Information Retrieval: Retrieval Models

Luo Si
Department of Computer Science
Purdue University
Retrieval Models

Representation

Query

Retrieval Model

Retrieved Objects

Evaluation/Feedback

Indexed Objects

Retrieval Models

Information Need

Indexed Objects

Retrieved Objects

Evaluation/Feedback

Retrieval Models
Overview of Retrieval Models

Retrieval Models

- **Boolean**
- **Vector space**
  - Basic vector space
  - Extended Boolean
- **Probabilistic models**
  - Statistical language models
  - Two Possion model
  - Bayesian inference networks
- **Citation/Link analysis models**
  - Page rank
  - Hub & authorities
Retrieval Models: Outline

Retrieval Models

- Exact-match retrieval method
  - Unranked Boolean retrieval method
  - Ranked Boolean retrieval method

- Best-match retrieval method
  - Vector space retrieval method
  - Latent semantic indexing
Retrieval Models: Unranked Boolean

Unranked Boolean: Exact match method

- **Selection Model**
  - Retrieve a document iff it matches the precise query
  - Often return unranked documents (or with chronological order)

- **Operators**
  - Logical Operators: AND OR, NOT
  - Approximately operators: #1(white house) (i.e., within one word distance, phrase) #sen(Iraq weapon) (i.e., within a sentence)
  - String matching operators: Wildcard (e.g., ind* for india and indonesia)
  - Field operators: title(information and retrieval)…
Retrieval Models: Unranked Boolean

Unranked Boolean: Exact match method

• A query example

(#2(distributed information retrieval) OR (#1 (federated search)) AND author(#1(Jamie Callan) AND NOT (Steve)))
Retrieval Models: Unranked Boolean

WestLaw system: Commercial Legal/Health/Finance
Information Retrieval System

- Logical operators
- Proximity operators: Phrase, word proximity, same sentence/paragraph
- String matching operator: wildcard (e.g., ind*)
- Field operator: title(#1(“legal retrieval”)) date(2000)
- Citations: Cite (Salton)
Retrieval Models: Unranked Boolean

Advantages:
- Work well if user knows exactly what to retrieve
- Predicable; easy to explain
- Very efficient

Disadvantages:
- It is difficult to design the query; high recall and low precision for loose query; low recall and high precision for strict query
- Results are unordered; hard to find useful ones
- Users may be too optimistic for strict queries. A few very relevant but a lot more are missing
Retrieval Models: Ranked Boolean

Ranked Boolean: Exact match

- Similar as unranked Boolean but documents are ordered by some criterion

Retrieve docs from Wall Street Journal Collection

Query: (Thailand AND stock AND market)

Which word is more important?

Many “stock” and “market”, but fewer “Thailand”. Fewer may be more indicative

Term Frequency (TF): Number of occurrence in query/doc; larger number means more important

Inversed Document Frequency (IDF):

Larger means more important

There are many variants of TF, IDF: e.g., consider document length

Total number of docs

Number of docs contain a term
Retrieval Models: Ranked Boolean

Ranked Boolean: Calculate doc score

- Term evidence: Evidence from term $i$ occurred in doc $j$: $(tf_{ij})$ and $(tf_{ij} \times idf_i)$
- AND weight: minimum of argument weights
- OR weight: maximum of argument weights

Query: (Thailand AND stock AND market)
Retrieval Models: Ranked Boolean

Advantages:
- All advantages from unranked Boolean algorithm
  - Works well when query is precise; predictive; efficient
- Results in a ranked list (not a full list); easier to browse and find the most relevant ones than Boolean
- Rank criterion is flexible: e.g., different variants of term evidence

Disadvantages:
- Still an exact match (document selection) model: inverse correlation for recall and precision of strict and loose queries
- Predictability makes user overestimate retrieval quality
Retrieval Models: Vector Space Model

Vector space model

- Any text object can be represented by a term vector
  - Documents, queries, passages, sentences
  - A query can be seen as a short document

- Similarity is determined by distance in the vector space
  - Example: cosine of the angle between two vectors

- The SMART system
  - Developed at Cornell University: 1960-1999
  - Still quite popular

