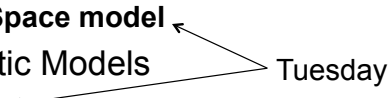


Information Retrieval

INFO 4300 / CS 4300

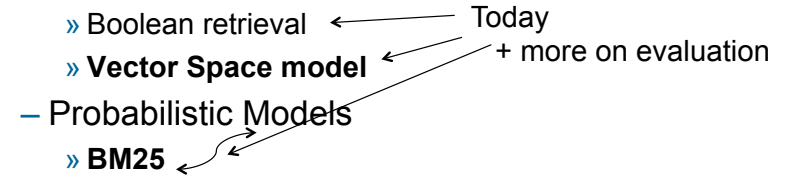
■ Retrieval models

- Older models
 - » Boolean retrieval
 - » **Vector Space model**
 - Probabilistic Models
 - » **BM25**
 - » Language models
 - Combining evidence
 - » Inference networks
 - » Learning to Rank
- Tuesday
- 

Information Retrieval

INFO 4300 / CS 4300

■ Retrieval models

- Older models
 - » Boolean retrieval
 - » **Vector Space model**
 - Probabilistic Models
 - » **BM25**
 - » Language models
 - Combining evidence
 - » Inference networks
 - » Learning to Rank
- Today
- + more on evaluation
- 

Retrieval Model Overview

- Older models
 - **Boolean retrieval**
 - Vector Space model
- Probabilistic Models
 - BM25
 - Language models
- Combining evidence
 - Inference networks
 - Learning to Rank

Boolean Retrieval

- Two possible outcomes for query processing
 - TRUE and FALSE
 - “exact-match” retrieval
 - simplest form of ranking
- Query usually specified using Boolean operators
 - AND, OR, NOT
 - proximity operators also used

Boolean Retrieval

- Advantages
 - Results are predictable, relatively easy to explain
 - Many different features can be incorporated
 - Efficient processing since many documents can be eliminated from search
- Disadvantages
 - Effectiveness depends entirely on user
 - Simple queries usually don't work well
 - Complex queries are difficult

Example: searching “by numbers”

- Sequence of queries driven by number of retrieved documents
 - e.g. “lincoln” search of news articles
 - president AND lincoln
 - president AND lincoln AND NOT (automobile OR car)
 - president AND lincoln AND biography AND life AND birthplace AND gettysburg AND NOT (automobile OR car)
 - president AND lincoln AND (biography OR life OR birthplace OR gettysburg) AND NOT (automobile OR car)

Vector Space Model

- Documents ranked by distance between points representing query and documents
 - *Similarity* measure more common than a distance or *dissimilarity* measure
 - e.g. Cosine correlation

$$\text{Cosine}(D_i, Q) = \frac{\sum_{j=1}^t d_{ij} \cdot q_j}{\sqrt{\sum_{j=1}^t d_{ij}^2 \cdot \sum_{j=1}^t q_j^2}}$$

Similarity Calculation

- Consider two documents D_1, D_2 and a query Q

$$\begin{aligned} \gg D_1 &= (0.5, 0.8, 0.3), D_2 = (0.9, 0.4, 0.2), Q = (1.5, 1.0, 1.0) \\ \text{Cosine}(D_1, Q) &= \frac{(0.5 \times 1.5) + (0.8 \times 1.0)}{\sqrt{(0.5^2 + 0.8^2 + 0.3^2)(1.5^2 + 1.0^2)}} \\ &= \frac{1.55}{\sqrt{(0.98 \times 3.25)}} = 0.87 \end{aligned}$$

$$\begin{aligned} \text{Cosine}(D_2, Q) &= \frac{(0.9 \times 1.5) + (0.4 \times 1.0)}{\sqrt{(0.9^2 + 0.4^2 + 0.2^2)(1.5^2 + 1.0^2)}} \\ &= \frac{1.75}{\sqrt{(1.01 \times 3.25)}} = 0.97 \end{aligned}$$

