CS 630 Notes: Lecture 9
Lecturer: Lillian Lee

Notes by Matt Connolly and Danish Mujeeb

February 23rd, 2006

1 Meta-Note on Modeling Tools

Thus far in the course we've been using some important tools for simple mathematical modeling. These are worth keeping in your "toolbox" as useful problem-solving techniques:

- Bayes flip
- (re)factorization
- Inserting random variables (as long as it's done appropriately)
- Knowing how to ground models in scenarios
- Maintaining good aesthetic sense
- Checking and challenging preconceptions (which we've seen result in the technique of pivoted normalization and, via a "different" Bayes flip, the language modeling approach)

2 A Basic LM Retrieval Paradigm (Continued)

Recall from last time that we had begun to develop a topic-based language modeling approach in which documents and queries were created from topic models. Thus,

$$\text{(document topic models) } T_D \rightarrow D$$

$$\text{(query topic models) } T_Q \rightarrow Q$$

(where T_D and T_Q are independent).

In this model, relevance is equivalent to a match between a document's and a query's topic model.

$$(R = y) \equiv (T_D = T_Q)$$

We can score documents with the function $P(R = y|D = d, Q = q)$, since d and q are the values we have available to us. However, once again we have to ask what this probability value means: why is the value of P not simply 0 or 1, since it appears that relevance is fixed if we know the identity of both the query and the document?

Consider the case of two documents d and d' that are copies of each other, but with d generated by t_1 and d' generated by t_2. Thus the two documents contain identical content but were generated by different topic models. We're really looking at *content*, not the physical document itself. (This could be considered a form of binning.)

Can we develop a useful scoring function from our probability term?

$$P(R=y|D=d,Q=q) \tag{1}$$

(introduce our new random variables; because a summation is used, all possible document and query topic models must be countable)

$$= \sum_{t,t'} P(R=y,T_Q=t,T_D=t'|D=d,Q=q) \tag{2}$$

(perform a Bayes flip)

$$= \sum_{t,t'} \frac{P(Q=q|R=y,T_Q=t,T_D=t',D=d)\cdot P(R=y,T_Q=t,T_D=t'|D=d)}{P(Q=q|D=d)} \tag{3}$$

(Here note that the denominator equals $P(Q=q)$ by independence and does not affect ranking)

$$\overset{rank}{=} \sum_{t} P(Q=q|R=y,T_Q=t,T_D=t,D=d)\cdot P(R=y,T_Q=t,T_D=t|D=d) \tag{4}$$

(because $R=y$ implies that $T_Q=T_D$. However, we can now see that most of the terms in our formula do not affect $Q=q$, and the $R=y$ term is made redundant.)

$$\overset{rank}{=} \sum_{t} P(Q=q|\cancel{R=y},T_Q=t,\cancel{T_D=t,D=d})\cdot P(\cancel{R=y},T_Q=t,T_D=t|D=d) \tag{5}$$

(Furthermore, we want to separate the second term into its two component events)

$$P(T_Q=t,T_D=t|D=d) = \overset{P(T_Q=t)\text{ by independence}}{P(T_Q=t|\cancel{T_D=t,D=d})}\cdot P(T_D=t|D=d) \tag{6}$$

(Note that in lecture the term $P(T_Q=t)$ was mistakenly dropped at this point, but that cannot be done unless $P(T_Q=t)=P(T_Q=t')$ for all t, t'. So we're left with:)

$$\overset{rank}{=} \sum_{t} P(Q=q|T_Q=t)\cdot P(T_Q=t)\cdot P(T_D=t|D=d) \tag{7}$$

(Assume a function $t^*(d)$ as a [MLE/MAP] estimate of a single "best" model to be inferred from d, where $P(T_D=t^*(d)|D=d)=1$)

$$= P(Q = q|T_Q = t^*(d)) \cdot P(T_Q = t^*(d)) \quad (8)$$

This is a form that resembles one we've seen before, namely, as $P(Q = q|D = d, R = y) \cdot P(R = y|D = d)$ in the first language model derivation. The second term, $P(T_Q = t^*(d))$, is a prior on whether $T_Q = t^*(d)$. If a uniform prior can be assumed (which requires that the topic set is finite), then the value for $P(Q = q|T_Q = t^*(d))$ can be recovered as our final score.

An extension common in practice is to consider the "distance" between the distributions of T_Q and T_D as a scoring function.

With the completion of this derivation, we've ended up with the following perspective on query/document modeling for the standard *ad hoc* retrieval setting: there are two basic models, the Vector Space Model and the probabilistic model. The latter can further be subdivided into two "flavors", "classic" probabilistic modeling and the Language Model approach.

3 Relevance Feedback

To set the stage for next time, consider the following observations about relevance:

- In probabilistic retrieval, we had problems with missing relevance information. For example, in the case of $P(A_j|R = y)$, we'd be set if we only had knowledge about relevance!
- The Vector Space Model makes no explicit mention of relevance; if we had relevance information, we would have to decide what to do with it.
- In the Language Modeling approach, we had relevance explicitly represented at one point but then got rid of it! (Though this depends on which scoring function is used.)

Relevance feedback means that we have relevance-labeled documents for a given query available to us (Ruthven and Lalmas '03). Why would a user be willing to provide this feedback after having seen some relevant documents (presumably satisfying her stated information need)?

