Introducing an Active Cluster-Based Information Retrieval Paradigm

Oscar Loureiro and Hava Siegelmann
Department of Computer Science, University of Massachusetts, Amherst, MA 01003.
E-mail: {oscar, hava}@cs.umass.edu

When a client interacts with an expert, e.g., a doctor, it falls upon the expert to ask questions that steer the process towards fulfilling the client's needs. This is most efficient given that the expert has more knowledge and a broader view of possible illnesses and treatments. On the other hand, when faced with an information retrieval (IR) task, most IR systems leave to the client the task of coming up with queries. We propose an information retrieval framework that assumes the responsibility of leading the users to the information, thus increasing efficiency and satisfaction.

Introduction

The access to useful information at relevant times may be the deciding factor in successful decision-making. Traditional information retrieval (IR) systems try to estimate the information beneficial to the user from the initial query. This technique is not satisfactory when the initial query is not sufficiently specific, as is frequently the case: First, the user has to know what information is available to be retrieved by the system in order to come up with an optimal query. Second, the user needs to be aware of the terms used in the collection of documents, as well as with their synonyms and similarity. Third, the user simply has to be aware of what information is important, which is difficult, especially when learning a new domain or during unpredictable environmental changes, such as in a disaster management situation.

There is a high level of dissatisfaction with current IR systems that rely on the user's initial query. If the query is not focused, the result will not be focused either, forcing the user to continue asking queries in a long and tedious search session.

Our proposal of a new IR methodology, Active Information Retrieval (AIR), is based on the notion that queries are best rendered by a process of dialog between the user and the system. A useful analogy would be to the patient–doctor interaction. The doctor is not limited to merely replying to questions, but is expected to actively engage the patient by formulating pertinent questions. This process is more expedient in arriving at a diagnosis and treatment options.

An AIR system allows the user to originate a query, which could be changed at any point, as in current search engines, but it also takes on the active role of asking questions of the user to clarify information needs. We call these questions made by the system to the user *reverse queries*. The user interface is based on a split-screen display: one part of the screen presents an ordered list of the documents retrieved based on their current ranking, while the other part presents the current reverse query posed to the user.

The active information retrieval paradigm incorporates the main results of both statistical experimental design (Atkinson & Donev, 1992; Fedorov, 1972) and active machine learning (Cohn, Ghahramani, & Jordan, 1996; Zhang, Cha, & Chen, 2001), as well as the insights of cluster-based information retrieval (Croft, 1980; Tombros, Anastasios, Villa, & Van Rijsbergen, 2002).

Active Information Retrieval is the logical extension of the notion of *Active Learning* to information retrieval tasks. While learning from examples, if those examples are chosen at random (*Passive Learning*), there is a certain probability that subsequent examples will be irrelevant for the learning task at hand (Cohn et al., 1996; Zhang et al., 2001). Thus, the idea behind *Active Learning* is the purposeful choosing of the examples to be used in the learning process.

There are two approaches in the literature on Active Learning that are characterized by the way of selecting the next learning example. In the first approach, the next example is chosen at random from the region in the space of examples that is least understood. In the second approach, the next example is chosen to minimize some global measure of uncertainty about the state of nature (entropy, margin, variance of estimates, etc.). The Active Information Retrieval system described here falls squarely into the second approach and uses entropy as the measure of uncertainty.

This article is organized as follows: in the next section, we present an example that illustrates our approach, while in the

Received October 28, 2003; revised April 26, 2004; accepted July 21, 2004

 ● Published online 31 May 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20193

section titled "Modeling the Information Needs of the User," we describe our probabilistic approach to modeling the information needs of the user. After a section describing the properties of the Dirichlet distribution, we dedicate a section to the user/system dialog, and especially to how the system should choose its next question to the user. We then compare our AIR system to previous attempts to introduce a dialog with the user in information retrieval systems. We close with suggestions for future work, both theoretical and applied.

Example

In a database, the query "apple" recalls 120 documents, with 100 related to computers and 20 related to plants and cooking. Due to the word frequencies in each document and/or the link structure in the database, the traditional IR system ranks higher all the documents about computers. Documents pertaining to plants and cooking are ranked 101 to 120.

Different users at different times may be interested in collecting information about Apple computers or the fruit of the apple tree. The traditional IR system will come back with the same list of documents in the same order every time. On the other hand, the kind of active information retrieval system proposed here may start by asking the user to express a preference between documents number 1 (the highest-ranked computer document) and 101 (the highest-ranked plant and cooking document). The user's response to this first reverse query will indicate what type of documents should be ranked higher, leading to a quick retrieval of the information sought by the user.

