Query Formulation in Linear Retrieval Models

S. K. M. Wong and Y. Y. Yao
Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2

The subject of query formulation is analyzed within the framework of adaptive linear models. Our study is based on the notions of user preference and an acceptable ranking strategy. Such an approach enables us to adopt a gradient descent algorithm to formulate the query vector by an inductive process. We also present a critical analysis of the existing relevance feedback and the probabilistic approaches. It is shown that Rocchio's method is a special case of our linear model and the independence assumption may be stronger than required for a linear system. Our method has the added advantages that it is applicable to both nonbinary document representation and a user preference relation inducing more than two classes.

Introduction

The primary objective of an information retrieval system is to identify the useful documents in response to a user's information request. Many retrieval models (Salton & McGill, 1983; Van Rijsbergen, 1979) have been proposed for this purpose. Among these, the linear models are perhaps best known. A linear decision function, the cosine similarity measure, was first used in the vector space model (Salton & McGill, 1983; Salton, 1971). Another linear system is the binary probabilistic independence model (Van Rijsbergen, 1979; Robertson & Sparck Jones, 1976; Yu & Salton, 1976). Both of these models and their variants have been studied extensively (Salton & McGill, 1983; Van Rijsbergen, 1979; Salton, 1971; Robertson & Sparck Jones, 1976; Yu & Salton, 1976; Raghavan & Wong, 1986; Wong, Ziarko, & Wong, 1985; Wong & Yao, 1987; Croft & Harper, 1977). The main advantage of a linear model lies in its simplicity and also its ability to provide acceptable performance. However, a number of important design and theoretical problems have remained to some extent unresolved. This article is an attempt to clarify some of the outstanding issues in linear retrieval models and hopefully through our discussions to provide useful insights for developing a more effective information retrieval system.

In this article, our focus is on the subject of query formulation which is being analyzed within the framework of adaptive linear models (Bollman & Wong, 1987). We will first introduce the notion of *user preference* which forms the basis of our approach to inductive learning. An adaptive linear retrieval model is then suggested based on an *acceptable* ranking strategy. Such a strategy enables us to adopt a gradient descent algorithm for inferring the query vector from a training set of documents provided that the user preference satisfies certain conditions.

After the basic concepts are defined and clarified, we present a critical analysis of the relevance feedback vector model. It is shown that the method originally proposed by Rocchio (1971) is in fact a special case or a first-order approximation of our linear model. We point out some of the difficulties in the gradient descent procedure suggested by Van Rijsbergen (1979). It seems unnecessary to introduce a threshold parameter in this method when one is primarily interested in document ranking rather than pattern classification. The relationship between our approach and the binary probabilistic independence model is also discussed. Our method seems to be less restrictive and is applicable to nonbinary document representation. An additional advantage of the proposed linear model is that it is not necessary to use the usual statistical independence assumption in our formulation. We conclude our discussions by emphasizing some of the difficulties encountered in the design of an adaptive system.

Preference Relation and Utility Function

Given two documents in a collection, a user would, more often than not, be able to decide which document is more useful based on his preference. It is also possible that the user may regard these two documents as belonging to the same category or as having the same value and therefore be unable to differentiate them. This *user preference* or judgment is largely governed by the content of the individual document and the type of information the user seeks. One of the objectives of this article is to show that the notion of user preference provides a better understanding to facilitate the development of information retrieval systems.

The concept of user preference was discussed in earlier literature on information retrieval although its usefulness

Received February 8, 1988; revised April 6, 1988; accepted May 23, 1988.

has perhaps not been fully explored. Robertson and Sparck Jones (1976) pointed out that user preference can be used as a basis for computing term weights. Bookstein (1983b) suggested using the preference structure to estimate the expected cost of retrieving a document. More recently, based on measurement theory, Bollmann and Wong (1987) discussed the necessary and sufficient conditions on the user preference to justify using a linear decision function in retrieval models.

To establish the framework for subsequent discussions, we will first clarify some of the important concepts about user preference.

Let D denote a finite set of documents. A user preference can be formally described by a relation $<\cdot$ on D: for $d, d' \in D$,

$$d <\cdot\, d' \iff \text{the user prefers } d' \text{ to } d. \qquad (1)$$

The relation $<\cdot$ is called a (strict) *preference relation* which reflects the user judgment on a set of documents. If $d <\cdot\, d'$ holds, one can say that d' is more relevant to the user than d. However, this preference relationship may not necessarily be defined between *any* two documents in a collection. In other words, for two arbitrary documents, $d, d' \in D$, it may happen that neither $d <\cdot\, d'$ nor $d' <\cdot\, d$ holds. In this case, an *indifference relation* $\sim$ on D can be defined as follows:

$$d \sim d' \iff (\text{not } (d <\cdot\, d'), \text{ not } (d' <\cdot\, d)). \qquad (2)$$

This indifference relationship between documents d and d' may arise in several ways. For instance, the user may consider d and d' equally useful to him. Another possibility is that d and d' are *incomparable* from the user point of view. This situation may occur when the user is asked to choose between two documents which are both irrelevant to his information needs.

