

Information Retrieval

Probabilistic Information Retrieval

Wessel Kraaij 2022

Based on Stanford IIR book slides

Christopher Manning and Pandu Nayak

Today

1. Introduction to the classical probabilistic retrieval model and the probability ranking principle
2. Three families of probabilistic modeling for IR
3. The Binary Independence (retrieval) Model: BIM
4. Relevance feedback, briefly
5. BM25 model
 - Revisiting Document Length normalization

Exploring a different IR modeling paradigm

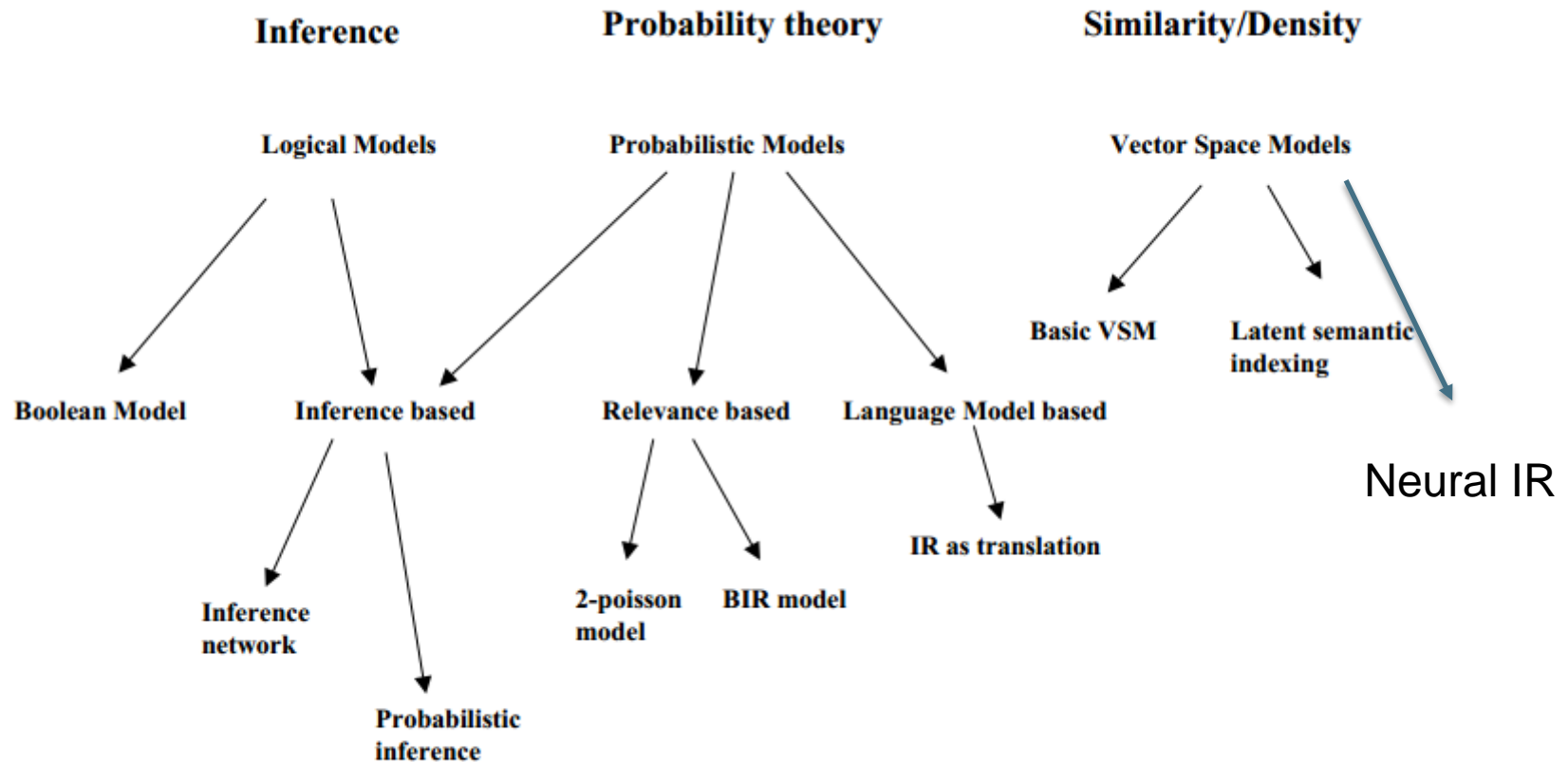
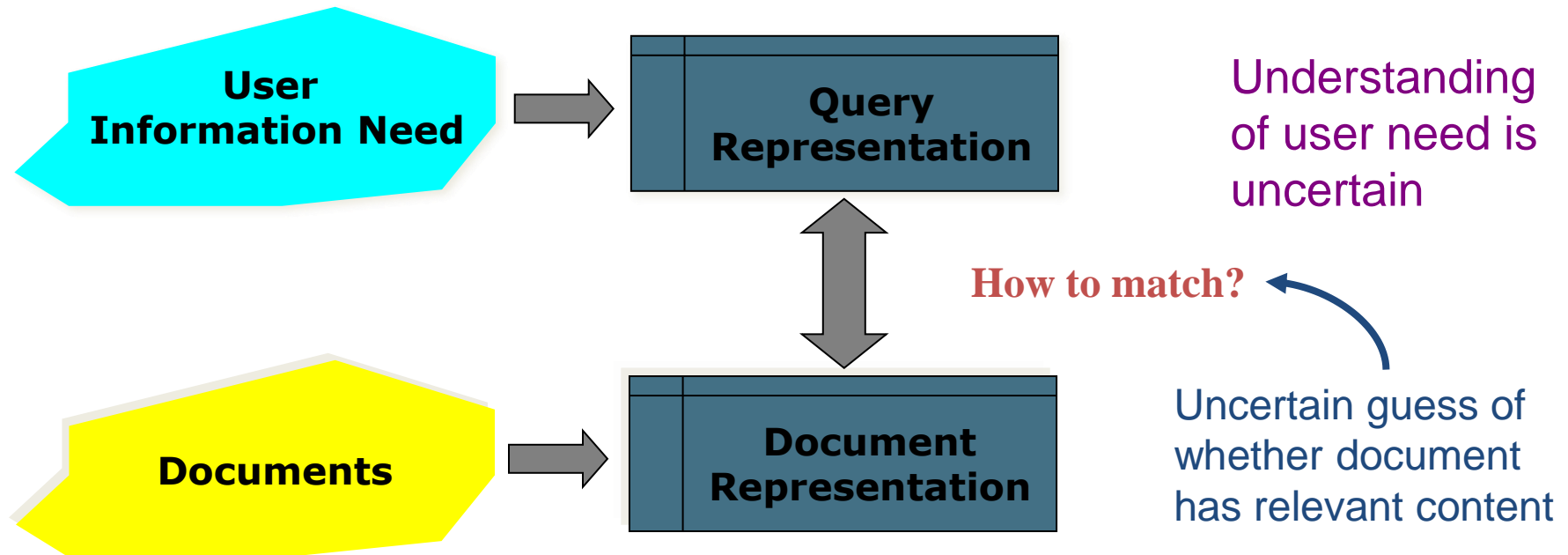


Figure 2.3. Taxonomy of IR models

1. Why probabilities in IR?



In traditional IR systems, matching between each document and query is attempted in a semantically imprecise space of index terms.

Probabilities provide a principled foundation for reasoning under uncertainty.

