

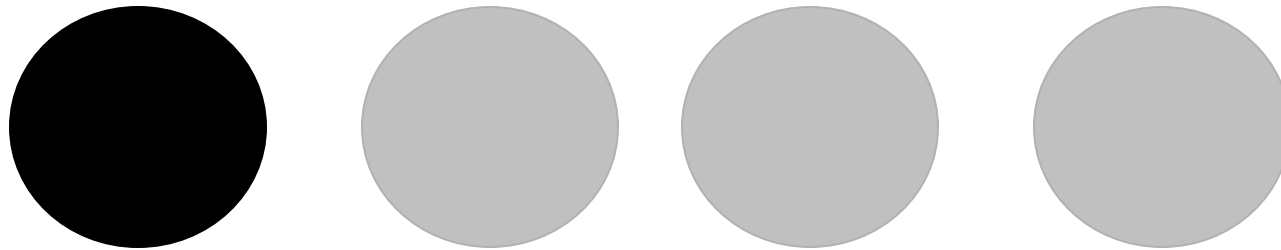
# Probabilistic Retrieval

# Probabilistic Model

- Use probability to estimate the “odds” of relevance of a query to a document.
- Need to know in advance which documents are relevant to query to compute an estimate of relevance.

# Some Background

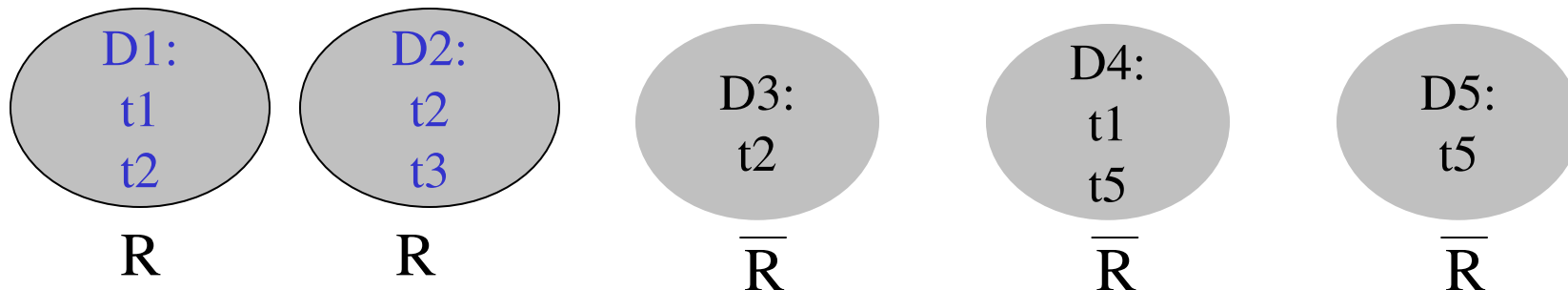
- If we have four balls, three gray and one black, and *it is equally likely that we could pick any of the balls*, we can estimate the probability that of:



- Choosing a black ball =  $1/4$
- Choosing (with replacement) two black balls in a row  $(1/4)(1/4) = (1/16)$

# Relevance Odds for One Term

- Now let's switch to documents. Let's say we want to estimate, for a given term, the odds it will be in a relevant document.



- Now we assume documents D1 and D2 are relevant, and D3 and D4 are not relevant. Need to estimate the odds that a document D is relevant given the query term  $t1$
- Odds that R is relevant given t1:*

*num relevant with t1 / num relevant*

$$O(R | t1) = \frac{\text{num relevant with } t1}{\text{num relevant}}$$

*num of docs with t1 / all documents*

$$O(R | t1) = (1 / 2) / (2 / 5) = .5 / .4 = 1.25 : 1$$

# Computing Odds of Relevance for Multiple Terms

- Now we are given query terms  $t_1, t_2, \dots, t_n$  so we want to compute the odds of relevance given these terms:

$$O(R \mid t_1, t_2, \dots, t_n)$$

- By repeated application of Bayes theorem we can take the product of these individual odds.

$$O(R \mid t_i) = \prod_{i=1}^{i=t} O(R \mid t_i)$$

- Since the log function is often used to scale the odds, the sum of the log odds (log of each odds) may be used:

$$\log\left(\prod_{i=1}^{i=t} O(R \mid t_i)\right) = \sum_{i=1}^{i=t} \log(O(R \mid t_i))$$

# Principles surrounding weights

(Robertson and Sparck Jones, 1976)

- Independence Assumptions
  - I1: The distribution of terms in relevant documents is independent and their distribution in all documents is independent.
  - I2: The distribution of terms in relevant documents is independent and their distribution in non-relevant documents is independent.
- Ordering Principles
  - O1: Probable relevance is based only on the presence of search terms in the documents.
  - O2: Probable relevance is based on both the presence of search terms in documents and their absence from documents.

# Parameters in Computing Term Weight

$N$  = total number of documents in collection

$R$  = total number of relevant documents for a query

$n$  = number of documents that contain the query term

$r$  = number of relevant documents that contain the query term

# Probabilistic Variations to Compute Term Weight

- I1 and O1:  $(r/R) / (n/N)$
- I2 and O1:  $(r/R) / ((n-r)/(N-R))$
- I1 and O2:  $(r/(R-r)) / (n / (N-n))$
- I2 and O2:  $(r/(R-r)) / ((n-r)/((N-n)-(R-r)))$
- Adding in a fudge factor of 0.5 for no good reason except that it helps:
- $((r+.5)/(R-r+.5)) / ((n-r+.5) / ((N-n)-(R-r))+.5)$

# Probabilistic Retrieval Example

- D1: “Cost of paper is up.” (*relevant*)
- D2: “Cost of jellybeans is up.” (*not relevant*)
- D3: “Salaries of CEOs are up.” (*not relevant*)
- D4: “Paper: CEO’s labor cost up.” (????)

