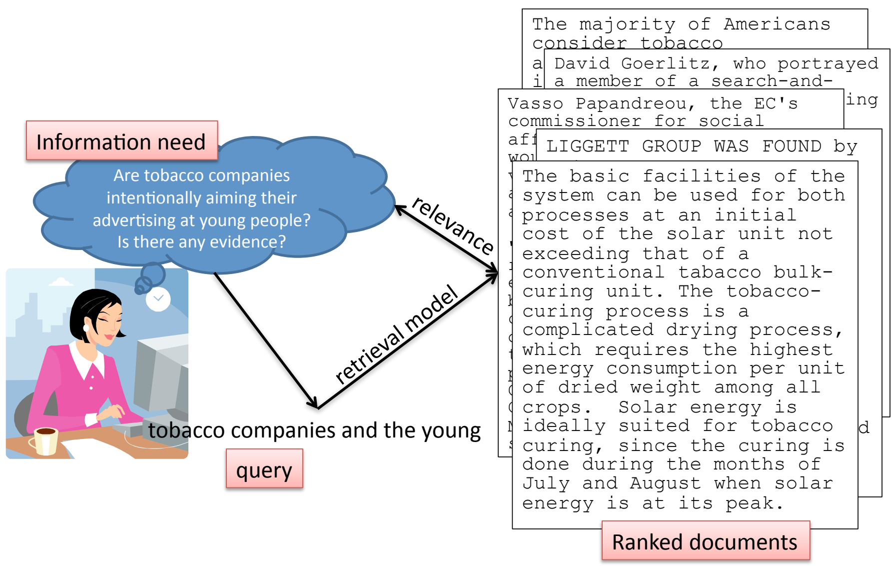
Queries and Information Needs

- A query can represent very different information needs
  - May require different search techniques and ranking algorithms to produce the best rankings
- A query can be a poor representation of the information need
  - User may find it difficult to express the information need
  - User is encouraged to enter short queries both by the search engine interface, and by the fact that long queries don't work

Retrieval Models

- Provide a mathematical framework for defining the search process
  - includes explanation of assumptions
  - basis of many ranking algorithms
  - can be implicit
- Theories about relevance

Retrieval Models



Vector Space Model

- Brief review:
  - Each term i has a weight w_{ik} in each document k.
  - These weights define a point in V-dimensional space.
  - Documents and queries are represented as vectors from the origin to its point.
  - Similarity between query and document is determined by the cosine angle between their vectors.

Vector Space Example

– Consider two documents D_1, D_2 and a query Q
  - D_1 = (0.5, 0.8, 0.3), D_2 = (0.9, 0.4, 0.2), Q = (1.5, 1.0, 0)

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

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

Term Weights

- Term weights w_{ik} are usually a function of tf and idf.
- There are many, many ways to define tf and idf and to combine them into a single weight.
- Very few of these have any mathematical motivation.
  - They are heuristics.
  - How can you predict which heuristic will work best for a task or domain or corpus?

Term Weighting Examples

- Term frequency: tf/len, tf/sqrt(len), log tf/len, log (tf/len + 1), (k+1)tf/(k+tf), …
- Inverse document frequency: N/n, (N+n)/n, log N/n, log (N/n + 1), …
- Combination: tf*idf, tf – idf, (tf+0.5)*(idf+1), …

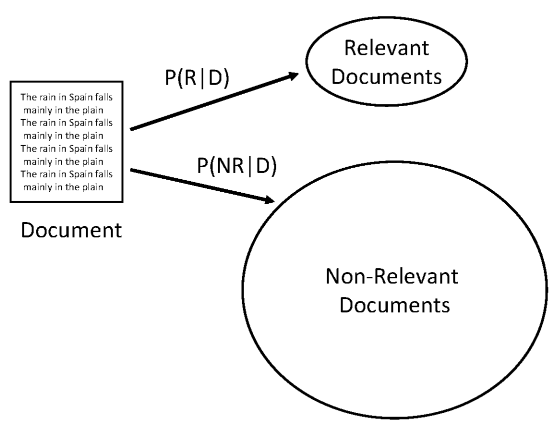
Probabilistic Models

- Use *statistics* of text to determine *probabilities* of relevance.
  - Mathematical framework founded in probability and statistics (and information theory).
  - Can potentially produce much less heuristic models.

