

LINEAR STRUCTURE IN INFORMATION RETRIEVAL

S.K.M. Wong, Y.Y. Yao and P. Bollmann

Abstract

Based on the concept of user preference, we investigate the linear structure in information retrieval. We also discuss a practical procedure to determine the linear decision function and present an analysis of term weighting. Our experimental results seem to demonstrate that our model provides a useful framework for the design of an adaptive system.

1. Introduction

Recently, Bollmann and Wong[BOLL87] have proposed an adaptive linear retrieval model. One of the main objectives of their work is to establish a theoretical basis for adopting linear models in information retrieval. Although they investigated the necessary and sufficient conditions for the existence of a linear retrieval function based on measurement theory, some of the important issues have not been fully explored.

In this paper, we focus more on the performance issues of a linear system and use an iterative algorithm (a gradient descent procedure) to compute the coefficients of a linear function. In particular, as an example for illustrating the usefulness of our approach, we present an analysis of term weighting for auto-indexing and show that some of the earlier results[SALT71, SALT83, RIJS79] can perhaps be better understood based on the linear structure. Our experimental results seem to demonstrate that our linear model provides a useful framework for designing an adaptive system for information retrieval.

S.K.M. Wong and Y.Y. Yao, Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2. P. Bollmann, Technische Universitat, Berlin, Germany.

Permission to copy without fee all part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

C 1988 ACM 0-89791-274-8 88 0600 0219 \$ 1,50

The paper is organized as follows. In Section 2, we first review the concept of user preference which forms the basis of our discussion. Then we concentrate on the study of a linear system in Section 3. In Section 4, we show how to construct a linear decision function by adopting an acceptable ranking strategy and compare our results with those obtained by other methods. The experimental results are summarized in Section 5. The main objectives of our experiments are to demonstrate the linear structure of some test document collections and to evaluate the performance of our method.

2. User Preference

Given any two documents in a collection, we assume that a user would *prefer* one to the other or regard both of them as being *equivalent* with respect to his information needs. In other words, it is assumed that the user's judgment on a set of documents D can be described by a (strict) preference relation $<\bullet$, namely,

$$d <\bullet d' \iff \text{the user prefers } d' \text{ to } d, \quad d, d' \in D. \quad (2.1)$$

This relation implies that there also exists an *indifference* relation \sim defined by:

$$d \sim d' \iff (\text{not } (d <\bullet d'), \text{not } (d' <\bullet d)) . \quad (2.2)$$

A preference relation $<\bullet$ which possesses the asymmetric and negatively transitive properties:

- (i) If $d <\bullet d'$, then $\text{not } (d' <\bullet d)$,
 - (ii) If $\text{not } (d <\bullet d')$ and $\text{not } (d' <\bullet d'')$, then $\text{not } (d <\bullet d'')$,
- (2.3)

is called a *weak order*[FISH70, ROBS76]. It can be seen that the indifference relation is an equivalence relation if the preference relation is a weak order.

Fig. 2.1 depicts a preference relation satisfying the axioms (2.3) of a weak order. Documents in the same level (rank) belong to the same equivalence class of the indifference relation \sim , and documents in the higher ranks are preferred to those in the lower ones.

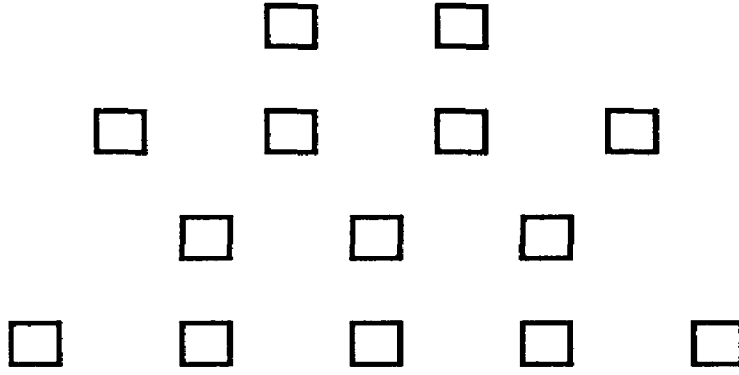


Figure 2.1 A (weak order) preference relation

3. The Linear Structure

It has been shown[BOLL87, FISH70, ROBS76, WONG88] that if a user preference relation is a weak order satisfying some additional conditions one can represent such a relation by a linear decision function. Assume that each document in a collection is described by a column vector \mathbf{d} in a p -dimensional vector space over a set of index terms $\{t_1, t_2, \dots, t_p\}$. Let \mathbf{D} be the set of document vectors. Then, there exists a linear function:

