Information Retrieval

Probabilistic Information Retrieval - BM25

5. Term frequency and the VSM

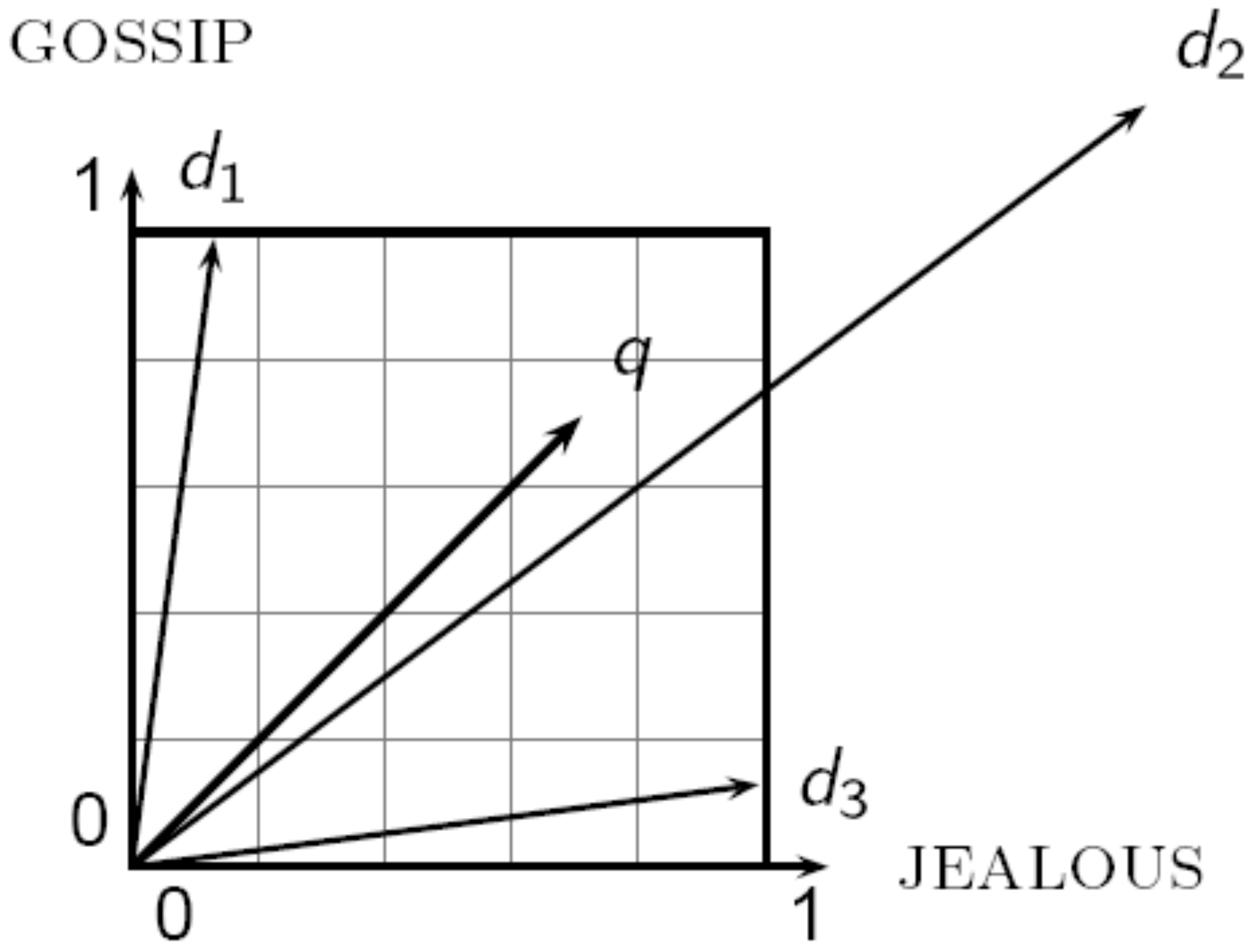
- Right in the first lecture, we said that a page should rank higher if it mentions a word more
  - Perhaps modulated by things like page length
- Why not in BIM? Much of early IR was designed for titles or abstracts, and not for modern full text search
- We now want a model with term frequency in it
- We'll mainly look at a probabilistic model (BM25)
- First, a quick summary of vector space model

Summary – vector space ranking (ch. 6)

