

AD 673362

RADC-TR- 68-178



SOME MATHEMATICS OF INFORMATION STORAGE AND RETRIEVAL

Dr. John W. Sammon, Jr.

TECHNICAL REPORT NO. RADC-TR- 68-178

June 1968

This document has been approved
for public release and sale; its
distribution is unlimited.

AD 673362 1968

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded, by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This document may be reproduced to satisfy official needs of US Govt agencies.

1. DATE	
2. BY	
3. FOR	
4. BY	
5. FOR	
6. BY	
7. FOR	
8. BY	
9. FOR	
10. BY	
11. FOR	
12. BY	
13. FOR	
14. BY	
15. FOR	
16. BY	
17. FOR	
18. BY	
19. FOR	
20. BY	
21. FOR	
22. BY	
23. FOR	
24. BY	
25. FOR	
26. BY	
27. FOR	
28. BY	
29. FOR	
30. BY	
31. FOR	
32. BY	
33. FOR	
34. BY	
35. FOR	
36. BY	
37. FOR	
38. BY	
39. FOR	
40. BY	
41. FOR	
42. BY	
43. FOR	
44. BY	
45. FOR	
46. BY	
47. FOR	
48. BY	
49. FOR	
50. BY	
51. FOR	
52. BY	
53. FOR	
54. BY	
55. FOR	
56. BY	
57. FOR	
58. BY	
59. FOR	
60. BY	
61. FOR	
62. BY	
63. FOR	
64. BY	
65. FOR	
66. BY	
67. FOR	
68. BY	
69. FOR	
70. BY	
71. FOR	
72. BY	
73. FOR	
74. BY	
75. FOR	
76. BY	
77. FOR	
78. BY	
79. FOR	
80. BY	
81. FOR	
82. BY	
83. FOR	
84. BY	
85. FOR	
86. BY	
87. FOR	
88. BY	
89. FOR	
90. BY	
91. FOR	
92. BY	
93. FOR	
94. BY	
95. FOR	
96. BY	
97. FOR	
98. BY	
99. FOR	
100. BY	

Do not return this copy. Retain or destroy.

SOME MATHEMATICS OF INFORMATION STORAGE AND RETRIEVAL

Dr. John W. Sammon, Jr.

This document has been approved
for public release and sale; its
distribution is unlimited.

FOREWORD

The research discussed in this document was accomplished under Project 5581, Task 558104.

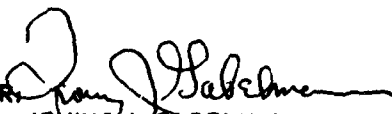
This technical report has been reviewed by the Foreign Disclosure Policy Office (EMLI) and the Office of Information (EMLS) and is releasable to the Clearinghouse for Federal Scientific and Technical Information.

This technical report has been reviewed and is approved.


Approved: FRANK J. TOMAINI
Chief, Information Processing Branch


Approved: JAMES J. DIMEL, Colonel, USAF
Chief, Intelligence and Information Processing Division

FOR THE COMMANDER:


IRVING J. GABELMAN
Chief, Advanced Studies Group

ABSTRACT

This report explains some of the mathematical techniques currently being used and some which are being considered for solving a problem of information storage and retrieval. Basically two problem characterizations are discussed. The first is a statistical description and the other is a vector space characterization. Specifically, we have neglected the interesting area of linguistic analysis which is sometimes used as the basis for information retrieval. Several examples, comments and suggestions are made regarding the use of the various techniques.

TABLE OF CONTENTS

<i>Contents</i>	<i>Page</i>
I. INTRODUCTION	1
II. GENERAL MATHEMATICAL MODEL	2
III. BOOLEAN ALGEBRAIC RETRIEVAL	4
IV. LINEAR STATISTICAL RETRIEVAL	7
V. STATISTICAL ASSOCIATION TECHNIQUES	12
VI. VECTOR SPACE REPRESENTATION	19
VII. DISCRIMINANT ANALYSIS	22
BIBLIOGRAPHY/REFERENCES	26

SECTION I

INTRODUCTION

There exist several hundred different methods for relating search requests to documents contained in a library. It would indeed be impossible to discuss all of these (and probably not desirable); therefore, this report shall be aimed at uncovering the basic mathematics which provide the foundation for most of the retrieval techniques. Specifically, this report will emphasize the mathematics of:

1. Boolean Algebraic Retrieval
2. Linear Statistical Retrieval
3. Statistical Association Techniques for expanding a query and/or for expanding the set of retrieved documents.
4. Vector Space representation of the retrieval process
5. Discriminant Analysis Techniques

It is not intended that this list of subjects exhaust the topic of mathematics for information retrieval. Specifically, we have neglected the very interesting area of linguistic analysis which is sometimes used as the basis for information retrieval. However, it is felt that the operational systems of today and those systems which will be operational in the near future can be adequately described in terms of the mathematics presented here.

The background material for this report was obtained for the most part from the sources listed in the bibliography. The descriptions, examples, comments and suggestions are those of the author.

SECTION II

GENERAL MATHEMATICAL MODEL

It is assumed throughout that each document and each query is characterized by a set of identifiers which include keywords, index terms, descriptors, phases, concepts, etc.. Furthermore, it is assumed that the necessary dictionaries, thesauri and algorithms exist for uniquely representing a document (or query) by an appropriate subset of identifiers.*

Let:

$\{D_i\}$ = set of documents composing the library

Q = query.

$\underline{d}$ = document vector defined on the set of identifiers

$\underline{q}$ = query vector defined on the set of identifiers.

d = No. of documents

i = No. of identifiers.