Can we use probabilities to quantify the uncertainties in our ranking function?

The document ranking problem

- We have a collection of documents
- User issues a query
- A list of documents needs to be returned
- **Ranking method is the core of modern IR systems:**
 - In what order do we present documents to the user?
 - We want the “best” document to be first, second best second, etc.
- **Idea: Rank by probability of relevance of the document w.r.t. information need**
 - $P(R=1 \mid \text{document}_i, \text{query})$

The Probability Ranking Principle (PRP)

“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.”

- [1960s/1970s] S. Robertson, W.S. Cooper, M.E. Maron; van Rijsbergen (1979:113); Manning & Schütze (1999:538)

Refresher a few probability basics

- For events A and B :

$$p(A, B) = p(A \cap B) = p(A | B)p(B) = p(B | A)p(A)$$

- Bayes' Rule

$$p(A | B) = \frac{p(B | A)p(A)}{p(B)} = \frac{p(B | A)p(A)}{\sum_{X=A, \bar{A}} p(B | X)p(X)}$$

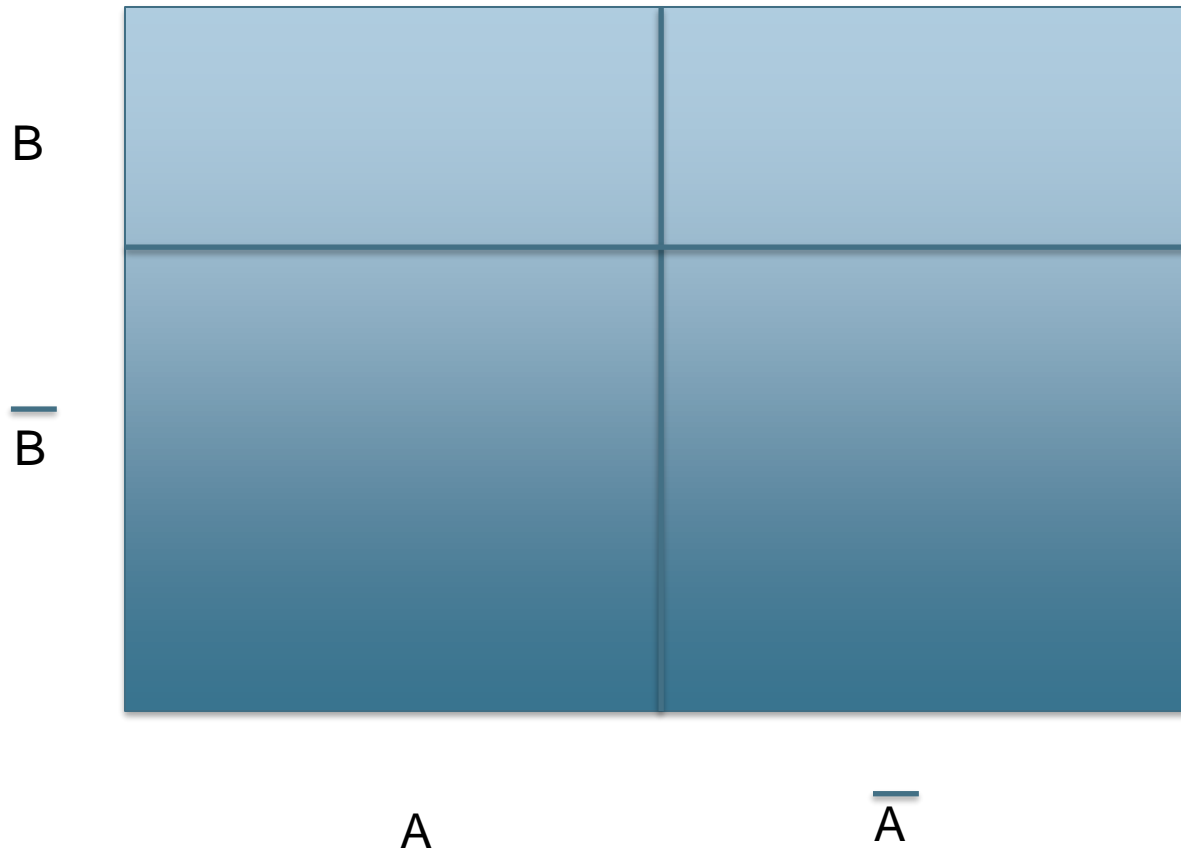
Posterior

Prior

- Odds ratio: $O(A) = \frac{p(A)}{p(\bar{A})} = \frac{p(A)}{1 - p(A)}$

Example

- $P(A,B) = ?$



The Probability Ranking Principle (PRP)

Let x represent a document in the collection.

Let R represent **relevance** of a document w.r.t. given (fixed) query and let $R=1$ represent relevant and $R=0$ not relevant.

Need to find $p(R=1 | x)$ – probability that a document x is **relevant**.

$$p(R = 1 | x) = \frac{p(x | R = 1)p(R = 1)}{p(x)}$$

$p(R=1), p(R=0)$ - prior probability of retrieving a relevant or non-relevant document at random

$$p(R = 0 | x) = \frac{p(x | R = 0)p(R = 0)}{p(x)}$$

$p(x/R=1), p(x/R=0)$ - probability that if a relevant (not relevant) document is retrieved, it is x .

$$p(R = 0 | x) + p(R = 1 | x) = 1$$

Probabilistic Retrieval Strategy

- First, estimate how each term contributes to relevance
 - How do other things like term frequency and document length influence your judgments about document relevance?
 - Not at all in BIM
 - A more nuanced answer is given by BM25
- Combine to find document relevance probability
- Order documents by decreasing probability
- Theorem: Using the PRP is optimal, in that it minimizes the loss (Bayes risk) under 1/0 loss
 - Provable if all probabilities correct, etc. [e.g., Ripley 1996]

2. Probabilistic IR approaches

1. Classical probabilistic retrieval model
 - Probability ranking principle, etc.
 - Binary independence model (\approx Naïve Bayes text cat)
 - (Okapi) BM25
2. Bayesian inference networks (Turtle and Croft)
3. Language model approach to IR (*IIR* ch. 12)
 - An important development in 2000s IR

Probabilistic methods are better grounded in theory than term weighting for vector space approaches

Some pioneers of probabilistic IR



Karen Spärck Jones



Stephen Robertson



Keith van Rijsbergen



Bruce Croft

3. Binary Independence Model

- **“Binary” = Boolean**: documents are represented as binary incidence vectors of terms (cf. IIR Chapter 1):
 - $\vec{x} = (x_1, \dots, x_n)$
 - $x_i = 1$ iff term i is present in document x .
- **“Independence”**: terms occur in documents independently
- Different documents can be modeled as the same vector

Binary Independence Model

- Queries: binary term incidence vectors
- Given query q ,
 - for each document d need to compute $p(R|q,d)$
 - replace with computing $p(R|q,x)$ where x is binary term incidence vector representing d
 - Interested only in ranking
- Will use odds and Bayes' Rule:

$$O(R|q, \vec{x}) = \frac{p(R=1|q, \vec{x})}{p(R=0|q, \vec{x})} = \frac{\frac{p(R=1|q)p(\vec{x}|R=1,q)}{\cancel{p(\vec{x}|q)}}}{\frac{p(R=0|q)p(\vec{x}|R=0,q)}{\cancel{p(\vec{x}|q)}}}$$