In conventional information retrieval models, documents are often divided into two disjoint subsets, a relevant set *rel* and a nonrelevant set *nrel*. This two-value scheme of classification is perhaps the simplest way to describe a preference relation, namely:

$$d <\cdot\, d' \iff (d \in nrel,\ d' \in rel), \qquad (3)$$

and similarly the indifference relation can be specified as:

$$\begin{aligned} d \sim d' \iff & (d \in rel,\ d' \in rel) \quad \text{or} \\ & (d \in nrel,\ d' \in nrel). \qquad (4) \end{aligned}$$

Such a description of user preference may be too restrictive because a user may wish to divide the documents into more than just two classes according to his preference.

So far we have been discussing the concepts of user preference in general terms without making specific assumptions. In order to simplify the problem, we will only consider hereafter a special kind of preference relation. A preference relation $<\cdot$ satisfying the following axioms:

(1) If $d <\cdot\, d'$, then not $(d' <\cdot\, d)$;
(2) If not $(d <\cdot\, d')$ and not $(d' <\cdot\, d'')$, then not $(d <\cdot\, d'')$,
(5)

is called a *weak order*. It can be seen that $\sim$ is an equivalence relation if the preference relation $<\cdot$ is a weak order. This equivalence relation partitions the documents into disjoint subsets.

Theorem 2.1

Suppose D is a finite set of documents and $<\cdot$ a relation on D. There exists a real-valued mapping $u{:}D \to \mathbf{R}$ satisfying the condition,

$$d <\cdot\, d' \iff u(d) < u(d') \qquad (6)$$

if and only if $<\cdot$ is a weak order. (u is uniquely defined up to a strictly monotonic transformation.) □

According to the above theorem (Roberts, 1976; Fishburn, 1970); if the preference relation $<\cdot$ is a weak order, it is then possible to assign a real number (a *utility*) to each document such that the sequence $u(d), u(d'), \cdots$ (arranged in ascending order) will indeed reflect the *order* of $d, d', \cdots$ under the preference relation $<\cdot$. Thus, the mapping u is called an order-preserving utility function. The utility $u(d)$ can be regarded as a measure of the relative *degree* of relevance of a document $d \in D$ with respect to a user preference relation. We can therefore rank the documents according to these utilities. The important point is that Theorem 2.1 provides a direct link between the notion of document ranking and that of a user preference relation satisfying the axioms of a weak order.

The axioms defined by equation (5) can be interpreted from two different points of view. The prescriptive or normative interpretation is concerned with the theoretical principles that a user must follow to specify his preference. From this perspective, the axioms are looked upon as conditions of rationality. A user cannot prefer d' to d and simultaneously prefer d to d'. If a user prefers neither d' to d nor d'' to d', it is reasonable to assume that he would not prefer d'' to d. In this case, Theorem 2.1 ensures that the preference of a *rational* user can be measured by a utility function. On the other hand, the descriptive interpretation treats the axioms as testable conditions. Whether or not one can rank the documents according to some value assigned to each document depends on whether or not the user preference relation is a weak order. An example is given below to demonstrate the existence of a utility function when the user preference relation is a weak order.

Example 2.1

Suppose a user preference relation $<\cdot$ on $D = \{d_1, d_2, d_3, d_4\}$ is specified by:

$$d_3 <\cdot\, d_1, \quad d_4 <\cdot\, d_1, \quad d_3 <\cdot\, d_2,$$
$$d_4 <\cdot\, d_2, \quad d_4 <\cdot\, d_3.$$

One can easily verify that the above relation is a weak order. The utility for each document can be calculated by the formula (Roberts, 1976):

$$u(d_i) = \text{the number of } d_j \text{ such that } d_j <\cdot\, d_i. \qquad (7)$$

Hence,

$$u(d_1) = 2, \qquad u(d_2) = 2, \qquad u(d_3) = 1, \qquad u(d_4) = 0,$$

which indeed satisfy the condition (6). That is, the sequence 0, 1, 2 of utilities reflects the order of the preference relation:

$$\{d_4\} <\cdot \{d_3\} <\cdot \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}.$$

From equation (2), it can be seen that the indifference relation $\sim$ has three equivalence classes $\{d_4\}, \{d_3\}, \{d_1, d_2\}$. According to the utilities computed, document(s) in the same equivalence class have the same rank. □

An Adaptive Linear System

One of the main tasks in information retrieval is to provide a ranking for the documents in accordance with a user preference so that the user would be able to find the required information by examining only the top-ranked documents. In the above discussions, we have shown that documents can be ordered by a utility function consistent with a preference relation if the relation is a weak order. Unfortunately, the user preference is not known *a priori* to the retrieval system. A user cannot *fully* specify his preference unless he has read *all* the documents. Obviously, this situation creates an apparent *impasse* in the design of an information retrieval system. However, one may adopt the inductive method—learning by example—to resolve such an impasse. Suppose the preference relationship is known within a sample of documents (a training set). A decision function (or a set of decision rules) can be inferred from this training set by an inductive process. That is, such a system can learn and refine the decision function which can be used to rank other documents in the collection. Many researchers (Van Rijsbergen, 1979; Salton, 1971; Rocchio, 1971; Wong & Ziarko, 1986; Bookstein, 1983a) have in fact applied different inductive methods to information retrieval. In the fifth section, we will give a detailed comparison between our approach and these methods.

To design an inductive retrieval system, the first step is to introduce a knowledge representation of documents. There are, of course, many ways to describe a document. One may characterize a document by its content, scope, etc. In this article, we assume that each document is represented by a set of unweighted or weighted index terms (keywords). That is, each document is mapped onto a vector in a p-dimensional vector space $\mathbf{R}^p$ spanned by the set of index terms $\{t_1, t_2, \ldots, t_p\}$:

$$h : D \longrightarrow \mathbf{R}^p \tag{8}$$

The mapping h may, for instance, be determined by using an autoindexing procedure (Salton, 1971). Each component of a vector indicates the importance of the corresponding index term in describing the document.