$\underline{r}$ = retrieval vector

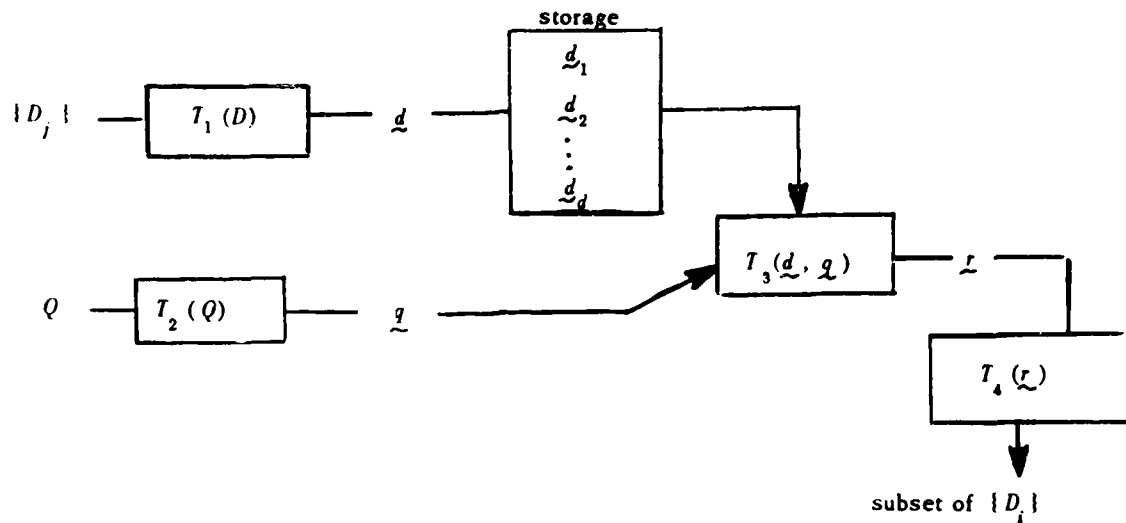
T = transformation

* Salton suggests the following techniques for generating identifiers from a document may not only be necessary but may also be more productive than the computation of higher-order statistical associations.

The following principal procedures are available for vocabulary control and normalization:

1. A stem-suffix cutoff procedure to reduce each text word, or index term, to word stem and word suffix, thus producing a common form for the many different words which exhibit the same stem (e.g., analyzer, analysis, analytic, analyst, etc.).
2. Use of a synonym dictionary, or thesaurus, to replace semantically equivalent words by a common identifier (or concept number).
3. Use of a hierarchical subject arrangement, such as a library classification system, capable of producing for a given concept number various types of related concepts, including more general ones, more specific ones, and a variety of cross references.
4. Use of phrase dictionaries to perform concept groupings by combining pairs or triples of concepts, previously included in a dictionary, into a single, more representative entity (e.g., the concepts "programming" and "language" might be transformed into a more meaningful unit such as "programming language").

Then the general model is



Explanation of Transformations

$T_1(D)$ - is a transformation on the set of all documents which maps each document into the vector space spanned by the identifiers.

$T_2(Q)$ - is a transformation on the set of all queries which maps every query into the vector space spanned by the identifiers.

$T_3(d, q)$ - is a transformation on the set of all document vectors and a query vector which generates a retrieval vector designated r .

$T_4(r)$ - is a transformation on the retrieval vector which generates a subset of the set of all documents

The contents of storage is represented by a C matrix of d row vectors, i.e.

$$C = \begin{bmatrix} \underline{d}_1^t \\ \underline{d}_2^t \\ \vdots \\ \underline{d}_d^t \end{bmatrix} = \begin{bmatrix} C_{ij} \end{bmatrix} \quad \begin{array}{c} \updownarrow \\ d \text{ rows} \end{array}$$

$\leftarrow t \text{ col.} \rightarrow$

Since there are d documents each represented by t identifiers, C is a $d \times t$ matrix.

It is assumed at the outset that the mechanism for generating the C matrix can be defined (that is, the identifiers have been selected and thus $T_1(D)$ can be found). It will turn out that the C matrix provides the fundamental starting point for all the analysis which follows.

SECTION III

BOOLEAN ALGEBRAIC RETRIEVAL

Perhaps the simplest and most widely used retrieval scheme is the Boolean Algebraic technique (sometimes called the Inverted Indices method).

Here the C matrix is a binary matrix

$$C = \begin{matrix} \begin{matrix} \updownarrow \\ d \end{matrix} & \begin{matrix} \leftarrow t \rightarrow \\ \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 \\ 1 \\ \vdots \\ \cdot \end{bmatrix} \end{matrix} \end{matrix}$$

where

$$C_{ij} = \begin{cases} 1 & \text{If identifier } j \text{ is present in document } i \\ 0 & \text{otherwise} \end{cases}$$

From the C matrix, t sets S_i , $i = 1 \dots t$ of documents are formed such that S_i contains those documents in which identifier i appears.

The query vector q , which is generated by the system user, is a ternary vector

$$\underline{q} = \begin{matrix} \begin{bmatrix} 1 \\ x \\ 1 \\ \cdot \\ 0 \\ \cdot \\ \cdot \end{bmatrix} & \begin{matrix} \updownarrow \\ t \end{matrix} \end{matrix}$$

where

$$q_i = \begin{cases} 1 & \text{if identifier } i \text{ is present in query } Q \\ x & \text{if identifier } i \text{ is not present in } Q \\ 0 & \text{if the negation of identifier } i \text{ is present in } Q \end{cases}$$

From the $\underline{q}$ vector the retrieval is obtained by intersection of all sets S_i corresponding to $q_i = 1$, $i = 1, \dots, t$ *

*Negation must be handled differently since the sets S_i contain only those documents which contain identifier i . Therefore, if identifier i is negated in the query, we could generate a new set which contains only those documents not contained in S_i . Although this is simple in theory the operation is time-consuming in practice.

Example 1

let $t = 3$ and $d = 6$

$$C = \begin{matrix} & t_1 & t_2 & t_3 \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{matrix} \Rightarrow \begin{matrix} S_1 = \{d_1, d_3, d_4, d_6\} \\ S_2 = \{d_1, d_2, d_3, d_5\} \\ S_3 = \{d_3, d_5, d_6\} \end{matrix}$$

Suppose the retrieval request is:

Retrieve all documents which contain identifier 1 and identifier 3

Based upon this request $\tilde{q} = \begin{bmatrix} 1 \\ x \\ 1 \end{bmatrix}$

$$q_1 = q_3 = 1$$

Since q_1 and q_3 are 1 we take the logical and of sets S_1 and S_3

$$\begin{aligned} \text{Subset of } \{D_i\} &= S_1 \cap S_3 \\ &= \{d_1, d_3, d_4, d_6\} \cap \{d_3, d_5, d_6\} \\ &= \{d_3, d_6\} \end{aligned}$$

$\therefore$ documents 3 and 6 are retrieved.