Probability Ranking Principle

- Robertson (1977)
  - "If a reference retrieval system's response to each request is a **ranking of the documents in the collection in order of decreasing probability of relevance** to the user who submitted the request,
  - where the **probabilities are estimated as accurately as possible on the basis of whatever data** have been made available to the system for this purpose,
  - the overall **effectiveness of the system to its user will be the best that is obtainable** on the basis of those data."

IR as Classification



Probability of Relevance

- P(R | D), P(NR | D)
  - Probability that document D is relevant, probability that document D is not relevant.
  - D would be represented as a vector of features (term features, document features, etc.)
- Very simple example:
  - Suppose there are 1000 documents in the collection.
  - 100 are relevant to a query, 900 are not.
  - Can you estimate P(R | D) and P(NR | D)?

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

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

    - Left-hand side is *likelihood ratio*

Estimating P(D|R)

- Assume independence

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

- *Binary independence model*
  - document represented by a vector of binary features indicating term occurrence (or non-occurrence)
  - $p_i = P(d_i \mid R)$ is probability that term i occurs (i.e., has value 1) in relevant document
  - $s_i = P(d_i \mid NR)$ is probability that term i occurs in non-relevant document

Binary Independence Model

$$\frac{P(D|R)}{P(D|NR)} = \prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \prod_{i:d_i=0} \frac{1-p_i}{1-s_i}$$

$$= \boxed{\prod_{i:d_i=1} \frac{p_i}{s_i} \cdot \left(\prod_{i:d_i=1} \frac{1-s_i}{1-p_i}\right.} \cdot \boxed{\left.\prod_{i:d_i=1} \frac{1-p_i}{1-s_i}\right) \cdot \prod_{i:d_i=0} \frac{1-p_i}{1-s_i}}$$

$$= \boxed{\prod_{i:d_i=1} \frac{p_i(1-s_i)}{s_i(1-p_i)}} \cdot \boxed{\cancel{\prod_{i} \frac{1-p_i}{1-s_i}}}$$

Binary Independence Model

- Classify a document as relevant if

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

Not necessary for ranking

- Scoring function is

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

- How can we estimate p_i and s_i?
  - Recall $p_i = P(d_i \mid R)$, $s_i = P(d_i \mid NR)$
  - If we randomly pick a document out of the relevant class R, what is the probability that it contains d_i?

Contingency Table

For term i:

	Relevant	Non-relevant	Total
$d_i = 1$	$\boxed{r_i}$	$\boxed{n_i} - r_i$	n_i
$d_i = 0$	$R - r_i$	$N - n_i - R + r_i$	$N - r_i$
Total	$\boxed{R}$	$\boxed{N} - R$	N

Number of relevant documents that contain term i

Number of relevant documents

Number of documents

Number of documents that contain term i

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

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

Gives scoring function:

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

Binary Independence Model

- Scoring function is

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

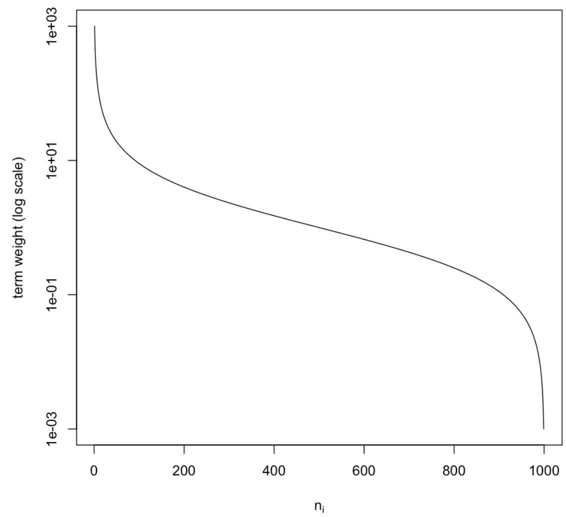
- Query provides information about relevant documents
- If we assume r_i is zero, n_i is all of the documents d_i occurs in, get *idf*-like weight

