

# Probabilistic Ranking Principle

Hongning Wang

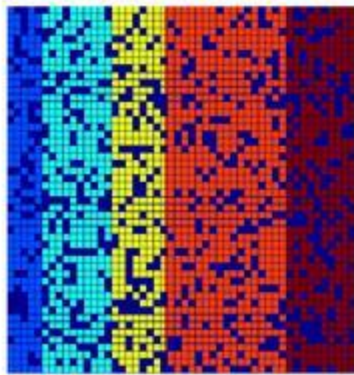
CS@UVa

# Recap: latent semantic analysis

- Solution

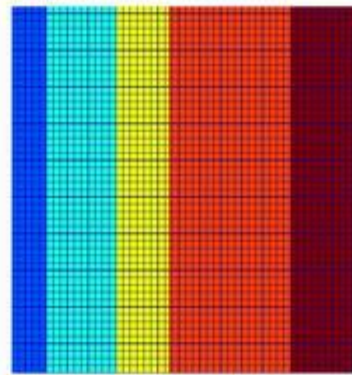
- Low rank matrix approximation

*Imagine this is \*true\* concept-document matrix*



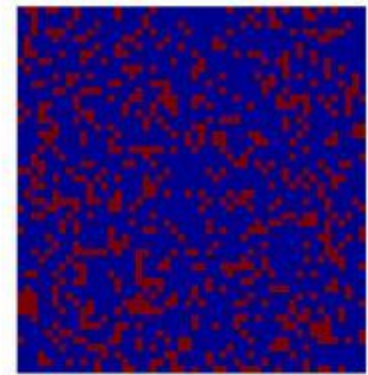
*Matrix of corrupted observations*

*Imagine this is our observed term-document matrix*



*Underlying low-rank matrix*

+

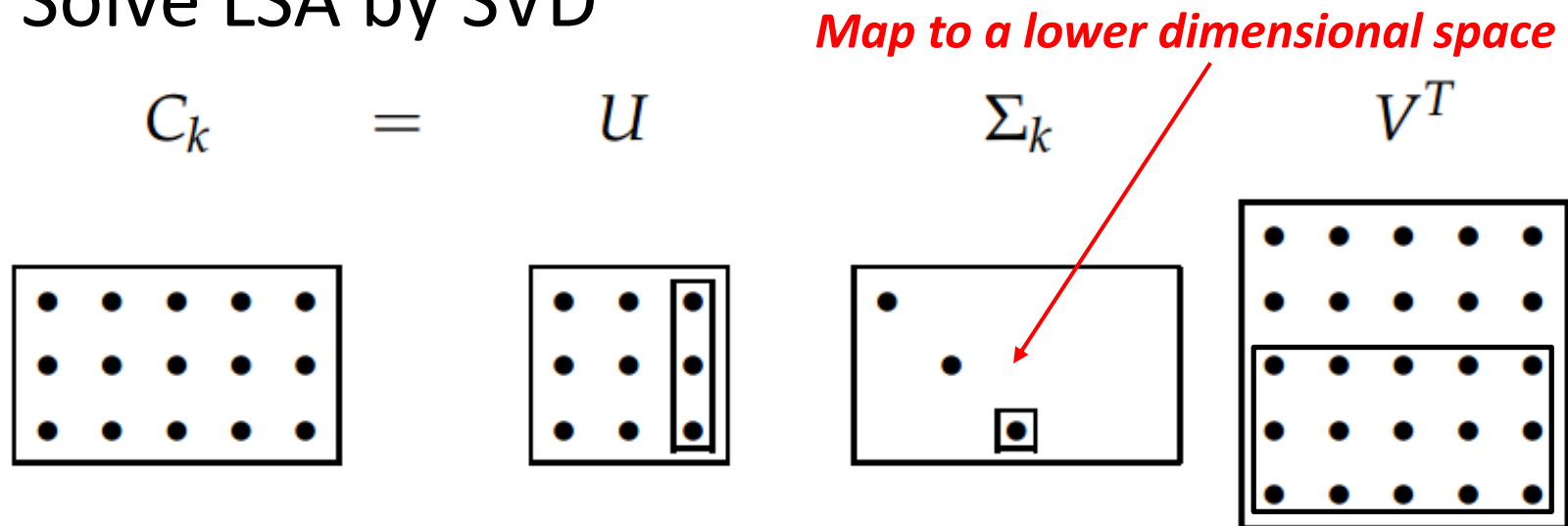


*Sparse error matrix*

*Random noise over the word selection in each document*

# Recap: latent semantic analysis

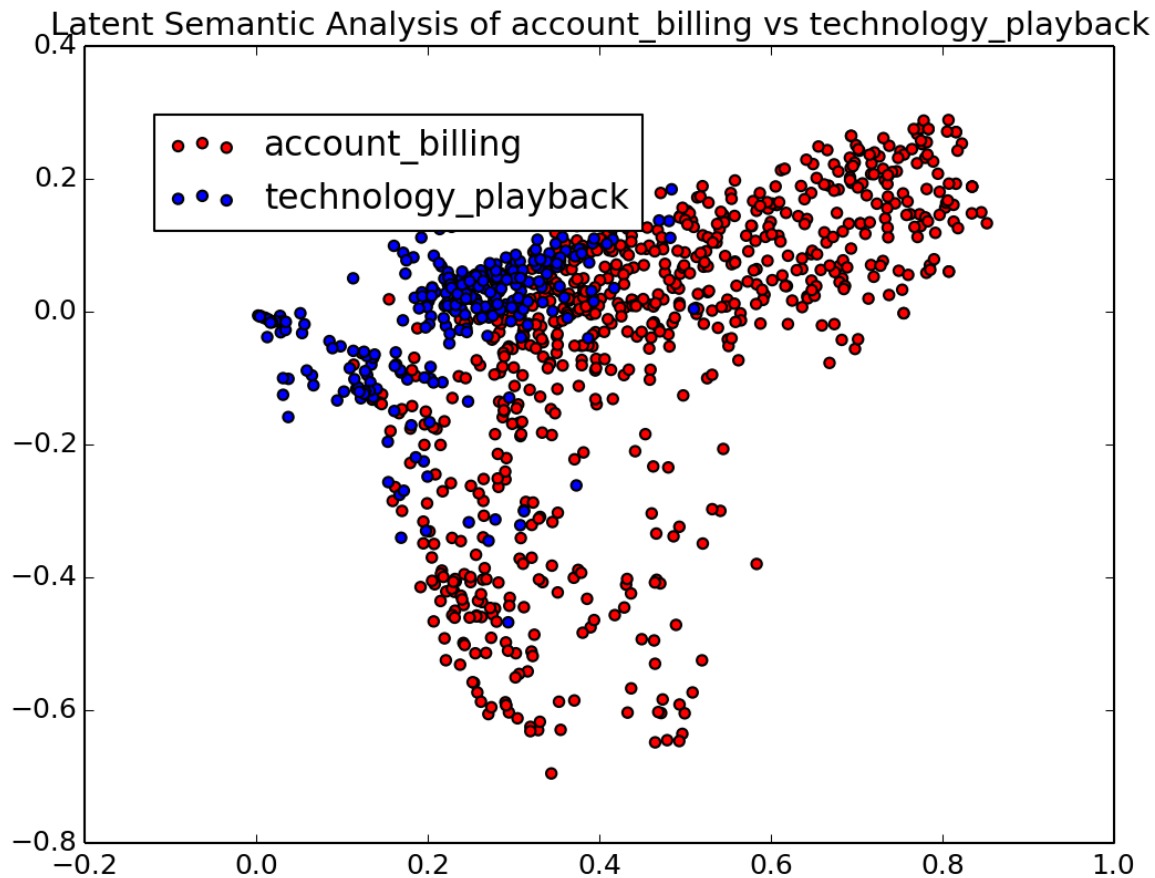
- Solve LSA by SVD



1. Perform SVD on document-term adjacency matrix
2. Construct  $C_{M \times N}^k$  by only keeping the largest  $k$  singular values in  $\Sigma$  non-zero