In this case the retrieval vector $\mathcal{L}$ is

$$\mathcal{L} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

More complicated Boolean expressions can be obtained using union and negation

Example 2:

Suppose the query request in Example 1 were changed to read:

Retrieve all documents which contain identifiers 1 and 3 or contain identifiers 1 and 2.

In this case the retrieval is accomplished in three steps:

Step 1: Retrieve documents which contain identifiers 1 and 3.

Step 2: Retrieve documents which contain identifiers 1 and 2.

Step 3: Obtain the logical or of the results of step 1 with those of step 2.

Step 1: From Example 1 $\{d_3, d_6\}$

Step 2: From query $q = \begin{bmatrix} 1 \\ 1 \\ x \end{bmatrix} \Rightarrow q_1 = q_2 = 1$

$$\begin{aligned} \text{Subset of } \{D_j\} &= S_1 \cap S_2 \\ &= \{d_1, d_3, d_4, d_6\} \cap \{d_1, d_2, d_3, d_5\} \\ &= \{d_1, d_3\} \end{aligned}$$

Step 3:

$$\begin{aligned} \text{Subset } \{D_j\} &= \{d_3, d_6\} \cup \{d_1, d_3\} \\ &= \{d_1, d_3, d_6\} \end{aligned}$$

Thus, documents 1, 3 and 6 are retrieved and

$$\underline{r} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

The main drawback of Boolean Retrieval is that it is a "yes" or "no" technique – that is, either the query exactly matches a document or else the document is not retrieved. This is a serious deficiency since it is unlikely that the user of the system would have the foresight to specify precisely the query corresponding to the documents which are relevant to him. What is needed is a retrieval vector $\underline{r}$ which is not binary but rather contains elements which indicate the relevance of each document to the query. In this way the documents can be rank ordered according to their relevance in answering the query. This property will be provided by Linear Statistical Retrieval.

SECTION IV

LINEAR STATISTICAL RETRIEVAL

In order to assign relevance numbers to the documents of the library, given the query vector q , the linear statistical model is normally used. Here the retrieval vector r is obtained by performing a linear transformation on the query vector q . The transformation matrix is the identifier document matrix C (usually a modified C matrix as will be seen).

Binary C Matrix

In the simplest case

$$r = Cq \quad \text{where}$$

$$C_{ij} = \begin{cases} 1 & \text{if the identifier } j \text{ is present in document } i \\ 0 & \text{otherwise} \end{cases}$$

$$q_i = \begin{cases} 1 & \text{if identifier } i \text{ is present in query } Q \\ 0 & \text{otherwise} \end{cases}$$

The relevance of the query q to document j would be indicated by the value of

$$r_j = \sum_{i=1}^t C_{ji} q_i$$

Note that r_j is simply the sum of the number of identifiers which are present in both the document and the query.

There exist at least two serious drawbacks to using this simple linear model. The first has to do with the fact that the C matrix is binary. Since we are interested in computing the relevance of a document based upon the query, it would seem that the elements of the C matrix should reflect the relevance to a document given an identifier had occurred in the document. That is to say, C_{ij} should be the relevance of identifier j to document i given that identifier j occurred in document i . The assignment of the C_{ij} 's could be accomplished either manually or algorithmically. If performed manually someone would have to estimate them at the time the document is stored in the library. The assignment could be done algorithmically by setting C_{ij} equal to the relative frequency of the occurrence of identifier j in document i .

The second deficiency of this simple linear model has to do with the fact that the binary relevance coefficient reflects only the number of identifiers which match in the document and the query, and does not take into account the number of mismatches.

For example let

$$\underline{q} = 100111000$$

$$\underline{d}_1^t = 000111000$$

$$\underline{d}_2^t = 111101111$$

where $\underline{d}_i^t$ is the i th row vector of C now

$$r_1 = \underline{d}_1^t \underline{q} = 3 \text{ and}$$

$$r_2 = \underline{d}_2^t \underline{q} = 3$$

$\therefore$ the relevance of document 1 equals the relevance of document 2

This is certainly counterintuitive since $\underline{d}_1^t$ is much closer to the query $\underline{q}$ than is $\underline{d}_2^t$.

Weighted C Matrix

Let C be a weighted matrix

$$C = \begin{matrix} \uparrow & \leftarrow t \rightarrow \\ d & \left[\begin{matrix} C_{ij} \end{matrix} \right] \\ \downarrow \end{matrix} \quad \text{where}$$

C_{ij} = relevance to document i given identifier j

Note that there is no reason why C_{ij} can't be negative. In fact if identifier j never appears in document i it would seem reasonable that $C_{ij} < 0$

Examining the linear model

$$\underline{r} = C \underline{q} \quad \text{where } \underline{q} \text{ is binary}$$

$$r_i = \sum_{j=1}^t C_{ij} q_j$$

the relevance of document i to the query $\underline{q}$ is simply the algebraic sum of the individual relevance coefficients (i.e., C_{ij} 's) which correspond to the identifiers in the query $\underline{q}$.

There are yet a few deficiencies present in our linear model. One of these involves the use of a binary query vector. The user may not consider each identifier in his query vector equally important in which case he may wish to weight the elements of $\underline{q}$.

Weighted C Matrix and Weighted q Vector

The linear model is

$$\underline{r} = C \underline{q} \text{ where } C \text{ is weighted as before}$$

and

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_t \end{bmatrix}$$

where

$$q_i = \text{the weight assigned to identifier } i.$$

Our linear model has generalized a good deal; however, there still exists a very important and fundamental question which has not been answered. This question involves the constraints upon the weights used in the C matrix and in the q vector. This question is treated in a vague way in the literature; however, it is of fundamental importance since the system retrieval will vary widely depending upon its answer.

In order to clarify (and answer) the problem of constraints, the linear retrieval method will now be interpreted as an operation in a linear vector space.

