CS 674/INFO 630: Advanced Language Technologies Fall 2007

Lecture 6 — September 13, 2007

Prof. Lillian Lee

Scribes: Nam Nguyen
Myle Ott

1 Recall

For any given query q and document d in a corpus C, we use the following scoring function to rank documents:

$$P(R_q = y \mid \vec{A} = \vec{d}) \stackrel{rank}{=} \prod_{\substack{j:q[j]\neq 0,\\ d[j]\neq 0}} \left(\frac{P(A_j = d[j] \mid R_q = y)}{P(A_j = d[j])} \times \frac{P(A_j = 0)}{P(A_j = 0 \mid R_q = y)} \right)$$

Note that the above scoring function was reached by applying the following strategies. See previous lecture for details.

- Bayes flip: condition on (relatively) more information
- reduce the number of unknowns (discard document-independent quantities)
- desparsify by "factoring" (linked dependence)

In section 2 we describe the binary attribute assumption made by Robertson and Spärck Jones (RSJ). In sections 3, 4 and 5 we will present several models for the occurrence probability of query terms in relevant documents.

2 Binary attribute assumption

In [3], RSJ propose binarization of the attribute vector $\vec{A}$. This allows us to rewrite our scoring function as follows:

$$P(R_q = y \mid \vec{A} = \vec{d}) \stackrel{rank}{=} \prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{P(A_j = 1 \mid R_q = y)}{P(A_j = 1)} \times \frac{1 - P(A_j = 1)}{1 - P(A_j = 1 \mid R_q = y)} \right)$$

3 Constant model

Unfortunately, our scoring function still has two unknowns per attribute, namely $P(A_j = 1)$ and $P(A_j = 1 \mid R_q = y)$. In [1], Croft and Harper (CH) propose solutions for both unknowns. For the unconditioned attribute occurrence probability:

$$\hat{P}(A_j = 1) = \frac{n_j}{N}$$

where n_j is the number of documents in the corpus for which attribute j occurs and N is the total number of documents in the corpus.[1]

For the attribute occurence probability of query terms in relevant documents, CH propose:

$$\hat{P}(A_j = 1 \mid R_q = y) = \alpha_{d,q,j}$$

where $\alpha_{d,q,j} \in [0, 1]$ is the *same* constant for all documents d and queries q where the j^{th} attribute is exhibited by both.

Question

After making the above substitutions, what does the scoring function look like? How does this scoring function relate to the Vector Space Model (VSM) discussed in previous lectures?

Answer

Given our scoring function from section 2, we can make substitutions for $P(A_j = 1)$ and $P(A_j = 1 \mid R_q = y)$ as follows:

$$\begin{aligned}
& \prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{P(A_j = 1 \mid R_q = y)}{P(A_j = 1)} \times \frac{1 - P(A_j = 1)}{1 - P(A_j = 1 \mid R_q = y)} \right) \\
= & \prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{\alpha_{d,q,j}}{\frac{n_j}{N}} \times \frac{\frac{N - n_j}{N}}{1 - \alpha_{d,q,j}} \right) \\
= & \prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{N}{n_j} - 1 \right) \times \left(\frac{\alpha_{d,q,j}}{1 - \alpha_{d,q,j}} \right)
\end{aligned}$$

For simplicity, $\alpha_{d,q,j} = 0.5$ is often used, giving the following scoring function:

$$\prod_{j:q[j]=1, d[j]=1} \left(\frac{N}{n_j} - 1 \right)$$

Notice that the quantity $\left(\frac{N}{n_j} - 1 \right)$ looks like inverse document frequency (IDF). In fact, we can cleverly rewrite the above scoring function to derive a matching function similar to ones used in the

[1] Also note that $\hat{P}(A_j = 1) \approx \hat{P}(A_j = 1 \mid R_q = n)$ since most documents in the corpus are probably not relevant to our given query. We make this observation because CH were working with RSJ's original derivation in which a different initial scoring function, $\log \left(P(R_q = y \mid \vec{A} = \vec{d}) \; / \; P(R_q = n \mid \vec{A} = \vec{d}) \right)$, was employed. We chose our scoring function, $P(R_q = y \mid \vec{A} = \vec{d})$, because it avoids the need for this extra approximation.

