Boolean and Vector Space Retrieval Models

Many slides in this section are adapted from Prof. Joydeep Ghosh (UT ECE) who in turn adapted them from Prof. Dik Lee (Univ. of Science and Tech, Hong Kong)

Retrieval Models

• A retrieval model specifies the details of:
  – Document representation
  – Query representation
  – Retrieval function
• Determines a notion of relevance.
• Notion of relevance can be binary or continuous (i.e. ranked retrieval).

Classes of Retrieval Models

• Boolean models (set theoretic)
  – Extended Boolean
• Vector space models (statistical/algebraic)
  – Generalized VS
  – Latent Semantic Indexing
• Probabilistic models

Other Model Dimensions

• Logical View of Documents
  – Index terms
  – Full text
  – Full text + Structure (e.g. hypertext)
• User Task
  – Retrieval
  – Browsing

Retrieval Tasks

• Ad hoc retrieval: Fixed document corpus, varied queries.
• Filtering: Fixed query, continuous document stream.
  – User Profile: A model of relative static preferences.
  – Binary decision of relevant/not-relevant.
• Routing: Same as filtering but continuously supply ranked lists rather than binary filtering.

Common Preprocessing Steps

• Strip unwanted characters/markup (e.g. HTML tags, punctuation, numbers, etc.).
• Break into tokens (keywords) on whitespace.
• Stem tokens to “root” words
  – computational \( \rightarrow \) comput
• Remove common stopwords (e.g. a, the, it, etc.).
• Detect common phrases (possibly using a domain specific dictionary).
• Build inverted index (keyword \( \rightarrow \) list of docs containing it).
Boolean Model

- A document is represented as a set of keywords.
- Queries are Boolean expressions of keywords, connected by AND, OR, and NOT, including the use of brackets to indicate scope.
  - [[Rio & Brazil]] || [Hilo & Hawaii] & hotel & !Hilton
- Output: Document is relevant or not. No partial matches or ranking.

Boolean Retrieval Model

- Popular retrieval model because:
  - Easy to understand for simple queries.
  - Clean formalism.
- Boolean models can be extended to include ranking.
- Reasonably efficient implementations possible for normal queries.

Boolean Models – Problems

- Very rigid: AND means all; OR means any.
- Difficult to express complex user requests.
- Difficult to control the number of documents retrieved.
  - All matched documents will be returned.
- Difficult to rank output.
  - All matched documents logically satisfy the query.
- Difficult to perform relevance feedback.
  - If a document is identified by the user as relevant or irrelevant, how should the query be modified?

Statistical Models

- A document is typically represented by a bag of words (unordered words with frequencies).
- Bag = set that allows multiple occurrences of the same element.
- User specifies a set of desired terms with optional weights:
  - Weighted query terms:
    - Q = < database 0.5; text 0.8; information 0.2 >
  - Unweighted query terms:
    - Q = < database; text; information >
- No Boolean conditions specified in the query.

Statistical Retrieval

- Retrieval based on similarity between query and documents.
- Output documents are ranked according to similarity to query.
- Similarity based on occurrence frequencies of keywords in query and document.
- Automatic relevance feedback can be supported:
  - Relevant documents “added” to query.
  - Irrelevant documents “subtracted” from query.

Issues for Vector Space Model

- How to determine important words in a document?
  - Word sense?
  - Word n-grams (and phrases, idioms,…) → terms
- How to determine the degree of importance of a term within a document and within the entire collection?
- How to determine the degree of similarity between a document and the query?
- In the case of the web, what is a collection and what are the effects of links, formatting information, etc.?
The Vector-Space Model

- Assume \( t \) distinct terms remain after preprocessing; call them index terms or the vocabulary.
- These “orthogonal” terms form a vector space.
  
  Dimension = \( t = |\text{vocabulary}| \)
- Each term, \( t_i \), in a document or query, \( j \), is given a real-valued weight, \( w_{ij} \).
- Both documents and queries are expressed as \( t \)-dimensional vectors:
  
  \[
  d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})
  \]

Document Collection

- A collection of \( n \) documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the “weight” of a term in the document; zero means the term has no significance in the document or it simply doesn’t exist in the document.

| \( T_1 \) | \( T_2 \) | \( \ldots \) | \( T_t \) |
| \( D_1 \) | \( w_{11} \) | \( w_{21} \) | \( \ldots \) | \( w_{t1} \) |
| \( D_2 \) | \( w_{12} \) | \( w_{22} \) | \( \ldots \) | \( w_{t2} \) |
| \( \vdots \) | \( \vdots \) | \( \vdots \) | \( \ddots \) | \( \vdots \) |
| \( D_n \) | \( w_{1n} \) | \( w_{2n} \) | \( \ldots \) | \( w_{tn} \) |

Graphic Representation

Example:

\[
D_1 = 2T_1 + 3T_2 + 5T_3 \quad D_2 = 3T_1 + 7T_2 + T_3 \quad Q = 0T_1 + 0T_2 + 2T_3
\]

- Is \( D_1 \) or \( D_2 \) more similar to \( Q \)?
- How to measure the degree of similarity? Distance? Angle? Projection?

