DOCUMENT RESUME

ED 160 041	IR 005 877

AUTHOR	Kar, B. Gautam; White, Lee J.
TITLE	A Distance Measure for Automatic Sequential Document Classification.
INSTITUTION	Ohio State Univ., Columbus. Computer and Information Science Research Center.
SPONS AGENCY	National Science Foundation, Washington, D.C. Div. of Science Information.
REPORT NO	OSU-CISRC-TR-75-7
PUB DATE	Aug 75
GRANT	GN-36340
NOTE	168p.; For related document, see IR 005 876
EDRS PRICE	MF-$0.83 HC-$8.69 Plus Postage.
DESCRIPTORS	*Algorithms; *Automatic Indexing; *Bayesian Statistics; *Classification; Cluster Grouping; Data Bases; Documentation; Feasibility Studies; Mathematical Models; *Measurement Techniques; Probability; *Sequential Approach; Statistical Analysis
IDENTIFIERS	*Keywords

ABSTRACT

The feasibility of using a distance measure, called the Bayesian distance, for automatic sequential document classification was studied. Results indicate that, by observing the variation of this distance measure as keywords are extracted sequentially from a document, the occurrence of noisy keywords may be detected. This property of the distance measure has been utilized to design a sequential classification algorithm which works in two phases. In the first phase keywords extracted from a document are partitioned into two groups, the good keyword group and the noisy keyword group. In the second phase these two groups are analyzed separately to assign primary and secondary classes to a document. The algorithm has been applied to the SPIN data base, and very encouraging results have been obtained. Appendices include descriptions and mathematical models of (1) Bayesian distance and classification error, (2) Bayesian distance and alpha-j values, (3) Bayesian distance and keyword vectors, and (4) the classification algorithm. (Author/CMV)

(OSU-CISRC-TR-75-7)

A DISTANCE-MEASURE FOR AUTOMATIC SEQUENTIAL DOCUMENT CLASSIFICATION

by

B. Gautam Kar and L. J. White

Work performed under

Grant No. GN-36340

Office of Science Information Service

National Science Foundation

The Computer and Information Science Research Center

The Ohio State University

Columbus, Ohio 43210

August 1975

ABSTRACT

This research has investigated the feasibility of using a distance measure, called the Bayesian distance, for automatic sequential document classification. It has been shown that by observing the variation of this distance measure as keywords are extracted sequentially from a document, the occurrence of noisy keywords may be detected. This property of the distance measure has been utilized to design a sequential classification algorithm which works in two phases. In the first phase keywords extracted from a document are partitioned into two groups -- the good keyword group and the noisy keyword group. In the second phase these two groups of keywords are analyzed separately to assign primary and secondary classes to a document. The algorithm has been applied to the SPIN data base and very encouraging results have been obtained.

PREFACE

This report is the result of research supported in part by Grant Number GN 36340 from the Office of Science Information Service, National Science Foundation to the Computer and Information Science Research Center, The Ohio State University.

The Computer and Information Science Research Center of The Ohio State University is an interdisciplinary research organization which consists of the staff, graduate students, and faculty of many University departments and laboratories. This report presents research accomplished in cooperation with the Department of Computer and Information Science.

The research was administered and monitored by The Ohio State University Research Foundation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

The main purpose of document classification is to aid the process of information retrieval from an information system. Such a system may contain a collection of written texts, books, summaries, abstracts, titles and so on, each of which is considered to be a document. A user of this system typically formulates a request in natural language describing the subject area or areas in which he seeks information. After such a request is formulated, the document collection is searched and all items considered to be relevant to the user's needs are retrieved.

The task of identifying documents which are relevant to a given request is a complex one and is usually done by comparing the contents of a document with that of the search request. This means that each document in the collection has to be analyzed for its content and represented by means of a set of content identifiers. These content identifiers, also referred to as attributes, are usually a set of distinguishing words, phrases or sentences that together describe the complete area of discourse of the entire document collection. The process of analyzing a document and representing it in terms of one or more of the above types of attributes is known as content analysis. It assumes three basic forms--abstracting, indexing and classification. Abstracting procedures generally use sentences as content identifiers, while indexing and classification techniques use phrases and words.

As opposed to the abstracting operation, where the abstracts produced are stored sequentially by item identifier number, indexing and classification generally presume an additional operation of file inversion. The index term or classification codes are first ordered in some fashion and corresponding to each index term or classification code a list of document identifiers is maintained. This is called an inverted file. As a result, now instead of searching through an entire document collection, only the appropriate attribute lists in the inverted file need be examined in order to extract documents relevant to a given query. Thus, basically, both indexing and classification partition a data base into groups. There is, however, a major difference between the two.

The technique of indexing induces a partition on a document collection which is very fine in nature. The entire data base is divided into as many groups as there are attributes. Classification, on the other hand, obtains a coarser structure. The entire file of documents is broken up into a much smaller number of groups, where each group contains documents whose attributes are similar to each other. More formally, the basic document classification problem consists of categorizing a set of documents by assigning them to a reasonably small number of subpopulations in such a way that the members within each group are sufficiently alike to justify ignoring the individual differences between them.

Most content analysis methods in use today, including classification of large information systems, are manual in nature. In practice this has proved to be very time consuming and expensive to operate. Therefore it is desirable to automate the entire process of content analysis. This research concentrates on devising an efficient automatic technique for the classification aspect of content analysis.

In general automatic document classification requires the following sets of operations.

(i) A set of categories or classes which form a desired partition of the entire document base is chosen.

(ii) A set of keywords which are most representative of the documents in the collection is selected.

(iii) An algorithm is designed, which given the description of a document in terms of its keywords, will be able to assign that document to one or more of the predetermined set of classes.

This research assumes the presence of a predetermined set of categories and keywords. It concentrates mainly on the third point, viz., on the design of efficient methods for assigning a document to one or more categories based on its description.

The problem of automatic document classification has been an area of research for a long time. One of the earliest and most significant contribution was that of Luhn [18] who showed that a statistical analysis of the words in a document can provide some clues as to its content. This work led to the development of several automatic classification methods from the early 1960's to the early 1970's, a number of which will be discussed in some detail in Chapter II. The basic approach used by all these methods is such that each document must be read entirely before it can be assigned to some category.

This research assumes a different approach. The basic philosophy employed here is based on the premise that it may not be necessary to examine a document entirely before some indication of its subject content can be obtained. If documents can be suitably classified by examining only limited portions thereof, then considerable time and money spent on processing entire documents can be saved. This realization has led to the development and implementation of a basic sequential technique for automatic document classification. In this method keywords are extracted sequentially from a document, and at each stage a statistical prediction technique is used to determine whether or not the document can be classified. If not, then more of it is read. Details of the basic sequential method, its feasibility and its limitations are discussed in Chapter III. Specifically, it is pointed out that the basic sequential algorithm is insensitive to the occurrence of 'noisy' or inappropriate keywords and lacks the capability of systematically assigning a document to more than one category.

Chapter IV addresses itself to these two problems and discusses the need for a measure which will be able to identify clusters of similar keywords. Such a measure, called the Bayesian distance, is then defined on a vector space representation of keywords. Chapter V explores the possibility of using the Bayesian distance for the purposes of automatic document classification. It is shown that by studying the variation of its magnitude and direction as keywords are read from a document, noisy words may be isolated and clusters of similar keywords can be identified. These clusters of similar keywords are such that they relate the document to different classes.

Chapter VI uses the properties of the Bayesian distance to design a classification technique which is called the revised sequential algorithm. This algorithm operates by first identifying two groups of keywords--the good keyword set and the noisy keyword set. It then analyzes the group of good keywords to obtain a primary class for a document. Chapter VII deals with the design of a method which uses the Bayesian distance measure to analyze the keywords contained in the noisy group. If these keywords are such that they indicate an additional class to which the document may be assigned, then this class is denoted as the secondary class.

Chapter VIII presents a summary of the basic achievements of this research and identifies several related areas in which further research could be pursued.

CHAPTER II

A SURVEY OF AUTOMATIC DOCUMENT CLASSIFICATION

In the last chapter the problem of automatic document classification was introduced as one aspect of the larger problem of automatic content analysis. Automatic document classification was defined as the process of categorizing a set of documents by assigning them to a reasonably small number of subpopulations or groups. The number of such groups obtained is dictated by the requirements demanded of the designer of the classification system. A larger number will be needed when a very fine distinction is required between the documents. A small number suffices when the distinction need only be coarse. A combination of coarse and fine distinction may be obtained by designing a hierarchical system and then operating at a suitable level of the hierarchy.