vector space model:

$$\prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{N}{n_j} - 1\right) \stackrel{rank}{=} \log\left(\prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{N}{n_j} - 1\right)\right)$$

$$= \sum_{\substack{j:q[j]=1,\\ d[j]=1}} \log\left(\tfrac{N}{n_j} - 1\right)$$

$$= \sum_{j=1}^{m} \left(q[j] \times d[j] \times \log\left(\tfrac{N}{n_j} - 1\right)\right)$$

Note that we have the binary analog of term frequency (TF) appearing—namely, the binary-valued $d[j]$. Hence, we have what can be called a TF-IDF–like term weight. This derivation is often referred to as the theoretical motivation for IDF.

4 Hyperbolic model

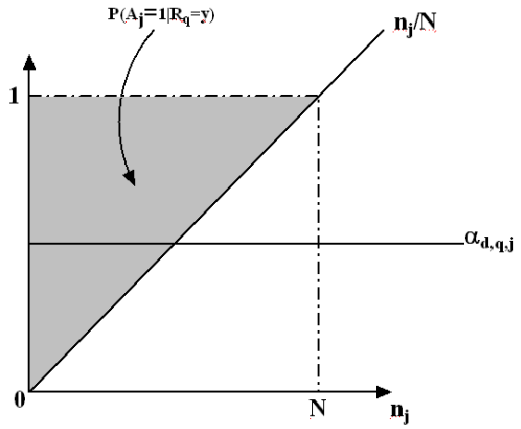
In [4], Robertson and Walker (RW) criticize the $\hat{P}(A_j = 1 \mid R_q = y) = \alpha_{d,q,j}$ assumption. They observe that $\log\left(\frac{N}{n_j} - 1\right)$ is negative when $\frac{N}{n_j} - 1 < 1$. Thus, for attributes that are shared by a document d and query q, we end up with a negative score in situations where $n_j > \frac{N}{2}$.

In general, RW observe that the problem of negative weights (when the log version of the matching function is used) can be avoided if our estimates obey the following condition:

$$\frac{\hat{P}(A_j = 1 \mid R_q = y)}{\hat{P}(A_j = 1)} \geq 1$$

$$\Rightarrow \quad \frac{\hat{P}(A_j = 1 \mid R_q = y)}{\frac{n_j}{N}} \geq 1$$

$$\Rightarrow \quad \hat{P}(A_j = 1 \mid R_q = y) \geq \frac{n_j}{N}$$



Note that the preceeding condition implies that:

$$\frac{1-\hat{P}(A_j=1)}{1-\hat{P}(A_j=1 \mid R_q=y)} = \frac{\hat{P}(A_j=0)}{\hat{P}(A_j=0 \mid R_q=y)} \geq 1$$

Thus, the product of the above two terms (our term weight) is also greater than or equal to 1.

Question

What are some possible choices for $\hat{P}(A_j=1 \mid R_q=y)$?

Answer

Recall that we want our estimation to never fall below $\frac{n_j}{N}$. Some possibilities are:

- Try $\hat{P}(A_j=1 \mid R_q=y)=1$. But this is clearly unrealistic; also after substitution,

  $$\frac{1-P(A_j=1)}{1-P(A_j=1 \mid R_q=y)}$$

  results in a division by zero.

- Try $\hat{P}(A_j=1 \mid R_q=y)=\frac{n_j}{N}$. Intuitively, if we want to design as simple a model as possible, noting that the occurrence probability of query terms in relevant documents must be greater than or equal to the occurrence probability of query terms in the corpus, estimating $\hat{P}(A_j=1 \mid R_q=y)=\hat{P}(A_j=1)$ might make sense. However, after substitution we have:

  $$\prod_{\substack{j:q[j]=1,\\ d[j]=1}} 1 = 1$$

  thereby assigning an equal score to all documents that share any attribute j with the query q, which is clearly undesireable for obtaining good ranks.