What would an AIR system need to be able to ask just the right question, presenting the documents originally ranked as 1 and 101 for a choice by the user? For starters, it would need to know that documents 1 to 100 and 101 to 120 are similar among themselves, but both groups are different from each other. In our approach, that knowledge is codified by a particular clustering of documents. In addition, the AIR system would need to be able to guess what the information needs of the user are and what additional pieces of information could make that guess more accurate. In our approach, such a guess is represented by a probabilistic representation of the user's information needs, which will be described in the next section. Finally, the AIR system would need to know what kinds of questions can be posed to the user and what kinds of responses can be expected. The rules for an optimal user/system dialog are analyzed in the section titled "The User/ System Dialog."

Modeling the Information Needs of the User

An AIR system consists of a collection of documents and an engine that retrieves those that are relevant to the information needs of the user. Let M be the number of documents $(d_1, \ldots, d_M)$ available in the collection and let θ_i be a measure of how relevant the i^{th} document in the collection is to satisfying the current information needs of the user. We will call θ_i the *relevance score* for document i. The information needs of the user are fully represented by the *relevance vector* $\theta = (\theta_1, \ldots, \theta_M)$, so that if θ were to be known to the system, an optimal representation of the information in the collection could be displayed to the user.

For example, if the information needs of the user would be satisfied by the first document in the collection and all other documents are completely unrelated to those information needs, then the relevance vector would be $\theta = (1, 0, \ldots, 0)$. On the other hand, if only documents one and two are relevant and the first is three times as relevant as the second, then $\theta = (0.75, 0.25, 0, \ldots, 0)$.

θ_i depends both on the information needs of the user and on what is available in the document collection. It is worth noticing that θ_i is unknown not only to the IR system interacting with the user, but to the user, who may know what kind of information is needed, but does not know exactly what is available in the collection.

Without loss of generality, we can assume that the relevance scores are non-negative. We also would like to be able to talk about how relevance scores change across queries or users; consequently, we need to normalize the relevance vector. In what follows, we will normalize θ by making $\sum_{i=1}^{N} \theta_i = 1$, and thus the relevance scores become proportions.

In most IR systems, the information provided by the initial user query is translated implicitly or explicitly into a guess $\hat{\theta}$ about the relevance vector θ, and that guess $\hat{\theta}$ is then used to rank the documents in the collection so that only the top-ranked ones are initially presented to the user. The AIR introduced here differs from this common approach by creating a dialog between the system and the user.

The goal of the dialog between the system and the user is to generate information useful for best estimating the relevance vector θ. This task of updating the system's understanding of the user in the context of uncertainty when new information is made available is best accomplished in a Bayesian framework. In such a framework, the system's current knowledge about θ is summarized by a probability distribution $P(\theta)$, which is updated with any new piece of information learned. The domain of such a probability distribution $P(\theta)$ is the simplex in R^N, defined as $L^N = \{Z \in R^N : z_i \geq 0 \ \forall i \text{ and } z_1 + \cdots + z_N = 1\}$. In this framework, the best guess about θ_i in mean-squared-error terms is $E_P(\theta_i)$, its expected value according to $P(\theta)$.

The Dirichlet Distribution

In many fields of science, the most popular family of probability distributions $P(\theta)$ for random vectors with non-negative elements that add up to one is the class of Dirichlet distributions (Aitchison, 1986). In Computer Science, Dirichlet distributions have been used in many areas, such as learning in Bayesian networks (Geiger & Heckerman, 1996, 1997), information retrieval (Blei, Jordan, & Ng, 2003; Lu, 2002), and learning naïve Bayesian classifiers (Hsu, Huang, & Wong, 2003).

The Dirichlet distribution with parameter $\alpha \in R_+^N$, represented by $\mathcal{D}_N(\alpha)$, is a probability distribution on the simplex L^N with density function

$$f(X_1 = x_1, \ldots, X_N = x_N) = \frac{\Gamma(\alpha_1 + \ldots + \alpha_N)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_N)} x_1^{\alpha_1 - 1} \ldots x_N^{\alpha_N - 1},$$

where Γ is the usual Gamma function, $\Gamma(z) \equiv \int_0^\infty t^{z-1} e^{-t}\, dt$. If X has a $\mathcal{D}_N(\alpha)$ distribution, then the expected value is $E(X_i) = \frac{\alpha_i}{\sum_{j=1}^N \alpha_j}$, the variance is $V(X_i) = \frac{\alpha_i\left(\sum_{k=1}^N \alpha_k - \alpha_i\right)}{\left(\sum_{k=1}^N \alpha_k\right)^2\left(1 + \sum_{k=1}^N \alpha_k\right)}$, and the covariance between two random variables is $COV(X_i, X_j) = -\sqrt{\frac{\alpha_i \alpha_j}{\left(\sum_{k=1}^N \alpha_k - \alpha_i\right)\left(\sum_{k=1}^N \alpha_k - \alpha_i\right)}}$.

The Dirichlet distribution for vectors of proportions is similar to the Normal distribution for continuous random variables. Both the Normal and the Dirichlet distributions are completely defined by their means and variances, and under very general conditions, the isoprobability contours for the Dirichlet distribution are convex, as they are ellipses for the Normal distribution.