This research assumes that the number of subpopulations or groups into which a document collection is to be partitioned is available. Each of these groups, also referred to as a category, denotes a distinct subject area and together they describe the entire area of discourse of the total document collection. The problem then is to design methods which will automatically examine each document and, based on its content, assign it to one or more of these categories.

The classification accuracy can then be estimated by comparing the results with those obtained by using manual methods.

In order that the outcome of this research may be evaluated, it is necessary to take a critical look at some of the more well known existing automatic classification systems. This chapter discusses the advantages and limitations of these systems. Since each of the systems discussed uses its own definitions and concepts, comparison becomes difficult unless a standardized set of definitions is used The following section attempts to do this briefly.

2.1 Basic Concepts and Definitions

Documents: For the purposes of this research a document will be considered to be any item in the form of an abstract, an article or any other coherent body of text. A document data base will be denoted by the set D where

$$D = \{d_1, d_2, \ldots, d_n\}$$

represents a collection of n documents.

Categories: Given a set of documents D, the number of groups into which the set is to be divided by the classification process is first determined. Each such group will be referred to as a category. The set of categories will be denoted by $C = \{C_1, C_2, \ldots, C_t\}$. The documents contained within each category will be sufficiently alike in their subject content to justify ignoring the individual differences between them.

Keywords: It was pointed out in Chapter I that in order to achieve classification each document in the set D has to be analyzed for its content and then represented by means of a set of content identifiers. These content identifiers, also referred to as attributes, are usually a set of distinguished words, phrases or sentences that together describe the complete area of discourse of the entire document collection. This research, like most other automatic document classification methods, uses a set of distinguished words to act as content identifiers. These distinguished words will be called keywords. Selection of an appropriate set of keywords is an area of research in its own right and has received a great deal of attention in the literature [22,32]. This research assumes that such a set is already available. It will be denoted by $K = \{k_1, k_2, \ldots, k_m\}$. Each keyword k_i contained in K relates a given document to one or more of the categories present in the set C defined above. The exact way in which this relationship between a keyword and a category is achieved will be discussed later in this chapter.

Using a set of documents, a set of categories and a set of keywords as the starting point, the following sections in this chapter discuss a number of automatic classification methods that are currently in use, or have been used in the past. Based on the general philosophy used in these classification methods, they have been divided into two broad groups, viz., statistical classification techniques and classification techniques based on clustering methods.

2.2 Statistical Methods of Automatic Document Classification

The possibility of characterizing the subject matter of a document by means of automatic content analysis was recognized in the early 1950's but it remained a relatively uncharted area until Luhn [18] showed that a statistical analysis of the words in a document would provide some clues as to its content. After his pioneering work several automatic document classification methods have been developed starting from the early 1960's to the early 1970's. The basic approach of all these methods is the same. Each document that is to be classified into one of the given categories is read entirely and all the keywords present in it are extracted. Using these keywords a prediction function is used to relate the document to each of the categories. The differences in the various methods lie in the nature of the prediction function that is used and the way the relationship between a category and a document is computed. These differences and similarities will be clarified when the various methods are discussed in some detail in the following sections.

2.2.1 Bayesian Technique

Maron [19] applied a statistical method to the problem of automatic classification which involved:

(i) the determination of certain probability relationships between individual keywords and subject categories, and

(ii) the use of these relationships to predict the category to which a document belongs by using Bayes rule.

The prediction method that he used was as follows: Given a set of categories $C_1, C_2, \ldots, C_t$ and a document which contains only one clue word k_i, the probability that the document belongs to each of the categories is computed using

$$P(C_j/k_i) = \frac{P(C_j)\cdot P(k_i/C_j)}{P(k_i)} \qquad (2.1)$$

Extension to the case of documents containing more than one clue word was made assuming keyword independence.

For experimentation purposes Maron chose 405 abstracts of computer literature published in the IRE Transactions on Electronic Computers, March 1959. He selected 32 categories manually, and used 260 of the 405 abstracts as sample documents from which keyword frequency statistics were obtained. Using a set of 90 keywords he achieved a classification accuracy of about 50% over the entire set of documents.

The classification techniques to be described in this research bear resemblance to Maron's method, in that the keyword class frequencies are calculated using sample documents and the a posteriori probabilities of the classes after a number of keywords are read and computed using Bayes rule. The basic difference lies in that this research hypothesizes that the keywords present in a document need not be examined in toto before class membership can be determined, but need only be read sequentially until a decision is reached. The philosophy behind such a sequential technique will be explicated in greater detail in Chapter III.

2.2.2 Techniques Based on Matrix Manipulation

Subsequent to the publication of Maron's work, Borko [3] devised a technique for automatic document classification based on factor analysis using Maron's 405 abstracts of computer literature. By means of a computer program, counts were made of the number of times each of the 90 keywords occurred in each of the documents in a 260 document sample set. Using this data a 90 x 90 keyword correlation matrix was derived. This matrix was then factor analyzed, as a result of which 21 subject categories were identified. A prediction technique based on the keyword frequencies in the documents and their factor loadings was developed. Suppose T_i denotes the number of occurrences of keyword k_i in a document, L_{ij} denotes the normalized factor loading of keyword k_i for category C_j. Then a value P_j is calculated for each category C_j as follows:

$$P_j = L_{1j} T_1 + L_{2j} T_2 + \dots + L_{mj} T_m . \qquad (2.2)$$

The document is classified in a class having the highest value of P_j. Using this technique a classification accuracy of about 48% was obtained.

At about the same time as Borko, Williams [29,30] devised a classification technique based on discriminant analysis. Instead of computing factor loadings, he computed a discriminant coefficient for each keyword and category. Using a set of 420 solid state abstracts published by the Cambridge Communications Corporation and a set of 48 keywords, he first classified 120 documents manually into three categories. The frequency of occurrence of a keyword in each document belonging to a given category is then empirically obtained. Using these frequencies two m x m matrices are computed as follows:

W ≡ a matrix whose elements are the pooled within-category sum of squares

A ≡ a matrix whose elements are the among-category sum of squares.

Suppose now a document contains words $k_{i_1}, k_{i_2}, \ldots, k_{i_s}$ whose mean frequencies are given by $x_1, x_2, \ldots, x_s$, then a prediction function of the following form is used:

$$X = B_1x_1 + B_2x_2 + \ldots + B_sx_s \qquad (2.3)$$

where the coefficients $B_1, B_2, \ldots, B_s$ are obtained by solving the equation

$$|W^{-1} A - \lambda I| = 0. \qquad (2.4)$$

One of the eigenvalues λ is chosen to give the best discrimination between the categories. The eigenvector corresponding to this λ then provides the set of coefficients $B_1, B_2, \ldots, B_s$. The discriminant score obtained from equation (2,3) is then the basis of assigning a test document to one of the categories. This method achieved a classification accuracy of about 75% on the set of 420 solid state abstracts.

Both the methods discussed in this section suffer from the same disadvantage. They require the inversion of matrices to compute the coefficients which relate a keyword to a category. For a sizable set of keywords, say 500, these methods become impracticable. More specifically, the increase in storage and time required by these methods is proportional to the square of the number of keywords.

It is clear from this discussion that methods which are dependent on inverting matrices whose dimensions depend on the size of the keyword set are impractical. The next section discusses a few methods of automatic document classification which essentially deal with matrices whose sizes depend on the number of documents in the collection. These are generally known as clustering techniques.

2.3 Automatic Document Classification Based on Clustering Techniques

The theory of clustering deals with the problem of finding natural groupings in a set of data. These natural groupings are obtained based on the similarity in the attributes of the data elements. An excellent description of various clustering techniques that have evolved over the years can be found in the book by Sokal and Sneath [24]. In this section we will discuss the basic philosophy of clustering and show how it has been applied for use in automatic document classification.

The starting point for most clustering algorithms is the similarity matrix. Let $D = \{d_1, d_2, \ldots, d_n\}$ be the set of documents and $K = \{k_1, k_2, \ldots, k_m\}$ be the set of keywords under consideration. Each document d_i in D is read entirely and all keywords occurring in it are extracted. An nxn matrix Q is obtained such that

$Q(i,j) = Q(j,i)$ = number of keywords occurring in common between documents d_i and d_j.

The similarity matrix S can now be directly obtained from the matrix D by normalizing each of its elements by using a standard similarity measure. There are several such measures that have been used in the literature and the one to be used for a particular implementation depends on the user requirements. One that has been used very widely is the Tanimoto similarity measure [25] which is as follows. Let S be of dimension nxn. Then each element S(i,j) is calculated as follows:

$$S(i,j) = \frac{Q(i,j)}{Q(i,i)+Q(j,j)-Q(i,j)} \qquad (2.5)$$

As can be seen, this similarity matrix is symmetric in nature, the diagonal entries being all equal to unity. Therefore during use in clustering only the upper or the lower triangular portion of the matrix needs to be stored.