# Recap: latent semantic analysis

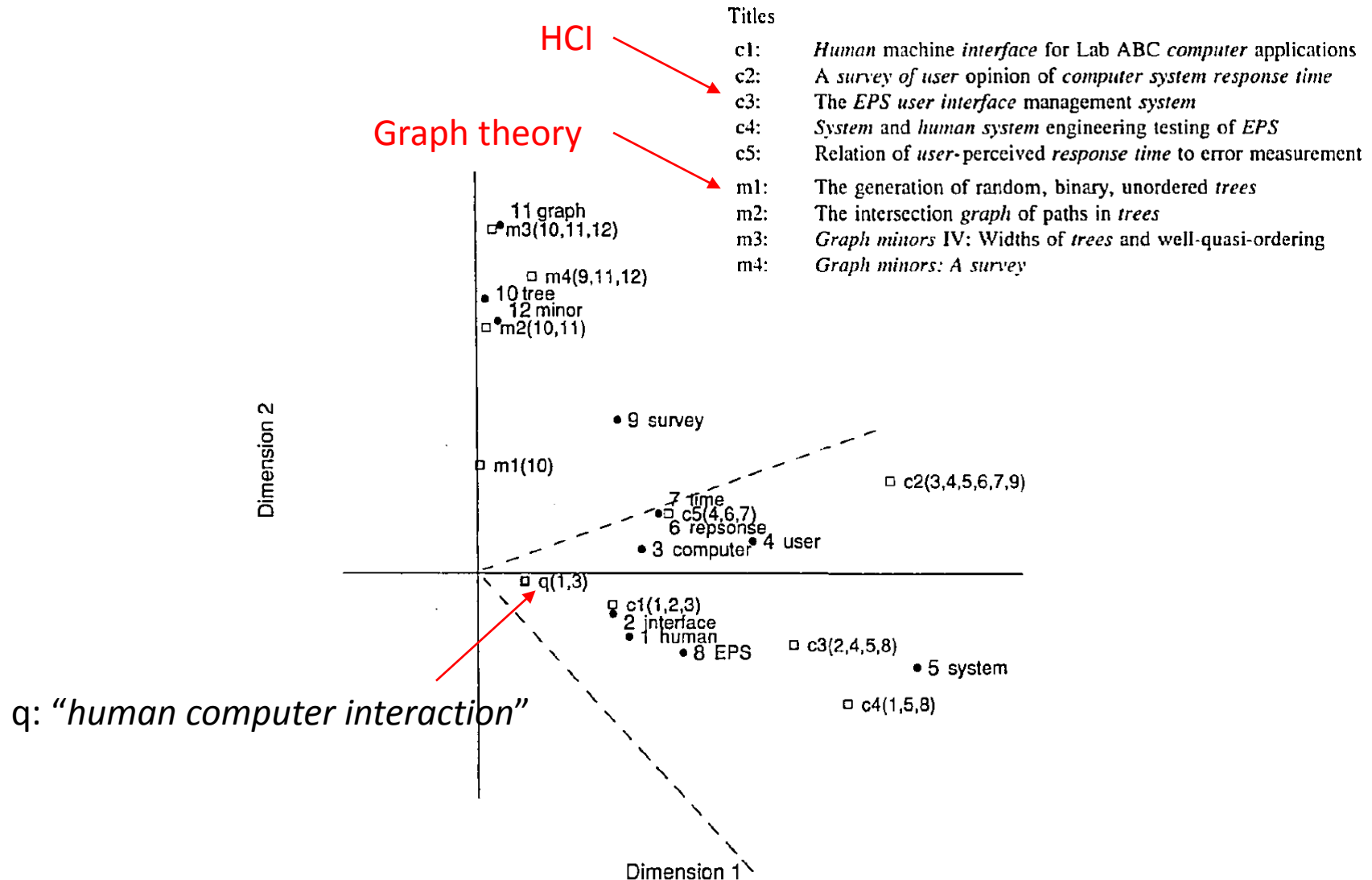
- Visualization



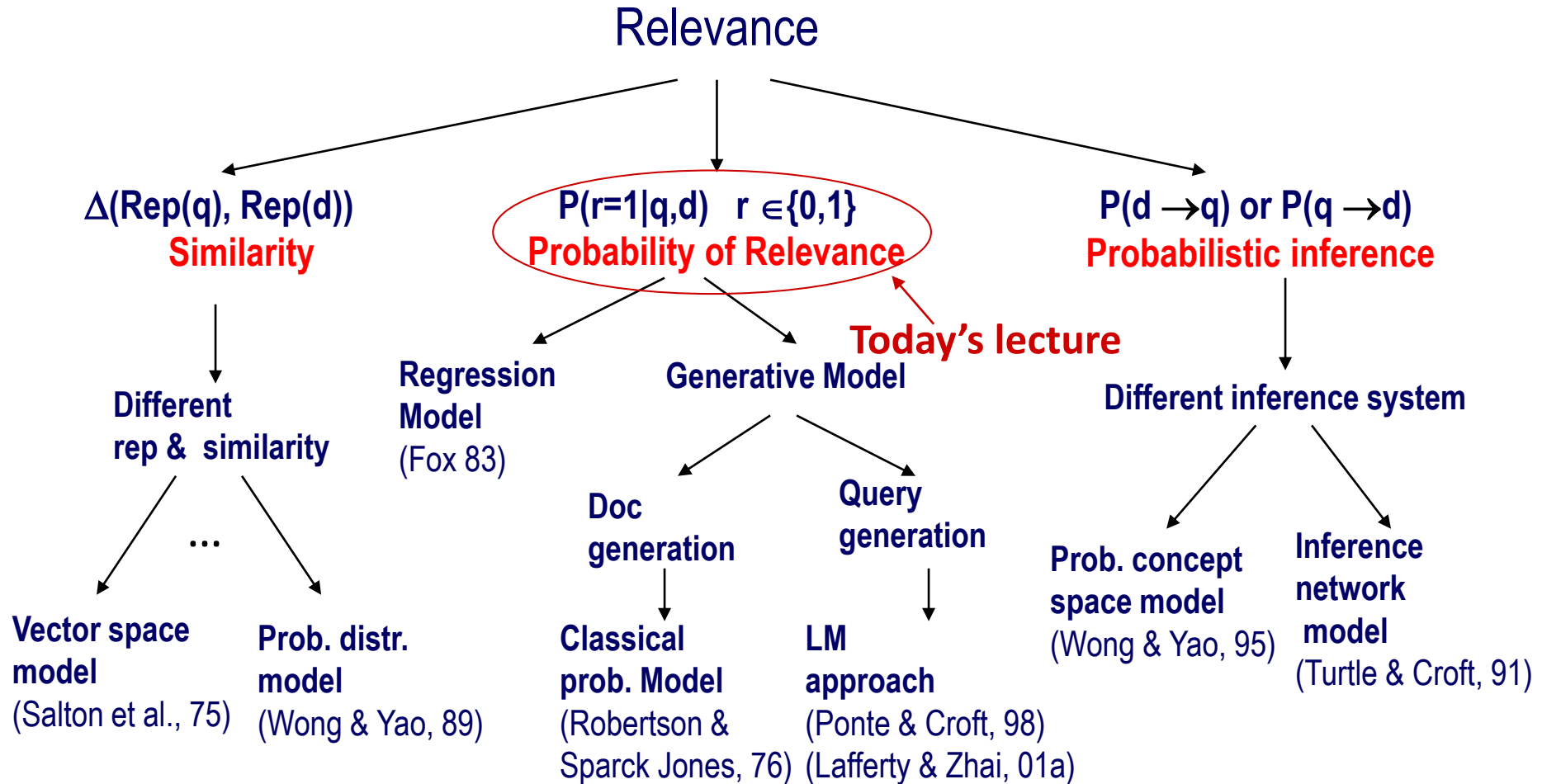
# Recap: LSA for retrieval

- Project queries into the new document space
  - $\tilde{q} = qV_{N \times k}\Sigma_{k \times k}^{-1}$ 
    - Treat query as a pseudo document of term vector
    - Cosine similarity between query and documents in this lower-dimensional space

# Recap: LSA for retrieval



# Notion of relevance



# Basic concepts in probability

- Random experiment
  - An experiment with uncertain outcome (e.g., tossing a coin, picking a word from text)
- Sample space ( $S$ )
  - All possible outcomes of an experiment, e.g., tossing 2 fair coins,  $S=\{HH, HT, TH, TT\}$
- Event ( $E$ )
  - $E \subseteq S$ ,  $E$  happens iff outcome is in  $S$ , e.g.,  $E=\{HH\}$  (all heads),  $E=\{HH, TT\}$  (same face)
  - Impossible event ( $\{\}$ ), certain event ( $S$ )
- Probability of event
  - $0 \leq P(E) \leq 1$



# Essential probability concepts

- Probability of events
  - Mutually exclusive events
    - $P(A \cup B) = P(A) + P(B)$
  - General events
    - $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
