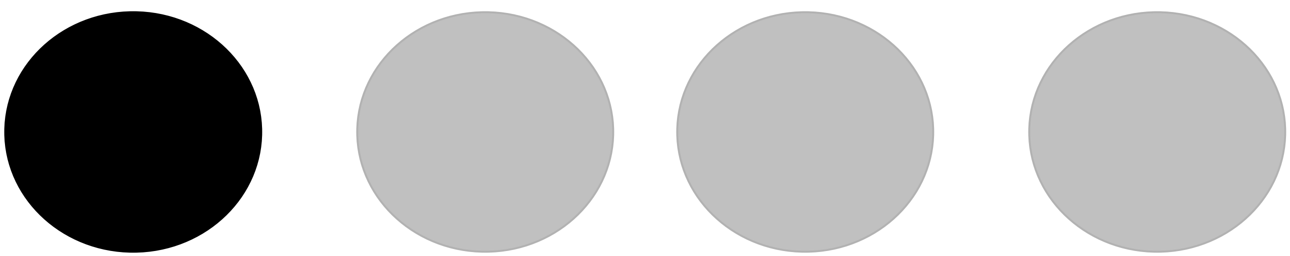
# Probabilistic Retrieval

# 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

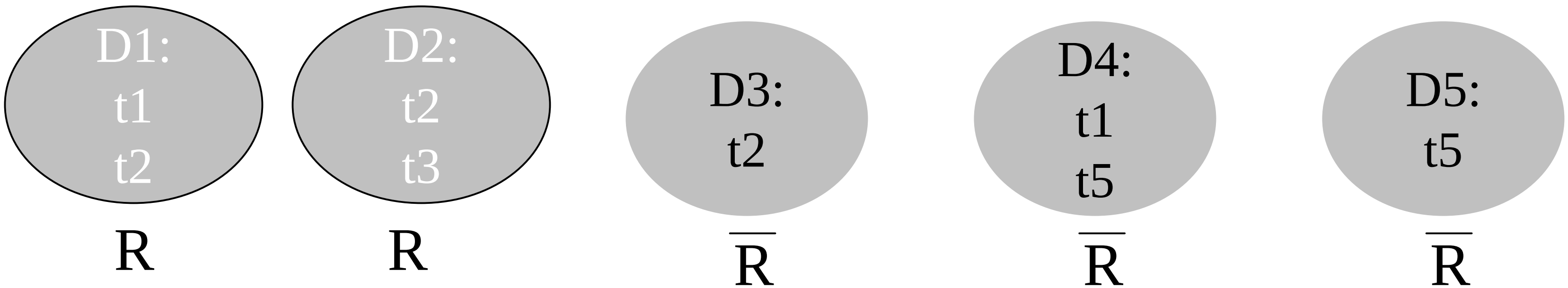
- If we have four balls, three red and one black, and *it is equally likely that we could pick any of the balls*, we can estimate the probability that of:



- Choosing a black ball = 1/4
- Choosing two black balls in a row (1/4)(1/4) = (1/8)

# Relevance Odds for One Term

- Now lets switch to documents.  Lets 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 non-relevant. Need to compute the estimate that a document D is relevant given the query term *t1*
- *Odds that R is relevant given t1:*

$$O(R \mid t1) = \frac{\textit{num relevant with t1 / num relevant}}{\textit{num of docs with t1 / all documents}}$$

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

# Computing Odds of Relevance for Multiple Terms

- Now we are given query terms $t_1, t_2, ..., t_n$ so we want to compute the odds of relevance given these terms:
- $O(R \mid t_1, t_2, ..., t_n)$
  - By repeated application of Bayes theorem we can take the product of these individual odds.
- $O(R \mid t_1) \times O(R \mid t_2) \times ... O(R \mid t_n)$
  - Note, 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 some fluff 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 CEO’s are up.” (*not relevant*)
- D4: “Paper: CEO’s labor cost up.” (????)

| **Q. Term** | **Relevant** | **Not relevant** | **Evidence** |
|---|---|---|---|
| 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
- *cost* appears in 1 of 2 non-relevant documents
  - odds are (1+.5)/(1+.5) = 1 to 1 that *cost* will appear
- 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." (????)

| **Term** | **Odds of Relevance** | |
|---|---|---|
| paper | (1.5/0.5)/(0.5/2.5) | = 15 |
| 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 |
| **TOTAL ODDS** (product of the individual odds) | | = **15** |

# Modifications to Basic Probabilistic Model

- Term frequency and document length are 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)

# Modifications to Basic Probabilistic Model

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 = number of terms in document (document length)

|Q| = number of terms in query

$\Delta$= average document length

$K_1$, $K_2$ , $K_3$ = tuning parameters

$$SC(Q, D_i) = \sum_{j=1}^{t} \log \left( \frac{\frac{r}{R-r}}{\frac{n-r}{(N-n)-(R-r)}} \right) \left( \frac{(k_1+1)tf_{ij}}{K+tf_{ij}} \right) \left( \frac{(k_3+1)qtf_j}{k_3+qt_{ij}} \right) + \left( k_2 \mid Q \mid \frac{\Delta - dl}{\Delta + dl_i} \right)$$

# Equivalence to Vector Space Model

- Now, if
  - Relevant set = {query}, and
  - Non-relevant set = {}
- Then probabilistic retrieval reduces to vector space retrieval.

# Summary of Basic Probabilistic Model

- Pros
  - Some theoretical basis
  - Sort of derives the *idf*
- Cons
  - no intuitive support for term frequency
  - lots of assumptions