Linear Vector Space Interpretation

Given the weighted C matrix,

$$C = \begin{matrix} \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \\ d \\ \downarrow \end{matrix} \begin{bmatrix} C_{ij} \end{bmatrix} \end{matrix} = \begin{bmatrix} \underline{d}_1^t \\ \underline{d}_2^t \\ \vdots \\ \underline{d}_d^t \end{bmatrix}$$

represents the t dimensional row vectors as

$$\underline{d}_i^t = [C_{i1} \ C_{i2} \ \dots \ C_{it}]$$

Here the documents are represented as t dimensional vectors in the vector space spanned by the t identifiers.

The query vector can be represented in the same space as a t dimensional vector.

The Linear Retrieval Model can now be interpreted as a set of vector operations in the linear vector space

$$\underline{r} = C \underline{q}$$

thus

$$r_i = \sum_{j=1}^t C_{ij} q_j \quad \text{or equivalently}$$

$$r_i = \langle \underline{d}_i, \underline{q} \rangle = \underline{d}_i^t \underline{q}$$

Thus, the relevance of document i to the query $\underline{q}$ is simply the inner product of the i th document vector with the query vector. Here it is obvious that the measure of relevance is directly related to the measure of "closeness" of the document vector to the query vector in the vector space. At this point one might be tempted to drop the Linear Model $\underline{r} = C \underline{q}$ in favor of using other metrics which measure the "closeness" between two vectors (such as Euclidean Distance, Box Car Norm, etc.). However, since we are concerned with analyzing the Linear Model we shall focus our attention on the inner product as our measure of "closeness" (or relevance).

Example 3:

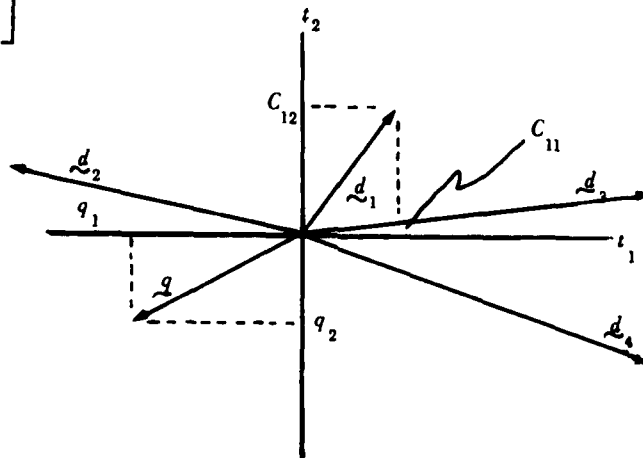
Let $d = 4$ and $t = 2$. That is, the library contains 4 documents each represented by 2 identifiers. Here the linear vector space is spanned by 2 identifiers and so the space has dimension 2.

$$C = \begin{matrix} & \begin{matrix} t_1 & t_2 \end{matrix} \\ \begin{matrix} \underline{d}_1 \\ \underline{d}_2 \\ \underline{d}_3 \\ \underline{d}_4 \end{matrix} & \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \\ C_{31} & C_{32} \\ C_{41} & C_{42} \end{bmatrix} \end{matrix}$$

$$\begin{aligned} \underline{d}_1^t &= [C_{11} \quad C_{12}] \\ \underline{d}_2^t &= [C_{21} \quad C_{22}] \\ \underline{d}_3^t &= [C_{31} \quad C_{32}] \\ \underline{d}_4^t &= [C_{41} \quad C_{42}] \end{aligned}$$

The query vector is similarly 2-dimensional

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$



We can now return to the important question of normalization. It has been shown that the Linear Statistical Model requires that the inner product between a document vector and the query vector be the relevance measure for that document. The inner product is given by:

$$r_i = \underline{d}_i^t \underline{q} = |\underline{d}_i| |\underline{q}| \cos \theta$$

where

$$\begin{aligned} |\underline{v}| &= \text{magnitude of the vector} \\ &= \sqrt{\underline{v}^t \underline{v}} \end{aligned}$$

and

θ is the angle between the vectors.

Notice that the inner product is directly proportional to the product of the magnitudes of the vectors. Herein lies the normalization problem. A user who assigns large weights to his query vector may get a completely different response than another user who uses the same relative weights but uses weights of a much lower magnitude. The documents will have the same rank ordering; however, the documents retrieved will depend upon $\underline{q}^t [\underline{L}]$. In view of this problem the most reasonable process is to require that the document vectors and the query vectors be normalized to unit vectors. In this case the measure of "closeness" (or relevance) is determined only by the cosine of the angle between the two vectors, i.e.,

$$r_i = \underline{d}_i^t \underline{q} = \cos \theta$$

Now the constraints on the weights of the C matrix and the $\underline{q}$ vector are specified. That is

$$|\underline{d}_i| = 1 = \sum_{j=1}^t C_{ij}^2 \quad \text{and}$$

$$|\underline{q}| = 1 = \sum_{i=1}^t q_i^2$$

At this point we have generalized the Linear Statistical Model to the point where the C matrix and the $\underline{q}$ vector are weighted and properly normalized. However, there still exists many deficiencies in the model. In particular consider the problem of formulating the query vector. Ideally the user should construct that query which best matches all the document vectors which are of interest to him. However, he cannot be expected to know the relevance of each identifier to every document and further, he will not be expected to assign a weight to every possible query identifier. To meet this need, some automatic Statistical Association Techniques can be employed to modify a user's query so as to generate a larger, more comprehensive, query. It will be shown that the same techniques used to broaden a query can be used to broaden the system response.

SECTION V

STATISTICAL ASSOCIATION TECHNIQUES

The central idea behind using Association Techniques (these techniques are sometimes called "clustering" or "clumping") is to add identifiers to a query by using the pair-wise statistical relations which exist between identifiers.

Therefore we wish to obtain a $t \times t$ matrix which reflects the similarity between the identifiers. Let S_t be such a similarity matrix

$$S_t = \begin{matrix} & \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \\ t \\ \downarrow \end{matrix} \\ \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \\ t \\ \downarrow \end{matrix} & \begin{bmatrix} S_{ij} \end{bmatrix} \end{matrix}$$

Here the ij th element indicates the degree of similarity between the i th identifier and the j th identifier. There exist many ways of generating similarity matrices but each method must use the association information inherently contained within the documents of the library. All this information is contained in the C matrix and so the C matrix is always used as the starting point. A useful Similarity Matrix is the Covariance Matrix* defined as

$$S_t = \begin{matrix} & \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \\ t \\ \downarrow \end{matrix} \\ \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \\ t \\ \downarrow \end{matrix} & \begin{bmatrix} S_{ij} \end{bmatrix} \end{matrix}$$

where

$$S_{ij} = \frac{1}{d} \sum_{k=1}^d (C_{ki} - \bar{C}_i) (C_{kj} - \bar{C}_j)$$

where

$$\bar{C}_i = \frac{1}{d} \sum_{k=1}^d C_{ki} \quad \left\{ \begin{array}{l} \text{This is the average of the} \\ \text{ith column vector of the} \\ \text{C matrix} \end{array} \right.$$

$\therefore S_{ij}$ is simply the covariance between the i th identifier and the j th identifier.