Using this similarity matrix an initial set of clusters is first obtained. This is usually done by defining a threshold parameter T such that two documents d_i and d_j are assigned to the same cluster if S(i,j) exceeds the value of T. It is clear that as T is increased two documents should have a greater number of keywords in common to be placed into the same cluster.

Several well known methods follow this general procedure. Their basic philosophy is to represent each document as the vertex of an undirected graph and then find connected components in this graph. The next section examines their methods briefly.

2.3.1 Graph Theoretic Document Clustering Methods

In order that this section may be clearly understood some elementary concepts in graph theory need to be defined.

Graph: A graph, $G = (V,E)$, consists of a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $E = \{e_1, e_2, \ldots, e_m\}$ such that each edge e_k is identified with an unordered pair (v_i,v_j) of vertices. The vertices v_i,v_j associated with edge e_k are called the end vertices of e_k. The edge e_k is said to be incident to v_i and v_j.

Subgraph: A graph g is said to be a subgraph of graph G if all the vertices and all the edges of g are in G, and each edge of g has the same end vertices in g as in G.

Path: A path is a finite alternating sequence of vertices and edges $v_{i_1} e_{j_1} v_{i_2} e_{j_2} v_{i_3} \ldots v_{i_{n-1}} e_{j_{n-1}} v_{i_n}$ such that

i) no edge or vertex appears more than once in the sequence, and

ii) each edge is incident with the vertices preceding and following it in the sequence.

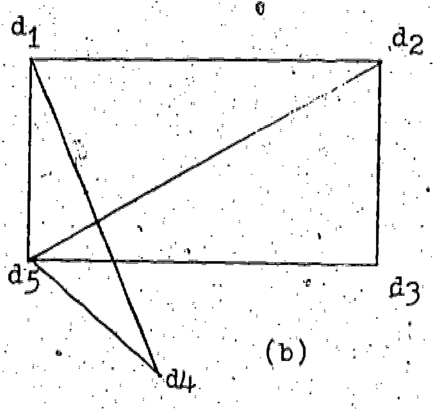
Connected Graph: A graph G is said to be connected if there is at least one path between every pair of vertices in G.

Complete Graph: A graph in which there exists an edge between every pair of vertices is called a complete graph.

Each document is represented by a vertex and there is an edge between two vertices d_i, d_j if $S(i,j)$ exceeds a threshold value T. An example is shown in Figure 2.1.

$$\begin{array}{c|ccccc|} & d_1 & d_2 & d_3 & d_4 & d_5 \\ \hline d_1 & . & 1 & 0 & 1 & 1 \\ d_2 & & . & 1 & 0 & 1 \\ d_3 & & & . & 0 & 1 \\ d_4 & & & & . & 1 \\ d_5 & & & & & . \\ \hline \end{array}$$

(a)



(b)

Figure 2.1 Similarity Matrix and Its Associated Graph

Using this representation Bonner [1] obtains all the complete subgraphs of the graph. Each of these subgraphs represents a set of documents which belong to one category. Unfortunately, some graphs have an exponential number of such subgraphs and, in that case, such a method becomes much too time consuming to be practical.

A variant of this method is to obtain the connected subgraphs instead of the complete subgraphs. This technique is used by Van Rigsbergen [26]. This is easier to do but it results in clusters where the documents within a cluster are not as closely related as in Bonner's method.

The methods discussed in this section not only require the computation of a similarity matrix but also a graph connectivity matrix. As larger and larger document collections are considered the storage space and computation time required increase as the square of the number of documents in the collection. Even for a 1,000 document collection, the storage and time required for these methods may become prohibitive. Some methods achieve a savings in storage space and computation time by eliminating the need for a graph-theoretic representation of the similarity matrix. Instead they use the method of centroids to assign documents into clusters. The feasibility of using such a technique has been investigated by various researchers and the following section briefly discusses some of the approaches taken.

2.3.2 Document Clustering by Finding Centroids

In this technique every document is examined serially and some documents are chosen to be centroids of the clusters if they satisfy certain criteria. For example, Rocchio's method [21] determines whether a document should be the centroid of a cluster by testing if there are sufficiently many documents located in its proximity.

Specifically, a document is a possible centroid only if there are at least n_1 documents which have a similarity of at least s_1, and n_2 documents which have a similarity of at least s_2 with the given document, where n_1, n_2, s_1 and s_2 are parameters which can be varied to obtain a different number of clusters.

Dattola's method [5] further simplifies this process by dividing the total document collection into an arbitrary number of groups. Each group is represented by a centroid whose dimension is equal to the number of keywords in the system. Each element of this vector is a quantity proportional to the frequency of occurrence of a keyword in all documents of the given group. Every document in the collection, each of which is also represented by such a vector, is now compared with each centroid and reassigned to the group to which it is the closest. The process is repeated until identical sets of clusters are obtained in successive iterations.

Both of the above techniques have been implemented in the classification phase of the SMART information storage and retrieval system [23]. They have been used to classify a subset of approximately 200 documents from the Cranfield collection [4]. Results have indicated that their classification is very dependent on the order in which the documents are examined. Dattola's method, even though it is faster than that of Rocchio, was found to be extremely dependent on the initial set of clusters that are chosen arbitrarily.

A major disadvantage of both these techniques is that the operations must be performed on the entire document collection to obtain satisfactory classification results. Therefore, for a collection whose size may change dynamically, these methods are generally not applicable.

Recently Yu [31] has proposed a technique which alleviates this problem to a certain extent. It is a very novel approach because instead of performing the clustering on a set of documents, it is performed on a set of user queries. First a representative set of queries is obtained. These keyword rich queries are then clustered using a standard technique. Each of these query clusters are now considered separately and every document relevant to all the queries in a given cluster is retrieved and placed in one group. Thus clusters in the query space are made to induce clusters in the document space.

This technique works well when a data base has associated with it a representative query set. However, for a document collection for which no such set exists, it would need considerable modification.

2.4 Limitations of Present Classification Methods

The techniques of document classification discussed in the previous sections are really representative of a wide spectrum of contemporary methods. They are representative in that they incorporate the same basic philosophy underlying the other methods. Each has certain advantages and disadvantages which have been discussed earlier. In this section several major philosophical points related to these techniques will be noted and it will be outlined how the classification algorithms developed in this research address themselves to these points.

The following observations can be made about the classification techniques that we have discussed.

(i) Techniques which use a similarity matrix are not practical for large data bases using existing facilities because of the storage space (both core and secondary storage) required to store such a matrix and the time required to process it.

(ii) Another disadvantage of using a similarity matrix for classification is that the entire document set has to be available before the classification process can be initiated. In other words, the document collection must be static in nature.

(iii) Most of the methods that we have discussed read each document of a collection completely before attempting classification. Given a keyword set $K = \{k_1, k_2, \ldots, k_m\}$, each document prior to classification is represented by an m-dimensional vector which depicts the presence or absence of each of the m keywords from the set K.

(iv) Most of the methods discussed work with a small keyword set. Each keyword in this set is related to a particular category in a binary fashion, i.e., it either belongs to a category or it does not. Given a keyword set $K = \{k_1, k_2, \ldots, k_m\}$ and a category set

$C = \{C_1, C_2, \ldots, C_t\}$, the set K is divided into t groups of keywords $G = \{g_1, g_2, \ldots, g_t\}$. Each group g_i contains keywords which are indicative of a category C_i. If a keyword k_i is present in two groups g_i and g_j, then it is considered to be equally indicative of classes C_i and C_j.

The algorithms developed in this research are based on the philosophy that an entire document need not be examined before a decision regarding its class membership can be made. As each keyword contained in a document is examined, information regarding its class membership is obtained. After a certain portion of the document has been read, any new keyword extracted from it may give only marginal information about its class membership. When this happens, it may be more efficient to stop reading the document any further and classify it into a category which is most appropriate at that point. In other words, the classification algorithms developed in this research process the words sequentially in a succession of stages. Each stage involves reading the document and extracting a fixed number of keywords. If at any stage a decision can be made about which class should be assigned to the document, then the process stops; otherwise it continues to the next stage. The motivation behind such a sequential technique is provided in the early section of Chapter III.

