Probabilistic IR
Information Needs and Queries

• “What are the courses at SFU talking about document indexes?”
  – Issue a query “course, SFU, document indexes” to a search engine

• Information need: the topic about which the user desires to know more
  – Unfortunately, often cannot be fed into a search engine

• Query: what the user conveys to the computer in an attempt to communicate the information need
  – Multiple queries may be formed to capture the same information need
  – A query may not capture the information need sufficiently
Why Retrieval Models?

• Understand and model how a person determines whether a document is relevant to her/his information need
  – An extremely complex mechanism – how language is represented and processed in human brains

• Evaluating retrieval models – comparing retrieval results to human actions
  – Ranking algorithms based on good models should retrieve relevant documents near the top of the ranking

• Ranking algorithms for general search improved in effectiveness by over 100% in 1990s, so did web search effectiveness in the last 10 years
Retrieval Models

• A retrieval model is a formal representation of the process of matching a query and a document
• Typically, retrieval models capture the statistical properties of text rather than the linguistic structure
  – More sophisticated models incorporate linguistic features, but tend to be of secondary importance
  – An important difference from NLP
• Commercial web search engines do not disclose their retrieval models
  – No doubt that the ranking algorithms there rely on solid mathematical foundations
Relevance

• A complex concept
  – Even a person may not explain why one document is more relevant than another
  – Different people may have different judgments

• Topical relevance versus user relevance
  – Topical relevance: whether the query and the document are about the same topic – a list of the U.S. presidents is topical relevant to query “Abraham Lincoln”
  – User relevance: many other factors such as age of the document, language, target audience, novelty, etc. – a list of the U.S. presidents may not be considered user relevant to query “Abraham Lincoln”
Binary vs Multi-valued Relevance

- Binary relevance: either a document is relevant or irrelevant
  - An over-simplified assumption – obviously some documents are less relevant than others, but still more relevant than documents that are completely off-topic
  - For query “Abraham Lincoln”, a list of the U.S. presidents is more relevant than an ad for a Lincoln car

- Multi-valued relevance
  - Just three levels (relevant, unsure, irrelevant) make judges’ task much easier
  - However, multi-valued relevance is hard to be used in ranking algorithms
The Boolean Retrieval Model

• We can pose any query which is in the form of a Boolean expression of terms, i.e., in which terms are combined with the operators AND, OR, and NOT
  – Each document is modeled as a set of words
• Ad hoc retrieval: retrieve documents that are relevant to an arbitrary user information need, communicated to the system by means of a one-off, user-initiated query
A Probabilistic View

• Is a document $d$ related to a query $q$?
  – Probability $P(R=1 \mid d, q)$

• Using probability in retrieval
  – Relevant documents $\{d \mid P(R=1 \mid d, q) > 1 - P(R=1 \mid d, q)\}$
  – Ranking in probability descending order

• Central issues
  – How to compute $P(R=1 \mid d, q)$?
  – How to rank documents according to probabilities?
Probability Theory Review

• Chain rule
  – \( P(A, B) = P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) \)
  – \( P(\neg A, B) = P(B|\neg A)P(\neg A) \)

• Partition rule
  – If event \( B \) can be divided into an exhaustive set of disjoint subcases \( B_1, \ldots, B_n \), then
    \[ P(B) = P(B_1) + P(B_2) + \ldots + P(B_n) \]
  – A special case: \( P(B) = P(A, B) + P(\neg A, B) \)
Bayes’ Rule

\[ P(A | B) = \frac{P(B | A)P(A)}{P(B)} = \frac{P(B | A)P(A)}{P(B | A)P(A) + P(B | \overline{A})P(\overline{A})} \]

- P(A) – the prior probability
- P(A|B) – the posterior probability
- Computing the posterior probability based on the likelihood of B occurring in the two cases where A does or does not hold
Probability Ranking Principle

- A binary notion of relevance: a document is either relevant or irrelevant to a query
  - Probability ranking principle (PRP): Rank documents by their estimated probabilities of relevance with respect to the information need $P(R=1 \mid d, q)$
  - But how should we calculate the probability?
- The 0/1 Loss Case: losing a point for either returning an irrelevant document or failing to return a relevant document
  - Task: return the best possible results as the top $k$ documents
  - According to PRP, return the top $k$ documents in $P(R=1 \mid d, q)$
Bayes Optimal Decision Rule

