Formalization and Evaluation of Linear Relevance Feedback

S. K. M. Wong
Y. Y. Yao
G. Salton*
C. Buckley*

TR 89-992
April 1989

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This study was supported by the National Science Foundation under grant IR 87-02735.

Formalization and Evaluation of Linear Relevance Feedback

S.K.M. Wong,[+] Y.Y. Yao,[+] G. Salton,[*] and C. Buckley [*]

Abstract

This study outlines an adaptive method which constructs improved query vectors based on the user preference judgments on sample document pairs. In particular, the user states that some documents are preferred to other documents and the system is then expected to rank the preferred documents ahead of the others. In the adaptive system, all needed parameter values are provided within the model, and a solution query vector is constructed under well defined conditions.

Certain relationships between the new adaptive and the conventional relevance feedback systems are discussed and evaluation data are provided to demonstrate the effectiveness of the system.

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

[*] G. Salton, C. Buckley, Department of Computer Science, Cornell University, Ithaca, New York, 14853.

1. Introduction

The formulation of useful user queries presents a main obstacle in information retrieval. In most information retrieval systems, the processes of indexing and retrieval are not transparent to the users. It is therefore difficult for the user to construct a query precisely according to the individual information needs. Relevance feedback is a well known technique to alleviate the difficult problem of query formulation and to enhance the effectiveness of information retrieval. In relevance feedback an initial query is usually used to retrieve a sample of documents. Based on user preferences on the sample, an improved query may be constructed. This process can be applied iteratively by treating the improved query as an initial query. As the size of the sample documents increases, it is hoped that a more accurate query may eventually be obtained.

Extensive studies of relevance feedback have been made within the frameworks of Boolean, vector space, and probabilistic models [2-4, 6, 8-14, 16-18, 22, 23, 25]. The evaluation results clearly demonstrate that relevance feedback provides an effective method for constructing user queries. With the recent developments in parallel processing, it now seems possible to implement relevance feedback in a practical retrieval system [13, 18]. However, serious drawbacks exist in the conventional approaches. For example, when the probabilistic model is used for relevance feedback, a document can only be represented as a *binary* vector, although attempts have been made to remove such a restriction [24]. Besides, it is often difficult to justify the probabilistic independence assumption. On the other hand, terms weights can easily be incorporated into the vector space model, but systematic methods do not exist for determining certain parameters in a vector based relevance feedback model [10-14].

Recently, an adaptive query formulation method was proposed based on the use of preference judgments for certain pairs of documents [20, 21]. There is no constraint on the document representation (i.e. both binary and weighted document representation are allowed). A salient feature of this new feedback method is that given the user preference

judgment for a set of sample documents, the query can be automatically generated by an iterative algorithm without the need to introduce specific formulas or parameters. Whether such an iterative scheme converges depends on the user preference structure. A more detailed discussion of the conditions under which a *linear* preference structure holds can be found in [1].

One of the objectives of this paper is to provide a theoretical basis for the development of a more effective adaptive information retrieval system. We also present an experimental evaluation of the proposed adaptive query formulation method [20]. These results are compared with evaluations of the conventional relevance feedback approaches [6, 14].

2. Basic Concepts

Given a set of documents D, a user preference can be defined by a binary relation $<\bullet$ on D as follows:

$$d <\bullet d' \quad \text{if the user prefers } d' \text{ to } d \ .$$

We may assume that each document $d \in D$ is represented by an n-dimensional vector $\mathbf{d} = (d_1, d_2, \ldots, d_n)$. From the ranking point of view, our primary objective is to find an *order-preserving* function f on the set of of document vectors $\mathbf{D}$ such that whenever d' is preferred to d,

$$\mathbf{d} <\bullet \mathbf{d}' \Rightarrow f(\mathbf{d}) < f(\mathbf{d}') \ . \tag{2.1}$$

A function satisfying condition (2.1) guarantees that the preferred documents will be ranked ahead of the less preferred. In other words, condition (2.1) implies:

$$f(\mathbf{d}) \geq f(\mathbf{d}') \Rightarrow \neg(\mathbf{d} <\bullet \mathbf{d}') \ . \tag{2.2}$$

We call such a function f an *acceptable ranking* function.

In the adaptive linear model [1, 20], f is chosen to be:

$$f(\mathbf{d}) = \sum_{j=1}^{n} w_j d_j = \mathbf{q} \cdot \mathbf{d} \ , \tag{2.3}$$

where $\mathbf{q} = (w_1, w_2, \ldots, w_n)$ is an n-dimensional vector. The preference relation $<\bullet$ is *weakly linear* if there exists a linear function f defined by eqn. (2.3) such that for any $\mathbf{d}, \mathbf{d}' \in \mathbf{D}$,

$$\mathbf{d} <\bullet \mathbf{d}' \Rightarrow f(\mathbf{d}) < f(\mathbf{d}') = \mathbf{q} \cdot \mathbf{d} < \mathbf{q} \cdot \mathbf{d}' \ , \tag{2.4}$$

where $\mathbf{q}$ is referred to as a query vector.