- Represent the query as a weighted term frequency/inverse document frequency (tf-idf) vector
  - (0, 0, 0, 0, 2.3, 0, 0, 0, 1.78, 0, 0, 0, …, 0, 8.17, 0, 0)
- Represent each document as a weighted tf-idf vector
  - (1.2, 0, 3.7, 1.5, 2.0, 0, 1.3, 0, 3.7, 1.4, 0, 0, …, 3.5, 5.1, 0, 0)

- Compute the cosine similarity score for the query vector and each document vector
- Rank documents with respect to the query by score
- Return the top *K* (e.g., *K* = 10) to the user

Cosine similarity



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

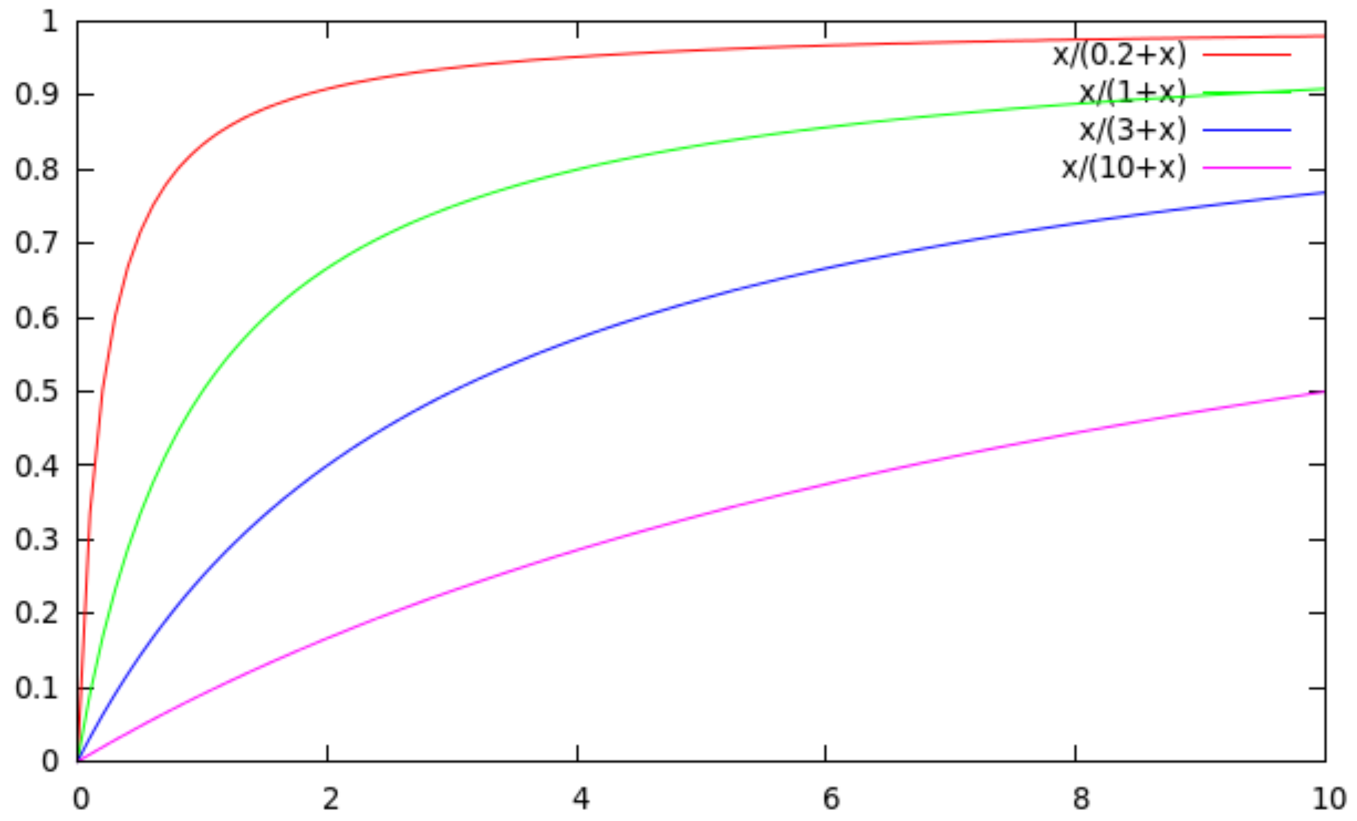
- Goal: be sensitive to term frequency and document length while not adding too many parameters
  - (Robertson and Zaragoza 2009; Spärck Jones et al. 2000)