Binary Independence Model

$$O(R | q, \vec{x}) = \frac{p(R = 1 | q, \vec{x})}{p(R = 0 | q, \vec{x})} = \frac{p(R = 1 | q)}{p(R = 0 | q)} \times \frac{p(\vec{x} | R = 1, q)}{p(\vec{x} | R = 0, q)}$$

Constant for a
given query

Needs estimation

- Using linked **dependence** Assumption:

$$\frac{p(\vec{x} | R = 1, q)}{p(\vec{x} | R = 0, q)} = \prod_{i=1}^n \frac{p(x_i | R = 1, q)}{p(x_i | R = 0, q)}$$

$$O(R | q, \vec{x}) = O(R | q) \times \prod_{i=1}^n \frac{p(x_i | R = 1, q)}{p(x_i | R = 0, q)}$$

Binary Independence Model

$$O(R | q, \vec{x}) = O(R | q) \times \prod_{i=1}^n \frac{p(x_i | R = 1, q)}{p(x_i | R = 0, q)}$$

- Since x_i is either 0 or 1:

$$O(R | q, \vec{x}) = O(R | q) \times \prod_{x_i=1} \frac{p(x_i = 1 | R = 1, q)}{p(x_i = 1 | R = 0, q)} \times \prod_{x_i=0} \frac{p(x_i = 0 | R = 1, q)}{p(x_i = 0 | R = 0, q)}$$

- Let $p_i = p(x_i = 1 | R = 1, q)$; $r_i = p(x_i = 1 | R = 0, q)$;
- Assume, for all terms not occurring in the query ($q_i=0$) $p_i = r_i$

$$O(R | q, \vec{x}) = O(R | q) \times \prod_{\substack{x_i=1 \\ q_i=1}} \frac{p_i}{r_i} \times \prod_{\substack{x_i=0 \\ q_i=1}} \frac{(1 - p_i)}{(1 - r_i)}$$

Overview of probabilities and parameters

	document	relevant (R=1)	not relevant (R=0)
term present	$x_i = 1$	p_i	r_i
term absent	$x_i = 0$	$(1 - p_i)$	$(1 - r_i)$

Any relation between p_i and r_i ?

Binary Independence Model

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{x_i=q_i=1} \frac{p_i}{r_i} \times \prod_{x_i=0, q_i=1} \frac{1-p_i}{1-r_i} = 1!$$

All matching terms

Non-matching query terms

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{\substack{x_i=1 \\ q_i=1}} \frac{p_i}{r_i} \times \prod_{\substack{x_i=1 \\ q_i=1}} \frac{1-r_i}{1-p_i} \times \prod_{\substack{x_i=0 \\ q_i=1}} \frac{1-p_i}{1-r_i}$$

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \times \prod_{q_i=1} \frac{1-p_i}{1-r_i}$$

All matching terms

All query terms

Binary Independence Model

$$O(R | q, \vec{x}) = O(R | q) \cdot \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \cdot \prod_{q_i=1} \frac{1-p_i}{1-r_i}$$

Diagram illustrating the components of the Binary Independence Model formula:

- $O(R | q)$ is labeled as "Constant for each query".
- The product term $\prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)}$ and the product term $\prod_{q_i=1} \frac{1-p_i}{1-r_i}$ are grouped together by a double-headed arrow, indicating they are the "Only quantity to be estimated for rankings".

Retrieval Status Value:

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

Binary Independence Model

[Robertson & Spärck-Jones 1976]

All boils down to computing RSV.

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

$$RSV = \sum_{x_i=q_i=1} c_i; \quad c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

The c_i are **log odds ratios** (of contingency table a few slides back)
They function as the term weights in this model

So, how do we compute c_i 's from our data?

Binary Independence Model

- Estimating RSV coefficients in theory
- For each term i look at this table of document counts:

$$c_i = \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

Documents	Relevant	Non-Relevant	Total
$x_i=1$	s	$n-s$	n
$x_i=0$	$S-s$	$N-n-S+s$	$N-n$
Total	S	$N-S$	N

• Estimates:

$$p_i \approx \frac{s}{S} \quad r_i \approx \frac{(n-s)}{(N-S)}$$

$$c_i \approx K(N, n, S, s) = \log \frac{s/(S-s)}{(n-s)/(N-n-S+s)}$$

For now, assume no zero terms. Remember smoothing.

Smoothed estimates

- Smoothing is necessary:
 - We want to avoid dividing by zero
 - We want to avoid taking a log of zero
- Avoid estimates where p_i or $r_i = 0$ or 1
- Laplace smoothing: add a small quantity to the counts
- $p_i = (s+1/2)/(S+1)$
- $r_i = (n-s+1/2)/(N-S+1)$

Estimation of r_i

- If non-relevant documents are approximated by the whole collection, then r_i (prob. of occurrence in non-relevant documents for query) is n/N and

$$\log \frac{1 - r_i}{r_i} = \log \frac{N - n - S + s}{n - s} \gg \log \frac{N - n}{n} \gg \log \frac{N}{n} = IDF!$$

- Inverse Document Frequency (IDF)
 - Spärck-Jones (1972)
 - A key, still-important term weighting concept

Collection vs. Document frequency

- Collection frequency of t is the total number of occurrences of t in the collection (incl. multiples)
- Document frequency is number of docs t is in
- Example:

Word	Collection frequency	Document frequency
<i>insurance</i>	10440	3997
<i>try</i>	10422	8760

- Which word is a better search term (and should get a higher weight)?

Estimation of p_i

- p_i (probability of occurrence in relevant documents) cannot be approximated as easily
- p_i can be estimated in various ways:
 - from relevant documents if you know some
 - Relevance weighting can be used in a feedback loop
 - constant (Croft and Harper combination match) – then just get idf weighting of terms (with $p_i=0.5$)

$$RSV = \prod_{x_i=q_i=1} \log \frac{N}{n_i}$$

Example smoothed estimates of c_i

N=2M	rel S	non rel N-S	p_i smoothed	r_i smoothed	$p_i(1-r_i)/r_i(1-p_i)$	c_i log RSJ
	10	1999990				
Leiden	10	100	0,95	5,02502E-05	417887,57	5,62
University	10	10000	0,95	0,00500027	4178,77	3,62
the	10	1999990	0,95	0,99999975	5,25002E-06	-5,28

C_i value for different p_i and r_i

		r								
		0,1	0,2	0,4	0,5	0,6	0,7	0,8	0,9	0,99999
p	0,1	0	-0,352	-0,778	-0,954	-1,130	-1,322	-1,556	-1,908	-5,954
	0,2	0,352	0	-0,426	-0,602	-0,778	-0,970	-1,204	-1,556	-5,602
	0,3	0,586	0,234	-0,192	-0,368	-0,544	-0,736	-0,970	-1,322	-5,368
	0,4	0,778	0,426	0,000	-0,176	-0,352	-0,544	-0,778	-1,130	-5,176
	0,5	0,954	0,602	0,176	0,000	-0,176	-0,368	-0,602	-0,954	-5,000
	0,6	1,130	0,778	0,352	0,176	0,000	-0,192	-0,426	-0,778	-4,824
	0,7	1,322	0,970	0,544	0,368	0,192	0,000	-0,234	-0,586	-4,632
	0,8	1,556	1,204	0,778	0,602	0,426	0,234	0,000	-0,352	-4,398
	0,9	1,908	1,556	1,130	0,954	0,778	0,586	0,352	-5,3E-16	-4,046
0,99999		5,954	5,602056	5,176087	4,999996	4,823904	4,632019	4,397936	4,045753	0