$$g(\mathbf{d}) = \mathbf{q}^T \mathbf{d} = \sum_{i=1}^p w_{t_i} d_{t_i} \quad (3.1)$$

such that

$$\mathbf{d} < \bullet \mathbf{d}' \iff \mathbf{q}^T \mathbf{d} < \mathbf{q}^T \mathbf{d}' \quad , \quad \mathbf{d}, \mathbf{d}' \in \mathbf{D} \quad , \quad (3.2)$$

which implies

$$\mathbf{d} \sim \mathbf{d}' \iff \mathbf{q}^T \mathbf{d} = \mathbf{q}^T \mathbf{d}' \quad . \quad (3.3)$$

($\mathbf{q}^T = (w_{t_1}, w_{t_2}, \dots, w_{t_p})$ denotes the transpose of vector \mathbf{q} .) We say that a linear function defined by eqn. (3.1), satisfying condition (3.2) provides a *perfect* ranking for the documents. That is, based on such a function the documents can be ordered precisely in accordance with the user preference as defined by eqns. (2.1) and (2.2).

In practice, to find a query vector representing both the preference and the indifference relations at the same time can be quite a difficult task. However, we can express condition (3.2) in terms of two single implications:

$$d < \bullet d' \Rightarrow q^T d < q^T d' , \quad (3.4)$$

and

$$d < \bullet d' \Leftarrow q^T d < q^T d' . \quad (3.5)$$

The first implication eqn. (3.4) is equivalent to

$$q^T d \geq q^T d' \Rightarrow \text{not } (d < \bullet d') , \quad (3.6)$$

which has a significant impact on designing retrieval models. Our primary concern in information retrieval is to ensure that those documents more relevant to the user information needs are ranked ahead of those less relevant ones. One is really not interested in making sure that those documents belonging to the same equivalence class of the indifference relation should have the same relevance value. Therefore, the less stringent condition (3.4) may be a more suitable strategy than condition (3.2).

In contrast to the perfect ranking strategy, condition (3.4) only guarantees that less preferred documents will not be listed in front of the more preferred ones. We say that a query vector q satisfying condition (3.4) provides an *acceptable* ranking.

It should perhaps be emphasized here that many existing ranking strategies [COOP68, SALT83, RIJS79] proposed in the past for information retrieval are in fact based on condition (3.4). In this paper, we will adopt the acceptable ranking strategy for the study of linear structure.

In order to facilitate the analysis of the linear structure and for comparison of some previous results, we assume in subsequent discussions that documents are represented by binary vectors. Each component of a document vector corresponding to an index term is assigned a value of 0 or 1 depending on whether the index term is absent or present in the document. Furthermore, it is assumed that the user preference has a simple structure with only two levels. That is, documents are divided into two equivalence classes -- *relevant* and *nonrelevant* (see Fig. 3.1). Thus, the user preference and indifference relations can be expressed as:

$$d < \bullet d' \Leftarrow d \in \text{nrel} , d' \in \text{rel} , \quad (3.7)$$

and

$$d \sim d' \Leftarrow (d \in \text{rel} , d' \in \text{rel}) \text{ or } (d \in \text{nrel} , d' \in \text{nrel}) , \quad (3.8)$$

where *rel* and *nrel* denote the subsets of relevant and nonrelevant documents, respectively. (It should be noted that our analysis here is applicable to non-binary document descriptions with a more complex preference structure.)

