CS 6740: Advanced Language Technologies April 2, 2010

Lecture 15: Implicit Relevance Feedback & Clickthrough Data

Lecturer: Lillian Lee Scribes: Navin Sivakumar, Lakshmi Ganesh, Taiyang Chen

Abstract

Explicit relevance feedback is easy to use, but hard to obtain and has the further weakness of not generalizing well across queries or users. In this lecture we explore the alternative of using implicit relevance feedback. We sketch several forms of implicit feedback, and focus on clickthrough data and how it may be used to extract relative relevance judgements. We find that clickthrough data, while imposing no overhead on the user, needs to be corrected for several confounding factors, and requires some work to incorporate into a retrieval system.

1 Introduction

Recall that explicit relevance feedback consists of $r/\bar{r}$ labels (corresponding to judgements of "relevant"/"non-relevant") given by the user for the top k results returned by the retrieval system for a given query. We identified the following problems with the explicit RF approach:

- Providing explicit RF requires extra work from the user, and as a result is hard to incentivize.
- Explicit RF does not generalize across queries or users. This exacerbates the incentivization problem, as it would appear that despite extra work from the user, the payoff is not high.

In this lecture, we shall discuss the *implicit relevance feedback* approach, which addresses the above problems with explicit RF. However, we shall only address the first issue in this lecture (although IF does address the second issue as well).

2 Implicit Relevance Feedback

Implicit relevance feedback (IF) consists of observations of user behavior during normal interaction. Thus, it requires no extra work on the part of the user, and avoids the incentivization issues of explicit RF. However, it has the disadvantage of being "noisier" since, not having explicit $r/\bar{r}$ labels, we must *guess* at the user's judgements. How effectively we can make use of this noisy data, and how much work it involves, is the subject of the rest of this lecture.

First let us consider the various kinds of implicit feedback that might be gathered:

1. **Clickthrough data:** Observations of which search results[1] the user clicks on. This is possibly the most widely used form of implicit feedback, and shall therefore be the focus of this lecture. The basic idea is that the user probably tends to click on results that are more relevant. We defer detailed treatment of this form of IF till the next section.
2. **User Query History:** Observations of the user's history of query submissions. This includes reformulations, or query rewrites, which can be used to infer the user's dissatisfaction with the results returned for the original formulation. An examination of the queries that immediately preceded a query can also be an indication of the user's interest, which can be used to disambiguate queries that have meanings in multiple domains. The canonical example of such a query is "Java", which could refer to coffee, the Indonesian island, or the programming language; knowing that one of the previous queries was "C++", another programming language, for instance, would help pinpoint the meaning of this query.
3. **User's Entire History:** Observations of all information created, copied, or viewed by the user. This could include everything from webpages viewed, to emails, calendar items, and documents in the user's filesystem. Teevan et al. [TDH05] proposed various ways that all this information may be used to infer relevance judgements.
4. **Reading Time:** Observation of the amount of time the user spends on each result. It seems reasonable that a user would spend more time on more relevant results. However, the jury is still out on this subject. Morita et al. [MS94] postulated that this quantity correlates positively with the user's interest, while Kelly et al. [KB04] refuted this claim.
5. **Eye Tracking:** Observation of features such as eye fixation and pupil dilation as the user observes the results. The hypothesis is that features such as duration of fixation and diameter of the pupil vary in a systematic way between relevant and non-relevant results; for example, larger pupil diameter might be an indication of relevance. Salojärvi et al. [SPS+05] conducted a small study measuring such effects.

3 Clickthrough Data

Let's take a closer look at implicit relevance feedback in the form of clickthrough data. In order to understand how clickthrough data may be used, we must first answer the following questions:

1. *Is clickthrough data equivalent to explicit RF?* In other words, we are asking

[1]Note that by "results" we actually mean *snippets* of the kind that Google returns for a query. This is because the user decides which results to click on based only on the summary snippets presented to him. (We shall use the masculine pronoun henceforth for brevity, but we intend that it be interpreted in a gender-neutral fashion.).

whether a click is equivalent to a label of r, and a non-click to a label of $\bar{r}$. It is easy to see that this is not the case. Here are some reasons why not:

(a) A user usually stops clicking on results once his information need has been met; therefore, a non-click might merely mean that the user is done, and probably not that those results are non-relevant.