For every document pair $\mathbf{d}, \mathbf{d}' \in \mathbf{D}$ satisfying the relationship $\mathbf{d} <\bullet \mathbf{d}'$, we can define a difference vector $\mathbf{b} = \mathbf{d}' - \mathbf{d}$. Let

$$\mathbf{B} = \{ \mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d}, \mathbf{d}' \in \mathbf{D} \text{ and } \mathbf{d} <\bullet \mathbf{d}' \} \ . \tag{2.5}$$

By definition, if the preference relation is weakly linear, $\mathbf{d} <\bullet \mathbf{d}'$ implies $\mathbf{q} \cdot \mathbf{d} < \mathbf{q} \cdot \mathbf{d}'$, i.e., $\mathbf{q} \cdot \mathbf{b} > 0$. It is therefore clear that the problem of finding a query vector $\mathbf{q}$ satisfying eqn. (2.4) for every document pair with $\mathbf{d} <\bullet \mathbf{d}'$ is equivalent to solving a system of linear inequalities as follows:

$$\mathbf{q} \cdot \mathbf{b} > 0 \ , \qquad \text{for every } \mathbf{b} \in \mathbf{B} \ . \tag{2.6}$$

Each difference vector $\mathbf{b} \in \mathbf{B}$ defines a hyperplane, $\mathbf{b} \cdot \mathbf{q} = 0$, with $\mathbf{b}$ as its normal vector. The hyperplane $\mathbf{b} \cdot \mathbf{q} = 0$ passes through the origin $\mathbf{0}$ and divides the space into two parts. The half-plane on the side of the normal vector is referred to as the positive side of the plane. It can be easily seen that for any vector $\mathbf{q}$ on the positive side of the hyperplane $\mathbf{q} \cdot \mathbf{b} = 0$ the condition will be satisfied $\mathbf{q} \cdot \mathbf{b} > 0$ as shown in Figure 1 (a). For two difference vectors $\mathbf{b}, \mathbf{b}' \in \mathbf{B}$, any vector $\mathbf{q}$ that lies in the overlap region of the positive sides of the corresponding two hyperplanes satisfies the conditions $\mathbf{q} \cdot \mathbf{b} > 0$ and $\mathbf{q} \cdot \mathbf{b}' > 0$ (see Figure 1 (b)). In general, any query vector that lies in the overlap region

of the positive sides of all hyperplanes defined by the vectors in **B** is a solution to the system of linear inequalities (2.6). This region is called the *solution region* and any vector in the solution region is called a *solution vector*. In fact, the solution region $\mathbf{C_D}$ is a *convex cone* [7]. That is, (a) $\mathbf{q} \in \mathbf{C_D} \Rightarrow k\mathbf{q} \in \mathbf{C_D}$ for all $k > 0$, and (b) $\mathbf{q}, \mathbf{q}' \in \mathbf{C_D} \Rightarrow (\mathbf{q} + \mathbf{q}') \in \mathbf{C_D}$. (Note that because $k > 0$, $\mathbf{C_D}$ is not a vector subspace.)

In what follows, an iterative algorithm is outlined for learning a solution vector from a given set sample of documents $\mathbf{S} \subseteq \mathbf{D}$ [20]. Note that the symbol **B** in the remainder of this paper will represent the set of difference vectors defined as follows:

$$\mathbf{B} = \{\, \mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d}, \mathbf{d}' \in \mathbf{S} \text{ and } \mathbf{d} <_{\bullet} \mathbf{d}' \,\} \ . \tag{2.7}$$

Algorithm 1

(i) Choose a starting query vector $\mathbf{q}_0$ and let $k = 0$;

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

$$\Gamma(\mathbf{q}_k) = \{\, \mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d}, \mathbf{d}' \in \mathbf{S},\ \mathbf{d} <_{\bullet} \mathbf{d}' \text{ and } \mathbf{q}_k \cdot \mathbf{b} \le 0 \,\} \subseteq \mathbf{B} \ ; \tag{2.8}$$

If $\Gamma(\mathbf{q}_k) = \varnothing$, then $\mathbf{q}_k$ is a solution vector and the procedure is terminated;

(iii) Let

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \sum_{\mathbf{b} \in \Gamma(\mathbf{q}_k)} \mathbf{b} \ ; \tag{2.9}$$

(iv) Let $k = k + 1$; go back to step (ii);

The above procedure may produce some negative components in the solution vector. For practical reasons, negative weights are not allowed and typically replaced by 0 in some relevance feedback methods [10-14]. Replacing a negative weight by 0 may inadvertently change a solution vector into a non-solution vector. Hence negative

components are allowed; indeed, negative weights are also used in the probabilistic models [8, 14, 19].

The following example demonstrates the operation of the algorithm.

