(dj \text{ relevant-to } q)}{P(dj \text{ non-relevant-to } q)} \)
  - This is the odds of the document \( dj \) being relevant
  - Taking the odds minimize the probability of an erroneous judgement

- Definition:
  - \( wij \in \{0, 1\} \)
  - \( P(R \mid \text{vec}(dj)) \) : probability that given doc is relevant
  - \( P(\neg R \mid \text{vec}(dj)) \) : probability doc is not relevant
The Ranking

- \( \text{sim}(d_j, q) = \frac{P(R \mid \text{vec}(d_j))}{P(\neg R \mid \text{vec}(d_j))} \)

\[
= \frac{[P(\text{vec}(d_j) \mid R) \cdot P(R)]}{[P(\text{vec}(d_j) \mid \neg R) \cdot P(\neg R)]} \\
\sim \frac{P(\text{vec}(d_j) \mid R)}{P(\text{vec}(d_j) \mid \neg R)}
\]

- \( P(\text{vec}(d_j) \mid R) \) : probability of randomly selecting the document \( d_j \) from the set \( R \) of relevant documents
The Ranking

\[ \text{sim}(d_j, q) \sim \frac{P(\text{vec}(d_j) \mid R)}{P(\text{vec}(d_j) \mid \neg R)} \]

~

\[ \frac{\prod P(k_i \mid R)}{\prod P(k_i \mid \neg R)} \times \frac{\prod P(\neg k_i \mid R)}{\prod P(\neg k_i \mid \neg R)} \]

\[ P(k_i \mid R) : \text{probability that the index term } k_i \text{ is present in a document randomly selected from the set } R \text{ of relevant documents} \]
The Ranking

\[ \text{sim}(d_j, q) \sim \log \left( \frac{\prod P(k_i | R) * \prod P(\neg k_j | R)}{\prod P(k_i | \neg R) * \prod P(\neg k_j | \neg R)} \right) \]

\[ \sim K \cdot \left( \log \prod \frac{P(k_i | R)}{P(\neg k_i | R)} + \log \prod \frac{P(k_i | \neg R)}{P(\neg k_i | \neg R)} \right) \]

\[ \sim \sum w_i q * w_j * \left( \log \frac{P(k_i | R)}{P(\neg k_i | R)} + \log \frac{P(k_i | \neg R)}{P(\neg k_i | \neg R)} \right) \]

where

\[ P(\neg k_i | R) = 1 - P(k_i | R) \]

\[ P(\neg k_i | \neg R) = 1 - P(k_i | \neg R) \]
The Initial Ranking

- \[ \text{sim}(d_j, q) \sim \sum w_i q \ast w_j \ast (\log \frac{P(k_i | R)}{P(\neg k_i | R)}) + \log \frac{P(k_i | \neg R)}{P(\neg k_i | \neg R)} \]

- Probabilities \( P(k_i | R) \) and \( P(k_i | \neg R) \)?

- Estimates based on assumptions:
  - \( P(k_i | R) = 0.5 \)
  - \( P(k_i | \neg R) = \frac{n_i}{N} \)
    
    where \( n_i \) is the number of docs that contain \( k_i \)
  - Use this initial guess to retrieve an initial ranking
  - Improve upon this initial ranking
Improving the Initial Ranking

\[ \text{sim}(d_j, q) \sim \sum wi_q \times wij \times (\log \frac{P(k_i | R)}{P(\neg k_i | R)} + \log \frac{P(k_i | \neg R)}{P(\neg k_i | \neg R)}) \]

- Let
  - \( V \) : set of docs initially retrieved
  - \( V_i \) : subset of docs retrieved that contain \( k_i \)

- Reevaluate estimates:
  - \( P(k_i | R) = \frac{V_i}{V} \)
  - \( P(k_i | \neg R) = \frac{n_i - V_i}{N - V} \)

- Repeat recursively
Improving the Initial Ranking

\[ \text{sim}(d_j, q) \sim \sum w_i q \ast w_j \ast (\log \frac{P(k_i | R)}{P(\neg k_i | R)} + \log \frac{P(k_i | \neg R)}{P(\neg k_i | \neg R)}) \]

To avoid problems with \( V=1 \) and \( V_i=0 \):

\[ \begin{align*}
P(k_i | R) &= \frac{V_i + 0.5}{V + 1} \\
P(k_i | \neg R) &= \frac{N - V + 0.5}{N - V + 1}
\end{align*} \]

Also,

\[ \begin{align*}
P(k_i | R) &= \frac{V_i + n_i/N}{V + 1} \\
P(k_i | \neg R) &= \frac{n_i - V_i + n_i/N}{N - V + 1}
\end{align*} \]
Pluses and Minuses

- Advantages:
  - Docs ranked in decreasing order of probability of relevance

- Disadvantages:
  - need to guess initial estimates for $P(ki \mid R)$
  - method does not take into account tf and idf factors
Brief Comparison of Classic Models

- Boolean model does not provide for partial matches and is considered to be the weakest classic model.
- Salton and Buckley did a series of experiments that indicate that, in general, the vector model outperforms the probabilistic model with general collections.
- This seems also to be the view of the research community.