Another major difference between the classification algorithms that have been discussed in this chapter and those developed for this research is in the representation of keywords. Instead of assuming that a keyword is either indicative of a category or it is not, each keyword is represented by a set of values which essentially indicate different degrees to which a keyword is related to a given category. More specifically each keyword k_j is represented by t probability values $[p_1, p_2, \ldots, p_t]$, where p_i represents the probability $P(k_j/C_i)$, i.e., the probability that a document which belongs to category C_i will contain keyword k_j. Chapter III indicates a method by which these probabilities may be calculated.

Such a representation of keywords has led to the development of a method which treats a keyword as a t-dimensional probability vector. A document therefore can be represented as a cluster of points in a t-dimensional vector space. Chapter IV elaborates on this concept and shows how clusters of keywords which relate the document to different categories may be isolated. It is also shown how certain keywords which mislead the classification algorithm into placing a document in a wrong category can be isolated as 'noisy' keywords.

CHAPTER III

THE SEQUENTIAL CLASSIFICATION METHOD

As pointed out in section 2.5 of the previous chapter, most classification algorithms in use today extract all the concepts or attributes present in a document before initiating classification. All the concepts present in a document need not always be measured before its subject content can be determined. This observation forms the basis of the sequential classification method that will be discussed in this chapter.

3.1 Sequential Decision Model for Pattern Classification

The problem of automatic document classification can be likened to the classical problem of statistical pattern recognition where there exists a set of categories, and each category is characterized by a set of attribute-value pairs. These characteristic attribute-value pairs are obtained from a set of training patterns each of which is known to belong to a certain category. When a test pattern arrives, measurements are made on the attributes or features contained in this test pattern, and based upon these measurements and a decision criterion, it is classified into one or more of the existing categories. For the problem of document classification the categories are the various subject classes which describe the area of discourse of a collection of documents. The attributes are keywords present in the document collection and the training patterns are preclassified sample documents from which characteristic keywords are obtained for the various classes.

With this analogy in mind we can now formulate a basis for sequential document classification techniques based on the theory of sequential pattern recognition of Wald [27] and Fu [12]. Suppose we have a set of categories $C = \{C_1, C_2, \ldots, C_t\}$ and a set of features $K = \{k_1, k_2, \ldots, k_m\}$ describing these categories. In non-sequential classification theory all the N features present in a test pattern are observed by the classifier at one stage. This might prove to be impractical if the cost of making feature measurements is taken into consideration. Instead only as many features may be measured as are needed to classify the pattern with a given classification accuracy. Besides, after a certain point more feature measurements will not necessarily increase classification accuracy. A trade-off between the error of misclassification and the number of features to be measured can be obtained by taking feature measurements sequentially and terminating the sequential process when a sufficient or desirable accuracy is obtained.

If there are two pattern classes C_1 and C_2 to be recognized, then at the n^{th} stage of the sequential process, that is, after the n^{th} feature measurement is made, the classifier computes the sequential probability ratio given by equation (3.1).

$$\lambda_n = \frac{P_n(X/C_1)}{P_n(X/C_2)} \qquad (3.1)$$

where $P_n(X/C_i)$, $i = 1, 2$ represents the multivariate conditional probability density function $P_n(X_1, X_2, \ldots, X_n/C_i)$, and $X_1, X_2, \ldots, X_n$ are the features that have been measured so far, that is, when the end of the n^{th} stage is reached. The λ_n computed by equation (3.1) is then compared with two stopping boundaries A and B. If $\lambda_n \geq A$ then the decision is $X \sim C_1$ and if $\lambda_n \leq B$ then the decision is $X \sim C_2$. If $B < \lambda_n < A$, then an additional feature measurement is made and the process proceeds to the $(n+1)^{st}$ stage. The two stopping boundaries are related to the misclassification error probabilities by equations (3.2) and (3.3).

$$A = \frac{1-e_{21}}{e_{12}} \qquad (3.2)$$

$$B = \frac{e_{21}}{1-e_{12}} \qquad (3.3)$$

where e_{ij} is the probability of deciding $X \sim C_i$ when actually $X \sim C_j$ is true. It has been shown by Wald [27] that the sequential probability ratio test (SPRT) is optimal in the case of two pattern classes, that is, for a given e_{12} and e_{21} there is no other procedure yielding a smaller average number of feature measurements than SPRT. For more than two pattern classes, say $C_1, C_2, \ldots, C_t$, a generalized sequential probability ratio test (GSPRT) can be devised. This is an extension of the two class case, but the optimality property pointed out above no longer holds. Fu [12], however, notes that in most cases it can be shown to be close to optimal.

Once again at the n^{th} stage the following statistic is computed

$$U_n(X/C_i) = \frac{P_n(X/C_i)}{[\prod_{j=1}^{t} P_n(X/C_j)]^{1/t}}, \quad j = 1, 2, . . ., t. \qquad (3.4)$$

$U_n(X/C_i)$ is then compared with the stopping boundary of the i^{th} pattern class C_i and the decision procedure is to reject the class C_i from further consideration if

$$U_n(X/C_i) < A(C_i), \quad i = 1, 2, ..., t,$$

where $A(C_i)$ is the stopping boundary for class C_i. At each stage one or more pattern classes may be rejected from consideration and a new set of generalized sequential probability ratios may be computed. The pattern classes are rejected sequentially until only one is left, which is accepted as the recognized class. In some practical cases, however, the number of feature measurements may become extremely large and it may become necessary to truncate the sequential process at a predetermined stage.

3.2 Sequential Classification Technique for Documents

As pointed out in the previous section the problem of automatic document classification is very similar to the classical pattern recognition problem. Given a document collection a number of different subject areas describing the total area of discourse of the collection is identified. Each of these subject areas denotes a different category. A test document, like a test pattern, is now examined to determine the category to which it belongs.

There are, however, several aspects in which the two applications differ. The problem of pattern recognition can usually be formulated by using rigorous mathematical techniques because the probability density functions on the feature space are usually known or can be estimated. In the case of document classification such rigor is lacking. The features utilized in the case of pattern recognition now take the form of keywords. Since keywords represent ideas and not numerical quantities, the decision as to which keywords best represent any category is always subjective. Also in most cases, exact probability distributions of keywords over the categories they describe cannot be obtained. This section describes a sequential document classification procedure which is based on the GSPRT outlined in the previous section.

As in the case of pattern recognition we have the following predetermined items to work with:

<u>category set</u>: $\{C_1, C_2, \ldots, C_t\}$. Each category C_i denotes a subject area, and taken together, all these subject areas describe the complete area of discourse of the document collection. For instance, if one is dealing with a set of scientific documents, the subject areas might be Mathematics, Physics, Chemistry, Biology, etc.

<u>keyword set</u>: $\{k_1, k_2, \ldots, k_m\}$. These correspond to the attribute set of a pattern recognition problem. Each keyword present in a document relates it to one or more of the subject areas chosen above.

keyword - category probability matrix: In a sequential pattern recognition problem, after a feature X is measured, the computation of the sequential probability ratio test requires the availability of the conditional probability densities $P(X/C_1)$, $P(X/C_2)$, ..., $P(X/C_t)$. In the context of document classification, these are the probability values $P(k_i/C_1)$, ..., $P(k_i/C_t)$, where now the features are represented by keywords. Instead of being probability densities, these are discrete probability values calculated from sample documents. Thus the matrix CP shown in Figure 3.1 is required to compute the probability ratios at each stage where $P(k_j/C_i)$ is the probability of keyword k_j being present in a document which belongs to category C_i.

$$CP = \begin{array}{c|ccc:cc:c}
 & k_1 & k_2 & \cdots & k_j & \cdots & k_m \\ \hline
C_1 & & & & & & \\
C_2 & & & & & & \\
\vdots & & & & & & \\ \hdashline
C_j & & & & & & \\
\vdots & & & & p(k_j/C_i) & & \\ \hdashline
\vdots & & & & & & \\
C_t & & & & & & \\
\end{array}$$

Figure 3.1 Conditional Probability Matrix

The method used to calculate this conditional probability matrix will be described later in the chapter. Classification of a document is then carried out as follows:

The document is read until one or more keywords are found. At each stage, i.e., after reading each keyword, one of the following three decisions is made:

(i) the document is classified into one of the t categories,

(ii) the document is termed unclassifiable, or

(iii) more of the document is read because neither of decisions (i) or (ii) is taken.

Let us turn to a consideration of the probability ratio test and the prediction methods to be used. Suppose we are at the n^{th} stage of the sequential process, i.e., n keywords have been read and they are $W_1, W_2, \ldots, W_n$, where each W_i is a member of the set $K = \{k_1, k_2, \ldots, k_m\}$. The probability $P(W_1, W_2, \ldots, W_n/C_j)$ that the sequence of words $W_1, W_2, \ldots, W_n$ will be in a document that belongs to category C_j has to be calculated. This will be computed by using the probability estimate $P(W_i/C_j)$ obtained from the matrix CP discussed earlier. Since here we are estimating the probability of finding a string of keywords in a document which belongs to a given category, an issue that should be discussed is whether or not the keywords should be considered to be dependent or independent of each other. The next section briefly addresses this problem.

3.2.1 Calculation of $P(W_1,W_2,\ldots,W_n/C_j)$

The calculation of $P(W_1,W_2,\ldots,W_n/C_j)$ can be based on the premise that the keywords are dependent on each other. A prediction formula which takes into account the dependence of keywords is presented in equation (3.6) below. It should be noted here that this is just one of the many ways in which the effect of keyword dependence may be captured. Using conditional probabilities and noting that

$$P(A,B/C) = P(A/C) \cdot P(B/A,C) \qquad (3.5)$$

we can say

$$P(W_1,W_2,\ldots,W_n/C_j) = P(W_1/C_j)\cdot P(W_2/W_1,C_j)\,..\,P(W_n/W_{n-1},\ldots,W_1,C_j). \qquad (3.6)$$

Let us isolate any one term from the right hand side of equation (3.6), say $P(W_i/W_{i-1},\ldots,W_1,C_i)$. This is the probability that a document which belongs to category C_i and contains the keywords $W_1,W_2,\ldots,W_{i-1}$ will also contain keyword W_i. Since each W_j in $W_1,W_2,\ldots,W_{i-1}$ can be any keyword in the set K defined before, this probability has to be calculated for m^{i-1} different cases, where m is the number of keywords in set K. Considering the complexity of this task, a simplifying assumption is necessary. Fried [11], in presenting his classification model, simplified equation (3.6) above by assuming that a keyword is independent of all others that precede it in the document. As we shall see later, the sequential probability ratio test does not depend on the assumptions of exactly how the calculation of $P(W_1,W_2,\ldots,W_n/C_j)$ is made. Hence if the classification algorithm performs well under the assumption of keyword independence, which is a worst case assumption, it should do at least as well when keyword dependence is considered. Thus using the assumption that keywords are independent of each other we have

$$P(W_1,W_2,\ldots,W_n/C_j) = \prod_{i=1}^{n} P(W_i/C_j) \qquad (3.7)$$

3.2.2 Probability Ratio Test and Stopping Boundary

After the n keywords $W_1, W_2, \ldots, W_n$ have been read and $P(W_1,W_2,\ldots,W_n/C_j)$ for $j = 1,2,\ldots,t$ has been calculated at the end of the n^{th} stage of the sequential process, a probability ratio test has to be performed to predict the class to which the document may be assigned. For each j and for each fixed n, let $P_n^*(W_1,\ldots,W_n/C_j)$ be a probability distribution over the sequence $W_1,W_2,\ldots,W_n$. Then for each j this is a discrete distribution over m^n points where m is the total number of keywords in set K. A sequential decision rule can then be formulated as in GSPRT, by computing the ratio

$$\alpha_j = \frac{P(W_1,W_2,\ldots,W_n/C_j)}{P_n^*(W_1,W_2,\ldots,W_n/C_j)} \qquad (3.8)$$

Fried and his coworkers [11] addressed themselves to this problem and formulated the following decision rule.

At the n^{th} stage of the sequential process the ratio α_j is computed for each value of j ranging from 1 to t. This α_j is then compared with a predetermined threshold value α. As long as more than one α_j is greater than α, the process moves on to the $(n+1)^{st}$ stage. If at any stage only one α_j is greater than α and all the others are less than α, then the process terminates and the document is classified in class C_j for which α_j is greater than α.

Fried then studied various forms of P_n^* that could be used to compute the probability ratio. The following forms of P_n^* were investigated:

(i) $P_n^* = \frac{1}{m^n}$

(ii) $P_n^* = \frac{(m-n)!}{m!}$

(iii) $P_n^*(W_1,W_2,\ldots,W_n/C_j) = \sum_{j=1}^{t} P_n(W_1,\ldots,W_n/C_j)P(C_j)$

where $P(C_j)$ is the a priori probability of class C_j. P_n^* given by (i) and (ii) is independent of the keywords and depends only on n, the stage of the sequential process. In their experiments they found that (iii) gave the best results in terms of the number of stages that are required by the sequential proces before a document can be classified.

In this research the sequential probability ratio is computed by using Bayes formula for conditional probability. It can be derived as follows.

Let the keywords that have been read at the n^{th} stage of the sequential process be $W_1,W_2,\ldots,W_n$, where each W_i is a member of the keyword set $K = \{k_1,k_2,\ldots,k_m\}$. Let $P(C_j)$ be the a priori probability of a class before any keyword has been read. Then for each class C_j the probability $P(W_1,W_2,\ldots,W_n/C_j)$ can be computed as before. Then the a posteriori probability of class C_j after $W_1,W_2,\ldots,W_n$ has occurred is given by

$$P(C_j/W_1,W_2,\ldots,W_n) = \frac{P(C_j)\cdot P(W_1,W_2,\ldots,W_n/C_j)}{\sum_{i=1}^{t} P(C_i)\cdot P(W_1,W_2,\ldots,W_n/C_i)} \quad (3.9)$$

This is the probability that a document containing keywords $W_1,W_2,\ldots,W_n$ belongs to class C_j. For simplicity $P(C_j/W_1,W_2,\ldots,W_n)$ will be denoted by α_j. The value of α_j is now compared with a preset threshold value α, as in the Fried model. Based on this comparison one of the following two decisions is made:

(i) if only one $\alpha_j \geq \alpha$ then the document is classified in the corresponding class C_j;

(ii) if more than one α_j's are such that $\alpha_j \geq \alpha$, the process moves on to the $(n+1)^{st}$ stage, i.e., one more keyword is read; if the document does not contain any more keywords then it is considered unclassifiable.

In case (ii), those classes for which the α_j values are less than the threshold are deleted from consideration and only the ones whose α_j values are greater than or equal to the threshold are retained. This is mainly done in the interest of reducing computation time as will be pointed out in section 3.3 when implementation of the sequential technique is discussed.

3.2.3 The Parameters T and R

In the actual implementation of the sequential method, the probability ratio test is not necessarily computed after every keyword. This is because in many instances the very first keyword may be a strong indicator for a particular class and a poor one for the rest of the classes. This might lead to a precipitous decision at a very early stage in the sequential process. To avoid this a parameter T is introduced such that no ratio is computed and no decision regarding the classification of a document is made until at least T keywords are read.

Again it might be computationally wasteful, depending on the quality of the keywords and the nature of the data base, to compute the α_j values after every keyword subsequent to the first T keywords. To control this, another parameter R is introduced so that after the first T keywords, the α-tests are conducted only after groups of R keywords have been read. At the end of the first stage of the sequential process, T keywords have been read, and at the end of the second stage T+R keywords have been read and so on until at the end of the n^{th} stage [T+(n-1)R] keywords have been read. Depending on the data base under consideration T and R may be varied to obtain the desired classification accuracy.

3.3 Calculation of the Conditional Probability Matrix CP

As indicated in section 3.2, the sequential classification technique assumes that the keyword-class conditional probabilities are available for all possible keyword-class pairs. This is represented as a conditional probability matrix CP, each element of which is the probability of a keyword occurring in a document which belongs to a particular class. We will now consider how to calculate the elements of such a matrix.

Given a set of documents, a set of classes $C_1, C_2, \ldots, C_t$, and a set of keywords $K = \{k_1, k_2, \ldots, k_m\}$, the problem is to compute $P(k_i/C_j)$

for each keyword-class pair. To do this, all the available documents are divided into two sets called the sample and the test documents. We assume that each document in the sample set is already classified into one or more classes manually. Keyword frequency tables by classes can then be prepared using the sample set. Let D_j represent the subset of the sample documents associated with class C_j, and r_{ix} denote the number of occurrences of keyword k_i in document d_x. Then the frequency of keyword k_i given that a document is in class C_j is

$$f(k_i/C_j) = \sum_{d_x \varepsilon D_j} r_{ix} \qquad (3.10)$$

These frequencies are calculated for each keyword and stored in a frequency table. Then each element of the CP matrix is calculated as follows:

$$P(k_i/C_j) = \frac{f(k_i/C_j)}{\sum_{s=1}^{m} f(k_s/C_j)} \qquad (3.11)$$

where m is the number of keywords. For any class C_j let $K_j=\{k_{j1},k_{j2},\ldots,k_{jp}\}$ denote the set of all keywords that have occurred in at least one sample document belonging to class C_j. Then if we sum both sides of equation (3.11) over all keywords present in K_j we have

$$\sum_{k_i \varepsilon K_j} P(k_i/C_j) = \frac{\sum_{k_i \varepsilon K_j} f(k_i/C_j)}{\sum_{s=1}^{m} f(k_s/C_j)} . \qquad (3.12)$$

Now, $\sum_{k_i \epsilon K_j} f(k_i/C_j) = \sum_{s=1}^{m} f(k_s/C_j)$ because

for any keyword k_i contained in K but not contained in K_j, $f(k_i/C_j)$ is equal to zero. Therefore $\sum_{k_i \epsilon K_j} P(k_i/C_j)=1$. Hence the set of values $\{P(k_{j1}/C_j),\ldots,P(k_{jp}/C_j)\}$ is a probability set.

Since during classification these quantities do not change, they need to be calculated only once and stored. The a priori probability of a class C_j, denoted $P(C_j)$, prior to the initiation of the algorithm is calculated as

$$P(C_j) = \frac{\sum_{i=1}^{m} f(k_i/C_j)}{\sum_{C_y \epsilon C} [\sum_{i=1}^{m} f(k_i/C_y)]} . \qquad (3.13)$$

Assume that a keyword k_i occurs only in sample documents belonging to a given class C_j. Then all values $P(k_i/C_p)$ such that p is not equal to j will be zero. Now assume that a test document is being classified by the sequential procedure and at a stage n keyword k_{i_n} occurs in this document. Let the n-1 previous keywords be denoted by $k_{i_1},k_{i_2},\ldots,k_{i_{n-1}}$, then since $P(k_{i_1},k_{i_2},\ldots,k_{i_{n-1}},k_{i_n}/C_p) = P(k_{i_1},\ldots,k_{i_{n-1}}/C_p)\cdot P(k_{i_n}/C_p)$, the numerator of equation (3.9) becomes zero for all classes C_p where $p \neq j$. Hence α_p for all $p \neq j$ becomes zero. Thus all classes C_p, $p \neq j$, are dropped from consideration and C_j is chosen as the correct class since $\alpha_j \geq \alpha$. Thus we see that the occurrence of one keyword has lead to such a drastic decision, no matter how high the α_p values had been before keyword

k_{i_n} occurred. The situation would be even more critical if k_{i_n} had been the first keyword read. Then the document would have been classified after just one keyword. To avoid such precipitous decisions, every zero entry in the CP matrix is replaced by a small value, smaller than all other values in the matrix. Also, it should be noted that if the sample set chosen were larger, then it is highly probable that the number of non-zero entries in the CP matrix would have increased considerably. This small value which replaced every $P(k_i/C_j)$ which equals zero is called the default probability. The implication of such a replacement is that equation (3.9) which calculates the α_i values is now not truly Bayesian because the set of probabilities of keywords associated with a given class no longer sum to unity as pointed out earlier in this section. Now every keyword is "associated" with every class because even those that did not occur in any sample document belonging to that class have a default probability assigned to them. Obviously the smaller the value used for this default probability, the closer is equation (3.9) to being truly Bayesian. When implementing the sequential method on a digital computer there is a limit to how small this value can be made. Scaling techniques have been developed to alleviate underflow problems caused by using very small values of the default probability.

3.4 Storage Technique for the CP Matrix

The CP matrix is stored as a hash table with the keywords being the hash keys. Each entry in the table stores the keyword and the associated probabilities of the classes. This method seems to be ideally suited for the sequential classification implementation, as an extensive amount of searching is required. Tests have shown that processing time is considerably lower for search techniques employing hashing than sequential or binary searches.

The size of the hash tables were designed to provide a 50-80% loading factor which yielded a low average number of probes of the table and resulted in few collisions. Day's algorithm [6] was used to resolve collisions.

It might seem that considerable amount of core storage may be saved by storing the keyword-frequency table in peripheral storage such as a disk. However, if this is done search time will increase sharply and may become prohibitive. For every keyword extracted from a document, the keyword table has to be accessed at least once. Therefore if this table is stored in a disk, the I/O time required may be so high that it offsets the saving obtained in core storage.

3.5 The Stop List

In order to speed up the process of extracting keywords from a document, a stop list of very commonly occurring words, such as "an", "the", "when", etc., is maintained. Every word extracted from a document is first searched in the stop list. If it is not contained in the stop list, then the keyword table is searched. Like the keyword list, the stop list is also stored as a hash table. Since most words present in a document are non-keywords, the probability of finding an extracted word in the stop list is higher than finding it in the keyword list. If there were no stop list, then the keyword table would have to be accessed a number of times before it could be determined that a given word was not a keyword. In general, considerably fewer accesses of the stop list would be required to reach the same decision. This is the main reason why maintaining a stop list is advantageous.

3.6 Implementation of the Sequential Method

The basic sequential method discussed in this chapter was developed and implemented by White and coworkers [28]. Detailed discussion of its implementation and the results obtained by its use on various data bases will be found in the reference cited. Here we shall present a few points of interest about the implementation.

A flowchart of the sequential method is presented in Figure 3.2. As can be seen in the flowchart the implementation varies in one aspect from the theoretical sequential algorithm. Instead of retaining all the classes until a decision is made, whenever the α_j value of a class C_j drops below the threshold α, it is eliminated from consideration. This is done in the interest of reducing compution time. For each class that is retained, an extra multiplication, division and comparison is required. Besides, classes whose α_j values are less than the threshold may cause underflow problems if they are retained. Since these classes have low α_j values it is very likely that keywords occurring in the document in the subsequent stages of the sequential process will have default or very low

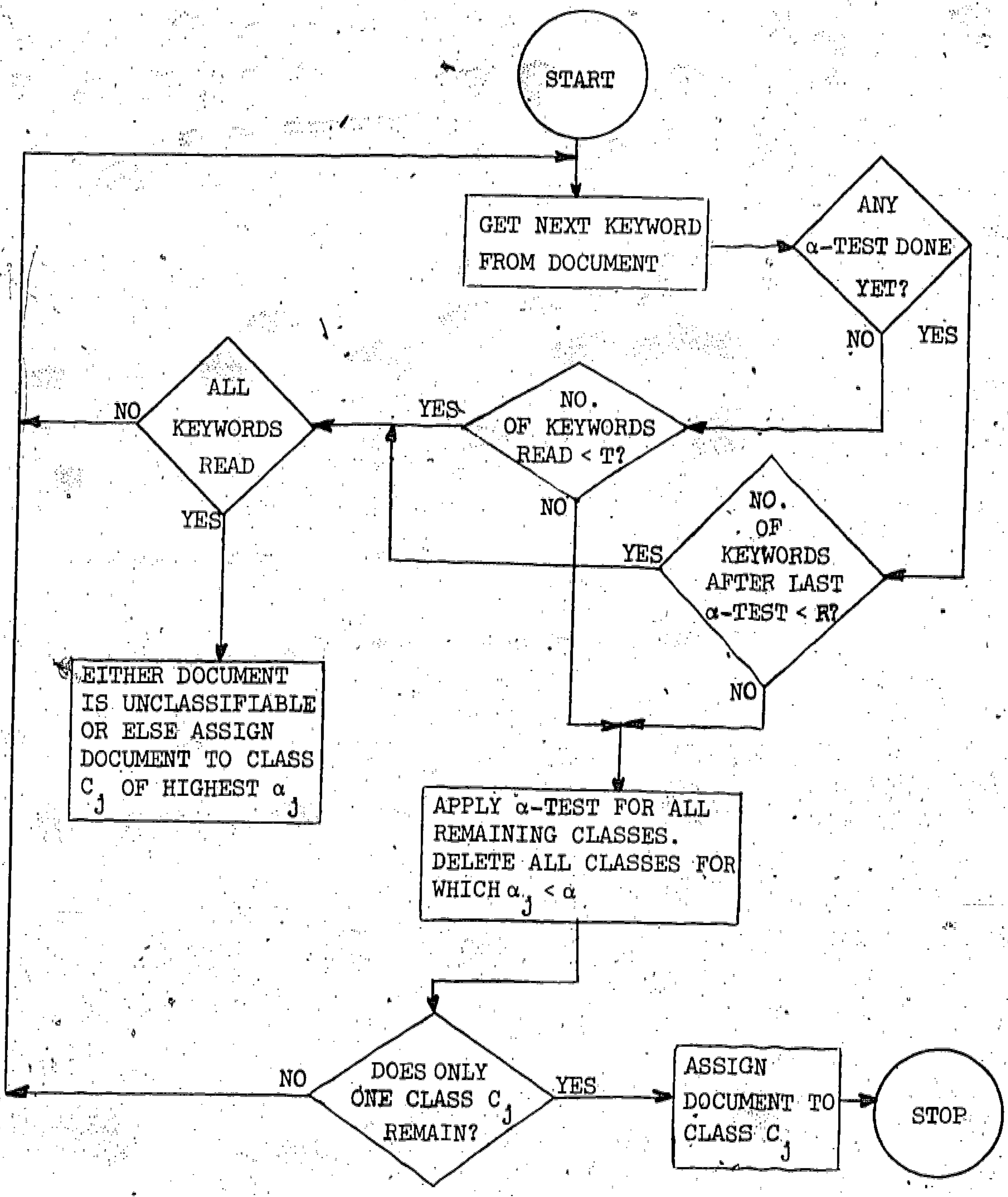

START
GET NEXT KEYWORD FROM DOCUMENT
ANY α-TEST DONE YET?
NO
YES
NO. OF KEYWORDS READ < T?
YES
NO
ALL KEYWORDS READ
NO
YES
NO. OF KEYWORDS AFTER LAST α-TEST < R?
YES
NO
EITHER DOCUMENT IS UNCLASSIFIABLE OR ELSE ASSIGN DOCUMENT TO CLASS C_j OF HIGHEST α_j
APPLY α-TEST FOR ALL REMAINING CLASSES. DELETE ALL CLASSES FOR WHICH $\alpha_j < \alpha$
DOES ONLY ONE CLASS C_j REMAIN?
NO
YES
ASSIGN DOCUMENT TO CLASS C_j
STOP

Figure 3.2 Flowchart of the Sequential Method

probability values associated with them. Hence, during the calculation of their α_j values, the numerator of equation (3.9) will involve multiplication of a large number of small quantities and may cause underflow. For systems with a very large number of classes this might prove to be a substantial overhead. Of course, if all classes were kept until a decision is made, a more accurate decision should result.

It should be noted here that as a result of this decision to drop classes whenever their α_j values fall below the prespecified threshold, a slight change has to be made in the calculation of the a priori probability of the classes $P(C_i)$. This was done at the very beginning of the sequential process by using equation (3.13). Now at each stage of the sequential process, when classes are dropped, the a priori probabilities $P(C_i)$ are recalculated considering only the set of classes that remain at that stage.

The sequential algorithm was applied to the SPIN data base containing a total of 500 documents. Each of these documents were classified by the American Institute of Physics (AIP) into one or more classes of a set of seven categories. Classification results obtained by the sequential algorithm were compared with those of the AIP to get an estimate of how well the method has performed. A sample of the best results obtained is presented in Table 3.1. Several values of the three parameters α, T and R were used.

Table 3.1 Experiments Varying T and R with α = 0.13

T	R	NUMBER OF DOCUMENTS CLASSIFIED CORRECTLY	NUMBER OF DOCUMENTS WHICH SATISFY THE T THRESHOLD	% CORRECT	NUMBER OF DOCUMENTS READ ENTIRELY
2	1	373	466	80.0	55
2	2	375	466	80.5	70
4	1	364	417	87.3	90
4	2	358	417	85.6	99

Variation of α did not produce significant change in classification accuracy. With an increase in T, however, an increase in accuracy was obtained. This is to be expected because as T is increased, more keywords are read before any α-tests are performed. This means that more keywords are examined before any classes are dropped or before an attempt is made to classify a document. Hence precipitous decisions are avoided. Variation in R did not produce significant change in accuracy. However, as T and R are increased, even though classification accuracy increases, fewer documents are actually classified by the algorithm. This is because now fewer documents have enough keywords to satisfy the T and R thresholds. These results will later be compared with the results obtained by applying the revised sequential method of Chapter VI to the same data base.

The next section takes a critical look at the sequential algorithm and points out some of its limitations. These form the basis of the philosophy of the revised sequential method that is discussed in Chapter IV and V.

3.7 Limitations of the Sequential Method

In order to understand some of the limitations of the sequential method, it is necessary to look at the most critical step of the algorithm, viz., the step that calculates the a posteriori probability of the classes, the α_j values. Suppose that at any given stage of the sequential process, a string of keywords $W_1, W_2, \ldots, W_n$

have been read from a document where each W_i is a member of the keyword set K. Then the a posteriori probability for a class C_i, denoted by α_i, is calculated as follows:

$$\alpha_i = \frac{P(C_i)P(W_1/C_i)\ldots P(W_n/C_i)}{\sum_j P(C_j)\ P(W_1/C_j)\ldots P(W_n/C_j)} \tag{3.14}$$

There are several salient points to note about this Bayesian technique. First, the keywords are treated as being completely independent, i.e., the following is assumed:

$$P(W_1,W_2,\ldots,W_n/C_i) = P(W_1/C_i)\cdot P(W_2/C_i)\ldots P(W_n/C_i) \tag{3.15}$$

whereas in fact a more rigorous and accurate technique would be to use the following:

$$P(W_1,W_2,\ldots,W_n/C_i) = P(W_1/C_i)\cdot P(W_2/C_i,W_1)\ldots P(W_1/C_i,W_1,W_2..W_{n-1}) \tag{3.16}$$

A critical look at equation (3.16) shows that in order to consider keyword dependence, frequences of co-occurrences of words have to be computed. Let us consider the calculation of $P(W_2/C_i,W_1)$ for instance. We note that

$$P(W_2/C_i,W_1) = \frac{P(W_2,W_1/C_i)}{P(W_1/C_i)} \tag{3.17}$$

$P(W_1/C_i)$ is already available from previous calculations. Therefore,

$P(W_2,W_1/C_i)$ has to be calculated. For every sample document that belongs to C_i, counts would have to be made for the number of times W_1 and W_2 occur together in the same document. The probability is then obtained by dividing this count by the total count of the frequencies of every keyword pair occurring in class C_i. This has to be done for every possible keyword pair. For three keyword dependencies the computation becomes even more cumbersome. Thus, it becomes clear from these two forms of the same expression and the discussion above that an assumption of keyword independence reduces the computational complexity of the problem immensely.

Besides being cumbersome to compute, equation (3.16) fails to take into account the entire dependency of a keyword on every other keyword that occurs in a document. In the string of keywords $W_1,W_2,\ldots,W_i\ldots,W_n$, equation (3.16) computes the probability based on the assumption that the occurrence of W_i depends on the keywords that have preceded it in the document. Therefore the dependence of W_i on keywords occurring after it is not captured.

The second major point is that because of the assumed independence, every keyword's effect on the α_j value for a particular class is directly proportional to its frequency of occurrence in that class. In other words, let us assume that we have three keywords of moderate probabilities p_1,p_2,p_3 from a class C_x and one keyword of very high probability p_4 from a class C_y. Let the default probability being used be denoted by d. Then at the end of the fourth keyword the ratio of the α_j values for the two classes will be given by:

$$\frac{\alpha_x}{\alpha_y} = \frac{p_1 \cdot p_2 \cdot p_3}{p_4 \cdot d \cdot d} \qquad (3.18)$$

Depending on the values of p_1, p_2, p_3 and p_4, it is possible that this ratio is less than unity, i.e., $p_4 \; d^2 > p_1 \cdot p_2 \cdot p_3$. If this is the case, then the effect of three keywords is nullified by that of a single keyword. If the document actually belongs to C_y, this is a desirable situation; if it belongs to C_x, then the fourth keyword is a noisy word and should be recognized as such.

From these two observations made above about the sequential method, one of its major drawbacks can be described as follows. The sequential method is unable to isolate inappropriate keywords from a set of good keywords. As in the example given above, the fourth keyword with probability p_4 may be a noisy keyword and should be recognized as such and not considered for the probability calculations. It is also obvious that a decision about the appropriateness of this keyword has to be taken in the context of the three keywords which have previously been identified. Thus, even though we do not want to have to calculate keyword dependencies at each stage, we do need a measure of "closeness" between keywords. This measure should be able to isolate noisy keywords from good ones based on the distance between keywords. The decision regarding which keywords should be considered as noisy may change as new keywords are read from the document. A distance measure that does this effectively will be introduced in the next chapter.