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

BIM Summary

- Documents are relevant if P(D | R)/P(D | NR) > P(NR)/P(R).
- The probability of observing document D in the relevant class R is modeled as the product of the probabilities of observing (or not observing) each term *i* in documents in the relevant class.
  - Similarly for P(D | NR).
- The probability of observing term i in a document in R is estimated as 0.5.
- The probability of observing term i in a document in NR is estimated as n_i/N.
- Documents are scored as $\sum_{i:d_i=q_i=1} \log \frac{N-n_i}{n_i}$

BIM Term Weighting



2-Poisson Model

- Generalize binary occurrence model to term frequency model.

$$\frac{P(D \mid R)}{P(D \mid NR)} = \prod_i \frac{P(F_i = f_i \mid R)}{P(F_i = f_i \mid NR)} \frac{P(F_i = 0 \mid NR)}{P(F_i = 0 \mid R)}$$

- Partition documents into those “elite” for term and those “not elite” for term.
  - $P(F_i \mid R) = P(F_i \mid E)P(E \mid R) + P(F_i \mid NE)P(NE \mid R)$
  - $P(F_i \mid NR) = P(F_i \mid E)P(E \mid NR) + P(F_i \mid NE)P(NE \mid NR)$
- $P(F_i \mid E)$, $P(F_i \mid NE)$ have Poisson distributions.

2-Poisson Model

- Model components:
  - $P(F_i = f_i \mid E) = \lambda^{f_i} e^{-\lambda} / f_i!$
  - $P(F_i = f_i \mid NE) = \mu^{f_i} e^{-\mu} / f_i!$
  - $P(E \mid R) = p'$
  - $P(E \mid NR) = q'$
- Many parameters to estimate.
  - λ, μ, p', q' for every term.

$$w = \log \frac{(p'\lambda^{tf} e^{-\lambda} + (1-p')\mu^{tf} e^{-\mu})\ (q' e^{-\lambda} + (1-q') e^{-\mu})}{(q'\lambda^{tf} e^{-\lambda} + (1-q')\mu^{tf} e^{-\mu})\ (p' e^{-\lambda} + (1-p') e^{-\mu})},$$

Approximating the 2-Poisson Model

- Start with Binary Independence Model weight:

$$w_i = \log \frac{N - n_i}{n_i}$$

- Modify with a document term frequency component and a query term frequency component.
  - Determine "shape" of these components using some constraints.

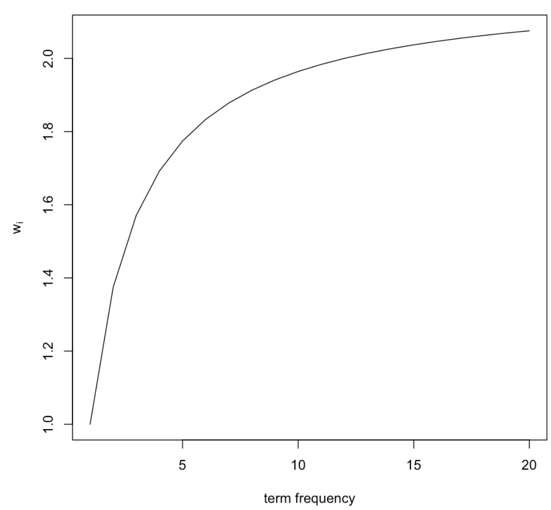
Ad Hoc Model of Term Frequency

- Full Poisson model has properties:
  - w = 0 if term frequency is 0.
  - w increases monotonically with tf.
  - w asymptotically approaches a maximum.
- So how about:

$$w'_i = \frac{(k_1 + 1)tf_i}{k_1 + tf_i} w_i$$

- k_1 is a term frequency parameter determined by developer.

Term Frequency Weighting



Ad Hoc Model of Document Length

- 2-Poisson model implicitly assumes all documents have the same length.
  - They do not.
- Two hypotheses about why:
  - "Scope hypothesis": long documents are like several short documents concatenated.
  - "Verbosity hypothesis": long documents are just longer versions of short documents.
- Verbosity more tractable.