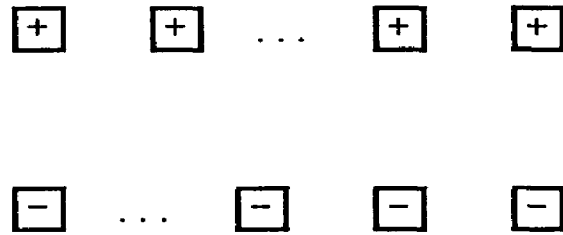


Figure 3.1 A user preference with two levels: relevant(+), nonrelevant(-)

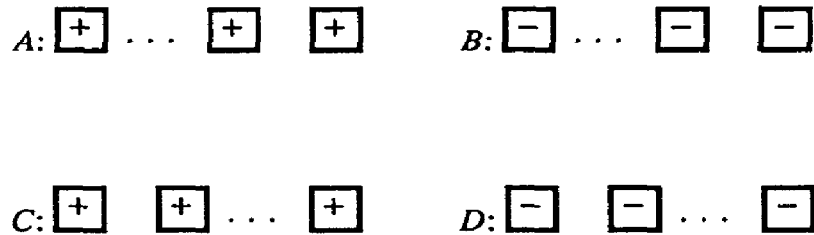


Figure 3.2 Partition of documents based on the value of index term t

Based on a given user preference, one can construct a contingency table for each index term t :

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

where

- N = total number of documents in the collection
- R = total number of relevant documents
- n = number of documents indexed by t
- r = number of relevant documents indexed by t .

Suppose documents are ranked according to the value of index term t . There are *four* possible groups of documents, A, B, C, D as shown in Fig. 3.2. From this classification scheme, we can form different sets of document pairs (Cartesian product sets):

- (i) $A \times D$
- (ii) $(A \times C) \cup (B \times D)$
- (iii) $B \times C$
- (iv) $(A \times B) \cup (C \times D)$.

As indicated in Table 3.1, the first two sets (i) and (ii) of document pairs are ordered correctly according to the acceptable ranking strategy (condition (3.4)), and the last two sets (iii) and (iv) of document pairs do not satisfy this criterion. Among these four sets, only sets (i) and (iii) are independent and moreover they are more significant for document classification. They can be used to construct a measure of the usefulness of each term t . For example, the ratio of the cardinality of set (i) over the cardinality of set (iii):

$$\frac{|A \times D|}{|B \times C|} = \frac{r(N - R - n + r)}{(n - r)(R - r)} \quad , \quad (3.9)$$

provides a plausible measure of how well the documents are ranked by using the index term t alone.

set of document pairs	condition (3.4) \Rightarrow	condition (3.5) \Leftarrow	condition (3.2) $\Leftarrow \Rightarrow$
$A \times D$	yes	yes	yes
$(A \times C) \cup (B \times D)$	yes	no	no
$B \times C$	no	no	no
$(A \times B) \cup (C \times D)$	no	yes	no

Table 3.1 Characteristics of the sets of document pairs induced by t
(yes: satisfying the condition, no: not satisfying the condition)

It is interesting to note that eqn. (3.9) is closely related to the formula for term precision weighting[ROBN76, YU76] based on the probabilistic approach:

$$w^1 = \log \frac{r(N - R - n + r)}{(n - r)(R - r)} \quad . \quad (3.10)$$

Using different assumptions, other formulas have also been suggested for term weighting[ROBN76]:

$$w^2 = \log \frac{r(N - n)}{n(R - r)} , \quad (3.11)$$

$$w^3 = \log \frac{r(N - R)}{R(n - r)} , \quad (3.12)$$

$$w^4 = \log \frac{rN}{Rn} . \quad (3.13)$$

In terms of the cardinalities of sets A, B, C, D , the above formulas can be rewritten as:

$$w^2 = \log \frac{r(R - r) + r(N - R - n + r)}{r(R - r) + (n - r)(R - r)} \quad (3.14)$$

$$w^3 = \log \frac{r(n - r) + r(N - R - n + r)}{r(n - r) + (n - r)(R - r)} \quad (3.15)$$

$$w^4 = \log \frac{rr + r(n - r) + r(R - r) + r(N - R - n + r)}{rr + r(n - r) + r(R - r) + (n - r)(R - r)} . \quad (3.16)$$

It has been pointed out[ROBN76] that for some document collections w^1 gives the best results and performance becomes progressively worse from w^1 to w^4 . If we assume $r(n - r) \geq r(R - r)$, it is not difficult to see that more *noise* is added to w^3 than w^2 and so on. Thus, from quite a different point of view, our analysis here seems to justify the earlier results.