• The decision that minimizes the risk of loss is to simply return the documents that are more likely relevant than irrelevant
  – Document d is relevant if and only if \( P(R=1 \mid d, q) > P(R=0 \mid d, q) = 1 - P(R=1 \mid d, q) \)
  – Since \( P(R=0 \mid d, q) + P(R=1 \mid d, q) = 1 \), d is relevant if and only if \( P(R=1 \mid d, q) > 0.5 \)

• The probability ranking principle is optimal in the sense that it minimizes the expected loss (also known as the Bayesian risk) under 0/1 loss
PRP with Retrieval Costs

• Modeling retrieval costs
  – $C_1$: the cost of retrieval of a relevant document
  – $C_0$: the cost of retrieval of an irrelevant document

• A document is the next to be retrieved if for any other document $d'$, $C_1 \times P(R=1 \mid d) + C_0 \times P(R=0 \mid d) \leq C_1 \times P(R=1 \mid d') + C_0 \times P(R=0 \mid d')$
Binary Independence Model (BIM)

- Documents and queries are represented as binary term incidence vectors
- Terms are modeled as occurring in documents independently
  - No association between terms – far from true in practice
- The relevance of each document is independent of the relevance of other documents
  - Far from true in practice
Probability $P(R \mid \bar{x}, \bar{q})$

- Using Bayes rule, we have

$$P(R = 1 \mid \bar{x}, \bar{q}) = \frac{P(\bar{x} \mid R = 1, \bar{q})P(R = 1 \mid \bar{q})}{P(\bar{x} \mid \bar{q})}$$

$$P(R = 0 \mid \bar{x}, \bar{q}) = \frac{P(\bar{x} \mid R = 0, \bar{q})P(R = 0 \mid \bar{q})}{P(\bar{x} \mid \bar{q})}$$

- $P(\bar{x} \mid R = 1, \bar{q})$ is the probability that if a relevant document is retrieved, the document’s representation is $\bar{x}$
  - Have to be estimated

- $P(R = 1 \mid \bar{q})$ and $P(R = 0 \mid \bar{q})$ are prior probabilities of retrieving a relevant or irrelevant document
  - $P(R = 1 \mid \bar{x}, \bar{q}) + P(R = 0 \mid \bar{x}, \bar{q}) = 1$
Ranking by Odds

• Computing the posterior probabilities is often costly on large data collections
• If only ranking is concerned, we can use odds
  – \( O(A) = \frac{P(A)}{P(\neg A)} \)

\[
O(R \mid \tilde{x}, \tilde{q}) = \frac{P(R = 1 \mid \tilde{x}, \tilde{q})}{P(R = 0 \mid \tilde{x}, \tilde{q})} = \frac{\frac{P(R = 1 \mid \tilde{q})P(\tilde{x} \mid R = 1, \tilde{q})}{P(\tilde{x} \mid \tilde{q})}}{\frac{P(R = 0 \mid \tilde{q})P(\tilde{x} \mid R = 0, \tilde{q})}{P(\tilde{x} \mid \tilde{q})}} = \frac{P(R = 1 \mid \tilde{q})}{P(R = 0 \mid \tilde{q})} \frac{P(\tilde{x} \mid R = 1, \tilde{q})}{P(\tilde{x} \mid R = 0, \tilde{q})}
\]

• \( \frac{P(R = 1 \mid \tilde{q})}{P(R = 0 \mid \tilde{q})} \) is a constant for a given query \( q \)
Estimating Prior Probability Ratio

- How to estimate \( \frac{P(\bar{x} \mid R = 1, \bar{q})}{P(\bar{x} \mid R = 0, \bar{q})} \)?

