Bayesian Model for Information Retrieval

Jaime B. Teevan
teevan@mit.edu

September 29, 2000

Abstract

This paper is a quick formalization of what David Karger and I have been discussing regarding developing a Bayesian framework for information retrieval. The goal of the framework presented here is to create a well founded basis for information retrieval that can incorporate relevance feedback in a consistent way. Efficiency is not an objective, although when we have found an effective way to do retrieval, we will investigate ways to make it efficient.

1 Introduction

Text retrieval is a difficult problem that has become both more difficult and more important in recent years. This is because of the increased amount of electronic information available and the greater demand for text search as a result of the World Wide Web. People are surrounded with large quantities of information, but unable to use that information effectively because of its overabundance.

The most common solution to this problem is the Web search engines. Here text search is done using a collection of heuristics, each new trick giving a slight improvement over previous methods. While people may develop models to describe these ad hoc methods, the methods are developed empirically based on what has been shown to work, and what has not. The result has been reasonably successful text retrieval. However, without a theoretical framework to describe a problem, it is difficult to talk about it. We are in the process of developing a model for information retrieval.

By developing a good model of text documents and their relationships to each other and to a query, it is easier to understand and improve on retrieval methods. The assumptions inherent in the retrieval are obvious and the trade offs between the complexity of indexing and retrieving and the benefit of improvements are clear.

2 Related Work

This is a list of relevant reading. Some I have already read, and comments are included explaining what useful information is contained in the paper. Others I have yet to read, and are included with comments describing what I hope to gain from them.

- [Hec95] is a good introduction to a Bayesian approach to probability and statistics. This paper has been particularly useful in its discussion of Dirichlet distributions. Is there a proof for the feedback to the thetas taking place in the likelihood maximization process? Is there discussion of partial feedback if you are not sure of what you have observed? If not here, perhaps it can be found starting from here.
- [BL99] proposes a probabilistic approach to information retrieval based upon the ideas and methods of statistical machine translation.
- [MKM] describes a Bayesian approach to one of the conventional probabilistic information retrieval models.
- [RW94] is a good model to compare our framework with, and also perhaps to compare results with. I've had this on my desk forever, and still haven't gotten around to reading it.
- [Gre98] investigates the way the terms are distributed in a TREC corpus and talks about the ramifications of this on information retrieval.

3 The Framework

The ideal model to use in information retrieval is, by definition, the one that works the best. Of course, if we could find the model that produced what we knew to be the "right" answers, then we would know the right answers, and should just return them. So instead we try to approximate the ideal model and look for the one that best fits the data we have.

Something fundamental that seems to be left out of many of the previous text retrieval models models are the terms that are *not* in the user's query that also describe the user's information need. The standard user of a text retrieval system uses just a few words to specify what he or she is looking for. However, this need is likely to be more accurately described by the inclusion of many words and possibly even the exclusion of others. Perhaps a better model also includes some form of probabilistic relevance feedback, where relevant documents reinforce each other.

We make the assumption that the documents comprising the corpus fall into one of two categories: those that are relevant to the user's information need, and those that are irrelevant. The documents that are relevant to the query share certain distinguishing features, namely the presence of those the features that make them relevant to the users needs and the absence of those which don't. The same holds for irrelevant documents. This results in what is essentially a clustering problem that involves finding the set of documents that share similar identifying features with each other and with the query, and don't share features with the irrelevant documents. A benefit of a model based on these assumption is that it can incorporate relevance feedback in a consistent manner.

We are in the process of investing several Bayesian frameworks to find the model that best fits the data. We assume that there exists some underlying feature distribution from which words are selected to generate a document. Furthermore, we assume that the corpus is entirely generated by selecting words from one of the two distributions, the relevant distribution, and the irrelevant distribution. These distributions are modeled as Dirichlet distributions. We are empirically testing the models we develop on TREK data and comparing the results to standard tf.idf results as well as the results from similar models.