  - Independent events
    - $P(A \cap B) = P(A)P(B)$

 Joint probability, or  
simply as  $P(A, B)$

# Essential probability concepts

- Conditional probability
  - $P(B|A) = P(A, B)/P(A)$
  - Bayes' Rule:  $P(B|A) = P(A|B)P(B)/P(A)$
  - For independent events,  $P(B|A) = P(B)$
- Total probability
  - If  $A_1, \dots, A_n$  form a non-overlapping partition of S
    - $P(B \cap S) = P(B \cap A_1) + \dots + P(B \cap A_n)$
    - $P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B|A_1)P(A_1) + \dots + P(B|A_n)P(A_n)} \propto P(B|A_i)P(A_i)$
    - This allows us to compute  $P(A_i|B)$  based on  $P(B|A_i)$

# Interpretation of Bayes' rule

Hypothesis space:  $H = \{H_1, \dots, H_n\}$ , Evidence:  $E$

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{P(E)}$$

If we want to pick the most likely hypothesis  $H^*$ , we can drop  $P(E)$

Posterior probability of  $H_i$



Prior probability of  $H_i$



$$P(H_i | E) \propto P(E | H_i)P(H_i)$$



Likelihood of data/evidence given  $H_i$

# Theoretical justification of ranking

- As stated by William Cooper

*“If a reference retrieval system’s response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data.”*