- The Lucene system
  - Open source information retrieval library; (Based on Java)
  - Work with Hadoop (Map/Reduce) in large scale app (e.g., Amazon Book)
Retrieval Models: Vector Space Model

Vector space model vs. Boolean model

- **Boolean models**
  - Query: a Boolean expression that a document must satisfy
  - Retrieval: Deductive inference

- **Vector space model**
  - Query: viewed as a short document in a vector space
  - Retrieval: Find similar vectors/objects
Retrieval Models: Vector Space Model

Vector representation

<table>
<thead>
<tr>
<th></th>
<th>Java</th>
<th>Sun</th>
<th>Starbucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Query</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>
Retrieval Models: Vector Space Model

Vector representation
Retrieval Models: Vector Space Model

Give two vectors of query and document

- query as \( q = (q_1, q_2, \ldots, q_n) \)
- document as \( d_j = (d_{j_1}, d_{j_2}, \ldots, d_{jn}) \)
- calculate the similarity

Cosine similarity: Angle between vectors

\[
sim(q, d_j) = \cos(\theta(q, d_j))
\]

\[
\cos(\theta(q, d_j)) = \frac{q \cdot d_j}{\|q\| \|d_j\|} = \frac{q_1 d_{j_1} + q_2 d_{j_2} + \ldots + q_n d_{jn}}{\sqrt{q_1^2 + \ldots + q_n^2} \sqrt{d_{j_1}^2 + \ldots + d_{jn}^2}}
\]
Retrieval Models: Vector Space Model

Vector representation

<table>
<thead>
<tr>
<th></th>
<th>Java</th>
<th>Sun</th>
<th>Starbucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Query</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Similarity Score</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>0.59</td>
<td>0.99</td>
<td>0.70</td>
</tr>
</tbody>
</table>
Retrieval Models: Vector Space Model

Vector Coefficients

- The coefficients (vector elements) represent term evidence/term importance
- It is derived from several elements
  - Document term weight: Evidence of the term in the document/query
  - Collection term weight: Importance of term from observation of collection
  - Length normalization: Reduce document length bias
- Naming convention for coefficients:

\[ q_{k} \cdot d_{j,k} = DCL \cdot DCL \]

First triple represents query term; second for document term
Retrieval Models: Vector Space Model

Common vector weight components:

- **Inc.ltc**: widely used term weight
  - “l”: $\log(tf)+1$
  - “n”: no weight/normalization
  - “t”: $\log(N/df)$
  - “c”: cosine normalization

\[
\frac{q_1 d_{j_1} + q_2 d_{j_2} + \ldots + q_n d_{j_n}}{\sqrt{\sum_k [(\log tf_q (k) + 1)^2]}} \sqrt{\sum_k [(\log tf_j (k) + 1 \log \frac{N}{df (k)})^2]}
\]
Retrieval Models: Vector Space Model

Other Common vector weight components:

- “d”: $1+\ln(1+\ln(tf))$
- “t”: $\log((N/df)$
- “b”: $1/(0.8+0.2*\text{doclen}/\text{avg\_doclen})$
Retrieval Models: Vector Space Model

- Standard vector space
  - Represent query/documents in a vector space
  - Each dimension corresponds to a term in the vocabulary
  - Use a combination of components to represent the term evidence in both query and document
  - Use similarity function to estimate the relationship between query/documents (e.g., cosine similarity)
Retrieval Models: Vector Space Model

Advantages:

- Best match method; it does not need a precise query
- Generated ranked lists; easy to explore the results
- Simplicity: easy to implement
- Effectiveness: often works well
- Flexibility: can utilize different types of term weighting methods
- Used in a wide range of IR tasks: retrieval, classification, summarization, content-based filtering…
Retrieval Models: Vector Space Model

Disadvantages:

- Hard to choose the dimension of the vector ("basic concept"); terms may not be the best choice
- Assume independent relationship among terms
- Heuristic for choosing vector operations
  - Choose of term weights
  - Choose of similarity function
- Assume a query and a document can be treated in the same way
Retrieval Models: Vector Space Model

Disadvantages:

