COM S/INFO 630: Representing and Accessing [Textual] Digital Information
Lecturer: Lillian Lee
Lecture 3: 2 February 2006
Scribes: Siavash Dejgosha (sd82) and Ricardo Hu (rh238)

Pivoted Length Normalization

I. Summary

Document length normalization schemes attempt to eliminate the advantage that long documents have over the shorter documents under a certain scoring scheme. However, Singhal et al's paper suggests that normalization over-penalizes long documents when compared to the actual document distribution. In particular, retrieval using cosine normalization tends to retrieve shorter documents more often and longer documents less often than the distribution of lengths of relevant documents tends to suggest.

Pivoted Document Length Normalization (PDLN) modifies the standard cosine normalization (CDLN) scheme to overcome this situation. PDLN seeks to transform CDLN to better match the retrieval and true relevance probabilities of a document with respect to length.

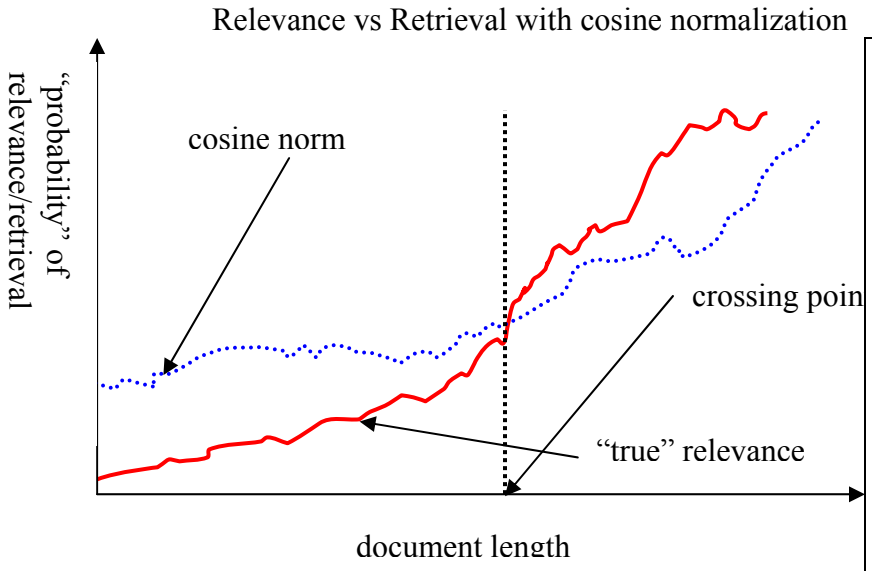
Notation
d: a document in the corpus
tf: term frequency
idf: inverse document frequency
w: weight of a particular term in the vocabulary

w = tf * idf

$$norm(d) = \sqrt{w_1^2 + w_2^2 + ... + w_m^2}$$

II. Review

Consider the distribution of relevant documents according to document length. We can construct a curve of the "probability" vs. length graph of the relevant documents in a corpus versus the presumed relevant documents retrieved by a standard cosine normalization scheme:



Graph Construction

1. Sort the set of all documents in corpus by length.
2. Create bins, B_i, with 1,000 documents each in sorted order.
3. The length of a particular bin is the median length of all documents in that bin: $len(B_i)$ = median(len(d), for all d ε B_i)
4. Each bin is assigned a probability by the ratio: relevant doc's in bin / total number of relevant doc's in the corpus: $prob(B_i) = P(d \varepsilon B \mid d \text{ is relevant})$

Observations

I. Neither curve is linear despite cosine normalization being a linear function of the document L_2 norm. This is because the underlying relevance distribution is apparently non-linear.

II. The shape of the cosine normalization curve does not match exactly that of the relevance curve. The shape of the curve is determined by the IR system's representation and ranking system of documents which is not necessarily a completely accurate function of the document length.
   a. For lengths < crossing point, the cosine curve is above the relevance curve. That is, short documents are more likely to be retrieved than their "true" likelihood of relevance.
   b. For lengths > crossing point, the cosine curve is below the relevance curve. That is, long documents are less likely to be retrieved than their "true" likelihood of relevance.

III. Reliability of statistics over different lengths:
Suppose there are only 2 documents at length 13,000 and 5,000 documents at length 13,001. The statistics for length 13,001 are much more reliable than the statistics at 13,000. We adjust for this by creating 1000 document bins.