Example 3.1

Consider a set of document vectors $\mathbf{D} = \{ \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4 \}$ with

$$\mathbf{d}_1 = (0.0, 1.0), \qquad \mathbf{d}_2 = (0.7, 1.0),$$
$$\mathbf{d}_3 = (1.0, 0.2), \qquad \mathbf{d}_4 = (1.0, 0.0).$$

and assume that the following preference relation is defined on **D**:

$$\mathbf{d}_1 <\bullet \mathbf{d}_2, \quad \mathbf{d}_1 <\bullet \mathbf{d}_3, \quad \mathbf{d}_4 <\bullet \mathbf{d}_2, \quad \mathbf{d}_4 <\bullet \mathbf{d}_3 .$$

Based on this preference relation, from eqn. (2.7) the following set of difference vectors is obtained:

$$\mathbf{B} = \{\mathbf{b}_{21}, \mathbf{b}_{24}, \mathbf{b}_{31}, \mathbf{b}_{34}\},$$

where

$$\mathbf{b}_{21} = \mathbf{d}_2 - \mathbf{d}_1 = (0.7, 1.0) - (0.0, 1.0) = (0.7, 0.0),$$
$$\mathbf{b}_{24} = \mathbf{d}_2 - \mathbf{d}_4 = (0.7, 1.0) - (1.0, 0.0) = (-0.3, 1.0),$$
$$\mathbf{b}_{31} = \mathbf{d}_3 - \mathbf{d}_1 = (1.0, 0.2) - (0.0, 1.0) = (1.0, -0.8),$$
$$\mathbf{b}_{34} = \mathbf{d}_3 - \mathbf{d}_4 = (1.0, 0.2) - (1.0, 0.0) = (0.0, 0.2).$$

Table 1 contains the detailed steps for constructing a query vector based on three different starting query vectors, from which three different solution vectors are derived.

It is perhaps worth mentioning here that if one starts with the null vector (i.e. case 3), the query vector obtained in the first iteration corresponds to Rocchio's *optimal* query [10]. This example demonstrates that Rocchio's optimal query may not necessarily be a solution vector satisfying all preference judgments.

An illustration of Case 1 is given in Figure 2. As shown in the figure the initial vector $\mathbf{q}_0$ is located at (0.3, 1.0). One can easily check that the conditions $\mathbf{q}_0 \cdot \mathbf{d}_1 < \mathbf{q}_0 \cdot \mathbf{d}_2$, $\mathbf{q}_0 \cdot \mathbf{d}_4 < \mathbf{q}_0 \cdot \mathbf{d}_2$, and $\mathbf{q}_0 \cdot \mathbf{d}_4 < \mathbf{q}_0 \cdot \mathbf{d}_3$ are satisfied by $\mathbf{q}_0$. The one condition clearly violated by $\mathbf{q}_0$ is $\mathbf{q}_0 \cdot \mathbf{d}_1 < \mathbf{q}_0 \cdot \mathbf{d}_3$. Geometrically, this means that $\mathbf{q}_0$ is much closer to $\mathbf{d}_1$ than to $\mathbf{d}_3$. A transformation that moves $\mathbf{q}_0$ closer to $\mathbf{d}_3$ can be carried out as $\mathbf{q}_1 = \mathbf{q}_0 + (\mathbf{d}_3 - \mathbf{d}_1) = \mathbf{q}_0 + \mathbf{b}_{31} = (1.3, 0.2)$. At this point, $\mathbf{d}_3 <\bullet\, \mathbf{d}_1$ is satisfied, but $\mathbf{d}_4 <\bullet\, \mathbf{d}_2$ is violated. The further transformation $\mathbf{q}_2 = \mathbf{q}_1 + (\mathbf{d}_2 - \mathbf{d}_4) = \mathbf{q}_0 + \mathbf{b}_{24} = (1.0, 1.2)$ now produces a query vector in the solution region. That is, $\mathbf{q}_2$ satisfies all the conditions as specified by the user preference:

$$\mathbf{q}_2 \cdot \mathbf{d}_1 < \mathbf{q}_2 \cdot \mathbf{d}_2 \ , \quad \mathbf{q}_2 \cdot \mathbf{d}_1 < \mathbf{q}_2 \cdot \mathbf{d}_3,$$
$$\mathbf{q}_2 \cdot \mathbf{d}_4 < \mathbf{q}_2 \cdot \mathbf{d}_2 \ , \quad \mathbf{q}_2 \cdot \mathbf{d}_4 < \mathbf{q}_2 \cdot \mathbf{d}_3.$$

It should be noted that in general the above iterative algorithm may not necessarily converge. We have shown [20] that if the user prefence relation is *weakly linear*, then independent of the choice of the starting vector, Algorithm 1 will find a solution vector in a finite number of iterations.

3. Experimental Evaluation

A number of experiments were performed to evaluate the effectiveness of the proposed adaptive learning procedure. Before presenting the experimental results, however, some important issues relating to the experiments must be clarified.

