CS630 Lecture 15: Preference Implicit Feedback

Lecture by Lillian Lee
Scribed by Randy Au, Nick Gerner

March 28, 2006

In this lecture, we outline a preference implicit feedback mechanism described by Joachims in 2002 and by Radlinski and Joachims in 2005. One key insight we've seen before is to use clickthrough data to identify pair-wise document summary preferences for queries. If document summary $d^{(i)}$ is listed before summary $d^{(j)}$ and $d^{(j)}$ was clicked on we can say that $d^{(j)}$ is preferred to $d^{(i)}$.

1 Insights and Problems

This can be extended to summaries over several queries. Suppose q_k yields a set of document summaries $\{s_1, s_2, ...\}$ and the user doesn't click on any of these. Then suppose the user reformulates their query to q_{k+1} with summaries $\{s'_1, s'_2, ..., s'_i, ...\}$ and the user clicks on s_i. In general we might like to say that s'_i is preferred to all other s_j and all other s'_l for $l < i$. However, we can't guarantee that the user actually considered all other s_j. Going back to the eye tracking research we've seen before, we can say that with high probability (in some settings) the user did consider s_1 and s_2. So we say that the user prefers s'_i over s_1 and s_2. This is a highly accurate heuristic (recall the preference results from the eye tracking study), but provides very sparse data.

We have two important problems though. First, while the data is highly accurate, it isn't in the same format as the implicit feedback data we've used before. We only have relative information between documents, no relevance judgments. Second, the relevance feedback we collect in this way may not generalize to other queries (since the judgments are with respect to the original query or query chain).

2 Generalizing Preference Implicit Feedback

The idea Radlinski and Joachims present is to group the queries with the documents. Consider a vector space. Traditionally we had the dimensions represent features of the documents. Now we will consider the dimensions also incorporating query features along with document features. Instead of representing a document by a vector in the vector space, we represent (q, d) by the vector $\phi(q, d)$. For example one coordinate of $\phi(q, d)$ might be the cosine between the traditional query and document vectors from the VSM. Some other possible coordinate rules follow:

$$\phi_i(q,d) = \begin{cases} 1 & \text{if } d = \text{ the CU homepage and } q \text{contains "big red"}, \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_i(q,d) = \begin{cases} 1 & \text{if } d \text{ is a Finnish website}, \\ 0 & \text{otherwise} \end{cases}$$

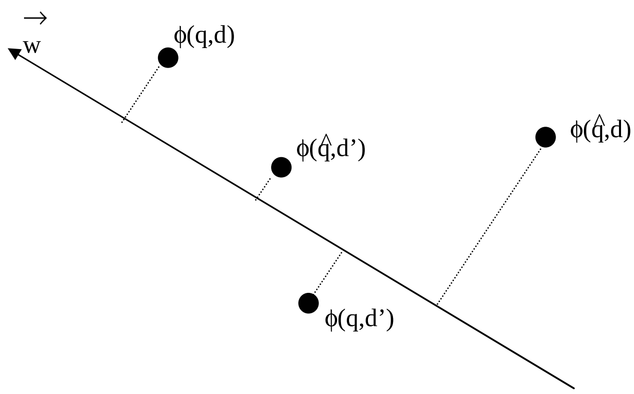
$$\phi_i(q,d) = \begin{cases} 1 & \text{if } d \text{ is ranked in the top ten for } q \text{ according to Google or some other external search engine }, \\ 0 & \text{otherwise} \end{cases}$$

This model is at least as expressive as the vector space model since we can represent the VSM's ranking method as a single feature. The new model allows us to incorporate many more query specific features. But notice that the second example above for the new model was also expressible in the original VSM.

However, on one hand, we have a *very* high-dimensional space now: at least $m \times n$ dimensions result just from the first case above. But on the other hand, the vectors themselves will probably be very sparse.

3 Training

Given that we have a model for representing queries and documents, we need a way to represent our preference implicit feedback in this model. Specifically, suppose we know that with respect to q, d is preferred to d'. We also know that with respect to $\hat{q}, d'$ is preferred to d. Our goal is to produce a vector $\vec{w}$ such that the above preference information can be encoded in the length of the projections on to $\vec{w}$ of the different query/document vectors we have. In our example above we want to preserve $\vec{w} \cdot \phi(q,d) > \vec{w} \cdot \phi(q,d')$ and $\vec{w} \cdot \phi(\hat{q},d') > \vec{w} \cdot \phi(\hat{q},d)$. This is illustrated below.



In general this is an NP-Complete problem (constraint satisfaction). However, in practice we try to minimize the constraint violations (the support vector machine approach).

In order to compute a ranking for a new query $\hat{\hat{q}}$ we compute $\phi(\hat{\hat{q}},d) \forall d$ and rank in descending order of $\vec{w} \cdot \phi(\hat{\hat{q}},d)$.

4 Exercise

Recall that our major problem with click-through data was that it is not the same sort of data as the relevance data we had access to in other feedback schemes. However, one new piece of information that we felt we could rely on was preference data of clicks provided we had a heuristic to determine what a user had probably looked at but didn't click on, and so a new scheme was required to encode the relative positions of document query pairs in a high-dimensional vector space.

Now consider this search situation. A user has an difficult search task that stretches their knowledge and ability to articulate their information need. It should be reasonable to assume that the user makes a "best effort" attempt at providing as best a query they have the ability to produce.

In such a search situation, the user's ability to judge the relevance of a document from a summary is obviously limited, which partly explains why there is a very low correlation between clickthroughs and relevance data. Another potential explanation being that summaries are not representative enough to the underlying documents.

1) Now, consider the relationship between the documents the user clicked and supposedly browsed through, judged inadequate, then reformulated their query. What has our class discussion stated about about the relationships between documents and query reformulations, a priori?

- Our class discussion makes no claim whatsoever about the relationship between the two other than assuming that they're within the same search context. It instead immediately starts analyzing the user click behavior.

2) Now, taking into account our assumption that users could do no better with their initial query, but after their initial search they then reformulate their query into a supposedly "better" one, how could this come about?

- In such a situation, it would seem that the only source of inspiration for a better query would be any documents that were supposedly looked at. Users either learn terms that potentially would make a better query, or learn terms to exclude in order to filter out irrelevant documents. We would therefore have some material link between their query reformation and the documents/summaries that the user has clicked on.

3) And finally, how can we integrate this scheme into our extended vector space?

- This link between new query and document suggests a possible $\phi(q, d)$ for the new vector space, since we can incorporate a condition such as "1 for any document that "is similar" (for some definition of similar) to a document that contains a term that was in q_k but not $q_{(k-1)}$." So as not to over-generalize, we might want to consider restricting to documents that bear similarity over some threshold with relation to q.