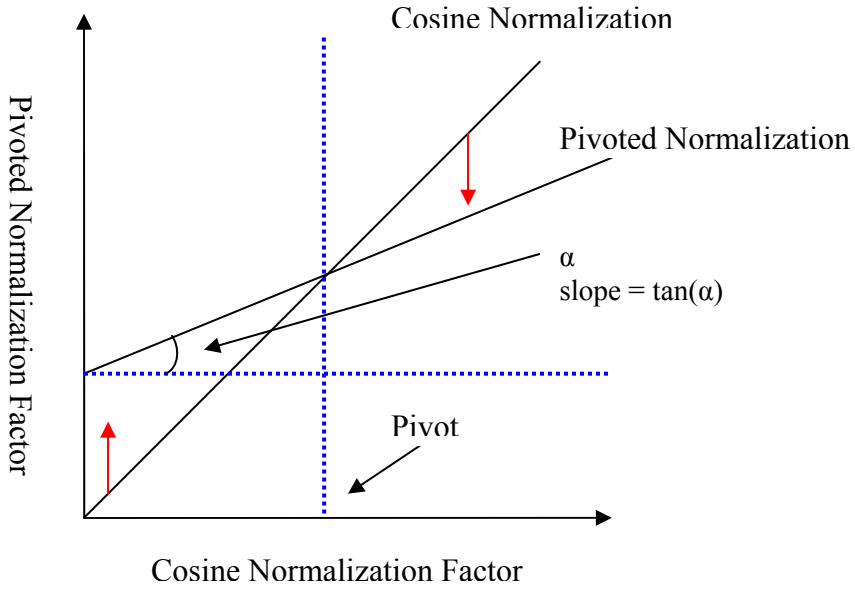
III. Derivation of Pivoted Length Normalization[1,2]

Motivation:
From observation II above, norm(d) should be lower for longer documents because longer documents are being penalized too much by cosine normalization.

Solution:
Map the old normalization to a new normalization, norm'(d) with an appropriate value for longer documents. We use a linear mapping:

[1] norm'(d) = m' norm(d) + b', where m' and b' are parameters

This equation defines pivoted document length normalization in terms of cosine normalization.



The pivot, p, is the norm of the document length corresponding to the crossing point in the previous graph. At this point, norm'(d_p) = norm(d_p).
Note that there is some leeway in specifying d_p. It is not clear if they would have used the document that has the median length in the bin that represents the crossing-point, or if they would have used all the relevant documents in the bin represented by the "crossing-point" to calculate the d_p.
We do not need an exact specification of d_p because the graph above is not plotted from actual data. Rather, it is intended to show how the "tilting" transformation works. We can just consider d_p to be a representative document of the length of approximately equal to the value of the crossing point.

---

[1] *Singhal, Salton, Mitra and Buckley. "Document length normalization." Information Processing & Management. vol 32 #5, Sep 1996, p619-33*
[2] *Singhal, Buckley, Mitra. "Pivoted Document Length Normalization." ACM SIGIR. 1996*

At the pivot, norm'(d_p) = norm(d_p) = p, and using [1] above we get

normd'(d_p) = m' norm(d_p) + b', where m' = tan(alpha)

→ p = m' p + b'

[2] b' = p(1-m')

Using [1] and [2], we can write

[3] norm'(d) = m' norm(d) + p(1-m')

Removing one parameter

Note that for any positive constant α with respect to a document d:

$$score(d) \overset{rank}{=} \alpha \bullet score(d)$$

since the constant term does not matter for ranking. We can use this equivalence to define a normalization function to simplify the constant term:

$$norm''(d) = \left(\frac{1}{p(1-m')} \right) norm'(d)$$

[4] $$norm''(d) = \frac{m'}{p(1-m')} norm(d) + 1$$

Let's try to fix one of the parameters and optimize the other one. How about a good p?

Let p = $\overline{norm(d)} = \overline{norm}$, the average normalization over all documents

We claim there exists an m'' such that $\frac{m''}{\overline{norm}(1-m'')} = \frac{m'}{p(1-m')}$.

Solving for that m'' we arrive at:

[5] $$m'' = \frac{\frac{m' \times \overline{norm}}{(1-m')p}}{\left(1 + \frac{\overline{norm} \times m'}{(1-m')p}\right)}$$

Now we define another new norm'''(d) using [4] and [5]

$$norm'''(d) = \frac{m''}{(1-m'')\overline{norm}} norm(d) + 1$$

Multiply by (1-m'') to define norm^{iv}(d) which is equivalent for ranking purposes:

[6] $$norm^{iv}(d) = m'' \frac{norm(d)}{\overline{norm}} + (1-m'')$$

Notice that for a document of average length, $norm^{iv}(d) = \overline{norm}$ so that norm^{iv}(d) = 1. That is, a document of average length is already of "appropriate" length and does not need to be scaled under this scheme.