Representation of preference relations

In most retrieval systems, it is implicitly assumed that a user preference is represented by a *two-level* structure (a two-level relevance scale). That is, the documents in a given collection D are divided only into two disjoint subsets, a relevant set D_2 and a non-relevant set D_1. In other words, the user prefers the documents in D_2 to those

in D_1, which defines the following preference relation $<\bullet$ on D:

$$d <\bullet d' \iff (d \in D_1, d' \in D_2) \ . \tag{3.1}$$

The two-level structure (i.e., $D_1 <\bullet D_2$) represents the simplest type of user preference. In this special case, there is little difference between the concepts of document ranking and classification. It can be shown that the problem of ranking a set of documents based on a two-level relevance scale is equivalent to that of classifying the documents into two distinct classes. In general, a user preference has a multi-level structure in which documents in D are partitioned into more than two subsets $D_1, D_2, \ldots, D_m$ with $D_1 <\bullet D_2 <\bullet \cdots <\bullet D_m$ (or more formally, the preference relation is a weak order [20]). The solution to the ranking and classification problems with a multi-level structure are fundamentally different.

In contrast to conventional feedback approaches, the proposed adaptive method is applicable to multi-level preference structures. However, in the available test collections the user preference relations are all described in terms of a two-level structure. To compensate for the use of such a simple structure, we construct an additional level of preference by assuming $\mathbf{0} <\bullet \mathbf{d}$ for all $\mathbf{d} \in \mathbf{S}$. In other words, every document in the sample S is assumed to be more preferred to a blank sheet of paper characterized by the null vector $\mathbf{0}$. This is a reasonable assumption if all sample documents are believed to contain some *useful* information for the user. For example, one may use this assumption when the samples are selected from the top ranked documents generated by the weighted queries in the standard vector space model [14, 15]. By doing so, additional difference vectors are generated, namely:

$$\{\mathbf{b} = \mathbf{d} - \mathbf{0} \mid \mathbf{d} \in \mathbf{S} \text{ and } \mathbf{0} <\bullet \mathbf{d}\} \ .$$

This set is simply equal to the set of sample documents S. Therefore, the complete set of difference vectors generated by S now becomes

$$\mathbf{B}^* = \mathbf{B} \cup \mathbf{S} \ , \tag{3.2}$$

where **B** is the set of original difference vectors derived from a two-level structure as defined by eqn. (3.1). In our experiments **B** was actually replaced by $\mathbf{B}^*$ in step (ii) of Algorithm 1.

Selection of sample documents

An initial search in [14] was performed for selecting sample documents. The term weights used for both documents and queries in the initial search are:

$$\frac{(0.5 + 0.5\frac{tf_j}{\mathbf{max}\ tf}) \log\frac{N}{n_j}}{\sqrt{(0.5 + 0.5\frac{tf_j}{\mathbf{max}\ tf})^2 (\log\frac{N}{n_j})^2}} \ , \tag{3.3}$$

where

tf_j is the occurrence frequency of term t_j in the document (query),

$\mathbf{max}\ tf = \mathbf{max}\ \{tf_j\}$ is the maximum occurrence frequency of any term in the document,

N is the number of documents in the collection, and

n_j is the number of documents indexed by term t_j .

The formula (3.3) is in the form of the ($tf \times idf$) (term frequency times inverse document frequency) weighting system [14, 15]. The cosine similarity measure between the original user query and documents was used to rank the documents. In each case the top 15 documents were selected as sample documents [14]. The user judgments, in terms of relevance and non-relevance, for these 15 documents were used to construct the preference relation. The main reason for choosing this method for selecting samples is the need to compare the results with those available from earlier feedback experiments [14]. There may be better ways to select a good representative sample, but we will not con-

sider this problem in the present study. In fact, the problem of how to select a small but effective training sample of documents is an outstanding issue.

Choice of starting query vectors

We used three different starting query vectors. Since we wish to take advantage of the information supplied by the user, one obvious choice is the user original query with term weights defined by eqn. (3.3). The second choice is a query generated as the sum of all relevant document vectors in the sample. The third choice is a combination of the first two starting query vectors.

Let $\mathbf{d}^+$ denote the sum of all relevant document vectors in the sample. If one assumes that the similarity between two relevant documents is greater than that between a relevance document and a non-relevant document, it follows that:

$$\mathbf{d}^+ \cdot (\mathbf{d}' - \mathbf{d}) = \mathbf{d}^+ \cdot \mathbf{b} > 0 , \quad \mathbf{b} \in \mathbf{B} . \tag{3.4}$$

Of course, in practice the above assumption is not always true. Nevertheless, $\mathbf{d}^+$ may represent a good starting query vector. Since the solution region is a convex cone, the sum of any two solution vectors is also a solution. Thus, one may also choose the sum of $\mathbf{d}^+$ and the user original query vector as an alternative starting vector.