(b) A user clicks on a result based on his perusal of the *snippet* he is presented with; in this sense, the click information judges the snippet and not the document itself.

(c) A user may click on the top k results simply because he trusts the retrieval system enough that he believes (without any further information) the top results will satisfy his information need; thus, his clicking on a result might only be an indication of a trust bias, and not a judgment of relevance.

(d) A user may get side-tracked and click a result simply because it looks interesting, even though it is not relevant to the current information need.

(e) A user may omit possibly relevant results simply because their sources are not as authoritative as the ones that he is previously aware of.

Note that we side-step issue (b) by simply treating clickthrough data as RF on the snippets themselves, rather than the underlying search result. We can then ask the question of whether clickthrough data provides accurate information about the snippets.

2. This brings us to the next question: *Can relevance information be extracted from clickthrough data, and, if so, how?* As we hinted at before, IF (such as clickthrough data) is noisier than explicit RF, and some work must be done to extract useful relevance information from it. We describe a series of studies that were performed between 2002-2005 that address this question.

Joachims et al. [JGP$^+$05, JGP$^+$07] set out to discover the correlation (if any) between clicks and relevance. The following information was gathered:

- *Clicks:* User group A was asked to complete a set of pre-specified web search tasks (they were allowed to formulate their own queries to achieve these tasks). From the results returned by the test retrieval system, only those the users clicked on were recorded.
- *Relevance Judgements:* User group B was presented the results shown to group A (all the returned results, not just the ones clicked on) and asked to rank them in order of their relevance to the search tasks (ties were allowed, to make the task easier). Note that group B was presented the result set in a random order to control for the effect of result ordering in clicking behavior.
- *Eye-tracking Information:* Eye-trackers were employed to gather information on which result snippets user group A viewed (where "viewed" is interpreted as "spent some time looking at"). This information helps to avoid the inference of negative feedback on results that were not viewed at all.

The study resulted in the following findings:

1. Users tend to scan results from top to bottom[2], in order of their rank - an unsurprising finding. More specifically, however, it was found that (using the notation s_i for the i^{th} snippet) s_1 and s_2 were viewed immediately after their presentation; and the subsequent snippets were viewed as follows: pause, s_3, pause, s_4, s_5, s_6, *long* pause, s_7. The long pause preceding s_7 was because the screen only fit 6 results, and the user needed to scroll down to view the next result.
2. For clicked snippet s_i, it was found that s_{i-1} and s_{i+1} were viewed more than 45% of the time, with this percentage being higher for smaller i's. This is a positive result, as it seems to indicate that clickthrough information encapsulates relevance judgement not only of s_i, but also of s_{i-1} and s_{i+1}.
3. It was found that s_3 was viewed less than 50% of the time. This shows conclusively that a non-click is *not* an indication of non-relevance, since many non-clicks are simply evidence that the user never viewed the result; hence clickthrough data is not good for negative feedback.
4. It was found that s_1 was viewed around 80% of the time, and clicked on about 40% of the time. Similarly, s_2 was viewed about 70% of the time, and clicked on approximately 10% of the time[3]. Compare this against the fact that the absolute relevance judgements showed s_1 to be more relevant than s_2 42% of the time, while the reverse was true 24% of the time. The fact that s_1 was clicked on about four times more often than s_2 despite the fact that it was more relevant than s_2 only twice as many times as the reverse suggests that there is a bias towards the top result. To confirm this, the authors tried secretly swapping the top two snippets presented to the users. It was found that the top result (what *was* s_2) was viewed about 75% of the time and clicked on 30% of the time, while the next result (which *used to be* s_1) was viewed about 70% of the time and clicked on 10% of the time. This near-identical result after the snippet-swap confirms the bias towards the top result. This bias is one of the "noise" elements that needs to be corrected when the clickthrough data is used to infer relevance judgements.

Thus we see that clickthrough data does encode relevance information, even if it is noisier than in explicit RF. This takes us to the question of *how* relevance information can be extracted from clickthrough data despite the noise elements described in the above study.