A very useful property of the Dirichlet distribution is its ability to be readily updated after new information is learned. For example, in the most common use of the Dirichlet distribution, the values of the random variables $X_1, \ldots, X_N$ indicate the chances that some event will happen among the N possible ones. In that case, additional information about the true value of $X_1, \ldots, X_N$ is given by what event actually occurred, which can be coded by a vector with a 1 for the event that happened and zeros for all other $N - 1$ events. Suppose $X_1, \ldots, X_N$ has a $\mathcal{D}_N(\alpha_1, \ldots, \alpha_N)$ distribution and the first event is observed to have happened so that the new information is coded $[1, 0, \ldots, 0]$. Since $X_1 = P([1, 0, K, 0] \mid X_1, K, X_N)$, then

$$\begin{aligned} &P(X_1, \ldots, X_N \mid [1, 0, \ldots, 0]) \\ &\quad \propto P([1, 0, \ldots, 0] \mid X_1, \ldots, X_N) P(X_1, \ldots, X_N) \\ &\quad = X_1 \frac{\Gamma(\alpha_1 + \ldots + \alpha_N)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_N)} X_1^{\alpha_1 - 1} \ldots X_N^{\alpha_N - 1} \\ &\quad = \frac{\Gamma(\alpha_1 + \ldots + \alpha_N)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_N)} X_1^{(\alpha_1 + 1) - 1} \ldots X_N^{\alpha_N - 1} \end{aligned}$$

and $P(X_1, \ldots, X_N \mid [1, 0, \ldots, 0]) = \mathcal{D}_N(\alpha_1 + 1, \ldots, \alpha_N)$. This result is summarized in Lemma 1.

Lemma 1: If a set of probabilities $X_1, \ldots, X_N$ has a $\mathcal{D}_N(\alpha_1, \ldots, \alpha_N)$, then after observing the i^{th} event happening, the posterior distribution for $X_1, \ldots, X_N$ is still a Dirichlet distribution, with the parameter α_i corresponding to the i^{th} event increased by one. ◆

Another important property of Dirichlet distributions is their self-similar nature, summarized in Lemma 2. Self-similarity means that if the set of proportions $X_1, \ldots, X_N$ has a Dirichlet distribution and it is partitioned into q clusters, then the proportions inside each of those clusters also follow a Dirichlet distribution.

Lemma 2: Let $X = (X_1, \ldots, X_N)$ be a random vector of proportions that follows a Dirichlet distribution with parameter vector $\alpha = (\alpha_1, \ldots, \alpha_N)$. Let $(X_1, \ldots, X_N)$ be partitioned into two clusters, $(X_1, \ldots, X_M)$, and $(X_{M+1}, \ldots, X_N)$, with totals $S_1 = \sum_{i=1}^M x_i$ and $S_2 = \sum_{i=M+1}^N x_i$. Then:

- The vector of cluster totals (S_1, S_2) has a Dirichlet distribution with parameters $\beta_1 = \sum_{i=1}^M \alpha_i$ and $\beta_2 = \sum_{i=M+1}^N \alpha_i$, equal to the sums of the parameters corresponding to the variables included in each cluster.
- The vector of relative proportions inside cluster 1, $\left(\frac{x_1}{S_1}, \ldots, \frac{x_M}{S_1}\right)$, has a Dirichlet distribution with parameters $\left(\frac{\alpha_1}{\beta_1}, \ldots, \frac{\alpha_M}{\beta_1}\right) = (\gamma_1^{(1)}, K, \gamma_M^{(1)}) = \gamma^{(1)}$.
- The vector of relative proportions inside cluster 2, $\left(\frac{x_{M+1}}{S_2}, \ldots, \frac{x_N}{S_2}\right)$ has a Dirichlet distribution with parameters $\left(\frac{\alpha_{M+1}}{\beta_2}, \ldots, \frac{\alpha_N}{\beta_2}\right) = (\gamma_1^{(2)}, K, \gamma_{N-M}^{(2)}) = \gamma^{(2)}$.
- The random vectors (S_1, S_2), $\left(\frac{x_1}{S_1}, \ldots, \frac{x_M}{S_1}\right)$, and $\left(\frac{x_{M+1}}{S_2}, \ldots, \frac{x_N}{S_2}\right)$ are independent.