* Again the literature is particularly vague on the subject of similarity measures. I suggest two other possible measures as follows:

$$1. \quad S_t = C^T C$$

Here S_{ij} = inner product of the i th column vector of C with the j th column vector

Once the similarity matrix has been generated we can interpret the ij th element as the strength of the association between identifier i and identifier j . It is convenient to represent this information as an undirected** graph where the nodes represent the identifiers and the weight of the links represent the association coefficients between the nodes (i.e., identifiers) which they connect. That is the links on this graph indicate the "strength" of the 1st order associations between nodes. By taking products of these "strengths" along paths of length two second order associations can be obtained. For example in Figure 1 a second order association between node i and node k is given by the product

$$S_{ij} S_{jk}$$

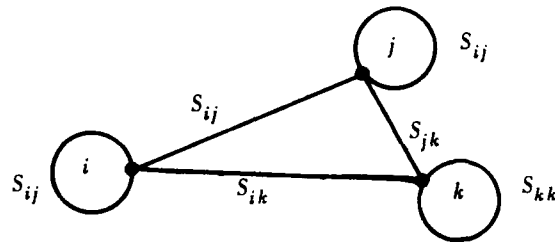


Figure 1

The sum of all second order associations between node i and node j is obtained by examining the ij th element of the matrix obtained by squaring the Similarity Matrix = S^2 . Therefore, second order associations are obtained from

$$S^2 = SS = \begin{matrix} \uparrow & \leftarrow t \rightarrow \\ \downarrow & \left[S_{ij}^{(2)} \right] \end{matrix}$$

* Continued

of C . The previous discussion on normalization is pertinent here.

2. Another measure of similarity could be obtained by considering the Euclidean distance between the identifier vectors (i.e., the column vectors of C) in the space spanned by the documents. Here we would have t identifier vectors represented in a d -dimensional vector space spanned by the d documents

$$S_{ij} = | \underline{t}_i - \underline{t}_j |^{-1}$$

where $\underline{t}_i$ is the i th column vector of C

** Note that the graph is undirected since $S_{ij} = S_{ji}$, that is the association from i to j . If the similarity matrix were not symmetric then the graph would be directed.

where

$$S_{ij}^{(2)} = \sum_k S_{ik} S_{kj} \quad \left\{ \begin{array}{l} \text{sum of all second order} \\ \text{associations between} \\ \text{node } i \text{ and node } j \end{array} \right.$$

Any order association between identifiers can be obtained by raising the S matrix to the desired power. That is, the n th order associations are obtained from

$$S^n = \overset{\leftarrow n \text{ times} \rightarrow}{S \cdot S \cdots S}$$

Returning to the question of expanding our query vector q by using higher order associations between identifiers, it is clear that we must have some means of taking into account the relative importance of the different order associations. It would seem reasonable that 1st order associations are more important than second which in turn are more important than third and so on. To accomplish this, the following method is suggested by Salton.

Let $\underline{q}^*$ = expanded query vector
 $\underline{q}$ = original query vector
 S_t = similarity matrix
 α = positive constant less than one $0 < \alpha < 1$

then define

$$\underline{q}^* = [I + \alpha S_t + (\alpha S_t)^2 + (\alpha S_t)^3 + \cdots] \underline{q}$$

Here we have weighted the higher order associations by the appropriate power of α and since $0 < \alpha < 1$, α^n is monotonically decreasing as n increases.

Example 4: Second Order Association.

For the purpose of simplicity, suppose that the similarity matrix has been threshold at the level θ .

That is

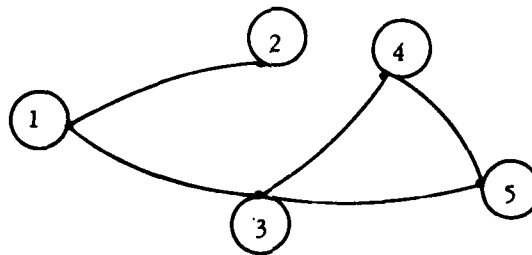
$$S_{ij} = \begin{cases} 1 & \text{if } S_{ij} > \theta \\ 0 & \text{if } S_{ij} < \theta \end{cases}$$

Then the similarity matrix is a binary matrix

Let

$$S_{threshold} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

the resulting graph is



The presence of a link indicates that the connected nodes are associated in $S_{threshold}$

Now

$$S^2 = SS = \begin{bmatrix} 3 & 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 4 & 3 & 3 \\ 1 & 0 & 3 & 3 & 3 \\ 1 & 0 & 3 & 3 & 3 \end{bmatrix}$$

S^2 yields the second order associations between pairs of nodes. Since $S_{threshold}$ is binary the ij element of the S^2 matrix is simply the number of paths of length two between node i and node j . This may be verified by examination of the graph.

Once the expanded query vector is obtained (i.e., $\underline{q}^*$ above) it must be normalized such that $|\underline{q}^*| = 1$.

The exact same techniques used for expanding the query vector can be used to expand the set of retrieved documents. Suppose that Linear Statistical Retrieval is used so that

$$\underline{r} = C\underline{q}$$

Remember that the elements of the $\underline{r}$ vector are the relevance indicators for the documents, that is

$$r_i = \text{relevance of document } i$$

Now a new relevance vector can be obtained by

$$\underline{r}^* = [I + (\alpha S_d) + (\alpha S_d)^2 + (\alpha S_d)^3 + \dots] \underline{r}$$

where S_d is a $d \times d$ similarity matrix defined on the documents. The ij th element of S_d is the association of the i th document to the j th document.

