Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

BM25
8 February 2018

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?

1. Okapi BM25 [Robertson et al. 1994, TREC City U.]

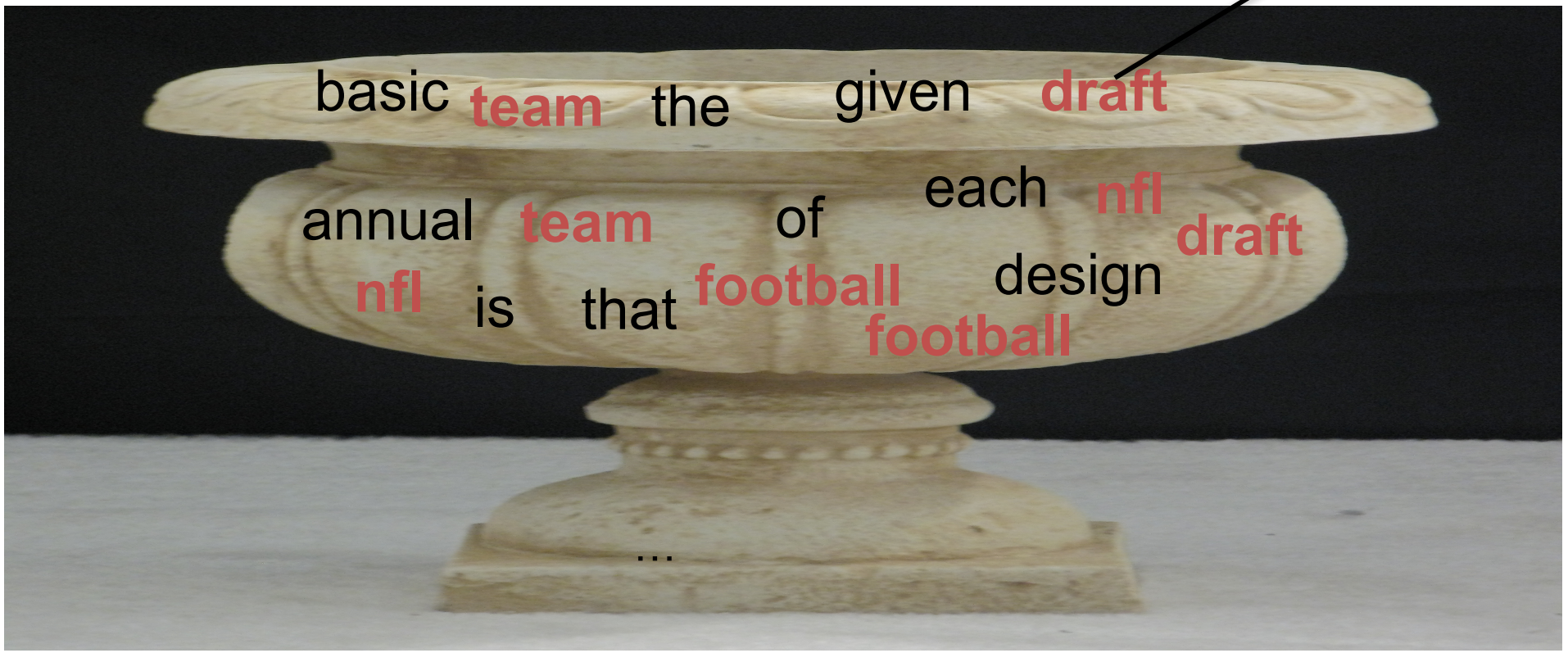
- BM25 "Best Match 25" (they had a bunch of tries!)
  - Developed in the context of the Okapi system
  - Started to be increasingly adopted by other teams during the TREC competitions
  - It works well

- Goal: be sensitive to term frequency and document length while not adding too many parameters
  - (Robertson and Zaragoza 2009; Spärck Jones et al. 2000)