4. A Gradient Descent Procedure

In this section, we present an algorithm for finding a query vector \mathbf{q} satisfying the acceptable ranking strategy[WONG88].

For convenience, the condition (3.4) can be expressed as

$$\mathbf{d} < \bullet \mathbf{d}' \Rightarrow \mathbf{q}^T \mathbf{b} > 0 , \quad (4.1)$$

where $\mathbf{b} = \mathbf{d}' - \mathbf{d}$ is called the *difference vector* for document pair $(\mathbf{d}, \mathbf{d}')$ such that $\mathbf{d} < \bullet \mathbf{d}'$. It is therefore clear that the problem of finding a query vector satisfying criterion (4.1) is equivalent to solving the following system of linear inequalities:

$$\mathbf{q}^T \mathbf{b}_\alpha > 0 , \quad \alpha = 1, 2, \dots, M , \quad (4.2)$$

where M denotes the number of document pairs specified by the preference relation $< \bullet$. An exact or approximate solution of this system of linear inequalities can be found by minimizing the well known perceptron criterion function[DUDA73]:

$$J(\mathbf{q}) = \sum_{\mathbf{b} \in \Gamma(\mathbf{q})} -\mathbf{q}^T \mathbf{b} , \quad (4.3)$$

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

$$\Gamma(\mathbf{q}) = \{ \mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d} \prec \bullet \mathbf{d}', \mathbf{q}^T \mathbf{b} \leq 0 \} . \quad (4.4)$$

We define $J(\mathbf{q}) = 0$ if $\Gamma(\mathbf{q}) = \emptyset$. It has been shown[WONG88] that the function $J(\mathbf{q})$ provides a measure of the total error induced by the query vector \mathbf{q} . Therefore, minimizing $J(\mathbf{q})$ is in fact equivalent to minimizing the error. With the criterion function (4.3), the gradient descent procedure[DUDA73, WONG88] is outlined below:

- (i) Choose an initial query vector \mathbf{q}_0 and let $k = 0$.
- (ii) Let \mathbf{q}_k be the query vector in the k th step. Identify the set of difference vectors $\Gamma(\mathbf{q}_k)$ using eqn. (4.4). If $\Gamma(\mathbf{q}_k) = \emptyset$ (i.e. \mathbf{q}_k is a solution vector), terminate the procedure.
- (iii) Let

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \sum_{\mathbf{b} \in \Gamma(\mathbf{q}_k)} \mathbf{b} . \quad (4.5)$$

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

Example 4.1

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

$$\mathbf{d}_1 = (1, 1, 0, 1)^T$$

$$\mathbf{d}_2 = (1, 0, 1, 0)^T$$

$$\mathbf{d}_3 = (0, 1, 1, 0)^T$$

$$\mathbf{d}_4 = (0, 1, 0, 1)^T$$

Suppose a user preference relation on \mathbf{D} is given by:

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

For this preference relation, one obtains the following set of difference vectors:

$$\mathbf{B} = \{ \mathbf{b}_{21}, \mathbf{b}_{31}, \mathbf{b}_{32}, \mathbf{b}_{24}, \mathbf{b}_{34} \},$$

where

$$\mathbf{b}_{21} = \mathbf{d}_2 - \mathbf{d}_1 = (0, -1, 1, -1)^T$$

$$\mathbf{b}_{31} = \mathbf{d}_3 - \mathbf{d}_1 = (-1, 0, 1, -1)^T$$

$$\mathbf{b}_{32} = \mathbf{d}_3 - \mathbf{d}_2 = (-1, 1, 0, 0)^T$$

$$\mathbf{b}_{24} = \mathbf{d}_2 - \mathbf{d}_4 = (1, -1, 1, -1)^T$$

$$\mathbf{b}_{34} = \mathbf{d}_3 - \mathbf{d}_4 = (0, 0, 1, -1)^T$$

By choosing $\mathbf{q}_0 = \mathbf{0}$, we have $\Gamma(\mathbf{q}_0) = \mathbf{B}$. Thus, in the first iteration the query vector \mathbf{q}_1 is equal to:

$$\begin{aligned}\mathbf{q}_1 &= \mathbf{q}_0 + (\mathbf{b}_{21} + \mathbf{b}_{31} + \mathbf{b}_{32} + \mathbf{b}_{24} + \mathbf{b}_{34}) \\ &= (-1, -1, 4, -4)^T.\end{aligned}$$

