INFSCI 2140
Information Storage and Retrieval
Lecture 3: Models of IR: Advanced Models

Peter Brusilovsky
http://www2.sis.pitt.edu/~peterb/2140-051/

Overview

- Boolean Models and Databases
- Problems with Boolean Models
- Simple Vector Model
- Extended Boolean Model
- Fuzzy model and Probabilistic Model
- Natural Language
- Things to mention

Models: Classic and New

- Boolean Model
  - Classic
  - Extended
  - Fuzzy
- Vector Model
  - Classic
  - Others (generalized, LSI, Neural Networks)
- Probabilistic Model

Benefits of Boolean model

- Integration of formatted databases and full-text document collections
- Object oriented databases
  - A record is an object and denotes a document
- Boolean Queries work with both (unlike vector model)!
- Example of Boolean search in complex databases

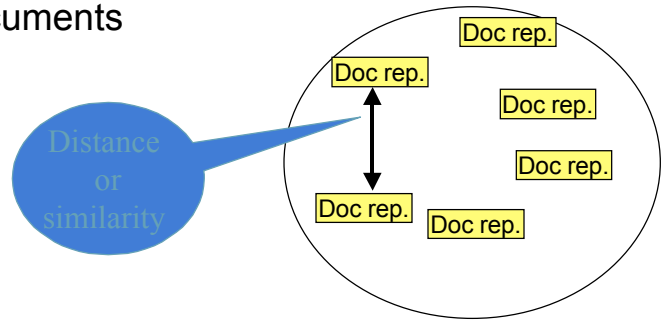
Problems of Boolean Model

- Can't assign significance for terms
- Boolean queries are hard for users
  - misstated queries
- Order of precedence of OR and AND does matter (A or B and C)
- The problem of NOT - open corpus
- Hard to make efficient queries
- Ordering of the results
- Controlling the size (Boolean function!)

Better GUI for Boolean search

Document Space

- Document space is organized in some way. For example it could be possible to calculate a distance or a similarity between different documents



Designer's prospects for Q & M

- Can we consider query as a document?
- No
  - let's consider a query as a characteristic function defined for document space
- Yes
  - let's put a query into a space and calculate "closeness" or similarity

The query is a part of the document space

- Document and query representation are built in order to catch the meaning of documents and queries.
- We can measure the distance between documents and query using the some method that we have used to calculate the document similarity
- Main problem: how to measure the distance between documents and query?

Vector Model: Documents

- In a vector model each document is represented by a *term vector*
- 0-1 vector (simple)
  - **0** if the term is not present in the document
  - **1** if the term is present in the document
- Weighed vector
  - **0** if the term is not present in the document
  - **term weight** if the term is present in the document (the term weight is usually related to the frequency of the term itself)

Vector Model: Queries

- The query is considered as a document so it is represented with a vector
- The system must be designed to ensure that the comparison is always based on comparing the same terms (in the query and in the document)
- The matching between query and documents can be made using the same techniques used to calculate document similarity.

Vector Model: Weighted Queries

- The user can be asked to assign weights to the terms of the query
- If the user assign weights freely some normalization is necessary to ensure that the weights used are compatible with those assigned by the system to documents

Weighted Queries: Normalization

- If u is the weight assigned by the user

$$u_{min} < u < u_{max}$$

and the scale of the system is between s_{min} and s_{max} then the new weight is

$$s = \frac{s_{max}(u - u_{min}) + s_{min}(u_{max} - u)}{u_{max} - u_{min}}$$

Vector Model: Presentation

- It is possible to calculate a distance between document and query - easier to deiced what documents the system should output to the user and how to order
- The document can be ranked so:
  - a fixed number of documents (the first 100 for example)
  - the document whose similarity is above a threshold (perhaps specified by the user)

Vector Model: Benefits and Problems

- Vector queries a bit easier for the user -- simply list of terms
- Can provide weights for terms
- Allow “like this” queries
- Basis for personalization and profiling
- BUT: It is not possible to express easily logical connectives