- Rank by probability of relevance leads to optimal retrieval performance

# Justification

- From decision theory
    - Two types of loss
      - $\text{Loss}(\text{retrieved} | \text{non-relevant}) = a_1$
      - $\text{Loss}(\text{not retrieved} | \text{relevant}) = a_2$
    - $\phi(d_i, q)$ : probability of  $d_i$  being relevant to  $q$
    - Expected loss regarding to the decision of including  $d_i$  in the final results
      - Retrieve:  $(1 - \phi(d_i, q))a_1$
      - Not retrieve:  $\phi(d_i, q)a_2$
- Your decision criterion?*

# Justification

- From decision theory
  - We make decision by
    - If  $(1 - \phi(d_i, q))a_1 < \phi(d_i, q)a_2$ , retrieve  $d_i$
    - Otherwise, not retrieve  $d_i$
  - Check if  $\phi(d_i, q) > \frac{a_1}{a_1 + a_2}$
  - Rank documents by descending order of  $\phi(d_i, q)$  would minimize the loss

# According to PRP, what we need is

- A relevance measure function  $F(q,d)$ 
  - For all  $q, d_1, d_2$ ,  
 $F(q,d_1) > F(q,d_2)$  iff.  $p(\text{Rel} | q,d_1) > p(\text{Rel} | q,d_2)$
  - Assumptions
    - Independent relevance
    - Sequential browsing

Most existing research on IR models so far has fallen into this line of thinking... (Limitations?)

# Probability of relevance

- Three random variables
  - Query  $Q$
  - Document  $D$
  - Relevance  $R \in \{0,1\}$
- Goal: rank  $D$  based on  $P(R=1 | Q, D)$ 
  - Compute  $P(R=1 | Q, D)$
  - Actually, one only needs to compare  $P(R=1 | Q, D_1)$  with  $P(R=1 | Q, D_2)$ , i.e., rank documents
- Several different ways to define  $P(R=1 | Q, D)$



# Conditional models for $P(R=1 | Q,D)$

- Basic idea: relevance depends on how well a query matches a document
  - $P(R=1 | Q,D)=g(\text{Rep}(Q,D) | \theta)$  ← a functional form
    - $\text{Rep}(Q,D)$ : feature representation of query-doc pair
      - E.g., #matched terms, highest IDF of a matched term, docLen
  - Using training data (with known relevance judgments) to estimate parameter  $\theta$
  - Apply the model to rank new documents
- Special case: logistic regression

# Regression for ranking?

- Linear regression

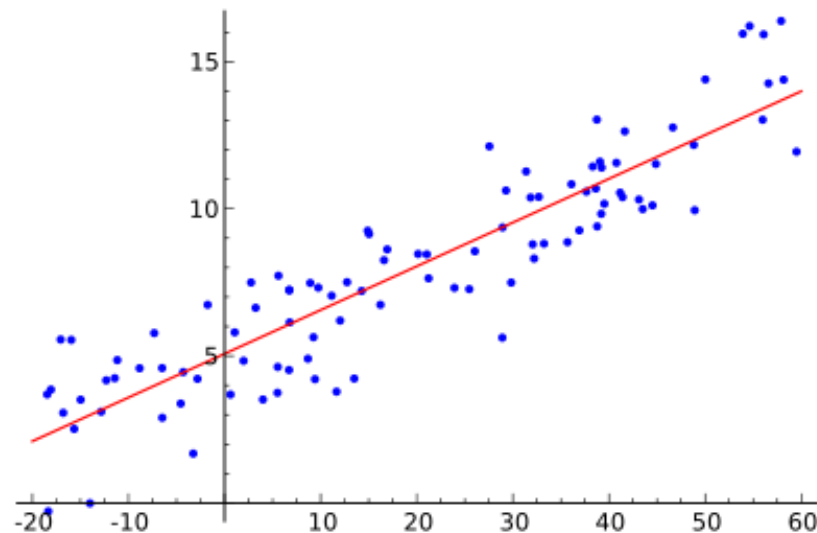
- $y \leftarrow w^T X$

- Relationship between a scalar dependent variable  $y$  and one or more explanatory variables

In a ranking problem:

$X$  features about query-document pair

$y$  relevance label of document for the given query



# Features/Attributes for ranking

- Typical features considered in ranking problems

$$X_1 = \frac{1}{M} \sum_1^M \log QAF_{t_j} \quad \text{Average Absolute Query Frequency}$$

$$X_2 = \sqrt{QL} \quad \text{Query Length}$$

$$X_3 = \frac{1}{M} \sum_1^M \log DAF_{t_j} \quad \text{Average Absolute Document Frequency}$$

$$X_4 = \sqrt{DL} \quad \text{Document Length}$$

$$X_5 = \frac{1}{M} \sum_1^M \log \frac{N - n_{t_j}}{n_{t_j}} \quad \text{Average Inverse Document Frequency}$$

$$X_6 = \log M \quad \text{Number of Terms in common between query and document}$$