Approximating the saturation function

- … 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:

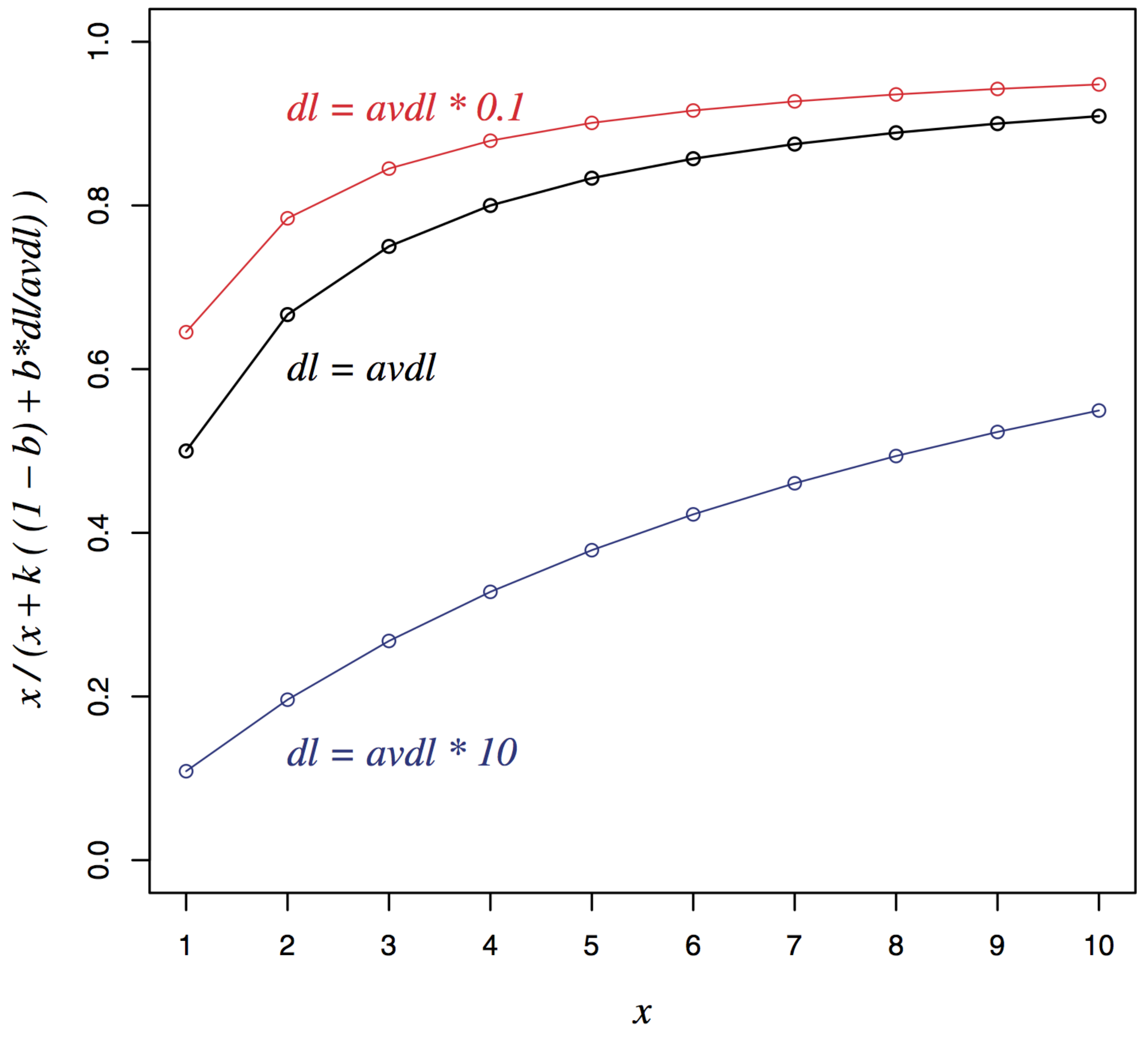
$$dl = \sum_{i \in V} tf_i$$

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

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

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

Document length normalization



Okapi BM25

- Normalize *tf* using document length

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

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

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

- 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

Assume
N = 2048
$df_{learning} = 16$
$df_{machine} = 2$

- tf-idf: $(1+ \log_2 tf) * \log_2 (N/df)$
  - 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**

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.