- Hard to choose the dimension of the vector ("basic concept"); terms may not be the best choice
- Assume independent relationship among terms
- Heuristic for choosing vector operations
  - Choose of term weights
  - Choose of similarity function
- Assume a query and a document can be treated in the same way
Retrieval Models: Vector Space Model

What are good vector representation:

- Orthogonal: the dimensions are linearly independent ("no overlapping")
- No ambiguity (e.g., Java)
- Wide coverage and good granularity
- Good interpretation (e.g., representation of semantic meaning)
- Many possibilities: words, stemmed words, "latent concepts"...
Retrieval Models: Latent Semantic Indexing

Dual space of terms and documents

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>machine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>gene</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mutation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>expression</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Retrieval Models: Latent Semantic Indexing

Latent Semantic Indexing (LSI): Explore correlation between terms and documents

- Two terms are correlated (may share similar semantic concepts) if they often co-occur
- Two documents are correlated (share similar topics) if they have many common words

Latent Semantic Indexing (LSI): Associate each term and document with a small number of semantic concepts/topics
Retrieval Models: Latent Semantic Indexing

Using singular value decomposition (SVD) to find the small set of concepts/topics

\( m \): number of concepts/topics

Representation of concept in document space; \( V^TV = I_m \)

Representation of concept in term space; \( U^TU = I_m \)

Diagonal matrix: concept space

\( X = USV^T \)

\( U^TU = I_m \)

\( V^TV = I_m \)
Retrieval Models: Latent Semantic Indexing

Using singular value decomposition (SVD) to find the small set of concepts/topics

$m$: number of concepts/topics
Retrieval Models: Latent Semantic Indexing

Properties of Latent Semantic Indexing

- Diagonal elements of S as $S_k$ in descending order, the larger the more important
- $x_k = \sum_{i \leq k} u_k S_k v_k$ is the rank-k matrix that best approximates $X$, where $u_k$ and $v_k$ are the column vector of U and V
Retrieval Models: Latent Semantic Indexing

Other properties of Latent Semantic Indexing

- The columns of $U$ are eigenvectors of $XX^T$
- The columns of $V$ are eigenvectors of $X^TX$
- The singular values on the diagonal of $S$, are the positive square roots of the nonzero eigenvalues of both $AA^T$ and $A^TA$
Retrieval Models: Latent Semantic Indexing

\[
\begin{array}{cccccccc}
\text{information} & C1 & C2 & C3 & C4 & B1 & B2 & B3 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{retrieval} & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{machine} & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
\text{learning} & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
\text{system} & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
\text{protein} & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\text{gene} & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
\text{mutation} & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\text{expression} & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
\end{array}
\]

\[
\begin{pmatrix}
-0.3467 \\
-0.3467 \\
-0.6215 \\
-0.4544 \\
-0.3329 \\
-0.0452 \\
-0.2245 \\
-0.0452 \\
-0.0401 \\
\end{pmatrix}
\begin{pmatrix}
3.1395 & 0 \\
0 & 2.3912 \\
\end{pmatrix}
\begin{pmatrix}
-0.5248 & -0.5635 & -0.5202 & -0.3427 & -0.0843 & -0.1003 & -0.0415 \\
-0.1578 & -0.1695 & 0.1462 & -0.0550 & 0.3754 & 0.6402 & 0.6092 \\
\end{pmatrix}
\]
## Retrieval Models: Latent Semantic Indexing

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>information</strong></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>retrieval</strong></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>machine</strong></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>learning</strong></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>system</strong></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>protein</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>gene</strong></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>mutation</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>expression</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{pmatrix}
-0.3467 & -0.1369 \\
-0.3467 & -0.1369 \\
-0.6215 & -0.0987 \\
-0.4544 & -0.0327 \\
-0.3329 & -0.0049 \\
-0.0452 & 0.5225 \\
-0.2245 & 0.4859 \\
-0.0452 & 0.5225 \\
-0.0401 & 0.4118 \\
\end{pmatrix}
\begin{pmatrix}
3.1395 & 0 \\
0 & 2.3912 \\
\end{pmatrix}
\begin{pmatrix}
-0.5248 & -0.5635 & -0.5202 & -0.3427 & -0.0843 & -0.1003 & -0.0415 \\
-0.1578 & -0.1695 & 0.1462 & -0.0550 & 0.3754 & 0.6402 & 0.6092 \\
\end{pmatrix}
\]
Retrieval Models: Latent Semantic Indexing