Term Weights

- *tf.idf* weight

- Term frequency weight measures importance

in document: $tf_{ik} = \frac{f_{ik}}{\sum_{j=1}^t f_{ij}}$

- Inverse document frequency measures

importance in collection: $idf_k = \log \frac{N}{n_k}$

- Some heuristic modifications

$$d_{ik} = \frac{(\log(f_{ik})+1) \cdot \log(N/n_k)}{\sqrt{\sum_{k=1}^t [(\log(f_{ik})+1.0) \cdot \log(N/n_k)]^2}}$$

Information Retrieval

INFO 4300 / CS 4300

- Retrieval models

- Older models

» Boolean retrieval ← Today
 » **Vector Space model** ← + more on evaluation

- Probabilistic Models

» **BM25**

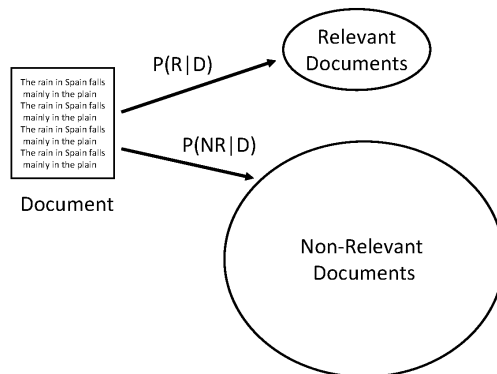
$$\sum_{i \in Q} \log \frac{(r_i+0.5)^j (R-r_i+0.5)}{(n_i-r_i+0.5) (N-n_i-R+r_i+0.5)} \cdot \frac{(k_1+1)f_i}{K+f_i} \cdot \frac{(k_2+1)qf_i}{k_2+qf_i}$$

– **Combining evidence**

» Inference networks

» Learning to Rank

IR as Classification



Bayes Classifier

- Bayes Decision Rule

- A document D is relevant if $P(R|D) > P(NR|D)$

- Estimating probabilities

- use Bayes Rule

$$P(R|D) = \frac{P(D|R)P(R)}{P(D)}$$

- classify a document as relevant if

$$\frac{P(D|R)}{P(D|NR)} > \frac{P(NR)}{P(R)}$$

» lhs is **likelihood ratio**

Estimating P(D|R)

- Assume term independence

$$P(D|R) = \prod_{i=1}^t P(d_i|R)$$

- Binary independence model**

- document represented by a vector of t binary features indicating term occurrence (or non-occurrence)
- p_i is probability that term i occurs (i.e., has value 1) in relevant document, s_i is probability of occurrence in non-relevant document

Binary Independence Model

$$\begin{aligned} \frac{P(D|R)}{P(D|NR)} &= \prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \prod_{i:d_i=0} \frac{1-p_i}{1-s_i} \\ &= \prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \left(\prod_{i:d_i=1} \frac{1-s_i}{1-p_i} \cdot \prod_{i:d_i=1} \frac{1-p_i}{1-s_i} \right) \cdot \prod_{i:d_i=0} \frac{1-p_i}{1-s_i} \\ &= \prod_{i:d_i=1} \frac{p_i(1-s_i)}{s_i(1-p_i)} \cdot \prod_i \frac{1-p_i}{1-s_i} \end{aligned}$$

Binary Independence Model

- Scoring function is

$$\sum_{i:d_i=1} \log \frac{p_i(1-s_i)}{s_i(1-p_i)}$$

- Query provides information about relevant documents.
- If we assume p_i constant, s_i approximated by entire collection, get **idf-like weight**

$$\log \frac{0.5(1-\frac{n_i}{N})}{\frac{n_i}{N}(1-0.5)} = \log \frac{N-n_i}{n_i}$$