4 Using Implicit Relevance Feedback

The study in [**Joa02**] proposed the following clever idea: Treat clicks as *relative* judgements. In other words, each click tells us something about the surrounding

[2]The experiment system installs a proxy to filter out ads in search results, so there are no distractions when users look through the results.

[3]Note that the low clicking statistic is partially due to users reformulating their queries after viewing the presented results.

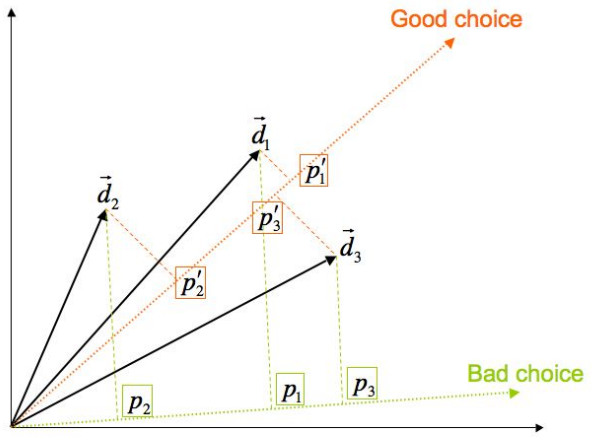

Figure 1: Choosing a Direction of Relevance

snippets. So, if s_i was clicked on and s_{i+k} was not, this does *not* imply that s_i is more relevant than s_{i+k}, since it is likely that the user did not view the lower ranked result; however, if s_{i+k} was clicked on and s_i was not, then we can infer that s_{i+k} is more relevant than s_i, since the user typically views results from top to bottom, and s_i was most likely viewed.

Along these lines, Radlinski et al. [RJ05] suggested that query reformulations can be used in a similar manner. The basic approach is as follows: suppose the user issues a sequence of queries $q_1, q_2, \ldots$. Assume that we have some method for determining whether query q_k represents a new information need or is a reformulation of the prior query q_{k-1}. Consider the case when q_k is a reformulation of the q_{k-1}. If the results s_1^{k-1} and s_2^{k-1} for query q_{k-1} were not clicked on, but the result s_i^k for query q_k was clicked on, we can infer that s_i^k is more relevant than both s_1^{k-1} and s_2^{k-1}. Note that we cannot draw any conclusions about the lower ranked results s_j^{k-1} for query q_{k-1}, since it is likely that the user never viewed those results.

To continue with the idea of relative judgements, Joachims et al. [Joa02] go on to suggest that pairwise relationships be used to infer a spatial notion of relevance, which can then be used to rank documents. Consider a fixed query q. Suppose we are given documents d_1, d_2, d_3 (where index does not correspond to presentation order) and relative judgements $\text{rel}(d_1) > \text{rel}(d_2), \text{rel}(d_1) > \text{rel}(d_3)$. Assume in addition that we have a vector representation $\vec{d_i}$ of each document d_i. The goal is to pick a direction in the vector space that induces an ordering on the document corpus that respects the inferred relative judgements. Fig. 1 illustrates this concept for a 2-dimensional document vector space. The direction labeled "bad choice" violates the relative document judgements since we see that the document vector projections along this direction (labeled p_i for document d_i) have the ordering $p_2 < p_1 < p_3$. On the other hand, the direction labeled "good choice" respects the relative judgements since the document projections along this direction (labeled p_i'

for document d_i) have the ordering: $p'_2 < p'_3 < p'_1$. Thus, this direction could be used to rank the documents.