Term Weights: Term Frequency

- More frequent terms in a document are more important, i.e. more indicative of the topic.

\[
f_{ij} = \text{frequency of term } i \text{ in document } j
\]

- May want to normalize term frequency (tf) across the entire corpus:

\[
\text{tf}_{ij} = \frac{f_{ij}}{\max\{f_{ij}\}}
\]

Term Weights: Inverse Document Frequency

- Terms that appear in many different documents are less indicative of overall topic.

\[
df_i = \text{document frequency of term } i = \text{number of documents containing term } i
\]

\[
\text{idf}_i = \log_2 (N/ df_i)
\]

- An indication of a term’s discrimination power.
- Log used to dampen the effect relative to \( tf \).

TF-IDF Weighting

- A typical combined term importance indicator is tf-idf weighting:

\[
w_{ij} = tf_{ij} \text{idf}_i = tf_{ij} \log_2 (N/ df_i)
\]

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.
- Many other ways of determining term weights have been proposed.
- Experimentally, tf-idf has been found to work well.
Computing TF-IDF -- An Example

Given a document containing terms with given frequencies:

A(3), B(2), C(1)

Assume collection contains 10,000 documents and
document frequencies of these terms are:

A(50), B(1300), C(250)

Then:

A: \( tf = \frac{3}{3}; \) \( idf = \log(10000/50) = 5.3; \) \( tf-idf = 5.3 \)
B: \( tf = \frac{2}{3}; \) \( idf = \log(10000/1300) = 2.0; \) \( tf-idf = 1.3 \)
C: \( tf = \frac{1}{3}; \) \( idf = \log(10000/250) = 3.7; \) \( tf-idf = 1.2 \)

Query Vector

- Query vector is typically treated as a
document and also tf-idf weighted.
- Alternative is for the user to supply weights
for the given query terms.

Similarity Measure

- A similarity measure is a function that computes
the degree of similarity between two vectors.

- Using a similarity measure between the query and
each document:
  - It is possible to rank the retrieved documents in the
    order of presumed relevance.
  - It is possible to enforce a certain threshold so that the
    size of the retrieved set can be controlled.

Similarity Measure - Inner Product

- Similarity between vectors for the document \( d \)
  and query \( q \)
  can be computed as the vector inner product:

  \[
  \text{sim}(d, q) = \sum_{i=1}^{n} d_i \cdot q_i = \sum_{i=1}^{n} w_{ij} \cdot w_{iq}
  \]

  where \( w_{ij} \) is the weight of term \( i \) in document \( j \)
  and \( w_{iq} \) is the weight of term \( i \) in the query

  - For binary vectors, the inner product is the number of
    matched query terms in the document (size of intersection).
  - For weighted term vectors, it is the sum of the products
    of the weights of the matched terms.

Properties of Inner Product

- The inner product is unbounded.

- Favors long documents with a large number of
  unique terms.

- Measures how many terms matched but not how
  many terms are not matched.

Inner Product -- Examples

Binary:

- \( D = 1, 1, 0, 1, 0, 1, 0 \)
- \( Q = 1, 0, 1, 0, 0, 1, 1 \)

Size of vector = size of vocabulary = 7
0 means corresponding term not found in
document or query

\( \text{sim}(D, Q) = 3 \)

Weighted:

- \( D_1 = 2T_1 + 1T_2 + 5T_3 \)
- \( D_2 = 3T_1 + 7T_2 + 1T_3 \)
- \( Q = 0T_1 + 0T_2 + 2T_3 \)

\( \text{sim}(D_1, Q) = 2 \cdot 0 + 3 \cdot 0 + 5 \cdot 2 = 10 \)
\( \text{sim}(D_2, Q) = 3 \cdot 0 + 7 \cdot 0 + 1 \cdot 2 = 2 \)
Cosine Similarity Measure

- Cosine similarity measures the cosine of the angle between two vectors.
- Inner product normalized by the vector lengths.

\[
\text{CosSim}(\mathbf{d}, \mathbf{q}) = \frac{\sum w_{ij} \cdot w_{ij}}{\sqrt{\sum w_{ij} \cdot \sum w_{ij}}}
\]

\[
\text{CosSim}(\mathbf{D}_1, \mathbf{Q}) = \frac{10}{\sqrt{(4+9+25)(0+0+4)}} = 0.81
\]

\[
\text{CosSim}(\mathbf{D}_2, \mathbf{Q}) = \frac{2}{\sqrt{(9+49+1)(0+0+4)}} = 0.13
\]

\[
\mathbf{D}_1 = 2\mathbf{T}_1 + 3\mathbf{T}_2 + 5\mathbf{T}_3
\]

\[
\mathbf{D}_2 = 3\mathbf{T}_1 + 7\mathbf{T}_2 + 1\mathbf{T}_3
\]

\[
\mathbf{Q} = 0\mathbf{T}_1 + 0\mathbf{T}_2 + 2\mathbf{T}_3
\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]

\[\mathbf{D}_1\]

\[\mathbf{D}_2\]

\[\mathbf{Q}\]

\[\theta_1 \theta_2 \theta_3\]

\[t_1 t_2 t_3\]