Moreover, the lemma would be true also for any number of clusters $q > 2$. ◆

Proof: See Aitchison (1986). ◆

Corollary: If $P(X)$ is a Dirichlet distribution, then it can be written in a hierarchical form as $P(X) = \mathcal{D}_N(\alpha) = \mathcal{D}_2(\beta) \times \mathcal{D}_M(\gamma^{(1)}) \times \mathcal{D}_{N-M}(\gamma^{(2)})$. ◆

Proof: By lemma 2, (S_1, S_2), $\left(\frac{x_1}{S_1}, \ldots, \frac{x_M}{S_1}\right)$, and $\left(\frac{x_{M+1}}{S_2}, \ldots, \frac{x_N}{S_2}\right)$ are independent, so

$$\begin{aligned} P(X) = P(X_1, K, X_N) &= P\left(S_1, S_2, \frac{X_1}{S_1}, \ldots, \frac{X_M}{S_1}, \frac{X_{M+1}}{S_2}, \ldots, \frac{X_N}{S_2}\right) \\ &= P(S_1, S_2) \times P\left(\frac{x_1}{S_1}, \ldots, \frac{x_M}{S_1}\right) \times P\left(\frac{x_{M+1}}{S_2}, \ldots, \frac{x_N}{S_2}\right) \\ &= \mathcal{D}_2(\beta) \times \mathcal{D}_M(\gamma^{(1)}) \times \mathcal{D}_{N-M}(\gamma^{(2)}). \blacklozenge \end{aligned}$$

This self-similarity property makes computations based on the Dirichlet distribution simpler than for any other probability distribution for proportions. In particular, it makes the process of Bayesian updating much more manageable computationally, as theorem 1 in section"Query Set Optimization" below will show. On the other hand, using a Dirichlet distribution to model the system's knowledge about the relevance vector imposes certain constraints that need to be evaluated in each particular case. By using the Dirichlet distribution, we accept that:

- the AIR system assumes that the user distributes relevance scores among the different document clusters independently of the relevance among the documents inside each cluster, and

- the AIR system also assumes that the user distributes relevance scores among the documents inside a cluster independently of the relevance among the documents inside another cluster.

The User/System Dialog

Let $D = \{d_1, \ldots, d_M\}$ be the set of documents in the collection, and $C = \{c_1, \ldots, c_N\}$ be the set of available clusters of documents for which appropriate summaries can be generated. In our presentation, we assume that the set of clusters is static and given to the IR system, but the results can be easily generalized for the case when the clustering is dynamic and updated by the system at each step of its interaction with the user.

Beyond the scope of this article is the matter of how to create summaries for clusters of documents. We are aware of the recent developments in this field and will incorporate that research into our implementation of AIR systems in future work.

The process of retrieving information from the collection D is initiated by a user U who sends a query Q_0 (*initial query*). To decide what documents to retrieve and in what order to present them to the user, any IR system has to estimate the information needs of the user. In other words, the system has to estimate the relevance vector θ, either implicitly or explicitly.

Following the probabilistic approach to IR (Crestani, Lalmas, Van Rijsbergen, & Campbell, 1998), the system's uncertainty about the true value of θ is reflected on a probability distribution $P(\theta) = f(I_D, I_Q, I_U)$, where I_D is the information available to the system about the documents in the collection, I_Q is the information available about the initial query by the user, and I_U is the information available about the user. Examples of I_D would be the word-frequency distribution of each document in the collection or the matrix of links between hypertext documents. Examples of I_Q would be what words are included in the query and in what order, while an example of I_U would be the history of queries by the same user.

Extensive research has gone into determining how to best define the information sets I_D, I_Q, and I_U, and what f function will generate more user satisfaction (Baeza-Yates & Ribeiro-Neto, 1999). Our proposal for an active IR system is to take such $P(\theta)$ as a starting point in the retrieval process and use a dialog with the user as a means to generate better estimates of what the information needs of the user are.

Let $P_0(\theta)$ be the available probability distribution after the initial query by the user. The key question then for an optimal dialog between user and system is: given $P_0(\theta)$, what question should the system ask the user so as to reduce as much as possible its uncertainty about the relevance vector θ?

The optimal question depends on what type of questions (reverse queries) the system can ask the user. What is optimal when the system can only ask the user to compare the relevance of two documents at a time will probably not be optimal if more complex questions are allowed.

The optimal question also depends on $P_0(\theta)$. This is just a corollary of the general principle that to choose what to ask next, one should have a guess about what the answer to each possible question is going to be; for instance, in a game of 20 questions to guess a number between 1 and 100, the optimal first question depends on our state of knowledge about what the chosen number is. If we have no prior information and all values from 1 to 100 have the same probability of being chosen, then a reasonable first question would be, "Is the number greater than 50?" However, if we already know that the chosen number is between 91 and 100, then a reasonable first question would be, "Is the number greater than 95?"

Finally, the optimal next question also depends on how we measure the system's uncertainty about the user's needs, the quantity that this next question is supposed to reduce as much as possible.