Another very interesting way of expanding a retrieval set is obtained by using bibliographic citations.¹ The mechanism for doing this is quite similar to the methods used for association.

We begin by obtaining a $d \times d$ binary matrix where the documents are chronologically ordered along the rows and columns. That is

$$M = \begin{matrix} & \begin{matrix} d_1 & d_2 & \dots & d_d \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_d \end{matrix} & \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} \end{matrix}$$

where

d_1 is the oldest document

d_2 the next oldest and so on.

The rows represent the documents being cited and the columns the source of the citation. Therefore

$$M_{ij} = \begin{cases} 1 & \text{if document } j \text{ cites document } i \\ 0 & \text{otherwise.} \end{cases}$$

The elements on or below the diagonal are zero since a document can only cite a previously published document and further no document cites itself. Now proceeding as before, higher order linkages can be examined by taking higher order powers of the M matrix.

For example taking the n th power of the M matrix and examining the ij element of M^n ($i < j$ and $j > n$)* we can obtain the sum of n th order linkages between document i and document j .

* It is easy to show that $M_{ij}^{(n)} = 0$

where

$$M^n = [M_{ij}^{(n)}]$$

For $j \leq n$.

Now by examining the M^n matrix, all documents which exhibit strong n th order links are collected into groups. Now we can expand the retrieved document set by adding the document groups which are strongly linked to the originally retrieved documents.

For an example of a proposed Linear Statistical Retrieval System see reference 2.

A Statistical Viewpoint

Viewing the problem of information retrieval from a statistical point we would like to compute the probability of a document being relevant given the query vector $\underline{q}$. That is to say we should like to compute

$$P(d_i/\underline{q}) = \text{Probability that document } i \text{ is relevant given query } \underline{q}$$

Following the usual procedure we employ Bayes Rule to get

$$P(d_i/\underline{q}) = \frac{P(\underline{q}/d_i) P(d_i)}{P(\underline{q})} = \frac{P(\underline{q}, d_i)}{P(\underline{q})}$$

Note that in theory $P(\underline{q}/d_i)$ could be estimated. $P(\underline{q}/d_i)$ is the probability of query $\underline{q}$ given that document i is relevant. We could accomplish this by observing the relative frequency of the $\underline{q}$ vector under the condition that document i is relevant to the user generating $\underline{q}$. Of course this procedure would have to be done many times for all possible query vectors. Clearly this is impossible in any practical sense.

$P(d_i)$ could be estimated by the relative frequency with which document i is considered relevant.

$P(\underline{q})$ is a constant given any query and therefore poses no problem of estimation.

In order to simplify our problem let us assume that the identifiers composing the $\underline{q}$ vector are statistically independent. In this case

$$P(\underline{q}/D) = \prod_k P(q_k/D)$$

Now

$$P(d_i/\underline{q}) = \frac{\prod_{k=1}^t P(q_k/d_i) P(d_i)}{\prod_k P(q_k)} = (\text{const.}) P(d_i) \prod_{k=1}^t P(q_k/d_i)$$

Since the log function is a monotonic function of its argument we can use $\log P(d_i/\underline{q})$ to estimate the relevance of document i . Taking the log

$$\log P(d_i/\underline{q}) = \text{const.} + \log P(d_i) + \sum_{k=1}^t \log P(q_k/d_i)$$

Now it is assumed that we can estimate the $P(q_k/d_i)$ in much the same way we obtained the weights in the C matrix. Given a $\underline{q}$ vector we can get $P(q_k/d_i)$ and the relevance factor can be obtained as a linear function.

$$r_i = \log P(d_i/q) = \text{const.} + \log P(d_i) + \sum_{k=1}^L \log P(q_k/d_i)$$

It should be noted that this simple linear relation is obtained under the assumption of statistical independence. For further discussion of statistical techniques see reference 3.

Before going on to other topics it is worth noting that the highest order statistics considered thus far are only second order. Even though higher order associations were employed they were generated taking account only of second order statistical relationships.

SECTION VI

VECTOR SPACE REPRESENTATION

The vector space representation has already been given where the d documents are represented as t dimensional vectors in the space spanned by the t identifiers. Similarly the query vector is represented as a t dimensional vector in the same space. If we were to implement the Linear Statistical System previously described we would have to store the d t -dimensional document vectors. This would require $d \times t$ numbers. Typically

$$d = 500,000 \text{ documents}$$

$$t = 1000 - 10,000 \text{ identifiers.}$$

$$\therefore d \times t = 5 \times 10^8 - 5 \times 10^9 \text{ numbers.}$$

If each coordinate were represented by 5 bits, the system would have to store up to 2.5×10^{10} bits. to represent d documents in the t dimensional vector space.

Because of the size of the required storage we are motivated to search for lower dimensional vector spaces in which we can represent the document vectors and still perform meaningful retrieval.

One possibility for accomplishing this function is to find a K dimensional subspace of the t dimensional vector space such that the K -space is "best" K dimensional space in the least squares sense.

The solution to this problem is well known in the field of linear algebra. It turns out that the solution is given by the K eigenvectors corresponding to the K largest eigenvalues of the covariance matrix defined earlier.

Let

$$S_t = \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \downarrow \\ t \end{matrix} \begin{bmatrix} S_{ij} \end{bmatrix} = \text{Covariance Matrix}$$

where

$$S_{ij} = \frac{1}{d} \sum_{k=1}^d (\bar{C}_{ki} - \bar{C}_i) (\bar{C}_{kj} - \bar{C}_j)$$

$$\bar{C}_i = \frac{1}{d} \sum_{k=1}^d C_{ki}$$

where

$$C = \begin{matrix} \begin{matrix} \leftarrow t \rightarrow \\ \uparrow \\ d \\ \downarrow \end{matrix} \left[\begin{matrix} \end{matrix} \right] \end{matrix} = \text{document identifier matrix}$$

Then, solving the following eigenvector problem yields the appropriate K eigenvectors

$$S \underline{e} = \lambda \underline{e}$$

Where $\underline{e}$ is a t dimensional eigenvector

Having solved this problem we then define a linear transformation Λ as follows

$$\Lambda = \begin{matrix} \begin{matrix} \leftarrow K \rightarrow \\ \uparrow \\ t \\ \downarrow \end{matrix} \left[\begin{matrix} \underline{e}_1 & \underline{e}_2 & \dots & \underline{e}_K \end{matrix} \right] \end{matrix}$$

Where the column vectors are the K eigenvectors obtained above.