Use \mathbf{q}_1 as query vector and we obtain $\Gamma(\mathbf{q}_1) = \{\mathbf{b}_{32}\}$. In the second iteration \mathbf{q}_2 becomes:

$$\mathbf{q}_2 = \mathbf{q}_1 + \mathbf{b}_{32} = (-2, 0, 4, -4)^T.$$

For query vector \mathbf{q}_2 , $\Gamma(\mathbf{q}_2) = \emptyset$. Therefore, $\mathbf{q} = \mathbf{q}_2 = (-2, 0, 4, -4)^T$ is a solution vector. According to eqn. (3.1), the linear function has the values $-6, 2, 4$, and -4 for documents $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$, and \mathbf{d}_4 , respectively. It can easily be verified that to rank the documents according to these numbers satisfies the acceptable ranking strategy. \square

In general, if we choose $\mathbf{q}_0 = \mathbf{0}$ as the initial query in the above gradient descent procedure, at the first iteration one obtains from eqn. (4.4):

$$\Gamma(\mathbf{q}_0) = \{ \mathbf{b} = \mathbf{d}' - \mathbf{d} \mid \mathbf{d} \in \text{nrel}, \mathbf{d}' \in \text{rel} \}, \quad (4.6)$$

and from eqn. (4.5) \mathbf{q}_1 is given by:

$$\mathbf{q}_1 = \mathbf{q}_0 + \sum_{\mathbf{b} \in \Gamma(\mathbf{q}_0)} \mathbf{b} = \sum_{\mathbf{d}' \in \text{rel}} \sum_{\mathbf{d} \in \text{nrel}} (\mathbf{d}' - \mathbf{d}). \quad (4.7)$$

Based on the contingency table in Section 3, eqn. (4.7) can be written as,

$$\begin{aligned}\mathbf{q}_1 &= \sum_{\mathbf{d}' \in \text{rel}} \sum_{\mathbf{d} \in \text{nrel}} (\mathbf{d}' - \mathbf{d}) \\ &= (N - R) \sum_{\mathbf{d}' \in \text{rel}} \mathbf{d}' - R \sum_{\mathbf{d} \in \text{nrel}} \mathbf{d}.\end{aligned} \quad (4.8)$$

The component of the query vector \mathbf{q}_1 assigns a weight w^5 to index term t :

$$\begin{aligned}
 w^5 &= (N - R)r - R(n - r) \\
 &= NR\left(\frac{r}{R} - \frac{n}{N}\right) .
 \end{aligned} \tag{4.9}$$

If $\frac{r}{R} = \frac{n}{N}$ (i.e. $w^5 = 0$), index term t gives no useful information for document classification. If $\frac{r}{R} > \frac{n}{N}$, the presence of term t (i.e. $d_t = 1$) in a document vector will contribute w^5 votes for the relevant class (+). On the other hand, if $\frac{r}{R} < \frac{n}{N}$ and $d_t = 1$, w^5 will contribute $|w^5|$ votes for the nonrelevant class (-). Thus, the absolute value $|\frac{r}{R} - \frac{n}{N}|$ provides an approximate measure of the usefulness of index term t for distinguishing relevant and nonrelevant documents. (Note that RN in eqn. (4.9) is a constant which does not affect the ranking of documents.)

As an example for illustrating the potential of our approach, we present here a brief discussion on term weighting. In indexing, one needs a guideline to select proper terms to index the documents. Since the usefulness of each index term with respect to a user preference is reflected by the value $|\frac{r}{R} - \frac{n}{N}|$, it may provide a criterion for term weighting. However, the ratio $\frac{r}{R}$ is not the same for different user preference relations and not known *a priori*. In order to estimate the average value for $|\frac{r}{R} - \frac{n}{N}|$, one may regard $\frac{r}{R}$ as the value of a random variable X . Based on the physical interpretation of the weighting function w^5 defined by eqn. (4.9), it is reasonable to assume that the probability density function (p.d.f) of this random variable for a *good* index term has the form (indicated by the solid curve) as shown in Fig. (4.1).