5 Problems

1. From the results in [JGP+05], we have seen that there is a user bias towards clicking the top result even if the second result is more relevant. Also, clickthrough data does not provide reliable negative feedback since the bottom results are usually not viewed at all (and hence, naturally, not clicked). Thus, clickthrough data is much less useful than it would be in a world where presentation-order didn't matter. In this context,
   (a) Can you think of a way to change the retrieval system to control for presentation-order bias?
   (b) What are the advantages and disadvantages of this altered system?
2. In this problem we explore the approach of [Joa02] for extracting relative feedback from clickthrough data and then applying the relative judgments to rank documents. Suppose we have a collection of documents with the following vector representations:

$$d_1 = (1, 1, 0, 0)$$
$$d_2 = (1, 0, 1, 0)$$
$$d_3 = (0, 1, 1, 0)$$
$$d_4 = (0, 1, 0, 1)$$
$$d_5 = (0, 2, 0, 0)$$

   Assume that the user is presented the results in the order d_1, d_2, d_3, d_4, d_5 and we have obtained clickthrough data for that sequence of results.
   (a) Suppose we know that the user clicked only on results d_2 and d_4. What relative judgments can be inferred from this data?
   (b) Find a vector $\vec{w}$ such that the projections of $d_1, \ldots, d_5$ satisfy the relative judgments inferred in part (a). What is the complete ranking of the results based on their projections onto $\vec{w}$? Can you find a different vector $\vec{w}'$ that gives a different overall ranking of the results while still satisfying the constraints from part (a)?
   (c) Suppose in addition that the user had clicked on d_5. How does this change the relative judgments that can be inferred? Can you still find a direction such that the projections of the documents satisfy the new constraints?

6 Solutions

1. (a) We could control for presentation-order bias by randomizing the order in which results are presented to the user. (Results that score below

some threshold of relevance are omitted.) Thus, presentation order is no longer correlated with scores of the results[4]; the user (apprised of this change) can no longer infer any relative relevance information from the presentation order, and should stop basing his clicking behavior on it.

(b) Let us make the following assumption: the "relevance threshold" filters out most documents in the corpus, and the presented results are few enough that the user can be reasonably expected to look at all of them. Since we believe the user is looking for the most-relevant result, we can further assume that he *does* scan through all the presented snippets before deciding which to click on (presentation order no longer being useful here). This is an awkward assumption to make, but perhaps reasonable when the user is aware that the system presents results in random order. In light of this assumption, we discuss this model's advantages and disadvantages:

- Advantages:
  - Clickthrough data now provides negative feedback! In this model, a result that was not clicked on is less relevant than one that was.
  - Clickthrough data now provides less noisy positive feedback, since presentation-order bias has been removed.
- Disadvantages:
  - Users expect the retrieval system to provide some ordering information on the returned results and will find this system frustrating. Since the retrieval system does compute ordering information (relevance scores), choosing to withhold such information seems quite user-unfriendly.
  - There is no information on the withheld results (ones that failed the "relevance threshold" filter) in this model. In particular, users won't have access to relevant documents that scored below the threshold, so the system will not receive direct feedback on documents whose scores are excessively low. (Note that this is a problem with standard clickthrough data as well.)

In order to mitigate some of the disadvantage of user inconvenience, we could offer randomized result ordering on a request-basis. That is, if the user expresses willingness to provide some valuable feedback to improve the retrieval system, then he will be presented his results in random order (as described above) and his clickthrough information is used to improve retrieval. There might also be ways to incentivize users to occasionally use this option. On the other hand, this essentially brings us back to the case of explicit relevance feedback. A possible variation on this idea is to have the system choose whether to present results in score order or in

[4]One way to achieve this might be to present the results in a manner where, unlike a list-style presentation, there is no inherent ordering; however, this might prove an elusive quest - people impose spatial ordering on most things.

random order; the system designer would have to balance the short-term user inconvenience incurred when presenting randomized results against the long-term user benefit from improved search results due to more useful relevance feedback.

2. (a) Because the user clicked on d_2, we can conclude that d_2 is preferred to any unclicked result that was positioned higher in the sequence of results; hence $\text{rel}(d_2) > \text{rel}(d_1)$. Note that we may not make any relative judgment between d_2 and d_3 (or d_5), even though d_2 was clicked and d_3 was not clicked. This is because d_3 was presented below d_2, so we do not know if d_3 was viewed by the user when making the decision to click d_2. Similarly, we can conclude that d_4 is preferred to the unclicked results d_1 and d_3, both of which were presented above d_4. Hence $\text{rel}(d_4) > \text{rel}(d_1)$ and $\text{rel}(d_4) > \text{rel}(d_3)$.