The interaction between user and AIR system proceeds as follows: based on the initial probability distribution $P_0(\theta)$, the system chooses a reverse query $Q_1 = Q_1(I_D, I_Q, I_U, P_0)$ from the space of possible reverse queries $\mathcal{Q}$. The user will answer Q_1 with a response $R_1(\theta)$ from the space of possible user responses $\mathcal{R}$, and the system will then update $P_0(\theta)$ to $P_1(\theta) = g(I_D, I_Q, I_U, R_1)$. After that, the system will decide the next reverse query, and so on. This iterative process will continue until stopped either by the user or by the system. Finally, the latest update of $P(\theta)$ will be used by the system to decide on an information presentation $T(I_D, I_Q, I_U, P)$ from the space of potential presentations $\mathcal{T}$. In most cases, the final information presentation T will be related to $\hat{\theta} = E[\theta]$, the system's estimate of the relevance vector. Examples of a final presentation T would be displaying to the user only those documents for which $\hat{\theta}$ is larger than a certain threshold value or only the 10 documents with the highest values for $\hat{\theta}$.

The Space of Reverse Queries

Both the degree of effectiveness of an active IR system and its complexity crucially depend on the type of reverse queries ($\mathcal{Q}$) that are allowed.

In principle, an active IR system could ask the user a reverse query conditional on all the information available about the documents in the collection (I_D), the user (I_U), the initial query (I_Q) or $P(\theta)$, the system's current knowledge about the user's information needs. However, in this presentation of the AIR framework, we will concentrate for simplicity on the basic reverse query strategies, where the system's next reverse query is based only on $P(\theta)$. It is worth noticing that this formulation is not overly restrictive because $P(\theta)$ incorporates the information contained in the initial user query and all the system/user interactions so far.

We consider only a relatively simple type of reverse query, called *contrastive selection* reverse queries in Jaakkola and Siegelmann (2001). In a contrastive selection reverse query, the AIR system chooses a query set S containing the clusters to be presented to the user, and the user is expected to choose only one cluster.

In each step, therefore, our particular AIR system proceeds as follows:

1. it finds a small subset S of clusters to present to the user;
2. it waits until the user selects one of the presented clusters;
3. it uses the evidence from the user's selections to update its belief of the user's information needs; and
4. it outputs the top documents so far, ranked by their estimated relevance score.

The iteration continues until terminated by the user or the system.

In every iteration, the system presents to the user a query Q_S, defined by the query set $S = \{S_1, \ldots, S_k\}$ containing k document clusters from the collection. For the time being, k, the size of the query sets, is assumed to be fixed, although in practice it will be chosen by the user. The user's response then can be written as $R = r$, where $1 \leq r \leq k$ is the cluster included in the query set that is most relevant to the user.

We assume that the user will compute relevance values $(\lambda_1, \ldots, \lambda_k)$ for all the document clusters in the query set by adding up the relevance values of the documents included in each one and then select the i^{th} cluster as the most relevant with probability $\lambda_i/(\lambda_1 + L + \lambda_k)$, its relative relevance. In future work, we will allow for users who are inconsistent in their answers or who base their responses on just the most relevant document in each cluster.

Hierarchical Representation of P(θ)

We assume that the knowledge of the system about the relevance vector θ can be represented by the Dirichlet probability distribution $P(\theta)$. Then, for each partition of θ, such distribution can be written in a hierarchical fashion, as shown by the corollary to Lemma 2.

When the system poses a reverse query Q_S based on the query set $S = \{S_1, \ldots, S_k\}$, a three-level partition of θ is defined: θ is partitioned into two (first-level) clusters, those documents belonging to the clusters included in the query set and those excluded; the first of those two clusters is again partitioned into k (second-level) clusters, $S_1, \ldots, S_k$. Finally, each cluster included in the query set is partitioned into the documents that it contains. This partition is illustrated in Figure 1.

This three-level partition of θ leads to a three-level hierarchical representation of $P(\theta)$ as the product $P(\theta^{(1)})P(\theta^{(2)}_{\bullet|1})P(\theta^{(2)}_{\bullet|2})[\prod_{l=1}^{k} P(\theta^{(3)}_{\bullet|l})]$, where $\theta^{(1)}$ represents first-level cluster totals, $\theta^{(2)}_{\bullet|1}$ and $\theta^{(2)}_{\bullet|2}$ stand for second-level proportions inside each of the two first-level clusters, and the $\theta^{(3)}$ terms correspond to the proportions inside the second-level clusters. From Lemma 2, each of the factors follows a Dirichlet distribution, and their parameters are readily obtained by gathering the appropriate components of the original parameter vector α; for example, $\alpha^{(1)}_0 = \sum_{x \in S} \alpha_x$; $\alpha^{(2)}_{j|1} = \sum_{x \in s_j} \alpha_x$, for $j = 1, \ldots, k$; $\alpha^{(2)}_{x|2} = \alpha_x$, for $x \notin S$; and $\alpha^{(3)}_{x|j} = \alpha_x$, whenever $x \in s_j, j = 1, \ldots, k$.