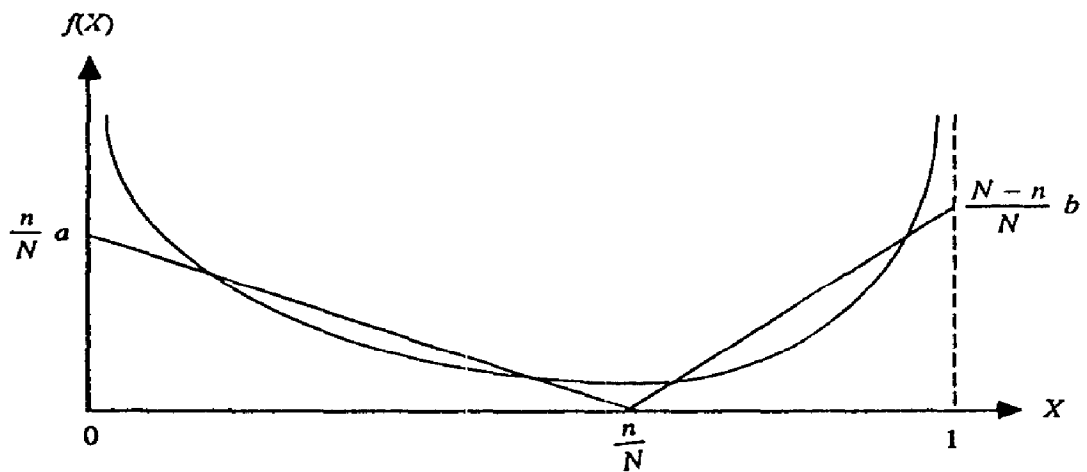


Figure 4.1 The p.d.f. for a good index term

Suppose a linear function is used to approximate the p.d.f (see Fig. 4.1). That is,

$$f(X) = \begin{cases} a(-X + \frac{n}{N}) & 0 \leq X \leq \frac{n}{N} , \\ b(X - \frac{n}{N}) & \frac{n}{N} \leq X \leq 1 , \end{cases} \quad (4.10)$$

where $a > 0$ and $b > 0$. By normalizing the above p.d.f,

$$\int_0^1 f(X) dX = \int_0^{\frac{n}{N}} a(-X + \frac{n}{N}) dX + \int_{\frac{n}{N}}^1 b(X - \frac{n}{N}) dX = 1 ,$$

one obtains:

$$b = \frac{2N^2 - an^2}{(N - n)^2} .$$

Clearly, $\frac{an^2}{2N^2}$ is the probability for $0 \leq X \leq \frac{n}{N}$ and $\frac{b(N - n)^2}{2N^2}$ is the probability for $\frac{n}{N} < X \leq 1$. For further simplification, one may assume that $\frac{an^2}{2N^2} \cong 1 - \frac{n}{N}$ and $\frac{b(N - n)^2}{2N^2} \cong \frac{n}{N}$. Under these approximations, we have

$$a \cong \frac{2N(N - n)}{n^2} , \quad b \cong \frac{2Nn}{(N - n)^2} .$$

Thus, the expected usefulness of term t can be computed from the probability density function as follows:

$$\begin{aligned} E(|X - \frac{n}{N}|) &= \int_0^{\frac{n}{N}} \frac{2N(N - n)}{n^2} (-X + \frac{n}{N}) (-X + \frac{n}{N}) dX + \\ &\quad \int_{\frac{n}{N}}^1 \frac{2Nn}{(N - n)^2} (X - \frac{n}{N}) (X - \frac{n}{N}) dX \\ &= \frac{4}{3} \frac{n(N - n)}{N^2} . \end{aligned} \quad (4.11)$$

It can be shown that the expected value is maximum when $n = 0.5N$. Thus, the above analysis provides a justification for choosing mid-frequency terms in indexing[SALT83, RIJS79].

5. Experimental Results

There are two main objectives in our experiments. The first is to test the effectiveness of the approximate weighting function defined by eqn. (4.9). The second is to test the convergence speed of the gradient procedure. The initial value for q_0 was chosen to be the null vector. Two standard document collections were used with binary vector representation. The ADINUL collection has 82 documents and 35 queries and the CRN4NUL collection has 424 documents and 155 queries. Some of their characteristics are summarized in Table 5.1.