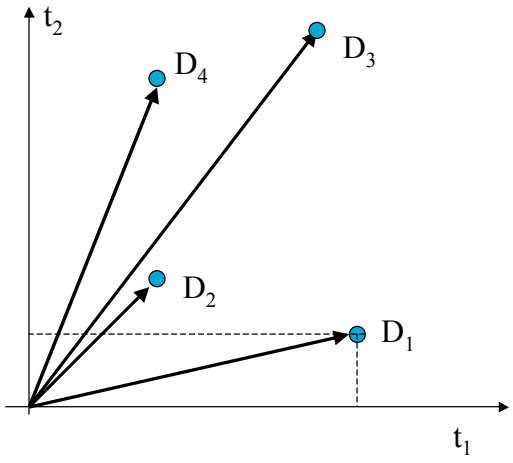
Vector-based matching

- Documents and queries are represented as a vector of weights, each elements of the vector corresponds to the importance of the index term in the document
- Measures
  - Distance measure between documents using metrics (L_1, L_2, etc)
  - Angular (cosine) measure between document vectors

Distance Measure

Distance Measure using Euclidean Norm (L_2 Norm)

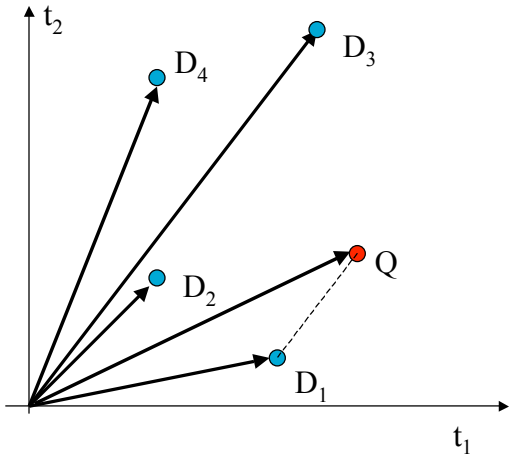
- Imagine that documents are represented using only two terms t_1 t_2



Distance Measure

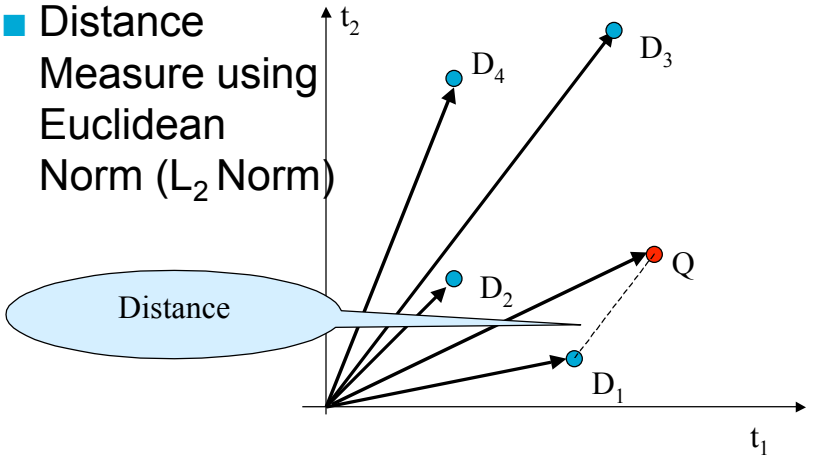
Distance Measure using Euclidean Norm (L_2 Norm)

- The query Q is also represented using the same two terms t_1 t_2



Distance Measure

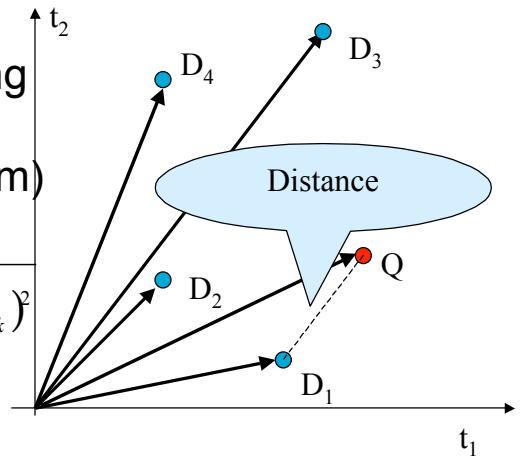
- Distance Measure using Euclidean Norm (L_2 Norm)