We know from Theorem 2.1 that there exists an order-preserving utility function u for a user preference if the relation $<\cdot$ is a weak order. Unfortunately, this theorem does not say how one can construct such a utility (decision) function in terms of document descriptions. Clearly, even if $<\cdot$ is a weak order, to find such a decision function consistent with a preference relation can easily become an intractable task without making some simplifying assumptions. For instance, in the probabilistic model the statistical independence approximation is used. In subsequent discussions, we assume that the decision function is linear (a first-order polynomial in terms of the components of the document vector). Based on the above document representation, one can now develop a method to construct a linear decision function.

Definition 3.1

Let $\mathbf{D} \subseteq \mathbf{R}^p$ be a finite set of column document vectors and $<\cdot$ a user preference relation on $\mathbf{D}$. We say that the preference relation $<\cdot$ is *linear* if there exists a query vector $\mathbf{q} \in \mathbf{R}^p$ such that for any $\mathbf{d}, \mathbf{d}' \in \mathbf{D}$,

$$\mathbf{d} <\cdot \mathbf{d}' \Longleftrightarrow \mathbf{q}^T\mathbf{d} < \mathbf{q}^T\mathbf{d}', \tag{9}$$

where $\mathbf{q}^T = (w_1, w_2, \ldots, w_p)$ denotes the transpose of $\mathbf{q}$. □

Condition 9 implies that

$$\mathbf{d} \sim \mathbf{d}' \Longleftrightarrow \mathbf{q}^T\mathbf{d} = \mathbf{q}^T\mathbf{d}'. \tag{10}$$

For a linear user preference relation, the utility function $u(\mathbf{d}) = \mathbf{q}^T\mathbf{d}$ enables us to rank the documents precisely in accordance with the user preference. Also, documents belonging to the same equivalence class of the indifference relation have the same utility. In this case we say that the linear function $u(\mathbf{d})$ provides a *perfect ranking*.

If a perfect ranking is the objective in the design of an information retrieval system, for a given preference relation defined on a training set of documents, then one has to formulate a query vector satisfying equations (9) and (10) exactly. However, there exist some serious difficulties in the construction of a linear decision function that would provide a perfect ranking even if the user preference relation is linear. An important question is whether perfect ranking is the strategy that should be adopted for information retrieval. We believe that a user would be satisfied as long as the preferred documents are ranked higher than the nonpreferred ones. Therefore, from the system design point of view, the perfect ranking strategy does seem to be stronger than required.

Definition 3.2

Let $\mathbf{D}$ be a finite set of document vectors and $<\cdot$ a user preference relation defined on $\mathbf{D}$. The preference relation $<\cdot$ is *weakly linear* if there exists a query vector $\mathbf{q}$ such that for any $\mathbf{d}, \mathbf{d}' \in \mathbf{D}$,

$$\mathbf{d} <\cdot \mathbf{d}' \Longrightarrow \mathbf{q}^T\mathbf{d} < \mathbf{q}^T\mathbf{d}'. \quad \square \tag{11}$$

The above definition implies that

$$\mathbf{q}^T\mathbf{d} \geq \mathbf{q}^T\mathbf{d}' \Longrightarrow \text{not } (\mathbf{d} <\cdot \mathbf{d}'). \tag{12}$$

The one-way implication in condition (11) has a significant impact on system design. In contrast to the perfect ranking strategy, this condition only guarantees that less preferred documents will not be listed in front of the preferred ones. However, documents in the same equivalence class of the indifference relation may not have the same utility. We say that such a decision function provides an *acceptable ranking*. It should be noted here that some of the ranking strategies suggested in the past are actually intended to provide an acceptable ranking rather than a perfect one (Salton, 1983; Van Rijsbergen, 1979; Cooper, 1968). In the next section, we will show how the query vector can be inferred from a training set of document vectors for a weakly linear preference relation.

Query Formulation By An Inductive Method

In this section, we discuss the problem of query formulation within the general framework of inductive learning. Our objective is to infer a query vector from a training set $\mathbf{S} \subseteq \mathbf{D}$.

It is understood in inductive learning that the decision rules or decision functions inferred from a training set may not necessarily be correct when applied to all objects in the universe of interest. In the context of information retrieval, this means that the query vector learned from a training set may not correctly classify all documents in the whole collection. However, by enlarging the training set, the system is expected to produce a more *accurate* query vector if the user preference relation is weakly linear. At each phase of the inductive process additional samples can be selected from the remaining documents and an improved query vector is thus produced. This process can be repeated until the user is satisfied with what has been retrieved.

Suppose a user preference relation $<\cdot$ on a set of document vectors $\mathbf{D}$ is weakly linear. (Obviously, this assumption implies that $<\cdot$ is also weakly linear on any subset of $\mathbf{D}$). With the acceptable ranking strategy, we now develop an inductive method to formulate a query vector $\mathbf{q}$ from a training set $\mathbf{S}$ of document vectors selected from $\mathbf{D}$.