# Regression for ranking

- Linear regression

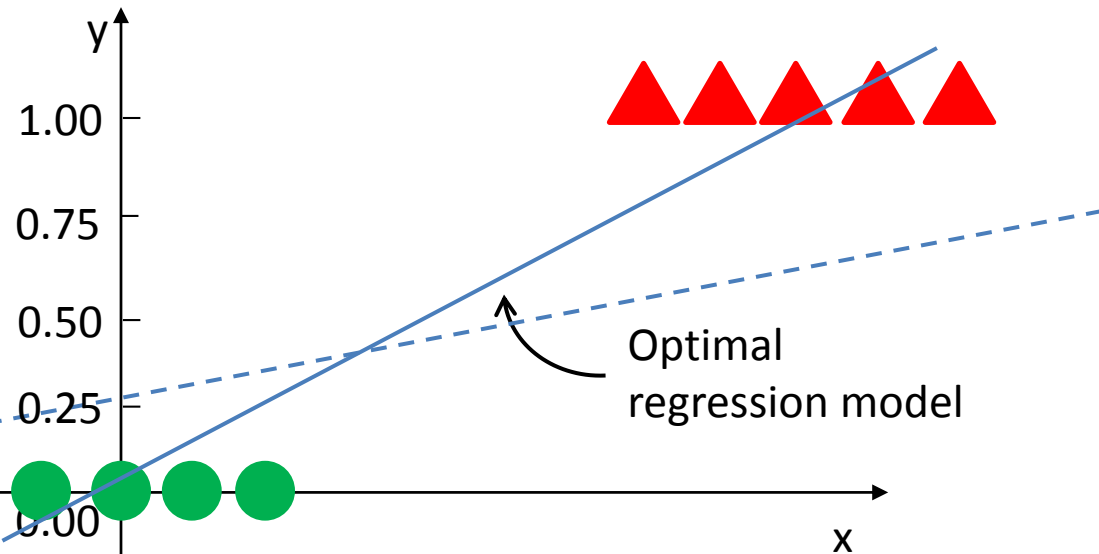
- $y \leftarrow w^T X$

- Relationship between a scalar dependent variable  $y$  and one or more explanatory variables

*Y is discrete in a ranking problem!*

$$y = \begin{cases} 1 & w^T X > 0.5 \\ 0 & w^T X \leq 0.5 \end{cases}$$

What if we have an outlier?



# Regression for ranking

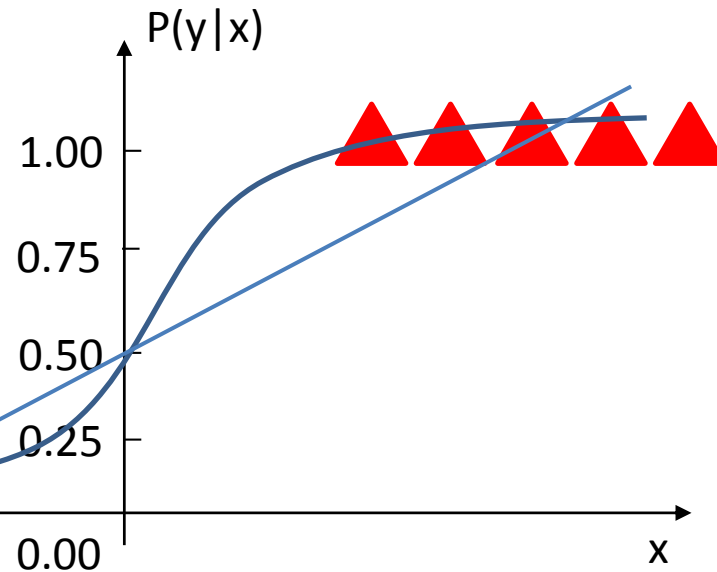
- Logistic regression

- $P(R=1 | Q,D) = \sigma(w^T X) = \frac{1}{1 + \exp(-w^T X)}$
- Directly modeling posterior of document relevance

Sigmoid function

$$\sigma(w^T X) = \frac{1}{1 + \exp(-w^T X)}$$

What if we have an outlier?



# Conditional models for $P(R=1 | Q, D)$

## Pros & Cons

- Advantages
  - Absolute probability of relevance available
  - May re-use all the past relevance judgments
- Problems
  - Performance heavily depends on the selection of features
  - Little guidance on feature selection
- Will be covered with more details in later learning-to-rank discussions

# Generative models for $P(R=1 | Q, D)$

- Basic idea
  - Compute  $\text{Odd}(R=1 | Q, D)$  using Bayes' rule

$$\text{Odd}(R=1 | Q, D) = \frac{P(R=1 | Q, D)}{P(R=0 | Q, D)} = \frac{P(Q, D | R=1)}{P(Q, D | R=0)} \frac{P(R=1)}{P(R=0)} \leftarrow \text{Ignored for ranking}$$

- Assumption
  - Relevance is a binary variable
- Variants
  - Document “generation”
    - $P(Q, D | R) = P(D | Q, R)P(Q | R)$
  - Query “generation”
    - $P(Q, D | R) = P(Q | D, R)P(D | R)$

# Document generation model

$$Odd(R=1 | Q, D) \propto \frac{P(Q, D | R=1)}{P(Q, D | R=0)}$$

	information	retrieval	retrieved	is	helpful	for	you	everyone
Doc1	1	1	0	1	1	1	0	1
Doc2	1	0	1	1	1	1	1	0

Assume independent attributes of  $A_1 \dots A_k \dots$  (why?)

Let  $D = d_1 \dots d_k$ , where  $d_k \in \{0, 1\}$  is the value of attribute  $A_k$  (Similarly  $Q = q_1 \dots q_k$ )

$$Odd(R=1 | Q, D) \propto \prod_{i=1}^k \frac{P(A_i = d_i | Q, R=1)}{P(A_i = d_i | Q, R=0)}$$

Terms *occur* in doc

Terms do *not occur* in doc

$$= \left[ \prod_{i=1, d_i=1}^k \frac{P(A_i = 1 | Q, R=1)}{P(A_i = 1 | Q, R=0)} \right] \left[ \prod_{i=1, d_i=0}^k \frac{P(A_i = 0 | Q, R=1)}{P(A_i = 0 | Q, R=0)} \right]$$

document	relevant(R=1)	nonrelevant(R=0)
term present $A_i=1$	$p_i$	$u_i$
term absent $A_i=0$	$1-p_i$	$1-u_i$



# Document generation model

$$Odd(R=1|Q,D) \propto \prod_{i=1}^k \frac{P(A_i = d_i | Q, R=1)}{P(A_i = d_i | Q, R=0)}$$

Terms *occur* in doc

Terms do *not occur* in doc

$$= \prod_{i \in 1, d_i=1}^k \frac{P(A_i = 1 | Q, R=1)}{P(A_i = 1 | Q, R=0)} \prod_{i \in 1, d_i=0}^k \frac{P(A_i = 0 | Q, R=1)}{P(A_i = 0 | Q, R=0)}$$