Now the document vectors are projected into the K dimensional subspace by the linear transformation Λ i.e.

$$C' = C \Lambda$$

Since C is $d \times t$ and Λ is $t \times K$, C' is $d \times K$. Therefore, we need to represent each of the d document vectors in the K space by only K numbers in lieu of the t numbers we originally needed. Since $K < t$ we have saved storage. The typical savings might be a factor of 1000.

Now when a query vector is generated we map it into the K space by

$$\underline{q}' = \Lambda^T \underline{q} \quad \text{where } \underline{q}' \text{ is } K \times 1$$

Retrieval is accomplished in the K -space just as before, i.e.

$$\underline{r} = C' \underline{q}'$$

Note that the vectors need not be re-normalized in the K -space since Λ is an orthogonal transformation which means that the vector magnitudes are invariant under this transformation*

*To show this let

$$\underline{y} = \Lambda \underline{z}$$

The magnitudes are $\underline{y}^T \underline{y}$ and $\underline{z}^T \underline{z}$

$$\underline{y}^T \underline{y} = \underline{z}^T \Lambda^T \Lambda \underline{z}$$

but since Λ is orthogonal $\Lambda^T = \Lambda^{-1}$; $\Lambda^T \Lambda = I$

$$\therefore \underline{y}^T \underline{y} = \underline{z}^T I \underline{z} = \underline{z}^T \underline{z}$$

Another interesting method for reducing the dimensionality of the vector space has been proposed by Assorio⁴. Assorio begins the problem by grouping the documents in the library into a number of fields. He then asks several experts working in a particular field to generate a t dimensional vector which typifies that field. This is accomplished by each expert going through all t identifiers and ranking their importance (or relevance) to his field. An average vector for each field is then obtained by averaging together the vectors generated by each expert in that field. If there are f fields, this process will generate f t dimensional vectors.

The underlying concept here is that the experts within a field will draw upon their knowledge and experience to generate a "good" representative vector for that field. Then the averaging of the expert vectors within the field together will further smooth the effects of each individual. The result should then be the "best" t dimensional vector for that field. Here "best" means that the resultant field vector fits all the document vectors in the "best" way possible.

We can represent Assorio's information at this point by the F matrix

$$F = \begin{matrix} & \xleftarrow{f} & \\ \uparrow & \begin{bmatrix} \vec{f}_1 & \vec{f}_2 \end{bmatrix} & \\ f & & \\ \downarrow & & \end{matrix}$$

where the column vectors are the average field vectors.

Now since the dimensionality of the space cannot exceed the $\min \{ t, f \}$ and typically $f \ll t$, we can represent our document vectors and query vectors in an f dimensional space. It may still be possible to solve our problem in an even smaller dimensional space by using the least squares subspace fit as described earlier. Assorio accomplishes a similar subspace fit using Factor Analysis.

In any case, if the f dimensional space defined by the f field vectors is not reduced further a Schmidt Orthogonalization procedure should be used in order to define an orthogonal f -dimensional subspace for representing the d documents.

Using this technique we have reduced the required storage from $d \times t$ down to $d \times f$ which typically is a factor of 1000 times!

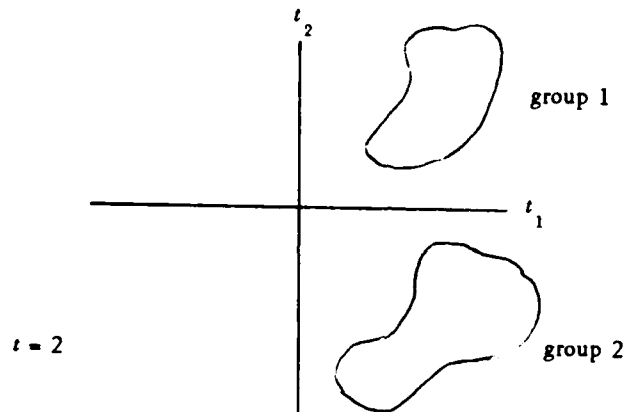
SECTION VII

DISCRIMINANT ANALYSIS

Up to this point our concern in dimension reduction has focused on fitting the document vectors in a subspace of the t dimensional space.

Here the emphasis changes abruptly. Our concern in this section is to find a $p - 1$ dimensional subspace which is optimal for discriminating between p groups (or classes) of documents. This problem is discussed by Williams⁵ and its solution is classically obtained using Discriminant Analysis.

To begin a discussion on Discriminant Analysis it is best to select a simple case so that the basic ideas are not clouded by the algebra. Therefore, assume that $p = 2$, that is, there are two groups of document vectors located in the t dimensional space.



Now we wish to project these two groups orthogonally onto a line so that the variation between the projected groups is as large as possible, relative to the variation within the two projected groups. The problem is to find the direction of projection which will accomplish this. It will turn out that this is equivalent to finding that direction of projection which maximizes the distance between the projected means relative to the sum of the variabilities of the projected groups.

Definitions:

μ_1 = mean vector of group 1 $t \times 1$

μ_2 = mean vector of group 2 $t \times 1$

Δ = $\mu_1 - \mu_2$

B = within Groups Scatter Matrix $t \times t$

B = between Groups Scatter Matrix $t \times t$

S = pooled Scatter Matrix $t \times t$

$X^{(i)}$ = data matrix for group i $i = 1, 2$ with mean μ_i subtracted from each column.

$$X^{(i)} = \begin{bmatrix} \tilde{x}_1^i & \tilde{x}_2^i & \dots & \tilde{x}_{NT}^i \end{bmatrix} = \begin{bmatrix} x_{ij}^{(i)} \end{bmatrix} \quad i = 1, 2$$

Now

$$W = \begin{bmatrix} w_{ij} \end{bmatrix}$$

$$w_{ij} = U_{ij}^{(1)} + U_{ij}^{(2)}$$

where

$$U_{ij}^{(g)} = \sum_{r=1}^N (X_{ir}^{(g)} - \bar{X}_i^{(g)}) (X_{jr}^{(g)} - \bar{X}_j^{(g)}) \quad g = 1, 2$$

$$= X^{(g)} X^{(g)T}$$

$$W = X^{(1)} X^{(1)T} + X^{(2)} X^{(2)T}$$

Notice that $X^{(g)} X^{(g)T}$ differs from the covariance matrix only by a $1/(N_g-1)$ normalizing factor.