Implementation of Algorithm 1

When there is reason to believe that the starting vector is a good approximation to the solution vector, it may be desirable not to make too much change to the starting vector. In other words, we introduce a bias towards a solution vector that is closer to the starting vector. Thus, instead of modifying the query vector by the whole set $\Gamma(\mathbf{q})$ as suggested in step (iii) of Algorithm 1, one may modify the query vector by adding to it the difference vector $\mathbf{b}_{max}$ with maximum error, i.e. $\mathbf{q} \cdot \mathbf{b}_{max}$ has the largest negative value. More precisely, $\mathbf{b}_{max}$ is defined by $\mathbf{q} \cdot \mathbf{b}_{max} = \max_{\mathbf{b} \in \Gamma(\mathbf{q})} \{-\mathbf{q} \cdot \mathbf{b}\}$. In this case, step

(iii) of Algorithm 1 should be replaced by:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{b}_{\max} \ . \tag{3.5}$$

It can be proved [5, 7] that if the preference relation is weakly linear, this modification of the query vector does not affect the convergence property of Algorithm 1. We actually adopted this modified procedure in our experiments.

Experimental results

The adaptive query formulation approach was evaluated by using five test document collections in various subject areas. These five collections are CACM (3204 documents, 64 queries), CISI (1460 documents, 112 queries), CRAN (1398 documents, 225 queries), INSPEC (12684 documents, 84 queries), and MED (1033 documents, 30 queries). The detailed statistics about these collections can be found in [14, 15]. The diversity of these collections, in terms of size, subject area, and document and query features, may provide a reliable testing environment.

The standard recall and precision measures are used for performance evaluation. The overall performance is determined by computing the average precision at three recall points of 0.75, 0.50, and 0.25, which represent the high, medium, and low recall points, respectively. In order to evaluate the effectiveness of the various adaptive approaches, a *residual collection* system [2, 14] was used. That is, the sample documents were removed so that evaluation was done on the reduced collection.

Throughout all experiments, document vectors with term weights defined by eqn. (3.3) were used. Table 2 summarizes the experimental results for the five test collections. The results of the *base* runs was obtained for the standard vector space model [14, 16] using term weights defined by eqn. (3.3) for both documents and queries. For purposes of comparison, the results obtained from the Ide strategies (i.e., *dec hi* [6, 14]) are also included because these results are among the best computed for the conventional

relevance feedback systems [14]. The column labeled by *Ide–all* represents the results of the Ide method in which the query vector is expanded by all terms, whereas the column labeled by *Ide–com* denotes the case where the query vector is expanded by the most common terms. The detailed specifications of these experiments can be found in [14]. The column labeled by *User* contains the results of the adaptive query formulation approach obtained by choosing the user original query vector as the starting vector. The column *Rel* presents the results obtained by using the sum of the relevant document vectors in the sample as the starting query vector. Finally, the results in column *User+Rel* are obtained by using the sum of the user original query plus all relevant document vectors as the starting query vector.

It can be seen from Table 2 that the performance results listed in column *User+Rel* are superior to the best results obtained by the Ide strategies. Comparing with the base case, the improvements are quite significant, from 53 per cent in CISI collection to 170 per cent in the CRAN collection. However, both the results in columns *User* and *Rel* are slightly worse than those in the *Ide* columns.

For a given sample of documents, there are a number of possible solution vectors. Although a sample solution vector correctly ranks all the documents in the sample, we have no *a priori* knowledge to judge how close the sample solution vector is to a *real* solution vector. (A real solution vector is a vector that correctly ranks all the documents in the entire collection.) Therefore, an arbitrary vector, though not necessarily a sample solution vector, may lie closer to a real solution vector. Such a vector may provide better performance results than a sample solution vector. An Ide vector may in fact be closer to a real solution vector than the sample solution vector generated by the iterative algorithm. This may explain why the Ide method can produce results slightly better than those in the *User* or *Rel* column.

In our method, it is important to choose a good starting query vector when the sample size is small. Therefore, by selecting a starting vector close to a real solution vector,

one may expect to obtain more satisfactory results. Indeed, the use of Ide vectors as starting vectors produces improved performance as shown in column *User+Rel.*

5. Conclusion

In this paper, we present and evaluate a theory of relevance feedback based on the notion of user preference. The experimental results show that the new feedback approach is very effective and comparable to some of the best existing relevance feedback strategies.

The important point to emphasize here is not the improvement of performance, but rather the fact that the approach has a sound theoretical base. In contrast to many feedback approaches, the adaptive method is applicable to binary or weighted document representations and to a multi-level preference structure. Furthermore, it does not depend on an *ad hoc* choice of ranking functions or parameters. We believe that the theory and experiments presented in this paper have demonstrated the importance of a systematic query formulation approach.

Reference

[1] Bollmann, P.; Wong, S.K.M. "Adaptive linear information retrieval models," *Proceedings of the Tenth ACM SIGIR Conference on Research and Development in Information Retrieval,* New Orleans, 157-163; 1987.

[2] Chang, Y.K.; Cirillo; Razon, J. "Evaluation of feedback retrieval using modified freezing, residual collection, and tests and control groups," in Salton, G. (ed.) *The SMART Retrieval System --- Experiments in Automatic Document Processing,* Englewood Cliffs, NJ: Prentice-Hall; 355-370, 1971.