- Naïve Bayes conditional independence assumption: the presence or absence of a word in a document is independent of any other words
  
  – An over-simplified assumption, but often works well in practice

\[
\frac{P(\bar{x} \mid R = 1, \bar{q})}{P(\bar{x} \mid R = 0, \bar{q})} = \prod_{i=1}^{M} \frac{P(x_i \mid R = 1, \bar{q})}{P(x_i \mid R = 0, \bar{q})} \quad \text{and} \quad O(R \mid \bar{x}, \bar{q}) = O(R \mid \bar{q}) \prod_{i=1}^{M} \frac{P(x_i \mid R = 1, \bar{q})}{P(x_i \mid R = 0, \bar{q})}
\]
Estimating Prior Probability Ratio

- Each $x_t$ is either 0 or 1
  \[
  O(R \mid \bar{x}, \bar{q}) = O(R \mid \bar{q}) \prod_{t: x_t = 1} \frac{P(x_t = 1 \mid R = 1, \bar{q})}{P(x_t = 1 \mid R = 0, \bar{q})} \prod_{t: x_t = 0} \frac{P(x_t = 0 \mid R = 1, \bar{q})}{P(x_t = 0 \mid R = 0, \bar{q})}
  \]
- $p_t = P(x_t = 1 \mid R = 1, \bar{q})$: the probability of a term appearing in a document relevant to the query
- $u_t = P(x_t = 1 \mid R = 0, \bar{q})$: the probability of a term appearing in a document irrelevant to the query
- Contingency

<table>
<thead>
<tr>
<th>Document</th>
<th>Relevant (R = 1)</th>
<th>Irrelevant (R = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term present $x_t = 1$</td>
<td>$p_t$</td>
<td>$u_t$</td>
</tr>
<tr>
<td>Term absent $x_t = 0$</td>
<td>$1 - p_t$</td>
<td>$1 - u_t$</td>
</tr>
</tbody>
</table>
Retrieval Status Value (RSV)

- Assumption: terms not in the query are equally likely to occur in relevant and irrelevant documents – \( p_t = u_t \) if \( q_t = 0 \)

- We only need to consider terms in the query

\[
O(R \mid \tilde{x}, \tilde{q}) = O(R \mid \tilde{q}) \prod_{t : x_t = q_t = 1} \frac{p_t}{u_t} \prod_{t : x_t = 0, q_t = 1} \frac{1 - p_t}{1 - u_t} = O(R \mid \tilde{q}) \prod_{t : x_t = q_t = 1} \frac{p_t(1 - u_t)}{u_t(1 - p_t)} \prod_{t : q_t = 1} \frac{1 - p_t}{1 - u_t}
\]

- \( \prod_{t : q_t = 1} \frac{1 - p_t}{1 - u_t} \) is a constant for a particular query

- Retrieval status value (RSV)

\[
RSV_d = \log \prod_{t : x_t = q_t = 1} \frac{p_t(1 - u_t)}{u_t(1 - p_t)} = \sum_{t : x_t = q_t = 1} \log \frac{p_t(1 - u_t)}{u_t(1 - p_t)}
\]
Computing RSV

- \( c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} + \log \frac{1-u_t}{u_t} \)
- \( RSV_d = \sum_{x_t, q_t = 1} c_t \)
- We use relevant documents to compute \( p_t / (1 - p_t) \)
- We use irrelevant documents to compute \( (1 - u_t) / u_t \)

<table>
<thead>
<tr>
<th>Document</th>
<th>Relevant</th>
<th>Irrelevant</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term present ( x_t = 1 )</td>
<td>( s )</td>
<td>( df_t - s )</td>
<td>( df_t )</td>
</tr>
<tr>
<td>Term absent ( x_t = 0 )</td>
<td>( S - s )</td>
<td>( N - df_t - (S - s) )</td>
<td>( N - df_t )</td>
</tr>
<tr>
<td>Total</td>
<td>( S )</td>
<td>( N - S )</td>
<td>( N )</td>
</tr>
</tbody>
</table>
Computing $c_t$

- $p_t = s / S$, $u_t = (df_t - s) / (N - S)$
- $c_t = K(N, df_t, S, s) = \log \frac{s / (S - s)}{(df_t - s) / ((N - df_t) - (S - s))}$
- Smoothing to avoid zeros
  $$\hat{c}_t = K(N, df_t, S, s) = \log \frac{(s + \frac{1}{2}) / (S - s + \frac{1}{2})}{(df_t - s + \frac{1}{2}) / ((N - df_t) - (S - s) + \frac{1}{2})}$$
- In practice, only a small portion of documents are relevant
  - We approximate $u_t = df_t / N$
  - $\log \frac{1 - u_t}{u_t} = \log \frac{N - df_t}{df_t} \approx \log \frac{N}{df_t}$ — aha, it is IDF!
  - $c_t = \log \left( \frac{p_t}{1 - p_t} \frac{1 - u_t}{u_t} \right) \approx \log \left( \frac{|V_t| + \frac{1}{2}}{|V| - |V_t| + 1 \cdot df_t} \right) = \log \frac{|V_t| + \frac{1}{2}}{|V| - |V_t| + 1} + \log \frac{N}{df_t}$
Estimating $p_t$