Contingency Table

	Relevant	Non-relevant	Total
$d_i = 1$	r_i	$n_i - r_i$	n_i
$d_i = 0$	$R - r_i$	$N - n_i - R + r_i$	$N - n_i$
Total	R	$N - R$	N

$$p_i = (r_i + 0.5)/(R + 1)$$

$$s_i = (n_i - r_i + 0.5)/(N - R + 1)$$

Gives scoring function:

$$\sum_{i:d_i=q_i=1} \log \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}$$

BM25

- Popular and effective ranking algorithm based on binary independence model
 - adds document and query term weights

$$\sum_{i \in Q} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \cdot \frac{(k_1 + 1)f_i}{K + f_i} \cdot \frac{(k_2 + 1)qf_i}{k_2 + qf_i}$$

- k_1 , k_2 and K are parameters whose values are set empirically
 - $K = k_1((1 - b) + b \cdot \frac{dl}{avdl})$ dl is doc length
- Typical TREC value for k_1 is 1.2, k_2 varies from 0 to 1000, $b = 0.75$

- r_i is the # of relevant documents containing term i
 - (set to 0 if no relevancy info is known)
- n_i is the # of docs containing term i
- N is the total # of docs in the collection
- R is the number of relevant documents for this query
 - (set to 0 if no relevancy info is known)
- f_i is the frequency of term i in the doc under consideration
- qf_i is the frequency of term i in the query
- k_1 determines how the tf component of the term weight changes as f_i increases. (if 0, then tf component is ignored.) Typical value for TREC is 1.2; so f_i is very non-linear (similar to the use of $\log f$ in term wts of the vector space model) --- after 3 or 4 occurrences of a term, additional occurrences will have little impact.
- k_2 has a similar role for the query term weights. Typical values (see slide) make the equation less sensitive to k_2 than k_1 because query term frequencies are much lower and less variable than doc term frequencies.
- K is more complicated. Its role is basically to normalize the tf component by document length.
- b regulates the impact of length normalization. (0 means none; 1 is full normalization.)

BM25 Example

- Query with two terms, "president lincoln", ($qf = 1$)
- No relevance information (r and R are zero)
- $N = 500,000$ documents
- "president" occurs in 40,000 documents ($n_1 = 40,000$)
- "lincoln" occurs in 300 documents ($n_2 = 300$)
- "president" occurs 15 times in doc ($f_1 = 15$)
- "lincoln" occurs 25 times ($f_2 = 25$)
- document length is 90% of the average length ($dl/avdl = .9$)
- $k_1 = 1.2$, $b = 0.75$, and $k_2 = 100$
- $K = 1.2 \cdot (0.25 + 0.75 \cdot 0.9) = 1.11$

BM25 Example

$$\begin{aligned} BM25(Q, D) &= \\ &= \log \frac{(0 + 0.5)/(0 - 0 + 0.5)}{(40000 - 0 + 0.5)/(500000 - 40000 - 0 + 0 + 0.5)} \\ &\quad \times \frac{(1.2 + 1)15}{1.11 + 15} \times \frac{(100 + 1)1}{100 + 1} \\ &+ \log \frac{(0 + 0.5)/(0 - 0 + 0.5)}{(300 - 0 + 0.5)/(500000 - 300 - 0 + 0 + 0.5)} \\ &\quad \times \frac{(1.2 + 1)25}{1.11 + 25} \times \frac{(100 + 1)1}{100 + 1} \\ &= \log 460000.5/40000.5 \cdot 33/16.11 \cdot 101/101 \\ &\quad + \log 499700.5/300.5 \cdot 55/26.11 \cdot 101/101 \\ &= 2.44 \cdot 2.05 \cdot 1 + 7.42 \cdot 2.11 \cdot 1 \\ &= 5.00 + 15.66 = 20.66 \end{aligned}$$

BM25 Example

- Effect of term frequencies

Frequency of "president"	Frequency of "lincoln"	BM25 score
15	25	20.66
15	1	12.74
15	0	5.00
1	25	18.2
0	25	15.66