Important tricks

Assumption: terms not occurring in the query are equally likely to occur in relevant and nonrelevant documents, i.e.,  $p_t = u_t$

document	relevant(R=1)	nonrelevant(R=0)
term present $A_i=1$	$p_i$	$u_i$
term absent $A_i=0$	$1-p_i$	$1-u_i$

# Robertson-Sparck Jones Model

(Robertson & Sparck Jones 76)

$$\log O(R=1 | Q, D) \stackrel{Rank}{\approx} \sum_{i=1, d_i=q_i=1}^k \log \frac{p_i(1-u_i)}{u_i(1-p_i)} = \sum_{i=1, d_i=q_i=1}^k \log \frac{p_i}{1-p_i} + \log \frac{1-u_i}{u_i} \quad (\text{RSJ model})$$

**Two parameters for each term  $A_i$ :**

$p_i = P(A_i=1 | Q, R=1)$ : prob. that term  $A_i$  occurs in a relevant doc

$u_i = P(A_i=1 | Q, R=0)$ : prob. that term  $A_i$  occurs in a non-relevant doc

How to estimate these parameters?

Suppose we have relevance judgments,

$$\hat{p}_i = \frac{\#(\text{rel. doc with } A_i) + 0.5}{\#(\text{rel.doc}) + 1} \quad \hat{u}_i = \frac{\#(\text{nonrel. doc with } A_i) + 0.5}{\#(\text{nonrel.doc}) + 1}$$

- “+0.5” and “+1” can be justified by Bayesian estimation as priors

***Per-query estimation!***

# Parameter estimation

- General setting:
  - Given a (hypothesized & probabilistic) model that governs the random experiment
  - The model gives a probability of any data  $p(D|\theta)$  that depends on the parameter  $\theta$
  - Now, given actual sample data  $X=\{x_1, \dots, x_n\}$ , what can we say about the value of  $\theta$ ?
- Intuitively, take our best guess of  $\theta$  -- “best” means “best explaining/fitting the data”
- Generally an optimization problem

# Maximum likelihood vs. Bayesian

- Maximum likelihood estimation
  - “Best” means “data likelihood reaches maximum”

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(X|\theta)$$

- Issue: small sample size

*ML: Frequentist's point of view*

- Bayesian estimation

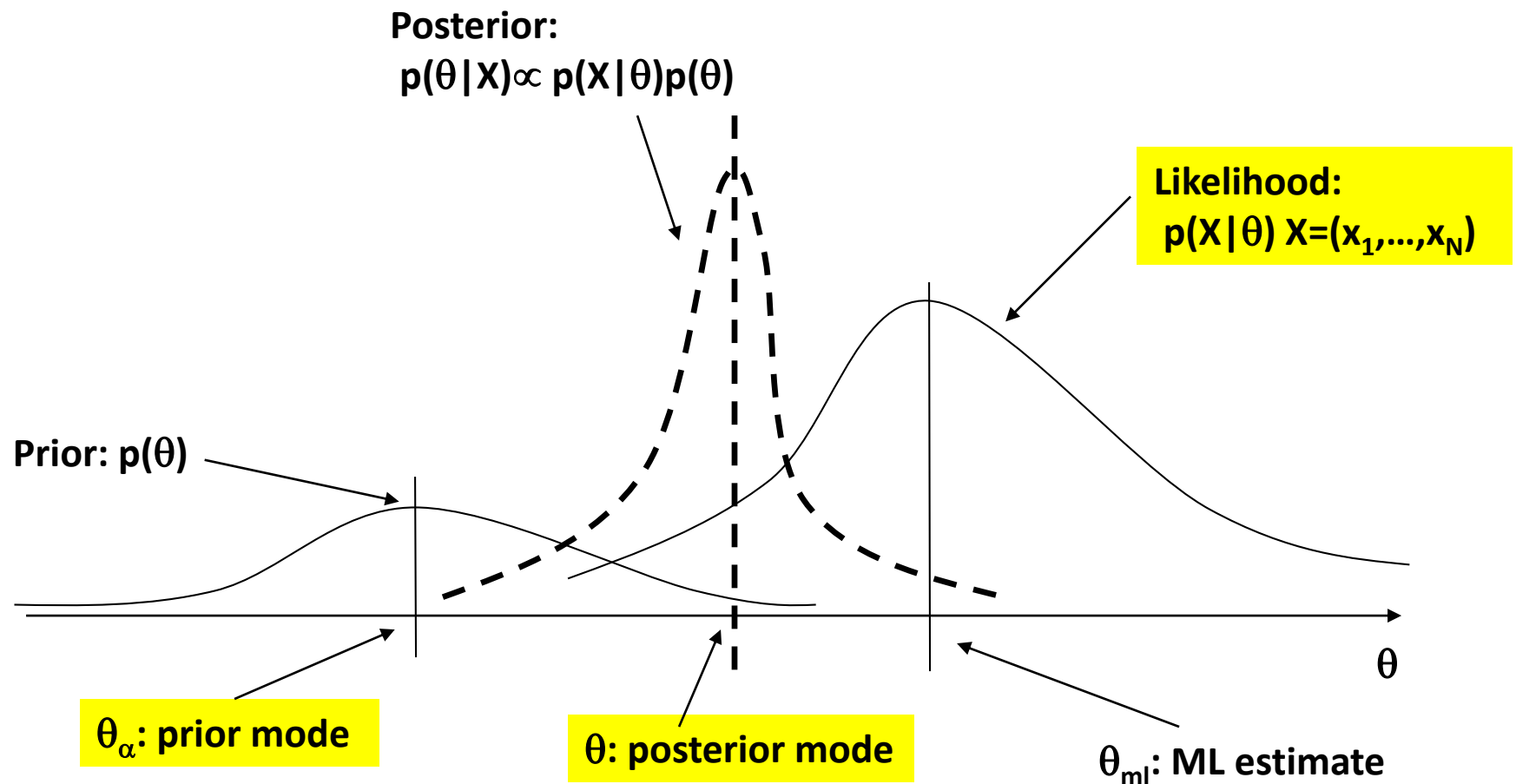
- “Best” means being consistent with our “prior” knowledge and explaining data well