There are several difficulties involved in developing a model for text retrieval. One is in finding a statistical model that accurately describes natural language. If the assumptions that are made in the similarities between relevant documents as compared to irrelevant documents are bad, then no model using those assumptions can be good. Associated with this are difficulties such as selecting the features to represent the corpus, and determining the representation of the query.

Another difficulty is making the model computationally feasible. Because we are saying that the interaction between documents is important, and because as a corpus grows, so do the number of relationships, it is likely that the best model is not one which can be solved quickly.

However, if these difficulties can be surmounted, improved information retrieval has ramifications in many areas, most obviously in Web search but also in any application that deals with large quantities of text. Having a good model for information retrieval in which relevance feedback can be incorporated will allow people to more easily talk about and develop new techniques.

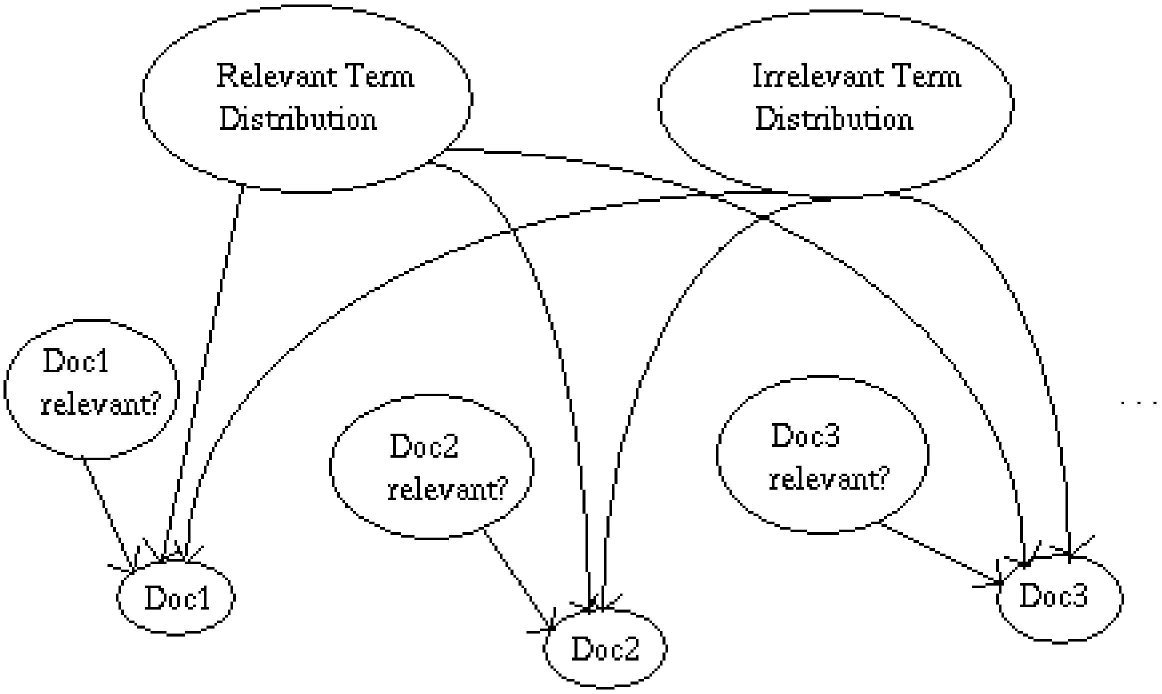

Figure 1: Model as a Bayes net.

The framework we are working with can be imagined pictorially. Figure ?? shows it represented as a Bayesian Network. As you can see in this figure, each document is a multinomial based of whether it is relevant, and the distribution of terms in relevant and irrelevant documents. Given the documents, we want to extrapolate back to find the relevance of each document, which entails finding the distribution of terms in relevant and irrelevant documents.