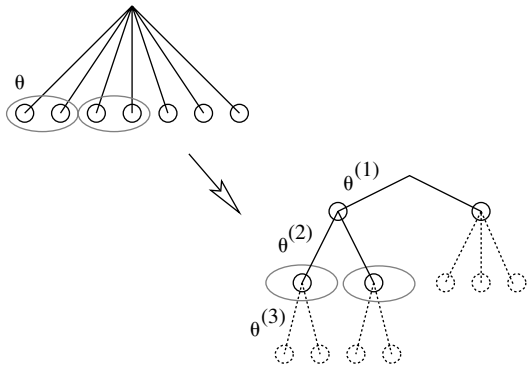

FIG. 1. A three-level hierarchical partition of the relevance vector θ.

Given $P(\theta)$, a reverse query Q_S, defined by the query set $S = \{S_1, \ldots, S_k\}$, and a user response $R = r$, it is easy to evaluate $P(\theta \mid Q_S, R)$ using Lemma 1, as shown in the next section.

Query Set Optimization

Our optimization criterion for choosing the query set S is the information that we stand to gain from querying the user with it. In other words, we will choose the reverse query that we expect will decrease, the most, our uncertainty about the relevance vector.

The mutual information between the user's response R and the relevance vector θ will be shown in theorem 1 below to be a simple function that is the sum of $k + 1$ terms, where k is the size of the query set. Each of those $k + 1$ terms depends on the Dirichlet coefficients (α's) of our best guess about θ so far, and the probability distribution of R.

To choose the best reverse query to be posed to the user, the system theoretically needs to go over all allowed subsets of clusters and calculate which one is expected to have the greatest impact in reducing the uncertainty about the information needs of the user. As in Jaakkola and Siegelmann (2001), we will measure the system's uncertainty level about the user's needs by using the entropy function H.

We can also optimize the choice of S with an approximation method that successively finds the next-best cluster to include in the query set. This algorithm scales as $O(Nk)$, where N is the number of clusters in our collection and k is the size of the query set.

Theorem 1. The mutual information between R and θ is given by

$$I(R; \theta) = H(R) + \sum_{r=1}^{k} \frac{(\alpha^{(2)}_{l|1} + L + \alpha^{(2)}_{k|1})}{\alpha^{(2)}_{r|1}} \theta^{(2)}_{r|1} P(R = r) \times \log\left[\frac{(\alpha^{(2)}_{l|1} + L + \alpha^{(2)}_{k|1})}{\alpha^{(2)}_{r|1}} \theta^{(2)}_{r|1} P(R = r) \right]$$

Proof: The user response (R) to the reverse query posed by the system is related only to the second level of the hierarchical representation of $P(\theta)$; therefore, $I(R; \theta) = I(R; \theta^{(2)}_{\bullet|1})$.

By definition of mutual information,

$$I(R;\theta^{(2)}_{\bullet|1}) = H(R) - H(R\,|\,\theta^{(2)}_{\bullet|1})$$
$$= H(R) + \sum_{r=1}^{k} P(R=r|\,\theta^{(2)}_{\bullet|1})\log P(R=r|\theta^{(2)}_{\bullet|1}) \qquad (1)$$

where the response $R = r$ corresponds to the user choosing the r^{th} cluster in the query set as the most relevant.

By Bayes rule, $P(R=r|\theta^{(2)}_{\bullet|1}) = \dfrac{P(\theta^{(2)}_{\bullet|1}|R=r)P(R=r)}{P(\theta^{(2)}_{\bullet|1})}$,

and we know from Lemma 1 that because $P(\theta^{(2)}_{\bullet|1})$ is a Dirichlet distribution, $P(\theta^{(2)}_{\bullet|1}|R=r)$ is also a Dirichlet distribution where the parameter corresponding to the cluster chosen by the user was increased by an amount c_s that depends on the query sets while all other parameters remain unchanged. Formally,

$$P(\theta^{(2)}_{\bullet|1})\colon \mathcal{D}(\alpha^{(2)}_{\bullet|1}),\ P(\theta^{(2)}_{\bullet|1}|R=r)\colon \mathcal{D}(\beta^{(2)}_{\bullet|1}),\ \beta^{(2)}_{r|1}$$
$$= \alpha^{(2)}_{r|1} + c_s \quad \text{and} \quad \forall r' \neq r\colon \beta^{(2)}_{r'|1} = \alpha^{(2)}_{r'|1}.$$

with $c_s = \sum_{RS} \theta_\kappa$

Thus, on one hand,

$$P(\theta^{(2)}_{\bullet|1}) = \mathcal{D}(\alpha^{(2)}_{\bullet|1}) = \frac{\Gamma(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1})}{\Gamma(\alpha^{(2)}_{l|1})\ldots\Gamma(\alpha^{(2)}_{k|1})}\,\theta^{(2)\alpha^{(2)}_{l|1}-1}_{l|1}\ldots\theta^{(2)\alpha^{(2)}_{k|1}-1}_{k|1},$$