'good indexing
terms', $p_i > r_i$

4. Probabilistic Relevance Feedback

1. Guess a preliminary probabilistic description of $R=1$ documents; use it to retrieve a set of documents
2. Interact with the user to refine the description: learn some definite members with $R = 1$ and $R = 0$
3. Re-estimate p_i and r_i on the basis of these
 - If x_i appears in V_i within **set of relevant top documents V** : $p_i = |V_i|/|V|$
 - Or can combine new information with original guess (use Bayesian prior):
4. Repeat, thus generating a succession of approximations to relevant documents

$$p_i^{(k+1)} = \frac{|V_i| + \kappa p_i^{(k)}}{|V| + \kappa}$$

κ is
prior
Weight, e.g '5'

Pseudo-relevance feedback

(iteratively auto-estimate p_i and r_i)

1. Assume that p_i is constant over all x_i in query and r_i as before
 - $p_i = 0.5$ (even odds) for any given doc
2. Determine guess of relevant document set:
 - V is fixed size set of highest ranked documents on this model
3. We need to improve our guesses for p_i and r_i , so
 - Use distribution of x_i in docs in V . Let V_i be set of documents containing x_i
 - $p_i = |V_i| / |V|$
 - Assume if not retrieved then not relevant
 - $r_i = (n_i - |V_i|) / (N - |V|)$
4. Go to 2. until converges then return ranking

Assessment of PRP and BIM

- It is possible to reasonably approximate probabilities
 - But either require partial relevance information or need to make do with somewhat inferior term weights
- Requires restrictive assumptions:
 - “Relevance” of each document is independent of others
 - Really, it’s bad to keep on returning **duplicates**
 - Term independence
 - Terms not in query don’t affect the outcome
 - Boolean representation of documents/queries
 - Boolean notion of relevance
- Some of these assumptions can be removed

6. BM25



OpenSource Connections

[What We Do](#)

[Case Studies](#)

[About Us](#)



BM25 The Next Generation of Lucene Relevance

Doug Turnbull – October 16, 2015

There's something new cooking in how Lucene scores text. Instead of the traditional "TF*IDF," Lucene just switched to something called BM25 in trunk. That means a new scoring formula for Solr (Solr 6) and Elasticsearch down the line.

Sounds cool, but what does it all mean? In this article I want to give you an overview of how the switch might be a boon to your Solr and Elasticsearch applications. What was the original TF*IDF? How did it work? What does the new BM25 do better? How do you tune it? Is BM25 right for everything?

Okapi BM25

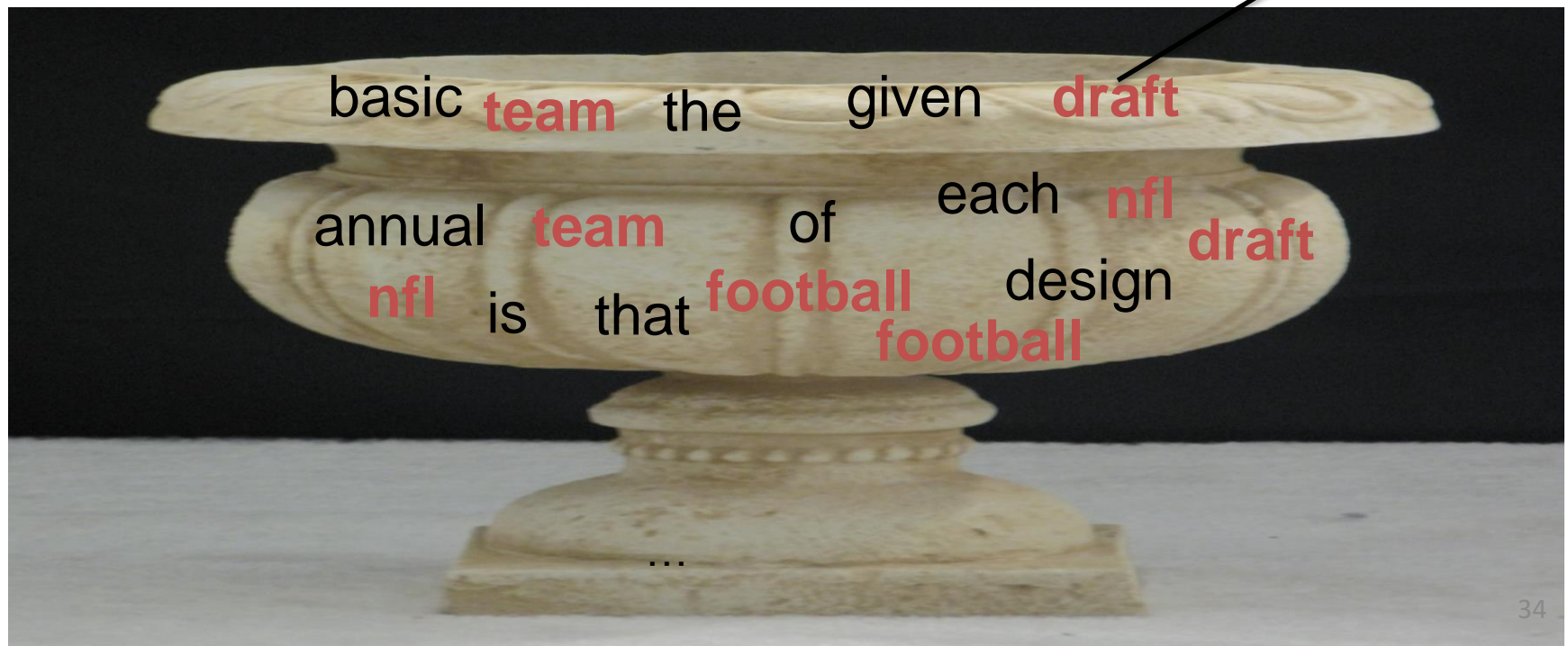
[Robertson et al. 1994, TREC City U., London UK]

- BM25 “Best Match 25” (they had a bunch of tries!)
 - Developed in the context of the Okapi system
 - Started to be increasingly adopted by other teams during the TREC competitions
 - Strong performance
- Goal: be sensitive to term frequency and document length while not adding too many parameters
 - (Robertson and Zaragoza 2009; Spärck Jones et al. 2000)
- First: have a more sophisticated model for tf