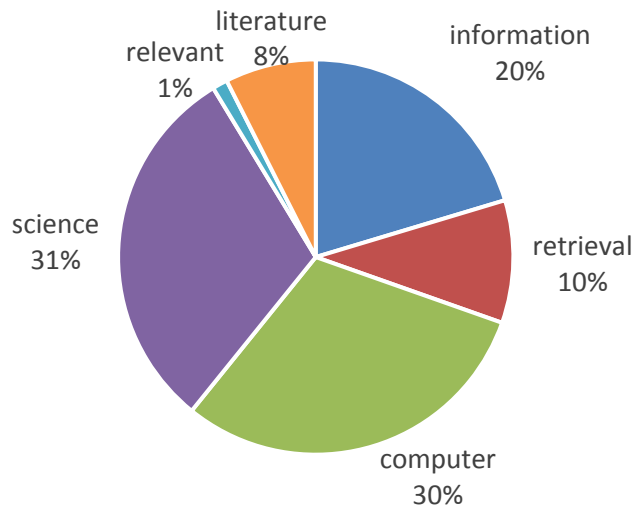
$$\hat{\theta} = \operatorname{argmax}_{\theta} P(\theta|X) = \operatorname{argmax}_{\theta} P(X|\theta)P(\theta)$$

- A.k.a, Maximum a Posterior estimation
  - Issue: how to define prior?

*MAP: Bayesian's point of view*

# Illustration of Bayesian estimation





■ information ■ retrieval ■ computer ■ science ■ relevant ■ literature



- Maximum likelihood estimator:  $\hat{\theta} = \operatorname{argmax}_{\theta} p(W|\theta)$

$$p(W|\theta) = \binom{N}{c(w_1), \dots, c(w_N)} \prod_{i=1}^N \theta_i^{c(w_i)} \propto \prod_{i=1}^N \theta_i^{c(w_i)} \Rightarrow \log p(W|\theta) = \sum_{i=1}^N c(w_i) \log \theta_i$$

$$\Rightarrow L(W, \theta) = \sum_{i=1}^N c(w_i) \log \theta_i + \lambda \left( \sum_{i=1}^N \theta_i - 1 \right)$$

$$\Rightarrow \frac{\partial L}{\partial \theta_i} = \frac{c(w_i)}{\theta_i} + \lambda \rightarrow \theta_i = -\frac{c(w_i)}{\lambda}$$

$$\Rightarrow \text{Since } \sum_{i=1}^N \theta_i = 1 \text{ we have } \lambda = -\sum_{i=1}^N c(w_i)$$

$$\Rightarrow \theta_i = \frac{c(w_i)}{\sum_{i=1}^N c(w_i)}$$

**Using Lagrange multiplier approach, we'll tune  $\theta_i$  to maximize  $L(W, \theta)$**

**Set partial derivatives to zero**

**Requirement from probability**

# Robertson-Sparck Jones Model

(Robertson & Sparck Jones 76)

$$\log O(R=1 | Q, D) \stackrel{Rank}{\approx} \sum_{i=1, d_i=q_i=1}^k \log \frac{p_i(1-u_i)}{u_i(1-p_i)} = \sum_{i=1, d_i=q_i=1}^k \log \frac{p_i}{1-p_i} + \log \frac{1-u_i}{u_i} \quad (\text{RSJ model})$$

**Two parameters for each term  $A_i$ :**

$p_i = P(A_i=1 | Q, R=1)$ : prob. that term  $A_i$  occurs in a relevant doc

$u_i = P(A_i=1 | Q, R=0)$ : prob. that term  $A_i$  occurs in a non-relevant doc

How to estimate these parameters?

Suppose we have relevance judgments,

$$\hat{p}_i = \frac{\#(\text{rel. doc with } A_i) + 0.5}{\#(\text{rel.doc}) + 1} \quad \hat{u}_i = \frac{\#(\text{nonrel. doc with } A_i) + 0.5}{\#(\text{nonrel.doc}) + 1}$$

- “+0.5” and “+1” can be justified by Bayesian estimation as priors

***Per-query estimation!***

# RSJ Model without relevance info

(Croft & Harper 79)

	information	retrieval	retrieved	is	helpful	for	you	everyone
Doc1	1	1	0	1	1	1	0	1
Doc2	1	0	1	1	1	1	1	0

Suppose we **do not** have relevance judgments,

- We will assume  $p_i$  to be a constant
- Estimate  $u_i$  by assuming **all** documents to be **non-relevant**

$$\log O(R=1 | Q, D) \approx \sum_{i=1, d_i=q_i=1}^k c + \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

Reminder:

$$IDF = 1 + \log \frac{N}{n_i}$$

**N: # documents in collection**

**$n_i$ : # documents in which term  $A_i$  occurs**

**IDF weighted Boolean model?**



# RSJ Model: summary

- The most important classic probabilistic IR model
- Use only term presence/absence, thus also referred to as Binary Independence Model
  - Essentially Naïve Bayes for doc ranking
  - Designed for short catalog records
- When without relevance judgments, the model parameters must be estimated in an ad-hoc way
- Performance isn't as good as tuned VS models

# Improving RSJ: adding TF

Let  $D=d_1...d_k$ , where  $d_k$  is the frequency count of term  $A_k$

$$\begin{aligned} \frac{P(R=1|Q,D)}{P(R=0|Q,D)} &\propto \prod_{i=1}^k \frac{P(A_i = d_i | Q, R=1)}{P(A_i = d_i | Q, R=0)} \\ &= \prod_{i=1, d_i \geq 1}^k \frac{P(A_i = d_i | Q, R=1)}{P(A_i = d_i | Q, R=0)} \prod_{i=1, d_i=0}^k \frac{P(A_i = 0 | Q, R=1)}{P(A_i = 0 | Q, R=0)} \\ &\propto \prod_{i=1, d_i \geq 1}^k \frac{P(A_i = d_i | Q, R=1)P(A_i = 0 | Q, R=0)}{P(A_i = d_i | Q, R=0)P(A_i = 0 | Q, R=1)} \end{aligned}$$