Consider two document vectors $\mathbf{d}, \mathbf{d}' \in \mathbf{S}$ such that $\mathbf{d} <\cdot \mathbf{d}'$ holds. By definition 3.2, $\mathbf{d} <\cdot \mathbf{d}'$ implies $\mathbf{q}^T\mathbf{d} < \mathbf{q}^T\mathbf{d}'$ or $\mathbf{q}^T\mathbf{b} > 0$, where $\mathbf{b} = \mathbf{d}' - \mathbf{d}$ is called a *difference* vector. Given a training set of documents, the set $\mathbf{B}$ of all difference vectors is defined as:

$$\mathbf{B} = \{\mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d}, \mathbf{d}' \in \mathbf{S} \text{ and } \mathbf{d} <\cdot \mathbf{d}'\}. \tag{13}$$

It is therefore clear that the problem of finding a query vector $\mathbf{q}$ satisfying the acceptable ranking strategy (condition (11)) is equivalent to solving the following system of linear inequalities:

$$\mathbf{q}^T\mathbf{b} > 0, \qquad \text{for every } \mathbf{b} \in \mathbf{B}. \tag{14}$$

Under the assumption that $<\cdot$ is weakly linear, there exists a solution vector $\mathbf{q}$ which satisfies equation (14). However, there may exist more than one solution for a given training set. (The region containing all solution vectors is referred to as the solution region.) It is understood in our formulation that *any* vector in the solution region would provide an acceptable ranking. Thus, our main concern here is to show how to find such a solution vector.

Many algorithms have been proposed for solving a system of linear inequalities. The approach adopted here is based on the minimization of a suitably chosen scalar function. Consider a query vector $\mathbf{q}$ and a difference vector $\mathbf{b} = \mathbf{d}' - \mathbf{d} \in \mathbf{B}$. If $\mathbf{q}^T\mathbf{b} > 0$, then we say that the vector $\mathbf{q}$ correctly specifies the preference relationship between $\mathbf{d}$ and $\mathbf{d}'$. If $\mathbf{q}^T\mathbf{b} \leq 0$, then an error occurs. In this case, the value $-\mathbf{q}^T\mathbf{b} \geq 0$ can be interpreted as a measure of the error. Therefore, the total error with respect to the vector $\mathbf{q}$ can be written as:

$$J(\mathbf{q}) = \sum_{\mathbf{b} \in \Gamma(\mathbf{q})} - \mathbf{q}^T\mathbf{b}, \tag{15}$$

where $\Gamma(\mathbf{q})$ is defined by:

$$\Gamma(\mathbf{q}) = \{\mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d}, \mathbf{d}' \in \mathbf{S}, \mathbf{d} <\cdot \mathbf{d}' \text{ and } \mathbf{q}^T\mathbf{b} \leq 0\} \subseteq \mathbf{B}. \tag{16}$$

We define $J(\mathbf{q}) = 0$ if $\Gamma(\mathbf{q}) = \emptyset$. Since $\mathbf{q}^T\mathbf{b} \leq 0$ for every $\mathbf{b} \in \Gamma(\mathbf{q})$, $J(\mathbf{q})$ is nonnegative. In particular, $J(\mathbf{q})$ is equal to 0 only if $\mathbf{q}$ is a solution vector (i.e., $\Gamma(\mathbf{q}) = \emptyset$) or $\mathbf{q}$ is on the boundary of the solution region. ($J(\mathbf{q})$ is called the *perceptron criterion function* in pattern recognition (Duda & Hart, 1973)).

A gradient descent procedure can be used to search for a query vector that minimizes the function $J(\mathbf{q})$ and at the same time satisfies the condition $\Gamma(\mathbf{q}) = \emptyset$. In this method, we start with an arbitrary vector $\mathbf{q}_0$ and compute the corresponding gradient vector. The next vector $\mathbf{q}_1$ is obtained by moving some distance from $\mathbf{q}_0$ in the direction of steepest descent (i.e., along the negative of the gradient), and so on. Let $\nabla J(\mathbf{q}_k)$ denote the gradient vector in the $(k + 1)$th iteration. The vector $\mathbf{q}_{k+1}$ can be expressed as:

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \alpha_k \nabla J(\mathbf{q}_k), \tag{17}$$

where α_k is a positive number that sets the step size.

For the perceptron criterion function (15), the gradient vector $\nabla J(\mathbf{q}_k)$ is defined by:

$$\nabla J(\mathbf{q}) = - \sum_{\mathbf{b} \in \Gamma(\mathbf{q})} \mathbf{b}. \tag{18}$$

Based on the above gradient vector, we adopt the following gradient descent algorithm (with $\alpha_k = 1$) to find a solution vector for the system of linear inequalities (14):

(1) Choose an initial query vector $\mathbf{q}_0$ and let $k = 0$;

(2) Let $\mathbf{q}_k$ be the query vector in the $(k + 1)$th iteration; identify the following set of difference vectors:

$$\Gamma(\mathbf{q}_k) = \{\mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d}, \mathbf{d}' \in \mathbf{S}, \mathbf{d} <\cdot \mathbf{d}' \text{ and } \mathbf{q}_k^T\mathbf{b} \leq 0\} \subseteq \mathbf{B}; \tag{19}$$

If $\Gamma(\mathbf{q}_k) = \emptyset$ (i.e., $\mathbf{q}_k$ is a solution vector), terminate the procedure;

(3) Let

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \nabla J(\mathbf{q}_k) = \mathbf{q}_k + \sum_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{b}; \tag{20}$$

(4) Let $k = k + 1$; go back to step (2);

We will show that if the preference relation $<\cdot$ is weakly linear, the sequence of vectors, $\mathbf{q}_0, \mathbf{q}_1, \cdots$ generated in the above procedure will indeed terminate at a solution vector. In order to prove the convergence of the above gradient descent algorithm, it is natural to attempt to show that each correction brings the vector $\mathbf{q}_k$ closer to the solution region. That is, one might try to show that if $\mathbf{q}$ is any solution vector, then $\|\mathbf{q}_{k+1} - \mathbf{q}\|$ is smaller than $\|\mathbf{q}_k - \mathbf{q}\|$. ($\|.\|$ denotes the magnitude of a vector.) Although this may not be true in general, one can show that it is true for solution vectors that are sufficiently long.