Generative model for documents

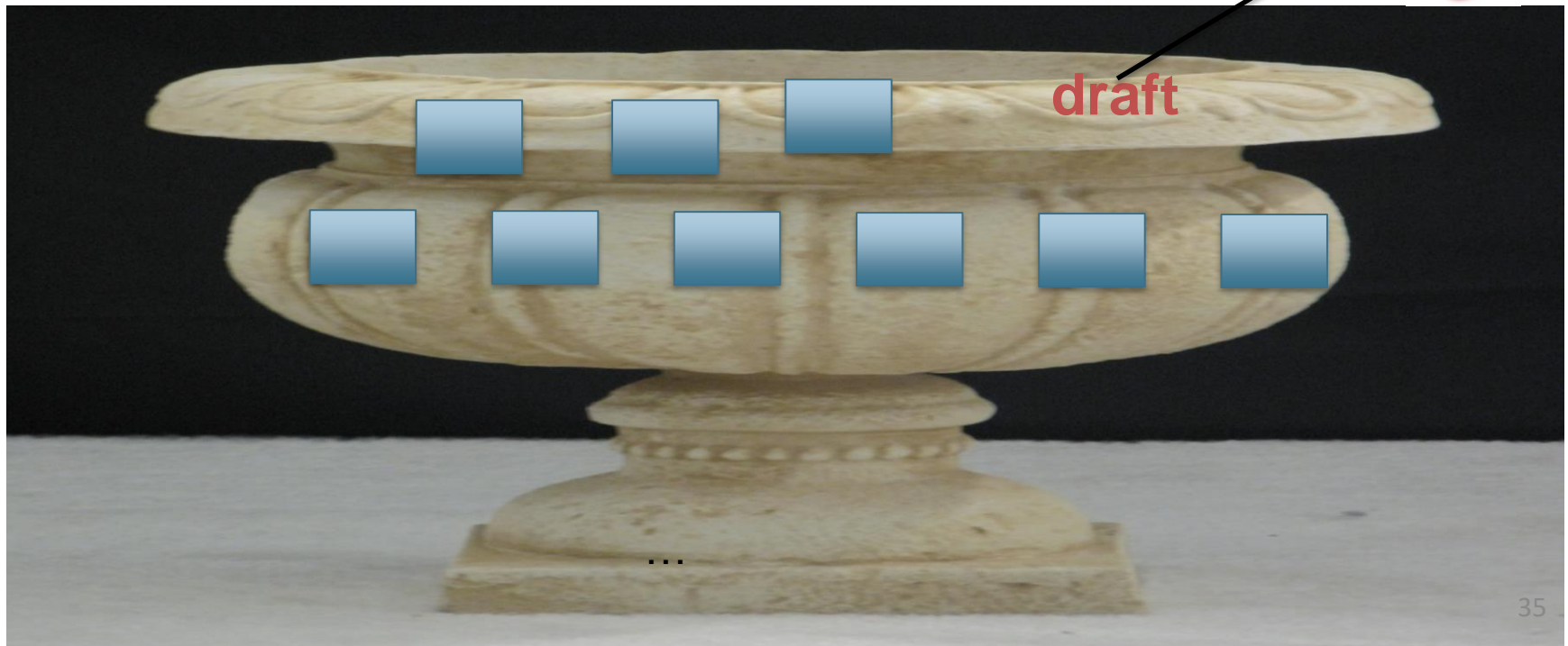
- Words are drawn independently from the vocabulary using a multinomial distribution

... the **draft** is that each **team** is given a position in the **draft** ...



Generative model for documents

- Distribution of term frequencies across documents (tf) follows a binomial distribution (12.1.3) – approximated by a Poisson distribution



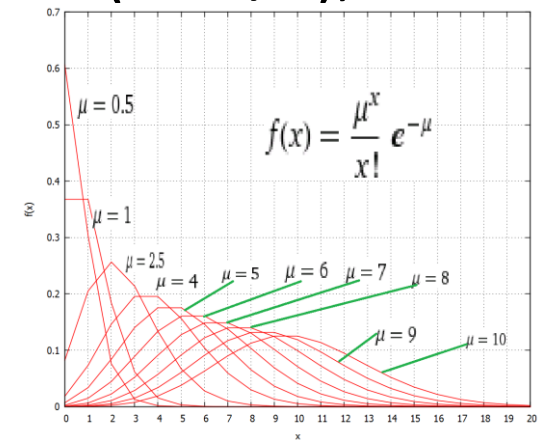
Poisson distribution

- The Poisson distribution models the probability of k , the number of events occurring in a fixed interval of time/space, with known average rate λ ($= cf/T$), independent of the last event

$$p(k) = \frac{\mu^k}{k!} e^{-\mu}$$

- Examples

- Number of cars arriving at a toll booth per minute
- Number of typos on a page



Poisson distribution

- If T is large and p is small, we can approximate a binomial distribution with a Poisson where $\lambda = Tp$

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

- Mean = Variance = $\lambda = Tp$.
- Example $p = 0.08$, $T = 20$. Chance of 1 occurrence is:

- Binomial $P(1) = \binom{20}{1} (.08)^1 (.92)^{19} = .3282$

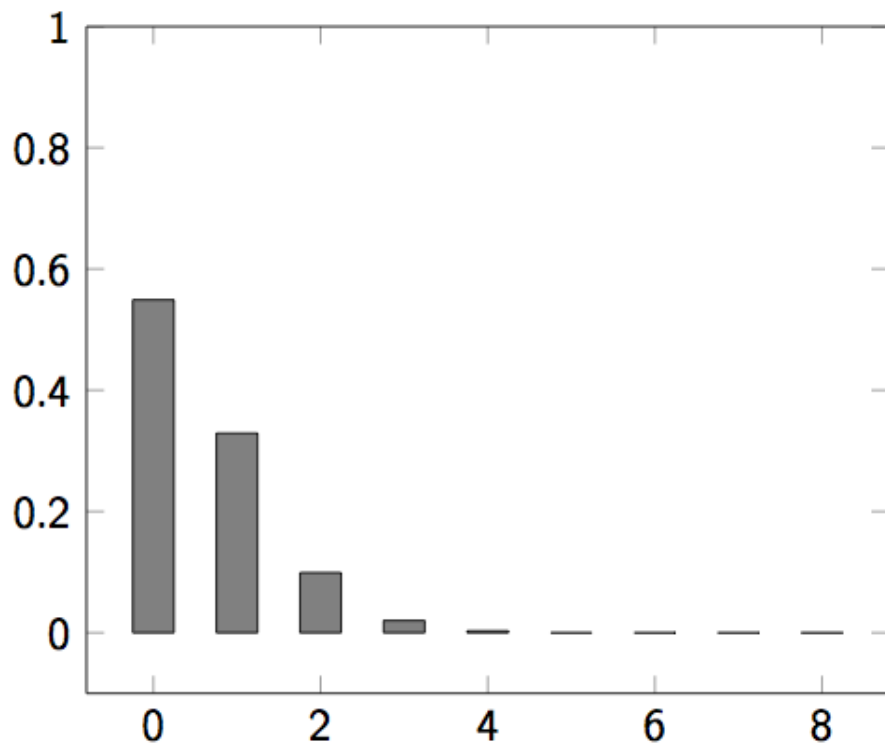
- Poisson $P(1) = \frac{[(20)(.08)]^1}{1!} e^{-(20)(.08)} = \frac{1.6}{1} e^{-1.6} = 0.3230 \quad \dots \text{already close}$