   (b) Note that the requirements on $\vec{w}$ from part (a) are the following:[5]

$$\vec{w} \cdot d_2 > \vec{w} \cdot d_1$$
$$\vec{w} \cdot d_4 > \vec{w} \cdot d_1$$
$$\vec{w} \cdot d_4 > \vec{w} \cdot d_3$$

   It is straightforward to verify that any direction $\vec{w} = (w[1], w[2], w[3], w[4])$ satisfying the following inequalities will satisfy the ranking constraints:

$$w[3] > w[2]$$
$$w[4] > w[1]$$
$$w[4] > w[3]$$

   (In fact, these inequalities are necessary and sufficient.) For example, we can see that $\vec{w} \cdot d_2 > \vec{w} \cdot d_1$ if and only if

$$w[1] + w[3] > w[1] + w[2]$$
$$w[3] > w[2]$$

   Now, we can demonstrate two directions $\vec{w}$ and $\vec{w}'$ that give different overall rankings. For example, consider $\vec{w} = (1, 2, 3, 4)$ and $\vec{w}' = (3, 1, 2, 4)$. In order to determine the rankings induced by these $\vec{w}$, we first compute the dot products with the document vectors:

$$\vec{w} \cdot d_1 = 1 + 2 = 3$$
$$\vec{w} \cdot d_2 = 1 + 3 = 4$$
$$\vec{w} \cdot d_3 = 2 + 3 = 5$$
$$\vec{w} \cdot d_4 = 2 + 4 = 6$$
$$\vec{w} \cdot d_5 = 2 \times 2 = 4$$

[5] Observe that we may use the dot product in place of projection because $\vec{w} \cdot d_i$ is proportional to the projection of d_i onto $\vec{w}$; since $\vec{w}$ is fixed, the dot product and projection induce the same ordering.

This gives the ordering

$$\vec{w} \cdot d_4 > \vec{w} \cdot d_3 > \vec{w} \cdot d_2 = \vec{w} \cdot d_5 > \vec{w} \cdot d_1$$

For $\vec{w}'$, we find

$$\begin{aligned}
\vec{w}' \cdot d_1 &= 3 + 1 = 4 \\
\vec{w}' \cdot d_2 &= 3 + 2 = 5 \\
\vec{w}' \cdot d_3 &= 1 + 2 = 3 \\
\vec{w}' \cdot d_4 &= 1 + 4 = 5 \\
\vec{w}' \cdot d_5 &= 2 \times 1 = 2
\end{aligned}$$

giving the ordering

$$\vec{w} \cdot d_4 = \vec{w} \cdot d_2 > \vec{w} \cdot d_1 > \vec{w} \cdot d_3 > \vec{w} \cdot d_5$$

(c) If the user in addition clicked d_5, then we introduce the following relative judgments:

$$\begin{aligned}
\text{rel}(d_5) &> \text{rel}(d_1) \\
\text{rel}(d_5) &> \text{rel}(d_3)
\end{aligned}$$

There is no direction $\vec{w}$ that satisfies these constraints simultaneously with the constraints in part (a). To see this, note that in part (b) we showed that the judgment $\text{rel}(d_2) > \text{rel}(d_1)$ induced the requirement $w[3] > w[2]$; now, in order to satisfy $\text{rel}(d_5) > \text{rel}(d_3)$, we must have $\vec{w} \cdot d_5 > \vec{w} \cdot d_3$, or equivalently

$$\begin{aligned}
2w[2] &> w[2] + w[3] \\
w[2] &> w[3]
\end{aligned}$$

References

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[JGP+07] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7, 2007.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.

[KB04] Diane Kelly and Nicholas J. Belkin. Display time as implicit feedback: understanding task effects. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 377–384, New York, NY, USA, 2004. ACM.

[MS94] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–281, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[RJ05] Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, New York, NY, USA, 2005. ACM.

[SPS+05] Jarkko Salojärvi, Kai Puolamäki, Jaana Simola, Lauri Kovanen, Ilpo Kojo, and Samuel Kaski. Inferring relevance from eye movements: Feature extraction. In *Helsinki University of Technology*, page 2005. No, 2005.

[TDH05] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM.