SIMULATION OF BIBLIOGRAPHIC DATA BASES FOR STUDIES OF AUTOMATIC DOCUMENT CLASSIFICATION

Gurcharan Singh Chahil

A Thesis

in

The Faculty

of

Engineering

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

November, 1978

ABSTRACT

SIMULATION OF BIBLIOGRAPHIC DATA BASES FOR STUDIES OF AUTOMATIC DOCUMENT CLASSIFICATION

Gurcharan Singh Chahil

This investigation deals with a simulation model of a document data base in order to study the performance of automatic classification procedures. The model proposed for description of the document categorizing process is dependent on prescription of a small number of global parameters. Using two statistical parameters that describe the nature of the categorization with respect to size and overlap of subject areas, the documents of the data base are divided into classification categories. Then the model is used to generate a document term matrix. This requires the assumption of Zipf's law, and a specification of statistical parameters that describe the richness of the data base with respect to content terms. At this stage the model is not subject to any other constraints regarding the associations of terms with documents. The model is then extended to incorporate the means by which the number of

content, non-content, or accidental associative terms may be varied and the effect of such variations on classification performance be studied.

The simulated data base was used to study the performance of three automatic classification procedures. The experimental results suggest the feasibility of using the simulation model for the study of automatic document classification. Furthermore, the results show that the predictive relevance rating method is superior to the attribute number method, and a high overlap among categories results in high performance of automatic classification procedures.

The effect of variations of statistical correlations of content terms with respective categories was examined. The experimental evidence indicates that classification efficiency is sensitive to changes in classification ratings of content terms in the simulated data base. This is not surprising but suggests that the parameters used in the model are appropriate for purposes of simulation.

ACKNOWLEDGEMENT

The author wishes to express his gratitude and deep appreciation to his thesis supervisor, Prof. H.S. Heaps for initiating the project and providing continued guidance throughout the investigation. The author is also thankfull to his wife for her perseverance and encouragement.

TABLE OF CONTENTS

NOMENCLATURE

I	The rank of a term.
D	Total number of different terms.
N	Total number of terms in the data base.
M_i	Word frequency for the i-th term.
M	Total number of documents in the data base.
p_{jk}	Joint probability that a document belongs to both the j-th and k-th category.
C_{mk}	Binary function whose value is 1 if the m-th document is present in the k-th category; otherwise zero.
$P_{ijkr}(I_1,I_2,I_3,I_4)$	The probability that $C_{mi}=I_1$, $C_{mj}=I_2$, $C_{mk}=I_3$ and $C_{mr}=I_4$.
P_{ijkr}	The unknown probability that a document belongs to the i,j,k,r th categories.
$M(I_1,I_2,I_3,I_4)$	The number of documents for which $C_{mi}=I_1$, $C_{m2}=I_2$, $C_{m3}=I_3$ and $C_{m4}=I_4$. The values of $I_1,I_2,\ldots$ are restricted to 0 and 1.
f_{ik}	Number of documents of the k-th category that contain the i-th term.

R_{ik}	Classification rating of the i-th term.
d_{mi}	Binary functions whose value is 1 if the m-th document contains the i-th term; otherwise zero.
$W_i(I_1, I_2, I_3, I_4)$, WI	Category weights of the i-th term.
p_c	The probability of a term being a content term.
p_n	The probability of a term being a non-content term.
$t(i), rc(i), s_i(n)$	Pseudo-random numbers uniformly distributed between 0 and 1.
$n(i)$	Pseudo-integer uniformly distributed between 1 and 4.
$V_k(m)$	Relevance rating of the m-th document to the k-th category.
a_{ki}	Constant.
U_{ii}	Proportion of documents that contain the i-th keyword.
V_{ik}	Proportion of documents that contain the i-th keyword and are in the k-th category.
TRT	Total number of documents retrieved at a given threshold.

RRT	Total number of relevant documents retrieved at a given threshold.
$P(K_j/W_i)$	Probability that a document containing term W_i belongs to category K_j.
$P(K_j)$	Probability that a document belongs to category K_j.
$P(W_n/K_j)$	Probability that a document indexed under category K_j will contain the term W_n.
P1	Probability that the first category in the ranked list of categories is correct.
P2	Probability that the first two categories in the ranked list of categories are correct.

CHAPTER 1

INTRODUCTION

In application to information retrieval from document data bases the purpose of document classification is primarily to organise and systematize the vast amounts of information to which accurate and fast access is required, and which becomes increasingly difficult as the data base grows in size. In many library data bases the documentary data are generated at a fast rate, and the growth may follow an exponential law. As a result the task of information retrieval becomes difficult. However failure to retrieve information may lead to much duplication of work. With the advent of mechanised systems much thought has been given to the organization of data by automatic means for fast and effective retrieval. But still the problem of effective retrieval has not been fully solved.

Suppose there is a store of documents and an enquirer poses a question to which the answer is to be a list of documents that satisfy his, or her, question. He can determine the set by reading all documents in the store, and by retrieving only the relevant ones. Obviously this solution is impractical although it may lead to information retrieval that is 100% effective. The enquirer will be happier if he has to read only a few documents from the entire store. His satisfaction may be regarded as a measure of system performance. In general the performance of an information retrieval system may be judged by the extent to which

it can satisfy the user population of the system. There are many parameters of the system that influence its performance. The important parameters that influence the performance may be enumerated as (1) The relevance of documents in the database, (2)The ability of the system to retrieve relevant information, (3)The ability of the system to reduce noise,or false retrievals, (4)The amount of user effort involved in getting the relevant information, (5)The type of output desired, (6)The type of user(naive,skilled,browsing), and (7)The data structure used in the system.

The main purpose of a retrieval system is to serve its user population, and its success in this regard depends on the satisfaction of its users. Therefore emphasis should be given to the design of systems that closely satisfy the needs of the users. The prime object is to retrieve all relevant information in response to a user inquiry with the involvement of minimum cost and effort on the part of user. It is virtually impossible to find a system with the property of providing 100% relevance retrieval. However,to achieve a given degree of relevance,the cost and effort may be reduced if the related documents are grouped together in some logical manner within the system. This suggests that knowledge should be diveded into groups of related phenomena in order to simplify the location and accessing of only related information. Dividing a given area of knowledge into related distinct groups for the specific purpose of aiding retrieval is,however,a major problem. The task of storing large

textual data, and the intellectual problem of characterising document content, is still quite complex. The computer may be able to read natural language, but to extract syntactic and semantic information in order to decide the relevance of documents is much more difficult.

Thus the problem of document classification reduces to automatic content analysis of the documents to be grouped. It becomes the problem of deciding what a given document is "about". The starting point of the classification may be a complete document text, an abstract, the title only, or a list of words deduced from its text. From this must be produced a document representation that a computer can handle. In production of such representations the emphasis is usually on statistical, rather than linguistic approaches, to automatic content analysis. The statistical approach is based on the original ideas of Luhn[1] which state that the frequency of a word occurrence in an article furnishes a useful measurement of word significance. His hypothesis is that frequency data can be used to extract words to represent a document. It is interesting to note that much of the subsequent work in information retrieval has been based on this basic idea. Good[2] and Fairthorne[3] were among the first to recommend that automatic classification might prove useful in document retrieval.

Automatic classification is defined to be the process of grouping together individual documents related to each other and which have been assigned to the same category numbers. They may

be said to be mutually relevant because their internal subject matter is the same. The reason for grouping documents in this way is based on the fact that otherwise the time taken by the operation of matching the search request against each document will be excessive. The document categorisation reduces the size of the data base that is relevant to the given request. It is not necassary that the categories be unique and distinct; they may overlap to some degree. Such a choice of categories should preferably be made in such a way that the relevant documents to the given search request are chosen from only a few categories.

At present,there are a number of classification methods in use in information retrieval. Spark Jones[4] has given a clear breakdown of classification methods in terms of some general characteristics of the resulting classificatory system. They are generally based on the relations between documents and classes. For example the classes may be either exclusive or overlapping, and the set of classes may be ordered or unordered. An example of an ordered classification is a hierarchy in which classes are ordered by inclusion,so that classes at one level are nested within the classes at the next level. For certain applications ordering is irrelevant whereas for others, such as document clustering,it is of importance.

The classification system used by the Association For Computing Macninery for its publication Computing Reviews is shown in figure(1.1). Prior to 1964 this classification had two levels of generality, and the terms were listed in alphabetical

order by major heading. But in the more recent system the alphabetic term listing is not used, and each subheading is accompanied by a category number. The new system could be called a hierarchical classification language although it is somewhat lacking in depth. Clearly subject headings represent a looser structure than a hierarchical language. Their use makes initial language design easier since there is less to predict, and makes future changes easier to implement because no elaborate structure need be perturbed by such changes.

An important aspect of research in automatic document classification is the development of a classification system that reduces, or eliminates,the human intellectual effort needed to classify documents. In the past,various techniques have been proposed to classify documents automatically. Most techniques employ some sort of statistical model that makes decisions only after the document representations have been read. In most of the research the document representations have been abstracts instead of complete documents, since transferring abstracts from hard copy text to machine readable form is less expensive. In general,the main methodology employed can be stated as follows:

A suitable document collection,in which the range of the subjects described by documents is not too heterogeneous, is selected. From some practical considerations, depending on the nature of collections, a suitable number of categories to fit the documents of the collection is decided upon. The documents are carefully read and indexed into the categories. The documents

are scanned for occurrence of different keywords that are arranged in decreasing order of frequency of their occurrences. Very high and low frequency words are not selected as keywords. Next, the keywords are correlated with categories, and this correlation is again a statistical one. Then the particular technique based on some mathematical model is used in order to classify the documents automatically into one or more categories. The classification effectiveness is compared with the result of manual indexing the documents. One may question the reliability of the manual indexing, and no claim can be made that the above procedure of manual indexing will result in perfect classification. But the general methodology has been to use this as criterion by which to evaluate the accuracy of automatic classification techniques.

A survey of different automatic classification procedures shows that in each such procedure the mathematical model calculates directly or indirectly the probable relevance of a document to a particular category. While indexing, a human indexer also tries to judge subjectively the relevance of the document to a particular category. But his basis of deciding relevance is very complex. It is of paramount importance to be able to quantify the relevance of a document to a particular category. Then the machine may attempt to predict this relevance by processing all the information contained in the document.

The use of probabilistic indexing for classification of documents into categories was introduced by Maron[5] and

subsequently "attribute number analysis" was developed by him [6]. Both methods depend on the prediction of the probabilities that a given document having certain keywords belongs to certain given categories. It assumes the statistical independence of keywords in documents within each category, and that subject categories are mutually exclusive and exhausitive. Neither of these assumptions are entirely valid in practice. His model does not allow for a varying degree of relevance and it asserts that either a document belongs to a given category or it does not. Application of factor analysis to classification has been described by Borko and Bernick[7] who have demonstrated factor ananlysis to be a useful technique for determination of the set of classification categories for the given population of documents.

Latent class analysis, as a means to document classification, has been applied by Baker[8]. The method depends on determination of eigenvalues of certain matrices whose elements are probabilities. If the eigenvalues are found to be real and to have values between 0 and 1 they may be used as probabilities that determine latent class structure. During the development of certain equations the statistical independence of keywords within each category is assumed. Since in practice this assumption is not true, and eigenvalues do not lie between 0 and 1, the procedure is inherently defective.

A theory of relevance for document classification has been proposed by Heaps[9] and is an attempt to quantify the notion of

relevance. The method treats the problem of automatic classification as an optimization problem where the errors between manual ratings and predicted model relevance rating to categories is minimized, and an expression is derived to give the indexing worth of additional keywords. Subsequently he has applied this procedure to automatic classification of documents in computer science[10]. The method does not require any assumptions regarding the statistical independence of keywords and categories.

As a result of the existence of many different classification techniques, it is imperative to have comparative tests of two or more systems used to index the same, or similar, set of documentary material. In this respect much of the work has suffered from the difficulty of comparing retrieval results obtained on different data bases. Researchers have carried out experiments with a large variety of document collections, and rarely has the same document collection been used in same form by more than one researcher. Sometimes one is led to suspect that the work of a researcher may be data specific and may not be valid were he to test it with other collections. Sometimes, one method may not perform to the same degree with different collections. This may give a very misleading picture of the advantages of one method relative to another. Detailed comparisons of the functioning of systems relative to each other are needed to show that one system is better suited to a particular environment, and unless many factors are taken into

account such comparisons rarely explain which system is best. To obtain a true understanding it is necessary to consider in detail the structural features of each system. Another limitation of most system evaluations has been the small size of the databases. One is left to conjecture as to the effectiveness of classification when a comparatively large collection is used.

In view of the above considerations it is tempting to consider the use of simulation techniques for the study of classification methods. This technique is more commonly used in hardware branches of computer science. Simulation may be defined as the operation of a model or simulator that is a representation of a real system [11]. In computer simulation experiments the validity of the model depends on whether the computer model is a valid model of a real data base and classification process. The extent to which it depends on our understanding of the real world and the way in which the model embodies its characteristics. Idealy the conclusions and inferences obtained from experiments on the model should be applicable to the real world.

It is important to recognize that there are many variables and subsystems within a retrieval system. In an ideal experiment it must be feasible to monitor all these and to detect any changes in the performance of the system. Because of the dimensionality and complexity of retrieval systems it is hoped that a simulation methodology may prove as useful as it has proved in some other fields of science [12,13].

However, a literature survey shows that little work has been done in regard to its application to the design and investigation of retrieval systems. A few projects where it has been used may be summarized as follows. The analysis of user behaviour in a library has been modeled by Reilly[14]. He considers the probability that a user will avail himself of library services, and the estimated service time, to constitute two independent variables in the simulation process. Blunt[15] has described the simulation approach to the analysis of alternative retrieval system design configuration, and has been concerned mainly with the measurement of system response time and equipment and personnel utilization. Fried et al[16] have explored the feasibility of an index simulation. The effect of varying the depth of indexing on retrieval was considered but no results were reported. Cooper[17] used simulation as a tool for designing and evaluating retrieval systems. The model allows examination of the effect of change of query characteristics on output performance, but the relevance of the document to the user was not simulated. He did not consider it as a good evaluative tool but discussed its merits in providing a framework for further development work.

It may be observed that in all the studies referred to above, little was done to apply the methodology to the problem of automatic classification. Most of the work was directed toward simulation of the evaluation and cost of retrieval systems in terms of equipment configuration and query chracteristics.

Before describing a simulation procedure, it is necessary to know the characteristics of the objects being simulated. The model used in this thesis incorporates certain rules which are used in the creation of a collection of pseudo-documents that form a database for the study of the automatic classification process. The individual words are the basic units for conveying content, and there is no sophisticated grammatical structure assumed. The model assumes the Zipf's law of word frequencies. In fact, it is well known that such an empirical law is satisfactory for application to many document databases that involve textual data. The advantage of creating pseudo-documents is to eliminate the differences in variability between different document collections, and thus avoid the introduction of any bias toward application of different classification techniques. This enables the true differences between different classification methods to be observed during the application of tests.

The retrieval system model is composed of three parts, (i)document-category associations generator, (ii)document-term associations generator, and (iii) different classification technique routines. The input to the first generator is a set of first and second order probabilities. The third and fourth order probabilities are calculated by the generator itself. Once these probabilities are known, the pseudo-documents are partitioned among the categories or category combinations. It may be observed that a knowledge of the particular terms is not required for the purpose. This categorised set of documents is then the

basis that is used for judging the effectiveness of other automatic classification methods. This is discussed in chapter 3.

The second generator simulates pseudo-documents and term associations. Since all the terms in a document do not convey the content of the document they cannot all be used as useful keywords. The model incorporates certain probabilities which, in fact, differentiate certain terms from the others. These probabilities have been labelled as p_c and p_n. The former is the probability that a given term is content bearing, and the latter is the probability that it is a non-content term. It is only content terms that discriminate documents of one category from those of another. The model simulates document-term associations by generating pseudo-random numbers between 0 and 1. These associations are not purely random; instead the document term association generation process includes a mechanism by which the associations of content terms with certain document categories is controlled in such a way that the content terms are constrained to occur in the documents pertinent to certain categories. This is analogous to the case of real document databases where documents in one particular category tend to contain the keywords that are associated with the category. The scheme is described in chapter 4. The input to the generator is the set of document numbers categorised into categories, p_c, p_n, the database size, and a set of category weights.

The third part of the simulation is the use of a simulated data base for a brief comparison of three different automatic classification techniques. The first two techniques are different versions of the predicted relevance method [10]. The third technique uses attribute numbers. It is observed that both predicted relevance rating and attribute number analysis [6] are stable and reliable when used on the model.

The model also computes the classification efficiency in terms of Precision (number of retrieved relevant documents divided by the total number of retrieved documents) and Recall (number of retrieved relevant documents divided by the total number of relevant documents). In the study of classification efficiency it is observed that the predictive relevance method is far superior to the attribute number method. A possible explanation is that the effect of the basic underlying assumptions of the attribute number method is difficult to assess. The detailed description of this is to be found in chapter 5.

Any simulation technique must be reliable. It should give accurate estimates of performance and allow measurement of what one desires to be measured. The simulator results should be comprehensive and accurate so that they form a basis for the design of real future systems. Furthermore, a desirable characteristic of the simulation procedure is that it be cost

effective in its ability to give results with minimum investment in time and with maximum reliability of the results.

The simulation model described in this thesis allows a number of parameters to be varied. They include:

(1) size of data base.
(2) first and second order input probabilities.
(3) the subject span of categories.
(4) the number of content terms in the collection.
(5) the category weights of content terms, or
(6) classification rating of content terms.
(7) frequecy distribution of terms in the collection.

The simulation model described in this thesis is not free from shortcomings. Although we have some detailed knowledge of information retrieval systems, yet some processes are too complex to be described adequately. The rules used to simulate documents are not complete they do not use all the available information such as, for example, the fact that different kinds of terms may convey different shades of meanings. The method described herein is dependent on a satisfactory description of certain statistical properties of document databases. The degree of reliability of these assigned probabilities may be questionable. Neverthless, it is conjectured that simulation is a useful tool for study of automatic classification procedures.

The application of simulation to automatic classification is believed to be a new approach, and further investigation is

needed in order to understand its potential as a useful tool. However, it is hoped that with increasing understanding of retrieval systems,more refinements in the model may be added with consequent reduction in its shortcomings as a tool for the study of information retrieval systems.

The simulation procedure outlined in subsequent chapters may be summarised as follows:

Step 1: Simulation of document category associations (chapter 3). Requires input data to specify the size of the data base, the number of categories, and the degree of overlap of categories.

Step 2: Simulation of document term matrix (chapter 4). Requires input data to specify the number of content terms and non-content terms and related weights WI as defined in chapter 4.

Step 3: Application of the simulation procedure in a study of three different methods of automatic document classification. Illustration of such an application is presented in chapter 5.

FIGURE 1.1: Subject headings used by the Association for Computing Machinery since 1964.

1. GENERAL TOPICS AND EDUCATION
   - 1.0 General
   - 1.1 Texts; Handbooks
   - 1.2 History; Biographies
   - 1.3 Introductory and Survey Articles
   - 1.4 Glossaries
   - 1.5 Education
   - 1.9 Miscellaneous

2. COMPUTING MILIEU
   - 2.0 General
   - 2.1 Philosophical and Social Implications
   - 2.2 Professional Aspects
   - 2.3 Legislation; Regulations
   - 2.4 Administration of Computing Centers
   - 2.9 Miscellaneous

3. APPLICATIONS
   - 3.1 Natural Sciences
     - 3.10 General
     - 3.11 Astronomy; Space
     - 3.12 Biology
     - 3.13 Chemistry
     - 3.14 Earth Sciences
     - 3.15 Mathematics; Number Theory.

FIGURE 1.1 (continued)

3.16 Meteorology

3.17 Physics; Nuclear Sciences

3.19 Miscellaneous

3.2 Engineering

3.20 General

3.21 Aeronautical; Space

3.22 Chemical

3.23 Civil

3.24 Electrical; Electronic

3.25 Engineering Science

3.26 Mechanical

3.29 Miscellaneous

3.3 Social and Behavioral Sciences

3.30 General

3.31 Economics

3.32 Education; Welfare

3.33 Law

3.34 Medicine; Health

3.35 Political Science

3.36 Psychology; Anthropology

3.37 Sociology

3.39 Miscellaneous

3.4 Humanities

3.40 General

3.41 Art

FIGURE 1.1 (continued)

3.42 Language Translation and Linguistics

3.43 Literature

3.44 Music

3.49 Miscellaneous

3.5 Management Data Processing

3.50 General

3.51 Education; Research

3.52 Financial

3.53 Government

3.54 Manufacturing; Distribution

3.55 Marketing; Merchandising

3.56 Military

3.57 Transportation; Communication

3.59 Miscellaneous

3.6 Artificial Intelligence

3.60 General

3.61 Induction and Hypothesis-formation

3.62 Learning and Adaptive Systems

3.63 Pattern Recognition

3.64 Problem-solving

3.65 Simulation of Natural Systems

3.66 Theory of Heuristic Methods

3.69 Miscellaneous

FIGURE 1.1 (continued)

3.7 Information Retrieval

3.70 General

3.71 Content Analysis

3.72 Evaluation of Systems

3.73 File Maintenance

3.74 Searching

3.75 Vocabulary

3.79 Miscellaneous

3.8 Real Time Systems

3.80 General

3.81 Communications

3.82 Industrial Process Control

3.83 Telemetry; Missiles; Space

3.89 Miscellaneous

3.9 Miscellaneous

4. PROGRAMMING

4.0 General

4.1 Processors

4.2 Programming Languages

4.3 Supervisory Systems

4.4 Utility Programs

4.9 Miscellaneous

FIGURE 1.1 (continued)

5. MATHEMATICS OF COMPUTATION
   - 5.0 General
   - 5.1 Numerical Analysis
   - 5.2 Metatheory
   - 5.3 Combinatorial and Discrete Mathematics
   - 5.4 Mathematical Programming
   - 5.5 Mathematical Statistics; Probability
   - 5.6 Information Theory
   - 5.9 Miscellaneous

6. DESIGN AND CONSTRUCTIONS
   - 6.0 General
   - 6.1 Logical Design; Switching Theory
   - 6.2 Computer Systems
   - 6.3 Components and Circuits
   - 6.9 Miscellaneous

7. ANALOG COMPUTERS
   - 7.0 General
   - 7.1 Applications
   - 7.2 Design; Construction
   - 7.3 Hybrid Systems
   - 7.4 Programming; Techniques
   - 7.9 Miscellaneous

8. FUNCTIONS

CHAPTER 2

BIBLIOGRAPHIC CHARACTERISTICS OF DOCUMENT DATA BASES

2.1 Zipf's Law

If a study of natural language,such as English ,is made it is common experience to find that a small number of words account for the major portion of the text. Zipf[18] was one of the first to detect this property of text,and subsequent studies of word distributions by other linguistics [19,20] led to emergence of Zipf's law concerning word frequencies. For illustration of this law,consider a sufficiently large sample of general English text. For each different word of the text,the word occurrences may be counted and the words arranged in decreasing order of their occurrence. The position of each word in the ordered list is called its rank,and the number of occurrences is its frequency. The Zipf's law states that if the frequency of a word is multiplied by its rank in the list,the product is constant for all the different words of the text. Thus

$$\text{rank} * \text{frequency} = \text{constant}. \qquad (1)$$

The Zipf relation can also be expressed in terms of the probability of occurrence pr of the r th word by the relation

$$pr = A/r \qquad (2)$$

where A is constant whose value will be different for each type of text.

The above relations are only approximately satisfied by large texts. The Zipf's law is based on empirical evidence and,in its original and simplest form, is an approximate description of an aspect of human behaviour in the generation of language.

The vocabulary in a natural language does not remain static and must be updated to conform to changes in the manner in which the documents in the subject field are being written. People sometimes measure the vocabulary of a writer by the total number of different words in his works. However,rare and unusual words make up a small fraction of written English. The environments in which a data base is generated influences the size and quality of the vocabulary. Perhaps all languages tend to develop a basic size of vocabulary which is governed by the capabilities and quality of the human brain. To this basic language the more literate people can add as many special and infrequently used words as they desire.

As new words are added to the vocabulary,the Zipf's law still continues to be satisfied. Zipf has expressed the law by means of a graph (Fig.2.1) which is plotted with respect to logarithmic scales. Curves A and B are for 260,430 running words of James Joyce's Ulysses and 43,989 running words from a newspaper,respectively[18]. The straight line C is Zipf's idealized graph. The heights of A and B are determined by the number of words in the sample. The steps at the lower right of the curves show that infrequent words can occur once,twice,thrice,and so forth,but cannot have fractional

occurences such as 1.5 or 2.67. The idealized curve with a slope of 45 degrees indicates that the number of different words in the sample must equal the number of occurrences of the most frequently used word.

The word frequency distribution and the values of r*pr are presented in Table (2.1) for the Chemical Abstract Titles tapes issued by the Chemical Abstract Services during 1965[33]. In each case N represents the total number of words present in the sample considered.

In practice, Zipf's law is not satisfied perfectly. It has been observed from word occurrences in texts that word probabilities cannot be inversely proportional to the rank of the word[21] for all words. The hyperbolic relation (r*f = c) makes no provision for the integrality of r, and hence is inadequate as an exact description of what is observed. However in the study of document data bases, word frequencies are found to be in approximate agreement with Zipf's law.

Zipf's law can be modified as follows in order to give the number of words which occur once, twice, or n times. Let

pr_n = probability of occurrence of word of rank r_n

r_n = rank of a word which occurs n times

N = total word occurrences

D = total number of different words

The rank r_n is given by the Zipf's law

Then

$$pr_n = A/r_n$$

$$n/N = A/r_n$$

$$r_n = AN/n \qquad (3)$$

In Figure 2.1, the horizontal steps at the right-hand side of the graph indicate that there are some words that occur the same number of times. For these words the ranking within the group of words that have the same frequency is arbitrary. Suppose it is assumed that the rank given by $r_n = AN/n$ applies to the last of the words that occur n times. Then there are r_n words that occur more than n times, and there are r_{n+1} words that occur more than n+1 times. The number of words that occur exactly n times is therefore

$$I_n = r_n - r_{n+1} = A*N/n - A*N/n+1 = A*N/[n(n+1)] \qquad (4)$$

The highest ranking term has rank D, and if there is at least one term that occurs once, then

$$D = A*N/1 \qquad (5)$$

Substituting AN from (5) into (4), gives

$$I_n/D = 1/[n(n+1)] \qquad (6)$$

$$I_n/I_1 = 2/[n(n+1)] \qquad (7)$$

The derivation of equations (6) and (7) is according to Booth[22] who verified them by examination of four English texts with regard to occurrence of words of low frequencies [22]. As indicated by Booth, one of the consequences of the above law is that it indicates that 50% of the different terms occur only once in the entire data base, 16% occur only twice, and 8% occur only three times. This relation can be used to predict the number of index terms of given low frequency occurrence that may be used to

index the documents of a data base.

In the present attempt at simulation of a document data base the words are designated by the integral numbers that denote their ranks, and their occurrences in the pseudo-documents are generated by means of pseudo-random numbers. The Zipf's law has been embodied in the simulation model as follows:

Let

M_I be the word frequency for the I th term

N be the total number of terms in the data base

D be the total number of different terms

I be the rank of the term

Using equation (3)

$$M_i = A*N/I \qquad (8)$$

where A is a constant that is different for each text sample. The constant A can be found by using the fact that the sum of the different word frequencies must be equal to the number of word occurrences. Hence,

$$A*N(1+1/2+1/3+...+1/D) = N \qquad (9)$$

The sum is a harmonic series which may be approximated by

$$A = 1/(\mathrm{Log}(d)+ \gamma) \qquad (10)$$

where γ is Euler's constant whose value is 0.5772

The details of the simulation procedure are described in chapters 4 and 5. The simulated word frequencies are shown plotted against their rank in Figure 2.2. The dotted line indicates the theoretical variation of Log(f) and Log(I) (I being

the rank of the term) as depicted by equation (1). The solid line is the experimental variation of position of occurrence of a particular word in the simulated data base and its word tokens (frequency). The close agreement between the two indicates how well the experimental graph follows the theoretical one.

2.2 VOCABULARY GROWTH

In any retrieval system an important component is the inverted index file. The inverted file provides a mechanism by which document descriptions can be compared with the descriptions of the requests for information. At the time of input of documents to the system the document is assigned some terms or classes which depend on the subject matter of the document. Efficient indexing is not necessarily achieved by labelling a document on the basis of its intrinsic subject matter. Rather, it is achieved by labelling a document according to the type of users who may be expected to derive some benefit from it, and according to the type of requests for which the document is likely to be regarded as relevant. Thus subject indexing must reflect the characteristics and requirements of users of the document collection. Thus the same document may be indexed correctly in six different ways in six different organizations. Normally documents belong to the same general subject area, but each document has slightly different subject matter which is helpful in allowing it to be differentiated from other documents that belong to the same broad area.

The relationship between documents and index terms may be represented in the form of the matrix of Figure 2.3. In this diagram the letters A through J represent index terms,and the numbers 1 through 10 represent documents indexed in the retrieval system. Each newly arrived document,on the basis of its subject matter,is assigned to some classes, and this assignment is indicated by a cross in a corresponding cell of the matrix. The classes may be designated by sets of index terms, and the complete set of these index terms is called the indexing language. In response to an information need,the subject request is translated into the indexing language. The search operation consists of matching the search profile against the document profiles in order to determine which documents correspond to the search profile.

Two major factors which affect the performance of an information retrieval system are (a) the exhaustivity,and (b) the specificity,of the index language. For any document its indexing exhaustivity is defined as the extent to which all the subjects discussed in the document are recognized by the indexing operation. A high level of exhaustivity is obtained by recognizing all the topics of a document during the conceptual analysis stage of indexing, and by expressing these topics by means of appropriate combinations of index terms. It has been established that a high level of exhaustivity of indexing leads to a high recall and low precision[23]. Conversely a low level of exhaustivity leads to a low recall and high precision. There

are basically two reasons for this. First, all the indexable topics are included for every document. It is likely that some of the topics will relate to the document only in a very minor way. As a result, many documents of low relevance will tend to be retrieved in response to the requests. The second reason is that the more topics that are recognized in indexing, and the more index terms that are used to express topics, the greater are the chances of fall-out (that is, irrelevant documents) in searching. It should be recognized that the level of exhaustivity applied in indexing is determined as a result of a policy decision by the managers of the system. It does not depend on the properties of the index language since the index language used is normally rich enough to handle the subject fields treated in the documents of a collection.

The index language specificity describes the ability of the index language to describe topics precisely. The greater the specificity of the index language, the more precisely can it define the subject matter and allow smaller document classes to be created. As a consequence a high precision will be achieved during searching. A high level of specificity in an index language will ensure high precision, and a low level will result in high recall values.

From considerations of their effect on system effectiveness it is of value to be able to quantify the terms exhaustivity and specificity. A few people[24,25] have attempted to relate them to document collection statistics. For example exhaustivity can

be assumed to be related to the number of index terms assigned to a given document,and specificity related to the number of documents to which a given term is assigned. From the above discussion it appears that there must be an optimum level of exhaustivity and specificity for a given user population.

The retrieval system vocabulary may be used to index and retrieve the documents in two ways. First an index term may be created to identify a class uniquely. Such an index term may be a logical combination of two simple terms. For example, the index term "Aluminium Machining" establfshes a relationship between two simple terms "Aluminium" and "Machining". This expresses the co-ordination of the class "Aluminium" and the class "Machining". The documents to which this class or concept number is applied are common to both the classes "Aluminium" and "Machining". Such a vocabulary, in which class relationships are expressed once and for all by labels used to define classes in the indexing operation,is called a Pre-co-ordinate Vocabulary. In this context,the term "Aluminium Machining" is the pre-co-ordination of the terms "Aluminium" and "Machining".

On the other hand,there are systems which do not use Pre-co-ordination. In such a system, at the time of indexing,only basic terms or class concepts are used for indexing. Following the above example,the topic "Aluminium Machining" is indicated by assigning to the document the separate terms "Aluminium" and "Machining". In conducting the search, the user can use Boolean functions of the individual terms in order to express his

interest in a complex topic. This type of vocabulary, in which class relations are exploited by manipulating the individual classes at the time of searching, is called a Post-co-ordinate Vocabulary.

The above method of establishing the relation between individual terms, at the time of vocabulary growth, is said to constitute pre-co-ordinate indexing. Similarly the other method, where the index terms at the time of vocabulary growth are uniterms and the user can manipulate these uniterms at the time of search, is known as post-co-ordinate indexing. Normally in post-co-ordinate indexing, during vocabulary growth, the uniterms[29] are not allowed to be merged together to create another class. But in some cases complying with this condition may give false results. For example, in searching for documents on "Ultrasonic welding", the terms "ultrasonic" and "welding" are coordinated. Documents retrieved on this class intersection may or may not deal with the subject of the search. A retrieved document indexed under the terms "electrical", "component", "welding", "cleaning", and "ultrasonic" may deal with welding of electrical components and subsequent cleaning by means of ultrasonics. To avoid this type of unwanted retrieval some further pre-co-ordinated headings may be adopted. Rather than being unit terms the index terms may become labels for unit concepts[27].

An increasing use of unit concepts will reduce the false co-ordination but cannot eliminate it entirely, especially in

manipulative systems which allow term coordination during question formulation. To some extent,false co-ordination can be eliminated by grouping together,at indexing time,all terms used to describe a complex subject,and separating these terms from others used to describe different subject topics. In spite of all these provisions,the cross-talk in manipulative indexes is difficult to resolve. For example,if it is desired to retrieve documents on design of digital computers,and "digital computer" and "design" are used as search terms, a number of irrelevant documents (such as digital computer applied to aircraft design) may be retrieved. These types of links cannot resolve the problem because the terms, being used in the description of the same subject complex,would appear in the links. To some extent this problem can be solved by the provision of syntax (rules governing positional relationships of words)[30] in the index language.

In the case of pre-co-ordinate indexing there is no facility for manipulating the classes. Instead,class relationships are built into the language implicitly. A pre-co-ordinate vocabulary is larger than that of a uniterm vocabulary. This is due to the fact that, as new classes are added to a pre-co-ordinate vocabulary,the vocabulary size is increased.

It is important to accept the fact that there is not much difference between pre-co-ordinate and post-co-ordinate systems. Some of the manipulative systems can be of mixed breed which manifest some amount of pre-co-ordination. The subject headings

of the ASTIA system [23] are pre-co-ordinate descriptors which can be manipulated by a post-co-ordinate system. The Medical Literature Analysis and Retrieval System (MEDLARS) of the National Library of Medicine is an example of a large mechanized manipulative system.

One important distinction between types of vocabulary is that they may be controlled or uncontrolled. The term controlled vocabulary refers to a list of approved index terms that an indexer is constrained to use for indexing purposes. The control on the language may also include specification of hierarchical relationships between index terms. Some other syntactic controls depending on the application can also be exercized on the language.

The other type of vocabulary, where the indexer need not establish and maintain an approved list of terms, is called an uncontrolled vocabulary. Lack of vocabulary control poses some problems. For example, different indexers may use different terms to describe an identical concept. This creates problems for the searcher who has to think of alternative ways of describing a particular subject in order to conduct a satisfactory search. This gives rise to the practice of choosing one of several synonymous terms and of introducing "see" references for the others.

Among the main components responsible for reduction of precision and recall in an information retrieval system, the

vocabulary, or index language, seems to be the major contributing factor. The inadequacies of the index language cannot be compensated by the provision of efficient indexing and searching. The index language failures are attributed to (a) lack of specificity in the terms and (b) ambiguous relationships between terms. The lack of specificity will cause precision failures but not recall failures provided appropriate references are incorpated to encompas the given concept in the vocabulary. If no terms relating to a specific question term are provided in the vocabulary then the indexer is likely to omit the topic, and indexing inconsistencies will occur. A search will then result in both recall and precision failures. Another factor responsible for precision failure is defective hierarchical structuring of the vocabulary.

Term co-ordination is found to be useful for increasing the specificity of the vocabulary. Term co-ordination reduces the size of document classes and increases the size of the index term vocabulary. A request to search for term A only when it occurs with term B asks for the class AB,which is likely to be substantially smaller than the class A+B. The classes may be intersected at the time of indexing(Pre-coordination) or at the time of searching(Post-coordination). It appears that class coordination is a successful means of improving precision. However it tends to reduce the recall level.

It follows that the vocabulary of an index language greatly influences the recall and precision performance of the system.

With a large size of vocabulary, more document classes can be defined uniquely and the size of the classes may be reduced. This amounts to saying that the size of an index term vocabulary is a reflection of the specificity of the index language.

The specificity of the index language is the main controlling factor for the precision capabilities of the system. Consider two retrieval systems using controlled vocabularies,system A with 2000, and system B with 1000 index terms. Suppose the same collection of documents on Aerodynamics have been indexed by each system. Suppose a search is made for documents on "slender delta wings". At the time of indexing in system A,having 2000 terms, the subject can be uniquely defined by class "slender delta wings". For system B,having a smaller vocabulary,the concept is not defined precisely and it is indexed under the more general class "delta wings". Naturally,a request for the documents will give more precision for the system A.

It has been observed that many document data bases exhibit similar characteristics in regard to vocabulary growth. As new documents are added to the database the vocabulary of terms increases. Since most of the terms in the new documents added are repetitions of terms already in the vocabulary of the document collection it is to be expected that the rate of growth of vocabulary is very small. An empirical relation connecting vocabulary growth and text length for general English text of up to 20,000 terms has been suggested [28] as follows. If

D = total number of different terms in vocabulary

N = total number of words of the text

K = constant depending upon the database

B = constant depending upon database

then

$$D = K*N**B \qquad (11)$$

$$LOG(D) = B*LOG(N)+LOG(K),$$

The above equation indicates that LOG(D) is a linear function of LOG(N). The variation of vocabulary size as a function of database size is presented in Fig.(2.4) for title words of the Chemical Titles tapes and both title words and subject headings of the Marc tapes. The empirical law has been found to hold approximately for the vocabularies in different fields within many document databases. This is a highly useful relation for predicting vocabulary characteristics of the databases as they increase in size.

Table 2.1: Values of rp_r for title terms in four different data bases.

	Kucera and Francis N=1,000,000			Chem. Titles 1965 N=1,058,359			MARC 01-58 N=317,581			Gas Chromatrography N=220,000 (approx)		
Rank	Term	Freq	rp_r	Term	Freq	rp_r	Term	Freq	rp_r	Term	Freq	rp_r
1	THE	69,971	.070	OF	107,687	.102	THE	25,647	.081	OF	23,569	.107
2	OF	36,411	.073	AND	37,578	.071	OF	21,471	.135	GAS	11,091	.101
3	AND	28,852	.086	THE	36,318	.103	AND	12,975	.123	THE	9,560	.130
4	TO	26,149	.104	IN	32,868	.124	IN	8,987	.113	AND	8,141	.148
5	A	23,237	.116	ON	10,984	.052	A	5,149	.081	CHROMATOGRAPHY	7,776	.177
6	IN	21,341	.128	BY	10,727	.070	TO	3,741	.071	IN	7,086	.193
7	THAT	10,595	.074	A	10,252	.068	FOR	3,464	.076	BY	4,031	.128
8	IS	10,099	.081	DI	8,419	.064	ON	2,726	.069	CHROMATOGRAPHIC	3,566	.130
9	WAS	9,816	.088	WITH	7,964	.068	HISTORY	1,213	.034	FOR	3,167	.130
10	HE	9,543	.095	FOR	6,509	.062	NEW	1,203	.038	ANALYSIS	3,061	.139
20	I	5,173	.103	METHYL	4,030	.076	LAW	649	.041	FROM	1,223	.111
30	THEY	3,618	.108	ACIDS	2,697	.076	AMERICA	511	.048	COMPOSITION	702	.096
40	THEIR	2,670	.107	5	2,236	.085	INTRODUCTION	439	.055	APPLICATION	576	.105
50	IF	2,199	.110	AN	1,890	.098	HIS	371	.058	OILS	492	.112
100	WELL	897	.090	PER	1,210	.114	BUSINESS	302	.095	MILK	256	.116
200	ALMOST	432	.086	SULFIDE	736	.139	INFORMATION	162	.102	AMINES	136	.124
300	HELP	311	.093	8	503	.143	CONSTRUCTION	118	.111	SPECTROSCOPY	89	.121
400	TURN	233	.093	METALLIC	395	.149	MODEL	93	.117	PEAKS	67	.122
500	STARTED	194	.097	FLUORESCENCE	316	.149	CHARACTERISTICS	78	.123	PRELIMINARY	53	.120
1,000	REACH	106	.106	TOXIN	147	.139	IMPLICATIONS	45	.142	STERIC	24	.109
2,000	SOLDIERS	56	.112	OXYTOCIN	67	.127	DIPLOMACY	20	.126	JUICES	9	.082
3,000	SURVEY	37	.111	DECREASE	36	.102	PSYCHIC	13	.123	DILUENTS	5	.068
4,000	SOUTHERERS	26	.108	GERMINATING	20	.076	FRICTION	9	.113	ANCHIMERICALLY	3	.055
5,000	ATTRACT	19	.095	RESONATOR	14	.066	PEASANT	7	.110	WORLD	3	.068

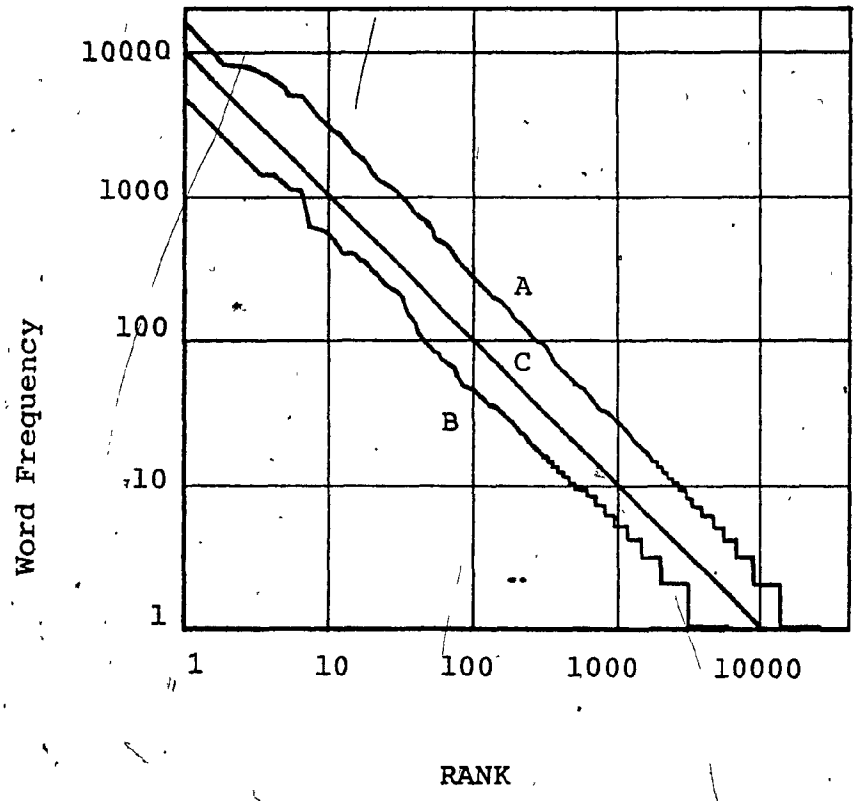


Fig. 2.1 Frequency versus Rank for different textual data

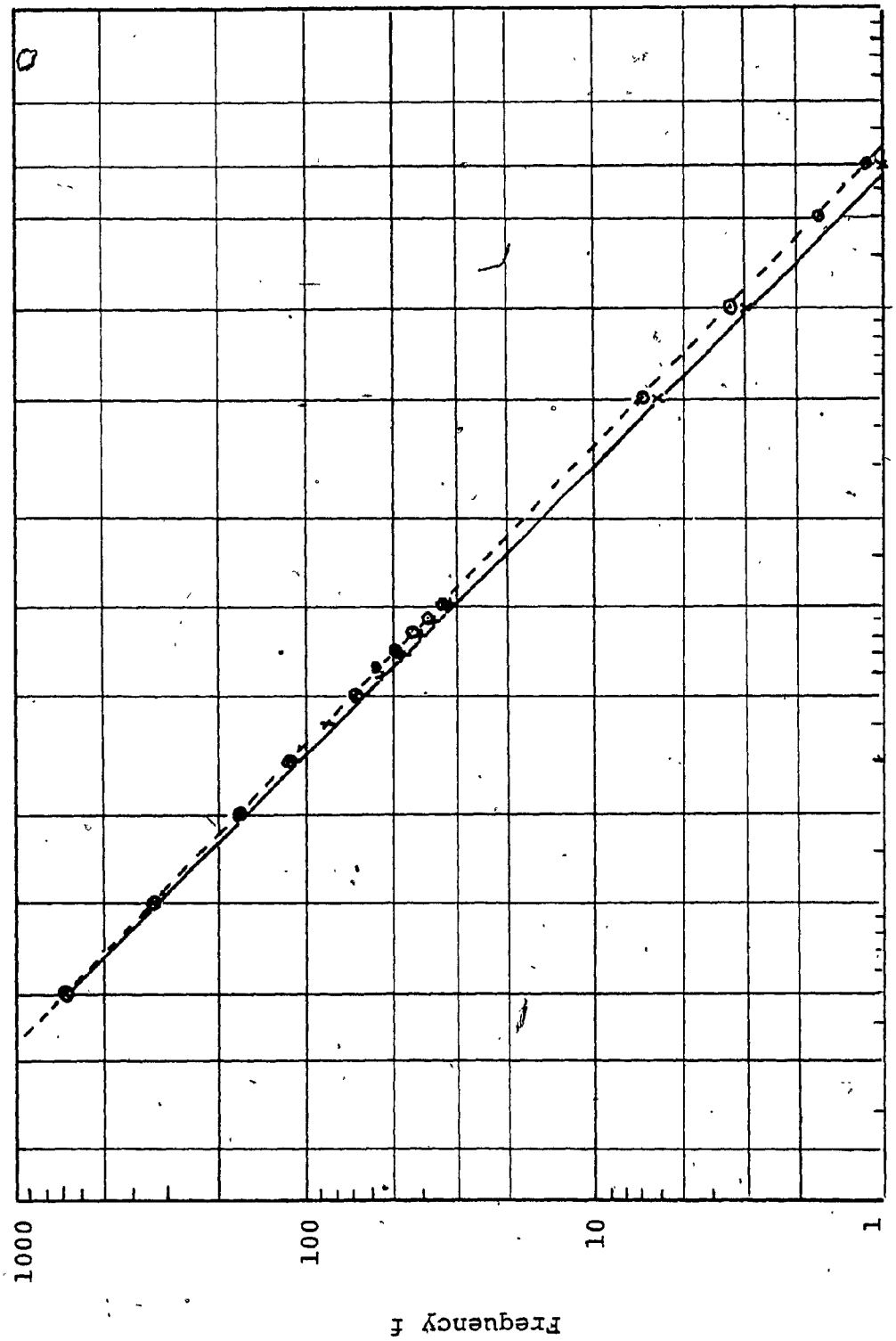


Fig. 2.2 Simulated and theoretical word frequency distribution

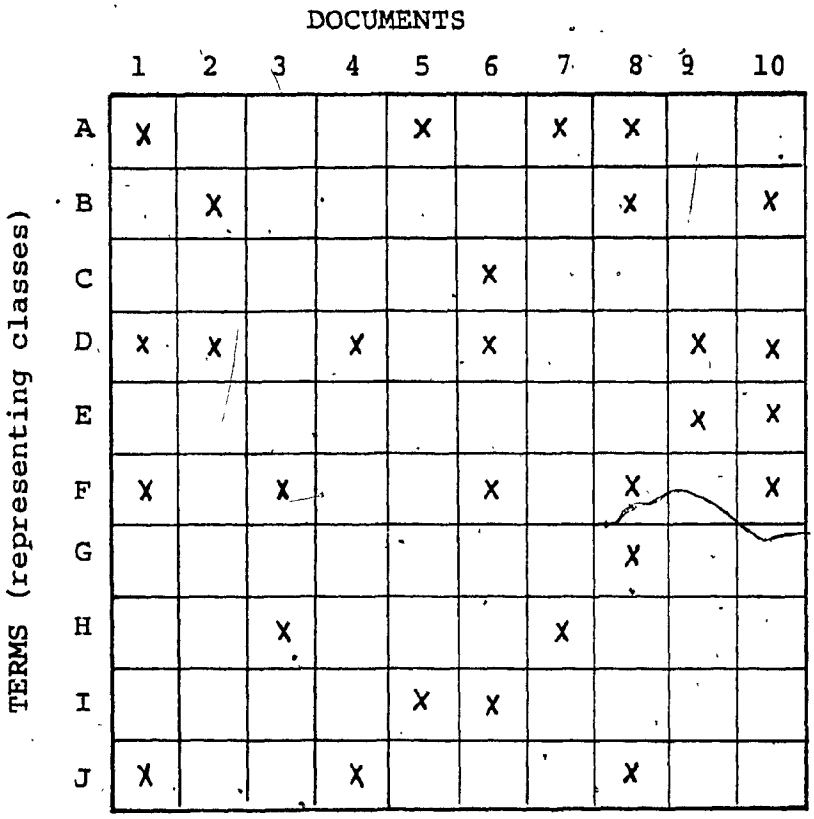


Fig. 2.3 Relationship between documents and index terms

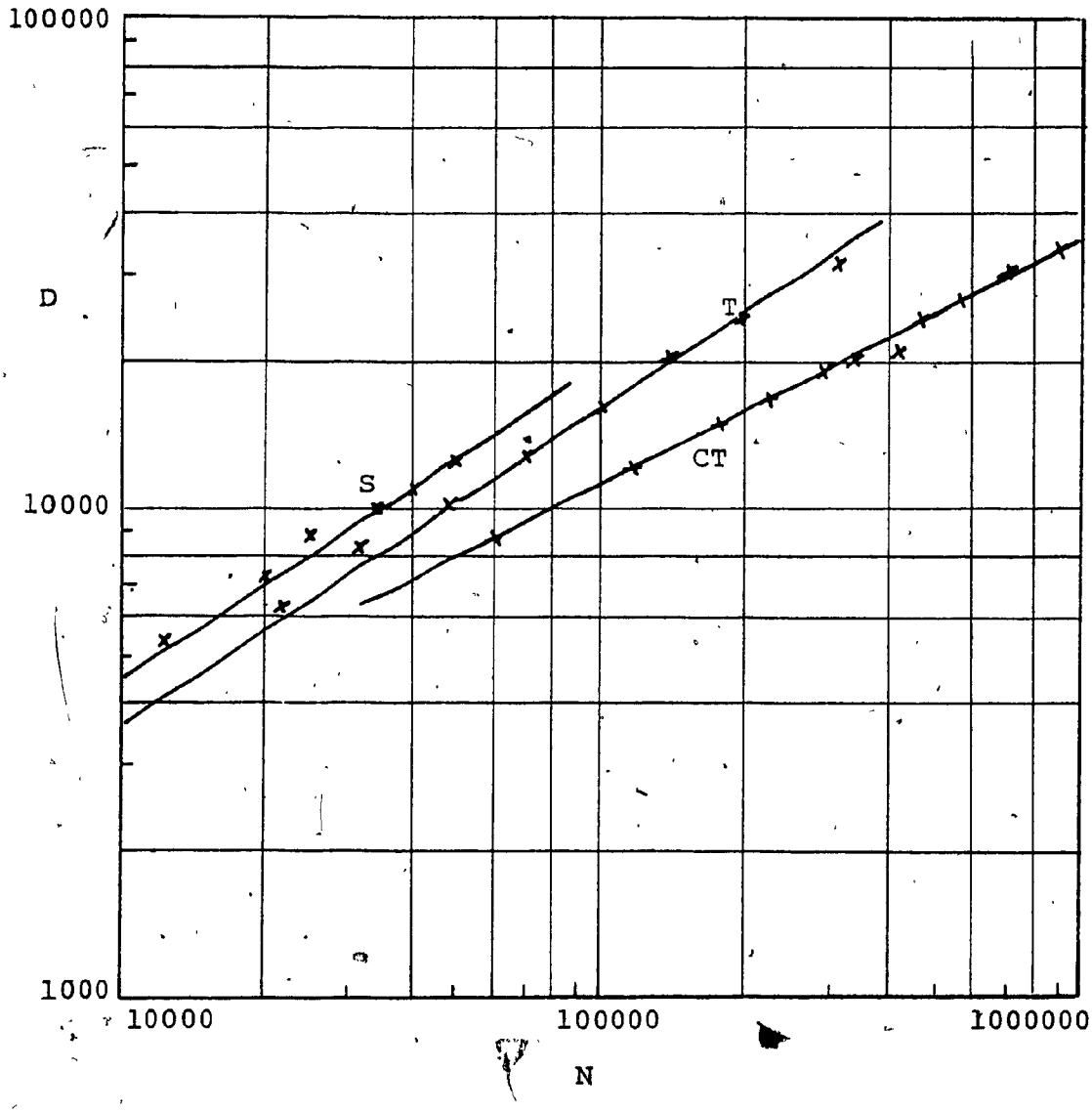


Fig. 2.4 Logarithmic relation between D and N of Chemical Titles (CT), and for MARC Titles (T) and subject headings (S).

CHAPTER 3

DOCUMENT CATEGORY MATRIX

Simulation is basically a technique that involves setting up a model of a real situation and then performing experiments on the model. The purpose of simulation is either to gain an understanding of the behaviour of the system or to evaluate the effect of various strategies being considered for the operation or control of the system. The model is often amenable to manipulations that would be impossible, too expensive, or impractical to perform on the real system that it represents. It is important to ensure that the simulation model accurately portrays the real system so that the operation of the model can be studied, and from this study the behaviour of the real system can be predicted.

It may be emphasised that simulation techniques are often based on use of probabilistic data and are therefore suitable for application to problems that cannot be solved easily by deterministic methods. Such problems are found in information retrieval where there are numerous variables and numerous sub systems within an information retrieval system. An ideal measurement technique should allow the monitoring of sub systems and variables in order to predict the behaviour of the system. Because of the complexity of an information retrieval system, it appears that simulation methodology might be suited for the

purpose.

The aim of this chapter is to describe simulation of the document and category matrix based on prior knowledge of certain probabilities. A knowledge of the particular descriptors (index terms) is not required for the simulation. Given some probabilities, the object is to illustrate how the documents can be allotted to different document categories and category combinations.

Consider a data base of M document records that contains a total of N terms of which only D terms are distinct. Let the terms be ranked with respect to decreasing frequency of occurrence, and suppose that the term frequencies are distributed according to the Zipf's law. Let M_i denote the number of occurrences of the I th term.

Assume that the documents are assigned to k categories and that the following probabilities are known.

$p_k = p_{kk}$ = probability that a document belongs to the k th category.

p_{jk} = joint probability that a document belongs to both the j th and k th category.

Thus the expected number of documents in the k th category is $p_k * M$.

Let $p_k(j)$ be the probability that a document in the j th category is also in the k th category. This denotes the conditional probability that given a document in the j th

category it is also present in the k th category. Then

$$p_k(j) = p_{jk}/p_j$$

$$p_{jk} = p_j p_k(j) = p_k p_j(k)$$

The association of documents with categories may be represented by a document category matrix of elements C_{mk} where

$C_{mk} = 1$ if m th document is in the k th category

$= 0$ otherwise

3.1 Probability Distribution of C_{mi} For Fixed m:

Let $P_{ij}(I_1,I_2)$ denote the probability that $C_{mi} = I_1$ and $C_{mj} = I_2$ where the values of I_1 and I_2 are restricted to 0 and 1. Then

$$P_{ij}(1,1) = p_{ij}$$

$$P_{ij}(0,1) = p_j - p_{ij}$$

$$P_{ij}(0,0) = 1 - p_i - p_j + p_{ij}$$

Since probabilities must be non-negative, it follows that

$$p_{ij} \leq p_j \text{ or } p_i$$

$$p_{ij} \geq p_i + p_j - 1$$

The further condition that the probabilities be not greater than 1 is automatically satisfied if the p_i and p_j are in the range 0 to 1.

Let $P_{ijk}(I_i,I_2,I_3)$ denote the probability that $C_{mi} = I_1$, $C_{mj} = I_2$, $C_{mk} = I_3$. If p_{ijk} denotes the unknown probability that a document belongs to both the I th, j th, and k th categories then

$$P_{ijk}(1,1,1) = p_{ijk}$$

$$P_{ijk}(1,0,1) = p_{ik} - P_{ijk}(1,1,1)$$

$$= p_{ik} - p_{ijk}$$

$$P_{ijk}(0,1,1) = p_{jk}-p_{ijk}$$

$$P_{ijk}(0,0,1) = p_k-P_{ijk}(0,1,1)-P_{ijk}(1,0,1)-P_{ijk}(1,1,1)$$

$$= p_k-p_{jk}-p_{ik}+p_{ijk}$$

$$P_{ijk}(0,0,0) = 1-p_i-P_{ij}(0,1)-P_{ijk}(0,0,1)$$

$$=1-p_i-p_j+p_{ij}-p_k+p_{jk}+p_{ik}-p_{ijk}$$

Since all the probabilities $P_{ijk}(I_1,I_2,I_3)$ must be non-negative it follows that

$$p_{ijk} \leq p_{ik}$$

$$p_{ijk} \leq 1-p_i-p_j-p_k+p_{ij}+p_{ik}+p_{jk}=c$$

And

$$p_{ijk} \geq p_{ik}+p_{jk}-p_k = q_k(i,j)$$

The condition for all probabilities to be less than or equal to 1 is automatically satisfied since for example the condition

$$1-p_i-p_k-p_j+p_{ij}+p_{ik}+p_j-p_{ijk} \leq 1$$

is equivalent to the condition

$$p_{ijk} \geq p_{ij}+p_{ik}+p_{jk}-p_i-p_j-p_k$$

Therefore the condition is satisfied if p_{ijk} is greater or equal to zero.

Thus if L_{ijk} and M_{ijk} denote

$$L_{ijk} = \text{Max}[q_i(j,k),q_j(k,i),q_k(i,j)]$$

$$M_{ijk} = \text{Min}[p_{ij},p_{ik},p_{jk},c]$$

then p_{ijk} must satisfy the inequality

$$L_{ijk} \leq p_{ijk} \leq M_{ijk}$$

3.2 Case Of Four Categories:

Let $P_{ijkr}(I_1,I_2,I_3,I_4)$ denote the probability that $C_{mi} = I_1, C_{mj} = I_2, C_{mk} = I_3$ and $C_{mr} = I_4$. If p_{ijkr} denotes the unknown probability that a document belongs to all of the i,j,k,r th categories then

$$P_{ijkr}(1,1,1,1) = p_{ijkr}$$

$$P_{ijkr}(1,1,0,1) = p_{ijr} - p_{ijkr}$$

$$P_{ijkr}(1,0,1,1) = p_{ikr} - p_{ijkr}$$

$$P_{ijkr}(1,0,0,1) = p_{ir} - P_{ijkr}(1,1,0,1) - P_{ijkr}(1,0,1,1) - P_{ijkr}(1,1,1,1)$$

$$= p_{ir} - p_{ijr} - p_{ikr} + p_{ijkr}$$

$$P_{ijkr}(0,0,0,1) = p_r - P_{ijkr}(1,0,0,1) - P_{ijkr}(0,1,0,1) - P_{ijkr}(0,0,1,1) - P_{ijkr}(1,1,0,1) - P_{ijkr}(1,0,1,1) - P_{ijkr}(0,1,1,1) - P_{ijkr}(1,1,1,1)$$

$$P_{ijkr}(0,0,0,1) = p_r - p_{ir} - p_{jr} - p_{kr} + p_{ijr} + p_{ikr} + p_{jkr} - p_{ijkr}$$

$$P_{ijkr}(0,0,0,0) = P_{ijk}(0,0,0) - P_{ijkr}(0,0,0,1)$$

$$= 1 - p_i - p_j - p_k + p_{ij} + p_{jk} + p_{jk} - p_r + p_{ir} + p_{jr} + p_{kr} - p_{ijr} - p_{ikr} - p_{jkr} + p_{ijkr}$$

Since all the probabilities must be non-negative it follows that

$$p_{ijkr} \leq p_{ijr}$$

$$p_{ijkr} \geq p_{ijr} + p_{ikr} - p_{ir}$$

$$= q_{ir}(j,k)$$

$$p_{ijkr} \leq p_{ijr} + p_{ikr} + p_{jkr} - p_{jr} - p_{kr} + p_r$$

$$= q_f(i,j,k)$$

$$p_{ijkr} \geq p_i+p_j+p_k+p_r-1 -p_{ij}-p_{ik}-p_{ir}$$
$$-p_{jk}-p_{jr}-p_{kr}+p_{ijk}+p_{ikr}+p_{jkr}$$
$$=c$$

Thus if L_{ijkr} and M_{ijkr} denote

$$L_{ijkr} = \text{Max}\{ q_{ir}(j,k), c\}$$

$$M_{ijkr} = \text{Min}\{p_{ijk}, p_{ijr}, p_{jkr}, q_i(j,k,r), q_j(k,r,i), q_k(r,i,j), q_r(i,j,k)\}$$

then p_{ijkr} must satisfy the inquality

$$L_{ijkr} \leq p_{ijkr} \leq M_{ijkr}$$

It may be noted that in the above expressions for $P_{ijkr}(I_1,I_2,I_3,1)$, if the p_r, p_{ir}, p_{ijr} and p_{ijkr} are replaced by $1, p_i, p_{ij}$, and p_{ijk} respectively, the new expressions denote $P_{ijk}(I_1,I_2,I_3)$ Similarly in the expressions $P_{ijk}(I_1,I_2,1)$, if p_k is replaced by 1, the new expressions denote $P_{ij}(I_1,I_2)$. Also if

$$C_{m1} = I_1, C_{m2} = I_2, \ldots\ldots, C_{mr} = I_r$$

the probability that $C_{mr+1} = 1$ is

$$P_{12\ldots,r+1}(I_1,I_2\ldots,I_r,1) / P_{12\ldots,r}(I_1,I_2,\ldots,I_r)$$

For a given value of the p_k, the only constraint on the $p_{ik}, p_{ijk}, p_{ijkr}$ is that they must be chosen within the ranges L_{ij} to M_{ij}, L_{ijk} to M_{ijk} and L_{ijkr} to M_{ijkr}. In the absence of any further information, it is assumed that each occurs with uniform probability distribution within the appropriate range. To assume any other probability distribution would be equivalent to assuming less uncertainty and hence that further information is available regarding constraints on the distribution.

3.3 Simulation of C_{mi} in terms of Proportions p_k and p_{jk}

Suppose that, instead of being given probabilities p_k and p_{jk}, the following quantities are given.

$p_k = p_{kk}$ = proportion of documents that belong to the k th category.

p_{jk} = proportion of documents that belong to both the j th and k th category.

The number of documents in the k th category is $p_k * M$, and the number of documents that belong to both the j th and k th category is $p_{jk} * M$.

let M(1) denote the number of documents for which $C_{m1} = 1$ and M(0) denote the number for which $C_{m1} = 0$. It means that M(0) documents are not in category 1. Then

$$M(1) = p_1 * M$$

$$M(0) = (1-p_1)M$$

Thus any choice of p_1 in the range $0 \leq p_1 \leq 1$ leads to a possible categorisation of documents with respect to the first category. Similarly p_2 must be chosen in the range $0 \leq p_2 \leq 1$.

3.4 Case of Two Categories:

Let $M(I_1, I_2)$ denote the number of documents for which $C_{m1} = I_1$ and $C_{m2} = I_2$ where the values of I_1 and I_2 are restricted to 0 and 1. Then with the $M(I_1, I_2)$ ordered as in figure (3.1) it is clear that

$$M(1,1) = p_{12}M$$

$$M(1,0) = p_1 M - M(1,1)$$

$$= (p_1 - p_{12})M$$

$$M(0,1) = p_2 M - M(1,1)$$

$$= (p_2 - p_{12})M$$

$$M(0,0) = M - M(1,0) - M(0,1) - M(1,1)$$

$$= (1 - p_1 + p_{12} - p_2 + p_{12} - p_{12})M$$

$$= (1 - p_1 - p_2 + p_{12})M$$

From the above relations it can be concluded that

$$0 \leq p_{12} \leq 1$$

$$p_{12} \leq p_1$$

$$p_{12} \leq p_2$$

$$p_{12} \geq p_1 + p_2 - 1$$

This last inequality can be written as

$$p_1 + p_2 - p_{12} \leq 1$$

which states that the number of documents in either category 1 or category 2 cannot exceed the total number of documents.

If L_{ij} and M_{ij} are defined as

$$L_{ij} = \text{Max } \{0, p_i + p_j - 1\}$$

$$M_{ij} = \text{Min } \{p_i, p_j, 1\}$$

then p_{12} must be chosen to satisfy the condition

$$L_{12} \leq p_{12} \leq M_{12}$$

This can always be satisfied since $p_1 + p_2 - 1$ is always less than both p_1 and p_2. It follows that two categories may be created by choosing any values of p_i and p_j in the range 0 to 1 and then choosing p_{ij} in the range

$$L_{ij} \leq p_{ij} \leq M_{ij}$$

3.5 Case of Three Categories:

Now suppose that it is desired to create third category. The probabilities p_3, p_{13}, p_{23} must satisfy the relation

$$0 \leq p_3 \leq 1$$

$$L_{13} \leq p_{13} \leq M_{13}$$

$$L_{23} \leq p_{23} \leq M_{23}$$

which may be written as

$$p_1+p_3-1 \leq p_{13} \leq \text{Min}(p_1,p_3)$$

$$p_2+p_3-1 \leq p_{23} \leq \text{Min}(p_2,p_3)$$

For three categories, with documents as ordered in figure(3.1), the following relations result

$$M(1,1,1) = p_{123}M$$

$$M(1,1,0) = p_{12}M - M(1,1,1)$$
$$= (p_{12}-p_{123})M$$

$$M(1,0,1) = p_{13}M - M(1,1,1)$$
$$= (p_{13}-p_{123})M$$

$$M(0,1,1) = p_{23}M - p_{123}M$$
$$= (p_{23}-p_{123})M$$

$$M(1,0,0) = p_1M - M(1,1) - M(1,0,1)$$
$$= (p_1-p_{12}-p_{13}+p_{123})M$$

$$M(0,1,0) = (p_2-p_{12}-p_{23}+p_{123})M$$

$$M(0,0,1) = (p_3-p_{13}-p_{23}+p_{123})M$$

$$M(0,0,0) = M(0,0) - M(0,0,1)$$
$$= M(0) - M(0,1) - M(0,0,1)$$
$$= (1-p_1-p_2+p_{12}-p_3+p_{13}+p_{23}-p_{123})M$$
$$= (1-p_1-p_2-p_3+p_{12}+p_{13}+p_{23}-p_{123})M$$

In order that the above proportions be non-negative the following inequalities must be satisfied.

$$0 \leq p_{123} \leq 1$$

$$p_{123} \leq p_{12}$$

$$p_{123} \leq p_{13}$$

$$p_{123} \leq p_{23}$$

$$p_{123} \geq p_{12}+p_{13}-p_1$$

$$p_{123} \geq p_{13}+p_{23}-p_3$$

$$p_{123} \geq p_{12}+p_{23}-p_2$$

$$p_{123} \leq 1-p_1-p_2-p_3+p_{12}+p_{13}+p_{23}$$

Thus p_{123} must be chosen in the range

$$L_{123} \leq p_{123} \leq M_{123}$$

where

$$L_{ijk} = \text{Max}(0, p_{ij}+p_{ik}-p_i)$$

$$M_{ijk} = \text{Min}(p_{ij}, 1-p_i-p_j-p_k+p_{ij}+p_{ik}+p_{jk})$$

It may be noted that the conditions

$$p_{123} \geq p_{12}+p_{13}-p_1$$

$$p_{123} \leq 1-p_1-p_2-p_3+p_{12}+p_{13}+p_{23}$$

are not contradictory if

$$p_{12}+p_{13}-p_1 \leq 1-p_1-p_2-p_3+p_{12}+p_{13}+p_{23}$$

$$0 \leq 1-p_2-p_3+p_{23}$$

which is always true provided p_{23} is chosen to satisfy

$$L_{23} \leq p_{23} \leq M_{23}$$

Similarly the conditions

$$p_{123} \leq p_{12}$$

$$p_{123} \geq p_{12}+p_{13}-p_1$$

$$p_{12}+p_{13}-p_1 \leq p_{12}$$

$$p_{13}-p_1 \leq 0$$

are satisfied since $p_{13} \leq p_1$. However the conditions

$$p_{123} \leq p_{23}$$

$$p_{123} \geq p_{12}+p_{13}-p_1$$

are contradictory unless

$$p_{12}+p_{13}-p_1 \leq p_{23}$$

or

$$p_{12}+p_{13}-p_{23} \leq p_1$$

This last inequality states that the number of documents of category 1 that are also present in category 2 or 3 cannot exceed the total number of documents in category 1.

In summary for the three categories

(a) p_i may be chosen arbitrarily between 0 and 1.

(b) p_{ij} must be chosen in the range

$$L_{ij} \leq p_{ij} \leq M_{ij}$$

subject to the constraints

C1: $p_{12}+p_{13}-p_{23} \leq p_1$

C2: $p_{12}+p_{23}-p_{13} \leq p_2$

C3: $p_{13}+p_{23}-p_{12} \leq p_3$

(c) p_{123} must be chosen in the range

$$L_{123} \leq p_{123} \leq M_{123}$$

It may be noted that the p_{ij} can be chosen simultaneously to satisfy (b) or, alternatively, the probabilities p_1, p_2, p_{12} may be chosen for two categories, and then $p_3, p_{13}, p_{23}, p_{123}$ chosen to satisfy (b) and (c) with C1, C2, C3 in the form

C1: $p_{13}-p_{23} \leq p_1-p_{12}$

C2: $p_{23}-p_{13} \leq p_2-p_{12}$

C3: $p_{13}+p_{23}-p_3 \leq p_{12}$

3.6 Case of Four Categories:

The numbers $M(I_1,I_2,I_3,I_4)$ must satisfy the equations

$$M(1,1,1,1) = p_{1234}\,M$$

$$M(1,1,1,0) = p_{123}M-M(1,1,1,1)$$
$$=(p_{123}-p_{1234})M$$

$$M(1,1,0,1) = (p_{124}-p_{1234})M$$

$$M(1,0,1,1) = (p_{134}-p_{1234})M$$

$$M(0,1,1,1) = (p_{234}-p_{1234})M$$

$$M(1,1,0,0) = p_{12}M-M(1,1,1,0)-M(1,1,0,1)-M(1,1,1,)$$
$$=(p_{12}-p_{123}-p_{124}+p_{1234})M$$

$$M(1,0,1,0) = (p_{13}+p_{123}-p_{134}+p_{1234})M$$

$$M(1,0,0,1) = (p_{14}-p_{124}-p_{134}+p_{1234})M$$

$$M(0,1,1,0) = (p_{23}-p_{123}-p_{234}+p_{1234})M$$

$$M(0,1,0,1) = (p_{24}-p_{124}-p_{234}+p_{1234})M$$

$$M(0,0,1,1) = (p_{34}-p_{134}-p_{234}+p_{1234})M$$

$$M(1,0,0,0) = p_1M-M(1,1)-M(1,0,1)-M(1,0,0,1)$$
$$=(p_1-p_{12}-p_{13}-p_{14}+p_{123}+p_{124}+p_{134}-p_{1234})M$$

$$M(0,1,0,0) = (p_2-p_{12}-p_{23}-p_{24}+p_{123}+p_{124}+p_{234}-p_{1234})M$$

$$M(0,0,1,0) = (p_3-p_{23}-p_{13}-p_{34}+p_{123}+p_{134}+p_{234}-p_{1234})M$$

$$M(0,0,0,1) = (p_4-p_{14}-p_{24}-p_{34}+p_{134}+p_{234}+p_{124}-p_{1234})M$$

$$M(0,0,0,0) = M(0,0,0)-M(0,0,0,1)$$
$$=M(0,0)-M(0,0,1)-M(0,0,0,1)$$
$$=M(0)-M(0,1)-M(0,0,1)-M(0,0,01)$$

$$= M - M(1) - M(0,1) - M(0,0,1) - M(0,0,0,1)$$

$$= (1 - p_1 - p_2 + p_{12} - p_3 + p_{23} + p_{13} - p_{123} - p_4 + p_{14} + p_{24} + p_{34} - p_{134} - p_{234} - p_{124} + p_{1234})M$$

$$= (1 - p_1 - p_2 - p_3 - p_4 + p_{12} + p_{13} + p_{14} + p_{23} + p_{24} + p_{34} - p_{123} - p_{124} - p_{134} - p_{234} + p_{1234})M$$

It follows that

$$0 \leq p_{1234} \leq 1$$

$$p_{1234} \leq p_{123}$$

$$p_{1234} \geq p_{123} + p_{124} - p_{12}$$

$$p_{1234} \leq p_{123} + p_{124} + p_{134} - p_{12} - p_{13} - p_{14} + p_1$$

$$p_{1234} \geq p_{123} + p_{124} + p_{134} + p_{234} - p_{12} - p_{13} - p_{14} - p_{23} - p_{24} - p_{34} + p_1 + p_2 + p_3 + p_4 - 1$$

The conditions

$$p_{1234} \leq p_{123}$$

$$p_{1234} \geq p_{123} + p_{124} - p_{12}$$

are not contradictory since $p_{124} \leq p_{12}$

The conditions

$$p_{1234} \leq p_{234}$$

$$p_{1234} \geq p_{123} + p_{124} - p_{12}$$

are not contradictory provided

$$p_{123} + p_{124} - p_{12} \leq p_{234}$$

and hence

$$p_{123} + p_{124} - p_{234} \leq p_{12}$$

which states that the number of documents of categories 1 and 2 that are also present in category 3 or 4 cannot exceed the total

number of documents in categories 1 and 2

The conditions

$$p_{1234} \leq p_{123}$$

$$p_{1234} \geq p_{123}+p_{124}+p_{134}+p_{234} - p_{12}-p_{13}-p_{14}-p_{23}-p_{24}-p_{34} + p_1+p_2+p_3+p_4-1$$

are not contradictory if

$$1-p_1-p_2-p_3+p_{12}+p_{13}+p_{23}-p_4 + p_{14}+p_{24}+p_{34}-p_{124}-p_{134}-p_{234} \geq 0 \qquad (3.1)$$

The conditions

$$p_{1234} \leq p_{123}+p_{124}+p_{134}-p_{12}-p_{13}-p_{14}+p_1$$

$$p_{1234} \geq p_{123}+p_{124}-p_{12}$$

are not contradictory if

$$p_{134}-p_{13}-p_{14}+p_1 \geq 0$$

or $p_{134} \geq p_{13}+p_{14}-p_1$

which is known to be true

The conditions

$$p_{1234} \leq p_{123}+p_{124}+p_{134}-p_{12}-p_{13}-p_{14}+p_1$$

$$p_{1234} \geq p_{134}+p_{234}-p_{34}$$

are not contradictory if

$$p_{123}+p_{124}-p_{234}-p_{12}-p_{13}-p_{14}+p_{34}+p_1 \geq 0$$

or $$p_{123}-p_{12}-p_{13}+p_1+p_{124}-p_{234}-p_{14}+p_{34} \geq 0 \qquad (3.2)$$

The conditions

$$p_{1234} \leq p_{123}+p_{124}+p_{134}-p_{12}-p_{13}-p_{14}-p_1$$

$$p_{1234} \geq p_{123}+p_{124}+p_{134}+p_{234}-p_{12}-p_{13} - p_{14}-p_{23}-p_{24}-p_{34}+p_1+p_2+p_3+p_4-1$$

are satisfied if

$$1-p_2-p_3-p_4+p_{23}+p_{24}+p_{34}-p_{234} \geq 0$$

which is known to be true from analysis of the case of three categories.

In summary for four categories

(a) Choose p_i, p_{ij} as for three categories

(b) Choose p_{ijk} in the range

$$L_{ijk} \leq p_{ijk} \leq M_{ijk}$$

subject to the conditions

C1: $p_{123}+p_{124}-p_{234} \leq p_{12}$

$p_{123}+p_{124}-p_{134} \leq p_{12}$

C2: $p_{123}+p_{134}-p_{124} \leq p_{13}$

$p_{123}+p_{134}-p_{234} \leq p_{13}$

C3: $p_{124}+p_{134}-p_{234} \leq p_{14}$

$p_{124}+p_{134}-p_{123} \leq p_{14}$

C4: $p_{123}+p_{234}-p_{124} \leq p_{23}$

$p_{123}+p_{234}-p_{134} \leq p_{23}$

C5: $p_{124}+p_{234}-p_{123} \leq p_{24}$

$p_{124}+p_{234}-p_{134} \leq p_{24}$

C6: $p_{134}+p_{234}-p_{124} \leq p_{34}$

$p_{134}+p_{234}-p_{123} \leq p_{34}$

(c) Choose p_{1234} in the range

$$L_{1234} \leq p_{1234} \leq M_{1234}$$

$$L_{1234}=\text{Max}(0, p_{ijk}+p_{ijr}-p_{ij}, p_{123}+p_{124}+p_{134}+p_{234} - p_{12}-p_{13}-p_{14}-p_{23}-p_{24}-p_{34}+p_1+p_2+p_3+p_4-1)$$

$$M_{1234}=\text{Min}(p_{ijk}, p_{ijk}+p_{ijr}+p_{ikr}-p_{ij}-p_{ik}-p_{ir}+p_i)$$

In order to illustrate the process of choosing a possible set of probabilities the following example is given.

Example:

Suppose that first order probabilities are given. By a first order probability is meant the probability that a document belongs to one given category. A second order probability denotes the probability of belonging to two categories, and so forth for third and fourth order probabiilities.

Given:

$p_1 = 0.4$ $p_2 = 0.3$ $p_3 = 0.2$ $p_4 = 0.1$

then

$$L_{ij} = \text{Max}(0, p_i + p_j - 1)$$

$$L_{12} = \text{Max}(0, 0.7 - 1)$$

$$L_{12} = 0$$

$$M_{12} = \text{Min}(p_1, p_2)$$

$$M_{12} = 0.3$$

$$0 \leq p_{12} \leq 0.3$$

p_{12} is assumed to be uniformly distributed in this range and might be chosen to have the particular value of $p_{12} = 0.25$.

Similarly the following second order probabilities may be selected.

$$p_{13} = 0.15 \qquad p_{14} = 0.05$$

$$p_{23} = 0.13 \qquad p_{24} = 0.04$$

$$p_{34} = 0.02$$

These satisfy the required constraints

C1: $p_{12} + p_{13} - p_1 \leq p_{23}$

since

$$0.25 + 0.15 - 0.4 \leq p_{23}$$

$$0 \leq p_{23}$$

C2: $p_{12}+p_{23}-p_2 \leq p_{13}$

since

$$0.25 + 0.13 - 0.3 \leq .15$$

$$0.08 \leq 0.15$$

C3: $p_{23}+p_{13}-p_3 \leq p_{12}$

since

$$0.13 + 0.15 - 0.15 \leq 0.25$$

Case Of Three Categories:

$$L_{123} = \text{Max}(0, p_{12}+p_{13}-p_1, p_{12}+p_{23}-p_2, p_{23}+p_{13}-p_3)$$

$$= \text{Max}(0, 0, .08, .08)$$

$$M_{123} = \text{Min}(p_{12}, p_{23}, p_{13}, 1-p_1-p_2-p_3+p_{12}+p_{13}+p_{23})$$

$$M_{123} = \text{Min}(.25, .13, .15, 1-.4-.3-.2+.25+.15+.13)$$

$$= \text{Min}(.25, .13, .15, .63)$$

It follows from the relation

$$L_{123} \leq p_{123} \leq M_{123}$$

that p_{123} might be selected as 0.1090 ;

Case Of Four Categories:

In the similar manner as above, the other third order probabilities may be selected as

$$p_{124} = 0.0380$$

$$p_{134} = 0.0157$$

$$p_{234} = 0.0060$$

chosen to satisfy the following constraints:

C1: $p_{123}+p_{124}-p_{12} \leq p_{234}$

$.109+.0380-.250 \leq p_{234}$

$-.10297 \leq p_{234}$

C2: $p_{123}+p_{134}-p_{13} \leq p_{124}$

$-.02527 \leq p_{124}$

C3: $p_{124}+p_{134}-p_{14} \leq p_{234}$

$.00375 \leq p_{234}$

C4: $p_{123}+p_{234}-p_{23} \leq p_{124}$

$-.01504 \leq p_{124}$

C5: $p_{124}+p_{234}-p_{24} \leq p_{123}$

$.00397 \leq p_{123}$

C6: $p_{134}+p_{234}-p_{34} \leq p_{124}$

$.00168 \leq p_{124}$

Then

$L_{1234} = \text{Max}(0, p_{ijk}+p_{ijr}-p_{ij}, c)$

$= \text{Max}(0, -.10297, -.02527, .00375,$

$-.01504, .00397, .00168, -.47129)$

$= 0.00397$

$M_{1234} = \text{Min}(p_{123}, p_{124}, p_{134}, p_{234}, p_{ijk}+p_{ijr}$

$+p_{ikr}-p_{ij}-p_{ik}-p_{ir}+p_i)$

$= \text{Min}(.109, .038, .0157, .006, .1127$

$.0329, .0306, .0497)$

$= .006$

and so

$.00397 \leq p_{1234} \leq .006$

Thus p_{1234} may be chosen as

$p_{1234} = .00487$

3.7 Probability Calculations:

The subject span of each category may be very general or very narrow. The number of categories should be just enough to categorise a certain area of knowledge. The size of the categories will also depend on how fine is the discrimination between categories within the particular area of knowledge. By varying the first order and second order probabilities, the number of categories to which a document will belong can be varied. Thus choice of the first and second order probabilities determines whether the subject span of each category is very general or very restricted. In order to observe this effect four cases of given first order probabilities and second order probabilities have been selected. The foregoing method of calculating third and fourth order probabilities has beeen programmed, and the results are tabulated in Table (3.1) for the following four cases.

Case (i)

$p_1 = 0.4$, $p_2 = 0.3$, $p_3 = 0.2$, $p_4 = 0.1$

$p_{12} = 0.25$, $p_{13} = 0.15$, $p_{14} = .05$,

$p_{23} = 0.13$, $p_{24} = 0.04$, $p_{34} = 0.02$

Case (ii)

$p_1 = p_2 = p_3 = p_4 = 0.3$

$p_{12} = p_{13} = p_{14} = p_{23} = p_{24} = p_{34} = 0.1$

Case (iii)

$p_1 = p_2 = p_3 = p_4 = 0.5$

$p_{12} = p_{13} = p_{14} = p_{23} = p_{34} = 0.2$

Case (iv)

$p_1 = p_2 = p_3 = p_4 = 0.5$

$p_{12} = .48 \quad p_{13} = .42 \quad p_{14} = 0.45$

$p_{23} = .43 \quad p_{24} = .445 \quad p_{34} = .44$

3.8 Document Categorisation:

Suppose it is decided to have four subject categories for the purpose of experiment. A given corpus of documents is to be indexed using these categories. For given first order probabilities,and using the procedure of the foregoing sections,the second,third,and fourth order probabilities may be calculated. The problem is then to assign the documents to their respective categories. Then with $M(I_1, I_2)$, etc as defined in section 3.3 it follows that

$M(1,1) = p_{12}M$

$M(1,0) = p_1M - M(1,1)$

$M(0,1) = p_2M - M(1,1)$

$M(0,0) = M - M(1,0) - M(0,1) - M(1,1)$

$M(1,1,1) = p_{123}M$

$M(1,1,0) = p_{12}M - M(1,1,1)$

$M(1,0,1) = p_{13}M - M(1,1,1)$

$M(0,1,1) = p_{23}M - M(1,1,1)$

$M(1,0,0) = p_1M - M(1,1) - M(1,0,1)$

$M(0,1,0) = M(0,1) - M(0,1,1)$

$= p_2M - M(1,1) - M(0,1,1)$

$M(0,0,1) = p_3M - M(0,1,1) - M(1,0,1) - M(1,1,1)$

$M(0,0,0) = M(0,0)-M(0,0,1)$

$=M-M(1,0)-M(0,1)-M(1,1)-M(0,0,1)$

Similarly for four categories:

$M(1,1,1,1) = p_{1234}M$

$M(1,1,0,1) = p_{124}M-M(1,1,1,1)$

$M(1,0,1,1) = p_{134}M-M(1,1,1,1)$

$M(1,0,0,1) = p_{14}M-M(1,0,1,1)-M(1,1,0,1)-M(1,1,1,1)$

$M(0,1,1,1) = p_{234}M-M(1,1,1,1)$

$M(0,1,0,1) = p_{24}M-M(1,1,0,1)-M(0,1,1,1)-M(1,1,1,1)$

$M(0,0,1,1) = p_{34}M-M(0,1,1,1)-M(1,0,1,1)-M(1,1,1,1)$

$M(0,0,0,1) = p_{4}M-M(0,0,1,1)-M(0,1,0,1)-M(0,1,1,1)$

$-M(1,0,0,1)-M(1,0,1,1)-M(1,1,0,1)-M(1,1,1,1)$

From the trend of the above results, a general recursive formula for computation of $M(I_1,I_2,..,1)$ may be derived in the following form:

$$M(I_1,I_2,\ldots I_r,1)=p_{n1n2..ns.r+1}M - \sum_{J_i} M(J_1,J_2,\ldots J_r,1)$$

Where

(a) n1,n2,.....ns are values of n for which $I_n=1$

(b) $J_i = I_i$ if $I_i = 1$

(c) The summation is over all combinations of $J_i(=0,1)$ for which $I_i = 0$ excluding the combination $J_1,J_2,\ldots\ldots,J_r = I_1,I_2,\ldots\ldots,I_r$

Similarly

$M(1,1,1,0) = M(1,1,1)-M(1,1,1,1)$

$M(1,1,0,0) = M(1,1,0)-M(1,1,0,1)$

$$M(1,0,1,0) = M(1,0,1) - M(1,0,1,1)$$

$$M(0,1,1,0) = M(0,1,1) - M(0,1,1,1)$$

$$M(1,0,0,0) = M(1,0,0) - M(1,0,0,1)$$

$$M(0,1,0,0) = M(0,1,0) - M(0,1,0,1)$$

$$M(0,0,1,0) = M(0,0,1) - M(0,0,1,1)$$

$$M(0,0,0,0) = M(0,0,0) - M(0,0,0,1)$$

Then $M(I_1,I_2,....I_r,0)$, in general may be calculated from

$$M(I_1,I_2,..I_r,0) = M(I_1,I_2,..I_r) - M(I_1,I_2,..I_r,1)$$

In order to partition the documents among different categories, the above recursive formulas have been programmed and the results of partitioning for two values of M, namely M=50 and M=5000, are presented in figures(3.2) to (3.5). For any value of m (document number) the determination of which of the first r categories contain the document is equivalent to determination of which of the ranges of document numbers contain the values of m.

TABLE 3.1: Calculated Probabilities

CASE NO.	p_1	p_2	p_3	p_4	p_{12}	p_{13}	p_{14}	p_{23}	p_{24}	p_{34}	p_{123}	p_{124}	p_{134}	p_{234}	p_{1234}
i	0.4	0.3	0.2	0.1	0.25	0.15	0.05	0.13	0.04	0.02	0.1090	0.0380	0.01570	0.0060	.0048
ii	0.3	0.3	0.3	0.3	0.10	0.10	0.10	0.10	0.10	0.10	0.0580	0.0400	0.0706	0.0298	0.02918
iii	0.5	0.5	0.5	0.5	0.20	0.20	0.20	0.20	0.20	0.20	0.058	0.038	0.0786	0.0298	0.01510
iv	0.5	0.5	0.5	0.5	0.48	0.42	0.45	0.43	0.445	0.44	0.41580	0.4443	0.4000	0.3984	0.39802

It may be noted that in Case (ii) the data base is covered by four categories that contain equal numbers of documents and have relatively little overlap. Thus the third and fourth order probabilities are very small. In contrast the Case (iv) corresponds to categories that have considerable overlap.

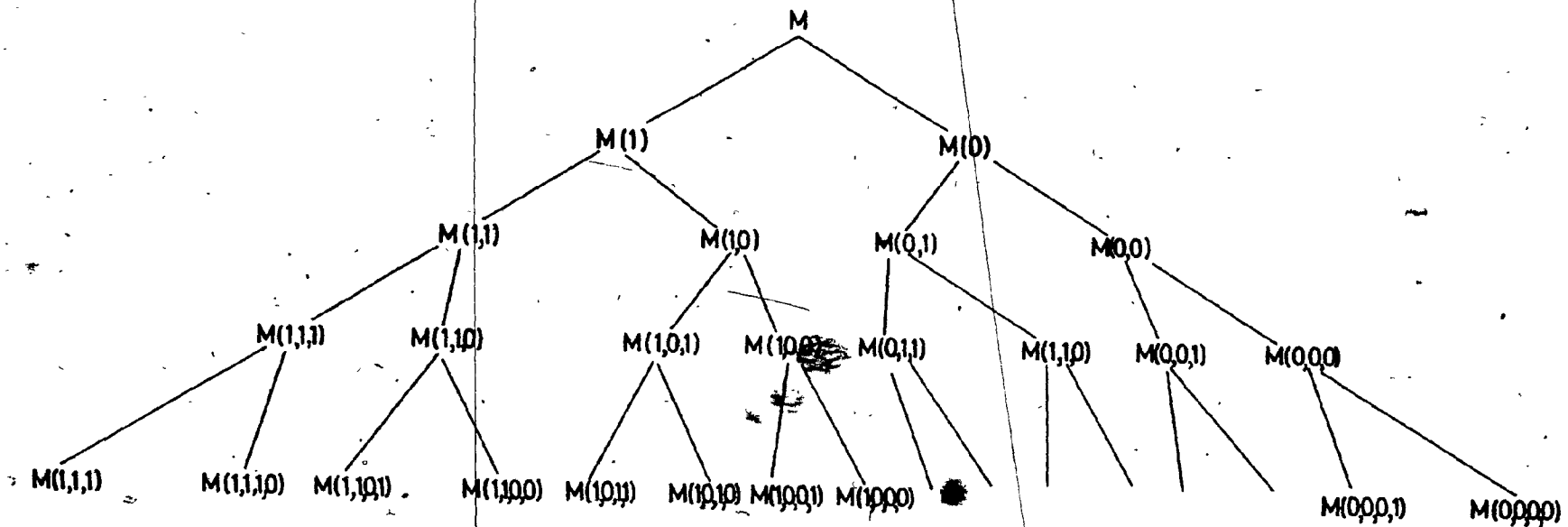


Fig. 3.1 Number of documents in certain combinations of categories.

Case(i)

M =50

Range of document number m				Properties
1	To	20	C_{m1}=	1
21		50		0
1		12	C_{m1}, C_{m2}=	1 1
13		20		1 0
21		22		0 1
23		50		0 0
1		5	C_{m1}, C_{m2}, C_{m3} =	1 1 1
6		12		1 1 0
13		14		1 0 1
15		20		1 0 0
21		21		0 1 1
22		22		0 1 0
23		23		0 0 1
24		50		0 0 0
0		0	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
1		5		1 1 1 0
6		7		1 1 0 1
8		12		1 1 0 0
13		13		1 0 1 1
14		14		1 0 1 0
0		0		1 0 0 1
15		20		1 0 0 0
0		0		0 1 1 1
21		21		0 1 1 0

0	0	0 1 0 1
22	22	0 1 0 0
0	0	0 0 1 1
23	23	0 0 1 0
24	26	0 0 0 1
27	50	0 0 0 0

Fig.3.2 Properties of ordered set of documents with respect to the first, first two, first three and first four categories.

Case(i)

M =5000

Range of document number m			Properties
1	To 2000	C_{m1}=	1
2001	5000		0
1	1250	C_{m1}, C_{m2}=	1 1
1251	2000		1 0
2001	2250		0 1
2251	5000		0 0
1	545	C_{m1}, C_{m2}, C_{m3} =	1 1 1
546	1250		1 1 0
1251	1455		1 0 1
1456	2000		1 0 0
2001	2105		0 1 1
2106	2250		0 1 0
2251	2395		0 0 1
2396	5000		0 0 0
1	24	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
25	545		1 1 1 0
546	710		1 1 0 1
711	1250		1 1 0 0
1251	1304		1 0 1 1
1305	1455		1 0 1 0
1456	1460		1 0 0 1
1461	2000		1 0 0 0
2001	2005		0 1 1 1
2006	2105		0 1 1 0

2106	2109	0 1 0 1
2110	2250	0 1 0 0
2251	2265	0 0 1 1
2266	2395	0 0 1 0
2396	2619	0 0 0 1
2620	5000	0 0 0 0

Fig.3.2 (continued)

Case(ii)

M =50

Range of document number m				Properties
1	To	15	C_{m1}=	1
16		50		0
1		5	C_{m1}, C_{m2}=	1 1
6		15		1 0
16		25		0 1
26		50		0 0
1		2	C_{m1}, C_{m2}, C_{m3} =	1 1 1
3		5		1 1 0
6		7		1 0 1
8		15		1 0 0
16		17		0 1 1
18		25		0 1 0
26		32		0 0 1
33		50		0 0 0
1		1	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
2		2		1 1 1 0
3		3		1 1 0 1
4		5		1 1 0 0
6		7		1 0 1 1
0		0		1 0 1 0
8		8		1 0 0 1
9		15		1 0 0 0
0		0		0 1 1 1
16		17		0 1 1 0

18	20	0 1 0 1
21	25	0 1 0 0
26	26	0 0 1 1
27	32	0 0 1 0
33	38	0 0 0 1
39	50	0 0 0 0

Fig.3.3 Properties of ordered set of documents with respect to the first,first two,first three and first four categories.

Case(ii)

M =5000

Range of document number m			Properties	
1	To	1500	C_{m1}=	1
1501		5000		0
1		500	C_{m1}, C_{m2}=	1 1
501		1500		1 0
1501		2500		0 1
2501		5000		0 0
1		290	C_{m1}, C_{m2}, C_{m3} =	1 1 1
291		500		1 1 0
501		710		1 0 1
711		1500		1 0 0
1501		1710		0 1 1
1711		2500		0 1 0
2501		3290		0 0 1
3291		5000		0 0 0
1		145	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
146		290		1 1 1 0
291		344		1 1 0 1
345		500		1 1 0 0
501		707		1 0 1 1
708		710		1 0 1 0
711		802		1 0 0 1
803		1500		1 0 0 0
1501		1503		0 1 1 1
1504		1710		0 1 1 0

1711	2006	0 1 0 1
2007	2500	0 1 0 0
2501	2643	0 0 1 1
2644	3290	0 0 1 0
3291	3846	0 0 0 1
3847	5000	0 0 0 0

Fig.3.3 (continued)

Case(iii)

M =50

Range of document number m			Properties	
1	To	25	C_{m1}=	1
26		50		0
1		10	C_{m1}, C_{m2}=	1 1
11		25		1 0
26		40		0 1
41		50		0 0
1		2	C_{m1}, C_{m2}, C_{m3} =	1 1 1
3		10		1 1 0
11		17		1 0 1
18		25		1 0 0
26		32		0 1 1
33		40		0 1 0
41		47		0 0 1
48		50		0 0 0
0		0	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
1		2		1 1 1 0
3		4		1 1 0 1
5		10		1 1 0 0
11		13		1 0 1 1
14		17		1 0 1 0
18		22		1 0 0 1
23		25		1 0 0 0
0		0		0 1 1 1
26		32		0 1 1 0

33	39	0 1 0 1
40	40	0 1 0 0
41	45	0 0 1 1
46	47	0 0 1 0
48	49	0 0 0 1
50	50	0 0 0 0

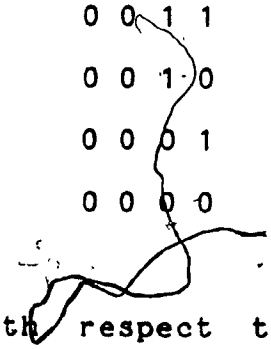


Fig.3.4 Properties of ordered set of documents with respect to the first,first two,first three and first four categories.

Case(iii)

M =5000

Range of document number m				Properties
1	To	2500	C_{m1}=	1
2501		5000		0
1		1000	C_{m1}, C_{m2}=	1 1
1001		2500		1 0
2501		4000		0 1
4001		5000		0 0
1		290	C_{m1}, C_{m2}, C_{m3} =	1 1 1
291		1000		1 1 0
1001		1710		1 0 1
1711		2500		1 0 0
2501		3210		0 1 1
3211		4000		0 1 0
4001		4790		0 0 1
4791		5000		0 0 0
1		75	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
76		290		1 1 1 0
291		404		1 1 0 1
405		1000		1 1 0 0
1001		1317		1 0 1 1
1318		1710		1 0 1 0
1711		2202		1 0 0 1
2203		2500		1 0 0 0
2501		2573		0 1 1 1
2574		3210		0 1 1 0

3211	3946	0 1 0 1
3947	4000	0 1 0 0
4001	4533	0 0 1 1
4534	4790	0 0 1 0
4791	4946	0 0 0 1
4947	5000	0 0 0 0

Fig.3.4 (continued)

Case(iv)

M =50

Range of document number m				Properties
1	To	25	C_{m1}=	1
26		50		0
1		24	C_{m1}, C_{m2}=	1 1
25		25		1 0
26		26		0 1
27		50		0 0
1		20	C_{m1}, C_{m2}, C_{m3} =	1 1 1
21		24		1 1 0
0		0		1 0 1
25		25		1 0 0
0		0		0 1 1
26		26		0 1 0
27		29		0 0 1
30		50		0 0 0
1		19	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
20		20		1 1 1 0
21		23		1 1 0 1
24		24		1 1 0 0
0		0		1 0 1 1
0		0		1 0 1 0
0		0		1 0 0 1
25		25		1 0 0 0
0		0		0 1 1 1
0		0		0 1 1 0

0	0	0 1 0 1
26	26	0 1 0 0
27	27	0 0 1 1
28	29	0 0 1 0
0	0	0 0 0 1
30	50	0 0 0 0

Fig.3.5 Properties of ordered set of documents with respect to the first,first two,first three and first four categories.

Case(iv)

M =5000

Range of document number m				Properties
1	To	2500	C_{m1}=	1
2501		5000		0
1		2400	C_{m1}, C_{m2}=	1 1
2401		2500		1 0
2501		2600		0 1
2601		5000		0 0
1		2709	C_{m1}, C_{m2}, C_{m3} =	1 1 1
2080		2400		1 1 0
2401		2421		1 0 1
2422		2500		1 0 0
2501		2571		0 1 1
2572		2600		0 1 0
2601		2929		0 0 1
2930		5000		0 0 0
1		1990	$C_{m1}, C_{m2}, C_{m3}, C_{m4}$=	1 1 1 1
1991		2079		1 1 1 0
2080		2310		1 1 0 1
2311		2400		1 1 0 0
2401		2409		1 0 1 1
2410		2421		1 0 1 0
2422		2439		1 0 0 1
2440		2500		1 0 0 0
2501		2501		0 1 1 1
2502		2571		0 1 1 0

2572	2572	0 1 0 1
2573	2600	0 1 0 0
2601	2798	0 0 1 1
2799	2929	0 0 1 0
2930	2977	0 0 0 1
2978	5000	0 0 0 0

Fig.3.5 (continued)

CHAPTER 4

DOCUMENT TERM MATRIX

Before describing the method of simulation of the document term matrix, it is appropriate to consider a collection of documents, each of which contains information about one or more subjects. Each document record is composed of words and sentences. For a given set of categories, the problem is how to fit the individual documents into the categories. If semantic considerations are ignored, the technique of automatic indexing is mainly statistical and involves determination of certain probability relationships between individual content-bearing words and categories. The resulting relationships are then used to predict the category to which the document may belong.

The approach is based on the fact that individual content words may be used to index a document, and the values of their probability distributions in a collection can be used to classify the documents. The frequency distributions of the words provide enough information to allow reasonable prediction about subject matter of the documents that contain the words. However not all the words present in a document can be regarded as providing a clue to document subject matter. The selection of key words from a document collection is another problem. Generally, very high frequency words and very low frequency words are not selected as key words. Experience shows that in many data bases about 20% of

the total set of distinct terms can be chosen as key words. Having selected keywords,the next problem is to find how they are associated with certain classes of documents. The resulting statistical information will enable a new document,having certain keywords,to be labelled as belonging to certain categories.

For simulation, in contrast to classification,the situation is somewhat reversed. For a given set of documents it is necessary to generate terms and their associations with documents. This may be specified by generation of a document term matrix. For example,suppose there are four categories and the documents have already been classed with respect to these four categories. The assignment of the keywords for the document set cannot be purely random but must be controlled so that certain keywords tend to associate with documents of certain classes. This is to say that keywords must be assigned so that there is a statistical correlation between keywords and categories. Such control is necessary in order to make the simulation model a correct portrayal of the real world. This chapter describes a document-term generation process with a mechanism to control the association of keywords with certain document categories in such a way that keywords are constrained to occur in the documents pertinent to certain categories.

4.1 Method of Keyword Assignment:

Suppose it is known that some terms tend to associate with certain categories. For each term and category a measure of

association may be expressed in terms of ratios of relative frequencies so that a term-category matrix is created whose i,k th entry represents a relative frequency f_{ik} defined as follows

V_{ik} =number of documents of k th category that contain the i th term.

M_k = number of documents in the k th category

$$f_{ik} = V_{ik}/M_k$$

The relative frequency can be interpreted to have the following significance. If, for a given term I and set of categories,the relative frequency is highest for the k th category then the I th term has more bias toward association with documents in that category. For a given document that contains this I th term there is more likelihood for the document to be associated with the k th category than with any other category.

A second relative frequency is defined to be M_i/M where

M_i =number of documents that contain I th term

M =number of documents in the data base

The ratio of the two relative frequencies is

$$R_{ik} = f_{ik}*M/M_i$$

and may be called the "classification rating" of the I th term with respect to the k th category.

Some particular values of R_{ik} that correspond to special situations are as follows:

(1) R_{ik} = 0 if no document of the k th category contains the I th term.

(2) $R_{ik} = 1$ if the proportion of documents of the k th category that contain the I th term is the same as the proportion of all documents that contain the I th term.

(3) $R_{ik} = M/M_i$ if the I th term appears in all the documents of the k th category.

(4) R_{ik} has an expected value of $1/p_k$ if the I th term appears only in documents of the k th category.

The last relation can be made more clear by considering an example. Suppose V_{ik} and M_k are as defined on the previous page. Then for (4) above $V_{ik} = M_k$, and

$$R_{ik} = \frac{V_{ik}/M_k}{V_{ik}/M} = M/M_k$$

$$= \frac{\text{number of document in database}}{\text{number of documents in k th category}}$$

$$= M/p_k * M = 1/p_k$$

A value of R_{ik} greater than 1 indicates the extent to which documents in the k th category are more likely to contain the I th term than are documents chosen from the entire database. Similarly, a value of R_{ik} less than 1 indicates the extent to which documents in the k th category are less likely to contain the I th term.

It may be noted that

number of documents in the k th category = $p_k * M$

The expected number of documents of the k th category that contain the I th term is

$$N_{ik} = f_{ik} * p_k * M$$
$$= R_{ik} * M_i * p_k$$

Therefore the expected proportion of documents in the k th category that contain the ith term is

$$f_{ik} = R_{ik} * M_i / M$$

The number of documents that contain the I th term but are not in the k th category is $M_i - p_k * R_{ik} * M_i$. The proportion of documents that contain the I th term but are not in the k th category is therefore

$$(1 - p_k * R_{ik}) M_i / (1 - p_k) M$$

The association of documents with terms may be described by a document term matrix, of M rows and D columns of elements d_{mi} where

d_{mi} = 1 if m th document contains I th term

= 0 otherwise

4.2 Constraints to be Satisfied by R_{ik}:

Let $r_k(i)$ denote the number of occurrences of the I th term in documents that are in the k th category but in no other category. Let $r_{jk}(i)$ denote the number of occurrences of the I th term in documents that are in both the j th and k th category but in no other. Let $r_o(i)$ denote the number of occurrences of the I th term in documents that are in no category.

Since the number of documents in the k th category that contain the I th term is $p_k * R_{ik} * M_i$ it follows that

$$r_k(i) + \sum_{J \neq K} r_{jk}(i) + \sum_{\substack{J \neq S \\ J \neq K \\ S \neq K}} r_{jsk}(i) + \ldots = p_k R_{ik} M_i$$

Also, since the number of documents that contain the I th term is M_i then

$$r_o(i) + \sum_{K} r_k(i) + \sum_{\substack{J,K \\ J \neq K}} r_{jk}(i) + \sum_{J,K,S \text{ (different)}} r_{jks}(i) + \ldots = M_i$$

The $r_k(i), r_{jk}(i)$, etc. must satisfy the above equations and also the following inequalities.

$0 \leq r_k(i) \leq$ number of documents in k th category $= p_k * M$

$0 \leq r_{jk}(i) \leq$ number of documents in both j and k th category $= p_{jk} * M$

$0 \leq r_o(i) \leq$ number of documents in no category

The above equations and inequalities place constraints on the allowed values of the R_{ik}. For example if k=2 the relations become

$r_1(i) + r_{12}(i) = p_1 R_{i1} M_i$

$r_2(i) + r_{12}(i) = p_2 R_{i2} M_i$

$r_o(i) + r_1(i) + r_2(i) + r_{12}(i) = M_i$

$0 \leq r_1(i) \leq p_1 M$

$0 \leq r_2(i) \leq p_2 M$

$0 \leq r_{12}(i) \leq p_{12} M$

$0 \leq r_o(i) \leq (1 - p_1 - p_2 + p_{12}) M$

If the first three equations are solved for $r_1(i)$, $r_2(i)$ and $r_{12}(i)$ the inequalities may be written as

$$M_i - r_o(i) - p_2 M \leq p_1 R_{i1} M_i \leq M_i - r_o(i)$$

$$M_i - r_o(i) - p_1 M \leq p_2 R_{i2} M_i \leq M_i - r_o(i)$$

$$M_i - r_o(i) \leq p_1 R_{i1} M_i + p_2 R_{12} M_i \leq M_i - r_o(i) + p_{12} M$$

$$0 \leq r_o(i) \leq (1 - p_1 - p_2 + p_{12}) M$$

Thus R_{i1} and R_{i2} must be constrained to satisfy the above equations.

Continuing as above, it may be seen that for values of k greater than 2, for any I th term ; the inequality constraints to be satisfied by R_{ik} become very involved. The corresponding computations necessary to determine the restrictions on occurrences of the I th term according to the above relations would also be unwarranted. It therefore appears impractical to specify R_{ik} directly. Rather, a simpler approach as described in the next section is suggested.

4.3 Term Simulation Algorithm:

The I th column of the document term matrix should contain M_i elements d_{mi} equal to 1. The remaining $M-M_i$ elements d_{mi} are equal to zero.

Consider the instance of k =4. If the association of the I th term with documents is purely random, and not dependent on the document category, then M_i values for which $d_{mi}=1$ may be distributed uniformly among the M elements d_{mi}. The number of

occurrences of the I th term in the documents for which

$C_{m1}, C_{m2}, C_{m3}, C_{m4} = I_1, I_2, I_3, I_4$

is then

$$M(I_1, I_2, I_3, I_4) * M_i / M$$

However, if the association of terms with documents is not independent of document categories then for some values of I_1, I_2, I_3, I_4, the number of occurrences of the term in the corresponding documents will be greater than $M(I_1, I_2, I_3, I_4) * M_i / M$. Likewise for other values of I_1, I_2, I_3, I_4 the number of occurrences will be less than $M(I_1, I_2, I_3, I_4) * M_i / M$.

Experience with databases used for information retrieval leads one to conclude that some terms are more closely associated with certain document classes than are other terms. Clearly there are correlations between terms and subject categories or classes, and term occurrence is not independent of the document categories.

The following procedure may be used to choose values of d_{mi}, for a given value of i, in such a manner that term occurrences depend on document categories.

For each value of i, let sixteen values of $W_i(I_1, I_2, I_3, I_4)$ be chosen in the range 0 to 1. These values may be called the category weights of the I th term. Choose two sequences of uniformly distributed pseudo-random numbers:

$r_i(1), r_i(2), r_i(3)$ ..... in range $1 \leq r_i(n) \leq M$

$s_i(1), s_i(2), s_i(3)$ ..... in range $0 \leq s_i(n) \leq 1$.

4.4 Algorithm 1:

Step 1: set $d_{mi}=0$ for all m=1,M. Set count=0,n=1.

Step 2: compute $r_i(n)$.

Step 3: if $d_{ri(n),i} = 1$ go to step 7.

Step 4: determine m_j and m_{j+1} (range of document number m according to section 3.7 of chapter 3) in which $r_i(n)$ lies

$$m_j < r_i(n) \leq m_{j+1}$$

This range determines the I_1 I_2,I_3,I_4 that describe the behaviour of the document with respect to category.

Step 5: if $s_i(n) \leq W_i(I_1,I_2,I_3,I_4)$ then set $d_{ri(n),i}=1$ and count=count+1; otherwise let $d_{ri(n),i}$ remain zero.

Step 6: if count=M_i then stop.

Step 7: n=n+1 then go to step 2.

The above procedure chooses M_i values of d_{mi} equal to 1. The following special cases are noteworthy.

(a) If all $W_i(I_1,I_2,I_3,I_4)$ are chosen equal to 1 then document term associations are independent of document categories. Likewise if all $W_i(I_1,I_2,I_3,I_4)$ are chosen to be equal. This case thus corresponds to $R_{ik}=1$ for all k.

(b) If $W_i(1,0,0,0)=1$; and all other $W_i(I_1,I_2,I_3,I_4)=0$, then the I th term will be assigned only to documents in the first, and no other, category. Also $R_{i1}=1/p_1$ and $R_{ik}=0$ for $k \neq 1$

The above steps 1-7 are meaningful only if $M_i \leq M(1,0,0,0)$

(c) If $W_i(1,I_2,I_3,I_4)=1$, and $W_i(0,I_2,I_3,I_4)=0$, then the I th term will be assigned only to documents of the first category. This case corresponds to $R_{i1}=1/p_1$ but does not determine R_{ik} for $k \neq 1$.

the above steps 1- 7 are meaningful for case (c) only if $M_i \leq M(1)$

(d) In case (b) if M_i is greater than $M(1,0,0,0)$ then eventualy the I th term will have been assigned to $M(1,0,0,0)$ documents. For the remaining $M_i - M(1,0,0,0)$ documents ,step 5 will never lead to an assignment of the I th term with a document, and the process will not terminate. A similar situation will arise for case (c) if M_i is greater than $M(1)$.

(e) If $W_i(1,0,0,0) = .9$, and all other $W_i(I_1,I_2,I_3,I_4) = .1$, the following two situations may arise:

(i) If $M_i \leq M(1,0,0,0)$ then <u>approximately</u> M_i occurrences of the I th term will be assigned to documents in the first, and no other, category. Thus

$R_{i1} \simeq 1/p_1$ and $R_{ik} \approx 0$ for $k \neq 1$

(ii) If $M_i > M(1,0,0,0)$ then approximately $M(1,0,0,0)$ occurrences of the I th term will be assigned to documents in the first,and no other, category. Thus $f_{i1} \approx M(1,0,0,0)/p_1 * M$ and

$$R_{i1} \approx M(1,0,0,0)/p_1 * M_i$$

The remaining $M_i - M(1,0,0,0)$ occurrences of the I th term

will be uniformly distributed among the remaining M-M(1,0,0,0) documents. This is due to the fact that for the remaining documents $W_i(I_1,I_2,I_3,I_4)$ is equal to 0.1. Any document generated in this range of category combinations is equally likely to be allotted to the I th term. The number of occurrences of the I th term in documents of the second category is therefore

$$(M(1,1)+M(0,1))(M_i-M(1,0,0,0)/(M-M(1,0,0,0)).$$

Thus

$$R_{i2} \simeq \left[\frac{M(1,1)+M(0,1)}{M-M(1,0,0,0)}\right]\left[\frac{M_i-M(1,0,0,0)}{M(1,1)+M(0,1)}\right]\cdot\frac{M}{M_i}$$

$$\simeq \frac{M_i-M(1,0,0,0)}{M-M(1,0,0,0)}\cdot\frac{M}{M_i}$$

Similarly

$$R_{i3} \simeq \frac{[M(1,1)+M(1,0,1)+M(0,1,1)+M(0,0,1)][M_i-M(1,0,0,0)]}{[M-M(1,0,0,0)][M(1,1,1)+,(1;0,1)+M(0,1,1)+M(0,0,1)]}\frac{M}{M_i}$$

$$\simeq \frac{M_i-M(1,0,0,0)}{M-M(1,0;0,0)}\cdot\frac{M}{M_i}$$

So

$$R_{ik} \simeq \frac{M_i-M(1,0,0,0)}{M-M(1,0,0,0)}\cdot\frac{M}{M_i}$$

(f) If $W_i(1,I_2,I_3,I_4)=.9$, and $W_i(0,I_2,I_3,I_4)=.1$, the following two situations may arise:

(1) If $M_i \leq M(1)$ then approximately M_i occurrences of the I th term will be assigned to documents in the first category. Thus

$R_{i1} \leq 1/p_1$ and R_{ik} cannot be determined

(2) If $M_i > M(1)$ then approximately $M(1)$ occurrences of the I th term will be assigned to documents in the first category. Thus

$$f_{i1} = M(1)/p_1 * M$$

$$R_{i1} \approx M(1)/p_1 * M_i$$

From the above it may be concluded that if for each I the 16 values of $W_i(I_1, I_2, I_3, I_4)$ are chosen as pseudo-random numbers uniformly distributed in the range 0 to 1 then the resulting values of d_{mi} will be such that.

(i) Some terms will tend to be associated with certain combination of categories.

(ii) No assumption has been made regarding the association of particular terms with particular categories.

This allotment procedure may not lead to strong associations of some terms with documents in particular categories. It does not clearly guarantee that some terms are more discriminating for certain categories. So such a choice of the $W_i(I_1, I_2, I_3, I_4)$ does not allow specification of the extent to which particular terms may associate with particular document classes.

Let the terms in the database be designated as "content","non-content",and "accidental associative" terms according to following definition.

(1) The I th term will be called a "content" term for the 1st category if $W_i(1, I_2, I_3, I_4) = .95$ and $W_i(0, I_2, I_3, I_4) = .05$. Such content terms are the ones that are likely to be useful

in determination of document categorisation.

Note: The definition may need modification by addition of the condition that $M_i \leq M(1)$ or $M_i \leq 2*M(1)$.

(2) The I th term will be called a "non-content" term if all $W_i(I_1, I_2, I_3, I_4)=1$. Such non-content terms are useless for the purpose of document classification.

(3) the I th term will be called an "accidentally associative" term if all $W_i(I_1, I_2, I_3, I_4)$ are chosen as pseudo-random numbers in the range 0 to 1. Such terms may appear to be associated with categories but the associations are accidental and of no use in prediction of the categories of additional documents that may subsequently be added to the data base.

Suppose the probabilities of a term being a content, non-content, or accidental associative term are assumed to be p_c, p_n, and $1-p_c-p_n$ respectively. Let the I th column of the document term matrix be simulated as follows:

For each I let t(i) denote a pseudo-random number uniformly distributed in the range 0 t(i) 1. Let n(i) denote a pseudo-random integer uniformly distributed among the values 1,2,3,4. Let the $W_i(I_1, I_2, I_3, I_4)$ be determined according to the scheme:

(i) If $t(I) \leq p_c$ then regard the I th term as a content term for the n(i) th category. Thus

$$W_i(I_1, I_2, I_3, I_4) = .95 \text{ if } I_{n(i)} = 1$$
$$= .05 \text{ if } I_{n(i)} = 0$$

(ii) If $p_c < t(i) \leq p_c+p_n$ then regard the I th term as a non-

content term, and set all $W_i(I_1, I_2, I_3, I_4) = 1$.

(iii) If p_c+p_n is less than t(i) then regard the I th term as an accidental associative term, and choose the $W_i(I_1, I_2, I_3, I_4)$ from a uniform distribution of random numbers in the range from 0 to 1.

After determination of the $W_i(I_1, I_2, I_3, I_4)$ as above, the Algorithm 1 for simulation of values of d_{mi} (m=1,M) is used in a form modified as follows:

4.5 Modified Version of Algorithm 1:

For a particular I th term, this algorithm chooses values of d_{mi}=1 in such a way that the I th term is constrained to occur in a manner as described in the previous section. The $s_i(n)$, rc(i) are pseudo-random numbers uniformly distributed in the range 0 to 1. The $r_i(n)$ and n(i) are integers uniformly distributed in the range 1 to M and 1 to 4 respectively. Let

p_c = probability that the term is a content term

p_n = probability that the term is a non-content term

$M_i = A*N/i$ where $A = 1/[\log(D)+\gamma]$,

N = total numbers of terms

The algorithm is as follows:

Step 1: Set $d_{mi} = 0$ for all m=1 to M. Set count=0.

Step 2: Compute rc(i). If $rc(i) \leq p_c$, label it as a content term and generate n(i) between 1 to 4 and go to step 3. If $rc(i) > p_c+p_n$ generate a random number $W_i(I_1, I_2, I_3, I_4)$ between 0 and 1 and go to step 3. Otherwise set

$W_i(I_1, I_2, I_3, I_4) = 1$.

Step 3: Compute $r_i(n)$.

Step 4: If $d_{ri(n),i}=1$ go to step 9.

Step 5: Determine m_j and m_{j+1} (range of document number m according to section 3.7 of chapter 3) for which $m_j \leq r_i(n) \leq m_{j+1}$

Hence determine the set I_1, I_2, I_3, I_4 that describes the behaviour of the document with respect to the categories.

Step 6: If the term is a content term then set

$W_i(I_1, I_2, I_3, I_4) = .95$ if $I_{n(i)}=1$

$= .05$ if $I_{n(i)}= 0$

and go to step 7.

Step 7: Generate $s_i(n)$ between 0 to 1. If $s_i(n) \leq W_i (I_1, I_2, I_3, I_4)$ then set $d_{ri(n),i} =1$, count=count+1, and go to step 8. Otherwise go to step 9.

Step 8: If count = M_i then stop.

Step 9: n=n+1 and go to step 3.

A flow chart of above algorithm is presented in Fig.(4.1).

In order to illustrate the steps of the above logic consider an example of 50 documents containing 70 distinct terms and a total number of terms equal to 400. Suppose the 50 documents are sorted into two categories as follows:

M(1,1) = documents numbered 1 to 12

M(1,0) = 13 to 20

M(0,1) = 21 to 22

M(0,0) = 23 to 50

For a given i, suppose that rc(i) is less than p_c, so that the I th term is a content term, and n(i) =1. This means that the term has bias toward documents of category 1. Therefore set $W_i(1,I_2)=.95$ and $W_i(0,I_2)=.05$. The steps of the algorithm proceed as follows.

1: n=1

2: rc(i) is less than p_c, so it is a content term

3: say $r_i(1)=14$

5: since $r_i(1)=14$ which is in the range 13 to 20, and this determines $I_1,I_2=1,0$

7: if $s_i(1) \leq .95$ set $d_{14,i}=1$ (This tends to set $d_{14,i}=1$)

.....

9: n = 2

3: $r_i(2) = 22$

5: $r_i(2)=22$ lies in the range 21 to 22, and this determines $I_1,I_2=0,1$

7: if $s_i(2) \leq .05$ set $d_{22,i}=1$ (This tends to set $d_{22,i}=1$)

From the above it is clear that as $r_i(1), r_i(2)$.....run through random document numbers there is a tendency to associate the I th term with documents in category 1 and not with documents in other categories.

The logic of the modified algorithm was tested on a data base with the following parameters.

M=50 N=400 D=70

$p_c=.20$ $p_n=.70$ k=4

The machine printed out a document term matrix in which any non-

zero element in the m th row indicates the presence of a term in the m th document. It is observed that the machine labelled 13 terms as content, 51 terms as non-content and 6 terms as accidental associative terms.

From the above document term matrix a term category matrix for content terms only was generated. Its i,k th element is the classification rating($R_{i,k}$) of the I th term, and this matrix is presented in Fig.(4.2). A careful observation of this matrix shows that:

Terms 4,18,50 have high rating for category 2.

Terms 6,34,54 have high rating for category 3.

Terms 17,20,37,58 have high rating for category 4.

Terms 10,67,69 have high rating for category 2.

The step 7 of the algorithm gives the values of n(i) for these terms and it is found that

n(4)=n(18)=n(50)=2

n(6)=n(34)=n(54)=3

n(17)=n(20)=n(58)=4

n(10)=n(67)=n(69)=1.

Thus the algorithm tends to make the terms of number (4,18,50),(6,34,54),(17,20,58),(10,67,69) biased toward categories 2,3,4,1 respectively.

In all these cases, the tendency of the algorithm to associate certain terms with certain categories is corroborated by $R_{i,k}$ being of high value for the I th term if the I th term is biased to that category except for terms 10,67,69. Although these terms

are biased to category 1 they also have a high value of $R_{i,2}$ for category 2. This is due to two main reasons (1) Although the algorithm tends to assign the I th term to the documents of the category to which the I th term is biased, it may happen that the probabilistic procedure allots the biased term to a document of another category. This is believed to be in agreement with the fact that a real document data base often contains unexpected associations of words. The fact that $R_{i,2}$ has a higher value than $R_{i,1}$ for terms 10,67 is due to the normalising effect of the denominator. The terms 10 and 67 appear in some documents of category 1 and 2, but the number of documents of category 2 is less than the number of documents of category 1 and this normalisation makes $R_{i,2}$ greater than $R_{i,1}$.

(2) Because of Zipf's law distribution of terms, the terms of high rank occur infrequently. Consequently, and particularly for any term whose frequency of occurrence is only 1, the term may become associated with a document of other category than the one for which it is important. The probability of such happening is low, but allowance for such is believed to be important and to reflect the behaviour of real document data bases. So it is not surprising that term 69, although discriminating for category 1, has been allotted to a document of category 2.

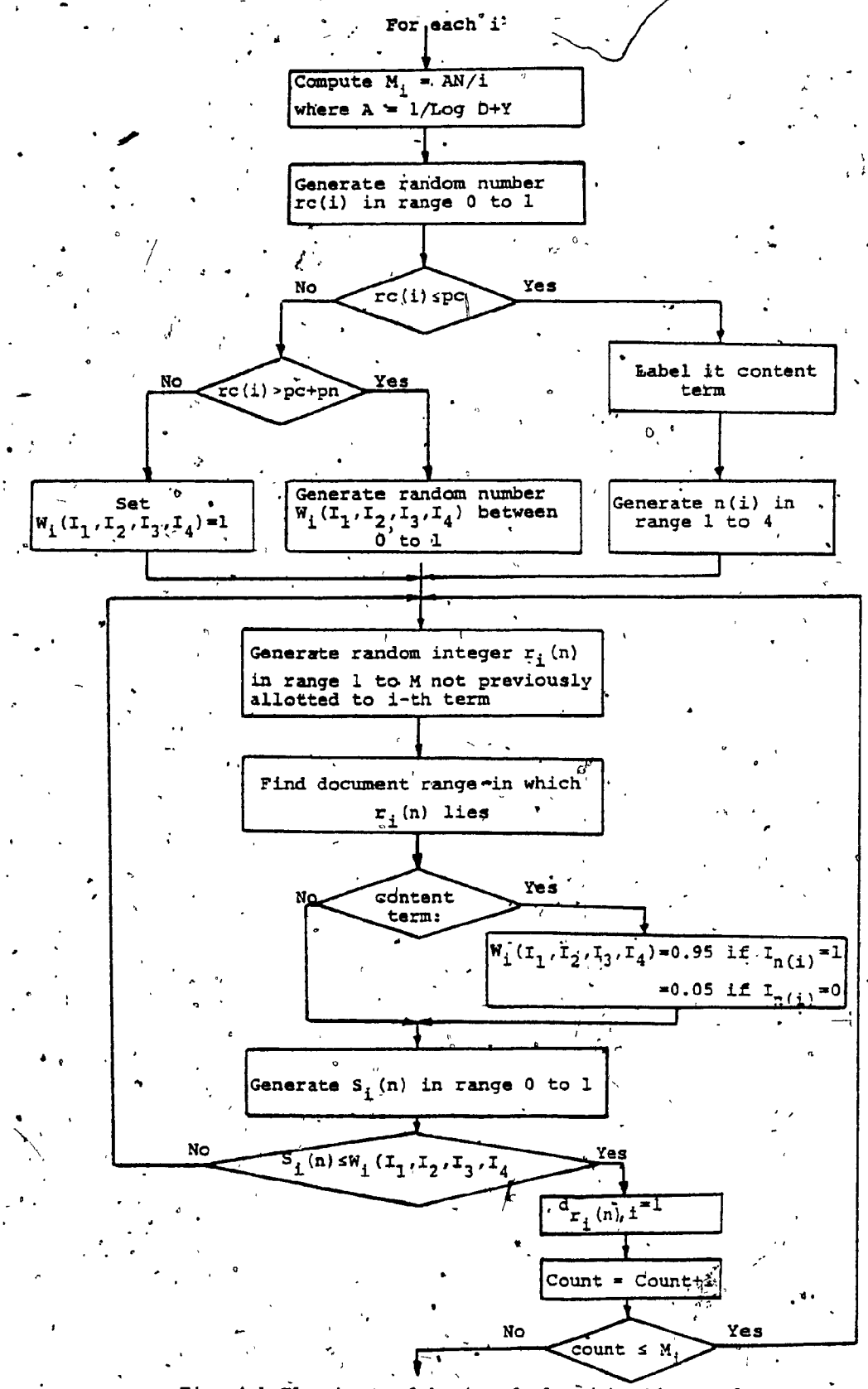


Fig. 4.1 Flowchart of logic of Algorithm 1

CONTENT TERMS	CATEGORY NUMBER 1	2	3	4
4	1.75	2.16	1.66	1.66
6	1.15	1.28	2.99	1.28
10	1.56	1.66	1.39	0
17	1.25	1.66	1.39	6.25
18	1.875	2.25	1.39	0
20	1.25	1.67	0	4.17
34	1.25	1.67	2.78	0
37	1.25	0	2.78	4.17
50	2.50	3.33	0	0
54	0	3.33	5.55	0
58	2.50	3.33	0	8.33
67	2.50	3.30	0	0
69	0	3.33	.10	0

Fig.(4.2) Content terms vs Category Matrix

CHAPTER 5

EXPERIMENTAL WORK

Automatic document classification is the process by which the category to which a given document belongs is decided by mechanical means. In the context of information retrieval the classification is to be used for some purpose which will naturally affect the choice of method and the results obtained. The purpose may be to enable documents to be grouped in such a way that retrieval is faster, and the effectiveness of the classification scheme may be observed during retrieval.

The choice of categories is normally pre-determined, and it is important that the pre-determined category structure is reasonably broad so that all the pertinent areas of knowledge are covered by the structure. It is not necessary that categories be mutually exclusive, and they may overlap to some degree. The subject span of each category should be fairly broad, and certainly more general than the average request for information.

The problem of document categorisation reduces to the process of determining the subject content of each document. The subject content of a document is often indicated by keywords that have been deduced from examination of the full text of the document. The degree of relevance of the document to a particular category may be determined from the correlation between the document

subject content and the category subject span. The fact that two documents belong to the same category,or categories,indicates that two documents have something common and are related to each other. The two documents may be regarded as mutually relevant if they contain some keywords in common. It follows that if a document belongs to a particular category, or categories,the allotment of keywords to the document has to be consistent (or in agreement) with the set of keywords associated with the relevant categories. This has been kept in view during the document term matrix generation in chapter 4, whereas the document category matrix generation process has been described in chapter 3.

It follows from the above discussion that determination of the relevances of given documents to particular categories is the crux of the whole classification process. The higher the relevance rating of a given document to a particular category k,the greater is the chance that the document will be indexed under that category k. During manual classification the classifier tries to make a subjective judgement of the relevance of a document to a particular category. The judgement is normally based on his experience and his knowledge of the area to which the document belongs. It may also be affected by noting the keywords of the document. Since the computer does not have the capabilities of the human brain, then to allow computer assessment of relevance it is important that relevance be expressed as some mathematical function of the keywords of the document.

The association of key words with a document can be described by a binary function d_{mi} which is equal to 1 if the I th key word is present in the m th document and equal to zero if the I th keyword is not present in the m th document. It is assumed for simplicity of computation that the relevance $V_k(m)$ is a linear function of d_{mi} and is given by[10]

$$V_k(m) = \sum_i^{\lambda} d_{mi} * a_{ki} \tag{5.1}$$

where a_{ki} are constants.

The a_{ki} may be computed automatically as follows:

$$a_{ki} = V_{ki}/U_{ii} \tag{5.2}$$

where

U_{ii} = proportion of documents that contain the I th keyword

V_{ki} = proportion of documents that contain the I th keyword and are in the k th category.

Alternatively a_{ki} can be decided manually, and in the following discussion its values have been chosen equal to 1, in which case the expression for the relevance rating of the m th document to the k th category reduces to the form

$$V_k(m) = \sum d_{mi} \tag{5.3}$$

where summation is over specified values of I for the content terms for the k th category

5.1 Experimental Program:

The motive behind the relevance rating analysis can be two fold (1) A new document enters the system for cataloging, and keywords associated with the document are known. The machine

looks to see which of the keywords in the document match with content terms of each category. For each category the machine computes the relevance rating given by equation(5.3) and prints out ranked output for each category. The document may then be allotted to a particular category for which this rating is highest, or some other criteria of allotment to more than one category can be used. (2) Suppose a given corpus of documents has been indexed by using some classification scheme. On the basis of occurrence of content terms associated with each category, and for a particular document,the machine may compute the relevance rating for the document. This rating can be calculated for all the documents of collection. For a given threshhold level of relevance rating the number of correctly categorised documents, and the total documents at that level of threshhold,can be determined. From this,one may decide on the effectiveness of the particular classification scheme. It is this second aspect of classification for which the experimental program is designed and applied as explained below.

There are four main components of the experiment,namely (1)Pseudo-documents (2)Pseudo term and document associations(3)Pseudo-document and category associations (4)A pseudo stochastic variate uniformly distributed between 0 to 1 to decide the relevance of the term to any given category. The pseudo quantities so generated constitute the data base for the experiments. The generation of pseudo document-category associations and pseudo term document associations have been described in detail in chapter 3 and 4 respectively.

Suppose the following parameters of the data base are known

M = total number of documents.

N = total number of terms.

D = total number of distinct terms.

p_c = probability that a given term is a content term.

p_n = probability that a given term is a non-content term.

$W_i(I_1,I_2,I_3,I_4)$ = category weights. The I_1,I_2,I_3,I_4 may be 1 or 0.

The set of four quantities I_1,I_2,I_3,I_4 describes the behaviour of the document with respect to the four categories.

It has been observed that Zipf's law for word frequencies applies to real document data bases. In agreement with this, the model assumes that pseudo term occurrences are according to Zipf's law, and that terms are ranked according to decreasing frequency of their occurrences. Suppose M_1 denotes the frequency of the occurrence of I th term. Then

$$M_i = A*N/I$$

where

$$A = 1/[\text{Log}(D)+.5772]$$

Suppose the M documents have been already categorised into four categories according to the scheme of chapter 3. The number of documents indexed under particular combinations of categories (total 16 combinations) is input to the program. For a particular given category, k, the program proceeds as follows:

All sixteen possible category combinations, along with the number of documents in each combination, are read. The first step is to find the total number of documents in category k. For a given I th term, its total number of occurrences is equal to M_i. Before the term and document associations are generated, it is necessary to know what type of term it is. This is determined by generating a pseudo-random number rc between 0 and 1. If rc is less than p_c, the term is a content term, otherwise it is not. The program considers only content terms because it is the content terms which can discriminate documents of one category from those of another. If the I th term is non-content, then the next term is considered and the same criterion is applied. If it is determined that the I th term is a content term the next step is to determine the category for which it behaves as a content term. This is done by generating another pseudo-random integer n(i) between 1 and 4. A separate seed is given for use in generation of this integer. The I th term is then a content term for the documents of category k = n(i).

At this stage the pseudo-document and term associations are created. A pseudo-random number is generated and it is converted into an integer R_n between 1 and M. This number constitutes the pseudo document number. By scanning the ranges of the category combination the combination (I_1, I_2, I_3, I_4), to which the document belongs, is determined. The next step is to determine the category weights($W_i(I_1, I_2, I_3, I_4)$). The weight $W_i(I_1, I_2, I_3, I_4)$ is then set to be .95 if $I_{n(i)} = 1$ and .05 if $I_{n(i)} = 0$. In order to

illustrate further, suppose that n(i) =1 and the pseudo document number R_n lies in the range $M(1,I_2,I_3,I_4)$. Since n(i) in this case is 1, the weight $W_i(I_1,I_2,I_3,I_4)$ =.95 and the term is biased to documents of category 1. Next another pseudo-random number $s_i(n)$ uniformly distributed between 0 and 1 is generated. If $s_i(n)$ is less than $W_i(I_1,I_2,I_3,I_4)$, then I th term is associated with R_n otherwise it is not. This process is repeated until M_i occurrences of the I th term have been assigned. It follows that if M_i is greater than $M(1,I_2,I_3,I_4)$, then approximately $M(1,I_2,I_3,I_4)$ occurrences of the I th term will take place in documents of category 1. The remaining $M_i-M(1,I_2,I_3I_4)$ occurrences will be distributed randomly among the other documents.

For a given k, as I ranges from 1 to D the program selects only those content terms that are biased to category k. As each I th term is allotted to the k th category, column vector $V_k(m)$ defined in equation (5.1), which contains relevance ratings for pseudo documents, is updated. Thus at the end of program when all content terms for category k, are processed as above, the relevance rating analysis can be carried out. This is done in another unit of the program.

For a given threshhold level of relevance, the number of documents whose relevance exceeds that level is determined. Let this number of documents be denoted by TRT. Out of the set of TRT documents let RRT denote the number of documents that also belong to category k. The ratio RRT/TRT is chosen as an

index of classification effectiveness for the given threshhold. This ratio is traditionally called the PRECISION. Another quantity ,the ratio of RRT to the total number of documents in category k is also computed. This quantity is called the RECALL.

A complete flow chart of the above procedure is presented in figure(5.1). As the threshhold level is increased, the total number of documents (TRT) having relevance rating equal to this new threshhold decreases. The proprtion RRT also decreases but the decrease in RRT is not as much as in TRT. As a result,with increase of threshhold;PRECISION increases and RECALL decreases. At the maximum level of the threshhold the PRECISION is 100% and the RECALL approaches zero.

5.2 Experimental Results:

Let us consider a small model data base for testing the program. The parameters for this test model are M = 50, N = 400, D = 70 p_c = .2, p_n = .7.

$W_i(I_1,I_2,I_3,I_4)$ = .95 if $I_{n(i)}$ = 1

= .05 if $I_{n(i)}$ =0

The machine prints out results as follows:

Total documents of category 1 = 20

The pseudo document numbers are= from 1 to 20 for category 1.

The content terms of category 1 = 10,67,69.

Of the total of 70 different terms,the machine prints out 13 as content terms. Out of 13,only 3 terms(10,67,69) are biased to category 1. The following document numbers of documents with

threshhold level =1 were printed out.

2,3,4,10,12,18,21,41,45,50.

TRT = 10

RRT = 6

PRECISION = 6/10 and RECALL = 6/20

There were no documents of threshhold level 2. For this small data base, the machine also printed out the term-document association matrix whose elements are d_{mi}. This was done to allow checking of the authenticity of the relevance rating analysis. From this, the number of documents in which terms 10,67,69 occur was counted and these documents matched exactly with those as printed with rating number=1. Manually observed documents relevant to category 1, and with threshhold = 1, were found to be in agreement with those printed out by the machine.

The program was also tested with a prototype database of the following parameters

$M = 5000$, $N = 40000$, $D = 7000$, $p_c = .2$, $p_n = .7$

$W_i(I_1, I_2, I_3, I_4) = .95$ if $I_{n(i)} = 1$

$= .05$ if $I_{n(i)} = 0$

For this data base the range from which content terms were selected was restricted to I values between 50 and D. This was because, as a consequence of Zipf's law, the most frequent 49 terms have very high frequencies and do not provide useful information about the subject content of a document. Their contribution to discriminion of documents of one category from another is negligible. Hence they are rejected as candidates for content

terms. Therefore the content terms are selected out of the range 50 to D.

The following results were printed by the machine for the probabilities of case (i) of chapter 3 and for category 1 only.

Category k = 1

Content terms of category 1 = 358

Total documents at threshhold 1 = 862

Total relevant documents at threshhold 1(RRT) = 777

PRECISION = 777/862=.901, RECALL = 777/2000 . = .389

At threshhold 2 TRT = 168

At threshhold 2 RRT = 167

PRECISION = 167/168 = .994, RECALL=167/2000=.084

At threshhold 3 TRT = 24

At threshhold 3 RRT = 24

PRECISION = 24/24=1,RECALL = 24/2000=.012

Similarly at thresh 4, PRECISION = 3/3=1,RECALL = 3/2000 = .002

There were no documents of threshhold level 4. The above results for each category for four different cases are plotted in figure (5.2,5.3,5.4,5.5).

5.3 Predicted Relevance Rating Analysis:

The relevance rating, as defined by equation(5.1),is a prediction of the extent to which a document belongs to the given category. This mathematical function cannot replace the complex function of the human brain as a subjective judge of the

relevance of the document to a particular category, but it may predict the relevance rating of a document to the category in question. Hence,the values of relevance ratings derived from this function are called predictive relevance ratings[10]. The values of a_{ki} in equation (5.2) be computed from a knowledge of the statistical distribution of content terms in the documents. In the further analysis described below the values of a_{ki} are computed by the machine.

For the purpose of further work,the equation (5.1) is interpreted in two ways. In the first case,the summation over I is carried out for all the content terms of category k. The basis for this procedure is the supposition that if a document m is categorised under category k,then it is likely to have content terms of category k. It may also have some terms that are content terms for categories kl other than k. But the number of occurrences of content terms of categories kl in document m is likely to be a small number in comparison with the number of occurrences of terms of category k. Based on this reasoning it is considered that terms of category kl will lead to a small contribution to the relevance rating of the m th document with respect to the k th category. This method in which summation in equation (5.1) I over the content terms of particular category under consideration will be termed the predicted relevance method 1.

On the other hand, it can be reasoned that a few terms of category kl that occur in document m will enhance(however small

it may be) the relevance rating of the m th document with respect to category k. For example suppose that for category 1 the content terms are of low frequency distribution as per Zipf's law. Because of their low frequency, they cannot occur in all the documents of category 1. Say, they do not occur in documents 1 and 2 of category 1. Suppose these documents 1,2 belong to category 2 also and terms of category 2 occur in these documents. In the first method the relevance rating of these documents 1,2 to category 1 will be equal to zero. But in the second case, the fact that content terms of category 2 occur in them, will contribute to the relevance of documents 1 and 2 to category 1 also. This is due to the correlation brought about by the term a_{ki} in equation (5.1). This method in which the summation is over all the content terms of the database will be called the relevance rating method 2.

5.4 Experimental Program:

Suppose a given corpus of documents has been indexed by using some classification scheme. On the basis of occurrences of content terms associated with each category then for particular documents the machine can compute the predictive relevance rating of the document to each category. This rating may be calculated for all documents of the collection. Then finally the accuracy of the classification can be checked by comparison with the criterian classification.

The equation(5.1) has been programmed on the computer. The procedure is according to the flow chart given in figure(5.1). The main modification to the previous procedure is that the vector Iv(5000) is replaced by the matrix YKM(5000,4) which is the relevance rating matrix.

At the end of execution of the program,when classification efficiency in terms of precision and recall is evaluated for all categories,the relevance rating matrix is printed. For the small database used above,and the probabilities of case(i),for method 1,out of a total of 50 documents the documents of numbers from 27 to 50 are not in any category. The first 26 documents are allotted to category, or category combinations, according to the scheme of chapter 3. Assuming that relevance ratings greater than .5 value indicates relevance to a category, then out of 41 category assignments there are 32 correct assignments; thus 79% are correct.

Finally a subroutine Rank is used to give ranked output so that for a given document the category with the highest relevance rating appears first and is followed by others in decreasing order of relevance. Each row of the ranked output also shows the category number. From the analysis of this ,it may be seen that the documents of number 15,19,20 do not have any content terms. So the question of their indexing to any category does not arise. For the remaining 23 documents, the question might be asked as to what is the probability that a category that appears at the top of the list is in fact the correct category. This amounts to

asking what is the probability that correct category has the highest relevance. This probability is denoted by P1. It is found that out of 23 documents there are 18 cases in which the highest relevance predicts the correct category; this corresponds to 78%. Similarly let P2 denote the probability that first two categories in the ranked list of categories are correct. Out of 14 such chances,10 are correct about 71%.

Examination of the relevance rating matrix shows that documents 1,5,6,7 have zero relevance for category 1. But according to the criterion classification they belong to category 1. This is explained by the fact that terms 10,67,69 which are content terms for category 1 do not occur in the above documents. In equation(5.1) the summation is over content terms only for the category in question. As a result,the respective relevance matrix entries for these documents with respect to category 1 have been set to zero. However it may be seen that content terms of category 2 (4,18) occur in these documents. If the summation over the content terms in equation 5.1 is to cover all the content terms of the data base,the constants a_{1i} for these terms with respect to category 1 will not be equal to zero. It follows that some contribution to the relevance of these documents by terms 4 and 18 with respect to category 1 will be made. In that case,the relevance rating for these documents will not be zero. This explains why the summation over I in equation (5.1) for method 2 is to include all content terms of the data base.

For method 2, the above program was modified to include the summation over all content terms of the data base. The above relevance rating of .5 was used as a threshold. The machine made a total of 51 category assignments of which 41 category assignments were correct, about 69%. Out of 23 documents, 19 assignments of highest relevance were correct, and this gave 82.61% as the percentage correct. Similarly P2 was found to be equal to 65.22%. As predicted above, although this method gave more correct category assignments, it also gave more false assignments.

5.5 Attribute Number Analysis:

Suppose the content terms of the pre-determined category structure are known. The categories are established before applying the present form of analysis. A document is scanned to determine the content terms it contains. Let the given document contain only one content term W_1. Let W_1 be random variable associated with W_1. Given the document contains W_1, what is the probability that it belongs to categories 1,2,3,4? This is given by the following expression[6]

$$P(K_j/W_i) = P(K_j)*P(W_i/K_j)/P(W_i) \qquad (5.4)$$

$P(W_i/k_j)$ is the probability that a document indexed under category K_j will contain the term W_i. Since $P(W_i)$ is constant with respect to j the above expression reduces to

$$P(K_j/W_i) = \text{cons.}*P(K_j)*P(W_i/K_j) \qquad (5.5)$$

Similarly, if a document contains n content terms then

$$P(K_j/W_1,W_2,\ldots W_n) = \text{cons.}*P(K_j)*P(W_n W_{n-1}\ldots W_1/K_j)$$

$= \text{cons.} * P(K_j) * P(W_n/W_{n-1} \ldots W_1, K_j) \ldots P(W_1/K_j)$ (5.6)

Assuming that relative to any given category, the two words W_n, W_{n-1} occur independently then

$P(K_j/W_1, W_2, \ldots W_n) = \text{cons.} * P(K_j) * P(W_n/K_j)\ P(W_{n-1}/K_j) \ldots P(W_1/K_j)$ (5,7)

The value given by this equation is called the Attribute Number of the given document with respect to category K_j [6]. It may be noted that this scheme of classification assumes that (1) keywords occurrences are independent, and (2) the categories are mutually exclusive and exhaustive. Both these assumptions are not valid in practice. However the assumptions simplify the process of automatic classification and have been made by other investigators. The conditional probabilities are estimated as follows:

$P(k_j)$ = (no of documents in category K_j)/(Total documents in database)

$P(W_n/k_j)$ = (no of documents that are in K_j and contain W_n) /(Total documents in K_j)

The equation (5.7) was programmed for the database with the following parameters:

$M = 50$, $N = 400$, $D = 70$, $p_c = .2$ and $p_n = .7$

Criterion classification---case (i)

The logic of the program is slightly different from that of the previous one. First of all the program generates document and content term associations for all content terms. For the term i, when M_i occurrences have been determined, the contribution to the attribute number of the given document m by term I (term I

occurs in m) is evaluated for each category and matrix YKM(M,K), for the m th row and k=1,2,3,4, is updated by use of equation (5.7). Then,for each category,the program calculates the classification efficiency in terms of precision and recall. In this case the threshhold level has to be kept very low. This is because of the nature of the computation by equation (5.7).

For a given category K,and for a particular threshhold level(IT),the program counts the number of documents that satisfy (IT) and belong to K. This gives RRT. Similarly TRT is the total documents which satisfy (IT) for that category k. For every category it was observed that,irrespective of threshhold,the precision and recall values were lower than those of the relevance rating analysis.

The attribute number matrix was also printed out. Out of 26 documents it was found that document numbers 15,19,20 have zero entries. These documents do not contain any of the content terms. The probability P1 for this case is 82%. It means that out of 23 cases,in 19 cases the correct category is at the top of the list. The probability P2 is 62%. There are 15 cases in which the first two categories are at the top of the list. It is surprising to note that looking at criterion classification case(i)(chapter 3) ,there are exactly 15 documents belonging to two categories.

Comparing this method, and the relevance rating method 2,for ranked output, it is found that in some cases,the category at the

top of the list is not the same. For example, document 3 for relevance rating category 1 is at the top of list, while in the attribute number method for the same document the category 3 is at the top of list.

These three methods were programmed on the computer for a data base of parameters

M = 5000,D = 7000,N = 40000,category weights .95 and .05.

The complete flow chart and logical components of the program have been dealt with in the previous pages. The relevance rating matrix, whose each element indicates the relevance of m th document to the k th category, is analysed, and results are presented in figures(5.6 to 5.13). These results are discussed in the following pages. It may be noted that, for the purpose of precision and recall, a relevance rating of .5 has been considered as the threshold.

The relevance rating matrix was also arranged in decreasing order of relevance rating of a document to all the categories. For illustration purposes only case(i) of probabilities is discussed. According to the criterian classification there are 2619 documents which belong to some category or category combinations. Out of 2619 documents 787 documents do not contain any of the content terms. So these 787 documents cannot be indexed. The remaining 1832 documents are found to have relevance values and are ranked in decreasing order of relevance for the categories. Out of 1832 cases which appear at the top of

the list,1727 such cases (94.27%) are of the correct category for method 1. For method 2 ,out of 1832 cases just 1777 are correct(97%). Similarly, the probability that the first two categories in the ranked output are correct is computed. Method 1 gave 517 out of 599 as correct (86.31%). Method 2 gave 1175 out of 1719 as correct(68.35%). These results for all the four cases of chapter 3 are presented in tables(5.1,5.2). It may be observed that case (iv), where there is high overlap of categories,shows higher values of P1 and P2 than does case(iii).

For the attribute number analysis method,the procedure using equation(5.7) was programmed in the same way as that for the relevance rating method. A slight change in initialisation of the attribute number matrix YKM(M,K), whose each element is the attribute number of the document to given category, was made. This matrix was initialised to -1 instead of zero. In the subsequent processing an entry in the attribute number matrix may become zero or positive and -1's remain as tags to indicate the elements that have not been set. This procedure did not disturb the generality of the method. It may be observed from the computation logic of equation (5.7) that, during processing, some of entries of YKM(M,K) can become zero. This makes it impossible to differentiate the entries that are made zero during processing and entries that are zero from initialisation. For example,for case(i) and the small data base document number 21 belongs to category 2 and 3. Term 54 having only one occurrence occurs in this document. This term will make some contribution to the

attribute number of the 21 st document to category 2 and 3. Since the term does not occur in documents of category 1 and 4,the respective entries in YKM matrix will be made equal to zero. Suppose that term 54 accidently occurs in document 41 which is not in any category. All the entries of YKM for this document will be made equal to zero. This will help to differentiate document 41 from others in which no term occurs and,hence their entries in YKM will be negative.

Another problem with the method of computation is the nature of processing of equation(5.7). For example, document 21 contains content term 69 of category 1,term 4 of category 2,and terms 6 and 54 of category 3. This document is relevant to category 2 and 3. Due to 4 occurrences of terms in it, its attribute number with respect to category 2 is reduced from .02 to .005. When ranked in decreasing order of attribute number its rank to category 2 will be markedly effected. Thus the more terms a document contains,the more its attribute number will be reduced and hence its ranking will be affected.

For the relevance rating method for case(i),2619 documents are found to be in some category. Out of this number 787 do not contain any content term. Out of the remaining 1832 documents,there are 1769 for which the correct category tops the list(96.56%). Similarly,the probability that the first two categories are correct is 73.37%. These results are tabulated in tables(5.1,5.2). From the tables it can be seen that this method compares very well with the relevance rating method 2.

The program also computes the classification efficiency in terms of precision and recall for each category. For this analysis the number of documents that are in category k, and whose attribute numbers satisfied threshhold level, are counted (RRT). Similarly, the total number of documents that satisfy threshold are also computed. For this method the threshhold has to be kept very low(starting from .00001). The threshhold for the two relevance rating methods was high (starting at .5). The results of the computations for the three methods for four cases are presented in figures (5.6 to 5.13). A careful observation of all these figures shows that the relevance rating method 2 appears to perform the best. Note the two cases (iii) and (iv),where in case (iii) the overlap of categories is very small and in case (iv) there is high overlap among categories. With a threshhold level of .5 for case (iv),for any category values of Precision .87 and Recall .78 are possible. Similarly for case (iii),Precision .75 and Recall .52 can be obtained. Thus high overlap among categories gives higher classification efficiency in terms of precision and recall.

The attribute number method accounts for very poor performance in this respect. The precision and recall for any threshhold are very low. Another noteworthy phenomenon in this method is the variation trend between precision and recall. As the threshhold level is increased,both precision and recall reduce in a linear manner. It is well-known[31] that precision and recall may not be inversely proportional to each other. For

the database under consideration this phenomenon is explained as follows:

The term occurrences are according to Zipf's law. The content terms of one category tend to occur in the documents of that category. Some of the terms may have term occurrences according to Zipf's law that are greater than the number of documents in the category. As a result the remaining occurrences will be randomly distributed among documents of other categories and documents which are not in any category. Such random occurrences in a document not belonging to any category will be of very small number. The equation (5.7) will give a high attribute number to this document having only one or two occurrences. The same is true for all documents not in any category. But other documents belonging to some categories will have more term occurrences and according to equation (5.7) will have a low attribute number. As a consequence,as the threshhold is increased the number of relevant documents decreases, and at the same time the total documents retrieved to satisfy the threshhold also decreases. But the decrease in the number of irrelevant documents in the total retrieved number is very low. As a result,more irrelevant documents not in any category,but which have attr ibute numbers as explained above, are retrieved. This explains the low rate of decrease in the number of total retrieved documents with increase of threshhold level. So it is not surprising to see a precision and recall relationship different from what might at first be expected.

In general, in design of experiments on an information retrieval system one is mainly concerned with the effects of assigning different values to variables and parameters in order to characterise the classification process. Since the number of variables involved in a retrieval process is quite high, and examination of the effect of all possible combinations of the values of these variables appears to be impractical, one is therefore constrained to study the effects of some base configuration of all the variables. The base configuration in the present study is determined mainly as a result of practical experience with information retrieval systems. We have selected some values of certain parameters because they are consistent with practical considerations. The base configuration is composed of some standard values of p_c, p_n, and Zipf's distribution of terms in the data base. The classification rating of a content term I to some category k is another parameter whose value can be changed and its effect on classification examined. This does not prevent us from making changes in base configuration variables. The base configuration may be modified by assigning other values to the variables. The number of possible experiments becomes very large, and representation of experimental results also poses problems. Therefore, in the present study, the base configuration is not changed, but the effect of changes of values of R_{ik}, and variations in subject span of categories, is examined.

R_{ik}, is the ratio of two relative frequencies f_{ik} and M_i/M, and this parameter may be changed by changing the value of f_{ik} which is the ratio between the number of documents of the k th category that contain the I th term and the number of documents in the k th category. If the I th term is a content term for the k th category its probable number of occurrences in a document of the k th category is governed by the value of the category weights $W_i(I_1, I_2, I_3, I_4)$. A high value of category weight determines a high probability of occurrence of the term I in a document of the k th category. Thus changes in the values of f_{ik} reflect changes in the values of R_{ik}. In order to effect such changes the category weights $W_i(I_1, I_2, I_3, I_4)$ were adjusted to values of 0.85 and 0.15. The results of the parametric change are presented in tables(5.3,5.4). A comparison with tables(5.1,5.2) shows that values of P1 and P2 are lower than with category weights of values 0.95 and 0.05. The classification efficiency in terms of Precision and Recall for the three methods and for the four cases of probabilities was also computed, and results are plotted graphically in figures(5.14 to 5.21). The same trend among the three methods may be noted. The prediction of relevance by method 2 is the most superior. For all cases of probabilities, a low value of category weights leads to a low classification performance of the system.

It appears that the classification process is sensitive to changes in the values of R_{ik}. This is explained as follows: Consider an I th term that is a content term for the k th

category. Reducing R_{ik} causes this I th term to occur with reduced probability in documents of the k th category. This is equivalent to saying that the term's tendency to occur in documents of other categories is increased. As a result, this term's power of discriminating documents between categories is reduced. In other words the statistical distribution of the term I does not reach a peak value for the documents of the k th category; rather it tends to have a somewhat flat variation as a function of k. It is, of course, desirable that content terms should show a peak distribution in content term-category matrix elements for the documents of category for which they are supposed to be content terms. In fact, decreasing the value of R_{ik} results in weak statistical correlation between documents of the k th category and the I th term, and is therefore responsible for low values of classification efficiency.

TABLE 5.1: Comparisons of automatic document classifications procedures.

M = total documents

N_K = documents belonging to some category or category combinations

N_0 = out of N_K, documents which do not contain any content term

$W_i(I_1, I_2, I_3, I_4) = .95, .05$

Case (I): M = 5000

N_K = 2619

N_0 = 787

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{1727}{1832} = 94.27$	$\frac{1777}{1832} = 97.00$	$\frac{1769}{1832} = 96.56$
P2 (%)	$\frac{517}{599} = 86.31$	$\frac{1175}{1719} = 68.35$	$\frac{1182}{1611} = 73.37$

Case (II): M = 5000

N_K = 3846

N_0 = 1554

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{2165}{2270} = 95.37$	$\frac{2193}{2292} = 95.68$	$\frac{2191}{2292} = 95.59$
P2 (%)	$\frac{425}{530} = 80.19$	$\frac{931}{2033} = 45.79$	$\frac{935}{1857} = 50.35$

TABLE 5.2: Comparisons of automatic document classification Procedures.

Case (IV): $M = 5000$

$N_K = 2977$

$N_O = 887$

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{2066}{2088} = 98.95$	$\frac{2077}{2090} = 99.38$	$\frac{2077}{2090} = 99.38$
P2 (%)	$\frac{863}{869} = 99.31$	$\frac{1947}{2067} = 94.19$	$\frac{1947}{2053} = 94.84$

Case (III): $M = 5000$

$N_K = 4946$

$N_O = 2277$

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{2572}{2650} = 97.06$	$\frac{2597}{2669} = 97.30$	$\frac{2594}{2669} = 97.19$
P2 (%)	$\frac{535}{589} = 90.83$	$\frac{1819}{2605} = 69.83$	$\frac{1812}{2582} = 70.8$

TABLE 5.3: Comparison of automatic document classification procedures.

M = total documents

N_K = documents belonging to some category or category combination

N_O = out of N_K, documents containing no content term

$W_i(I_1, I_2, I_3, I_4) = 0.85, 0.15$

Case (I): $M = 5000$

$N_K = 2619$

$N_O = 856$

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{1491}{1682} = 88.64$	$\frac{1647}{1763} = 93.42$	$\frac{1653}{1763} = 93.76$
P2 (%)	$\frac{373}{535} = 69.72$	$\frac{1088}{1626} = 66.91$	$\frac{1697}{1536} = 71.42$

Case (II): $M = 5000$

$N_K = 3846$

$N_O = 1603$

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{1823}{2156} = 84.55$	$\frac{1951}{2243} = 86.98$	$\frac{1946}{2243} = 86.76$
P2 (%)	$\frac{329}{571} = 57.62$	$\frac{824}{1987} = 41.47$	$\frac{827}{1799} = 45.97$

TABLE 5.4: Comparisons of automatic document classification procedures.

Case (III): $M = 5000$

$N_K = 4946$

$N_O = 2314$

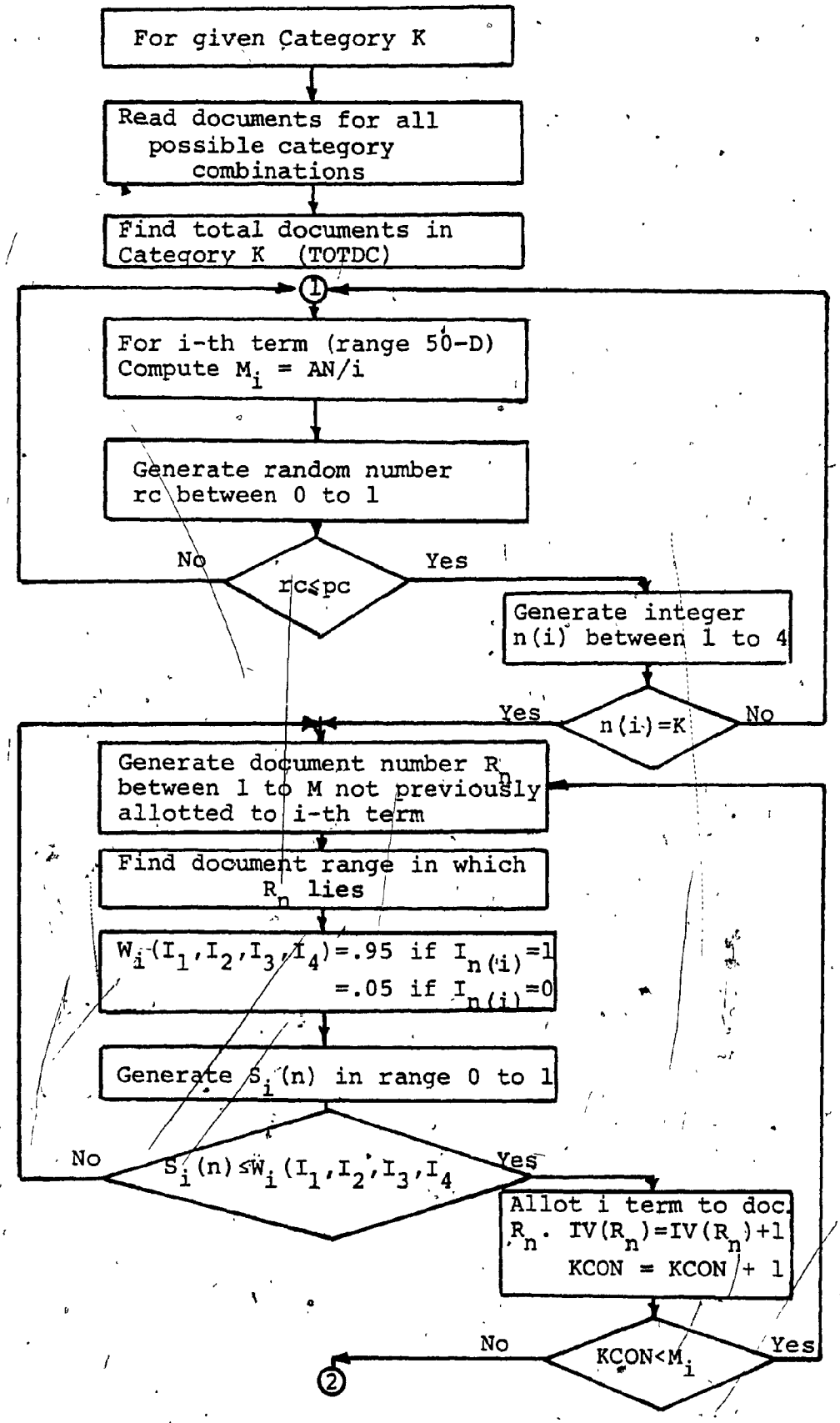
	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{2313}{2561} = 90.32$	$\frac{2426}{2632} = 92.17$	$\frac{2426}{2632} = 92.17$
P2 (%)	$\frac{451}{631} = 71.47$	$\frac{1675}{2563} = 65.35$	$\frac{1668}{2531} = 65.90$

Case (IV): $M = 5000$

$N_K = 2977$

$N_O = 974$

	Relevance Rating Method 1	Relevance Rating Method 2	Attribute Number Method
P1 (%)	$\frac{1949}{1993} = 97.74$	$\frac{1981}{2003} = 98.90$	$\frac{1981}{2003} = 98.90$
P2 (%)	$\frac{736}{772} = 95.34$	$\frac{1852}{1980} = 93.54$	$\frac{1852}{1960} = 94.49$


For given Category K
Read documents for all possible category combinations
Find total documents in Category K (TOTDC)
1
For i-th term (range 50-D) Compute M_i = AN/i
Generate random number rc between 0 to 1
No
rc≤pc
Yes
Generate integer n(i) between 1 to 4
Yes
n(i)=K
No
Generate document number R_n between 1 to M not previously allotted to i-th term
Find document range in which R_n lies
$W_i(I_1,I_2,I_3,I_4)$=.95 if $I_{n(i)}$=1
=.05 if $I_{n(i)}$=0
Generate $S_i(n)$ in range 0 to 1
No
$S_i(n) \leq W_i(I_1,I_2,I_3,I_4)$
Yes
Allot i term to doc. R_n. $IV(R_n)=IV(R_n)+1$
KCON = KCON + 1
No
KCON<M_i
Yes
2

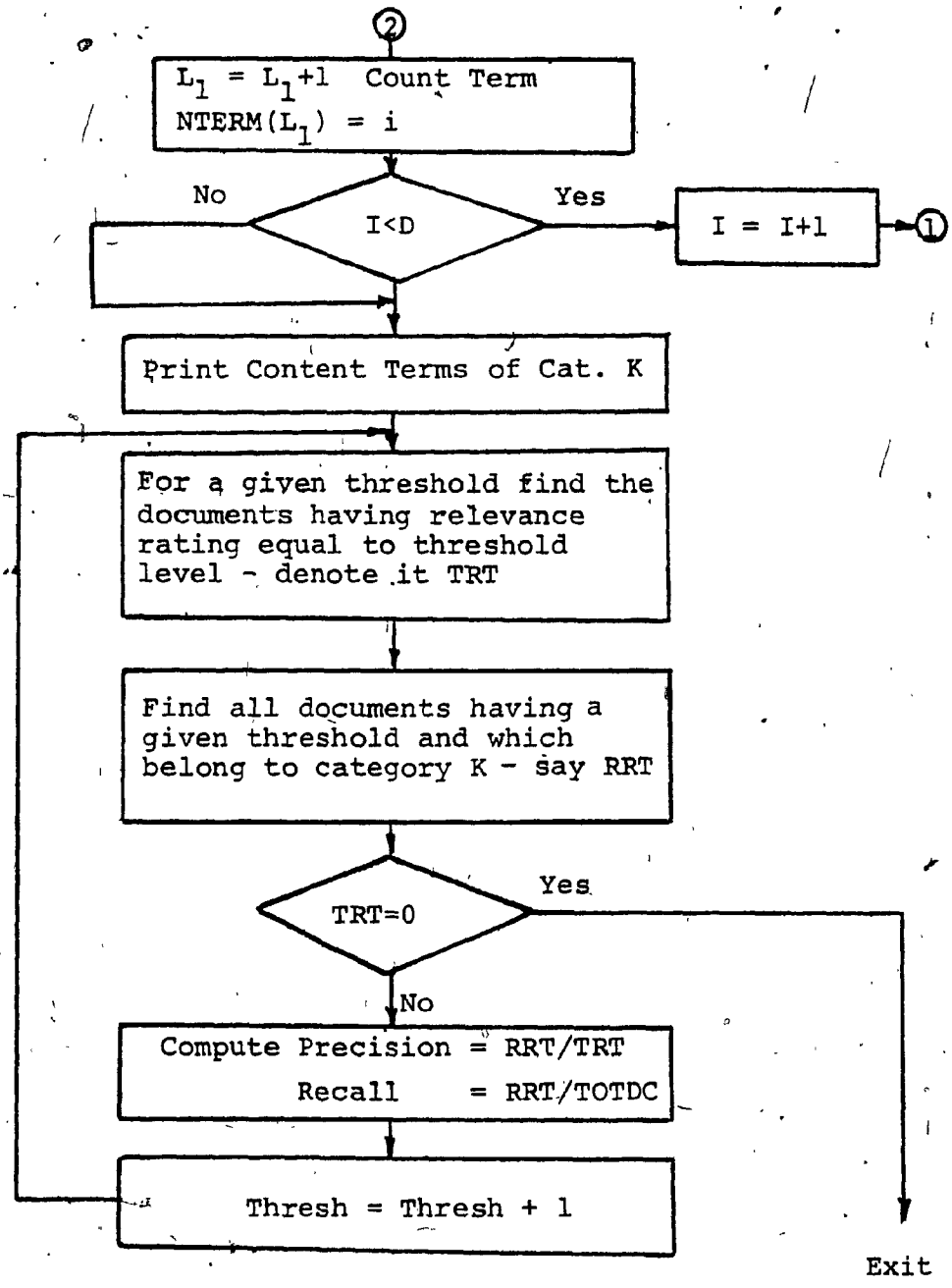


Fig. 5.1 Schematic flowchart for the evaluation of classification efficiency for different automatic classification procedures.

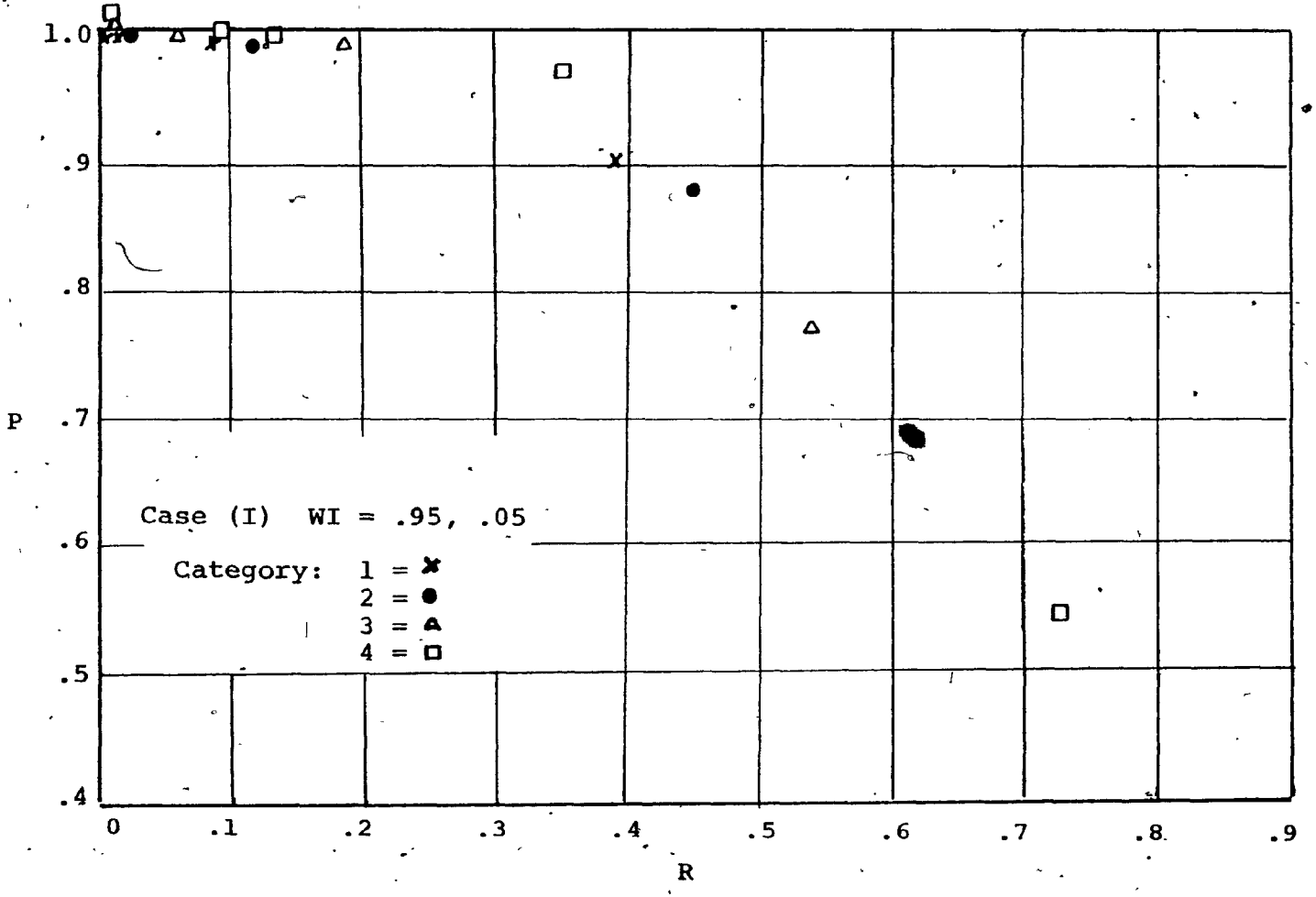


Fig. 5.2 Classification efficiency in terms of precision and recall.

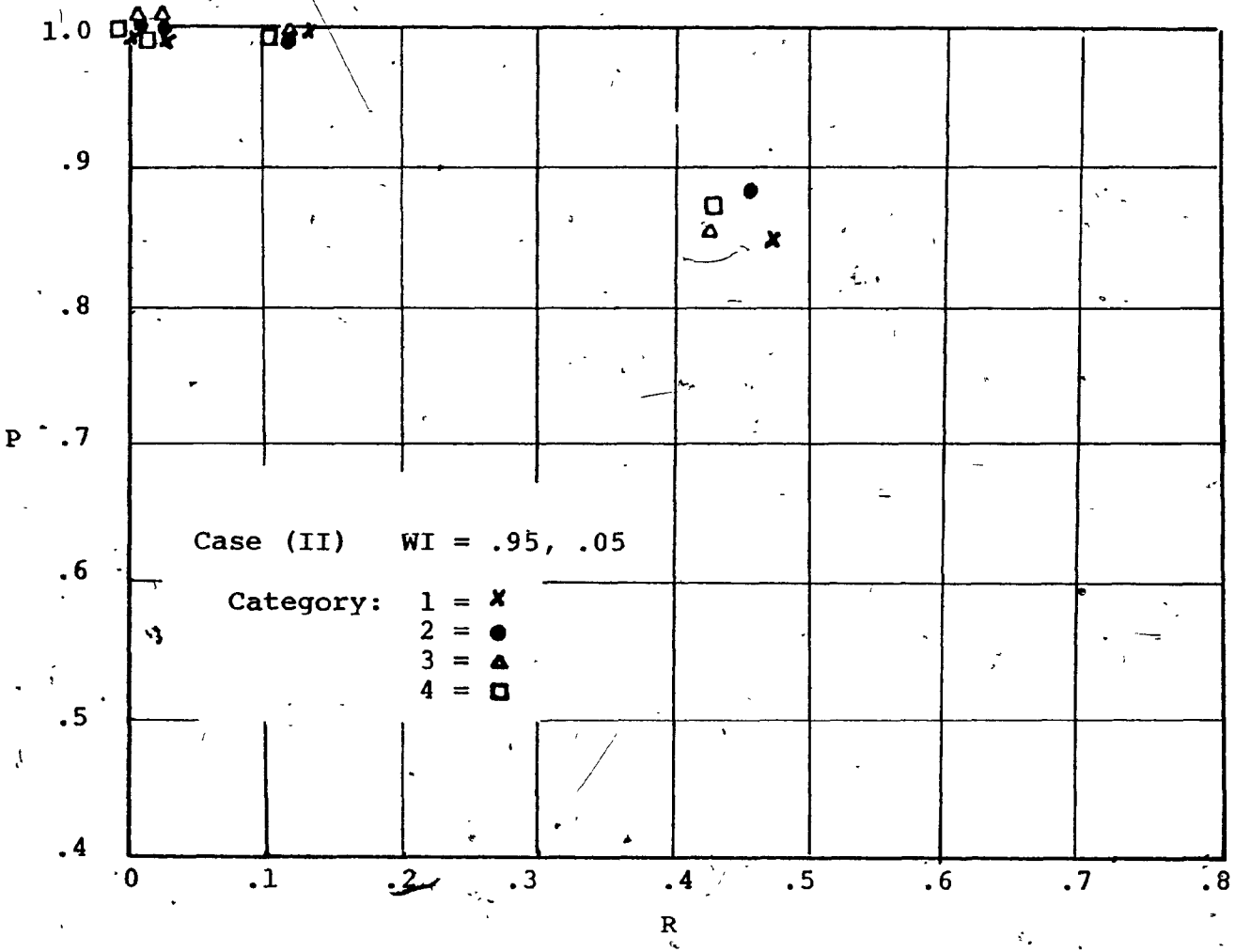


Fig. 5.3 Classification efficiency in terms of precision and recall.

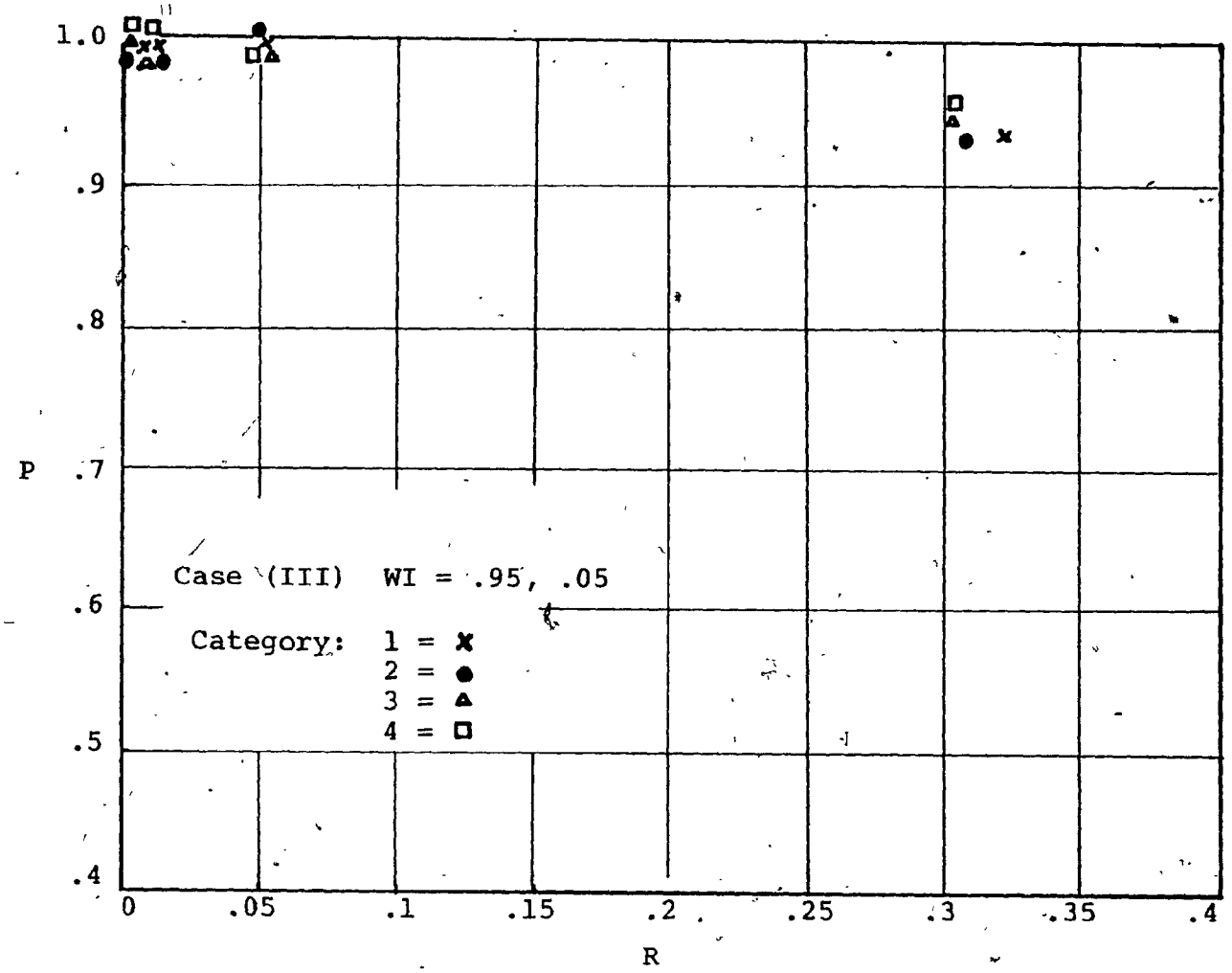


Fig. 5.4 Classification efficiency in terms of precision and recall.

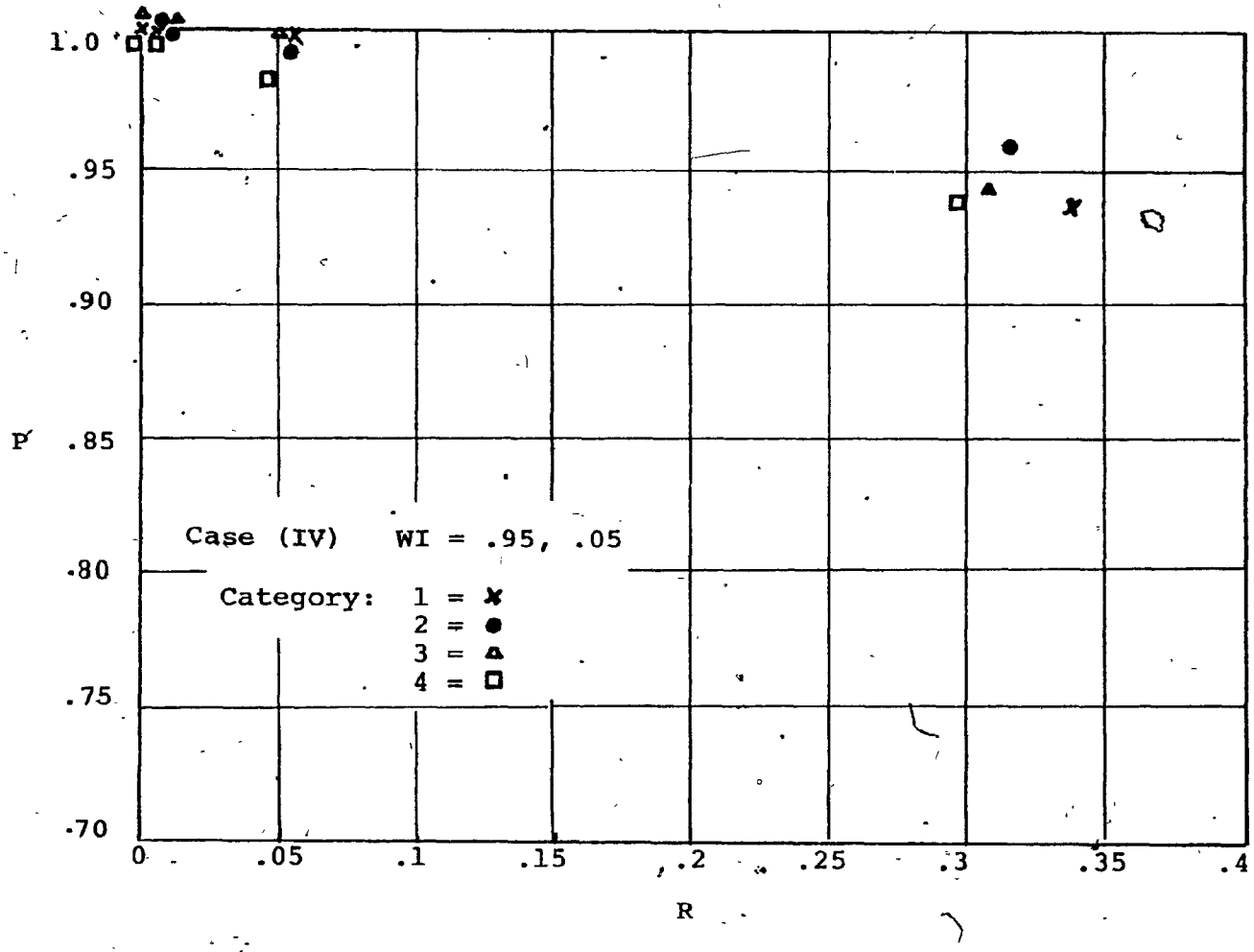
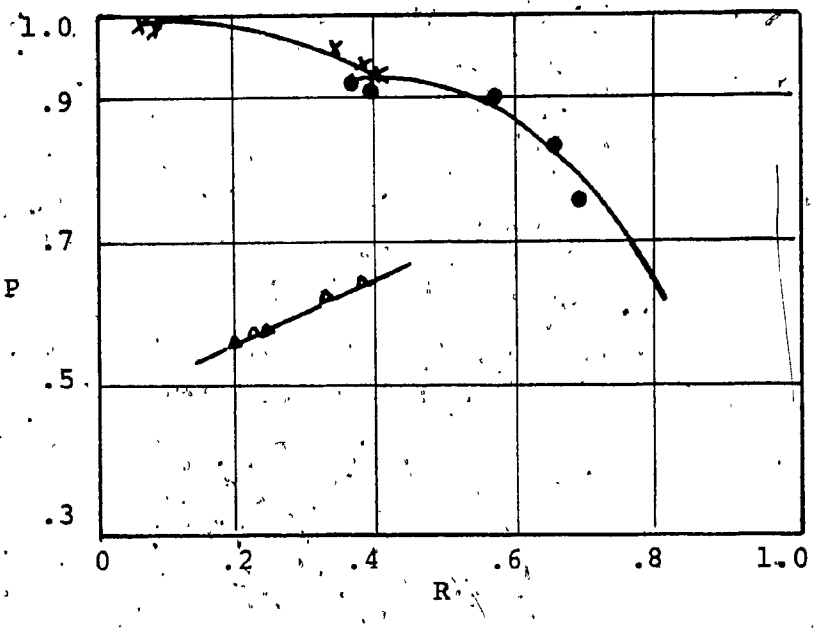


Fig. 5.5 Classification efficiency in terms of precision and recall.

a)



x Relevance rating 1 CASE (I)
● Relevance rating 2 WI = .95, .05
▲ Attribute number

b)

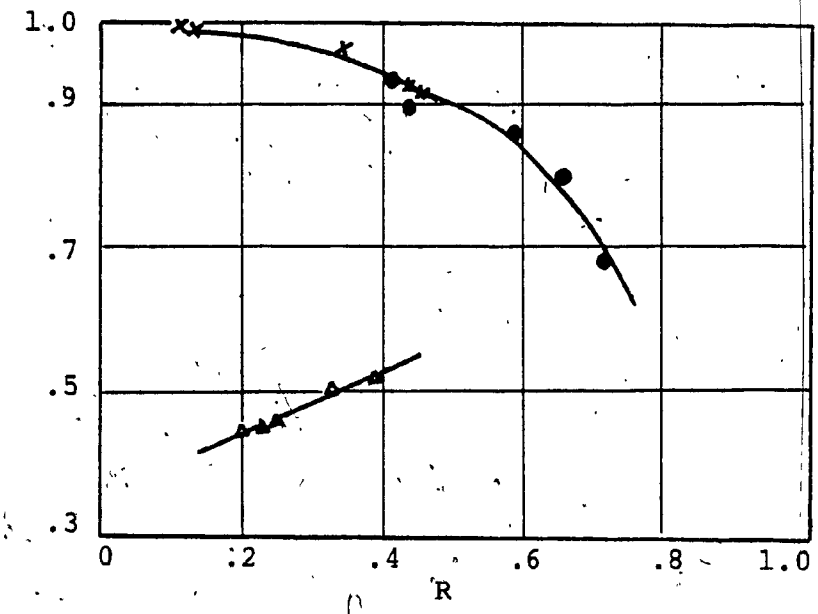


Fig. 5.6 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2.

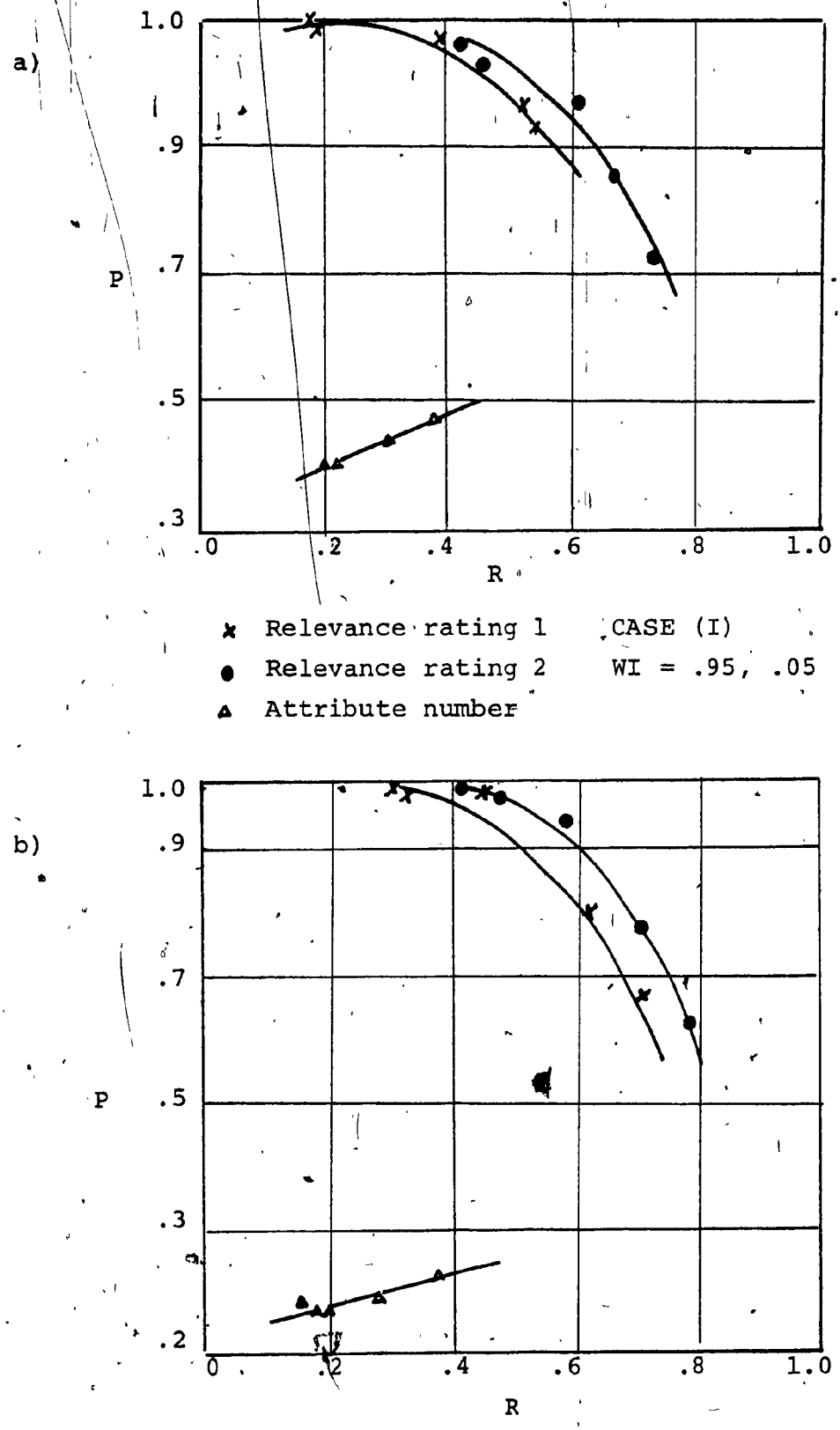
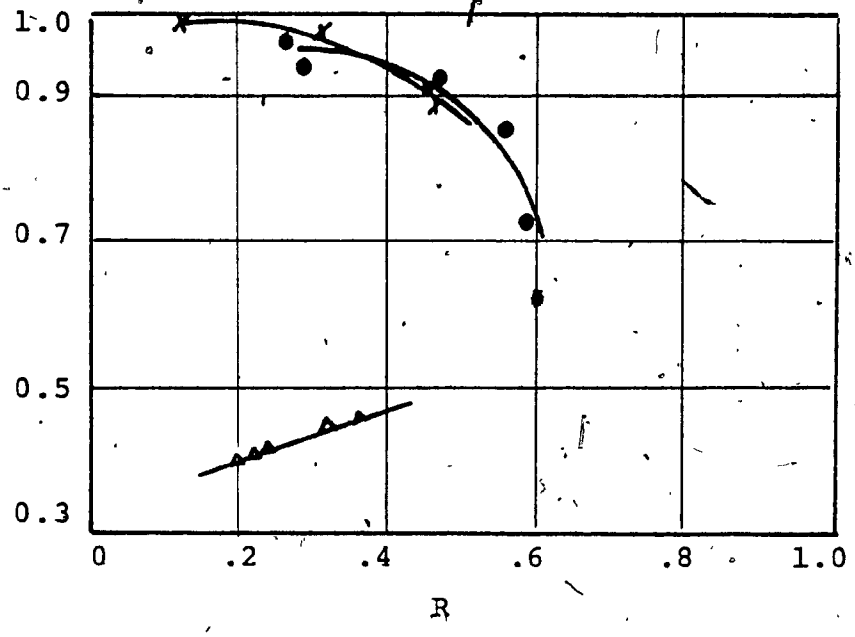


Fig. 5.7 Classification efficiency for three automatic document classification procedures: a) category 3, (b) category 4.

a)



✘ Relevance rating 1 CASE (II)
● Relevance rating 2 WI = 0.95, 0.05
▲ Attribute number

b)

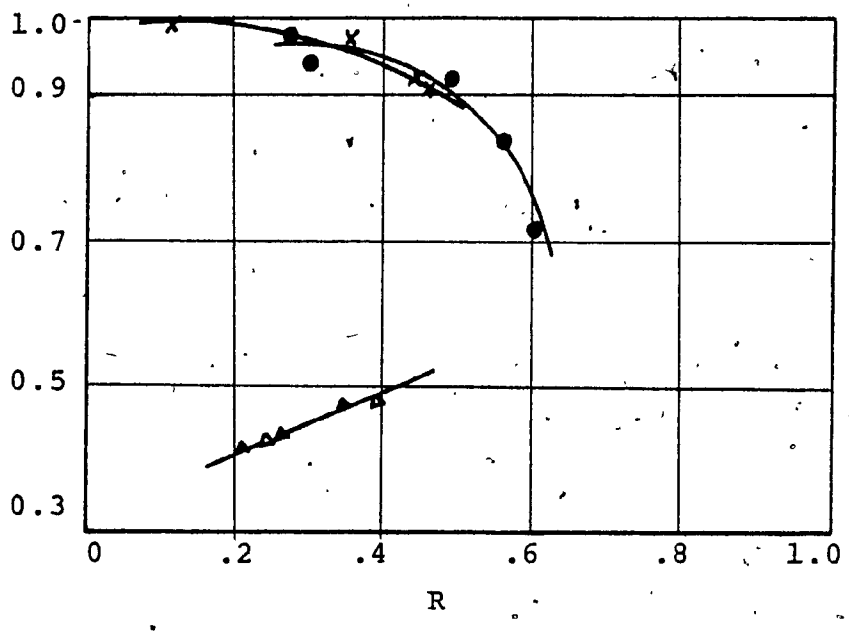
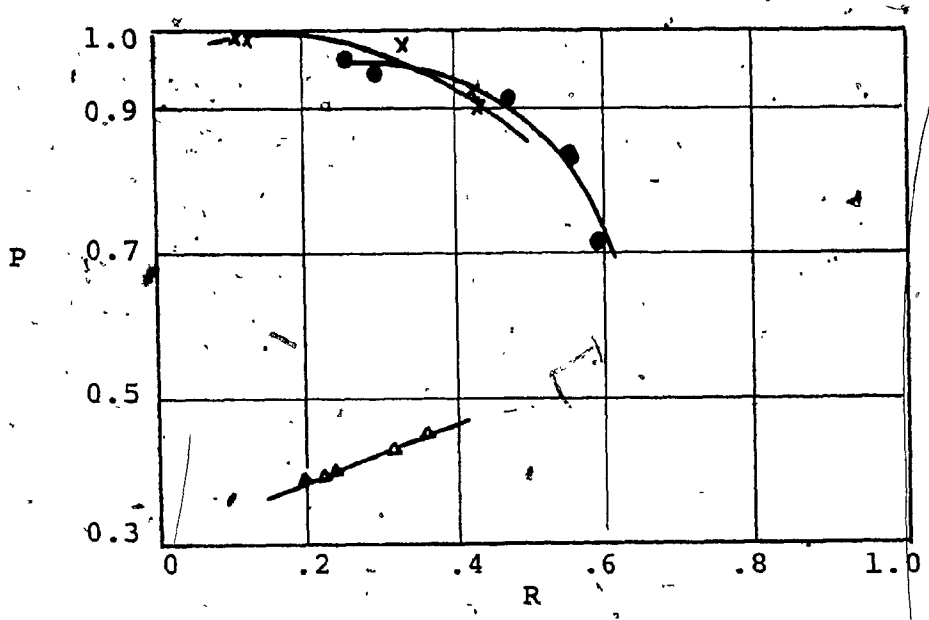


Fig. 5.8 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2.

a)



× Relevance rating 1 CASE(II)
● Relevance rating 2 WI = 0.95, 0.05
▲ Attribute number

b)

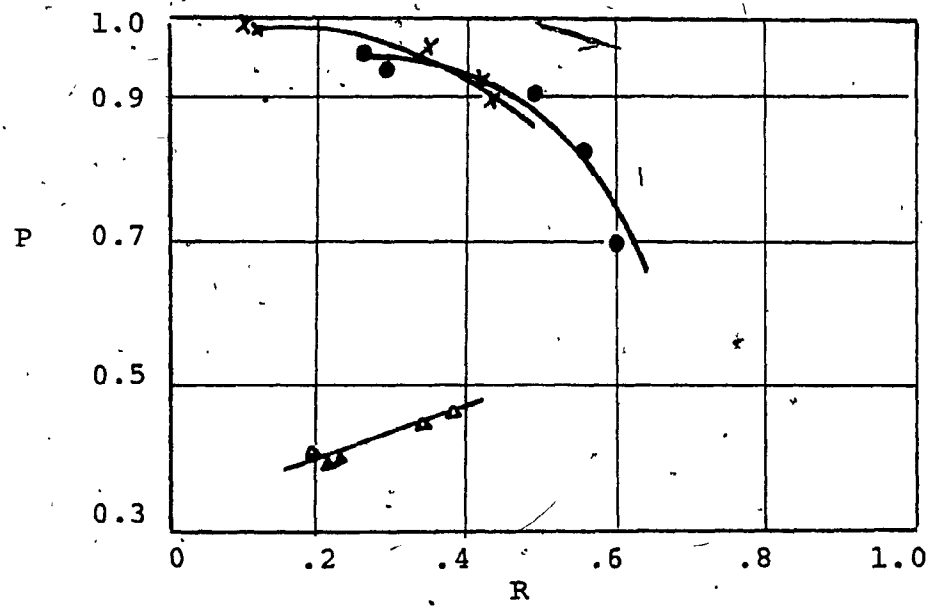
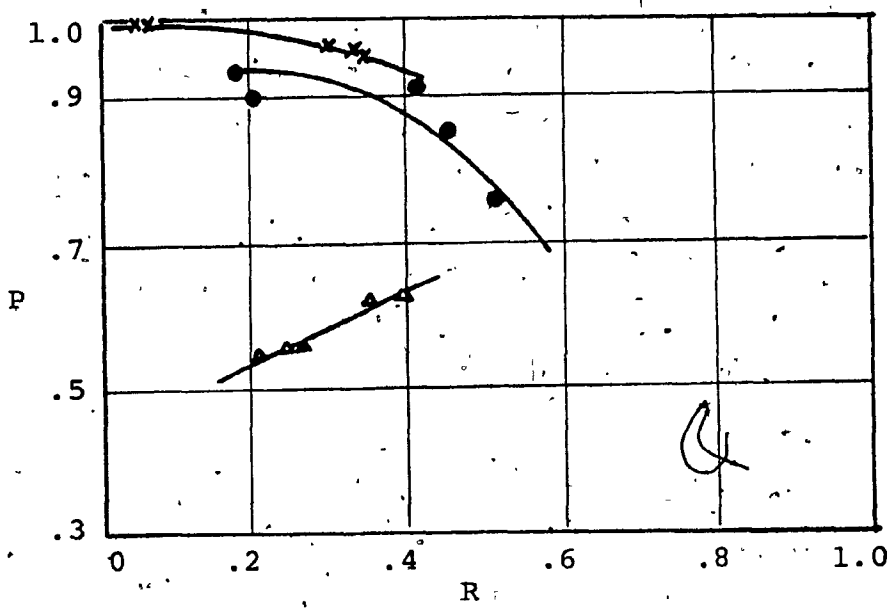


Fig. 5.9 Classification efficiency for three automatic document classification procedures: a) category 3, (b) category 4.

a)



× Relevance rating 1 CASE (III)
● Relevance rating 2 WI = .95, .05
△ Attribute number

b)

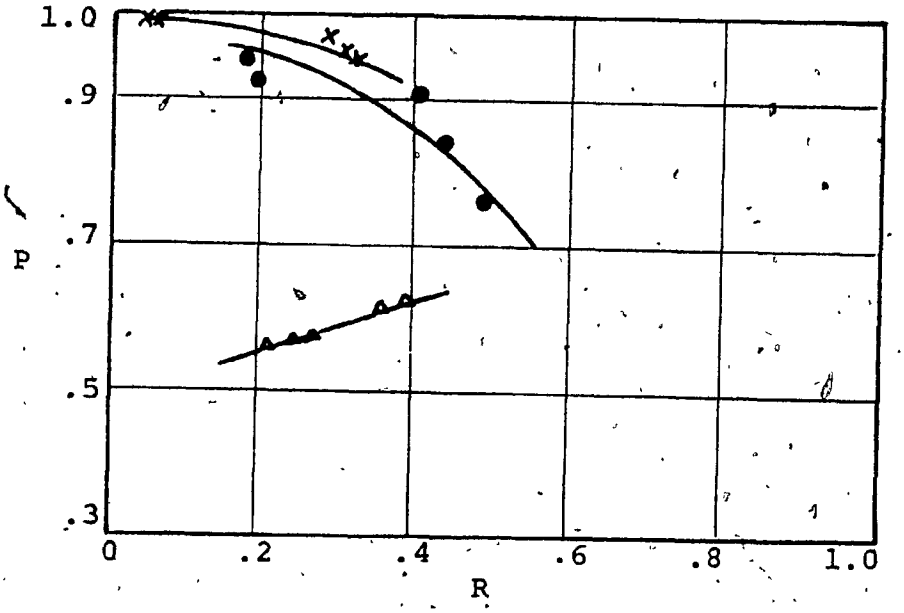
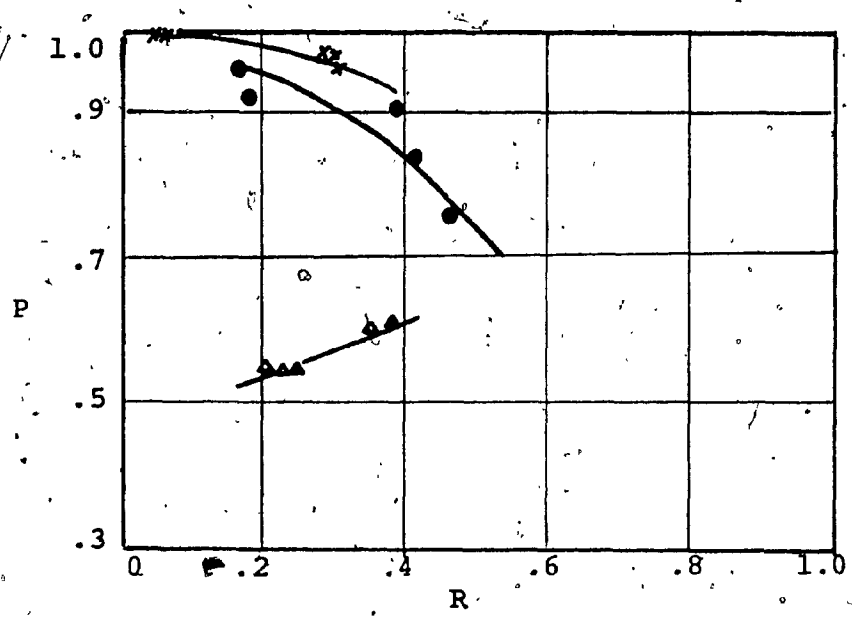


Fig. 5.10 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2.

a)



× Relevance rating 1 CASE (III)
● Relevance rating 2 WI = .95, .05
▲ Attribute number

b)

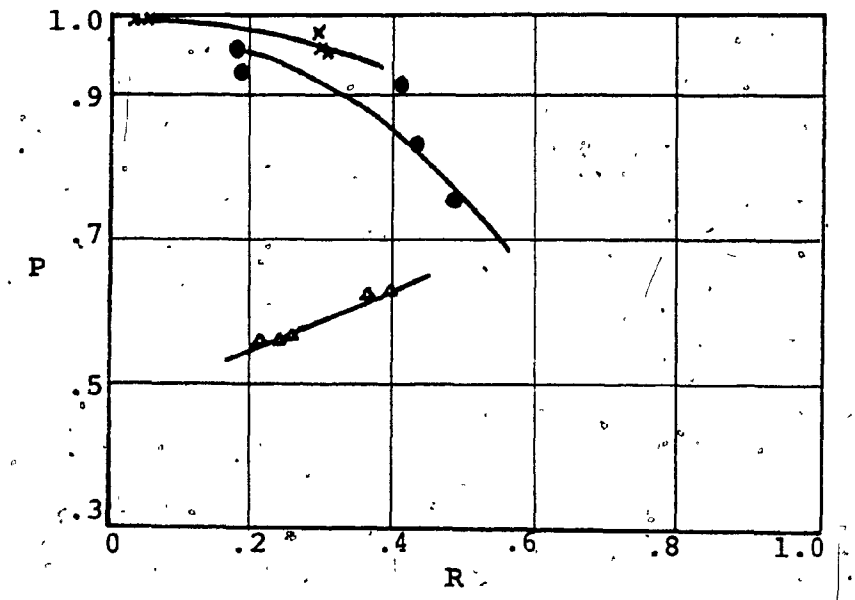
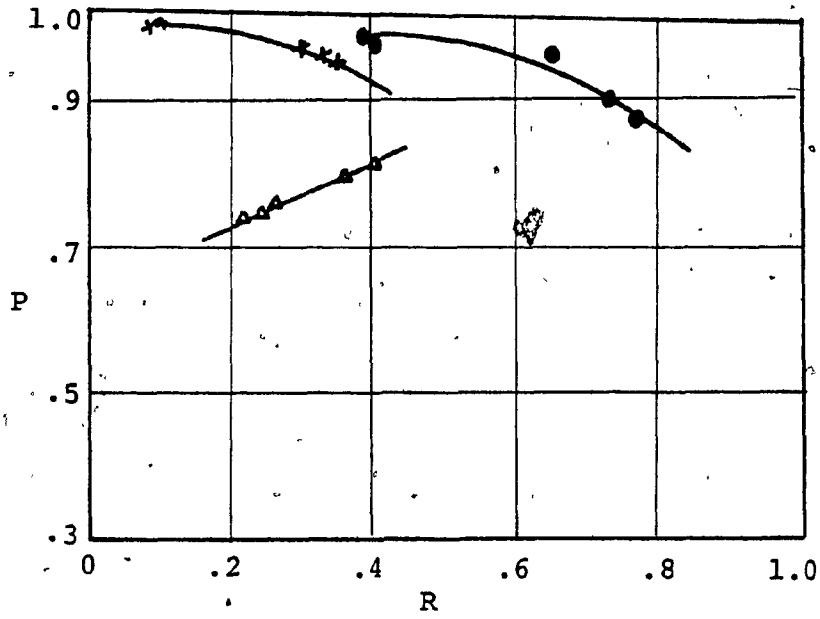


Fig. 5.11 Classification efficiency for three automatic document classification procedures: a) category 3, (b) category 4.

a)



✗ Relevance rating 1 CASE (IV)

● Relevance rating 2 WI = .95, .05

▲ Attribute number

b)

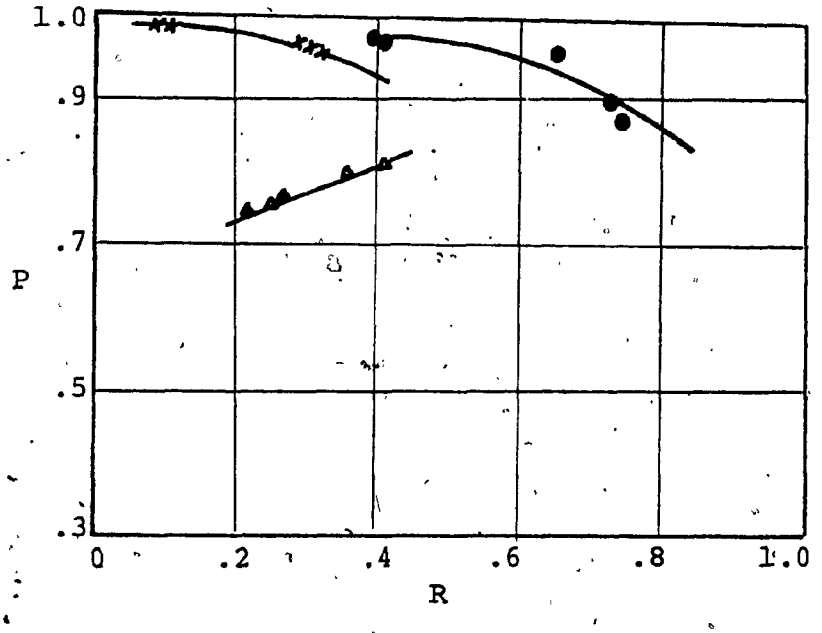
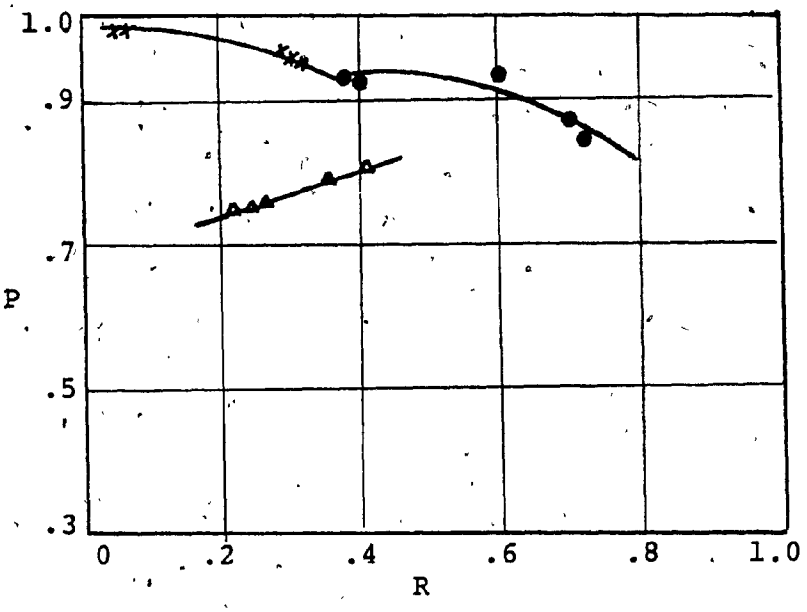


Fig. 5.12 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2.

a)



× Relevance rating 1 CASE (IV)

● Relevance rating 2 WI = .95, .05

▲ Attribute number

b)

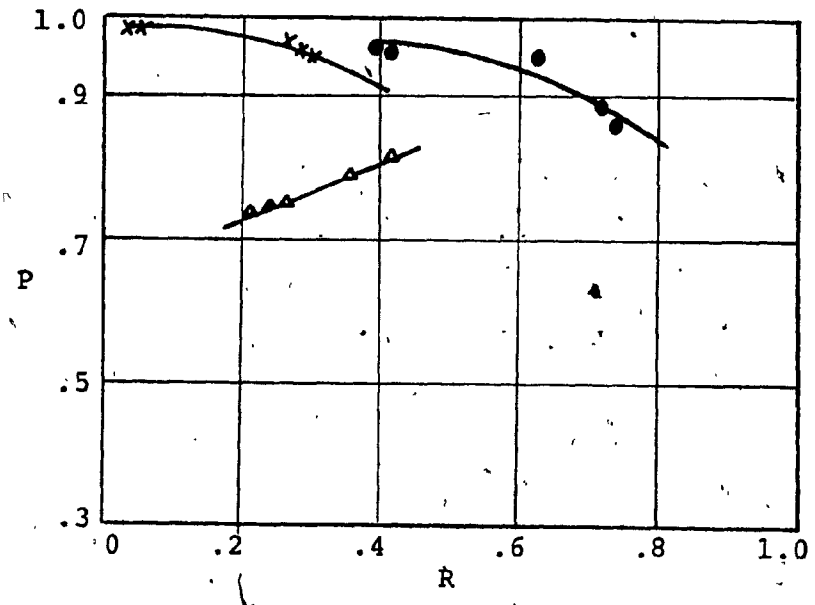


Fig. 5.13 Classification efficiency for three automatic document classification procedures: a) category 3, (b) category 4.

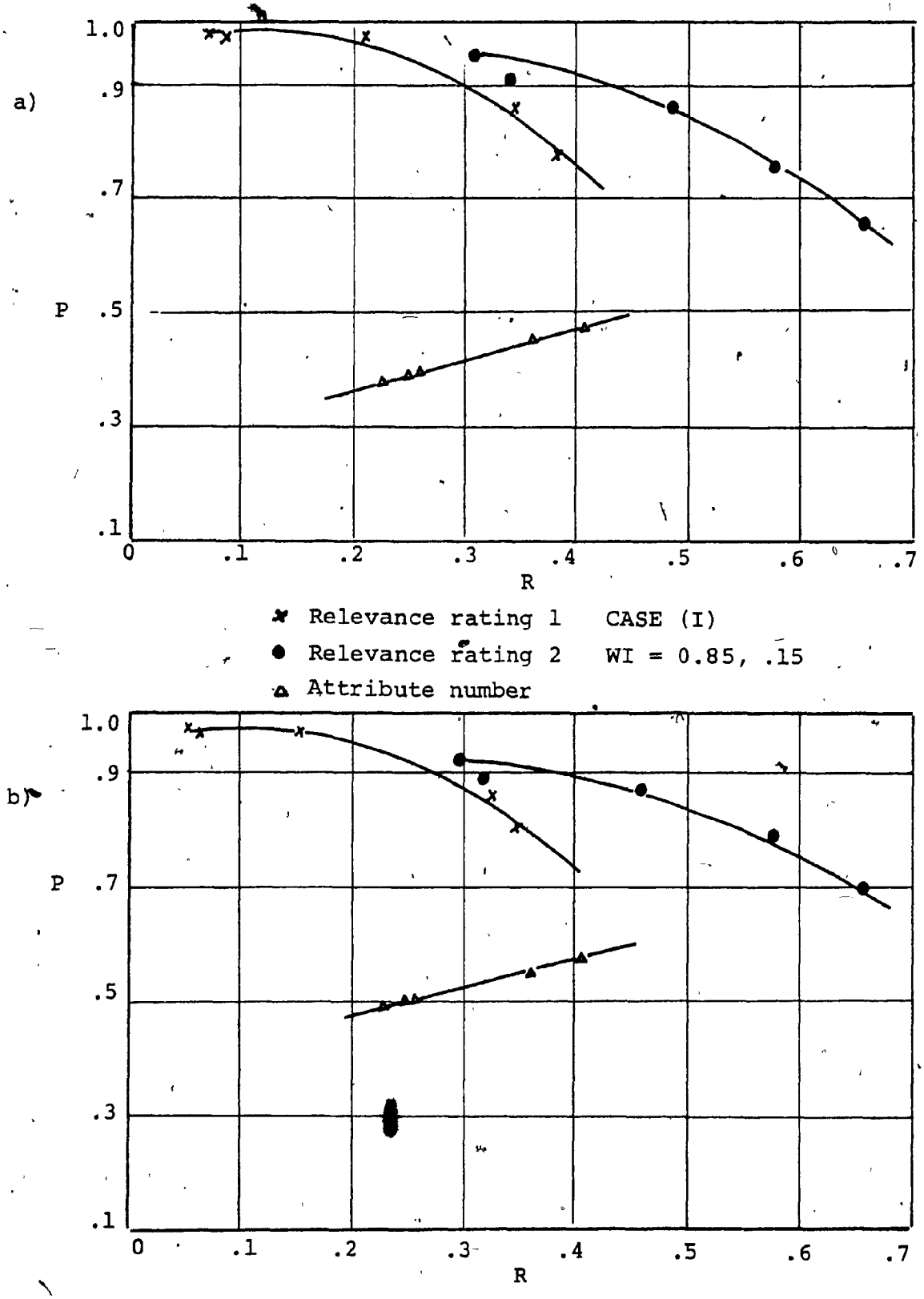
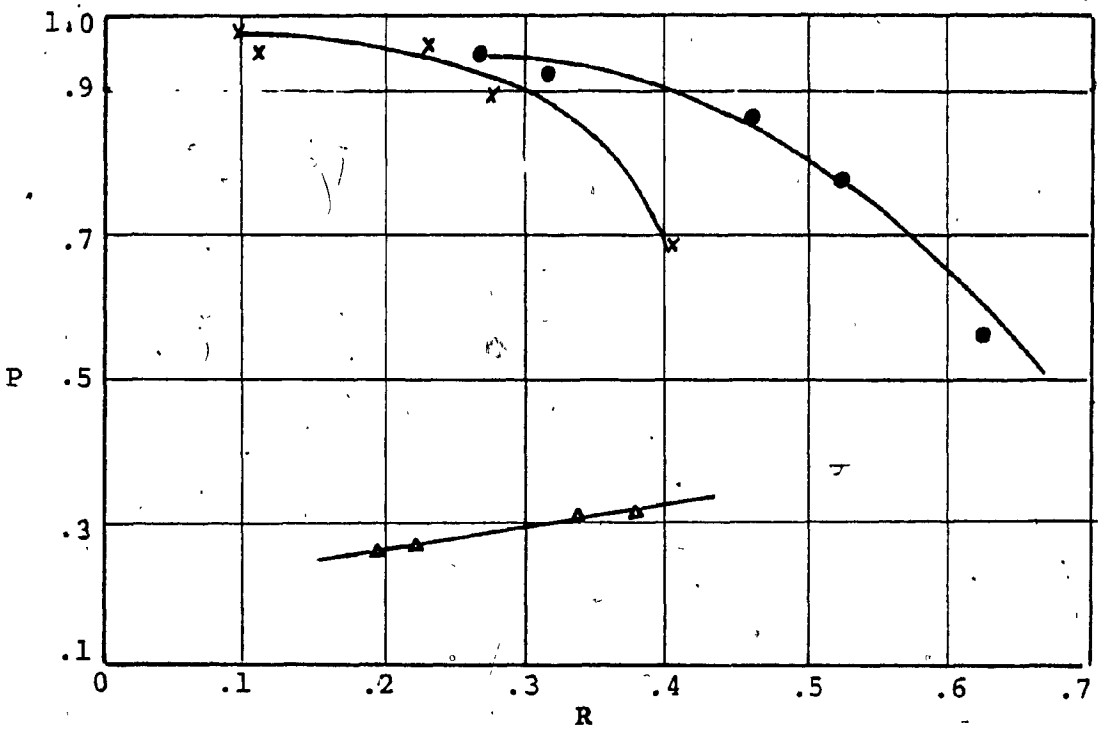


Fig. 5.14 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2.

a)



× Relevance rating 1 CASE (I)
● Relevance rating 2 WI = 0.85, .15
▲ Attribute number

b)

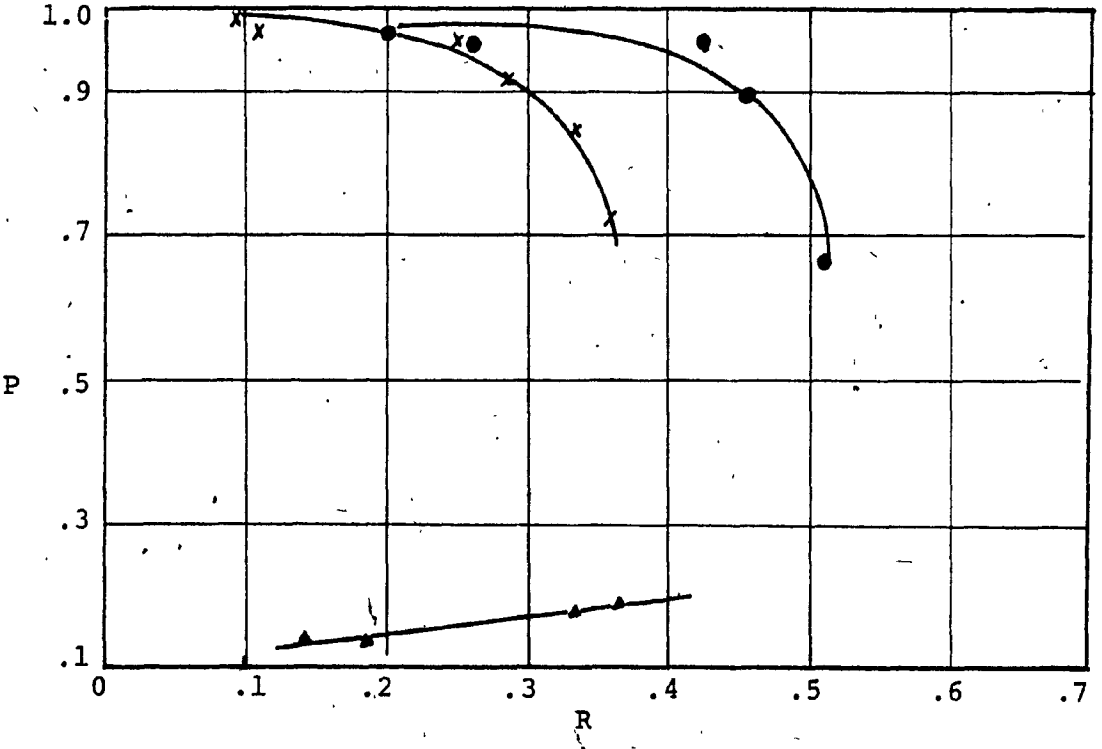
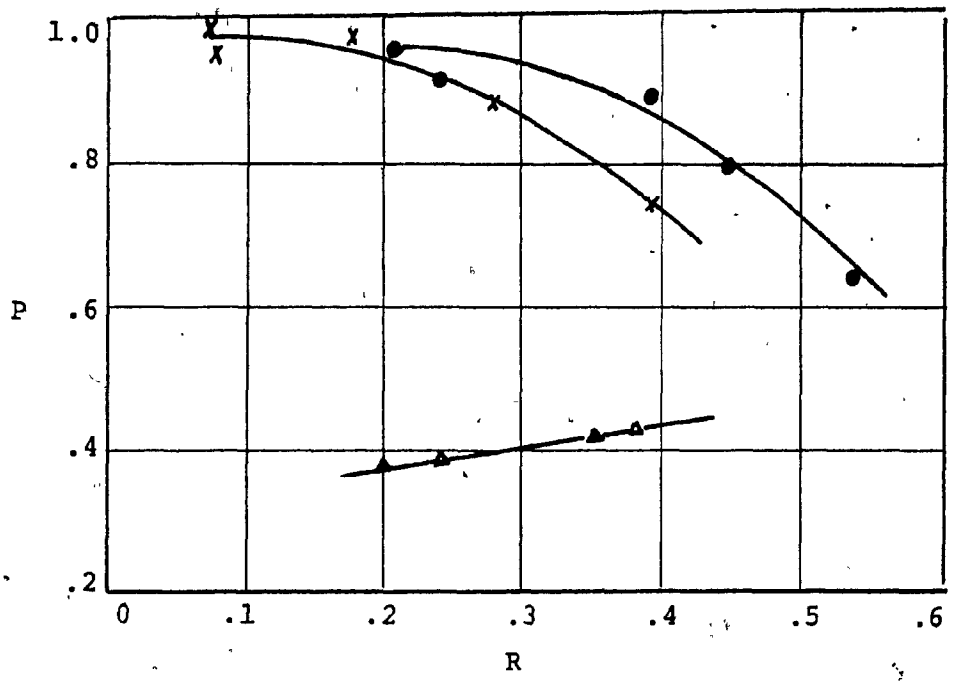


Fig. 5.15 Classification efficiency for three automatic document classification procedures: a) category 3, (b) category 4.

a)



✕ Relevance rating 1 CASE (II)
● Relevance rating 2 WI = .85, .15
△ Attribute number

b)

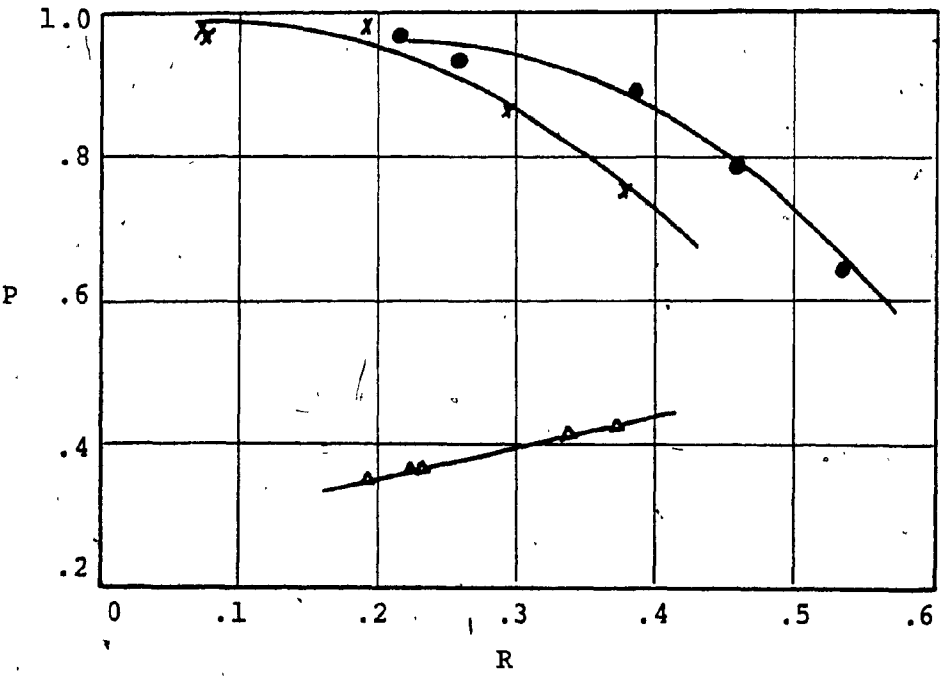
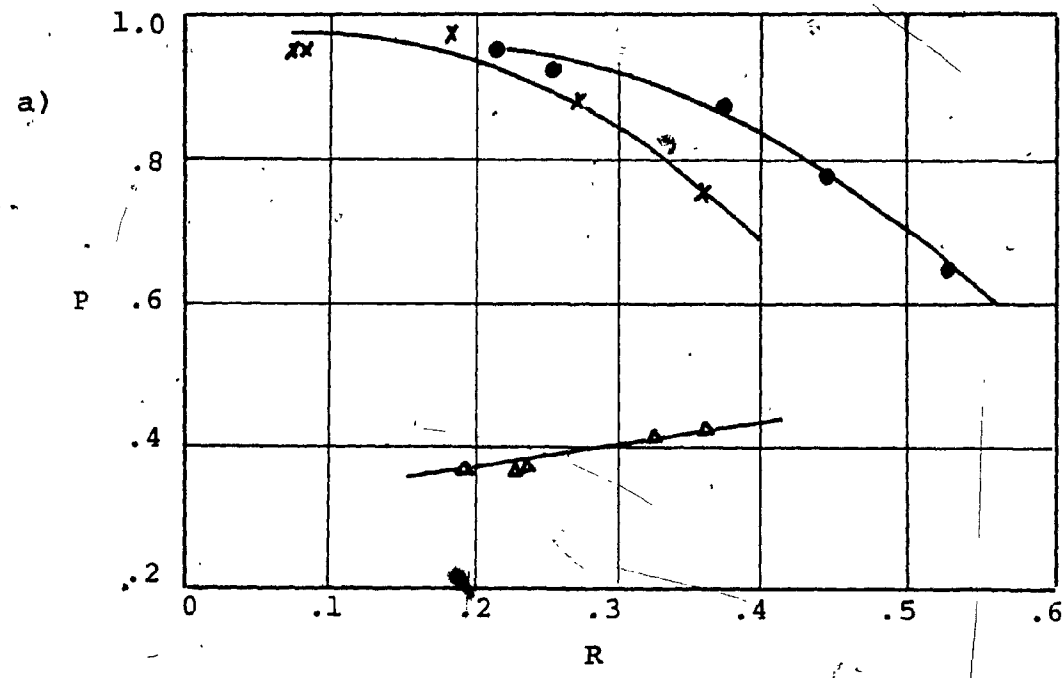


Fig. 5.16 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2



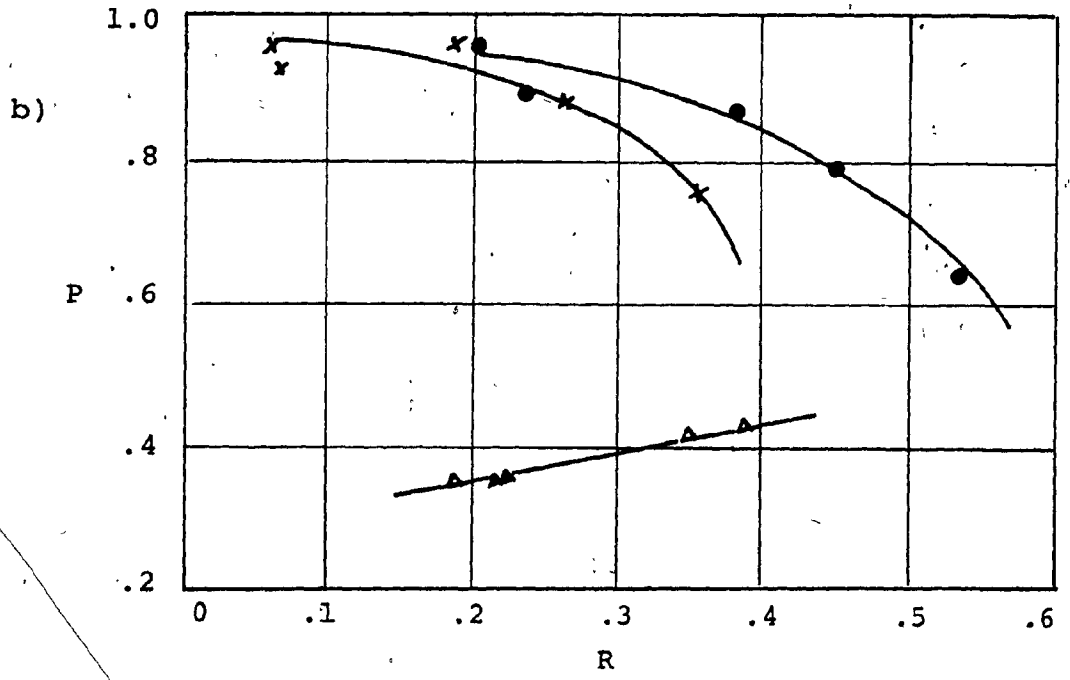
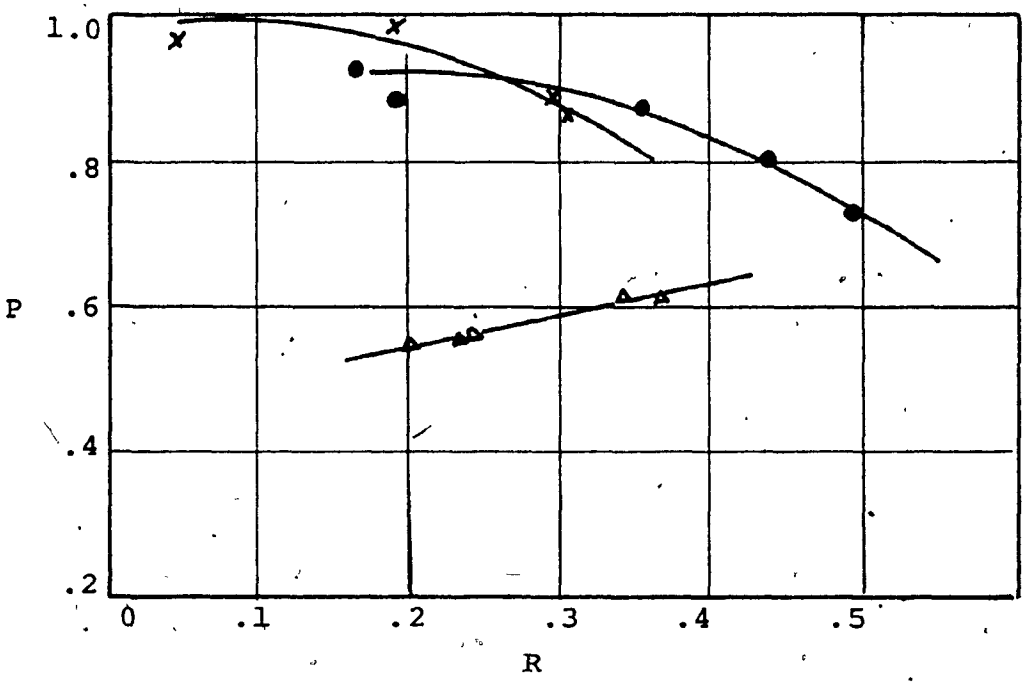


Fig. 5.17 Classification efficiency for three automatic document classification procedures: a) category 3, (b) category 4.

a)



✗ Relevance rating 1 CASE(III)
● Relevance rating 2 WI = .85, .15
△ Attribute number

b)

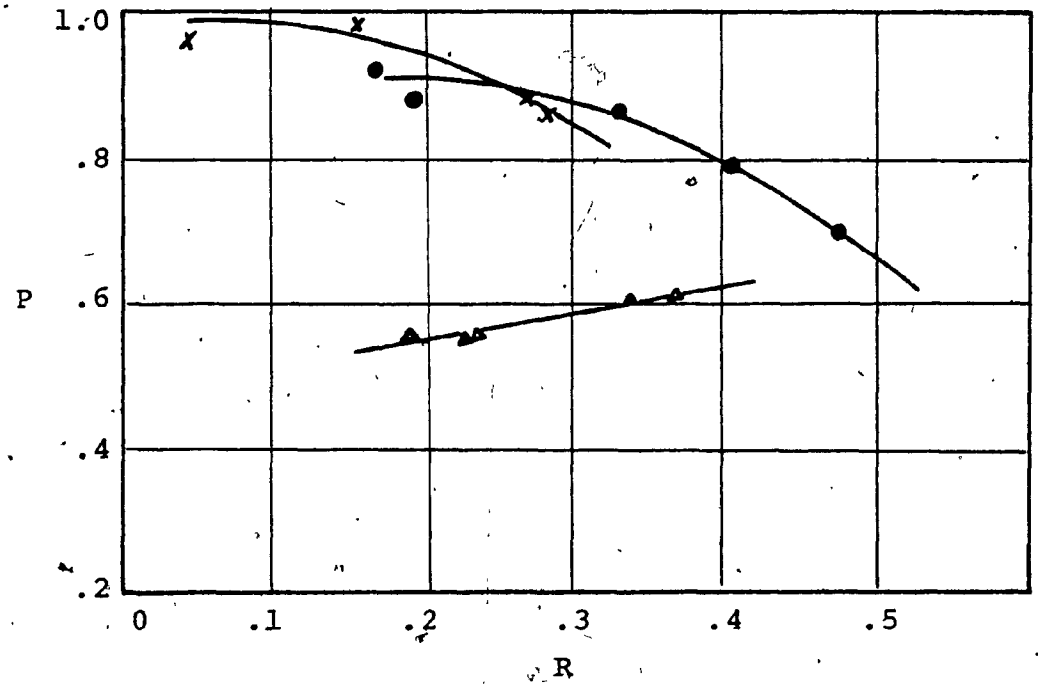
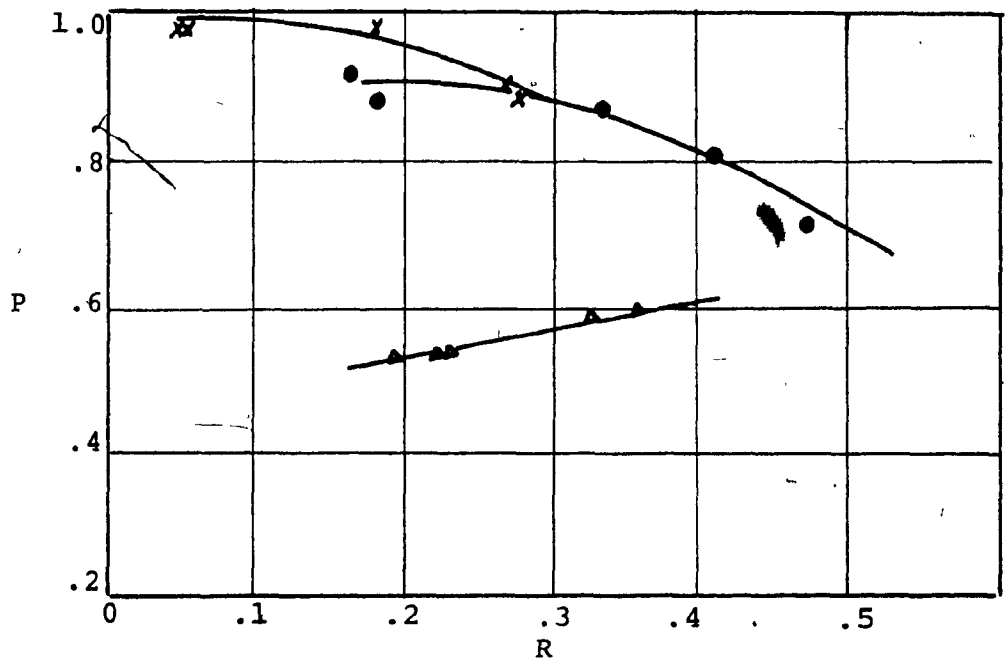


Fig. 5.18 Classification efficiency for three automatic document classification procedures: a) category 1, (b) category 2.

a)



x Relevance rating 1 CASE (III)
● Relevance rating 2 WI = .85, .15
△ Attribute number

b)

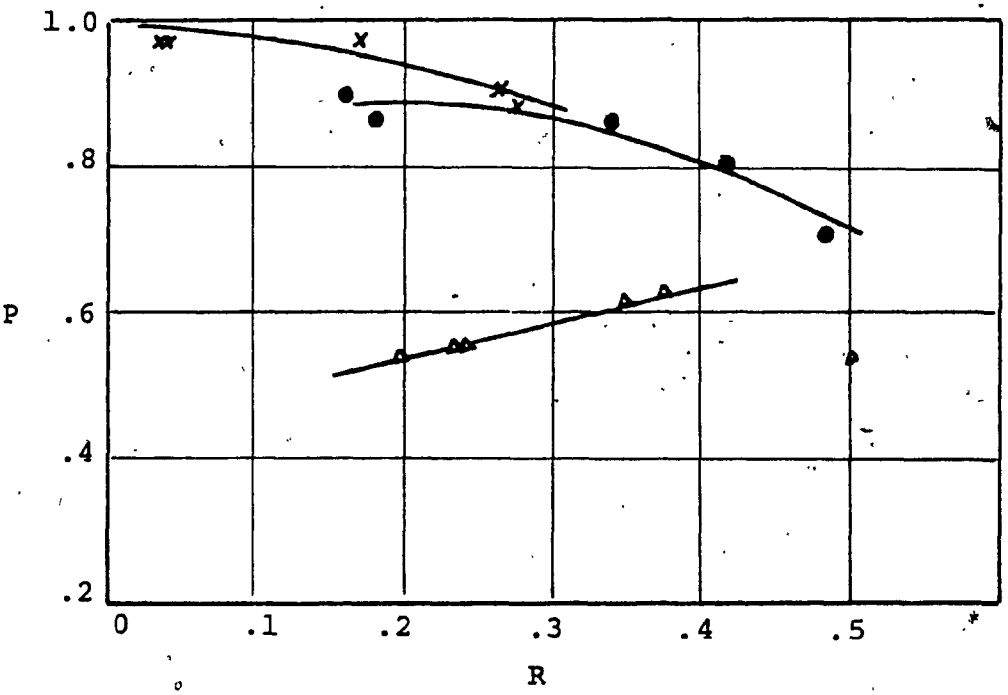
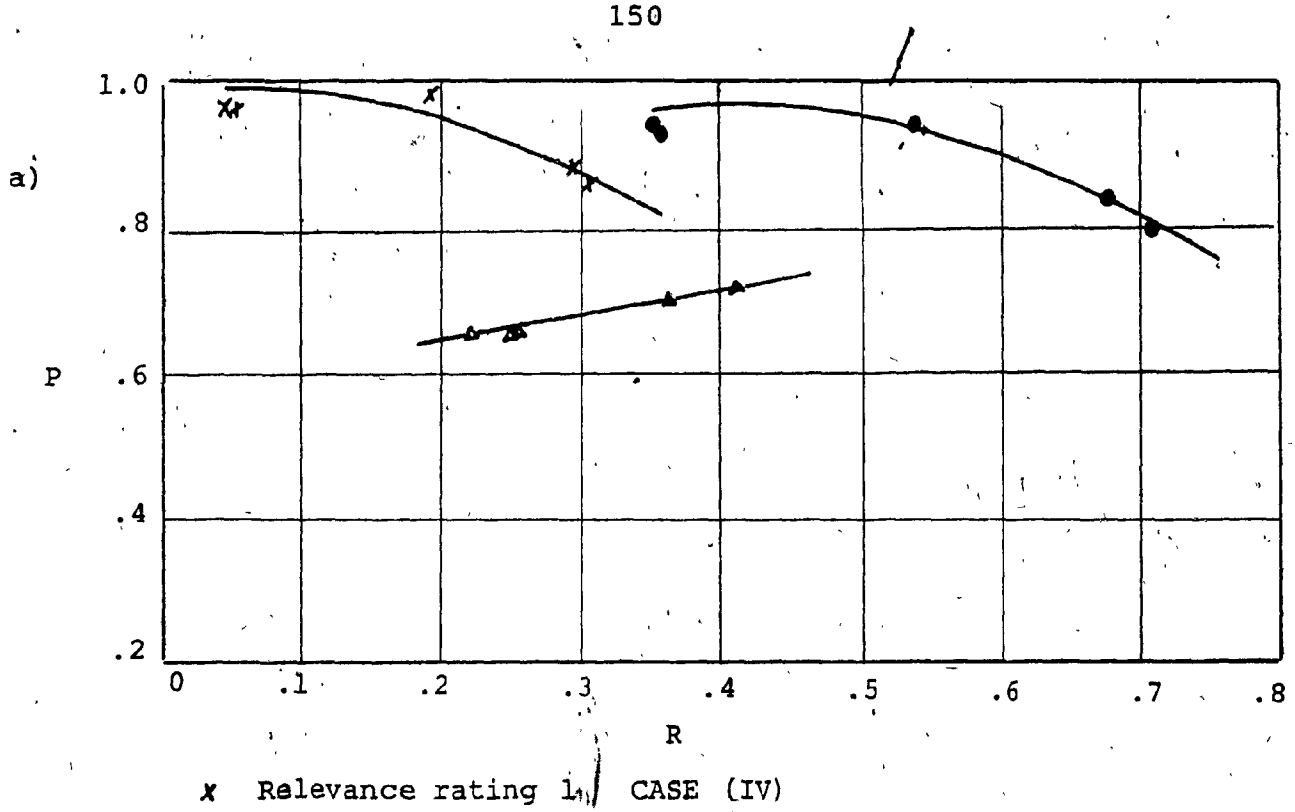


Fig. 5.19 Classification efficiency for three automatic document classification procedures: a) category 3, b) category 4.



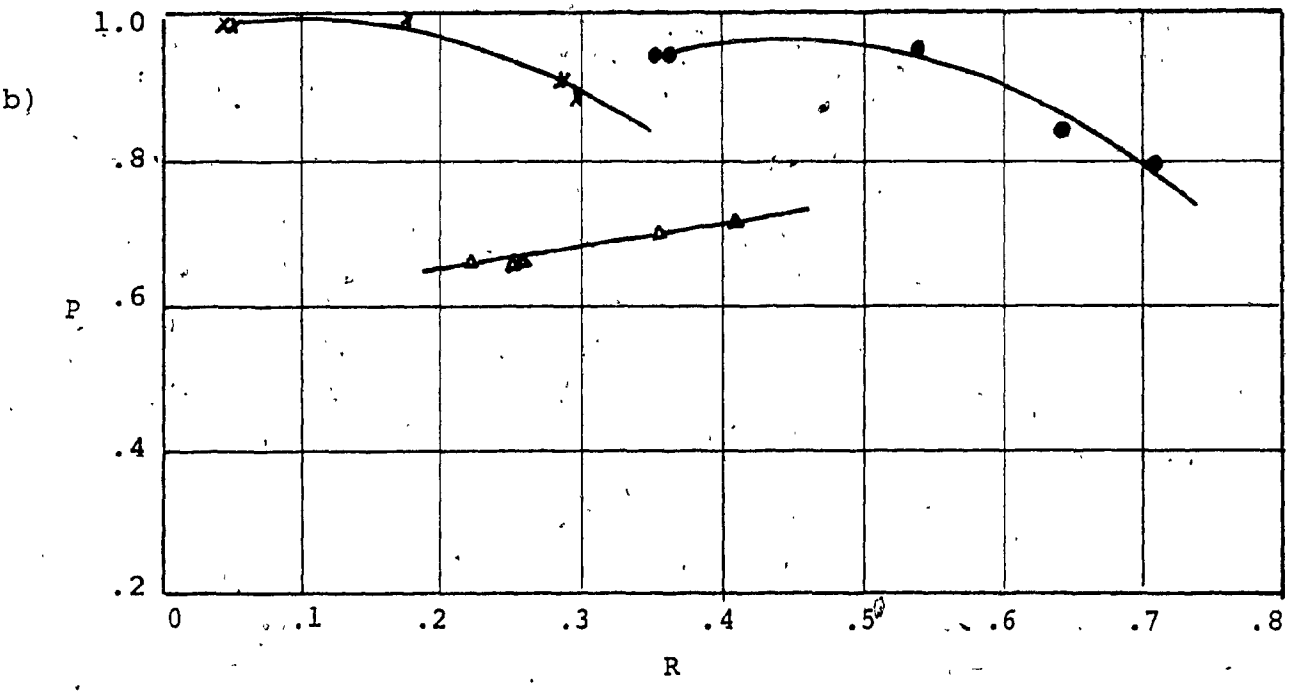


Fig. 5.20 Classification efficiency for three automatic document classification procedures: a) category 1 b) category 2.

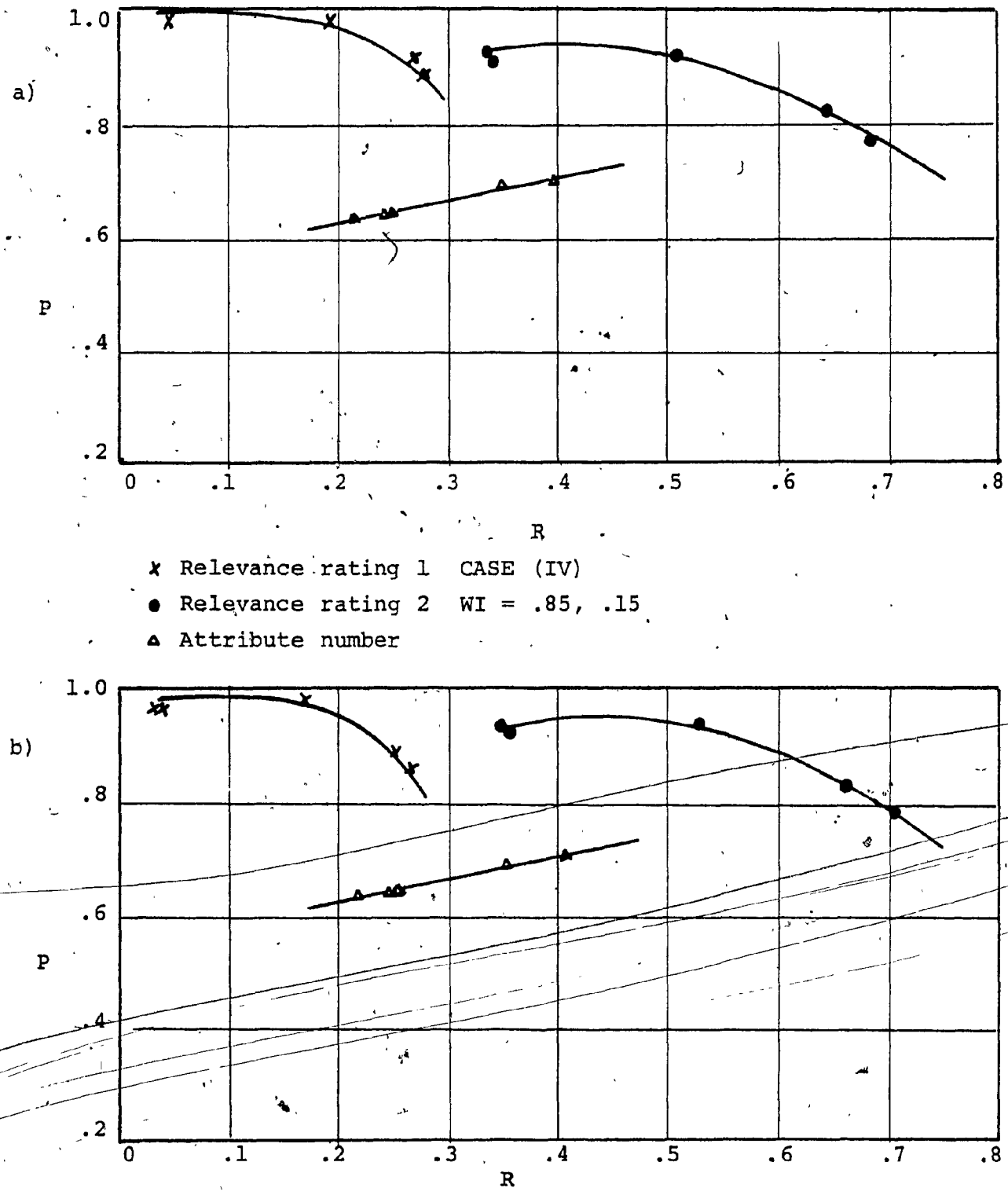


Fig. 5.21 Classification efficiency for three automatic document classification procedures: a) Category 3; b) Category 4.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY

As stated at the end of chapter 1, the present investigation has proceeded by means of three steps. Step 1 involved the simulation of document category associations as described in chapter 3. Step 2 was concerned with simulation of a document term matrix as described in chapter 4. Use of the simulated document data base in a study of three methods of automatic document classification was outlined in chapter 5.

An important feature of the simulation procedure is its dependence on a minimum number of given properties and assumptions regarding the nature of the bibliographic document data base being simulated. Thus in step 1 the only parameters that must be input to describe the size of the data base, and the number of different categories, are the following:

M=total number of indexing terms present in the data base

N=number of distinct indexing terms

K=number of categories with which the documents are associated

More significantly, the only parameters that must be input to describe the nature of the categorisation with respect to size and overlap of subject areas are the following:

p_k= proportion of documents that belong to the k th category.

p_{jk}= proportion of documents that belong to both the j th and k th categories.

For the input data of form described above the simulation procedure may be used to generate a sample data base whose categorisation of documents is consistent with the above input,but which is subject to no other assumptions regarding the properties of the categorisation scheme. If the simulation is repreated several times it generates successive examples of document data bases that satisfy the imposed conditions with respect to parameter values,but which are subject to no further constraints. Continued repetitions of the simulation procedure leads to generation of all possible such data bases.

The procedure of step 2,which generates the document term matrix, requires the assumption of Zipf's law,which is known to be valid for a large class of bibliographic data bases. It also requires a specification of the following.

p_c = probability that any term is a content term for some category in the sense that it tends to associate with documents of that category rather than with documents of other categories.

WI= a set of weights that quantify the above "tendecy to associate". Thus if the I th term tends to associate with documents of the k th category with a probability of p then

$$W_i(\ ,\ ,k,\ ,\ )= p$$

p_n = probability that any term is a non-content term in the

sense of having equally probable associations with documents of each category.

For given values of p_c, p_n, and WI the simulation is very general in that it produces a sample data base that is subject to no other constraints regarding the associations of terms with documents. It is not necessary that $p_c+p_n=1$ since it may be desired to simulate a data base in which a proportion $1-p_c-p_n$ of terms have unspecified properties with respect to being content or non-content terms. Such terms have been called "accidentally associative" terms. If the simulation procedure is repeated continually it will generate all possible document data bases that satisfy Zipf's law for the given values of p_c, p_n, and WI.

It should be noted that the input to the simulation program consists only of global size parameters M,N,D,K, the statistical parameters p_k and p_{jk} to describe categorisation, and the statistical parameters p_c, p_n, and WI that describe the richness of the data base with respect to content terms.

For any real document data base whose documents are classed with respect to subject categories the parameters M,N,D,K,p_k,p_{jk}, p_c, p_n, and WI could be determined experimentally. Such a data base could be regarded as an example of a whole family of data bases that could be generated by input of the same values of these parameters into the simulation procedure. It is suggested that an examination could be made of several small data bases whose documents are divided into categories,for example data

bases consisting of one year of ACM journal and communication papers, to determine the values of the parameters p_c, p_n, and WI. Simulations could then be made with the same values of the parameters, and comparisons could be made between the real and simulated data bases. Such a study is beyond the scope of the present investigation.

A data base whose documents fall into non-overlapping categories will be such that all $p_{jk}=0$, whereas if categories exhibit considerable overlap then the p_{jk} have relatively large values. Various schemes, whether manual or automatic, for document classification could be used on document data bases in order to determine the dependence of classification efficiency on the values of p_c and on the amount of category overlap. The extent to which increasing values of $1-p_c-p_n$, the probability of accidental term-associations, degrade the classification efficiency could be studied for various numbers of categories and category overlap. This could form the subject of a further investigation based on the simulation techniques described in the present thesis.

The comparison of three automatic classification schemes described in chapter 5 serves to illustrate use of the simulation procedure. It suggests that the simulation technique is applicable. Furthermore the graphs of figures 5.2-5.21 contain no features that suggest pecularities in the behaviour of the simulated data base.

The classification efficiency in terms of precision and recall for each automatic classification procedure has been presented in figures 5.6-5.21. An examination of all these graphical presentations of experimental results suggest that the predicted relevance rating method is far superior in performance to the attribute number method. Judging from the theory of the two automatic procedures one is intuitively inclined to accept the veracity of this result. A possible explanation for the poor performance of the attribute method is that the effect of the underlying assumptions is difficult to assess. The experimental results also show the effect of variations in the subject span of categories. This is indicated by comparisons of the graphs of figures 5.10,5.11,5.18,5.19 for case(iii) of the probabilities, and by figures 5.12,5.13,5.20,5.21 for case(iv) of the probabilities. The results suggest that a high overlap among categories gives high performance of automatic classification procedures.

The effect of variations of the statistical correlation of the I th content term with documents of the category for which it is a content term was observed by making changes in R_{ik}. The effect may be studied by making comparisons of tables 5.1,5.2 with tables 5.3,5.4. The values of P1 and P2 from tables 5.1,5.2 are higher than those of tables 5.3,5.4. The same trend can be seen from the plots of figures 5.6-5.13 and those of figures 5.14-5.21. The results confirm that higher value of R_{ik} give higher performance of classification for all the methods. It is

concluded that the classification efficiency is sensitive to the changes in classification ratings of content terms in the simulated data base.

REFERENCES

1. Lunh, H.P., "The Automatic Creation of Literature Abstracts", IBM Journal of Research and Development, vol.2, 159-165 (1958).

2. Good, I.J., "Speculations Concerning Information Retrieval", Research Report PC-78, IBM Research Centre, Yorktown Heights, New York (1958).

3. Fairthorne, R.A., "The Mathematics of Classification", Towards Information Retrieval, Butterworths, London, 1-10 (1961).

4. Sparck, Jones K., "Some thoughts on Classification for Retrieval", Journal of Documentation, 26, 89-101 (1970).

5. Maron, M.E. and Kuhns, J.L., "On Relevance, Probabilistic Indexing and Information Retrieval", J. ACM, vol.7, 1960, pp. 216-244.

6. Maron, M.E., "Automatic Indexing: An Experimental Inquiry", J. ACM, vol.8, 1961, pp. 404-417.

7. Borko, H. and Bernick, M., "Automatic Document Classification", J. ACM, vol.10, 1960, pp. 233-242.

8. Baker, F.B., "Information Retrieval based on Latent Class Analysis", J. ACM, vol. 9, 1962, pp.512-521.

9. Heaps, H.S., "A Theory of Relevance for Automatic Document Classification", Information and Control, vol. 22, 1973, pp. 268-278.

10. Heaps, H.S. and Leung, K.W., "Automatic Document Classification based on Theory of Relevance", Third International Study Conference on Classification Research (Bombay, 6-11, January 1975).

11. Naylor, T.H., et al, Computer Simulation Techniques, John Wiley & Sons, Inc., New York (1967).

12. Akers, Lex A., "Simulation of Semiconductor Devices", Simulation (SCS), vol. 29, no. 2, Aug (77) pp.33-41.

13. Surh, D.S. and Talavage, J.J., " A GASP IV Model of Steel Ingot Processing", Simulation (SCS), vol.29, no.2, Aug (77), pp. 49-55.

14. Reilly, Kevin D., User Determination of Library Request Presentation: A Simulation, Institute of Library Research, Univ. of California, L.A., (68).

15. Blunt, Charles R., "An Information Retrieval System Model", HRB-Singer, AD-623, 590 (1965).

16. Fried, J.B., et al, "Index Simulation Feasibility and Automatic Document Classification", Ohio State University, PB 182, 597 (1968).

17. Cooper, M.D., "A Simulation Model of an Information Retrieval System", Inform. Stor. Retr., vol.9, pp. 13-32 (1973).

18. Zipf, G.K., Human Behaviour and Principle of Least Effort, Addison Wesley, 1949.

19. Kucera, H. and Francis, W.N., Computational Analysis of Present-Day American English, Brown University Press, 1967.

20. Dewey, G., "Relative (Sic) Frequency of English Speech Sounds", Harvard University Press, Cambridge, Mass., Revised Edition, 1950.

21. Pierce, J.R., "Symbols, Signals and Noise: the Nature and Process of Communication", Harper & Row Publishers, N.Y. & London.

22. Booth, A.D., "A 'Law' of Occurrences for Words of Low Frequencies", Information and Control, vol.10, pp. 386-393, 1967.

23. Lancaster, F.W., "Information Retrieval Systems: Characteristics, Testing and Evaluation", Wiley, N.Y., 1968.

24. Sparck, Jones K., "A Statistical Interpretation of Term Specificity and its application in Retrieval", Journal of Documentation, 28, 11-21 (72).

25. Salton, G. and Yang, C.S., "On the Specification of Term Values in Automatic Indexing, Journal of Documentation, 29, 351-372 (73).

26. Sparck, Jones K., "Does Indexing Exhaustivity Matter", Journal of American Society for Information Science 24, 313-316 (1973).

27. Wadington, J.P., "Unit Concept Coordinate Indexing", American Documentation, 9, no. 2, (April 1958), pp. 107-113.

28. Herdun, G, The Advanced Theory of Language as Choice and Chance, Springer-Verlag, N.Y., 1966.

29. Taube, Mortimer, Gull, C.D. and Irma, S. Wachtel, "Unit Terms in Co-ordinate Indexing", American Documentation, 3, 4 (1952), 213, 18.

30. Meadow, Charles T., The Analysis of Information Systems, Melville Publishing Company, L.A., California, 1973.

31. Bookstein, A., "The Anamalous Behabiour of P in the Swet's Model and its Resolution", Journal of Documentation, 30, 374-380, 1974.

32. Heinc, M.H., "The Inverse Relation of P and R in terms of Swet's Model", Journal of Documentation, 29, 81-84, 1973.

33. Heaps, H.S., "Data Compression of Large Document Data Bases", Journal of Chemical Information, National Academy of Sciences, May 22-23, 1974.

34. Waltson, Claude E., "Information Retrieval"in"Advances in Computers", Vol. 6, ed: Franz L. ALT and Morris Rubinoff, Academic Press, 1965.

35. Yu, C.T., and Salton, G., "Precision Weighting - An Effective Automatic Indexing Method", JACM, (1976), 23, pp. 76-88.