- The user might be very interested in recall and want to see *all* the documents relevant to the query. This could be for the purpose of novelty (or lack thereof) detection: checking extant research for a certain idea, or watching for plagiarism. Another reason might be to mitigate (some) sample bias.
- The user might be browsing the document collection.
- The system might be giving no wholly relevant documents (but some partially relevant ones), and the user *really* needs the information he asked for.

4 Questions

4.1 Incorporating Relevance feedback in the VSM

What could be one way of incorporating the relevance feedback by the user into a ranking engine that uses the vector space model?

Solution

We propose a possible solution as follows. Please refer to figure 1 for a visual representation

1. Initially, a document clustering algorithm is employed beforehand to split the corpus into groups of similar documents. In our example, we have cluster A and cluster B.
2. A query q is issued by the user. Lets assume this query is closer to cluster B than cluster A. However, one document in cluster A, that is closest to q, is retrieved in addition to all the documents in cluster B. Moreover, the single retrieved document from cluster A is given a weak ranking, due to its distance from q.
3. The user provides relevance feedback. From the feedback, we find that the sole retrieved document from cluster A is, in fact, very relevant. Hence, more of the documents from cluster A, if not all, should be retrieved in the refined search.
4. In order to incorporate the user feedback into our next iteration, we take the cluster of documents in which the user-identified relevant document belongs to and rotate it some angle θ around the origin towards the direction of the query vector. Hence, in our example, all documents belonging to cluster A are rotated, say, $\theta = 15^o$ about the origin towards the query vector. Hence, in the next set of search results, more documents from cluster A will be included in the set of relevant documents and ranked higher than before. (This is a form of "document expansion", as opposed to query expansion. Is there any tradeoff involved in moving the document vector instead of the query vector?)

4.2 Further Thinking: Alternate Derivation

We saw in our derivation of a scoring function that a Bayes flip was used to express the result as a conditional probability of $Q = q$ (which eventually became $P(Q = q|T_q = t)P(T_d = t|D = d)$). Note, however, that the terms $D = d$ and $Q = q$ carry equal "weight" in the sense that they appear as part of the condition in the initial expression $P(R = y|D = d, Q = q)$. It is possible to carry out a parallel derivation in which $D = d$ is flipped to the other side instead of $Q = q$; this results in the scoring function $P(D = d|T_D = t^*(q)) \cdot P(T_D = t^*(q))$. Does this form offer any advantage to us in practice, or are the probability calculations equally problematic (or equally simple) for documents and queries?

Alternate Derivation

Applying the Bayes flip on equation 2

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

Employing the same tricks as we employed in section 2 of these notes, we arrive with the following expression.

$$Score = P(D = d|T_D = t^*(q)) \cdot P(T_D = t^*(q))$$

Is there advantage in doing this?

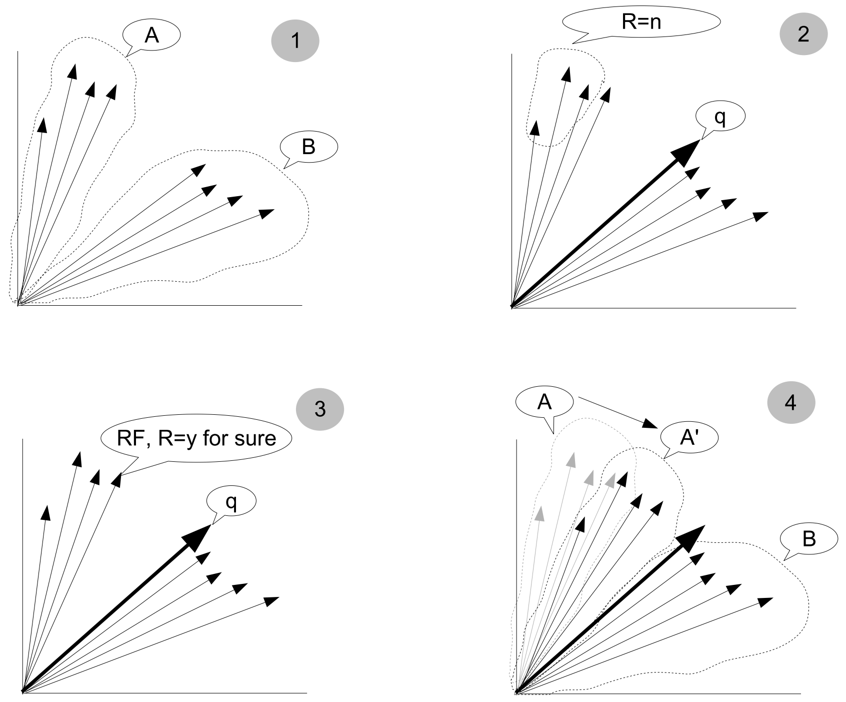

Figure 1: Incorporating Relevance Feedback in VSM

4.3 Further Thinking: Motivating the user to give relevance feedback

We believe, based on our own personal experiences, that the user might be more willing to provide feedback if he could get some sort of a numeric measure on the improvement that the user might be able to achieve by giving his feedback.

For example, while presenting the user with the first set of results, the user can be informed that *The accuracy of your search will improve by X% for each document that is marked relevant or not relevant.*

We need to answer the following questions to make this possible:

1. Is it actually possible to put a numeric value on the possible improvement *before* incorporating the user feedback into our ranking function?
2. If the answer to question no.1 is yes, how can we design a model to generate the statistics that could be presented to the user?