Generative model for documents

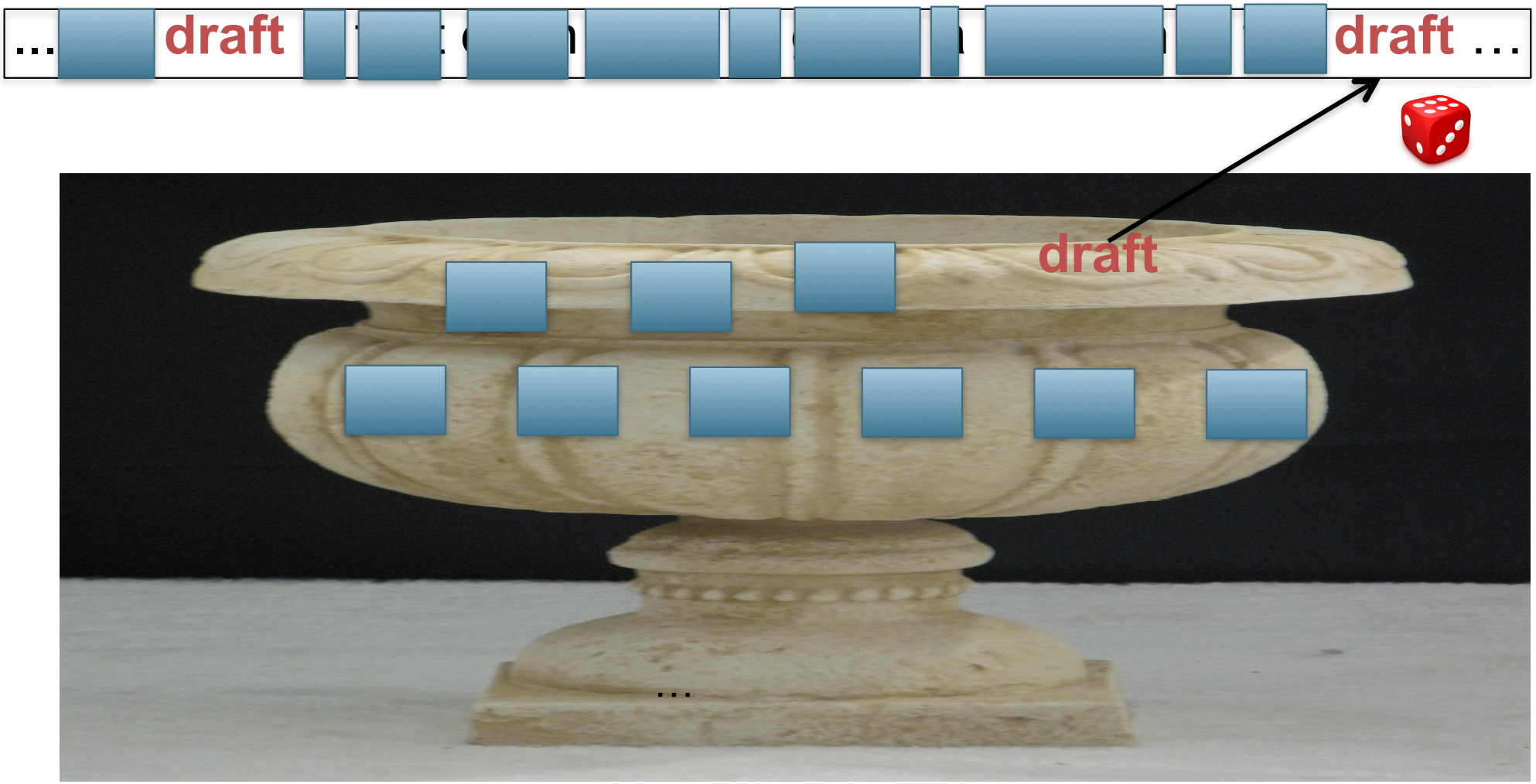
- Words are drawn independently from the vocabulary using a multinomial distribution