Assuming the linear correction, we have reduced the two-parameter search problem to a search for just one parameter. All the newly defined norm's are equivalent under ranking since we have only multiplied the functions by a constant in d.

Results
Empirically, applying this new normalization resulted in:

1. relevance and retrieved graphs becoming much more similar
2. 6-12% improvement in retrieval precision
3. relatively stable optimal (m'') values over 6 corpora

IV. Document Relevancy[3]

In the motivation for PDLN, we assumed that we had the distribution of "truly" relevant documents. But how valid is this assumption?

In practice, the set of valid documents in TREC is determined by humans who (1) judge the top 100 results from several different information retrieval systems for a particular query, and (2) pooling those results together. With this system, it is possible that there are relevant documents that are not in the top 100 results and have therefore not been marked. If a significant number of relevant documents are not marked or if there are inherent biases in the retrieval methods (e.g. over-retrieving short documents), these artifacts could skew the observed probability distribution. If we based our motivation for PDLN in that skewed distribution, then PDLN would be fundamentally flawed.

Motivation
It is impractical to have humans judge every document in a corpus of thousands or millions.

Goal
How can we count the number of truly relevant documents in a corpus without judging them all manually?

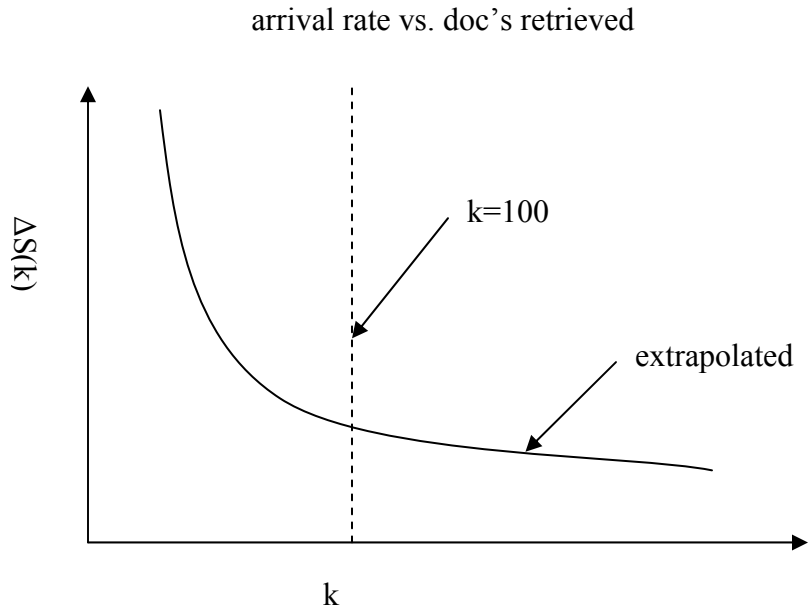
Idea: "Arrival Rates"
Instead of considering the static set of the top k results from a particular retrieval method, let k be a free variable. Estimate the number of relevant documents from the arrival rate of newly relevant documents with respect to k.

- "Pool" the top k documents from different retrieval methods by taking the union of the retrieved sets from different methods and retaining only the relevant documents. Call this R_k.
- Compare this R_k with the pooled relevant (k+1) document set, R_{k+1}.
- Define the "arrival rate" $\Delta R(k) = R_{k+1} - R_k$ where k is the "pool depth."
- We expect $\Delta R(k)$ to be a decreasing function of k if (1) there are a finite number of relevant documents and (2) our retrieval methods retrieve relevant documents.
- Notice that if $\Delta R(k)$ can be fitted with a smooth function delR(k) that converges to zero, we can calculate the total number of relevant documents as:
    - $reldocs = \int_1^{\infty} delR(k)dk$

[3] *"How Reliable are the Results of Large-Scale Information Retrieval Experiments" Zobel. SIGIR 1998*

Results

It turns out that the arrival rate does fit a smooth converging function very well. TREC results only extend to the top 100 relevant documents for a given query so results beyond k=100 must be extrapolated. Since the fit is so good for k < 100, extrapolation seems justifiable in this case.



In practice, we cut the upper limit of the integral earlier than infinity because we have less confidence in the fit for large k: for $k_{max} = 200$, $\int_1^{200} delR(k)dk = 6707$, for $k_{max} = 500$, $\int_1^{500} delR(k)dk = 9358$. However, the TREC results (which end at k=100) say that there are only 5040 relevant documents. Thus many relevant documents are not being marked and we are underestimating recall.