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>machine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>gene</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mutation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>expression</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{pmatrix}
-0.3467 & -0.1369 \\
-0.3467 & -0.1369 \\
-0.6215 & -0.0987 \\
-0.4544 & -0.0327 \\
-0.3329 & -0.0049 \\
-0.0452 & 0.5225 \\
-0.2245 & 0.4859 \\
-0.0452 & 0.5225 \\
-0.0401 & 0.4118
\end{pmatrix}
\begin{pmatrix}
3.1395 & 0 \\
0 & 2.3912
\end{pmatrix}
\begin{pmatrix}
-0.5248 & -0.5635 & -0.5202 & -0.3427 & -0.0843 & -0.1003 & -0.0415 \\
-0.1578 & -0.1695 & 0.1462 & -0.0550 & 0.3754 & 0.6402 & 0.6092
\end{pmatrix}
Retrieval Models: Latent Semantic Indexing

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>machine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>gene</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mutation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>expression</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{pmatrix}
-0.3467 & -0.1369 \\
-0.3467 & -0.1369 \\
-0.6215 & -0.0987 \\
-0.4544 & -0.0327 \\
-0.3329 & -0.0049 \\
-0.0452 & 0.5225 \\
-0.2245 & 0.4859 \\
-0.0452 & 0.5225 \\
-0.0401 & 0.4118
\end{pmatrix}
\begin{pmatrix}
3.1395 & 0 \\
0 & 2.3912
\end{pmatrix}
\begin{pmatrix}
-0.5248 & -0.5635 & -0.5202 & -0.3427 & -0.0843 & -0.1003 & -0.0415 \\
-0.1578 & -0.1695 & 0.1462 & -0.0550 & 0.3754 & 0.6402 & 0.6092
\end{pmatrix}
Importance of concepts

Reflect Error of Approximating $X$ with small $S$

Size of $S_k$
Retrieval Models: Latent Semantic Indexing

- SVD representation
  - Reduce high dimensional representation of document or query into low dimensional concept space
  - SVD tries to preserve the Euclidean distance of document/term vector

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>machine</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>system</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>gene</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mutation</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>expression</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept 1</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.5248</td>
<td>-0.1578</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept 2</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.5635</td>
<td>-0.1695</td>
</tr>
</tbody>
</table>
Retrieval Models: Latent Semantic Indexing

- SVD representation

Representation of the documents in two dimensional concept space
Retrieval Models: Latent Semantic Indexing

Retrieval with respect to a query

- Map (fold-in) a query into the representation of the concept space
  \[ q_{\text{concept}} = q^T U_k \text{Inv}(S_k) \]

- Use the new representation of the query to calculate the similarity between query and all documents
  - Cosine Similarity
Retrieval Models: Latent Semantic Indexing

Qry: Machine Learning Protein

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>machine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>gene</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mutation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>expression</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Representation of the query in the term vector space:

\[
[0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^T
\]
Retrieval Models: Latent Semantic Indexing

Representation of the query in the latent semantic space (2 concepts):

\[ q' = q^T U_k \text{Inv}(S_k) = [-0.3571 \ 0.1635]^T \]
## Retrieval Models: Latent Semantic Indexing

Comparison of Retrieval Results in term space and concept space

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>retrieval</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>machine</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>system</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>protein</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>gene</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>mutation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>expression</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Query Similarity in term space:
- Information: 0.29
- Retrieval: 0.58
- Machine: 0.58
- Learning: 0.82

Query Similarity in concept space:
- Information: 0.75
- Retrieval: 0.75
- Machine: 0.98
- Learning: 0.83

Qry: Machine Learning Protein
Problems with latent semantic indexing
- Difficult to decide the number of concepts
- There is no probabilistic interpolation for the results
- The complexity of the LSI model obtained from SVD is costly
Retrieval Models: Outline

Retrieval Models

- Exact-match retrieval method
  - Unranked Boolean retrieval method
  - Ranked Boolean retrieval method

- Best-match retrieval
  - Vector space retrieval method
  - Latent semantic indexing