- Use the frequency of term occurrence in known relevant documents
  - A (pseudo) relevance feedback approach
- Use a constant, say 0.5, for all terms in the query
  - The document ranking is determined simply by which query terms occur in documents scaled by their IDF weighting
  - This simple technique works very well on short documents
- Estimate $p_t$ from the collection level: $p_t = \frac{df_t}{N}$
An Iterative Approach

• Guess initial estimates of $p_t$ and $u_t$, for example, set $p_t$ to 0.5
• Use the current estimates of $p_t$ and $u_t$ to determine a best guess at the set of relevant documents $R = \{d: R_{d,q} = 1\}$
  – Retrieve a set of candidate documents and present to the user
• Learn from the user relevance judgments on a subset of documents $V \subseteq R$
  – Divide $R$ into two exclusive subsets: $VR = \{d \in R: R_{d,q} = 1\}$ and $VIR = VR = \{d \in R: R_{d,q} = 0\}$
• Re-estimate $p_t$ and $u_t$ on the basis of known relevant and irrelevant documents
  – $p_t = \frac{|VR_t|}{|VR|}$
  – Smoothing: $p_t = \frac{|VR_t| + \frac{1}{2}}{|VR| + 1}$
• Repeat the above process from step 2
Bayesian Updating Process

- $|V|$ in relevance feedback process is often very small
  - Using $|V|$ to estimate directly is quite unreliable
- $p_{t}^{k+1} = \frac{|VR_t| + \kappa p_t^k}{|VR| + \kappa}$
  - $p_t^k$ is the $k$-th estimate for $p_t$ in an iterative updating process and is used as a Bayesian prior in the next iteration with a weighting of $\kappa$
  - $\kappa$ can be set to 5
Using Pseudo Relevance Feedback

- Idea: assuming VR = V
- Implementation: use a fixed size set V of highest ranked documents
  - Update $p_t$ and $u_t$ accordingly
  - $p_t = (|VR_t| + \frac{1}{2}) / (|VR| + 1)$
  - $u_t = \frac{df_t - |V_t| + \frac{1}{2}}{N - |V| + 1}$
Assumptions Revisited

- In the binary independence model (BIM), we assume
  - A Boolean representation of documents/queries/relevance
  - Term independence
  - Terms not in the query do not affect the outcome
  - Document relevance values are independent

- More assumptions, more difficult to achieve good performance
  - BIM also requires partial relevance information or otherwise derives apparently inferior term weighting models

- A probabilistic IR system scores queries not by cosine similarity and TF/IDF in a vector space, but by a slightly different formula motivated by probability theory
Tree-Structured Dependencies

- Removing the assumption that terms are independent
  - In practice, the assumption does not hold
- A weaker assumption: each term can be directly dependent on only one other term
  - Probability estimation using the tree-augmented Naïve Bayes model
Okapi BM25: A Nonbinary Model

• For long documents, we should consider term frequencies and document length
• When there is no relevance feedback, $S = s = 0$, then
  $$RSV_d = \sum_{t \in q} \log \frac{N - df_t + \frac{1}{2}}{df_t + \frac{1}{2}}$$
  – Factoring in the frequency of each term and document length
  $$RSV_d = \sum_{t \in q} \log \frac{N}{df_t} \cdot \frac{(k_1 + 1)tf_{td}}{k_1((1 - b) + b(L_d / L_{ave}))) + tf_{td}}$$
  – $k_1 = 0$ corresponds to no consideration of term frequency, a large $k_1$ value corresponds to using raw term frequency – often set to 1.2
  – $b = 0$ means no length normalization, and $b = 1$ means fully scaling the term weight by the document length – often set to 0.75
Summary

• Probabilistic ranking principle
• Binary independence model (BIM)
• Ranking in BIM
  – Using (pseudo) relevance feedback iteratively
• Extensions
  – Tree-structured dependencies
  – BM25: using term frequency information
To-Do List

• Read Chapter 7.2
• Exercises in Chapter 7