Document Length Normalization

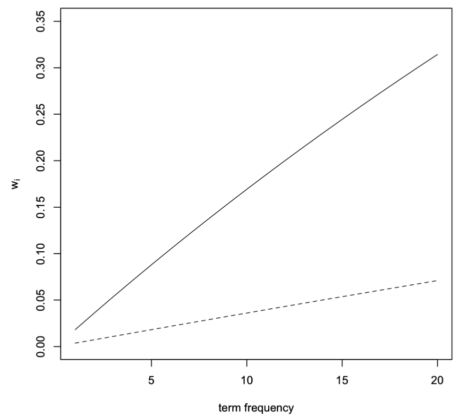
- Normalize tf by length normalization factor *NF*.

$$w'_i = \frac{(k_1+1)\frac{tf}{NF}}{k_1 + \frac{tf}{NF}} w_i = \frac{(k_1+1)tf}{k_1 NF + tf} w_i$$

- What should NF be?
  - NF = document length = *dl*
  - NF = scaled document length = *dl/avgdl*
  - NF = mixed length = (1 – b) + b**dl/avgdl*

Document Length Normalizations

NF = doc length

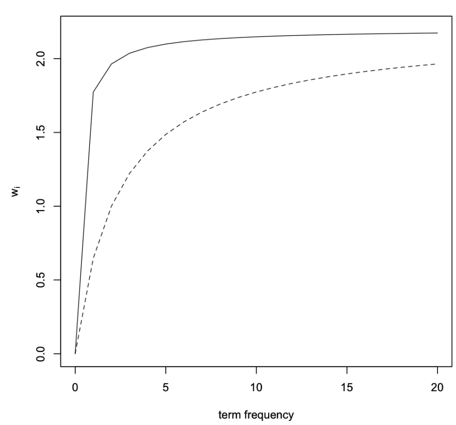


len = 100

len = 1000

Document Length Normalizations

NF = doc length/average doc length



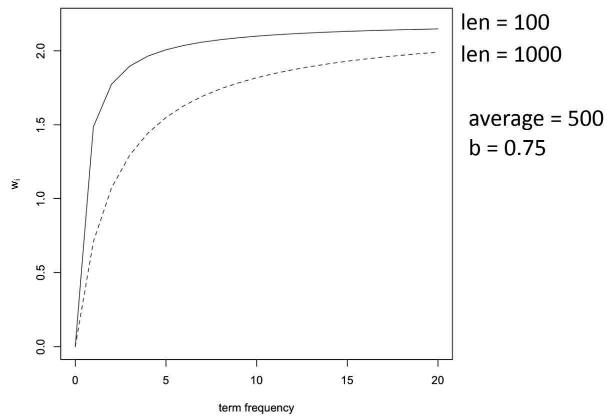
len = 100

len = 1000

average = 500

Document Length Normalizations

NF = (1-b) + b*doc length/average doc length



b is a length normalization parameter determined by developer.

Query Term Frequency

- Treat query term frequency like document term frequency.

$$w'_i = \frac{(k_3 + 1)qtf}{k_3 + qtf} w_i$$

- k_3 is a query term frequency parameter determined by developer.

BMn: Putting it all Together

- Combine BIM weight with term frequency weight (normalized by document length) and query term frequency weight.
- BM1: $\sum_{i \in Q} \frac{(k_3+1)qtf_i}{k_3+qtf_i} \log \frac{N-n_i}{n_i}$
- BM11: $\sum_{i \in Q} \frac{(k_1+1)tf_i}{k_1 \frac{dl}{avgdl} + tf_i} \frac{(k_3+1)qtf_i}{k_3+qtf_i} \log \frac{N-n_i}{n_i}$
- BM25: $\sum_{i \in Q} \frac{(k_1+1)tf_i}{k_1\left(1-b+b\frac{dl}{avgdl}\right) + tf_i} \frac{(k_3+1)qtf_i}{k_3+qtf_i} \log \frac{N-n_i}{n_i}$

BM25

- BM25 is a popular and effective approximation

$$\sum_{i \in Q} \frac{(k_1+1)tf_i}{k_1\left(1-b+b\frac{dl}{avgdl}\right) + tf_i} \frac{(k_3+1)qtf_i}{k_3+qtf_i} \log \frac{N-n_i}{n_i}$$

- tf, document length, and idf components
- Three parameters:
  - k_1, k_3, b
  - Determined empirically
- Good values: $k_1 = 1.2$, $k_3 = 0$, $b = 0.75$

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 · (0.25 + 0.75 · 0.9) = 1.11

BM25 Example

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

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