TF-IDF and Okapi BM25 LM, session 3

Northeastern University College of Computer and Information Science

CS6200: Information Retrieval



Binary Independence Models

- In Bayesian classification, we rank documents by their **likelihood ratios** calculated from some probabilistic model.
- The model predicts the features that a relevant or non-relevant document is likely to have.
- Our first model is a unigram language model, which independently estimates the probability of each term appearing in a relevant or non-relevant document.

Any model like this, based on independent binary features $f_i \in F$, is called a **binary independence model**.

$$\frac{P(D|R=1)}{P(D|R=0)}$$
Likelihood Batio

$$\frac{\prod_{i=1}^{|F|} P(f_i | R = 1)}{\prod_{i=1}^{|F|} P(f_i | R = 0)}$$

Binary independence Model

Ranking with B.I. Models

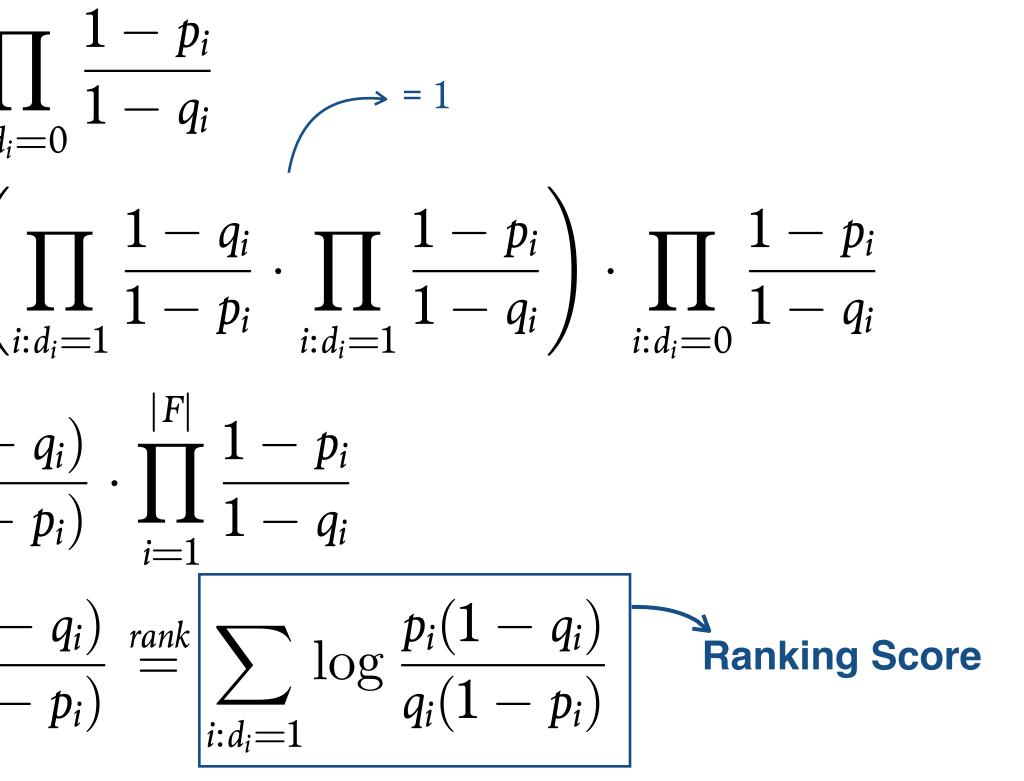
terms not found in the document. This is important for efficient queries.

Let
$$p_i := P(f_i | R = 1), q_i := P(q_i)$$

 $d_i \in \{0, 1\} := \text{ value of } f_i$
Then $\frac{P(D|R=1)}{P(D|R=0)} = \prod_{i:d_i=1} \frac{p_i}{q_i} \cdot \prod_{i:d_i=1$

$$\stackrel{rank}{=} \prod_{i:d_i=1} \frac{p_i(1-q_i(1-q_i))}{q_i(1-q_i)}$$

- Simplifying the binary independence model leads to a ranking score which allows us to ignore
 - $(f_i|R=0),$
 - in doc D.



Relationship to IDF

Under certain assumptions, the ranking score is just IDF:

- 1. All words have a fixed uniform probability of appearing in a relevant document: $p_i = 1/2$.
- 2. Most documents containing the term are non-relevant, so $q_i \approx df_i/D_i$
- 3. Most documents do not contain the term, so $D - df_i \approx D$.