and on the other hand,

$$P(\theta^{(2)}_{\bullet|1}|R=r) = \mathcal{D}(\beta^{(2)}_{\bullet|1})\ldots$$
$$= \frac{\Gamma(\beta^{(2)}_{l|1} + \ldots + \beta^{(2)}_{k|1})}{\Gamma(\beta^{(2)}_{l|1})\ldots\Gamma(\beta^{(2)}_{k|1})}\,\theta^{(2)\beta^{(2)}_{l|1}-1}_{l|1}\ldots\theta^{(2)\beta^{(2)}_{k|1}-1}_{k|1}$$
$$= \frac{\Gamma(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1} + c_s)}{\Gamma(\alpha^{(2)}_{l|1})\ldots\Gamma(\alpha^{(2)}_{k|1})\left(\dfrac{\Gamma(\alpha^{(2)}_{r|1} + c_s)}{\Gamma(\alpha^{(2)}_{r|1})}\right)}$$
$$\times\ \theta^{(2)\alpha^{(2)}_{l|1}-1}_{l|1}\ldots\theta^{(2)\alpha^{(2)}_{k|1}-1}_{k|1}\theta^{(2)}_{r|1}c_s$$

leading to

$$\frac{P(\theta^{(2)}_{\bullet|1}|R=r)}{P(\theta^{(2)}_{\bullet|1})} = \frac{\Gamma(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1} + c_s)\Gamma(\alpha^{(2)}_{r|1})}{\Gamma(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1})\Gamma(\alpha^{(2)}_{r|1} + c_s)}\,\theta^{(2)}_{r|1}c_s$$
$$= \frac{(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1})}{\alpha^{(2)}_{r|1}}\,\theta^{(2)}_{r|1}c_s \quad \text{and}$$

$$P(R=r|\theta^{(2)}_{\bullet|1}) = \frac{P(\theta^{(2)}_{\bullet|1}|R=r)P(R=r)}{P(\theta^{(2)}_{\bullet|1})}$$
$$= \frac{(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1})}{\alpha^{(2)}_{r|1}}\,\theta^{(2)}_{r|1}c_s P(R=r) \quad (2)$$

Combining (1) and (2), we get our final result,

$$I(R;\theta^{(2)}_{\bullet|1}) = H(R) + \sum_{r=1}^{k} P(R=r|\theta^{(2)}_{\bullet|1})\log P(R=r|\theta^{(2)}_{\bullet|1})$$
$$= H(R) + \sum_{r=1}^{k} \frac{(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1})}{\alpha^{(2)}_{r|1}}\,\theta^{(2)}_{r|1}\,c_s P(R=r)$$
$$\times \log\left[\frac{(\alpha^{(2)}_{l|1} + \ldots + \alpha^{(2)}_{k|1})}{\alpha^{(2)}_{r|1}}\,\theta^{(2)}_{r|1}c_s P(R=r)\right].\blacklozenge$$

Comparison to Previous Work

There are two well-known approaches to IR where the system tries to improve the results of an initial query-based search. One is the Scatter/Gather algorithm. The other is the relevance feedback approach. Both these approaches can be seen as special cases of our AIR, which are more limited in two aspects:

1. our system chooses optimally the reverse queries to pose to the user, whereas the others are typically sub-optimal, and
2. our AIR system incorporates the user's responses to solve an estimation problem where the goal is to recover the unknown document weights or relevance assessments in terms of probabilities, which is particularly superior in cases where users make a few errors.

In the Scatter/Gather algorithm for browsing information systems (Cutting, Karger, Pederson, & Tukey, 1996; Hearst, Karger, & Pedersen, 1995), the reverse queries are constrained to be a fixed number of clusters that contain all the documents being considered by the system, excluding all documents contained in less-relevant clusters. When documents are not chosen, they are permanently removed from the pool, making this technique highly sensitive to any user's error.

Relevance feedback IR systems (Rocchio, 1971; Salton & Buckley, 1990) use a much smaller space of possible reverse queries than the AIR. A very simple AIR system will present, as a reverse query, n documents and ask the user to mark them with 1's (for relevant documents) and 0's (for irrelevant documents). The documents presented to the user could be the ones with highest $\hat{\theta}$'s (occupying the top of the ranking so far) or not, depending of what reverse query is predicted to be the most informative. On the other hand, a relevance feedback system is constrained to present only the n top-ranked documents to be marked by the user.

Recall the example in the "Example" section, where the query "apple" returns 120 documents, and where the system first ranks all the documents about computers, while the plants and cooking documents are ranked in positions 101 to 120. An Active IR system could ask as the first reverse query to mark documents number 1 (the highest-ranked computer document) and 101 (the highest-ranked plant and cooking document). The user's answer to this first reverse query would indicate what type of documents should be

ranked higher. On the other hand, a Relevance Feedback IR system would present the first n (e.g., 10) documents according to the initial ranking. No matter what the user's response, the system still would not know if the "computer" or "plant" documents are the most relevant for that particular user.

