CS630 Lecture 9: End of Probabilistic Retrieval, Introduction to Relevance Feedback

Lecture by Lillian Lee
Scribed by Randy Au, Nick Gerner, Blazej Kot

February 23, 2006

In this lecture, we finish treating the basic retrieval paradigms, and move onto introducing the relevance feedback paradigm.

1 Meta-lessons

During the past several lectures, we have been trying to develop the following skills and important meta-tools to do mathematical modeling. These are tools with broad application within and outside of information retrieval.

- Bayes flip : Apply Bayes Rule when appropriate.
- (Re)-factorizations : Consider independent random variables and decomposing into products thereof. This rephrases the problem to hopefully eliminate terms to appropriately simplify a derivation.
- Insert R.V.'s : Capture the system you are trying to model by inserting (mathematically appropriately) random variables
- Ground models : Incorporate semantic knowledge and scenarios to yield intuition and take reasonable next steps (sometimes making appropriate assumptions).
- Aesthetic sense : Many choices during a derivation may seem arbitrary, but must be taken with care to move towards a desirable result. These choices rely on your intuition and ”aesthetic sense”
- Check/challenge preconceptions : Important, interesting and valuable new results are often achieved by questioning previous assumptions and models (consider pivoted length normalization or the language modeling approach to IR ranking).
- Math formalisms : By formalizing a model mathematically, assumptions and preconceptions are made explicit, making the applicability of a technique clear and suggesting directions for future work. Also consider taking advantage of previous formalisms (see challenging preconceptions above)

2 Simple re-derivation of a probabilistic scoring function

Recall we have the topic models

$$T_D \rightarrow D$$

and independent query models

$$T_Q \rightarrow Q$$

We can now define what we mean by $R = y$ to be that the topic model that generates a document is the same as the topic model which generates the query.

Our scoring function now (relying on the above intuition of $R = y$) is:

$$P(R = y|D = d, Q = q)$$

Again relevance status would seem to be determined by d and q and so apparently there is no "room" for probabilities. To address this issue (which we have faced before in probabilistic models of information retrieval) we consider the contents of documents rather than the documents themselves. Here, two documents might have been generated by different topic models but have the same content. This gives us a binning intuition which we considered in the RSJ model to give a random choice. We can rewrite our scoring function as follows:

$$\sum_{t,t'} P(R = y, T_Q = t, T_D = t'|D = d, Q = q)$$

Now we want to move Q to the LHS, so use Bayes flip:

$$\frac{\sum_{t,t'} P(Q = q|R = y, T_Q = t, T_D = t', D = d)P(R = y, T_Q = t, T_D = t'|D = d)}{P(Q = q|D = d)}$$

Note that the $P(R = y, T_Q = t, T_D = t'|D = d)$ term is zero if $t \neq t'$ since our interpretation of relevance is that these two topic models must be the same (equal the same t). Also, by the independence of the query and document the denominator term becomes $P(Q = q)$, which is constant for all documents, so it can be ignored under rank.

This now gives:

$$\sum_{t} P(Q = q|R = y, T_Q = t, T_D = t, D = d)P(R = y, T_Q = t, T_D = t|D = d)$$

Firstly, notice $R = y$ is in both cases implied because of $T_Q = t, T_D = t$ - that is our very definition of relevance. Therefore, we can drop this term from both parts of the equation.

Secondly, notice $Q = q$ does not depend on the document choice at all - therefore we can remove $T_D = t$ and $D = d$.

Together, this now gives:

$$\sum_t P(Q = q|T_Q = t)P(T_Q = t, T_D = t|D = d)$$

Now, notice that in the right-most term $T_Q = t$ is independent of $D = d$. However, we have this annoying term $T_D = t$. Note that from basic probability, this term can be re-written in a split form:

$$P(T_Q = t|T_D = t, D = d)P(T_D = t|D = d)$$

Plugging this back into our scoring function gives:

$$\sum_t P(Q = q|T_Q = t)P(T_Q = t)P(T_D = t|D = d)$$

In practice, we assume $t^*(d)$ is an MLE/MAP for the topic model of d and that $P(T_D = t^*(d)|D = d) = 1$ removing the necessity to sum over all possible topic models for d. This gives:

$$P(Q = q|T_Q = t^*(d))P(T_Q = t^*(d))$$

Note that this model has a problem in that it looks for exact matching between the the topic model for the document and the topic model for the query. We might address this with some notion of scoring by the distance between the models to allow for partial matching.

3 Relevance Feedback (prelude)

3.1 Introduction to Relevance Feedback

In our coverage of classic probabilistic information retrieval we had to overcome the issue of missing relevance information: we always needed some way to gather statistics conditioned on the (hidden) value of R. Now we will consider the case when some of this relevance information is available. Vector space models have no a priori method of incorporating this information and rely on further ad hoc methods. The probabilistic retrieval framework depends on some probabilistic scoring function that can (and does) include $R = y$ explicitly.