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



Generative model for documents

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



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

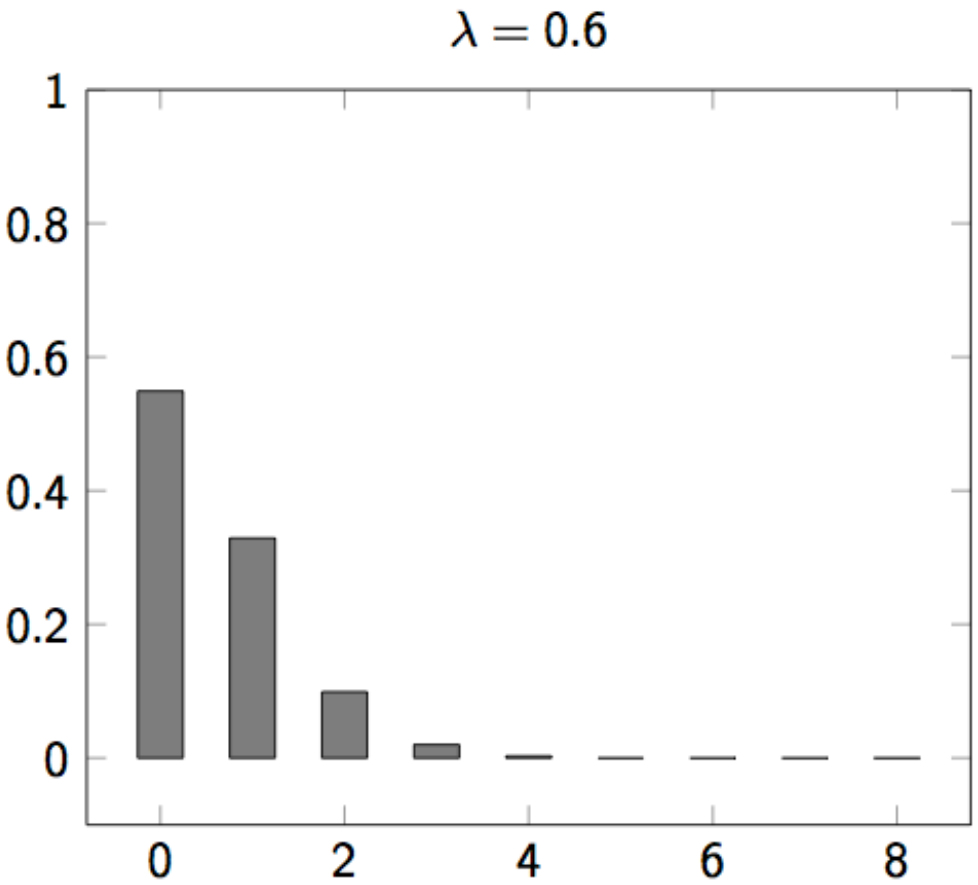
$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