Poisson model

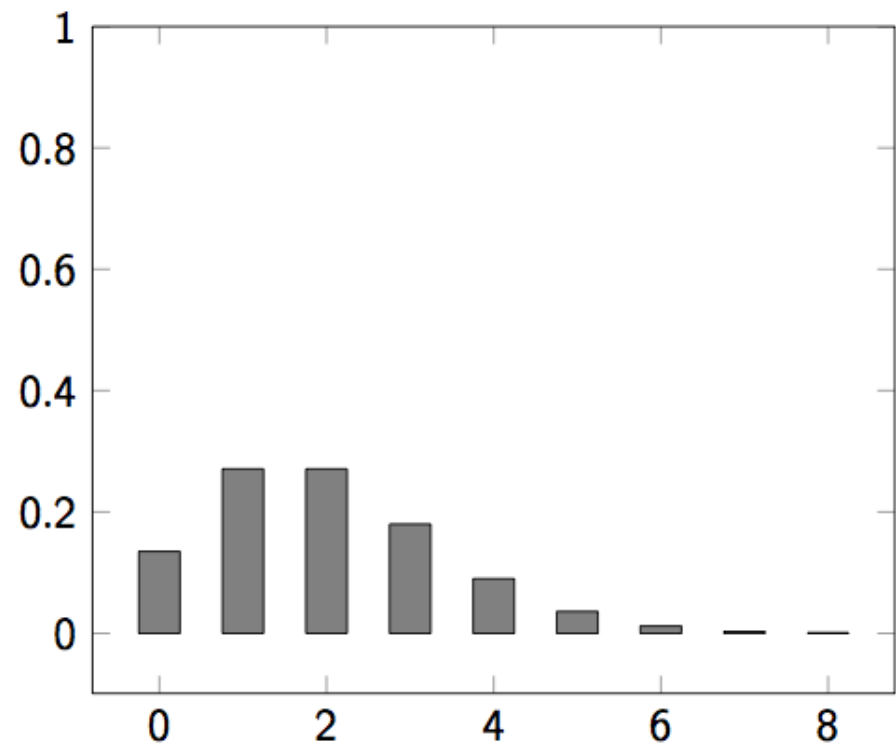
- Assume that term frequencies in a document (tf_i) follow a Poisson distribution
 - “Fixed interval” implies fixed document length ... think roughly constant-sized document abstracts
 - ... will fix later

Poisson distributions

$\lambda = 0.6$



$\lambda = 2$



(One) Poisson Model flaw

- Is a reasonable fit for “general” words
- Is a poor fit for topic-specific words
 - We observe higher $p(k)$ than predicted too often

Burstiness (Polya)

cf=53/T=650

Sum	Works of Freud	Documents containing k occurrences of word ($\lambda = 53/650$)												
Freq	Word	0	1	2	3	4	5	6	7	8	9	10	11	12
53	expected	599	49	2										
52	based	600	48	2										
53	conditions	604	39	7										
55	cathexis	619	22	3	2	1	2	0	1					
51	comic	642	3	0	1	0	0	0	0	0	0	1	1	2

words, words which possess little value for indexing purposes, tend to be distributed at random in a collection of homogeneous documents. In contrast, specialty words tend not to be so distributed.

Eliteness (“aboutness”)

- Model term frequencies using *eliteness*
- What is eliteness?
 - Hidden variable for each document-term pair, denoted as E_i for term i
 - Represents *aboutness*: a term is elite in a document if, in some sense, the document is about the concept denoted by the term
 - Eliteness is binary
 - Term occurrences depend only on eliteness...
 - ... but eliteness depends on relevance

Elite terms

Text from the Wikipedia page on the NFL draft showing **elite terms**

The **National Football League Draft** is an annual event in which the **National Football League (NFL)** teams select eligible college football players. It serves as the league's most common source of player recruitment. The basic design of the **draft** is that each **team** is given a **position** in the **draft order** in **reverse order** relative to its **record** ...

Retrieval Status Value

- Similar to the BIM derivation, we have

$$RSV^{elite} = \sum_{i \in q, tf_i > 0} c_i^{elite}(tf_i);$$

Presence only model

where

$$c_i^{elite}(tf_i) = \log \frac{p(TF_i = tf_i | R = 1) p(TF_i = 0 | R = 0)}{p(TF_i = 0 | R = 1) p(TF_i = tf_i | R = 0)}$$

IDF

We can rewrite, introducing eliteness variable, so moving towards 2 poisson

and using eliteness, we have:

$$\begin{aligned} p(TF_i = tf_i | R) &= \sum_{i=\{0,1\}} p(E_i, TF_i = tf_i | R) = p(TF_i = tf_i | E_i = 1) p(E_i = 1 | R) \\ &\quad + p(TF_i = tf_i | E_i = 0) (1 - p(E_i = 1 | R)) \\ &= \pi p(TF_i = tf_i | E_i = 1) + (1 - \pi) p(TF_i = tf_i | E_i = 0) \end{aligned}$$

π depends
on term i

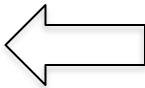
2-Poisson model

- The problems with the 1-Poisson model (flaws) suggests fitting two Poisson distributions
- In the “2-Poisson model”, the distribution is different depending on whether the term is elite or not

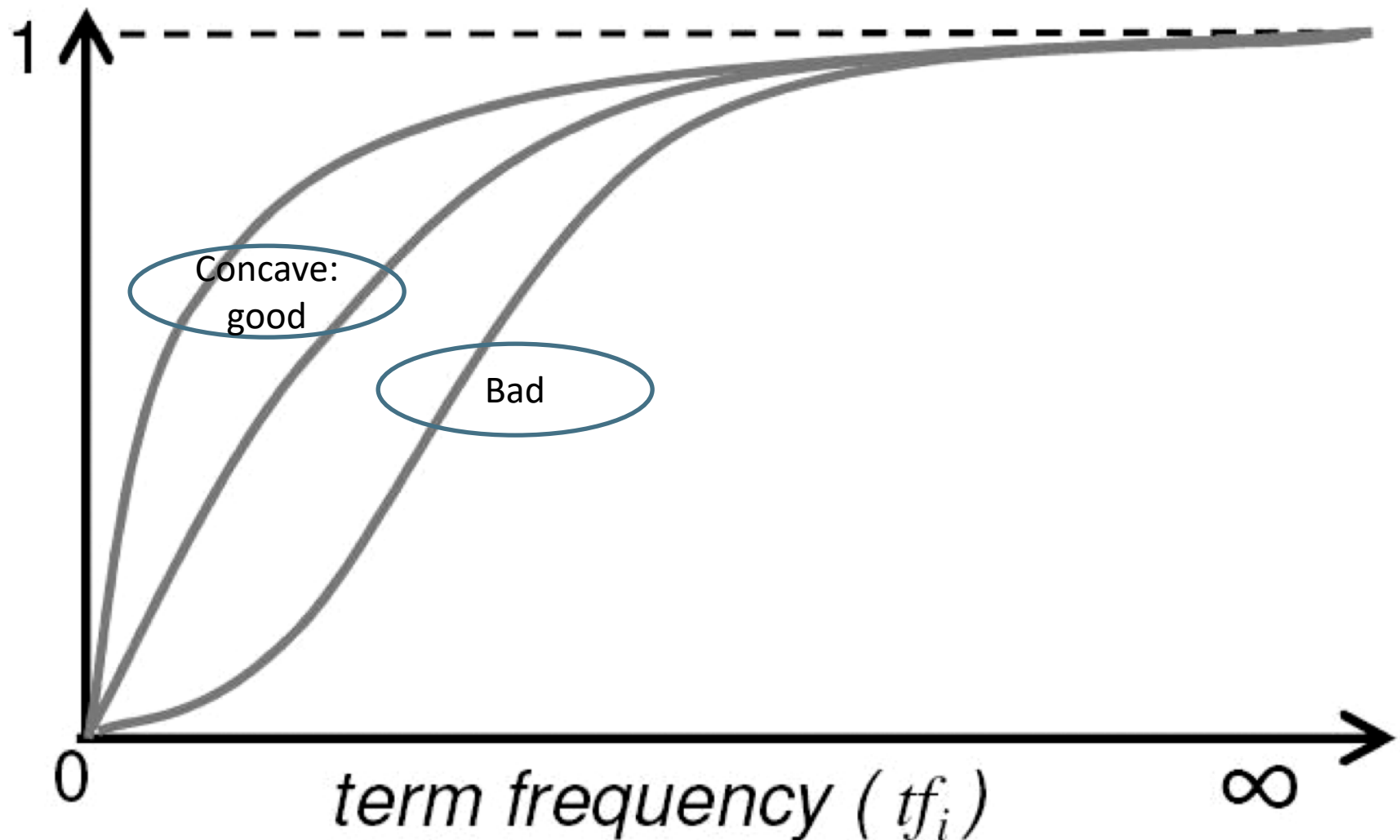
$$p(TF_i = k_i | R) = \pi \frac{\lambda^k}{k!} e^{-\lambda} + (1 - \pi) \frac{\mu^k}{k!} e^{-\mu}$$

- where π is probability that term is elite for document
- but, unfortunately, we don't know π, λ, μ (for each i)
- Note this is not the same k as in BM 25!