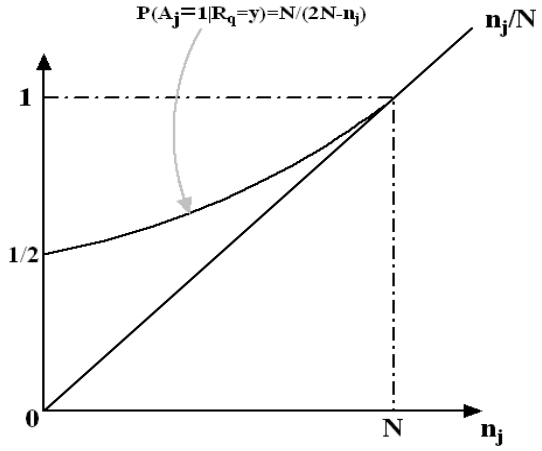
- RW claim that finding a straight line solution is "intractable." They instead propose:

  $$\hat{P}(A_j=1 \mid R_q=y) = \frac{1}{1+\frac{N-n_j}{N}} = \frac{N}{2N-n_j}$$

  In addition to satisfying our conditions, this solution has the nice benefit of simplifying the ranking function to an IDF:

  $$\begin{aligned} P(R_q=y \mid \vec{A}=\vec{d}) &= \prod_{\substack{j:q[j]=1,\\ d[j]=1}} \left(\frac{\frac{N}{2N-n_j}}{\frac{n_j}{N}} \times \frac{\frac{N-n_j}{N}}{\frac{N-n_j}{2N-n_j}} \right) \\ &= \prod_{\substack{j:q[j]=1,\\ d[j]=1}} \frac{N}{n_j} \qquad \text{(IDF)} \end{aligned}$$

  But the RW solution offers no clear justification for using this particular function.



5 Lift model

In [2], Lee questions RW's original claim that a "straight line" model is "intractable." Lee proceeds by claiming that in general, and as is true of RW's estimate, $\hat{P}(A_j = 1 \mid R_q = y)$ should in fact be strictly greater than $\hat{P}(A_j = 1) = \frac{n_j}{N}$, i.e., given that a document is relevant to a query, the occurrence probability of query terms in relevant documents should be greater than the occurrence probability of query terms in the corpus. Given our previous constraints, Lee proposes the following:

$$\hat{P}(A_j = 1 \mid R_q = y) = \frac{n_j + L}{N + L}$$

where L in the numerator corresponds to the "lift" given to documents known to be relevant (the L in the denominator is to ensure that the estimate is bounded above by 1). After substitution we have:

$$P(R_q = y \mid \vec{A} = \vec{d}) = \prod_{\substack{j:q[j]=1, \\ d[j]=1}} \left(1 + \frac{L}{n_j}\right)$$

which is IDF-like when N is substituted for L.

Question

Suppose that L is a function of n_j. Using Lee's lift model, provide a justification for RW's solution.

Answer

We claim that the RW solution can be explained by Lee's model with $L = N - n_j$:

$$\begin{aligned}
\hat{P}(A_j = 1 \mid R_q = y) &= \frac{n_j + L}{N + L} \\
&= \frac{n_j + (N - n_j)}{N + (N - n_j)} \\
&= \frac{N}{2N - n_j}
\end{aligned}$$

Namely, for any shared attribute, the lift given to a document known to be relevant to a query is equal to the number of documents that do not possess that attribute. The intuitive justification for

this is that query terms that are rare in the corpus are going to be the most helpful in discriminating relevant from non-relevant documents (since few documents will have these rare terms). Thus, the lift should be proportional to the rarity of the attribute in the corpus. This provides a good intuitive justification for the RW solution.

References

[1] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 1997.

[2] L. Lee. IDF revisited: A simple new derivation within the Robertson-Spärck Jones probabilistic model. In *Proceedings of SIGIR*, 2007.

[3] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[4] S. E. Robertson and S. Walker. On relevance weights with little relevance information. In *Proceedings of SIGIR*, 1997.