It would take only one reverse query for the Active IR system to concentrate on the plant and cooking documents, whereas the Relevance Feedback IR system would need at least 10 user interactions. The AIR is able to collect useful information more efficiently.

Discussion and Future Work

An active information retrieval paradigm is particularly superior to other methods when the initial user query cannot reflect the specific information needs.

Complex document collections, or ones with which the user is unfamiliar, constitute cases in which optimal initial queries are difficult to be formed and where a dialog will reflect more accurately the information needs of the user. Disaster management is another case requiring dialog because the information needs are non-ergodic in the sense of being divergent from and following different paths than those at normal times. All standard (non-interactive) information retrieval methods are ergodic, predicated on the assumption that the present is very much like the past and will generate sub-optimal results in the face of non-ergodic information needs. Here, the AIR will fully demonstrate its adaptability to the information needs of the user.

Our next implementation is in Bioinformatics, where the documents in the collection do not have a natural similarity metrics among them because documents may be images, texts, or gene files. There, the application would be to rank the genes according to their potential as fruitful targets for further study, and reverse queries will be requesting the user to perform additional experiments with these genes.

There are a number of extensions to the approach presented in this fundamental article that we plan to explore. We will consider more-sophisticated reverse queries, such as asking the user to rank the clusters. We will allow for some errors from the user. We will analyze the trade-offs between the size of the query set (resource constraints) and the expected completion time of the retrieval process. Finally, while the Dirichlet distribution has been used to describe compositional data in disciplines such as geology, biology, ecology, economics, and chemistry, we will explore its limitations and implement our AIR system using several alternatives to the Dirichlet distribution to allow for general searches to be encompassed by the AIR paradigm.

References

Aitchison, J. (1986). The statistical analysis of compositional data. London: Chapman & Hall.

Atkinson, A.C., & Donev, A.N. (1992). Optimum experimental designs. New York: Clarendon Press.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern information retrieval. Harlow, UK: Addison Wesley.

Blei, D.M., Jordan, M.I., & Ng, A.Y. (2003). Hierarchical Bayesian models for applications in information retrieval. In J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, & M. West (Eds.), Bayesian statistics 7 (pp. 25–44). Oxford, UK: Clarendon Press.

Cohn, D.A., Ghahramani, Z., & Jordan, M.I. (1996). Active learning with statistical models. Journal of Artificial Intelligence Research, 4, 129–145.

Crestani, F., Lalmas, M., Van Rijsbergen, C.J., & Campbell, I. (1998). "Is this document relevant?… Probably": A survey of probabilistic models in information retrieval. ACM Computing Surveys, 30(4), 528–552.

Croft, W.B. (1980). A model of cluster searching based on classification. Information Systems, 5, 189–195.

Cutting, D.R., Karger, D.R., Pederson, J.O., & Tukey, J.W. (1996). Scatter/gather: A cluster based approach to browse document collections. Proceedings of the Fifteenth Annual International ACM SIGIR Conference (pp. 318–329). New York: ACM.

Fedorov, V.V. (1972). Theory of optimal experiments. New York: Academic Press.

Geiger, D., & Heckerman, D. (1996). A characterization of the Dirichlet distribution with application to learning Bayesian networks. In K.M. Hanson & R. N. Silver (Eds.), Maximum entropy and Bayesian methods. Beachwood, OH: Institute of Mathematical Statistics.

Geiger, D., & Heckerman, D. (1997). A characterization of the Dirichlet distribution through global and local independence. The Annals of Statistics, 25, 1344–1369.

Hearst, M.A., Karger, D.R., & Pedersen, J.O. (1995). Scatter/gather as a tool for the navigation of retrieval results. Working Notes of the 1995 AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval.

Hsu, C.-N., Huang, H.-J., & Wong, T.-T. (2003). Implications of the Dirichlet assumption for discretization of continuous variables in naïve Bayesian classifiers. Machine Learning, 53(3), 235–263.

Jaakkola, T., & Siegelmann, H. (2001). Active information retrieval. Advances in Neural Information Processing Systems, 14, 777–784.

Lu, I.-L. (2002). The Dirichlet-multinomial model for Bayesian information retrieval. Hawaii International Conference on Statistics and Related Fields.

Rocchio, J.J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), The smart retrieval system—Experiments in automatic document processing (pp. 313–323). New York: Prentice Hall.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41(4), 288–297.

Tombros, A., Villa, R., & Van Rijsbergen, C.J. (2002). The effectiveness of query-specific hierarchic clustering in information retrieval. Information Processing and Management, 38, 559–582.

Zhang, C., & Chen, T. (2001). Active learning for information retrieval: Using 3D models as an example (Technical Report AMP 01-04). Pittsburgh, PA: Electrical and Computer Engineering, Carnegie Mellon University.