The S matrix is computed in a similar fashion by first pooling both groups together, then computing the covariance matrix Σ .

That is $S = (N_1 + N_2 - 1) \Sigma$

Now

$$S = B + W$$

so that B can be computed

$$B = S - W$$

For the case where there are only two groups

$$B = \frac{N_1 N_2}{N_1 + N_2} \tilde{\Delta} \tilde{\Delta}^T$$

The formal statement of the problem is: Find a direction A which maximizes the projected between class scatter for a fixed value of the projected within class scatter. It is easy to show that the projected scatters are given by:

$$\begin{aligned}\underline{A}^T B \underline{A} &= \text{projected between class scatter} \\ \underline{A}^T W \underline{A} &= \text{projected within class scatter}\end{aligned}\quad (1)$$

$\therefore$ we wish to maximize $\underline{A}^T B \underline{A}$ under the constraint that $\underline{A}^T W \underline{A}$ remain constant. This is conveniently handled using Lagrange multipliers.

$$\therefore \frac{\partial}{\partial \underline{A}} \left[\underline{A}^T B \underline{A} - \lambda (\underline{A}^T W \underline{A} - \text{Const}) \right] = 0 \quad (2)$$

λ = Lagrange Multiplier

This gives

$$\begin{aligned}B \underline{A} - \lambda W \underline{A} &= 0 \quad \text{or} \\ (B - \lambda W) \underline{A} &= 0\end{aligned}\quad (3)$$

In order for a non-trivial solution to exist (i.e. other than $\underline{A} = 0$) the determinant of $(B - \lambda W)$ must vanish.

$$|B - \lambda W| = 0$$

This problem is recognized as the generalized form of an eigenvector problem where λ is an eigenvalue.

Now extending the problem to the case of P groups, the discriminant analysis solution will result in the identical eigenvector problem where the $(P - 1)$ eigenvectors are the desired optimal subspace for discrimination. See Wilks⁶ (Pg 576) for further discussion.

For the special case of two groups the solution $\underline{A}$ can be found directly from equation 2 by substituting $B = K \underline{\Delta} \underline{\Delta}^T$ into the expression $\underline{A}^T B \underline{A}$ ($K = \text{const.}$)

i.e.

$$\underline{A}^T B \underline{A} = K \underline{A}^T \underline{\Delta} \underline{\Delta}^T \underline{A} = K (\underline{\Delta}^T \underline{A})^2$$

$$\frac{\partial}{\partial \underline{A}} \left[K (\underline{\Delta}^T \underline{A})^2 - \lambda (\underline{A}^T W \underline{A} - \text{Const.}) \right] = 0$$

$$= (K \underline{\Delta} \underline{\Delta}^T) \underline{A} - \lambda W \underline{A} = 0$$

$$\underline{A} = \alpha W^{-1} \underline{\Delta}$$

where $\alpha = \text{const.} = \frac{K \underline{\underline{A}}^T \underline{\underline{A}}}{\lambda}$.

Therefore, the direction of $\underline{\underline{A}}$ is obtained which solves the discrimination problem between two groups.

BIBLIOGRAPHY/REFERENCES

1. Yagi, Eri, "An Application of a Type of Matrix to Analyze Citations of Scientific Papers," American Documentation, Jan. 1965,
2. Stiles, H. E., "The Association Factor in Information Retrieval," JACM, Vol. 8, 1961.
3. Maron, M. E. and Kuhns, J. L., "On Relevance Probabilistic Indexing and Information Retrieval," JACM, Vol. 7, 1960.
4. Assorio, P. G., "Classification Space Analysis," RADC-TDR-64-287, Oct. 64, AD 608 034.
5. Williams, J. H., "Discriminant Analysis for Content Classification," Feb. 1966, RADC-TR-66-6, Feb. 66, AD 630 127.
6. Wilks, S. S., Mathematical Statistics, John Wiley & Sons, 1962.
7. Assorio, P. G., "Dissemination Research," RADC-TDR-65-314, Dec. 65, AD 625 905.
8. Cooley & Lohnes, Multivariate Procedures for the Behavioral Sciences, John Wiley & Sons, 1962.
9. Kanal, L. N., "Adaptive Modelling of Likelihood Classification," RADC-TR-66-190, Jun 66, AD 636 519.
10. Salton, G., "Progress in Automatic Information Retrieval," IEEE Spectrum, Aug. 1965.
11. Salton, G., "Language Data Processing," Harvard Summer School Note, Aug. 10-21, 1964.
12. Hayes, R. M., "Mathematical Models for Information Retrieval," Natural Language and the Computer, Ed. Paul Garvin, McGraw Hill, 1963.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
RADC (EMIIO) GAFB, N.Y. 13440		Unclassified	
3. REPORT TITLE		2b. GROUP	
SOME MATHEMATICS OF INFORMATION STORAGE AND RETRIEVAL			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
In-House			
5. AUTHOR(S) (First name, middle initial, last name)			
Dr. John W. Sammon, Jr			
6. REPORT DATE		7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1968		26	12
8a. CONTRACT OR GRANT NO.		8b. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO. 5581 c. Task No. 558104 d.		RADC-TR-68-178	
9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)			
10. DISTRIBUTION STATEMENT			
This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Rome Air Development Center (EMIIO) Griffiss Air Force Base, New York 13440	
13. ABSTRACT			
This report explains some of the mathematical techniques currently being used and some which are being considered for solving a problem of information storage and retrieval. Basically two problem characterizations are discussed. The first is a statistical description and the other is a vector space characterization. Specifically, we have neglected the interesting area of linguistic analysis which is sometimes used as the basis for information retrieval. Several examples, comments and suggestions are made regarding the use of the various techniques.			

DD FORM 1473
1 NOV 66

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information Storage and Retrieval Mathematical Statistics Pattern Recognition						

UNCLASSIFIED

Security Classification