Distance Measure

- Distance Measure using Euclidean Norm (L_2 Norm)

$$ed(D,Q) = \sqrt{\sum_{i=1}^{n} (t_k - q_k)^2}$$



Distance Measure

- We need some function to transform the distance between documents into a similarity measure
- A linear transformation is not effective:

  if the distance is m and k a constant linear transformation is like:

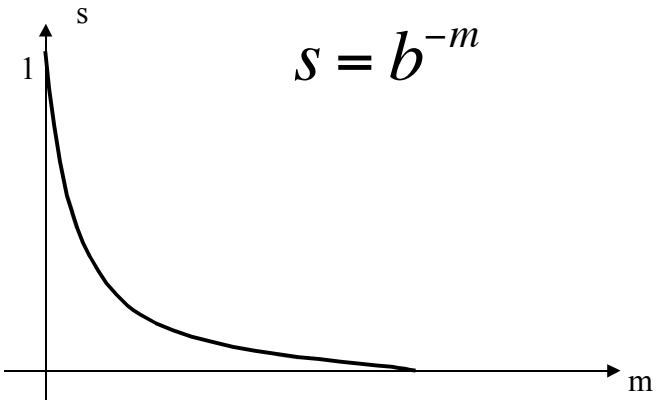
$$s = k - m$$

$s < 0$ if $m > k$

Distance Measure

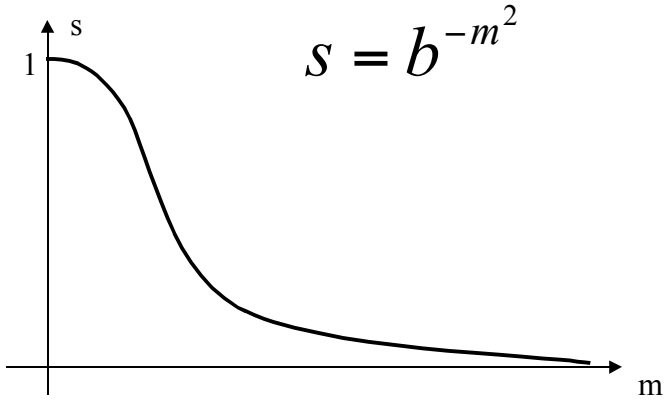
- A more convenient function is:

$$s = b^{-m}$$



Distance Measure

- Another function is:



$$s = b^{-m^2}$$

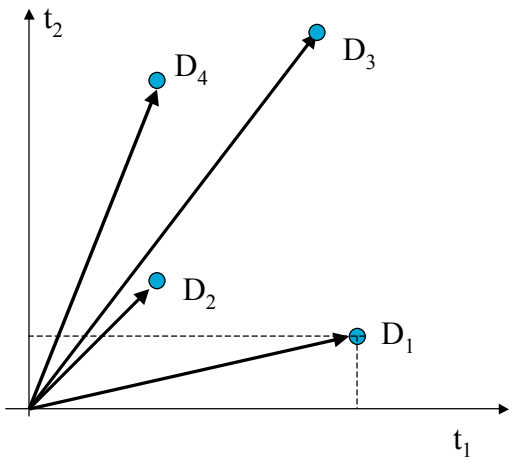
Angular Measure

- Angular Measure
  - is a measure that does not take into account the length (the norm of the vectors) that represent documents and query
- The idea
  - Vectors pointing to the same direction are close to each other.
- How to measure?