3.1 The Corpus

While we already mentioned that we are using a Bayesian framework to find the model which best fits the data, there are more details that need to be fleshed out to fully understand how this is done. We assume that there exists some underlying distribution from which words are selected to generate a document. Further more, we assume that the corpus is entirely generated by selecting words from one of two distributions, a distribution used to create the documents that are relevant to the query being performed, and a distribution used to create the documents that are irrelevant to the query. We chose to model this distribution as a Dirichlet distribution, as described in [Hec95]. Note that a document is either relevant or irrelevant. Each document therefor has some probability of being relevant. We will describe how to find the model that best fits the data, yielding a set of results that are ideally the best that can be found within this framework, by maximizing the likelihood that the relevant distribution, irrelevant distribution, and set of relevant documents fit our data.

To make assumptions about the corpus without actually looking at what is there is silly. We plan to look at the TREC documents to see what sort of information we can gather from them about their organization. For example, we will compare the distribution of query, and other, terms in the documents relevant to the query and not relevant to the query. There are surely other ways to look for useful information within the documents, and this is something we still need to investigate and think about further. Relevant to this is Greiff's paper from SIGIR 98 [Gre98].

3.2 The Query

The query can be accommodated into this model in several ways. One way to deal with the query is to directly alter the relevance distribution. The query consists of hand selected words that have been determined by the user to be highly representative of the relevant distribution. By fiddling with the prior, the query terms in the distribution can be made more likely. This allows for a very direct control of the way that the query affects the framework.

Another way to deal with the query is to treat it as a relevant document. The user generates the query terms from the relevant distribution, and then the user tells the system that this "document" is relevant. While this fits nicely into the framework it has the problem that this very short document is overpowered by the other documents that are also determined to fall into the relevant distribution, and will often allow the topic to drift.

The first approach is nice because it expresses more clearly the true importance of the query terms. The second is nice because it seems less arbitrary that fiddling with numbers. The system is currently implemented with the second approach. Some combination of the two approaches could be used, such as perhaps, while still treating the query as a document, understanding that the terms it contains are particularly important, and instead treating it like several relevant documents, not just one.

3.3 The Math

Let us first define some notation. The corpus D is made up of individual documents, d_j representing the jth document in the corpus. R is a vector of relevance judgments, each individual judgment r_j corresponding to a d_j being either zero, irrelevant, or one, relevant. The superscript i denotes irrelevant, and the superscript r relevant. p^i is the probability that a document is irrelevant, and p^r is the probability that it is relevant. Obviously, p^i and p^r sum to one.

We will also need to talk about the term distribution, which will be represented by the vector θ. All values in θ sum to one. Each element in θ corresponds to a term, with the kth term in θ being the probability of that term appearing.

As we discussed earlier, the term distribution is a Dirichlet distribution. This distribution, discussed in greater detail in [Hec95], is based on a vector α, each element of which corresponds to a term, represents our prior belief of the distribution of terms. Heckerman discusses in greater detail some elements of the Dirichlet distribution which we will use here, such as the how to find the most probable θ given α, how to find the probability of θ given α, and how to update α based on observation.

Because we make an assumption commonly used in text retrieval that the terms in a document are not related, the probability of a document can be expressed simply in terms of θ.

$$\begin{aligned}\Pr(d_j|\theta) &= \prod_k \Pr(d_j^k|\theta) \\ &= \prod_{t_k \in d_j} \theta_k\end{aligned}$$

Our framework has thus been defined in terms of a distribution of terms for relevant documents, θ^r, a distribution of terms for irrelevant documents, θ^i, and a set of relevance judgments, R. Our goal now is to find the model which is most likely given the corpus.

$$\max_{\theta^r, \theta^i, R} \Pr(\theta^r, \theta^i, R|D) \tag{1}$$

Let us further investigate $\Pr(\theta^r, \theta^i, R|D)$. Using Bayes Rule we arrive at the following:

$$\Pr(\theta^r, \theta^i, R|D) = \frac{\Pr(D|\theta^r, \theta^i, R)\Pr(\theta^r, \theta^i, R)}{\Pr(D)} \quad (2)$$