[3] Croft, W.B; Harper, D.J. "Using probabilistic models of document retrieval without relevance information," *Journal of Documentation,* **35:** 285-295; 1979.

[4] Dillon, N; Desper, J. "Automatic relevance feedback in Boolean retrieval systems," *Journal of Documentation,* **36:** 197-208; 1980.

[5] Duda, R.O.; Hart, P.E. *Pattern Classification And Scene Analysis,* New York: Wiley; 1973.

[6] Ide, E. "New experiments in relevance feedback," in Salton, G. (ed.) *The SMART Retrieval System --- Experiments in Automatic Document Processing,* Englewood Cliffs, NJ: Prentice-Hall; 337-354, 1971.

[7] Minsky, M.; Papert, S. *Perceptrons -- An Introduction to Computational Geometry,* Massachusetts: MIT Press; 1969.

[8] Robertson, S.E.; Sparck Jones, K. "Relevance weighting of search terms," *Journal of the American Society for Information Science,* **27:** 129-146; 1976.

[9] Robertson, S.E.; Van Rijsbergen, C.J.; Porter, M.F. "Probabilistic models of indexing and searching," in Oddy, R.N., Robertson, S.E., Van Rijsbergen, C.J., and Williams, P.W. (ed.) *Information Retrieval Research,* London: Butterworths, 35-56, 1981.

[10] Rocchio, Jr. J.J. "Relevance feedback in information retrieval," in Salton, G. (ed.) *The SMART Retrieval System --- Experiments in Automatic Document Processing,* Englewood Cliffs, NJ: Prentice-Hall; 313-323, 1971.

[11] Salton, G. "Relevance feedback and the optimization of retrieval effectiveness," in Salton, G. (ed.) *The SMART Retrieval System --- Experiments in Automatic Document Processing,* Englewood Cliffs, NJ: Prentice-Hall; 324-336, 1971.

[12] Salton, G. (ed.) *The SMART Retrieval System --- Experiments in Automatic Document Processing,* Englewood Cliffs, NJ: Prentice-Hall; 1971.

[13] Salton, G.; Buckley, C. "Parallel text search methods," *Communications of the ACM,* **31:** 202-215, 1988.

[14] Salton, G.; Buckley, C. "Improving Retrieval Performance by Relevance Feedback," *Technical Report,* Department of Computer Science, Cornell University, No. 88-898; 1988.

[15] Salton, G.; Buckley, C. "Term weighting approaches in automatic text retrieval," *Information Processing and Management,* **24:** 513-523, 1988.

[16] Salton, G.; McGill, M.H. *Introduction to Modern Information Retrieval,* New York: McGraw-Hill; 1983.

[17] Salton, G.; Voorhees, E.; Fox, E.A. "A comparison of two methods for Boolean query relevance feedback," *Journal of the American Society for Information Science,* **36:** 200-210, 985.

[18] Stanfill, G; Kahle, B. "Parallel free-text search on the connection machine," *Communications of the ACM,* **29:** 1229-1239, 1986.

[19] Van Rijsbergen, C.J. *Information Retrieval,* London: Butterworths; 1979.

[20] Wong, S.K.M; Yao, Y.Y. "Query formulation in linear retrieval models," To appear in *Journal of the American Society for Information Science,* 1988.

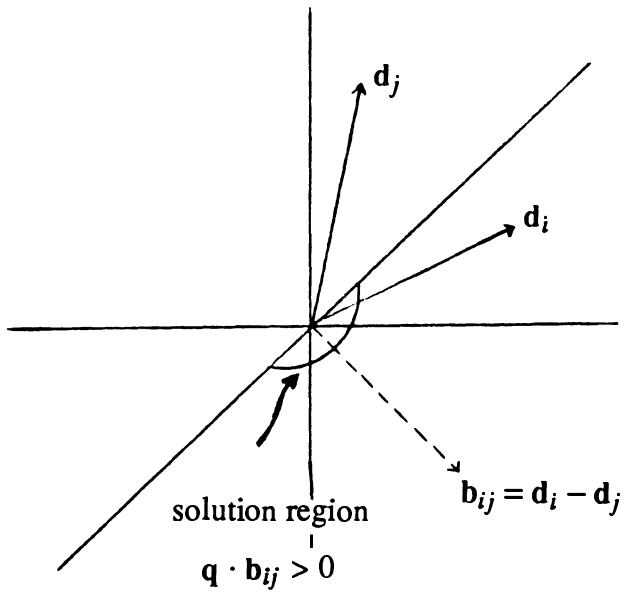
[21] Wong, S.K.M.; Yao, Y.Y.; Bollmann, P. "Linear structure in information retrieval," *Proceedings of the Eleventh ACM SIGIR Conference on Research and Development in Information Retrieval,* Grenoble, France, 219-232; 1988.