	ADINUL 82 DOC. 35 QUE.	CRN4NUL 424 DOC. 155 QUE.
(document length)		
max	95	164
min	17	13
average	40	53
(relevant documents per query)		
max	33	22
min	1	3
average	4	6
average document frequency	2.9	8.6

Table 5.1 Summary of collection statistics
(document length is measured by the number of terms)

Each collection includes information for every query as to which of the documents are relevant. We used this information (user preference) to compute the values of eqn. (4.9). The standard recall and precision measures are used for performance evaluation of the weighting function. Recall is defined as the proportion of relevant documents retrieved and precision is the proportion of the retrieved documents actually relevant. The overall performance is determined by computing the average precision over all the queries for recall values 0.1, 0.2, . . . , and 1.0. The speed of the gradient descent procedure is measured in terms of the number of iterations required for convergence.

The performance results of our weighting function are given in Table 5.2. These results indicate that the weighting function is very effective for both collections. It should

be emphasized that the above results were obtained by using the complete relevance information and the whole document collection because our objective here is to evaluate the effectiveness of the proposed linear model. We are planning for more experiments to study the inductive process.

Recall	Precision	
	ADINUL	CRN4NUL
0.10	1.0000	0.9990
0.20	1.0000	0.9990
0.30	1.0000	0.9968
0.40	1.0000	0.9968
0.50	1.0000	0.9941
0.60	1.0000	0.9829
0.70	1.0000	0.9623
0.80	1.0000	0.9375
0.90	1.0000	0.8718
1.00	0.9976	0.8377

Table 5.2 Performance of the weighting function

We obtained 34 out of 35 query vectors in the ADINUL collection and 97 out of 155 in the CRN4NUL collection in the first iteration, which satisfy the acceptable ranking strategy. Only one query in ADINUL needs 4 iterations to converge. There are many more documents in CRN4NUL, but within 50 iterations more than 85% of the queries already converged (see Table 5.3). Our preliminary results are quite encouraging and lay some groundwork for designing a viable adaptive system.

No. of iterations(\leq)	10	15	20	25	30	35	40	45	50
No. of queries(converged)	102	105	108	112	115	120	126	130	134
Percentage (%)	66	68	70	72	74	78	81	84	86

Table 5.4 Convergence speed for CRN4NUL collection
(The maximum number of iterations required is 275)

6. Conclusion

Our discussions presented here provide further support that the concept of linear structure is useful for the understanding and development of an information retrieval system.

The next task will be to implement an adaptive system. In order to test the predictive capability of such a system, one needs first to develop an effective method to select an appropriate set of samples (a training set of documents) for generating the query vector by the proposed inductive method.

References

- [BOLL87] Bollmann, P.; Wong, S.K.M. "Adaptive linear information retrieval models." *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 1987.
- [COOP68] Cooper, W.S. "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; 1968.
- [DUDA73] Duda, R.O.; Hart, P.E. *Pattern Classification And Scene Analysis*. John Wiley; 1973.
- [FISH70] Fishburn, P.C. *Utility Theory for Decision Making*, New York: Wiley; 1970.
- [ROBN76] Robertson, S.E.; Sparck Jones, K. "Relevance Weighting of Search Terms." *Journal of the American Society for Information Science*, 27:129-146; 1976.
- [ROBS76] Roberts, F.S. *Measurement Theory*, New York: Academic Press; 1976.
- [SALT71] Salton, G. (ed.) *The SMART Retrieval System --- Experiments in Automatic Document Processing*, Englewood Cliffs, NJ: Prentice-Hall; 1971.
- [SALT83] Salton, G.; McGill, M.H. *Introduction to Modern Information Retrieval*, New York: McGraw-Hill; 1983.
- [RIJS79] Van Rijsbergen, C.J. *Information Retrieval*. London: Butterworth; 1979.
- [WONG88] Wong, S.K.M; Yao, Y.Y. "Query Formulation in Linear Retrieval Models." *submitted for publication*.
- [YU76] Yu, C.T.; Salton, G. "Precision Weighting -- An Effective Automatic Indexing Method." *Journal of the Association for Computing Machinery*. 23(1):76-88; 1976.