3.2 Relevance Feedback Intuition

For a given query we assume that some relevance-labeled documents are available. This may seem counter-intuitive (this assumes that the user chooses relevant documents rather than the system!) Here are some possible scenarios in which the availability of such information is plausible:

- The user is very interested in recall for novelty verification (paper publishing or plagiarism detection)

- Mitigate sample bias: to use a returned set of relevant documents as a random sample we must account for ranking bias
- If a system is giving no relevant documents, but some partially relevant documents, feedback can improve results and so the user might be willing to provide it. This may arise if the system was unable to interpret the query (because of user or system error is left to the reader's judgement)

4 Questions

The scoring function derived in section 2 results in the same function as derived in the previous lecture. Specifically we can use the same multinomial estimation model for document and query generation. Recall an important parameter of this model: μ.

Let's consider how these two derivations differ qualitatively by exploring this parameter.

Imagine a world in which we have the following corpus. We've included topic models which can generate the documents; however, notice how we have not been concrete in defining the topic models. They are, for your reference in parameter decisions and to help conceptualize the corpus, as follows:

$$t_1 \text{ embodies the idea of "}\textit{dogs hate cats}\text{"}$$

which generates

$$d_1 = \text{"}\textit{dogs chase cats}\text{"}$$

$$d_2 = \text{"}cats\ fear\ dogs\text{"}$$

$$d_3 = \text{"}\textit{dogs hate cats}\text{"}$$

and

$$t_2 \text{ embodies the idea of "}\textit{dogs love cats}\text{"}$$

which generates

$$d_4 = \text{"}\textit{dogs chase cats}\text{"}$$

$$d_5 = \text{"}\textit{dogs like cats}\text{"}$$

$$d_6 = \text{"}\textit{cats hate dogs}\text{"}$$

We now have a case of 1 unique document per topic, and 2 shared documents between the two that are semantically different but the same when treated as a bag of words.

The entire vocabulary space is 6 words: $\mathbf{V} = \textit{dogs},\ \textit{cats},\ \textit{chase},\ fear,\ \textit{like},\ \textit{hate}$

4.1 Question 1

In your own words describe the function of the parameter μ.

(a) Quantitatively (but in words), for the scoring function used in both derivations, compare the ranking function as μ varies from zero.

(b) Qualitatively, what is the meaning of this parameter for this derivation with respect to topic model document generation? Perhaps reference the corpus in question (above).

4.2 Question 2

Now that you have an idea of the function of the parameter μ, what might be one good choice for this parameter? Provide a value and a short justification with respect to the corpus in question (above).

4.3 Question 3

Now, suppose some user out there has the information need

$$t_3 \text{ embodies the idea of ``}\textit{what do dogs hate}\text{''}$$

which generates the query

$$q = \text{``}\textit{dogs hate}\text{''}$$

Calculate the rank of each document using your choice of μ from before.

5 Answer

5.1 Question 1

(a) μ is a smoothing parameter. Conceptually it changes "emphasis" from the relative-frequency distribution in the document to the observed relative-frequency distribution in the corpus. It can also be thought of as adding artificial counts (distributed with respect to the corpus counts.) At zero this parameter only allows the model to generate other documents containing terms from the vocabulary induced by the document in question.

(b) In the case of this derivation (with respect to topic model document generation), μ helps to compensate for the fact that each document under each topic model only contains three terms, however other related documents contain other terms.

5.2 Question 2

In choosing an appropriate μ we should determine how much we want to emphasize the observed relative-frequency distribution in the corpus vs that in the document (see answer 1). One approach considers how disjoint the contents of each document under each model are. Notice that under both topics each document contains two terms in common with each other document, but one term which neither other document contains. The number of unique terms occurring in any specific document is three. However the number of unique terms occurring in all related documents is five. This might lead to a (somewhat arbitrary) decision to emphasize the observed document frequency with $\mu = 2$ in order to account for these other terms. However, it can be argued that this will move too much emphasis on the corpus count distribution for unseen terms because the topic only generated documents containing five of the six vocabulary terms and the unseen counts we add will be assigned to all three unseen vocabulary terms. $\mu < 2$ but still close to 2 may be a better choice, such as $\mu = 1$.

5.3 Question 3

Calculations Using the model

$$\prod_j \left(\frac{tf_j(d) + \mu tf_j(c)/|c|}{|d| + \mu} \right)^{tf_j(q)} , \mu = 1$$

we get:

$$P_{d_1}(q) = \left(\frac{1 + 6/18}{4}\right)^1 * \left(\frac{1 + 2/18}{4}\right)^0 * \left(\frac{1 + 6/18}{4}\right)^0 * \left(\frac{2/18}{4}\right)^1 * \left(\frac{1/18}{4}\right)^0 * \left(\frac{1/18}{4}\right)^0$$

Simplifying,

$$= \frac{1}{3}\frac{1}{36}$$

In the end, we obtain:

$$P_{d_1}(q) = P_{d_2}(q) = P_{d_4}(q) = P_{d_5}(q) = \frac{1}{3}\frac{1}{36}$$

$$P_{d_3}(q) = P_{d_6}(q) = \frac{1}{3}\frac{10}{36}$$

Which means that the score for the documents which match both query terms is ten times higher than those that match just one.