Let $\mathbf{q}$ be any solution vector. This means that $\mathbf{q}^T\mathbf{b} > 0$ for every $\mathbf{b} \in \mathbf{B}$. Thus, from equation (20) one obtains:

$$\mathbf{q}_{k+1} - \gamma\mathbf{q} = (\mathbf{q}_k - \gamma\mathbf{q}) + \mathbf{v}_k, \tag{21}$$

where γ is a positive scale factor and

$$\mathbf{v}_k = \sum_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{b}. \tag{22}$$

Therefore:

$$\|\mathbf{q}_{k+1} - \gamma\mathbf{q}\|^2 = \|\mathbf{q}_k - \gamma\mathbf{q}\|^2 + 2(\mathbf{q}_k - \gamma\mathbf{q})^T\mathbf{v}_k + \|\mathbf{v}_k\|^2. \tag{23}$$

From equation (19), we know that $\mathbf{q}_k^T\mathbf{b} \le 0$ for every $\mathbf{b} \in \Gamma(\mathbf{q}_k)$ and hence:

$$\mathbf{q}_k^T\mathbf{v}_k = \left(\sum_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{q}_k^T\mathbf{b}\right) \le 0. \tag{24}$$

By combining equations (22)–(24), we arrive at the following inequality:

$$\|\mathbf{q}_{k+1} - \gamma\mathbf{q}\|^2 \le \|\mathbf{q}_k - \gamma\mathbf{q}\|^2 - 2\gamma\mathbf{q}^T\mathbf{v}_k + \|\mathbf{v}_k\|^2. \tag{25}$$

Due to the fact that $\mathbf{q}^T\mathbf{v}_k$ is strictly positive, the second term in equation (25) will dominate the third if γ is sufficiently large. Now let

$$\delta^2 = \max_{\mathbf{b}\in\mathbf{B}} \|\mathbf{b}\|^2, \qquad \sigma = \min_{\mathbf{b}\in\mathbf{B}} \mathbf{q}^T\mathbf{b} > 0. \tag{26}$$

Note that δ^2 and σ are independent of k since the max and min operations are performed over the set of *all* difference vectors $\mathbf{B}$. Since $\Gamma(\mathbf{q}_k) \subseteq \mathbf{B}$, we have:

$$\max_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \|\mathbf{b}\|^2 \le \max_{\mathbf{b}\in\mathbf{B}} \|\mathbf{b}\|^2 = \delta^2, \cdot \tag{27}$$

and

$$\min_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{q}^T\mathbf{b} \ge \min_{\mathbf{b}\in\mathbf{B}} \mathbf{q}^T\mathbf{b} = \sigma. \tag{28}$$

It follows:

$$\begin{aligned}\|\mathbf{v}_k\|^2 &= \left\|\sum_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{b}\right\|^2 \le \sum_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \|\mathbf{b}\|^2 \le \rho_k \max_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \|\mathbf{b}\|^2 \\ &\le \rho_k \max_{\mathbf{b}\in\mathbf{B}} \|\mathbf{b}\|^2 = \rho_k\delta^2, \\ \mathbf{q}^T\mathbf{v}_k &= \sum_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{q}^T\mathbf{b} \ge \rho_k \min_{\mathbf{b}\in\Gamma(\mathbf{q}_k)} \mathbf{q}^T\mathbf{b} \ge \rho_k \min_{\mathbf{b}\in\mathbf{B}} \mathbf{q}^T\mathbf{b} = \rho_k\sigma,\end{aligned} \tag{29}$$

where $\rho_k \ge 1$ is the cardinality of the set $\Gamma(\mathbf{q}_k)$. With the choice $\gamma = \delta^2/\sigma$, we immediately obtain from equations (25) and (29):

$$\|\mathbf{q}_{k+1} - \gamma\mathbf{q}\|^2 \le \|\mathbf{q}_k - \gamma\mathbf{q}\|^2 - \rho_k\delta^2. \tag{30}$$

Clearly, the squared distance from $\mathbf{q}_k$ to $\gamma\mathbf{q}$ is reduced by at least $\rho_k\delta^2$ at each iteration. We can therefore conclude that in the gradient procedure $\mathbf{q}_k$ will eventually be inside the solution region. That is, $\mathbf{q}_k$ converges to a solution vector after a finite number of iterations.