 $\log \frac{p_i(1-q_i)}{q_i(1-p_i)}$ Ranking Score, $pprox \log rac{0.5(1-rac{df_i}{D})}{rac{df_i}{D}(1-0.5)}$ approximated using assumptions, $= \log \frac{1 - \frac{df_i}{D}}{\frac{df_i}{D}}$ $= \log \frac{D}{df_i} - \frac{df_i \cdot D}{df_i \cdot D}$ $= \log \frac{D - df_i}{df_i}$ $\approx \log \frac{D}{df_i}$ **becomes IDF**



Improving on IDF

It turns out that we can do better than IDF. To get there, we'll start by considering the contingency table of all combinations of d_i and R.

	<i>R</i> = 1	R = 0	Total
$d_i = 1$	r _i	$df_i - r_i$	df_i
$d_i = 0$	$R - r_i$	$D - R - df_i + r_i$	$D - df_i$
Total	R	D – R	D

$$\sum_{i:d_i=1} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} = \sum_{i:d_i=1} \log \frac{(num(d_i=1, R=1) + 0.5)/(num(d_i=0, R=1) + 0.5)}{(num(d_i=1, R=0) + 0.5)/(num(d_i=0, R=0) + 0.5)}$$
$$= \sum_{i:d_i=1} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(df_i - r_i + 0.5)/(D - R - df_i + r_i + 0.5)}$$

We will estimate p_i and q_i using this table and a technique called "add- α smoothing," with α =0.5.

$$p_i = \frac{r_i + 0.5}{R+1}; q_i = \frac{df_i - r_i + 0.5}{D-R+1}$$

This leads to a slightly different ranking score:



Is it better?

Let's unpack this formula to understand it better.

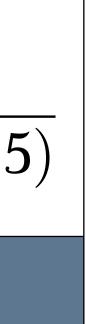
The numerator is a ratio of counts of relevant documents the term does and does not appear in. It's a likelihood ratio giving the amount of "evidence of relevance" the term provides.

The denominator is the same ratio, for nonrelevant documents. It gives the amount of "evidence of non-relevance" for the term.

If the term is in many documents, but most of them are relevant, it doesn't discount the term as IDF would.

$$\log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(df_i - r_i + 0.5)/(D - R - df_i + r_i + 0.5)}$$

A better IDF?



Okapi BM25

Okapi BM25 is one of the strongest "simple" scoring functions, and has proven a useful baseline for experiments and feature for ranking.

It combines:

- The IDF-like ranking score from the last slide,
- the document term frequency tf_{i,d}, normalized by the ratio of the document's length dl to the average length avg(dl), and
- the query term frequency $tf_{i,q}$.

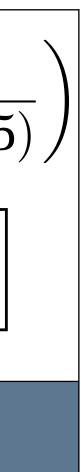
$$\sum_{i:d_i=q_i=1} \left[\log\left(\frac{(r_i+0.5)/(R-r_i+0.5)}{(df_i-r_i+0.5)/(D-R-df_i+r_i+0.5)} + \frac{tf_{i,d}+k_1 \cdot tf_{i,d}}{tf_{i,d}+k_1((1-b)+b \cdot \frac{dl}{avg(dl)})} \cdot \frac{tf_{i,q}+k_2 \cdot tf_{i,q}}{tf_{i,q}+k_2} \right]$$

Okapi BM25

 k_1 , k_2 , and b are empirically-set parameters. Typical values at TREC are:

$$k_1 = 1.2$$

 $0 \le k_2 \le 1000$
 $b = 0.75$



Example: BM25

Example query: "president lincoln"

- $tf_{president,q} = tf_{lincoln,q} = 1$
- No relevance information: $R = r_i = 0$
- "president" is in 40,000 documents in the collection: $df_{president} = 40,000$
- "lincoln" is in 300 documents in the collection: $df_{lincoln} = 300$
- The document length is 90% of the average length: *dl/avg(dl)* = 0.9
- We pick $k_1 = 1.2$, $k_2 = 100$, b = 0.75

tf _{president,d}	tflincoln,d	BM25
15	25	20.66
15	1	12.74
15	0	5.00
1	25	18.2
0	25	15.66

The low df term plays a bigger role.

Wrapping Up

Binary Independence Models are a principled, general way to combine evidence from many binary features (not just unigrams!)

anchor text, into account.

topics of machine learning.

- The version of BM25 shown here is one of many in a family of scoring functions. Modern alternatives can take additional evidence, such as

Next, we'll generalize what we've learned so far into the fundamental