Qualitative properties

- $c_i^{elite}(0) = 0$
- $c_i^{elite}(tf_i)$ increases monotonically with tf_i
- ... but asymptotically approaches a maximum value as $tf_i \rightarrow \infty$ [not true for simple scaling of tf]
- ... with the asymptotic limit being c_i^{BIM} 

Let's get an idea: Graphing $c_i^{elite}(tf_i)$ for different parameter values of the 2-Poisson

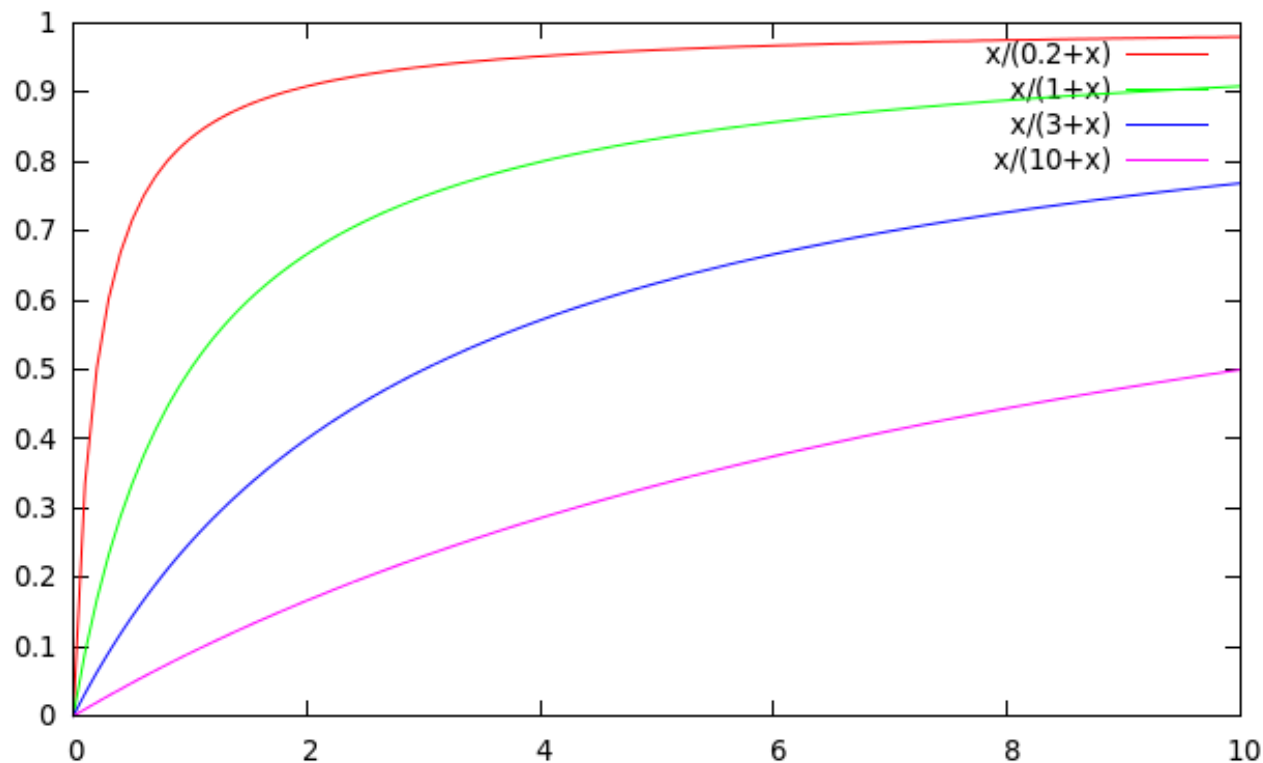


Approximating the saturation function

- Estimating parameters for the 2-Poisson model is not easy
- ... So approximate it with a simple parametric curve that has the same qualitative properties

$$\frac{tf}{k_1 + tf}$$

Saturation function



- For high values of k_1 , increments in tf_i continue to contribute significantly to the score
- Contributions tail off quickly for low values of k_1

“Early” versions of BM25

- Version 1: using the saturation function

$$c_i^{BM25v1}(tf_i) = c_i^{BIM} \frac{tf_i}{k_1 + tf_i}$$

- Version 2: BIM simplification to IDF

$$c_i^{BM25v2}(tf_i) = \log \frac{N}{df_i} \cdot \frac{(k_1 + 1)tf_i}{k_1 + tf_i}$$

- $(k_1 + 1)$ factor doesn't change ranking, but makes term score 1 when $tf_i = 1$
- Similar to $tf-idf$, but term scores are bounded

code	term weight	global	explanation
BM0	1	–	Coordination level matching
BM1	$\log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_i}{k_3+tf_i}$	–	Robertson/Sparck Jones weights plus query term reweighting
BM15	$s_1 s_3 \cdot \frac{tf_i}{k_1+tf_i} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_i}{k_3+tf_i}$	$k_2 \cdot Q ^{\frac{\Delta-d}{\Delta+d}}$	BM1 plus within-document term frequency correction and document length correction
BM11	$s_1 s_3 \cdot \frac{tf_i}{\frac{k_1+d}{\Delta}+tf_i} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_i}{k_3+tf_i}$	$k_2 \cdot Q ^{\frac{\Delta-d}{\Delta+d}}$	BM15 plus within-document term frequency normalization by document length
BM25	$s_1 s_3 \cdot \frac{tf_i^b}{K^c+tf_i^b} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_i}{k_3+tf_i}$	$k_2 \cdot Q ^{\frac{\Delta-d}{\Delta+d}}$	Combination of BM11 and BM15, where $K = k_1((1-b) + b\frac{d}{\Delta})$

Table B.1. Okapi term weighting functions

Document length normalization

- Longer documents are likely to have larger tf_i values
- Why might documents be longer?
 - Verbosity: suggests observed tf_i too high
 - Larger scope: suggests observed tf_i may be right
- A real document collection probably has both effects
- ... so should apply some kind of partial normalization