We have shown in the above discussions that if a user preference relation is weakly linear, by adding more sample documents to the training set the query vector inferred by our method will definitely approach the solution vector applicable to the *whole* collection. It should be emphasized here that one has no a priori knowledge to determine whether a user preference relation is weakly linear or not on the whole collection. As a matter of fact, in any inductive process it is not possible to predetermine if a certain assumption is valid in the universe of discourse. Therefore, in practice one may assume that the user preference relation is weakly linear (linearity assumption) unless one observes nonweakly linear cases in the training set. When it is discovered that the user preference relation is not weakly linear, there are two plausible alternatives: (1) one may have to select an appropriate nonlinear decision function for classifying the documents, and (2) one may regard the linearity assumption as a first-order approximation. In the latter case, one can still minimize the perceptron criterion function to obtain an approximate query vector such that it can correctly specify the preference relationships between as many document pairs as possible. (A detailed discussion of this minimization procedure can be found in Duda and Hart (1973).)

Critical Remarks on Other Inductive Methods

There are two basic inductive methods in information retrieval, the relevance feedback and the probabilistic approaches (Salton & McGill, 1983; Van Rijsbergen, 1979; Rocchio, 1971). In these methods, a two-value relevance scale is usually used for classifying the documents. That is, a document is considered as being either relevant or nonrelevant. We have already mentioned that such a dichotomy of classification schemes may be too restrictive. Nevertheless, a user preference relation $<\cdot$ on $\mathbf{D}$ can be defined as follows:

$$<\cdot\ = \{(\mathbf{d}, \mathbf{d}') | \mathbf{d} \in \mathbf{nrel}, \mathbf{d}' \in \mathbf{rel}\}, \tag{31}$$

where **rel** and **nrel** denote the relevant and nonrelevant sets of document vectors, respectively. Assume that the preference relation $<\cdot$ is weakly linear. By adopting the proposed acceptable retrieval strategy, the problem of ranking is reduced to finding a linear decision function from a training set $\mathbf{S} \subseteq \mathbf{D}$ of document vectors.

Now within the framework of our inductive linear model, we present a critical analysis of the existing relevance feedback and probabilistic methods based on a preference relation given by equation (31).

Relevance Feedback in Vector Space Model

Rocchio's Method. In the vector space model, the similarity (correlation) between a document $\mathbf{d}$ and the query $\mathbf{q}$ is measured by the cosine function $\beta(\mathbf{q}, \mathbf{d})$,

$$\beta(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{\|\mathbf{q}\| \, \|\mathbf{d}\|} = \frac{1}{\|\mathbf{q}\| \, \|\mathbf{d}\|} \mathbf{q}^T \mathbf{d} . \tag{32}$$

Rocchio defined an *ideal* query as one which ranks all the relevant documents in **rel** higher than those in the nonrelevant set **nrel**, namely,

$$\beta(\mathbf{q}, \mathbf{d}) < \beta(\mathbf{q}, \mathbf{d}') , \tag{33}$$

for all $\mathbf{d} \in \mathbf{nrel}$ and $\mathbf{d}' \in \mathbf{rel}$. From our formulation, it is clear that the ideal query in Rocchio's method is a solution vector and the cosine similarity measure is a special kind of linear decision function.

Since an ideal query may not exist for an arbitrary user preference relation, Rocchio suggested a method for computing an approximate solution vector by maximizing the following criterion function with respect to $\mathbf{q}$:

$$C = \frac{1}{n_0} \sum_{\mathbf{d}' \in \mathbf{rel}} \beta(\mathbf{q}, \mathbf{d}') - \frac{1}{n_1} \sum_{\mathbf{d} \in \mathbf{nrel}} \beta(\mathbf{q}, \mathbf{d}) , \tag{34}$$

where n_0 is the number of relevant documents in **rel** and n_1 the number of nonrelevant documents in **nrel**. By the definition of $\beta(\mathbf{q}, \mathbf{d})$, C can be expressed as:

$$C = \frac{\mathbf{q}}{\|\mathbf{q}\|} \cdot \left[\frac{1}{n_0} \sum_{\mathbf{d}' \in \mathbf{rel}} \frac{\mathbf{d}'}{\|\mathbf{d}'\|} - \frac{1}{n_1} \sum_{\mathbf{d} \in \mathbf{nrel}} \frac{\mathbf{d}}{\|\mathbf{d}\|} \right] . \tag{35}$$

It can be easily verified that C is maximized if $\mathbf{q}$ is chosen to be:

$$\hat{\mathbf{q}} = k \left[\frac{1}{n_0} \sum_{\mathbf{d}' \in \mathbf{rel}} \frac{\mathbf{d}'}{\|\mathbf{d}'\|} - \frac{1}{n_1} \sum_{\mathbf{d} \in \mathbf{nrel}} \frac{\mathbf{d}}{\|\mathbf{d}\|} \right] , \tag{36}$$

where k is a positive constant. This query $\hat{\mathbf{q}}$ is referred to as the *optimal* query by Rocchio, which has been widely adopted for the modification of the query vector in relevance feedback models (Salton, 1971; Rocchio, 1971).