[22] Wong, S.K.M.; Ziarko, W.; Raghavan, V.V.; Wong, P.C.N. "Extended Boolean query processing in the generalized vector space model," to appear in *Information Systems,* 1989.

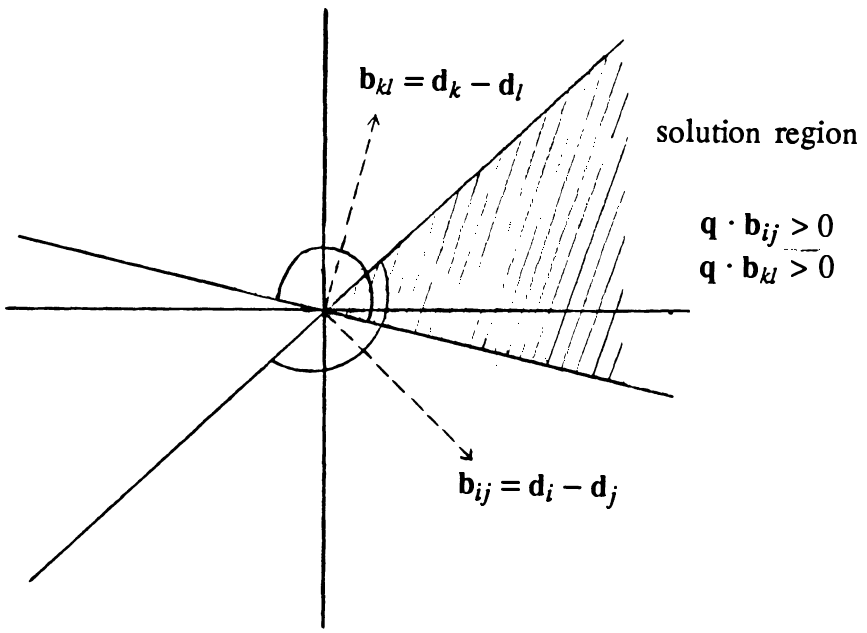
[23] Wu, H.; Salton, G. "The estimation of term relevance weights using relevance feedback," *Journal of Documentation,* **37:** 194-214, 1981.

[24] Yu C.T.; Lee, T.C. "Non-binary independence model," *Proceedings of the Ninth ACM SIGIR Conference on Research and Development in Information Retrieval,* Pisa, Italy, 265-268; 1986.

[25] Yu C.T.; Salton, G. "Precision weighting -- an effective automatic indexing method," *Journal of the ACM,* **23:** 76-85, 1976.



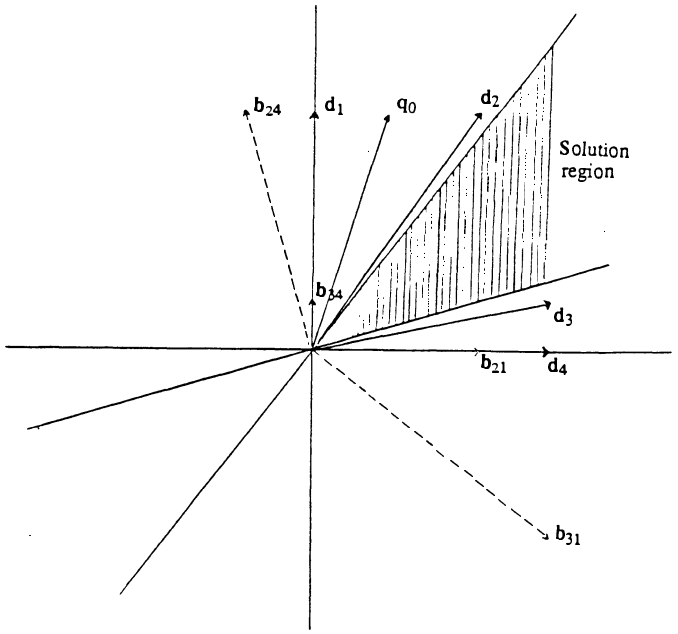
(a)



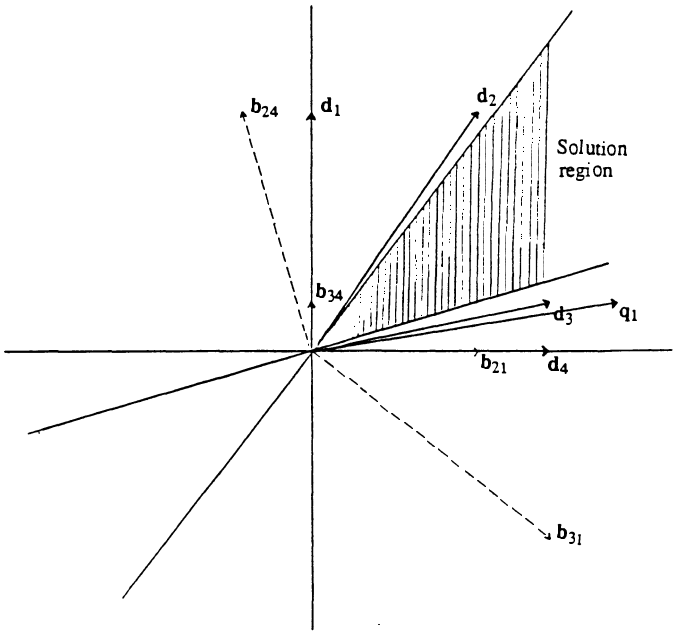
(b)