We will now look at each element of this equation separately. The likelihood portion of equation ?? represents the probability of the corpus given model. Because we assume that each document is independent, we are able to reduce it as follows:

$$\Pr(D|\theta^r, \theta^i, r) = \prod_j \Pr(d_j|\theta^r, \theta^i, r)$$

Now we know that when a document is relevant, it only depends on the relevant distribution, and when it is irrelevant, it only depends on the irrelevant distribution, so we can say:

$$\begin{aligned}
&= \prod_j r_j \Pr(d_j|\theta^r) + \bar{r_j}\Pr(d_j|\theta^i) \\
&= \prod_{r_j=1} \Pr(d_j|\theta^r) \prod_{r_j=0} \Pr(d_j|\theta^i) \\
&= \prod_{r_j=1} \Pr(d_j|\theta^r) \frac{\prod_j \Pr(d_j|\theta^i)}{\prod_{r_j=1} \Pr(d_j|\theta^i)} \\
&= \prod_{r_j=1} \frac{\Pr(d_j|\theta^r)}{\Pr(d_j|\theta^i)} \prod_j \Pr(d_j|\theta^i) \\
&= \prod_{r_j=1} \prod_{t_k \in d_j} \frac{\theta^r_k}{\theta^i_k} \prod_j \prod_{t_k \in d_j} \theta^i_k
\end{aligned}$$

The prior $\Pr(\theta^r, \theta^i, R)$ in equation ?? can be further broken down to $\Pr(\theta^r)\Pr(\theta^i)\Pr(R)$ since we assume that these three values are independent. We already discussed how to find $\Pr(\theta^r)$ and $\Pr(\theta^i)$ above since they are Dirichlet distributions. $\Pr(R)$ can be solved in a straightforward manner $(\prod_j r_j p^r + \bar{r_j} p^i)$ since it can be each relevance judgment can be assumed to be independent.

$P(D)$ in equation ??? is a normalizing constant, and consistent across all models. It can therefor safely be ignored when this value is used for comparison.

4 Finding the Best Model

Finding the best model, as mentioned in equation ??, means finding the parameters that maximize equation ??. This is a large search space, but there are several things that we can do to make it more feasible. In the following several sections we will investigate ways to solve the maximization problem.

It seems fair to assume that $p^r \ll p^i$. In fact, it is likely that the irrelevant distribution (θ^i) is essentially the same as the corpus distribution, meaning that this is really only a maximization problem over θ^r and R. Combining this information, equation ??, and equation ??, we arrive at the following maximization problem:

$$\max_{\theta^r,\theta^i,R} \Pr(\theta^r, \theta^i, R|D) = \max_{\theta^r,R} \Pr(D|\theta^r, R)\Pr(\theta^r)\Pr(R) \tag{3}$$

4.1 Simple method

Show the deriving that it's equal to tf/idf (more or less) here.

Try this also using the posterior probability of relevance. I still don't exactly understand why aren't the two more related? How are they related? Think about this some.

4.2 Iterative hill climbing

A fairly straightforward way of finding a local maximum is to do simple hill climbing. By first holding θ^r fixed, the optimal R for that θ^r can be found. Iterating over that step and holding R fixed while optimizing for θ^r, assuming we start with good initial estimates, a good local solution can be found. We will now discuss how each of these steps is done in further detail.