Cosine Measure

Cosine Measure

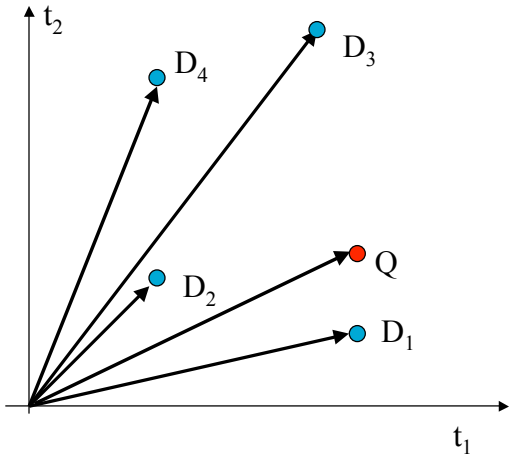
– Imagine that documents are represented using only two terms t_1 t_2



Cosine Measure

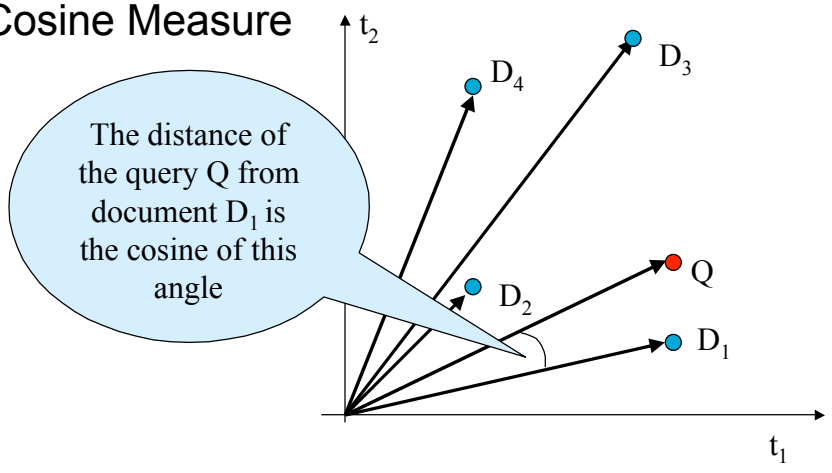
Cosine Measure

– The query is also represented using only two terms t_1 t_2



Cosine Measure

Cosine Measure



Cosine Measure

- Cosine Measure
  - t_k is the weight of the term k in the doc. D
  - q_k is the weight of the term k in the query Q
  - n is the total number of index terms

$$\sigma(D,Q) = \frac{\sum_{k=1}^{n} t_k \cdot q_k}{\sqrt{\sum_{k=1}^{n} t_k^{\,2}} \cdot \sqrt{\sum_{k=1}^{n} q_k^{\,2}}}$$

Demo of the system

http://kt2.exp.sis.pitt.edu:8080/VectorModel/index.html

Boolean Model

- Documents form a large set
- A query defines a subset
- Elementary query has a clearly defined subset
  - Each document in the subset matches the query
- To make a complex query one can use Boolean functions for set operation

Extended Boolean Model

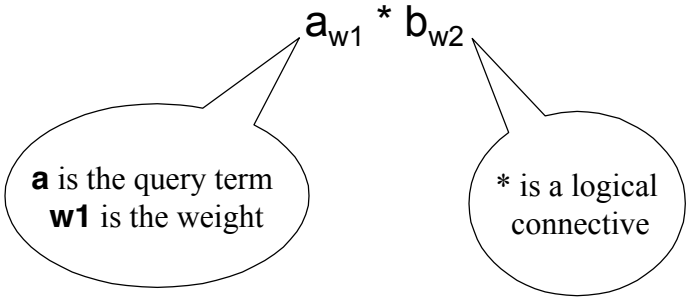
- Elementary query can have a weight!
  - weight $w \in [0, 1]$
- All Boolean operations are defined for weighed queries
  - $A(w_1)$ or $B(w_2)$
  - $A(w_1)$ and $B(w_2)$
  - $A(w_1)$ and not $B(w_2)$

Fuzzy Model

- Fuzzy matching is defined for elementary terms
  - A fuzzy set corresponds to every elementary term
- Boolean operations are reconsidered for new situation as fuzzy set operations
  - AND as min
  - OR as max
  - NOT as 1-x

Extended Boolean queries

- Attempts to introduce weights in Boolean queries
- The query



Extended Boolean queries

- The weight operation depends on a distance between the two document sets:
  - The set **A** corresponding to the term **a**
  - The set **B** corresponding to the term **b**

Extended Boolean queries

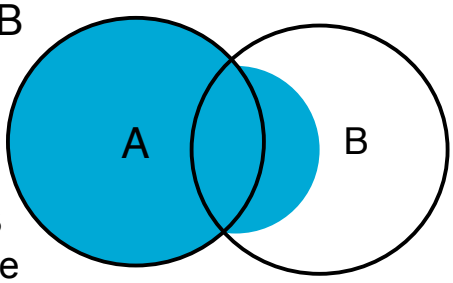
- A specific set **S** depends on the operation * considered
- To explain the mechanism take w_1=1.0 (including all elements of the set **A**) and consider w_2 increasing from 0.0 to 1.0 (so the set **B** from not considered if w_2=0.0 to the standard Boolean operation if w_2=1.0)

Ext. Boolean queries: OR

- A_1 OR B_0 =A
- A_1 OR B_1=A OR B

The set S is B-A

the weighted OR gradually includes more elements of B beginning with those closest to A

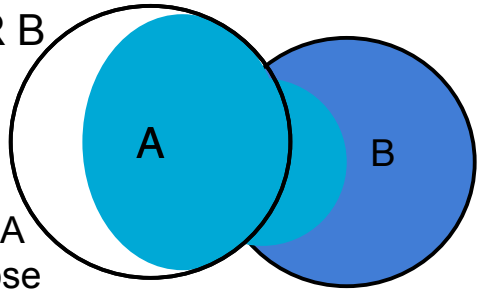


Ext. Boolean queries: OR

- A_0 OR B_1 =B
- A_1 OR B_1=A OR B

  The set S is A-B

  the weighted OR gradually includes more elements of A beginning with those closest to B

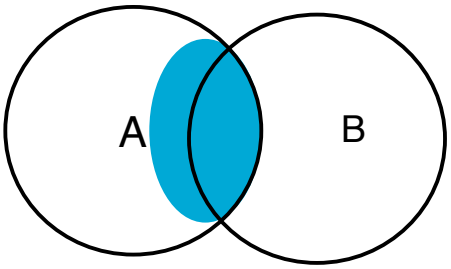


Ext. Boolean queries: AND

- A_1 AND B_0 =A
- A_1 AND B_1=A AND B

The set S is A-B

the weighted AND gradually delete those elements of A that are farthest from A AND B

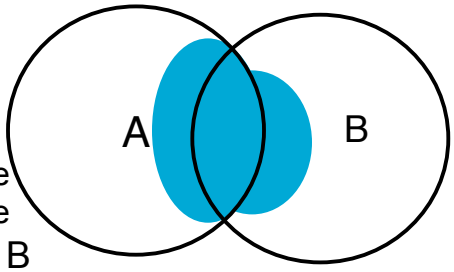


Ext. Boolean queries: AND

- A_0 AND B_1 =B
- A_1 AND B_1=A AND B

The set S is B-A

the weighted AND gradually delete those elements of B that are farthest from A AND B



Extended Boolean Queries

- It is not clear that this model results in significant better retrieval system performances
- It is difficult to predict the effect of any specific weighting

Fuzzy Sets

- In ordinary set theory an element is either in a given set or not
- In a fuzzy set each element has associate a membership function
- Expensive car
  - Price > 50K with probability 0.6
  - 40K < Price <= 50K with probability 0.3
  - 30K < Price <= 40K with probability 0.1

Fuzzy sets

- Suppose U the universal set (the set of all the entities that can be considered)
- A fuzzy set S can be defined as

$$\left\{ \langle x, \mu_s(x) \rangle \middle| \mu_s > 0 \right\}$$

where x is a member of U and μ_s is the membership function

- By definition every x for which $\mu_s(x)>0$ is an element of S to some extent

Fuzzy sets

- Given two fuzzy sets A and B

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))$$
$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))$$
$$\mu_{\bar{A}}(x) = 1.0 - \mu_A(x)$$

Fuzzy information retrieval

- The system and often the user cannot accurately tell whether a given document will meet the information need
- This uncertainty is modeled in a fuzzy evaluation of the document with respect to the query

Fuzzy Matching

- In fuzzy matching a query q is used to define a fuzzy set (the query fuzzy set)
- Then it is possible for each document d_j to calculate the membership to this set using a membership function:

$$\mu_q(d_j)$$

- Document are retrieved if the membership function is over a defined threshold

Fuzzy Matching

- Fuzzy matching appears to be similar to the probability matching but we do not estimate the probability of a document to be relevant but the degree of relevance for a document to a particular query

"Size" problem

- Test document collections an be a very small subset of a real database and processing of all the documents can be time-consuming.
- To solve this problem it can be helpful to make the matching in two steps:
  1. Using simple and fast techniques extract a subset of candidates
  2. Refine the set using more sophisticated techniques

Things to mention

- Natural language retrieval
- Missing terms in vector model
- Proximity
- Weighting
- Complexity and Scaling
- Data Fusion and Meta-Search

Assignment 3 (1 of 3)

Using the system at the URL (vector model)
http://kt2.exp.sis.pitt.edu:8080/VectorModel/index.html
answer to the following questions:

1. Explain why for the query Brown(1) you will have the cosine measure 0 with document D9,D10 and D12 (look at the formula)
2. Report a query that will rank 1 the document D3
3. Report a query that will rank 1 the document D9
4. Explain why for the following query:
   Quick(1);Dog(2);Jumps(1);Over(1);Lazy(1)
   You have the cosine measure with D10 equal to 1 (look at the formula and the document)

Assignment 3 (2a of 3)

- Consider the two documents sets

  A={4,7,18,21,25}

  B={1,5,7,18,22,25}

  Corresponding to the terms A and B and the query ($A_{0.7}$ and $B_{0.5}$). Find, using the system at the URL :

http://www2.sis.pitt.edu/~ir/Projects/Spg01/FinalProjects/HoahDerSu/

which documents satisfy the query

Assignment 3 (2b of 3)

Consider the two documents sets

A={1,3,5,7,10,11,13}

B={1,2,4,6,8,10,12,14}

Corresponding to the terms A and B and the query

(A_x and B_y).

Using the system at the URL

http://www2.sis.pitt.edu/~ir/Projects/Spg01/FinalProjects/HoahDerSu/

Find, if exist, the values of x and y that give the following retrieved sets:

S_1={1,10,2,4,3,5} , S_2={10,2,4,5} , S_3={1,10}, S_4={1,2,4, 7}

Assignment 3 (2c of 3)

Consider the two documents sets

A={1,3,5,7,10,11,13}

B={1,2,4,6,8,10,12,14}

Corresponding to the terms A and B and the query

(A_x or B_y).

Using the system at the URL

http://www2.sis.pitt.edu/~ir/Projects/Spg01/FinalProjects/HoahDerSu/

Find, if exist, the values of x and y that give the following retrieved sets:

S_1={1,10,2,4,3,5} , S_2={10,2,4,7} , S_3={1,10} , S_4={1,2,10, 7}

Assignment 3 (3 of 3)

- Considering the document represented by the vector D=(0.5 0.6) and the query Q=(0.25 0.5) calculate the similarity _(D,Q) using the euclidean distance _ and the two similarity functions

$$\sigma = b^{-\mu}$$

$$\sigma = b^{-\mu^2}$$