## 2-Poisson mixture model for TF

**Eliteness:** if the term is about  
the concept asked in the query

$$\begin{aligned} p(A_i = f | Q, R) &= p(E_i | Q, R) p(A_i = f | E) + P(\bar{E}_i | Q, R) p(A_i = f | \bar{E}) \\ &= p(E_i | Q, R) \frac{\mu_E^f}{f!} e^{-\mu_E} + P(\bar{E}_i | Q, R) \frac{\mu_{\bar{E}}^f}{f!} e^{-\mu_{\bar{E}}} \end{aligned}$$

**Many more parameters to estimate! (how many exactly?)**

**Compound with document length!**

# BM25/Okapi approximation

(Robertson et al. 94)

- Idea: Approximate  $p(R=1 | Q, D)$  with a simpler function that approximates 2-Poisson mixture model


- Observations: 
$$\frac{P(R=1 | Q, D)}{P(R=0 | Q, D)} \propto \prod_{i=1, d_i \geq 1}^k \frac{P(A_i = d_i | Q, R=1)P(A_i = 0 | Q, R=0)}{P(A_i = d_i | Q, R=0)P(A_i = 0 | Q, R=1)}$$

- $\log O(R=1 | Q, D)$  is a sum of term weights occurring in both query and document
  - Term weight  $W_i = 0$ , if  $TF_i = 0$
  - $W_i$  increases monotonically with  $TF_i$
  - $W_i$  has an asymptotic limit
- The simple function is 
$$W_i = \frac{TF_i(k_1 + 1)}{K + TF_i} \log \frac{p_i(1 - u_i)}{u_i(1 - p_i)}$$

# Adding doc. length

- Incorporating doc length
  - Motivation: the 2-Poisson model assumes equal document length
  - Implementation: penalize long doc
    - $$W_i = \frac{TF_i(k_1 + 1)}{K + TF_i} \log \frac{p_i(1 - u_i)}{u_i(1 - p_i)}$$

where  $K = k_1((1 - b) + b \times \frac{|d|}{\text{avg } |d|})$



*Pivoted document length normalization*

# Adding query TF

- Incorporating query TF
  - Motivation
    - Natural symmetry between document and query
  - Implementation: a similar TF transformation as in document TF

$$W_i = \frac{QTF_i(k_s + 1)}{k_s + QTF_i} \frac{TF_i(k_1 + 1)}{K + TF_i} \log \frac{p_i(1 - u_i)}{u_i(1 - p_i)}$$

- The final formula is called BM25, achieving top TREC performance

*BM: best match*

# The BM25 formula

$$\sum_{T \in Q} w^{(1)} \frac{(k_1 + 1)tf}{K + tf} \frac{(k_3 + 1)qtf}{k_3 + qtf} \quad (1)$$

where

**“Okapi TF/BM25 TF”**

$Q$  is a query, containing terms  $T$

$w^{(1)}$  is the Robertson/Sparck Jones weight [5] of  $T$  in  $Q$

$$\log \frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - R + r + 0.5)} \quad (2)$$

$N$  is the number of items (documents) in the collection

$n$  is the number of documents containing the term

$R$  is the number of documents known to be relevant to a specific topic

$r$  is the number of relevant documents containing the term

$K$  is  $k_1((1 - b) + b.dl/avdl)$

$k_1$ ,  $b$  and  $k_3$  are parameters which depend on the nature of the queries and possibly on the database;  $k_1$  and  $b$  default to 1.2 and 0.75 respectively, but smaller values of  $b$  are sometimes advantageous; in long queries  $k_3$  is often set to 7 or 1000 (effectively infinite)

$tf$  is the frequency of occurrence of the term within a specific document

$qtf$  is the frequency of the term within the topic from which  $Q$  was derived

$dl$  and  $avdl$  are respectively the document length and average document length measured in some suitable unit.

.. . . .

*becomes IDF when no  
relevance info is available*

# The BM25 formula

- A closer look

$$rel(q, D) = \sum_{i=1}^n \overbrace{IDF(q_i) \frac{tf_i(k_1 + 1)}{tf_i + k_1(1 - b + b \frac{|D|}{avg |D|})}}^{TF-IDF \text{ component for document}} \underbrace{\frac{qtf_i(k_2 + 1)}{k_2 + qtf_i}}_{TF \text{ component for query}}$$

- $b$  is usually set to  $[0.75, 1.2]$
- $k_1$  is usually set to  $[1.2, 2.0]$
- $k_2$  is usually set to  $(0, 1000]$

**Vector space model  
with TF-IDF schema!**

# Extensions of “Doc Generation” models

- Capture term dependence [Rijsbergen & Harper 78]
- Alternative ways to incorporate TF [Croft 83, Kalt96]
- Feature/term selection for feedback [Okapi's TREC reports]
- Estimate of the relevance model based on pseudo feedback [Lavrenko & Croft 01]

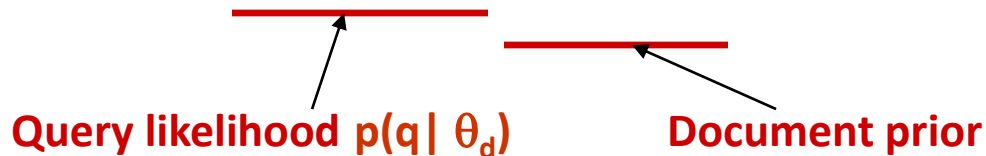


to be covered later



# Query generation models

$$O(R=1|Q,D) \propto \frac{P(Q,D|R=1)}{P(Q,D|R=0)}$$



Assuming uniform document prior, we have

$$O(R=1|Q,D) \propto P(Q|D,R=1)$$

Now, the question is how to compute  $P(Q|D,R=1)$  ?

Generally involves two steps:

- (1) estimate a language model based on D
- (2) compute the query likelihood according to the estimated model

***Language models, we will cover it in next lecture!***

# What you should know

- Essential concepts in probability
- Justification of ranking by relevance
- Derivation of RSJ model
- Maximum likelihood estimation
- BM25 formula

# Today's reading

- Chapter 11. Probabilistic information retrieval
  - 11.2 The Probability Ranking Principle
  - 11.3 The Binary Independence Model
  - 11.4.3 Okapi BM25: a non-binary model