There are two basic problems in Rocchio's method. Firstly, the optimal query is not necessarily a solution vector even if the preference relation $<\cdot$ is weakly linear. This fact can be seen from the following arguments. By choosing a null vector $\mathbf{q}_0 = \mathbf{0}$ as the initial query in our gradient descent procedure (described in the fourth section), at the first iteration (i.e., $k = 0$) one obtains from equations (19) and (31) the set $\Gamma(\mathbf{q}_0 = \mathbf{0})$:

$$\Gamma(\mathbf{0}) = \left\{ \mathbf{b} = \left(\frac{\mathbf{d}'}{\|\mathbf{d}'\|} - \frac{\mathbf{d}}{\|\mathbf{d}\|} \right) \middle| \, \mathbf{d} \in \mathbf{nrel}, \mathbf{d}' \in \mathbf{rel} \right\} , \tag{37}$$

which is obviously equal to the set **B** defined by equation (13). (Note that in order to be consistent with Rocchio's notation, normalized document vectors have been used to define the difference vector **b** in the above equation.) With $\mathbf{q}_0 = \mathbf{0}$, according to equations (20) and (37), $\mathbf{q}_1$ can be written as:

$$\begin{aligned} \mathbf{q}_1 &= \mathbf{q}_0 + \sum_{\mathbf{b} \in \mathbf{B}} \mathbf{b} = \sum_{\mathbf{d}' \in \mathbf{rel}} \sum_{\mathbf{d} \in \mathbf{nrel}} \left(\frac{\mathbf{d}'}{\|\mathbf{d}'\|} - \frac{\mathbf{d}}{\|\mathbf{d}\|} \right) \\ &= n_0 n_1 \left[\frac{1}{n_0} \sum_{\mathbf{d}' \in \mathbf{rel}} \frac{\mathbf{d}'}{\|\mathbf{d}'\|} - \frac{1}{n_1} \sum_{\mathbf{d} \in \mathbf{nrel}} \frac{\mathbf{d}}{\|\mathbf{d}\|} \right] , \end{aligned} \tag{38}$$

which is identical to the optimal query vector $\hat{\mathbf{q}}$ defined by equation (36). Therefore, Rocchio's method is only a first-order approximation in our linear model.

The second problem associated with Rocchio's approach is that the criterion function C defined by equation (35) does not have a maximum value unless the query vector $\mathbf{q}$ is *normalized.* On the other hand, it can be easily verified that whether the query vector $\mathbf{q}$ is normalized or not in equation (32) will not affect the ranking of documents. For this reason, it is not entirely clear how one can justify the maximization of the criterion function suggested by Rocchio from a theoretical standpoint although it is intuitively appealing.

Van Rijsbergen's Method. Van Rijsbergen (1979) suggested that the fix-increment error correction procedure (Duda & Hart, 1973; Nilsson, 1965; Minsky & Papert, 1969) can be used to construct a linear decision function to distinguish the relevant documents from the nonrelevant ones. In this method, the query vector at the $(k + 1)$th iteration is defined by:

$$\begin{aligned} \mathbf{q}_{k+1} &= \mathbf{q}_k + c\mathbf{d} \quad \text{if } \mathbf{q}_k^T \mathbf{d} - H \leq 0 \quad \text{and} \quad \mathbf{d} \in \mathbf{rel} \\ \mathbf{q}_{k+1} &= \mathbf{q}_k - c\mathbf{d} \quad \text{if } \mathbf{q}_k^T \mathbf{d} - H > 0 \quad \text{and} \quad \mathbf{d} \in \mathbf{nrel} , \end{aligned} \tag{39}$$

where c is a positive constant determining the step size and H a threshold. It can be easily shown (Nilsson, 1965) that this iterative method is in fact a gradient descent procedure designed to find a query vector $\mathbf{q}$ such that the following decision rules hold:

$$\begin{aligned} &\text{Decide } \mathbf{d} \text{ is relevant} && \text{if } \mathbf{q}^T \mathbf{d} > H , \\ &\text{Decide } \mathbf{d} \text{ is nonrelevant} && \text{if } \mathbf{q}^T \mathbf{d} \leq H . \end{aligned} \tag{40}$$

The above rules enable the system to decide if a document is relevant or not.

The main reason for incorporating a threshold parameter H in the gradient descent procedure is to formulate a set of decision rules in order to separate the documents into two classes. However, in information retrieval our objective is to rank documents consistent with the user preference relation. The problem is not just to separate documents into disjoint classes by employing strict decision rules, without taking into account the relationships between classes.

Therefore, from our analysis it seems unnecessary to introduce a pre-determined threshold parameter because we are primarily interested in finding a decision function which reflects the user preference relationships between documents. Besides, it may be difficult in practice to decide which threshold value to use. It may also be difficult to apply this procedure to a classification problem with multiple classes.

Binary Probabilistic Independence Model

In the binary probabilistic model, each document is described by a binary vector $\mathbf{d} = (x_1, x_2, \ldots, x_p)^T$ with $x_i = 0$ or 1. From the statistical independence assumptions, one obtains the following well known linear decision function:

$$g(\mathbf{d}) = \sum_{i=1}^{p} x_i \log \frac{p_i(1 - q_i)}{(1 - p_i)q_i} + \text{constant}, \tag{41}$$

where $p_i = P(x_i = 1|\mathbf{rel})$ and $q_i = P(x_i = 1|\mathbf{nrel})$. These probabilities for each component x_i can be conveniently estimated from a 2 × 2 contingency table:

	Relevant	Nonrelevant	
$x_i = 1$	r	$n - r$	n
$x_i = 0$	$R - r$	$N - n - R + r$	$N - n$
	R	$N - R$	N

The symbols in the above table are self-explanatory. It immediately follows:

$$p_i = \frac{r}{R}, \qquad q_i = \frac{n - r}{N - R}.$$

Substituting these values into equation (41), $g(\mathbf{d})$ can be rewritten as:

$$g(\mathbf{d}) = \sum_{i=1}^{p} x_i[\log r(N - n - R + r) - \log (n - r)(R - r)]_i + \text{constant}. \tag{42}$$