Document length normalization

- Document length:

$$dl = \sum_{i \in V} tf_i$$

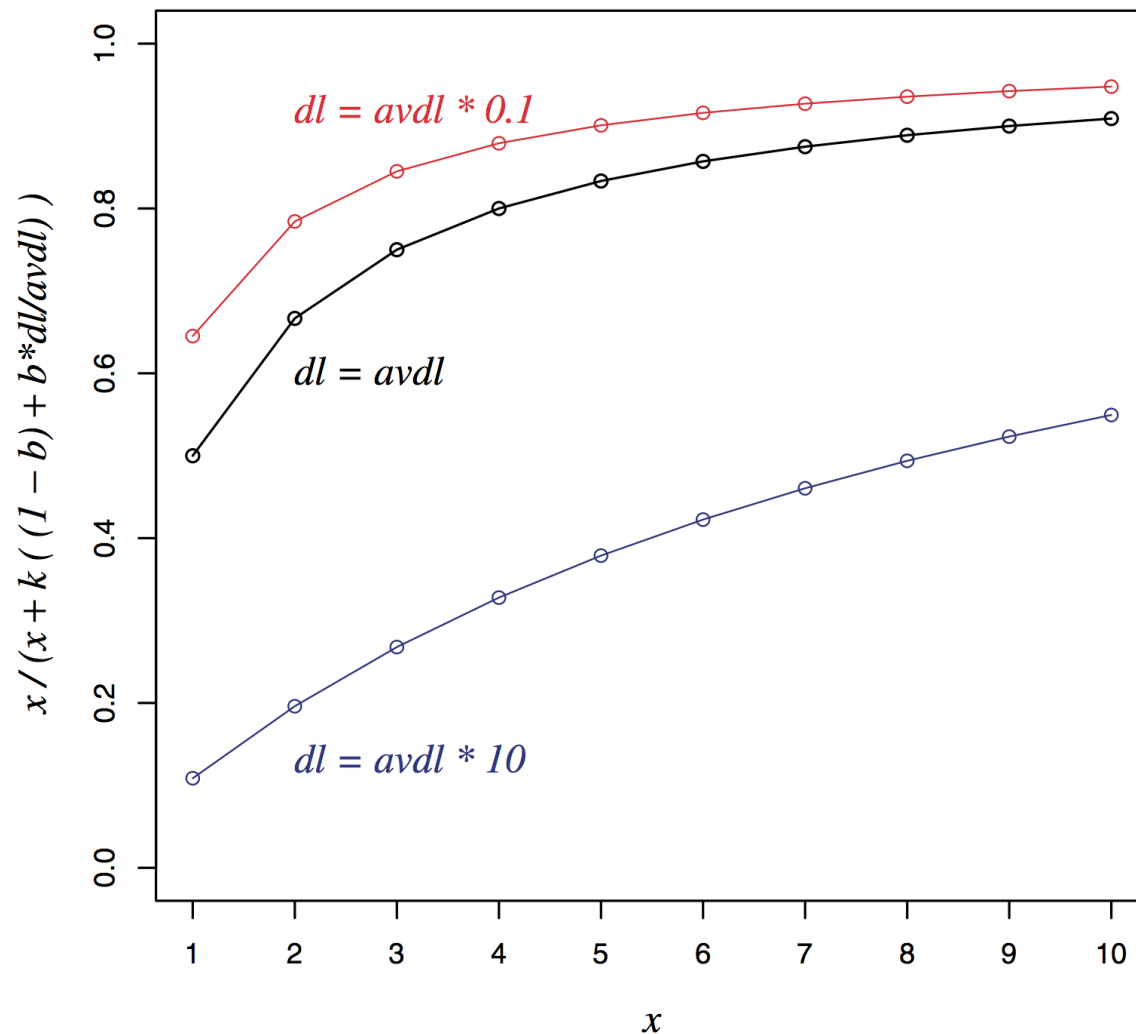
Note: Singhal used #unique terms

- avdl*: Average document length over collection
- Length normalization component

$$B = \frac{dl}{avdl} (1 - b) + b, \quad 0 \leq b \leq 1$$

- $b = 1$ full document length normalization
- $b = 0$ no document length normalization

Document length normalization



Okapi BM25

- Normalize tf using document length

$$tf_i^{\text{norm}} = \frac{tf_i}{B}$$

Fill in B

$$\begin{aligned} c_i^{BM25}(tf_i) &= \log \frac{N}{df_i} + \frac{(k_1 + 1)tf_i^{\text{norm}}}{k_1 + tf_i^{\text{norm}}} \\ &= \log \frac{N}{df_i} + \frac{(k_1 + 1)tf_i}{k_1((1 - b) + b \frac{dl}{avdl}) + tf_i} \end{aligned}$$

- BM25 ranking function

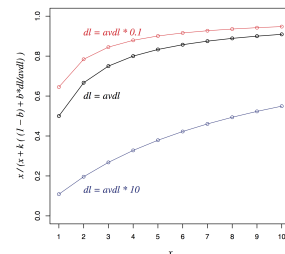
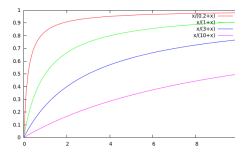
$$RSV^{BM25} = \sum_{i=1}^q c_i^{BM25}(tf_i);$$

BM25 has 2
parameters, k_1 and
 b

Okapi BM25

$$RSV^{BM25} = \frac{1}{\hat{q}} \log \frac{N}{df_i} \times \frac{(k_1 + 1)tf_i}{k_1((1 - b) + b \frac{dl}{avdl}) + tf_i}$$

- k_1 controls term frequency scaling (saturation function)
 - $k_1 = 0$ is binary model; k_1 large is raw term frequency
- b controls document length normalization
 - $b = 0$ is no length normalization; $b = 1$ is relative frequency (fully scale by document length)
- Typically, k_1 is set in range [1.2–2] and b around 0.75
- IIR sec. 11.4.3 discusses incorporating query term weighting and (pseudo) relevance feedback



Why is BM25 better than VSM tf-idf?

- Suppose your query is [machine learning]
- Suppose you have 2 documents with term counts:
 - doc1: learning 1024; machine 1
 - doc2: learning 16; machine 8
- tf-idf: $\log_2 \text{tf} * \log_2 (N/\text{df})$
 - doc1: **11** * 7 + 1 * 10 = **87**
 - doc2: **5** * 7 + 4 * 10 = **75**
- BM25: $k_1 = 2 \rightarrow$ bounded tf-weight!
 - doc1: **3** * 7 + 1 * 10 = **31**
 - doc2: **2.67** * 7 + 2.4 * 10 = **42.7**

Comparing models

	Effectiveness	Efficiency	Explain/Use	Parsimony
Boolean	No ranking	++ (presence only)	++/-	+
Vector Space (Inc.ntc)	Ranking	++ (presence only)	fair	fair
Lnu.ltu	Ranking ++	++ (presence only)	fair	- (more parameters)
Neural IR	Precision oriented ++ (needs BM25 1 st stage)	--	Better than LSI	-
BIM	No Tf!	++ (presence only)	Theory is clear	0 hyperparameters
BM25	++	++ (presence only)	Complex derivation	2 hyperparameters

Resources

- S. E. Robertson and K. Spärck Jones. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information Sciences* 27(3): 129–146.
- C. J. van Rijsbergen. 1979. *Information Retrieval*. 2nd ed. London: Butterworths, chapter 6. <http://www.dcs.gla.ac.uk/Keith/Preface.html>
- K. Spärck Jones, S. Walker, and S. E. Robertson. 2000. A probabilistic model of information retrieval: Development and comparative experiments. Part 1. *Information Processing and Management* 779–808.
- S. E. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3(4): 333-389.

<https://irsg.bcs.org/informer/2020/01/bm25-the-magic-dust-of-search/>

Kamphuis et al, Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants ,
ECIR 2020