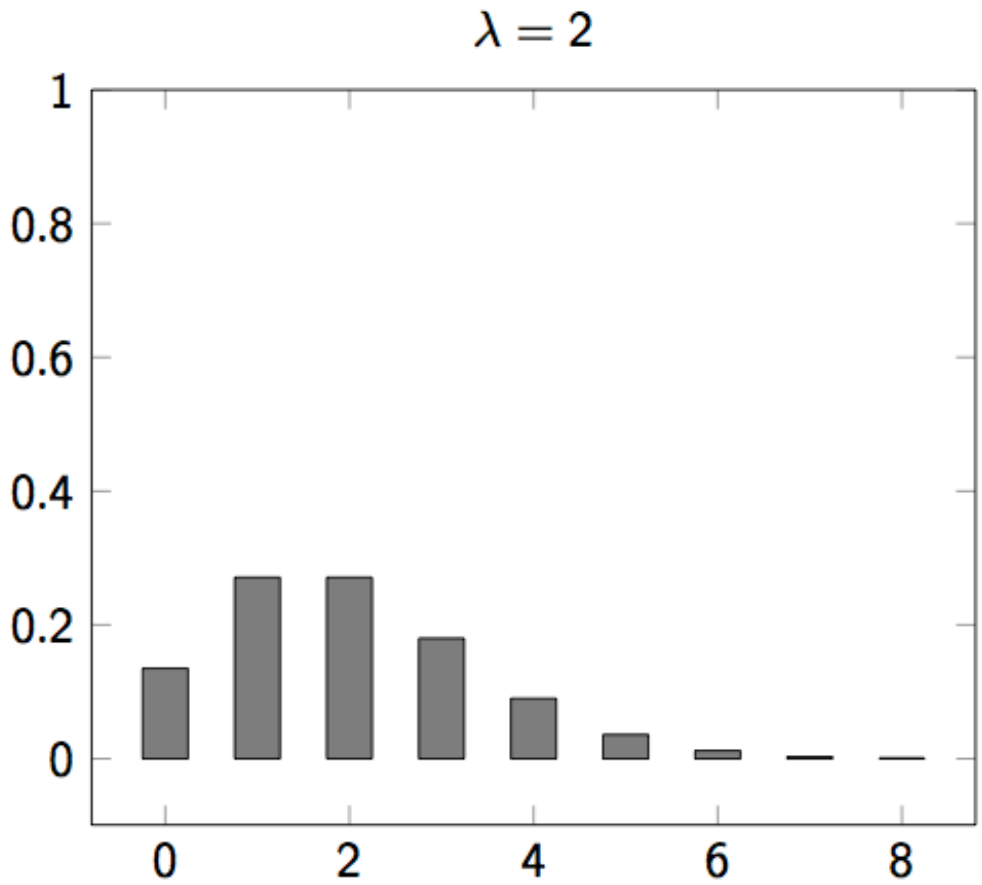
- Examples
  - Number of cars arriving at the toll booth per minute
  - Number of typos on a page

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





(One) Poisson Model

- Is a reasonable fit for "general" words
- Is a poor fit for topic-specific words
  - get higher *p(k)* than predicted too often

		Documents containing *k* occurrences of word (λ = 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

Harter, "A Probabilistic Approach to Automatic Keyword Indexing", JASIST, 1975

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** …

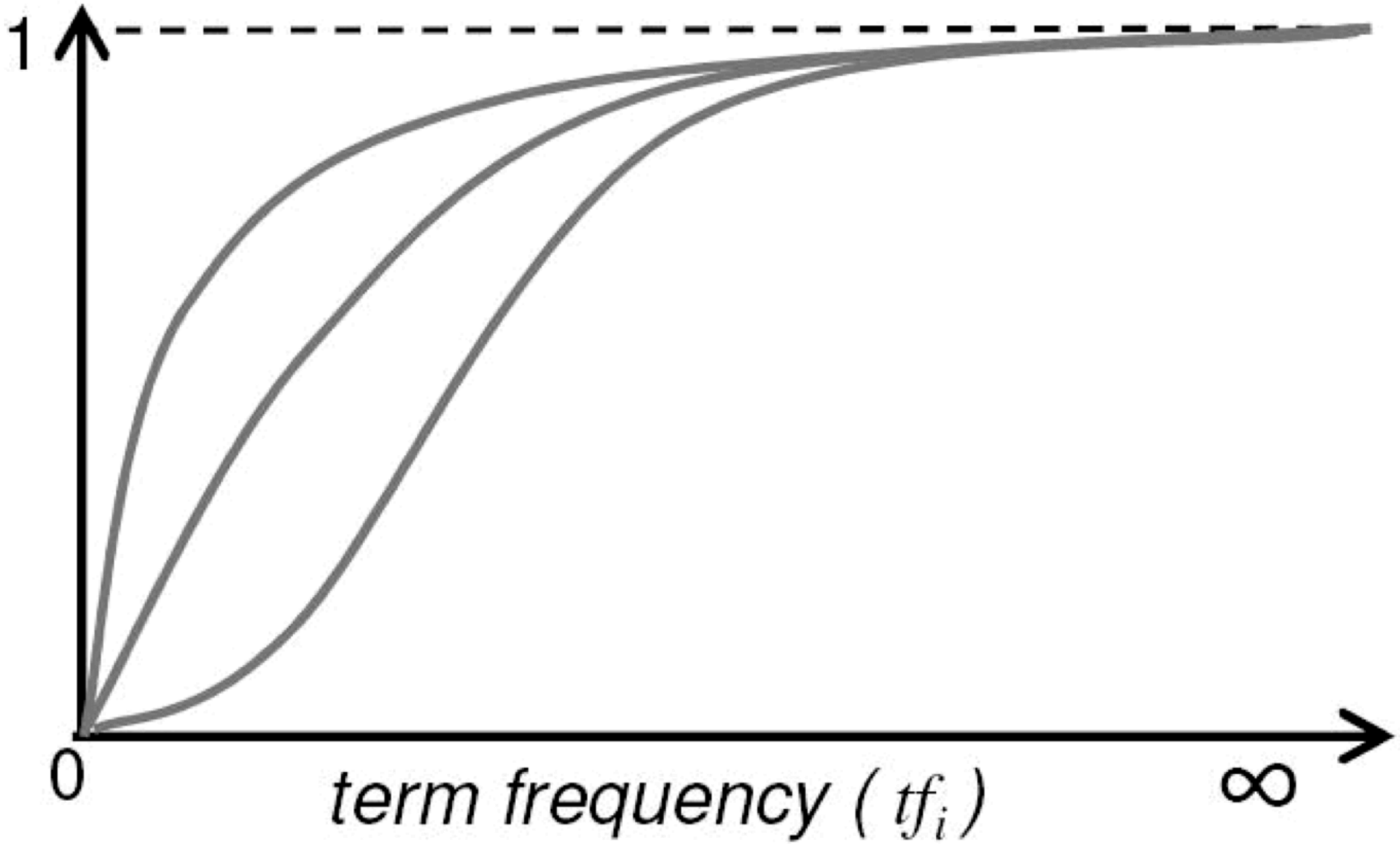
2-Poisson model

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

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

- where π is probability that document is elite for term
- but, unfortunately, we don't know π, λ, μ