Figure 1. Determination of solution region for $\mathbf{q} \cdot \mathbf{b} > 0, \mathbf{b} \in \mathbf{B}$

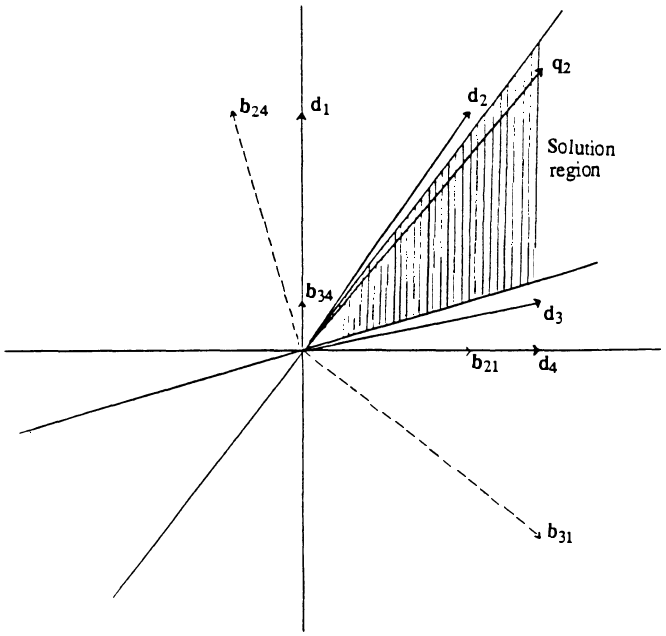
$$\mathbf{B} = \{\mathbf{b}_{ij} = \mathbf{d}_i - \mathbf{d}_j, \mathbf{b}_{kl} = \mathbf{d}_k - \mathbf{d}_l\}$$



(a) Initial configuration



(b) Transformation $\mathbf{q}_1 = \mathbf{q}_0 + (\mathbf{d}_3 - \mathbf{d}_1)$



(b) Transformation $\mathbf{q}_2 = \mathbf{q}_1 + (\mathbf{d}_2 - \mathbf{d}_4)$

Figure 2. A geometric illustration of the iterative algorithm

Iteration	Case 1	Case 2	Case 3
	$q_0 = (0.3, 1.0)$	$q_0 = (1.0, 0.1)$	$q_0 = (0.0, 0.0)$
1	$\Gamma(q_0) = \{b_{31}\}$ $q_1 = q_0 + b_{31} = (1.3, 0.2)$	$\Gamma(q_0) = \{b_{24}\}$ $q_1 = q_0 + b_{24} = (0.7, 1.1)$	$\Gamma(q_0) = \mathbf{B}$ $q_1 = q_0 + \sum_{b \in B} b = (1.4, 0.4)$
2	$\Gamma(q_1) = \{b_{24}\}$ $q_2 = q_1 + b_{24} = (1.0, 1.2)$	$\Gamma(q_1) = \{b_{31}\}$ $q_2 = q_1 + b_{31} = (1,7, 0.3)$	$\Gamma(q_1) = \{b_{24}\}$ $q_2 = q_1 + b_{24} = (1.1, 1.4)$
3	$\Gamma(q_2) = \varnothing$ Solution: (1.0, 1.2)	$\Gamma(q_2) = \{b_{24}\}$ $q_3 = q_2 + b_{24} = (1.4, 1.3)$	$\Gamma(q_2) = \{b_{31}\}$ $q_3 = q_2 + b_{31} = (2.1, 0.6)$
4		$\Gamma(q_3) = \varnothing$ Solution: (1.4, 1.3)	$\Gamma(q_3) = \{b_{24}\}$ $q_4 = q_3 + b_{24} = (1.8, 1.6)$
5			$\Gamma(q_4) = \varnothing$ Solution: (1.8, 1.6)

Table 1. Query formulation process for three starting query vector

	Base	Ide-all	Ide-com	User	Rel	User+Rel
CACM	.1459	.2704 +86%	.2479 +70%	.2482 +70%	.2274 +56%	.2762 +89%
CISI	.1184	.1742 +47%	.1924 +63%	.1751 +48%	.1644 +39%	.1805 +53%
CRAN	.1156	.3011 +160%	.2498 +116%	.2666 +131%	.3155 173%	.3118 +170%
INSPEC	.1368	.2140 +56%	.1976 +44%	.2036 +49%	.2018 +48%	.2245 +64%
MED	.3346	.6305 +88%	.6218 +86%	.6138 +84%	.6262 +87%	.6361 +90%

Table 2. Experimental results for five test collections