4.2.1 Optimizing with respect to R

Finding the optimal R when θ^r is held fixed turns out to be a straightforward decision that can be made one document at a time. Note that for the same reasons that we can ignore θ^i in equation ?? we can also ignore the second part of equation ??? ($\prod_j \prod_{t_k \in d_j} \theta^i_k$) and focus on the first when substituting it in below.

$$
\begin{aligned}
\max_R \Pr(D|\theta^r, R)\Pr(\theta^r)\Pr(R) &= \max_R \Pr(D|\theta^r, R)\Pr(R) \\
&= \max_R \prod_{r_j=1} \prod_{t_k \in d_j} \frac{\theta^r_k}{\theta^i_k} \prod_j (r_j p^r + \bar{r_j} p^i) \\
&= \max_R \prod_{r_j=1} (p^r \prod_{t_k \in d_j} \frac{\theta^r_k}{\theta^i_k}) \prod_{r_j=0} p^i
\end{aligned}
$$

Maximizing this equation, and finding the optimal R, merely involves, for each document, deciding a document is relevant if

$$
p^r \prod_{t_k \in d_j} \frac{\theta^r_k}{\theta^i_k} > p^i \tag{4}
$$

4.2.2 Optimizing with respect to θ^r

This is primarily from heckermans. Need to discuss it more . . .

Partial feedback to θ^r Also talked about looking at the posterior probability of relevance, and feeding back to the thetas only so much as we are confident. Discuss how partial feedback is found. Look into how ranking by this works. Is this an ok way to do the feedback (Heckerman?)?

4.3 Other ways of optimizing

The other way to find the best model to match our data is to integrate over all possible θ^r's.

$$
\Pr(event|D) = \int d\theta \sum_r \Pr(\theta^r)\Pr(r)\Pr(event|\theta, r) \tag{5}
$$

This is nice because the probability of a specific document being relevant, $r^i = 1$, can be found, allowing us to rank the documents.

$$
\Pr(r^i = 1|D) = \int d\theta \sum_{r^i=1} \Pr(\theta^r)\Pr(r) \tag{6}
$$

5 Further discussion

Unfortunately, we've had some problems with this approach, mainly that so far the results have been worse than standard tf.idf calculations. Most queries run didn't find any documents that fell into the relevant set (Optimizing with respect to R, discussed in ??4.2.1??) and therefor never iterated.

Those that did find some documents belonging to R initially worked well, but the set of relevant documents grew to be much larger than the true relevant set in a sort of topic expansion/drift. Whether it iterated or not varied greatly as a function of the prior belief of probability of relevance, but it is interesting to note that if it did iterate, the set it of documents it resulted in was most often much larger than suggested by the prior probability of relevance.

When it didn't iterate, but was ranked by the same number used to decide if it was relevant or not (equation ??4??), the results were better than when it didn't iterate, but still worse than tf.idf. This ranking function we managed to reduce to ranking the documents by query term/term corpus frequency. (Do math for this here . . .) Ranking by the posterior probability of relevance at this point did worse.

One approach we are considering taking, but need to investigate the theoretic foundation for a little more, is to not take relevance as binary during the iterations, but rather incorporate documents based on how likely we believe they're relevant (or the posterior probability of relevance, which, as noted above, worked surprisingly poorly for ranking).

It is important to note a major problem with our approach, and that is that the correct document set is not the most likely. When the correct documents were seeded as the start, during subsequent iterations the results set grew substantially, and occasionally even booted some relevant documents, finding new sets of documents with higher likelihood. This suggests a possible flaw in the model.

We've also talked about when iterating to find $\Pr(d_j|\theta^r)$ perhaps not fixing the alphas, but rather having them effected by each other. David, are there other things we've discussed that I've left out? Going through my green notebook I don't really find anything else, but I feel like there are other things.

References

[BL99] A. Berger and J. Lafferty. Information retrieval as statistical translation, 1999.

[Gre98] Warren R. Greiff. A theory of term weighting based on exploratory data analysis, 1998.

[Hec95] D. Heckerman. A tutorial on learning with bayesian networks, 1995.

[MKM] David D. Lewis Michelle Keim and David Madigan. Bayesian information retrieval: Preliminary evaluation.

[RW94] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval, 1994.