On the other hand, in our approach the solution vector can be approximated by equation (38) without the normalization factors, namely:

$$\mathbf{q} \cong \mathbf{q}_1 = \sum_{\mathbf{d}' \in \mathbf{rel}} \sum_{\mathbf{d} \in \mathbf{nrel}} (\mathbf{d}' - \mathbf{d}). \tag{43}$$

Based on the contingence table and equation (43), the ith component of $\mathbf{q}_1 = (w_{11}, w_{12}, \ldots, w_{1p})^T$ is given by:

$$w_{1i} = [r(N - n - R + r\,) - (n - r)(R - r)]_i\,.$$

Thus, we obtain another linear decision function:

$$\begin{aligned} f(\mathbf{d}) &\cong \mathbf{q}_1^T\mathbf{d} = \sum_{i=1}^{p} x_i w_{1i} \\ &= \sum_{i=1}^{p} x_i[r(N - n - R + r) - (n - r)(R - r)]_i \\ &= NR \sum_{i=1}^{p} x_i\left(\frac{r}{R} - \frac{n}{N}\right)_i. \end{aligned} \tag{44}$$

Some useful results are observed from the above linear function. If $r/R = n/N$ (i.e., $w_{1i} = 0$), index term t_i gives no useful information for document classification. If $r/R > n/N$, the presence of term t_i (i.e., $x_i = 1$) in a document vector will contribute w_{1i} *votes* for the relevant class (**rel**). On the other hand, if $r/R < n/N$, the presence of term t_i will contribute $|w_{1i}|$ votes for the nonrelevant class (**nrel**). Thus, the absolute value $|r/R - n/N|$ provides an approximate measure of the usefulness of index term t_i for distinguishing relevant and nonrelevant documents.

By comparing the results given by equations (42) and (44), it is evident that there exists a close relationship between the probabilistic approach and our linear model. However, our method has the advantage that it is applicable to nonbinary document representation and to a retrieval problem with multiple classes. The need for making the independence assumption can also be avoided. It should be noted that the independence assumption is a sufficient condition but not a necessary condition for the existence of a linear decision function in a variety of classification problems.

Conclusion

We have presented some evidence to support that the concepts of user preference are indeed useful in the design of a retrieval system. In particular, when the preference relation is weakly linear, we have shown that the notion of document ranking is directly linked to that of an acceptable ranking strategy. Such a strategy leads to a convenient algorithm for inferring the query vector by an inductive process. Our analysis indicates that the relevance feedback method can be considered as a special case of our adaptive linear model and the independence assumption in the probabilistic model may be too strong an approximation for a linear system.

Although we have provided a framework for designing an adaptive retrieval system, the problem of how to select a small but effective training set of documents remains an outstanding issue. Another problem is related to the large number of index terms used in the document representation. Obviously, this has a significant negative impact on system performance particularly in an interactive environment. Without a satisfactory resolution of these problems, one may not be able to build a viable retrieval system with learning capability.

References

Bollmann, P., & Wong, S. K. M. (1987). Adaptive linear information retrieval models. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.*

Bookstein, A. (1983a). Information retrieval: A sequential learning process. *Journal of the American Society for Information Science, 34,* 331–342.

Bookstein, A. (1983b). Outline of a general probabilistic retrieval model. *Journal of Documentation, 39,* 63–72.

Cooper, W. S. (1968). Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *Journal of the American Society for Information Science, 19,* 30–41.

Croft, W. B., & Harper, D. J. (1977). Using probabilistic models of document retrieval without relevance information. *Journal of Documentation, 35,* 106–119.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* New York: John Wiley.

Fishburn, P. C. (1970). *Utility theory for decision making.* New York: Wiley.

Minsky, M., & Papert, S. (1969). *Perceptrons—an introduction to computational geometry.* Cambridge, MA: MIT Press.

Nilsson, N. J. (1965). *Learning machine—foundations of trainable pattern classifying systems.* New York: McGraw-Hill.

Raghavan, V. V., & Wong, S. K. M. (1986). A critical analysis of vector space model in information retrieval. *Journal of the American Society for Information Science, 37,* 279–287.

Roberts, F. S. (1976). *Measurement theory.* New York: Academic Press.

Robertson, S. E., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science, 27,* 129–146.

Rocchio, Jr. J. J. (1971). Relevance feedback in information retrieval. In Salton, G. (Ed.), *The SMART retrieval system—experiments in automatic document processing.* Englewood Cliffs, NJ: Prentice-Hall.

Salton, G. (Ed.) (1971). *The SMART retrieval system—experiments in automatic document processing.* Englewood Cliffs, NJ: Prentice-Hall.

Salton, G., & McGill, M. H. (1983). *Introduction to modern information retrieval.* New York: McGraw-Hill.

Van Rijsbergen, C. J. (1979). *Information retrieval.* London: Butterworth.

Wong, S. K. M., & Yao, Y. Y. (1987). A statistical similarity measure. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.*

Wong, S. K. M., & Ziarko, W. (1986). A machine learning approach to information retrieval. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval,* 228–233.

Wong, S. K. M., Ziarko, W., & Wong, P. C. N. (1985). Generalized vector space model in information retrieval. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval,* 18–25.

Yu, C. T., & Salton, G. (1976). Precision weighting—an effective automatic indexing method. *Journal of the Association for Computing Machinery, 23,* 76–88.