APPENDIX

COMPUTER PROGRAMS

```
      PROGRAM PIJ(INPUT,OUTPUT)
C
C       P IS SYMMETRIC MATRIX OF JOINT PROBABILITIES . PIJK IS
C       JOINT PROBABILITY THAT A DOCUMENT BELONGS TO CATEGORY I,J,K.
C       KCON=COUNT FOR POSSIBLE PERMUTATION OF N OBJECTS TAKEN N
C       AT A TIME.
C       LCON=COUNT FOR NO. OF COMBINATION OF 4 OBJECTS TAKEN
C       3 AT A TIME.
C       L,K KEEPS INTEGER SEQUENCE UP TO 4
C
      DIMENSION P(4,4),Q(10),PIJK(10),K(5),L(5),QI(3,4),QS(4)
      READ 101,((P(I,J),J=1,4),I=1,4)
101   FORMAT(1X,4F5.3)
      PRINT 101,((P(I,J),J=1,4),I=1,4)
      M=4
      N=3
      I1=0
      LCON=0
9     KCON=0
      CALL NCOMB(L,M,N,I1)
      PRINT 7,(L(I),I=1,N)
7     FORMAT(* COMBI    =*,4I1)
15    Q(L(1))=0
C       FIND Q,R(I,J)=P(I,R)+P(J,R)-P(R,R)
      DO 14 I=2,N
      TEMP=P(L(I),L(1))
14    Q(L(1))=Q(L(1))+TEMP
      Q(L(1))=Q(L(1))-P(L(1),L(1))
      PRINT 8,Q(L(1)),P(L(2),L(3)),(L(I),I=1,N)
8     FORMAT(* Q(L(1))    =*,F10.5,* <=  P(L(2),L(3)) = *,F10.5,
     1* CYCLE  =    *,4I1)
      CALL CYCLE(L,N)
      KCON=KCON+1
      IF(KCON.EQ.N) GO TO 20
      GO TO 15
20    CONTINUE
      C=P(L(1),L(1))+P(L(2),L(2))+P(L(3),L(3))-1-P(L(1),L(2))-
     2P(L(1),L(3))-P(L(2),L(3))
      PRINT 89,C
89    FORMAT(*     C     = *,F10.5)
C
C       FIND LOWER AIJK BOUND AND UPPER BIJK BOUND
      AIJK=AMAX1(Q(L(1)),Q(L(2)),Q(L(3)),0.0)
      BIJK=AMIN1(P(L(1),L(2)),P(L(1),L(3)),P(L(2),L(3)),-C)
      X=RANF(YY)
      PRINT 149,X
149   FORMAT(*     X    =   *,F10.7)
C       GET PIJK UNIFORMLY DISTRIBUTED BETWEEN AIJK AND BIJK
      PIJK(L(1)+L(2)+L(3))=AIJK+X*(BIJK-AIJK)
103   FORMAT(1X,2F10.5)
      PRINT 103,AIJK,BIJK
      PRINT 102,L(1),L(2),L(3),PIJK(L(1)+L(2)+L(3))
102   FORMAT(1X,3I1,F10.4)
      LCON=LCON+1
      IF(LCON.EQ.4) GO TO  50
      GO TO 9
```

```
50      MCON=0
        M=4
        N=4
        I1=0
        CALL NCOMB(L,M,N,I1)
        PRINT 200,(L(I),I=1,N)
200     FORMAT(* COMB =*,4I1)
99      NCON=0
C
C        FIND Q,R(I,J,K)=P(I,J,R)+P(I,K,R)+P(J,K,R)-P(I,R)-P(J,R)
C        -P(K,R)+P(R,R)
        QS(L(1))=PIJK(L(1)+L(2)+L(3))+PIJK(L(1)+L(2)+L(4))
       1+PIJK(L(1)+L(3)+L(4))-P(L(1),L(2))-P(L(1),L(3))
       1-P(L(1),L(4))+P(L(1),L(1))
        PRINT 137,QS(L(1)),L(1)
137     FORMAT(* QS      = *,F10.6,*   I  = *,I1)
100     QI(L(1),L(2))=PIJK(L(1)+L(3)+L(2))+PIJK(L(1)+L(4)+L(2))
       1-P(L(1),L(2))
        PRINT 130,QI(L(1),L(2)),(L(I),I=1,2)
130     FORMAT(* QI    = *,F10.5,*  IJ  =*,4I1)
        NN=N-1
C        KEEP L(1) FIXED AND CHANGE L(2) TO L(4) IN CYCLIC ORDER
        DO 107 I=1,NN
107     K(I)=L(I+1)
        CALL CYCLE(K,N-1)
        DO 108 I=1,NN
108     L(I+1)=K(I)
        NCON=NCON+1
        IF(NCON.EQ.3) GO TO 109
        GO TO 100
109     CALL CYCLE(L,N)
        PRINT 211,(L(I),I=1,N)
211     FORMAT(*CYCLE L, N *,4I1)
        MCON=MCON+1
        IF(MCON.EQ.4) GO TO 115
        GO TO 99
115     CONTINUE
        C=P(L(1),L(1))+P(L(2),L(2))+P(L(3),L(3))+P(L(4),L(4))
       2-1-P(L(1),L(2))-P(L(1),L(3))-P(L(1),L(4))-P(L(2),L(3))
       2-P(L(2),L(4))-P(L(3),L(4))+PIJK(6)+PIJK(7)+PIJK(8)+PIJK(9)
        PRINT 97,C
97      FORMAT(*     C       = *,F10.5)
C
C        CALCULATE UPPER AND LOWER BOUND OF PIJKR
        AIJK=AMAX1(QI(1,2),QI(1,4),QI(1,3),QI(2,3),QI(2,4),QI(3,4),C)
        BIJK=AMIN1(PIJK(5),PIJK(7),PIJK(8),PIJK(9)
       1,QS(1),QS(2),QS(3),QS(4))
        PRINT 117,AIJK,BIJK
117     FORMAT(1X,2F10.5)
        X=RANF(YY)
        PIJKR=AIJK+X*(BIJK-AIJK)
        PRINT 118,L(1),L(2),L(3),L(4),PIJKR
118     FORMAT(1X,4I1,F10.5)
        STOP
        END
```

```
      SUBROUTINE NCOMB(L,M,N,KCN)
C
C       L GIVES COMBINATION SEQUENCE-RETURNED
C       N=NO. OF INTEGERS IN COMBINATION(INPUT TO SUBROUTINE)
C       M=TOTAL NO. OF INTEGERS OUT OF WHICH COMBINATION TO BE SELECTED
C       KCN =VARIABLE FOR BYPASSING FIRST THREE STATEMENTS
C
      DIMENSION MAX(10),L(5)
      IF(KCN.EQ.1) GO TO 11
      DO 10 I=1,N
      MAX(I)=M-N+I
10    L(I)=I
      KCN=KCN+1
      RETURN
11    L(N)=L(N)+1
      NN=N
25    IF(L(1).EQ.MAX(1)) GO TO 40
      IF(L(NN).GT.MAX(NN)) 30,40
30    NN=NN-1
      MM=L(NN)
      DO 35 I=NN,N
35    L(I)=MM+I-NN+1
      GO TO 25
40    RETURN
      END


      SUBROUTINE CYCLE(LL,NUM)
      DIMENSION LL(5)
      ITEM=LL(1)
      NN=NUM-1
      DO 10 I=1,NN
      LL(I)=LL(I+1)
10    CONTINUE
      LL(NUM)=ITEM
      RETURN
      END
```

```
      PROGRAM DCTR(INPUT,OUTPUT,TAPE2)
C
      DIMENSION P(4,4),PIJK(10),IP(16),AM(4,20),NN(4)
C     IP CONTIINS BIT SEQUENCE.AM CONTAINS NO. OF DOCUMENTS
C     WHICH ARE ALLOTED TO PARTICULAR SEQUENCE (M(I1,I2,..IN)
C     NB= NO. OF BITS IN   AND J=VALUE GIVEN BY BIT SEQUENCE
C     SEQUENCE IN COMPUTER WORD
C     IN AM(N,J) N= NUMBER OF BITS IN SEQUENCE. J=1+SUMMATION
C      2**(N-K)*I(K) WHERE K=SUMMATION INDEX FOR BIT SEQUENCE
C      PIJK(10) CONTAINS JOINT PROBABILITIES OF DOCUMENTS
C      BELONGING TO I,J,K,P CATEGORIES.
C     M=NUMBER OF DOCUMENTS
C
      READ*,M
    4 FORMAT(*     M     = *,I6)
      READ 1,((P(I,J),J=1,4),I=1,4)
    1 FORMAT(4F10.5)
      READ 1,(PIJK(I),I=6,10)
      PRINT 1,((P(I,J),J=1,4),I=1,4)
      PRINT 1,(PIJK(I),I=6,10)
      PRINT 4,M
      DO 3 I=1,4
      DO 3 J=1,20
    3 AM(I,J)=0
      AM(1,2)=P(1,1)*M
      AM(1,1)=(1-P(1,1))*M
      IP(1)=1
      NS=1
      NE=AM(1,2)
      PRINT 9
    9 FORMAT(20X,* AM *,5X,*   CUMULATIVE   *,6X,*   DOC. NO.   *,
     23X,* CAT. COMBINATION*)
      PRINT 7,AM(1,2),NS,NE,IP(1)
    7 FORMAT(15X,F10.2,20X,I5,5X,*TO*,I7,5X,4I2)
C      NS=STARTING DOC. NO. IN CATEGORY
C      NE=LAST DOC.NO. IN CATEGORY
      NS=NE+1
      NE=AM(1,1)+AM(1,2)+.001
      IP(1)=0
      PRINT 7,AM(1,1),NS,NE,IP(1)
C
      DO 200 NB=2,4
      NS=1
      CUML=0
      MM=2**NB
C
      DO 60 MC=1,MM
      CALL PERM(IP,MC,NB)
      PRINT 2,(IP(I),I=1,NB)
    2 FORMAT(4I2)
C      FIND IF LAST BIT IN PERMUTATION IS ZERO
      IF(IP(NB).EQ.0) GO TO 70
C     CALCULATE P,PSUM1=P(N1,N2...NS,R+1)*M
C      N1,N2 .... ARE VALUES OF N FOR WHICH I(N1)=1
      KCON=0
C     KCON IS COUNTER FOR NO. OF 1,S IN SEQUENCE
```

```
      II=0
      IJK=0
      DO 5 I=1,NB
 5    NN(I)=0
      LB=NB-1
      DO 10 L=1,LB
      IF(IP(L).EQ.1) GO TO 15
      GO TO 10
C     COUNT NO. OF 1,S IN SEQUENCE,NOTE,INDEX
 15   NN(L)=L
      LJ=L
      KCON=KCON+1
      IJK=IJK+NN(L)
 10   CONTINUE
      IF(KCON.EQ.3) GO TO 50
      IF(KCON.EQ.2) GO TO 20
      IF(KCON.EQ.1) GO TO 25
      PSUM1=P(NB,NB)*M
      GO TO 35
 20   LB=NB-1
      DO 30 L=1,LB
      IF(NN(L).EQ.0) GO TO 30
      IF(II.NE.0) GO TO 40
      II=NN(L)
 40   JJ=NN(L)
 30   CONTINUE
      PSUM1=PIJK(II+JJ+NB)*M
      GO TO 35
 25   PSUM1=P(NN(LJ),NB)*M
      GO TO 35
 50   PSUM1=PIJK(NN(1)+NN(2)+NN(3)+NB)*M
 35   PRINT 1,PSUM1
C
C     CALCULATE PSUM2 BYE CALLING PARTSUM
C     PSUM2=SUMMATION M(J1,J2,...J(R),1) SUMM IS OVERJ(I) (1,0)
C     FOR WHICH I(I)=0 EXCLUDING COMBINATION J(1),J(2),....J(R)
C     =I(1),I(2),...I(R)
C     J(I)=1 IF I1=0
      CALL PARTSUM(IP,NB,PSUM2,AM)
      CALL MADDR(NB,J,IP)
      AM(NB,J)=PSUM1-PSUM2
      PRINT 1,PSUM2
      GO TO 59
C     THE SEQUENCE IS I1,I2,..IR,0
C     FIRST FIND I1,I2,IR
C     M(I1,I2,..IR,0)=M(I1,I2,..IR)-M(I1,I2,..IR,1)
 70   CALL MADDR(NB-1,J,IP)
      SUM1=AM((NB-1),J)
C     PUT LAST BIT IN SEQUENCE=1
      IP(NB)=1
C     ABOVE SETS LAST BIT OF SEQUENCE 1
      CALL MADDR(NB,J,IP)
      SUM2=AM(NB,J)
C     RESTORE THE SEQUENCE
      IP(NB)=0
      CALL MADDR(NB,J,IP)
```

```
      AM(NB,J)=SUM1-SUM2
   59 CONTINUE
      CUML=CUML+AM(NB,J)+.001
      IF(CUML.GE.NS) GO TO 55
      NS=NS-1
      IF(NS.GE.NSAVE) GO TO 53
      NE=CUML+.001
      GO TO 57
   53 NS=0
      NE=0
      GO TO 57
   55 NE=CUML+.001
   57 PRINT 101,AM(NB,J),CUML,NS,NE,(IP(I),I=1,NB)
      IF(NB.NE.4) GO TO 69
      NCUML=CUML
      WRITE(2,103) NCUML
  103 FORMAT(I6)
   69 CONTINUE
      NE=CUML+.001
  101 FORMAT(15X,F10.2,5X,F10.2,5X,I5,5X,*TO*,I7,5X,4I2)
      NSAVE=NS
      NS=NE+1
   60 CONTINUE
  200 CONTINUE
      STOP
      END


      SUBROUTINE PERM(P,M,N)
C     THIS GENERATES A SEQUENCE P,M=SEQUENCE NUMBER,N=NO. OF
C     BITS IN SEQUENCE
C
      INTEGER P(15)
      MAX=2**N
      MM=MAX-M
      DO 1 I=1,N
      P(N-I+1)=MOD(MM,2)
      MM=SHIFT(MM,-1)
    1 CONTINUE
      RETURN
      END


      SUBROUTINE MADDR(NR,JC,II)
C     IT FINDS ADDR. OF EVERY AM(NB,J),NR=NO. OF BITS IN
C     BIT SEQUENCE,II=BIT SEQUENCE.
C     JC=1+SUMMATION 2**(NR-I)*II(I) AND ITS VALUE RETURNED.
C
      DIMENSION II(4)
      JC=1
      DO 10 I=1,NR
      JC=JC+2**(NR-I)*II(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PARTSUM(I,N,PSUM2,BM)
      DIMENSION I(16),II(16),BM(4,20)
C     LCON COUNTS  NO. OF ZEROS IN SEQUENCE
C
      LCON=0
      LB=N-1
C     COUNT NO. OF ZEROS IN SEQUENCE
   57 DO 60 L=1,LB
      IF(I(L).EQ.1)GO TO 60
      LCON=LCON+1
   60 CONTINUE
      IF(LCON.EQ.1) GO TO 62
      IF(LCON.EQ.2) GO TO 70
      IF(LCON.EQ.3) GO TO 90
C     FIND INDEX POSITION FOR WHICH BIT IS ZERO
      CALL MADDR(N,J,I)
      PSUM2=BM(N,J)
      RETURN
   62 DO 65 L=1,LB
      IF(I(L).EQ.1) GO TO 65
      K1=L
      I(K1)=1
   65 CONTINUE
      CALL MADDR(N,J,I)
      PSUM2=BM(N,J)
      I(K1)=0
C     THIS RESTORES SEQUENCE
      RETURN
   70 K1=0
C     FIND BIT POSITION WHERE THERE IS ZERO
      DO 75 L=1,LB
      IF(I(L).EQ.1) GO TO 75
      IF(K1.NE.0) GO TO 73
      K1=L
   73 K2=L
   75 CONTINUE
C     COPY SEQUENCE I1,I2..IN INTO II1,II2...II(N)
      DO 76 K=1,N
      II(K)=I(K)
   76 CONTINUE
      PSUM2=0
C     WHEN THERE ARE 2 ZEROS IN SEQUENCE FIND 2**2 PERMS
      DO 78 MM=1,4
      CALL PERM(II,MM,2)
      I(K1)=II(1)
      I(K2)=II(2)
      CALL MADDR(N,J,I)
      PSUM2=PSUM2+BM(N,J)
C     ORIGINAL SEQUENCE IS RESTORED
   78 CONTINUE
      RETURN
C     WHEN THERE ARE 3 ZEROS IN SEQUENCE FIND 2**3=8 PERMS
C     CALLING SUBROUTINE PERM.
   90 PSUM2=0
```

```
      DO 100 MM=1,8
      CALL PERM(II,MM,N-1)
      CALL MADDR(N,J,II)
      PSUM2=PSUM2+BM(N,J)
  100 CONTINUE
C     ORIGINAL SEQUENCE RESTORED
      RETURN
      END
```

```
      PROGRAM DTERM4(INPUT,OUTPUT)
C
C     SEQ IS A MATRIX WHOSE FIFTH COLUMN CONTAINS CUMULATIVE
C     DOCUMENTS. THE FIRST FOUR COLUMNS DESCRIBE THE
C     BEHAVIOUR OF GIVEN DOCUMENT TO CATEGORY COMBINATIONS.
C     D=NUMBER OF DISTINCT TERMS.
C     IA IS A DOCUMENT TERM MATRIX WHOSE ELEMENTS ARE 1 OR
C     0 DEPENDING ON WHETHER A DOCUMENT CONTAINS THE GIVEN
C     TERM OR NOT. IP IS A COLUMN VECTOR CONTAINING
C     SEQUENCE OF 1 AND 0.
C     PC=THE PROBABILITY THAT THE GIVEN TERM IS A CONTENT
C     TERM.
C     PN=THE PROBABILITY OF A TERM BEING NON-CONTENT
C     VECTOR NCN STORES THE SPECIFICATION OF A TERM
C
      INTEGER SEQ(16,5),D,RN
      DIMENSION IP(4),IA(50,70),NCN(70)
      DATA M,N,D,PC,PN/50,400,70, 0.2,0.7/
      DATA ((SEQ(I,5),I=1,16)=0,5,7,12,13,14,14,20,20,21,21,
     C22,22,23,26,50)
C     INITIALISE THE DOCUMENT TERM MATRIX
      DO 1 I=1,50
      DO 1 J=1,70
 1    IA(I,J)=0
      DO 5 I=1,16
      CALL PERM(IP,I,4)
      DO 3 J=1,4
      SEQ(I,J)=IP(J)
 3    CONTINUE
 5    CONTINUE
      PRINT 9,M,N,D,PC,PN
 9    FORMAT(* M =*,I5,* N =*,I8,* D =*,I6,* PC =*,F3.2,* PN =*,F3.2)
      PRINT 4,((SEQ(I,J),J=1,5),I=1,16)
 4    FORMAT(1X,4I1,5X,I2)
C     GENERATE RANDOM NUMBER RC BETWEEN 0 AND 1
      AD=D
      A=1/(ALOG(AD)+.5772)
      DO 100 I=1,D
C     CALCULATE TERM OCCURRENCES AS PER ZIPF'S LAW
      MI=A*N/I
      KCON=0
      IF(MI.LT.M) GO TO 7
      MI=M
 7    J=0
      Y=.5757*I+(5**.5+1)/2
      CALL RANSET(Y)
C     GENERATE RANDOM NUMBER RC
      RC=RANF(YY)
      PRINT 108,I,RC,Y
108   FORMAT(*START   TERM *,I2,*      RC=  *,F12.10,*   Y  =*,F10.5)
      IF(RC.LE.PC) GO TO 27
      IF((RC.GT.PC).AND.(RC.LE.PC+PN)) GO TO 26
C     OTHERWISE ACCIDENTAL TERM
      NCN(I)=1HA
      YO=.33*I+(5**.5+1)/2
      CALL RANSET(YO)
```

```
      WI=RANF(YY)
      GO TO 10
 26   NCN(I)=1HN
C     IT IS A NON-CONTENT TERM
      WI=1
      GO TO 10
 27   NCN(I)=1HC
C     IT IS A CONTENT TERM
C     GENERATE INTEGER NI BETWEEN 1 AND 4
      Y2=I**3+(5**.5+1)/2
      CALL RANSET(Y2)
      NI=RANF(YY)*4+1
      PRINT 99,NI
 99   FORMAT(*  NI    =*,I5)
 10   CONTINUE
      PRINT 200,NCN(I)
200   FORMAT(* TERM TYPE =*,A1)
      Y1=I+(5**.5+1)/2
      CALL RANSET(Y1)
 11   X=RANF(YY)
105   FORMAT(*X=  *,F12.10,*      RN=  *,I2)
C     GENERATE DOCUMENT NUMBER RN
      RN=X*M+1
C     PRINT 105,X,RN
      IF(IA(RN,I).EQ.1) GO TO 11
C      FIND DOC RANGE IN WHICH RN LIES
      DO 21 L=1,16
      KK=L
      IF(RN.LE.SEQ(L,5)) GO TO 22
21    CONTINUE
22    CONTINUE
      IF(NCN(I).NE.1HC) GO TO 35
C     FIND,CATGEROY NUMBER TO WHICH CONTENT TERM HAS MORE BIAS
      IF(SEQ(KK,NI).EQ.1) GO TO 30
      WI=0.05
      GO TO 35
 30   WI=.95
C      GENERATE RAND NUMBER BETWEEN 0 AND 1
 35   SN=RANF(YY)
      PRINT 107,SN,WI,RN
107   FORMAT(*SN=  *,F12.10,*      WI=  *,F5.3,*   RN=   *,I3)
      IF(SN.GT.WI) GO TO 11
C
C     ALLOT THE TERM TO THE DOCUMENT
      IA(RN,I)=1
      KCON=KCON+1
      IF(KCON.LT.MI) GO TO 11
100   CONTINUE
      PRINT 101,(NCN(K),K=1,60)
101   FORMAT(5X,60(1X,A1))
      DO 104 L=1,50
      PRINT 103,L,(IA(L,K),K=1,60)
104   CONTINUE
103   FORMAT(I3,2X,60I2)
      STOP
      END
```

```
C
C
C
C
      SUBROUTINE PERM(P,M,N)
C
C     THIS GENERATES BIT SEQUENCE  P WHICH IS RETURNED TO
C     MAIN PROGRAM. M=SEQUENCE COUNT
C     N=NUMBER OF BITS IN SEQUENCE
C
      INTEGER P(4)
      MAX=2**N
      MM=MAX-M
      DO 1 I=1,N
      P(N-I+1)=MOD(MM,2)
      MM=SHIFT(MM,-1)
    1 CONTINUE
      RETURN
      END
```

```
      PROGRAM DTERM(INPUT,OUTPUT,TAPE3)
C
C     SEQ. IS MATRIX WHOSE FIFTH COLUMN CONTAINS CUMULATIVE
C     DOCUMENTS.THE FIRST FOUR COLUMNS DESCRIBE THE BEHAVIOUR
C     OF GIVEN DOCUMENT TO GIVEN CATEGORY COMBINATIONS.
C     D=NUMBER OF DIFFERENT TERMS AND RN IS PSEUDO-DOCUMENT
C     NUMBER.IA IS A COLUMN VECTOR WHOSE ELEMENTS ARE 1 OR 0.
C     THE ELEMENT IS 1 IF THE I TH TERM OCCURS IN THE DOCUMENT
C     OTHERWISE ZERO. NTERM GIVES CONTENT TERMS FOR THE GIVEN
C     CATEGORY. IP CONTAINS I1,I2,I3,I4 WHOSE COMBINATION DESCRIBE
C     THE BEHAVIOUR OF THE GIVEN DOCUMENT TO CATEGORY COMBINATION
C     IV IS RELEVANCE COLUMN VECTOR.
C
      INTEGER SEQ(16,5),D,RN
      DIMENSION IP(4),IA(5000),NCN(70),IV(5000),NTERM(500)
      DATA M,N,D,PC,PN/5000,40000,7000, 0.2,0.7/
C     READ THE FIRST AND LAST CATEGORY NUMBER
      READ*,KNBE,KNLA
      DO 5 I=1,16
      READ(3,29) SEQ(I,5)
29    FORMAT(I6)
      CALL PERM(IP,I,4)
      DO 3 J=1,4
      SEQ(I,J)=IP(J)
 3    CONTINUE
 5    CONTINUE
      PRINT 9,M,N,D,PC,PN
9     FORMAT(* M =*,I5,* N =*,I6,* D =*,I6,* PC =*,F5.2,* PN =*,F5.2//)
      PRINT 4,((SEQ(I,J),J=1,5),I=1,16)
 4    FORMAT(1X,4I1,5X,I10)
C     GENERATE RANDOM NUMBER RC BETWEEN 0 AND 1
      AD=0
      A=1/(ALOG(AD)+.5772)
C
      DO 300 K=KNBE,KNLA
      DO 2 I=1,M
2     IV(I)=0
C     FIND TOTAL DOCS IN CATEGORY K
      TOTDC=0
      DO 15 J=1,16
      IF(SEQ(J,K).NE.1) GO TO 15
      TOTDC=TOTDC+SEQ(J,5)-SEQ(J-1,5)
15    CONTINUE
      PRINT 108,K,TOTDC
108   FORMAT(//* TOTAL DOCS IN CATEGORY  *,I1,* = *,F10.1)
      L1=0
C
      DO 100 I=50,D
C     COMPUTE TERM OCCURRENCES AS PER ZIPF'S LAW
      MI=A*N/I
      KCON=0
      IF(MI.LT.M) GO TO 7
      MI=M
 7    J=0
C     GENERATE RANDOM NUMBER RC BETWEEN 0 AND 1
      Y=.5757*I+(5**.5+1)/2
```

```
      CALL RANSET(Y)
      RC=RANF(YY)
C     PRINT 106,I,RC,Y
106   FORMAT(*START   TERM *,I2,*      RC=  *,F12.10,*   Y  =*,F10.5)
      IF(RC.LE.PC) GO TO 27
      GO TO 100
 27   CONTINUE
C      GENERATE INTEGER NI BETWEEN 1 AND 4
      Y2=I**3+(5**.5+1)/2
      CALL RANSET(Y2)
      NI=RANF(YY)*4+1
      IF(NI.NE.K) GO TO 100
      DO 8 I5=1,M
8     IA(I5)=0
C     PRINT 99,NI
99    FORMAT(*  NI   =*,I5)
C     PRINT 200,NCN(I),I
 200  FORMAT(* TERM TYPE =*,A1, *  TERM NO. = *,I6)
      Y1=I+(5**.5+1)/2
      CALL RANSET(Y1)
 11   X=RANF(YY)
 105  FORMAT(*X=  *,F12.10,*     RN=  *,I2)
      RN=X*M+1
C     PRINT 105,X,RN
      IF(IA(RN).EQ.1) GO TO 11
C      FIND DOC.RANGE IN WHICH RN LIES
      DO 21 L=1,16
      KK=L
      IF(RN.LE.SEQ(L,5)) GO TO 22
 21   CONTINUE
 22   CONTINUE
C     IF(NCN(I).NE.1HC) GO TO 35
C     FIND,CATGERCY NUMBER TO WHICH CONTENT TERM HAS MORE BIAS
      IF(SEQ(KK,NI).EQ.1) GO TO 30
      WI=0.05
      GO TO 35
 30   WI=.95
C      GENERATE RAND NUMBER BETWEEN 0 AND 1
 35   SN=RANF(YY)
C     PRINT 107,SN,WI,RN
 107  FORMAT(*SN= *,F12.10,*      WI=  *,F5.3,*   RN=   *,I3)
      IF(SN.GT.WI) GO TO 11
C     OTHERWISE ASSOCIATE THE TERM WITH RN DOCUMENT
      IA(RN)=1
C     UPDATE THE RELEVANCE COLUMN VECTOR
      IV(RN)=IV(RN)+1
      KCON=KCON+1
      IF(KCON.LT.MI) GO TO 11
      L1=L1+1
      NTERM(L1)=I
 100  CONTINUE
      PRINT 349,K
349   FORMAT(/* CONTENT TERMS OF CATEGORY  *,I2)
      PRINT 348,(NTERM(I1),I1=1,L1)
348   FORMAT(2015)
      PRINT 346
```

```
346   FORMAT(/* ATTRIBUTE NUMBERS OF DOCS.    *)
      PRINT 350,(IV(LL),LL=1,50)
350   FORMAT(/50I2)
C        MAKE NTERM=0
      DO 347 I1=1,L1
347   NTERM(I1)=0
      DO 150 ITHRESH=1,6
      TRT=0
      RRT=0
      DO 50 J=1,M
      IF(IV(J).LT.ITHRESH) GO TO 50
      TRT=TRT+1
C      FIND WHETHER J TH DOC. BELONGS TO CATEGORY K
      DO 23 JJ=1,16
      IF(J.GT.SEQ(JJ,5)) GO TO 23
19    IF(SEQ(JJ,K).EQ.1) GO TO 31
      GO TO 50
23    CONTINUE
      GO TO 50
31    RRT=RRT+1
400   FORMAT(/* RRT = *,F10.1)
50    CONTINUE
      PRINT 400,RRT
      PRINT 430,TRT
430   FORMAT(/* TRT = *,F10.1)
      IF(TRT.EQ.0) GO TO 300
      PR=RRT/TRT
      REC=RRT/TOTDC
      PRINT 111,PR,REC,ITHRESH,K
111   FORMAT(/* PRECISION = *,F6.3,* RECALL =*,F6.3,*  THRESH= *,I2
     +,*  CATEGORY =*,I1)
150   CONTINUE
300   CONTINUE
C     PRINT 101,(NCN(KI),KI=1,60)
101   FORMAT(5X,60(1X,A1))
C     DO 104 L=1,50
C     PRINT 103,L,(IA(L,K),K=1,60)
C104  CONTINUE
103   FORMAT(I3,2X,60I2)
      STOP
      END


      SUBROUTINE PERM(P,M,N)
C     THIS GENERATES BIT SEQUENCE P,M=SEQUENCE COUNT,
C     N=NUMBER OF BITS IN SEQUENCE
C
      INTEGER P(4)
      MAX=2**N
      MM=MAX-M
      DO 1 I=1,N
      P(N-I+1)=MOD(MM,2)
      MM=SHIFT(MM,-1)
1     CONTINUE
      RETURN
      END
```

```
      PROGRAM KRATE(INPUT,OUTPUT,TAPE3)
C
C     SEQ. IS MATRIX WHOSE FIFTH COLUMN CONTAINS CUMULATIVE
C     DOCUMENTS.THE FIRST FOUR COLUMNS DESCRIBE THE BEHAVIOUR OF
C     GIVEN DOCUMENT TO GIVEN CATEGORY COMBINATIONS.
C     D= NUMBER OF DIFFERENT TERMS AND KN IS PSEUDO DOCUMENT
C     NUMBER.IA IS COLUMN VECTOR WHOSE ELEMENTS ARE 1 OR 0
C     IT IS 1 IF THE I TH TERM OCCURS IN THE DOCUMENT OTHERWISE 0.
C     NTERM KEEPS COUNT OF CONTENT TERMS FOR GIVEN CATEGORY.
C     YIK COUNTS DOCUMENTS OF CATEGORY K THAT CONTAIN TERM I.
C     COUNT IS COUNTER FOR DOCS WHICH DO NOT CONTAIN ANY CONTENT
C     TERM.
C     LCON IS NUMBER OF DOCUMENTS HAVING NO CONTENT TERM OUT OF
C     THOSE DOCS.WHICH ARE ASSIGNED TO SOME CATEGORY.
C     LCON1 IS NUMBER OF DOCUMENTS IN FIRST 2 CATEGORIES WHICH
C     DO NOT CONTAIN ANY CONTENT TERM.
C     LCON2 IS DOCUMENTS WHICH ARE IN ANY 2 CATEGORIES BUT DO
C     NOT CONTAIN CONTENT TERMS.
C
C
      INTEGER SEQ(16,5),D,PN
      LOGICAL FLAG
      DIMENSION IP(4),IA(5000),NTERM(500)
      COMMON YKM(5000,4),KCON1,KCON2,TOT1,TOT2,COUNT,LCON,LCON1
     +,LCON2
      DATA M,N,D,PC,PN/5000,40000,7000, 0.2,0.7/
      READ*,KNBE,KNLA
      DO 5 I=1,16
      READ(3,29) SEQ(I,5)
29    FORMAT(I6) /
      CALL PERM(IP,I,4)
      DO 3 J=1,4
      SEQ(I,J)=IP(J)
 3    CONTINUE
 5    CONTINUE
      PRINT 9,M,N,D,PC,PN
 9    FORMAT(* M =*,I5,* N =*,I8,* D =*,I6,* PC =*,F5.2,* PN =*,F5.2//)
      PRINT 4,((SEQ(I,J),J=1,5),I=1,16)
 4    FORMAT(1X,4I1,5X,I10)
C     INITIALISE RELEVANCE MATRIX
      DO 39 I=1,M
      DO 39 J=1,4
39    YKM(I,J)=-1
      AD=0
      A=1/(ALOG(AD)+.5772)
      DO 300 K=KNBE,KNLA
C     FIND TOTAL DOCS IN CATEGORY K
      TOTDC=0
      DO 15 J=1,16
      IF(SEQ(J,K).NE.1) GO TO 15
      TOTDC=TOTDC+SEQ(J,5)-SEQ(J-1,5)
15    CONTINUE
      PRINT 108,K,TOTDC
108   FORMAT(//* TOTAL DOCS IN CATEGORY. *,I1,* = *,F10.1)
      L1=0
C     GENERATE   TERM ASSOCIATIONS WITH DOCUMENTS
```

```
C
      DO 100 I=50,D
      VIK=C
C     COMPUTE TERM OCCURANCES AS PER ZIPF,S LAW
      MI=A*N/I.
      KCON=0
      IF(MI.LT.M) GO TO 7
      MI=M
 7    J=C
C      GENERATE,RANDOM NUMBER RC BETWEEN 0 AND 1
      Y=.5757*I+(5**.5+1)/2
      CALL RANSET(Y)
      RC=RANF(YY)
C     PRINT 108,I,RC,Y
108   FORMAT(*START   TERM *,I2,*      RC=  *,F12.10,*   Y  =*,F10.5)
      IF(RC.LE.PC) GO TO 27
      GO TO 100
27    CONTINUE
C      GENERATE INTEGER NI BETWEEN 1 AND 4
      Y2=I**3+(5**.5+1)/2
      CALL RANSET(Y2)
      NI=RANF(YY)*4+1
      IF(NI.NE.K) GO TO 100
      DO 8 I5=1,M
8     IA(I5)=C
C     PRINT 99,NI
99    FORMAT(*  NI   =*,I5)
C     PRINT 200,I
200   FORMAT(* COUNT OF RELEVANT DOCUMENT TO CAT. K FOR TERM =*,I6)
      Y1=I+(5**.5+1)/2
      CALL RANSET(Y1)
 11   X=RANF(YY)
105   FORMAT(*X=  *,F12.10,*     RN=  *,I2)
      RN=X*M+1
C     PRINT 105,X,RN
      IF(IA(RN).EQ.1) GO TO 11
C      FIND DOC RANGE IN WHICH RN LIES
      DO 21 L=1,16
      KK=L
      IF(RN.LE.SEG(L,5)) GO TO 22
21    CONTINUE
22    CONTINUE
C     FIND,CATGEROY NUMBER TO WHICH CONTENT TERM HAS MORE BIAS
      IF(SEG(KK,NI).EQ.1) GO TO 30
      WI=C.15
      FLAG=.FALSE.
      GO TO 35
 30   WI=.c5
      FLAG=.TRUE.
C      GENERATE RAND NUMBER BETWEEN 0 AND 1
 35   SN=RANF(YY)
C     PRINT 107,SN,WI,RN
107   FORMAT(*SN= *,F12.10,*      WI=  *,F5.3,*  RN=   *,I3)
      IF(SN.GT.WI) GO TO 11
      IA(RN)=1
C     AFTER TERM ASSIGNED TO DOCUMENT RN  COUNT RELEVANCE
```

```
C       OF RN TO CAT. K
        IF(FLAG) VIK=VIK*1
C       PRINT 97,RN,VIK
 97     FORMAT(* RN  = *,I5,*  VIK  =  *,F5.1)
        KCON=KCON+1
        IF(KCON.LT.MI) GO TO 11
C        AFTER MI OCCURANCES OF TERM IS COMPLETED
C        COMPUTE RELEVANCE RATING CONTRIBUTION OF I TH TERM.
C        TO EVERY DOCUMENT
C       PRINT 255
255     FORMAT(* AFTER CONTRIBUTION TO RELEVANCE OF DOCUMENT TO CAT. K
     + BY THE ABOVE TERM*)
        DO 250 I6=1,M
        IF(IA(I6).EQ.0) GO TO 250
        IF(YKM(I6,K).EQ.-1) YKM(I6,K)=0.0
        YKM(I6,K)=YKM(I6,K)+VIK/KCON
C       PRINT 254,I6,YKM(I6,K)
254     FORMAT(* RN = *,I5,4F10.4)
250     CONTINUE
        L1=L1+1
        NTERM(L1)=I
 100    CONTINUE
        PRINT 349,K
349     FORMAT(/* CONTENT TERMS OF CATEGORY    *,I2)
        PRINT 348,(NTERM(I1),I1=1,L1)
348     FORMAT(20I5)
C          MAKE NTERM=0
        DO 347 I1=1,L1
347     NTERM(I1)=0
        DO 150 ITHRESH=5,13,2
        THRESH=.1*ITHRESH
        IRT=0
        RRT=0
        DO 50 J=1,M
        IF(YKM(J,K).LT.THRESH) GO TO 50
        IRT=IRT+1
C        FIND WHETHER J TH DOC. BELONGS TO CATEGORY K
        DO 23 JJ=1,16
        IF(J.GT.SEQ(JJ,5)) GO TO 23
19      IF(SEQ(JJ,K).EQ.1) GO TO 31
        GO TO 50
23      CONTINUE
        GO TO 50
31      RRT=RRT+1
400     FORMAT(/* RRT = *,F10.1)
50      CONTINUE
        PRINT 400,RRT
        PRINT 430,IRT
430     FORMAT(/* IRT = *,F10.1)
        IF(IRT.EQ.0) GO TO 300
        PR=RRT/IRT
        REC=RRT/TOTDC
        PRINT 111,PR,REC,THRESH,K
111     FORMAT(/* PRECISION = *,F6.3,* RECALL =*,F6.3,*  THRESH= *,F4.2
     +,*  CATEGORY =*,I1)
150     CONTINUE
```

```
300     CONTINUE
        DO 104 L=1,50
        PRINT 103,L,(YKM(L,K),K=1,4)
 104    CONTINUE
103     FORMAT(I3,2X,4F6.2)
        COUNT=0
        LCON1=0
        LCON2=0
        LCON=0
        KCON1=0
        KCON2=0
        TOT1=0
        TOT2=0
        DO 500 I=1,M
        CALL RANK(SEQ,4,I)
500     CONTINUE
        PRINT*,KCON1,TOT1,KCON2,TOT2
        P1=KCON1/TOT1
        P2=KCON2/TOT2
        PRINT 501,P1,P2,COUNT
501     FORMAT(*  P1  = *,F6.4,*  P2  = *,F6.4,* DOCS CONTAINING NO TERM =
      + *,F6.1)
        PRINT 503,LCON,LCON1,LCON2
503     FORMAT(*  LCON  =  *,I6,*  LCON1  =*,I6,*  LCON2  = *,I6)
        STOP
        END


        SUBROUTINE PERM(P,M,N)
C
C       THIS GENERATES BIT SEQUENCE P,M=SEQUENCE COUNT
C       N=NUMBER OF BITS IN SEQUENCE
C
        INTEGER P(4)
        MAX=2**N
        MM=MAX-M
        DO 1 I=1,N
        P(N-I+1)=MOD(MM,2)
        MM=SHIFT(MM,-1)
 1      CONTINUE
        RETURN
        END


        SUBROUTINE RANK(SEQ,N,II)
C
C        THIS GIVES RANKED OUTPUT OF A DOCUMENT FOR FOUR
C       CATEGORIES. IT IS IN DECREASING ORDER OF RELEVANCE
C       OF A DOCUMENT TO A CATEGORY.
C       KCON1 COUNTS THE NUMBER OF CHANCES ,WHEN CORRECT
C       CATEGORY TOPS THE RANKED LIST. KCON2 COUNTS CHANCES
C       ,WHEN FIRST TWO CATEGORIES (CORRECT) TOP THE LIST.
C
        INTEGER CAT(4),SEQ(16,5)
        LOGICAL FINISH
```

```
      COMMON YKM(5000,4),KCON1,KCON2,TOT1,TOT2,COUNT,LCON,LCON1
     +,LCON2
      DO 3 J=1,4
3     CAT(J)=J
      DO 6 K=2,N
      FINISH=.TRUE.
      LP=N-K+1
      DO 5 L=1,LP
      IF(YKM(II,L).GE.YKM(II,L+1)) GO TO 5
      TEMP=YKM(II,L+1)
      YKM(II,L+1)=YKM(II,L)
      YKM(II,L)=TEMP
      TEMP=CAT(L+1)
      CAT(L+1)=CAT(L)
      CAT(L)=TEMP
      FINISH=.FALSE.
5     CONTINUE
      IF(FINISH) GO TO 11
6     CONTINUE
11    CONTINUE
C     PRINT 10,(CAT(J),J=1,4)
10    FORMAT(4X,4I5)
C     PRINT 12,II,(YKM(II,J),J=1,4)
12    FORMAT(I4,4F5.2)
      IF(YKM(II,1).NE.-1) GO TO 19
      COUNT=COUNT+1
19    DO 23 L=1,16
      IF(L.EQ.16) RETURN
      IF(II.GT.SEQ(L,5)) GO TO 23
      IF(YKM(II,1).NE.-1) GO TO 20
      LCON=LCON+1
      IF(SEQ(L,CAT(1)).EQ.1.AND.SEQ(L,CAT(2)).EQ.1) LCON1=LCON1+1
C      COUNT DOCS.WHICH ARE IN ANY 2 CATEGORIES BUT DO NOT HAVE
C      CONTENT TERM.
      DO 30 KL=1,4
      IF(SEQ(L,CAT(KL)).NE.1) GO TO 31
      JJ=KL+1
      IF(JJ.EQ.5)GO TO 31
      DO 29 JL=JJ,4
      IF(SEQ(L,CAT(JL)).NE.1) GO TO 29
      LCON2=LCON2+1
      RETURN
29    CONTINUE
31    CONTINUE
      RETURN
20    IF(YKM(II,1).EQ.0.0) RETURN
      IF(SEQ(L,CAT(1)).EQ.1) KCON1=KCON1+1
      TOT1=TOT1+1
      IF(YKM(II,2).EQ.0.0.OR.YKM(II,2).EQ.-1) GO TO 24
      IF(SEQ(L,CAT(1)).EQ.1.AND.SEQ(L,CAT(2)).EQ.1) KCON2=KCON2+1
      TOT2=TOT2+1
      GO TO 24
23    CONTINUE
24    CONTINUE
      RETURN
      END
```

```
      PROGRAM RRATE1(INPUT,OUTPUT,TAPE3)
C
C     SEQ IS MATRIX WHOSE FIFTH COLUMN CONTAINS CUMULATIVE
C     DOCUMENTS.THE FIRST FOUR COLUMNS DESCRIBE BEHAVIOUR OF
C     GIVEN DOCUMENT TO GIVEN CATEGORY COMBINATION.
C     D=NUMBER OF DIFFERENT TERMS AND RN IS PSEUDO DOCUMENT
C     NUMBER.IA IS COLUMN VECTOR WHOSE ELEMENTS ARE 1 OR 0
C     IT IS 1 IF I TH TERM OCCURS IN THE DOCUMENT
C     NTERM KEEPS COUNT OF CONTENT TERMS FOR THE GIVEN CATEGORY.
C     VECTOR CK COUNTS DOCUMENTS IN CATEGORY K AND CIK COUNTS
C     NUMBER OF DOCUMENTS OF CATEGORY K THAT CONTAIN TERM I.
C     YKM-DOCUMENT CATEGORY MATRIX WHOSE EACH COMPONENT IS
C     RELEVANCE RATING OF M TH DOCUMENT TO CATEGORY K. GIVEN
C     A DOCUMENT CONTAINS TERMS W1,W2...WN ,ITS RELEVANCE TO
C     CATEGORY K IS DEFINED AS
C     LCON COUNTS DOCUMENTS HAVING NO CONTENT TERM
C     OUT OF THOSE DOCUMENTS WHICH ARE ASSIGNED TO SOME CATEGORY.
C      LCON1 IS NUMBER OF DOCUMENTS WHICH ARE IN 2 CATEGORIES
C     AND DO NOT CONTAIN CONTENT TERMS
C
C
      INTEGER SEQ(16,5),D,RN
      DIMENSION IP(4),IA(5000),NTERM(500),CK(4),CIK(4)
      COMMON YKM(5000,4),KCON1,KCON2,TOT1,TOT2,COUNT,LCON,LCON1
      DATA M,N,D,PC,PN/5000,40000,7000, 0.2,0.7/
      READ*,KNBE,KNLA
      DO 5 I=1,16
      READ(3,29) (SEQ(I,J),J=1,5)
29    FORMAT(16I6)
      CALL PERM(IP,I,4)
      DO 3 J=1,4
      SEQ(I,J)=IP(J)
3     CONTINUE
5     CONTINUE
      PRINT 9,M,N,D,PC,PN
9     FORMAT(* M =*,I5,* N =*,I8,* D =*,I6,* PC =*,F5.2,* PN =*,F5.2//)
      PRINT 4,((SEQ(I,J),J=1,5),I=1,16)
4     FORMAT(1X,4I1,5X,I10)
C      INITIALISE RELEVANCE RATING MATRIX
      DO 39 I=1,M
      DO 39 J=1,4
39    YKM(I,J)=-1
      DO 1 L=1,4
1     CK(L)=0
C     FIND TOTAL DOCUMENTS IN EACH CATEGORY
      DO 15 J=1,16
      DO 49 L=1,4
      IF(SEQ(J,L).NE.1) GO TO 49
      CK(L)=CK(L)+SEQ(J,5)-SEQ(J-1,5)
49    CONTINUE
15    CONTINUE
      AD=D
      A=1/(ALOG(AD)+.5772)
      DO 300 K=KNBE,KNLA
C      FIND TOTAL DOCS IN CATEGORY K
      TOTDC=CK(K)
```

```
      PRINT 108,K,TOTDOC
108   FORMAT(//* TOTAL DOCS IN CATEGORY  *,I1,* = *,F10.1)
      L1=0
C     GENERATE I TERM ASSOCIATION WITH DOCUMENT
      DO 100 I=50,D
      DO 26 L=1,4
26    CIK(L)=0
C     COMPUTE TERM OCCURANCES AS PER ZIPF'S LAW
      MI=A*N/I
      KOUN=0
      IF(MI.LT.M) GO TO 7
      MI=M
 7    J=0
C      GENERATE RANDOM NUMBER RC BETWEEN 0 AND 1
      Y=.5757*I+(5**.5+1)/2
      CALL RANSET(Y)
      RC=RANF(YY)
C     PRINT 105,I,RC,Y
105   FORMAT(*START   TERM *,I2,*      RC= *,F12.10,*   Y  =*,F10.5)
      IF(RC.LE.PC) GO TO 27
      GO TO 100
27    CONTINUE
C      GENERATE INTEGER NI BETWEEN 1 AND K
      Y2=1**3+(5**.5+1)/2
      CALL RANSET(Y2)
      NI=RANF(YY)*4+1
      IF(NI.NE.K) GO TO 100
      DO 8 I5=1,M
C     INITIALISE VECTOR IA.
8     IA(I5)=0
C     PRINT 99,NI
99    FORMAT(* NI =*,I5)
C     PRINT 200,I
 200  FORMAT(* COUNT OF RELEVANT DOC. TO EACH CAT.FOR TERM =*,I6)
      Y1=I+(5**.5+1)/2
      CALL RANSET(Y1)
 11   X=RANF(YY)
105   FORMAT(*X=  *,F12.10,*     RN=  *,I2)
      RN=X*M+1
C     PRINT 105,X,RN
      IF(IA(RN).EQ.1) GO TO 11
C      FIND DOC RANGE IN WHICH RN LIES
      DO 21 L=1,16
      KK=L
      IF(RN.LE.SEG(L,5)) GO TO 22
21    CONTINUE
22    CONTINUE
C     FIND,CATGEROY NUMBER TO WHICH CONTENT TERM HAS MORE BIAS
      IF(SEG(KK,NI).EQ.1) GO TO 30
      WI=0.15
      GO TO 35
 30   WI=.75
C      GENERATE RAND NUMBER BETWEEN 0 AND 1
 35   SN=RANF(YY)
C     PRINT 107,SN,WI,RN
107   FORMAT(*SN= *,F12.10,*     WI=  *,F5.3,*   RN=   *,I3)
```

```
      IF(SN.GT.WI) GO TO 11
      IA(RN)=1
C     AFTER TERM IS ASSIGNED TO RN TH DOCUMENT COUNT RELEVANCE
C     OF RN TO EACH CATEGORY
      DO 40 J1=1,4
      IF(SEQ(KK,J1).EQ.1) CIK(J1)=CIK(J1)+1
40    CONTINUE
C     PRINT 97,RN,(CIK(L),L=1,4)
 97   FORMAT(*  RN  = *,I5,4F5.1)
      KCON=KCON+1
      IF(KCON.LT.MI) GO TO 11
C
C     AFTER MI OCCURANCES OF TERM IS COMPLETED
C      COMPUTE RELEVANCE CONTRIBUTION OF I TH TERM
C      TO EVERY DOCUMENT
C     PRINT 255
255   FORMAT(* AFTER CONTRIBUTION TO RELEVANCE OF DOCUMENT TO CATEGORIES
     +BY THE ABOVE TERM *)
      DO 250 I6=1,M
      IF(IA(I6).EQ.0) GO TO 250
      DO 252 L=1,4
      IF(YKM(I6,L).EQ.-1) YKM(I6,L)=0.0
      YKM(I6,L)=YKM(I6,L)+CIK(L)/KCON
252   CONTINUE
C     PRINT 254,I6,(YKM(I6,L),L=1,4)
254   FORMAT(*  RN  = *,I5,4F8.4)
250   CONTINUE
      L1=L1+1
      NTERM(L1)=I
100   CONTINUE
      PRINT 349,K
349   FORMAT(//* CONTENT TERMS OF CATEGORY   *,I2)
      PRINT 348,(NTERM(I1),I1=1,L1)
348   FORMAT(20I3)
C        MAKE NTERM=0
      DO 347 I1=1,L1
347   NTERM(I1)=0
300   CONTINUE
C
C     COMPUTE CLASSIFICATION EFFICIENCY IN TERMS OF PRECISION
C     RECALL.
      DO 310 K=KNBE,KNLA
      DO 150 ITHRESH=5,13,2
      THRESH=.1*ITHRESH
      TRT=0
      KRT=0
      DO 50 J=1,M
      IF(YKM(J,K).LT.THRESH) GO TO 50
      TRT=TRT+1
C      FIND WHETHER J TH DOC. BELONGS TO CATEGORY K
      DO 23 JJ=1,16
      IF(J.GT.SEQ(JJ,5)) GO TO 23
19    IF(SEQ(JJ,K).EQ.1) GO TO 31
      GO TO 50
23    CONTINUE
      GO TO 50
```

```
      31    RRT=RRT+1
     400    FORMAT(/* RRT = *,F10.1)
      50    CONTINUE
            PRINT 400,RRT
            PRINT 430,TRT
     430    FORMAT(/* TRT = *,F10.1)
            IF(TRT.EQ.0) GO TO 310
            PR=RRT/TRT
            REC=RRT/CK(K)
            PRINT 111,PR,REC,THRESH,K
     111    FORMAT(/* PRECISION = *,F6.3,* RECALL =*,F6.3,*  THRESH= *,F4.2
           +,*  CATEGORY =*,I1)
      150   CONTINUE
      310   CONTINUE
            DO 104 L=1,50
            PRINT 103,L,(YKM(L,K),K=1,4)
            LCON1=0
      104   CONTINUE
      103   FORMAT(I3,2X,4F10.4)
            COUNT=0
            LCON=0
            KCON1=0
            KCON2=0
            TOT1=0
            TOT2=0
            DO 500 I=1,M
            CALL RANK(SEQ,4,I)
      500   CONTINUE
            PRINT *,KCON1,TOT1,KCON2,TOT2
            P1=KCON1/TOT1
            P2=KCON2/TOT2
            PRINT 501,P1,P2,COUNT
      501   FORMAT(*  P1  = *,F8.4,*  P2  = *,F8.4,* DOCS CONTAINING NO TERM *
           +,F8.1)
            PRINT 502,LCON,LCON1
      502   FORMAT(*  LCON  = *,I8,*   LCON1  =  *,I8)
            STOP
            END


            SUBROUTINE PERM(P,M,N)
      C
      C     THIS GENERATES BIT SEQUENCE P,M=SEQUENCE COUNT
      C     N=NUMBER OF BITS IN SEQUENCE
      C
            INTEGER P(4)
            MAX=2**N
            MM=MAX-M
            DO 1 I=1,N
            P(N-I+1)=MOD(MM,2)
            MM=SHIFT(MM,-1)
        1   CONTINUE
            RETURN
            END
```

```
      SUBROUTINE RANK(SEQ,N,II)
C
C     THIS GIVES RANKED OUTPUT OF A DOCUMENT FOR FOUR CATEGORIES
C     IT IS IN DECREASING ORDER OF RELEVANCE OF DOCUMENTS
C     TO CATEGORIES.KCON1 IS COUNTER FOR COUNTING NUMBER OF
C     CHANCES WHEN CORRECT CATEGORY AT TOP OF LIST.KCON2 COUNTS
C     WHEN FIRST TWO CATEGORIES(CORRECT) TOP THE LIST
C
C
      INTEGER CAT(4),SEQ(10,5)
      LOGICAL FINISH
      COMMON YKM(5000,4),KCON1,KCON2,TOT1,TOT2,COUNT,LCON,LCON1
      DO 3 J=1,4
3     CAT(J)=J
      DO 6 K=2,N
      FINISH=.TRUE.
      LP=N-K+1
      DO 5 L=1,LP
      IF(YKM(II,L).GE.YKM(II,L+1)) GO TO 5
      TEMP=YKM(II,L+1)
      YKM(II,L+1)=YKM(II,L)
      YKM(II,L)=TEMP
      TEMP=CAT(L+1)
      CAT(L+1)=CAT(L)
      CAT(L)=TEMP
      FINISH=.FALSE.
5     CONTINUE
      IF(FINISH) GO TO 11
6     CONTINUE
11    CONTINUE
C1    PRINT 10,(CAT(J),J=1,4)
10    FORMAT(4X,4I10)
C     PRINT 12,II,(YKM(II,J),J=1,4)
12    FORMAT(I4,4F10.4)
      IF(YKM(II,1).NE.-1) GO TO 19
      COUNT=COUNT+1
19    DO 23 L=1,16
      IF(L.EQ.16) RETURN
      IF(II.GT.SEQ(L,5)) GO TO 23
      IF(YKM(II,1).NE.-1) GO TO 20
      LCON=LCON+1
      IF(SEQ(L,CAT(1)).EQ.1.AND.SEQ(L,CAT(2)).EQ.1) LCON1=LCON1+1
      RETURN
20    IF(SEQ(L,CAT(1)).EQ.1) KCON1=KCON1+1
      TOT1=TOT1+1
      IF(YKM(II,2).EQ.0) GO TO 24
      IF(SEQ(L,CAT(1)).EQ.1.AND.SEQ(L,CAT(2)).EQ.1) KCON2=KCON2+1
      TOT2=TOT2+1
      GO TO 24
23    CONTINUE
24    CONTINUE
      RETURN
      END
```

```
      PROGRAM ATTRI(INPUT,OUTPUT,TAPE3)
C
C     SEQ IS MATRIX WHOSE FIFTH COLUMN CONTAINS CUMULATIVE
C     DOCUMENTS.THE FIRST FOUR COLUMNS DESCRIBE BEHAVIOUR OF
C     GIVEN DOCUMENT TO GIVEN CATEGORY COMBINATIONS.
C     O=NUMBER OF DIFFERENT TERMS AND RN IS PSEUDO-DOCUMENT
C     NUMBER.IA IS COLUMN VECTOR WHOSE ELEMENTS ARE 1 OR 0
C     IT IS 1 IF THE I TH TERM OCCURS IN THE DOCUMENT
C     NTERM KEEPS COUNT OF CONTENT-TERMS FOR GIVEN CATEGORY.
C     VECTOR CK COUNTS DOCUMENTS IN CATEGORY K AND CIK COUNTS
C     NUMBER OF DOCUMENTS OF CATEGORY K THAT CONTAIN TERM I.
C     YKM-DOCUMENT CATEGORY MATRIX WHOSE EACH COMPONENT IS
C     ATTRIBUTE NUMBER OF M TH DOCUMENT TO CATEGORY K. GIVEN
C     A DOCUMENT CONTAINS TERMS W1,W2...WN THE PROB. OF ITS
C     BELONGING TO CATEGORY K IS KNOWN AS ATTRIBUTE NUMBER.
C     LCON COUNTS DOCUMENTS HAVING NO CONTENT TERM AND DOCS
C     WHICH ARE IN SOME CATEGORY.
C     COUNT IS COUNTER FOR DOCS HAVING NO TERM FROM WHOLE
C     DATA BASE.
C
C
      INTEGER SEQ(16,5),O,RN
      DIMENSION IP(4),IA(5000),NTERM(500),CK(4),CIK(4)
      COMMON YKM(5000,4),KCON1,KCON2,TOT1,TOT2,COUNT,LCON,LCON1
      DATA M,N,O,PC,PN/5000,40000,7000, 0.2,0.7/
      READ*,KNBE,KNLA
      DO 5 I=1,16
      READ(3,29) SEQ(I,5)
29    FORMAT(I5)
      CALL PERM(IP,I,4)
      DO 3 J=1,4
      SEQ(I,J)=IP(J)
 3    CONTINUE
 5    CONTINUE
      PRINT 9,M,N,O,PC,PN
9     FORMAT(* M =*,I5,* N =*,I2,* O =*,I6,* PC =*,F5.2,* PN =*,F5.2//)
      PRINT 4,((SEQ(I,J),J=1,5),I=1,16)
 4    FORMAT(1X,4I1,5X,I10)
C      INITIALISE ATTRIBUTE NUMBER MATRIX
      DO 39 I=1,M
      DO 39 J=1,4
39    YKM(I,J)=-1
      DO 1 L=1,4
1     CK(L)=0
C     FIND TOTAL DOCUMENTS IN EACH CATEGORY
      DO 15 J=1,16
      DO 49 L=1,4
      IF(SEQ(J,L).NE.1) GO TO 49
      CK(L)=CK(L)+SEQ(J,5)-SEQ(J-1,5)
49    CONTINUE
15    CONTINUE
      AO=0
      A=1/(ALOG(AO)+.5772)
      DO 300 K=KNBE,KNLA
C      FIND TOTAL DOCS IN CATEGORY K
      TOTDC=CK(K)
```

```
      PRINT 108,K,TOTDC
108   FORMAT(//* TOTAL DOCS IN CATEGORY *,I1,* = *,F10.1)
      L1=0
C     GENERATE I TERM ASSOCIATIONS WITH DOCUMENTS
C
      DO 100 I=50,0
      DO 26 L=1,4
26    CIK(L)=0
C     COMPUTE TERM OCCURANCES AS PER ZIPF'S LAW
      MI=A*N/I
      KOUN=0
      IF(MI.LT.M) GO TO 7
      MI=M
 7    J=0
C      GENERATE RANDOM NUMBER RC BETWEEN 0 AND 1
      Y=.5757*I+(5**.5+1)/2
      CALL RANSET(Y)
      RC=RANF(YY)
C     PRINT 106,I,RC,Y
C106  FORMAT(*START   TERM *,I2,*      RC=  *,F12.10,*   Y  =*,F10.5)
      IF(RC.LE.PC) GO TO 27
      GO TO 100
 27   CONTINUE
C      GENERATE INTEGER NI BETWEEN 1 AND 4
      Y2=I**3+(5**.5+1)/2
      CALL RANSET(Y2)
      NI=RANF(YY)*4+1
      IF(NI.NE.K) GO TO 100
      DO 8 I5=1,M
C     INITIALISE VECTOR IA.
8     IA(I5)=0
C     PRINT 99,NI
99    FORMAT(*  NI  =*,I5)
C     PRINT 200,I
200   FORMAT(* COUNT OF RELEVANT DOC. TO EACH CAT.FOR TERM =*,I5)
      YI=I+(5**.5+1)/2
      CALL RANSET(YI)
 11   X=RANF(YY)
105   FORMAT(*X=  *,F12.10,*    RN=  *,I2)
      RN=X*M+1
C      PRINT 105,X,RN
      IF(IA(RN).EQ.1) GO TO 11
C      FIND DOC. RANGE IN WHICH RN LIES
      DO 21 L=1,16
      KK=L
      IF(RN.LE.SEQ(L,5)) GO TO 22
 21   CONTINUE
 22   CONTINUE
C     FIND CATEGORY NUMBER TO WHICH CONTENT TERM HAS MORE BIAS
      IF(SEQ(KK,NI).EQ.1) GO TO 30
      WI=0.15
      GO TO 35
 30   WI=.85
C      GENERATE RAND NUMBER BETWEEN 0 AND 1
 35   SN=RANF(YY)
C     PRINT 107,SN,WI,RN
```

```
107   FORMAT(*SN= *,F12.10,*      *I=  *,F5.3,*   RN=   *,I3)
      IF(SN.GT.WI) GO TO 11
      IA(RN)=1
C     AFTER TERM IS ASSIGNED TO RN DOCUMENT COUNT RELEVANCE
C     OF RN TO EACH CATEGORY
      DO 40 J1=1,4
      IF(SEG(KK,J1).EQ.1) CIK(J1)=CIK(J1)+1
40    CONTINUE
C     PRINT 97,RN,(CIK(L),L=1,4)
97    FORMAT(*  RN  = *,I5,4F5.1)
      KCON=KCON+1
      IF(KCON.LT.MI) GO TO 11
C     AFTER MI OCCURANCES OF TERM IS COMPLETED
C      COMPUTE ATTRIBUTE NUMBER CONTRIBUTION OF I TERM
C     TO EVERY DOCUMENT
C     PRINT 255
255   FORMAT(* AFTER CONTRIBUTION TO ATTRIBUTE NUMBER FOR DOCUMENT BY
     +THE ABOVE TERM *)
      DO 250 I6=1,M
      IF(IA(I6).EQ.0) GO TO 250
      DO 252 L=1,4
      IF(YKM(I6,L).EQ.-1) YKM(I6,L)=CK(L)/M
      YKM(I6,L)=YKM(I6,L)*CIK(L)/CK(L)
252   CONTINUE
C     PRINT 254,I6,(YKM(I6,L),L=1,4)
254   FORMAT(*  RN  = *,I5,4F8.4)
250   CONTINUE
      L1=L1+1
      NTERM(L1)=I
100   CONTINUE
      PRINT 349,K
349   FORMAT(/* CONTENT TERMS OF CATEGORY  *,I2)
      PRINT 348,(NTERM(I1),I1=1,L1)
348   FORMAT(20I5)
C        MAKE NTERM=0
      DO 347 I1=1,L1
347   NTERM(I1)=0
300   CONTINUE
C     COMPLTE CLASSIFICATION EFFICIENCY IN TERMS OF PRECISION
C     AND RECALL:
      DO 310 K=KNBE,KNLA
      DO 150 ITHRESH=1,41,10
      THRESH=.00001*ITHRESH
      IRT=0
      RRT=0
      DO 50 J=1,M
      IF(YKM(J,K).LT.THRESH) GO TO 50
      IRT=IRT+1
C      FIND WHETHER J TH DOC. BELONGS TO CATEGORY K
      DO 23 JJ=1,15
      IF(J.GT.SEG(JJ,5)) GO TO 23
19    IF(SEG(JJ,K).EQ.1) GO TO 31
      GO TO 50
23    CONTINUE
      GO TO 50
31    RRT=RRT+1
```

```
  400   FORMAT(/* KRT = *,F10.1)
   50   CONTINUE
        PRINT 400,RRT
        PRINT 430,TRT
  430   FORMAT(/* TRT = *,F10.1)
        IF(TRT.EQ.0) GO TO 310
        PR=RRT/TRT
        REC=RRT/CK(K)
        PRINT 111,PR,REC,THRESH,K
  111   FORMAT(/* PRECISION = *,F6.3,* RECALL =*,F6.3,*  THRESH= *,F10.5
       +,*  CATEGORY =*,I4)
  150   CONTINUE
  310   CONTINUE
        DO 104 L=1,50
        PRINT 103,L,(YKM(L,K),K=1,4)
  104   CONTINUE
  103   FORMAT(I2,2X,4F10.4)
        COUNT=0
        LCON=0
        KCON1=0
        LCON1=0
        KCON2=0
        TOT1=0
        TOT2=0
        DO 500 I=1,M
        CALL RANK(SEQ,4,I)
  500   CONTINUE
        PRINT*,KCON1,TOT1,KCON2,TOT2
        P1=KCON1/TOT1
        P2=KCON2/TOT2
        PRINT 501,P1,P2,COUNT
  501   FORMAT(*  P1   = *,F6.4,*  P2   = *,F6.4,*DOCS CONTAINING NO TERM =
       +  *,F5.1)
        PRINT 503,LCON,LCON1
  503   FORMAT(*  LCON     = *,I8,*    LCON1  =  *,I8)
        STOP
        END


        SUBROUTINE PERM(P,M,N)
C
C       THIS GENERATES BIT SEQUENCE P,M=SEQUENCE COUNT,
C       N=NUMBER OF BITS IN SEQUENCE
C
        INTEGER P(4)
        MAX=2**N
        MM=MAX-M
        DO 1 I=1,N
        P(N-I+1)=MOD(MM,2)
        MM=SHIFT(MM,-1)
    1   CONTINUE
        RETURN
        END
```

```
      SUBROUTINE RANK(SEQ,N,II)
C
C     THIS GIVES RANKED OUTPUT OF A DOCUMENT FOR FOUR CATEGORIES
C     IT IS IN DECREASING ORDER OF ATTRIBUTE NO. OF DOCUMENTS
C     TO CATEGORIES.KCON1 IS COUNTER FOR COUNTING NUMBER OF
C     CHANCES WHEN CORRECT CATEGORY AT TOP OF THE LIST.KCON2 COUNTS
C     WHEN FIRST TWO CATEGORIES(CORRECT) TOP THE LIST
C
C
      INTEGER CAT(4),SEQ(16,5)
      LOGICAL FINISH
      COMMON YKM(5000,4),KCON1,KCON2,TOT1,TOT2,COUNT,ICON,LCON1
      DO 3 J=1,4
3     CAT(J)=J
      DO 6 K=2,N
      FINISH=.TRUE.
      LP=N-K+1
      DO 5 L=1,LP
      IF(YKM(II,L).GE.YKM(II,L+1)) GO TO 5
      TEMP=YKM(II,L+1)
      YKM(II,L+1)=YKM(II,L)
      YKM(II,L)=TEMP
      TEMP=CAT(L+1)
      CAT(L+1)=CAT(L)
      CAT(L)=TEMP
      FINISH=.FALSE.
5     CONTINUE
      IF(FINISH) GO TO 11
6     CONTINUE
11    CONTINUE
C1    PRINT 10,(CAT(J),J=1,4)
10    FORMAT(4X,4I10)
C     PRINT 12,II,(YKM(II,J),J=1,4)
12    FORMAT(I4,4F10.4)
      IF(YKM(II,1).NE.-1) GO TO 19
      COUNT=COUNT+1
19    DO 23 L=1,16
      IF(L.EQ.16) RETURN
      IF(II.GT.SEQ(L,5)) GO TO 23
      IF(YKM(II,1).NE.-1) GO TO 20
      ICON=ICON+1
      IF(SEQ(L,CAT(1)).EQ.1.AND.SEQ(L,CAT(2)).EQ.1) LCON1=LCON1+1
      RETURN
20    IF(SEQ(L,CAT(1)).EQ.1) KCON1=KCON1+1
      TOT1=TOT1+1
      IF(YKM(II,2).EQ.0) GO TO 24
      IF(SEQ(L,CAT(1)).EQ.1.AND.SEQ(L,CAT(2)).EQ.1) KCON2=KCON2+1
      TOT2=TOT2+1
      GO TO 24
23    CONTINUE
24    CONTINUE
      RETURN
      END
```