<b>Q. Term</b>	<b>Relevant</b>	<b>Not relevant</b>	<b>Evidence</b>
paper	1	0	for (strong)
CEO	0	1/2	against
labor	0	0	none
cost	1	1/2	for (weak)
up	1	1	none

# Probabilistic Retrieval Example

## (Cont'd)

- *cost* appears in 1 of 1 relevant document
  - odds are  $(1+.5)/(0+.5) = 3$  to 1 that *cost* will appear in a relevant document
- *cost* appears in 1 of 2 non-relevant documents
  - odds are  $(1+.5)/(1+.5) = 1$  to 1 that *cost* will appear in an irrelevant document
- So if *cost* appears in D, then the odds are  $(3/1)/(1/1) = 3$  to 1 that D is relevant.

# Probabilistic Retrieval Example

## (Cont'd)

- D1: “Cost of paper is up.” (*relevant*)
- D2: “Cost of jellybeans is up.” (*not relevant*)
- D3: “Salaries of CEO’s are up.” (*not relevant*)
- D4: “Paper: CEO’s labor cost up.” (????)

<b>Term</b>	<b>Odds of Relevance</b>	
paper	$(1.5/0.5)/(0.5/2.5)$	= 3/2
CEO	$(0.5/1.5)/(1.5/1.5)$	= 1/3
labor	$(0.5/1.5)/(0.5/2.5)$	= 5/3
cost	$(1.5/0.5)/(1.5/1.5)$	= 3
up	$(1.5/0.5)/(2.5/0.5)$	= 3/5
<b>TOTAL ODDS</b>	(product of the individual odds)	= <b>1.5</b>

# Modifications to Basic Probabilistic Model

- Term frequency and document length were not considered in original probabilistic model.
- Performed worse than vector space model (VSM).

Thus:

- Modification to Probabilistic model:
  - Incorporating tf-idf (Croft and Harper, 1979)
  - Incorporating document length (Robertson and Walker 1995)

# Modifications to Basic Probabilistic Model [Okapi BM25; TREC-3]

$$SC(Q, D_i) = \sum_{j=1}^t w \left( \frac{(k_1 + 1)tf_{ij}}{K + tf_{ij}} \right) \left( \frac{(k_3 + 1)qtf_j}{k_3 + qtf_j} \right) + \left( k_2 |Q| \frac{avdl - dl}{avdl + dl_i} \right)$$

$$w = \log \left( \frac{\frac{r + 0.5}{R - r + 0.5}}{\frac{n - r + 0.5}{N - n - R - r + 0.5}} \right) \leftarrow \textit{Robertson/Spark Jones weight}$$

# Modifications to Basic Probabilistic Model (Cont'd)

$N$  = number of documents

$n$  = number of documents having the term

$R$  = total number of relevant documents for a query

$r$  = number of relevant documents that contain the query term

$tf$  = term frequency of term in document

$qtf$  = term frequency of query term

$dl$  = document length (arbitrary units)

$|Q|$  = number of terms in query

$avdl$  = average document length

$k_1, k_2, k_3, b$  = tuning parameters; depends on the collection and queries

$K = k_1(1-b) + b(dl_i/avdl)$

$b$  = is tuning parameter based on the nature of document collection

# Variation of Okapi BM25

[Robertson & Walker 1997; TREC-7]

$$w = \frac{k_5}{k_5 + \sqrt{R}} \left( k_4 + \log \frac{N}{N-n} \right) + \frac{\sqrt{R}}{k_5 + \sqrt{R}} \log \frac{r+0.5}{R-r+0.5} - \frac{k_6}{k_6 + \sqrt{S}} \log \frac{n}{N-n} - \frac{\sqrt{S}}{k_6 + \sqrt{S}} \log \frac{s+0.5}{S-s+0.5}$$

S = number of document known to be non-relevant to a query

s = number of non-relevant documents having the term

$k_4, k_5, k_6$  = tuning parameters

( $k_5, k_6$  indicate the weight given to relevance and non-relevance information, respectively. In TREC-7 experiments:  $k_5$  is 0-4; and  $k_6$  is 4- $\infty$ )

# *a priori* Relevance Information

- *a priori* Relevance Information not always known
- In on-line systems not possible to have relevant information as training data ( $r$ ,  $R$ )
- Alternative:
  - Relying on user's feedback
  - Without any relevance information

# *Without* Relevance Information

- If no relevance information exist, then  $R=S=0$ . Thus, the weight:

$$w = \frac{k_5}{k_5 + \sqrt{R}} \left( k_4 + \log \frac{N}{N-n} \right) + \frac{\sqrt{R}}{k_5 + \sqrt{R}} \log \frac{r+0.5}{R-r+0.5} - \frac{k_6}{k_6 + \sqrt{S}} \log \frac{n}{N-n} - \frac{\sqrt{S}}{k_6 + \sqrt{S}} \log \frac{s+0.5}{S-s+0.5}$$

- Will be:

$$w = k_4 + \log \frac{N}{N-n}$$

# Summary of Basic Probabilistic Model

- Pros
  - Some theoretical basis
- Cons
  - For many applications *a priori* relevance is not known