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



Qualitative properties

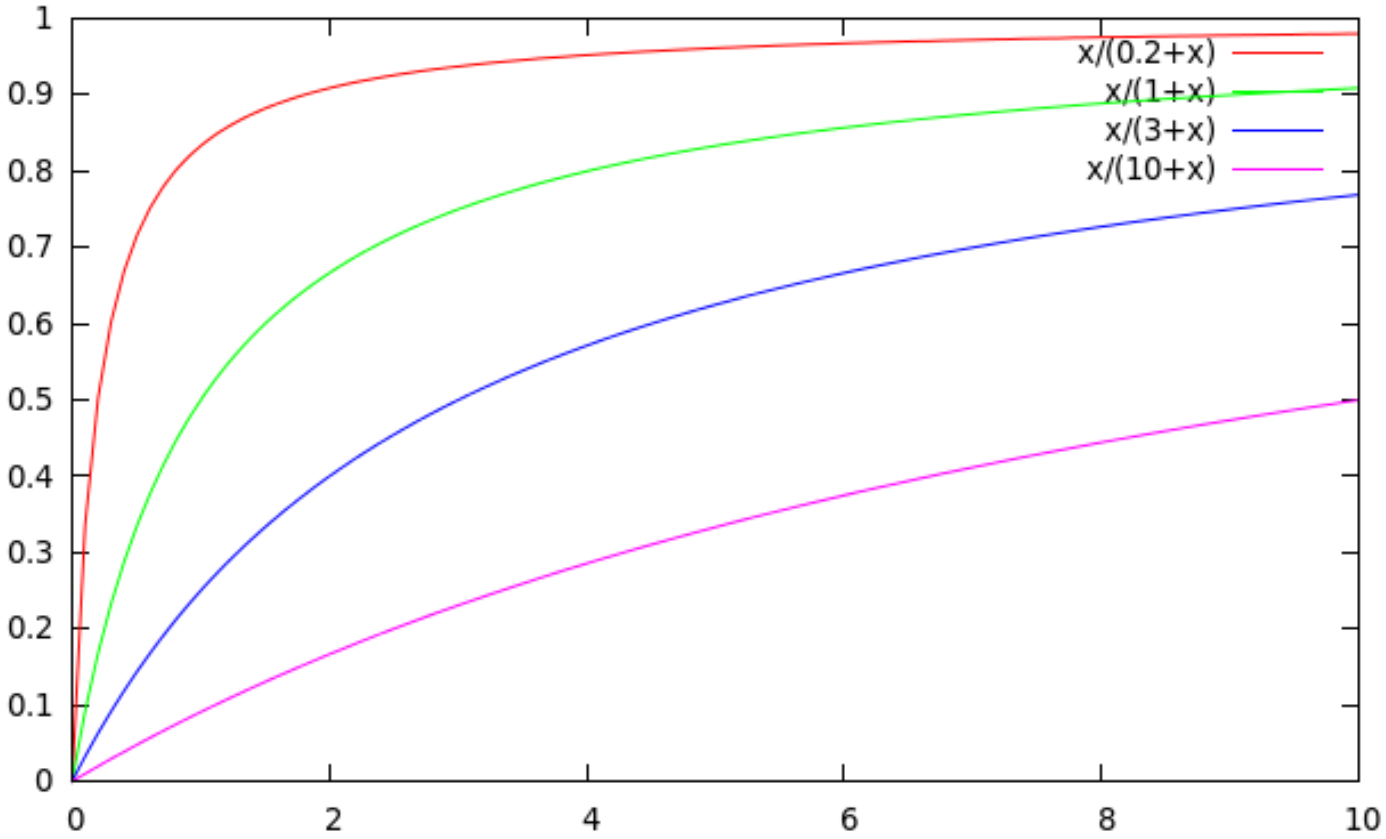
- $c_i^{elite}(0) = 0$
- $c_i^{elite}(tf_i)$ increases monotonically with tf_i
- … but asymptotically approaches a maximum value as $tf_i \rightarrow \infty$ [not true for simple scaling of tf]
- … with the asymptotic limit being c_i^{BIM} ⇐ Weight of eliteness feature

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

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

  - (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

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:

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

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

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

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

Okapi BM25

- Normalize *tf* using document length

$$tf_i' = \frac{tf_i}{B}$$

$$c_i^{BM25}(tf_i) = \log\frac{N}{df_i} \times \frac{(k_1+1)tf_i'}{k_1 + tf_i'}$$

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

- BM25 ranking function

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

Okapi BM25

$$RSV^{BM25} = \sum_{i \in q} \log \frac{N}{df_i} \cdot \frac{(k_1 + 1) tf_i}{k_1((1 - b) + b \frac{dl}{avdl}) + tf_i}$$

- k_1 controls term frequency scaling
  - $k_1 = 0$ 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 around 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: $\log_2$ tf * $\log_2$ (N/df)
  - doc1: 11 * 7 + 1 * 10 = **87**
  - doc2: 5 * 7 + 4 * 10 = **75**
- BM25: $k_1 = 2$
  - doc1: 7 * 3 + 10 * 1 = **31**
  - doc2: 7 * 2.67 + 10 * 2.4 = **42.7**