Another disadvantage which is not apparent from the above discussion is that the sequential method lacks the power to classify documents into more than one class in a systematic fashion. Very often, based on its subject content, a document should be classified into more than one class. Since at any stage of the classification process, the α_j values denote the a posteriori probability of the classes at that point, the sequential method can use this information to assign a document to a second class. Let us say that we have a three-class problem, i.e., an incoming document may be classified in classes C_1, C_2, or C_3. At the point of classification let the a posteriori probabilities be given by α_1, α_2, α_3, where we will assume without loss of generality that classes C_1, C_2 and C_3 are ordered such that $\alpha_1 > \alpha > \alpha_2 > \alpha_3$, α being the preset threshold level. The primary class is of course C_1, but how about a secondary class?

Several very simple minded techniques may be used to obtain a secondary class at this stage. Since $\alpha_2 > \alpha_3$ we may decide to denote C_2 as a secondary class. However, this technique will obtain a secondary classification in every case irrespective of whether or not it is appropriate to assign such a class. Even though α_2 and α_3 each might be very small, one of them will always be chosen as a secondary class.

One way to handle this problem is to require that α_2 should be comparable to α_1. This can be done by means of a ratio test as follows:

i) calculate the ratio α_2/α_1;

ii) if the ratio exceeds a predetermined threshold α_s then denote C_2 as a secondary class.

A variant of the ratio test where the correlation coefficient between the primary and the possible secondary class is taken into account was actually implemented in the sequential method by White and coworkers [28]. A secondary class would be obtained only if, in addition to satisfying the ratio test, the correlation coefficient between the two classes were higher than a given threshold. The rationale behind this method can be explained as follows. Let us assume that classes C_1 and C_2 are the possible choices for a primary and secondary class respectively. Then a Venn diagram representation of the keywords belonging to these classes can be portrayed by Figure 3.3. Region I represents words which exclusively belong to class C_1, region II represents words which have non-default probability values for both classes C_1 and C_2, and region III represents words which belong exclusively to class C_2. Let n_1, n_2 and n_3 be equal to the number of keywords in regions I, II and III respectively. Then the correlation coefficient between classes C_1 and C_2 is calculated as

$$\rho(C_1,C_2) = \frac{n_2}{n_1+n_2+n_3} . \qquad (3.19)$$

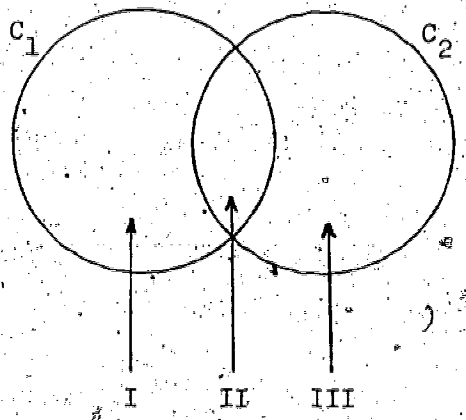


Figure 3.3 Representation of Correlation Between Two Classes

If the value of $\rho(C_1,C_2)$ is high, then it is likely that a given document may belong to both these classes. If, however, n_2 is low and n_1 and n_3 are high, then $\rho(C_1,C_2)$ will have a small value. In such a case, C_2 will never be assigned as a secondary class even though a document may contain keywords occurring from region III.

This suggests that, instead of using a correlation measure, a method should be devised which will be able to separate groups of keywords occurring from regions I and III and analyse them separately to obtain primary and secondary classes. The next chapter addresses the problem of isolating noisy keywords and identifying clusters of similar keywords. It defines a distance measure, called the Bayesian distance, and shows how its properties can be utilized to design a technique for handling these problems.

CHAPTER IV

THE BAYESIAN DISTANCE

The sequential method that has been described in Chapter III is a fast and fairly accurate method for document classification. However, in section 3.7 where some of its limitations were noted, it was pointed out how the sequential method may be vulnerable to the occurrence of noisy keywords and how it fails to achieve secondary classification in a systematic fashion. It was shown that if an appropriate distance measure could be defined, then noisy keywords could be isolated and clusters of similar keywords could be identified. These keywords could then be analyzed to obtain primary and secondary classes for a document. This chapter defines such a distance measure and outlines some of its properties. Before this is done, however, some required concepts are defined.

Good Keyword: It was noted in Chapter III that a keyword has an associated set of probabilities represented as k_i: $[p_1, p_2, \ldots, p_t]$. If k_i occurs in a document then the quantity p_j, which denotes $p(k_i/C_j)$, measures the strength with which this keyword relates the document to class C_j. Suppose a document belongs to class C_j. Then a keyword k_i contained in this document is a good keyword if p_j is greater than or equal to any of the other probability values associated with k_i.

Noisy Keyword: Referring to the discussion above, keyword k_i is a noisy keyword if there exists probability values associated with k_i which are greater than p_j.

Primary and Secondary Classes: Very often, based on its subject content, a document may be classified into more than one class, say C_i and C_j. If this is the case then this document should contain keywords which are indicative of both class C_i and C_j. This may happen in three ways.

(i) The document may contain a group of keywords $\{k_1, k_2, \ldots, k_n\}$ such that their probability components for C_i and C_j are predominantly higher than for the other classes.

(ii) The document may contain groups of keywords $\{k_{i_1}, k_{i_2}, \ldots, k_{i_n}\}$ and $\{k_{j_1}, k_{j_2}, \ldots, k_{j_n}\}$ such that the first group is a set of good keywords for class C_i and the second group is a set of good keywords for class C_j.

(iii) The document may contain keywords which fit into both categories (i) and (ii). Based on an analysis of these keywords it may be determined that the document should be classified into both C_i and C_j. If the keywords are more indicative of C_i than of C_j, then C_i is denoted as the primary class and C_j as the secondary class.

This chapter deals with the definition of a measure which is able to isolate groups of words which are similar in nature in the sense that words in the same group are all indicative of a particular class. More specifically, a distance function is defined which is able to isolate noisy keywords from the good keywords and discard the noisy ones from consideration while obtaining a primary class. Noisy keywords so isolated may then be analyzed to determine whether they indicate the fact that the document may be assigned a secondary class.

This distance measure, called the Bayesian distance, has been defined on a t-dimensional vector space which may be used to represent keywords. The next section discusses such a representation.

4.1 Vector Representation of Keywords

The keyword set representation for the sequential method consists of a matrix of conditional probabilities. Each of these probability values represents a numerical measure of the extent to which a keyword describes a particular class. This matrix, referred to as the conditional probability matrix CP, was introduced in section 3.2 of the previous chapter.

If a row corresponding to keyword k_i is isolated from this matrix the following is obtained:

$$k_i: \ [p_1, p_2, \ldots, p_t]$$

where

$$p_j = p(k_i/C_j).$$

As the sequential method reads documents and extracts keywords, it uses only these probabilities associated with a keyword to calculate the α_j values for the set of classes. Therefore, for the purposes of classification, a keyword can be represented by a t-dimensional vector. Two words, even though they are distinct, will have the same effect as far as the probability calculations are concerned if the probability vectors associated with them are the same. By the same token, two words with unequal probability vectors will have different effects in the a posteriori probability calculations.

A keyword therefore can be represented as a point in a t-dimensional vector space where each component of such a vector is a number between zero and unity. Hence, a document which contains several such keywords can be represented as a collection of points in this space. The next section elaborates on this idea.

4.2 Vector Space Representation of Document

A document, for the purposes of classification, is a group of keywords $\{k_1, k_2, \ldots, k_n\}$. As noted earlier, each of these keywords can be represented as a point in a t-dimensional space and the relationship between each keyword can be studied. For the sake of simplicity, let us consider a 3-class case and a document that has a total of seven keywords. These seven points in a 3-dimensional space may be visualized as shown in Figure 4.1. Suppose that in this three-class situation, we were to represent every keyword in the system, i.e., the complete set of keywords $K = \{k_1, k_2, \ldots, k_m\}$, in this 3-dimensional space. Each keyword k_i will have an associated probability vector of the form:

$$k_i: [p_1, p_2, p_3].$$

Then three distinct groups of keywords, G_1, G_2, G_3 could be identified.

(i) G_1 represents all keywords for which p_1 is greater than the default probability value.

(ii) G_2 represents all keywords for which p_2 is greater than the default probability value.

(iii) G_3 represents all keywords for which p_3 is greater than the default probability value.

A keyword depending on its probability components may belong to any one, any two or all three of these groups. In Figure 4.1, G_1, G_2, and G_3 have been drawn to represent these groups.

Looking at the figure, we can say intuitively that the document belongs to class C_1 with keywords k_1, k_2, k_3, k_4 and k_5 totally or partially representative of that class. In other words

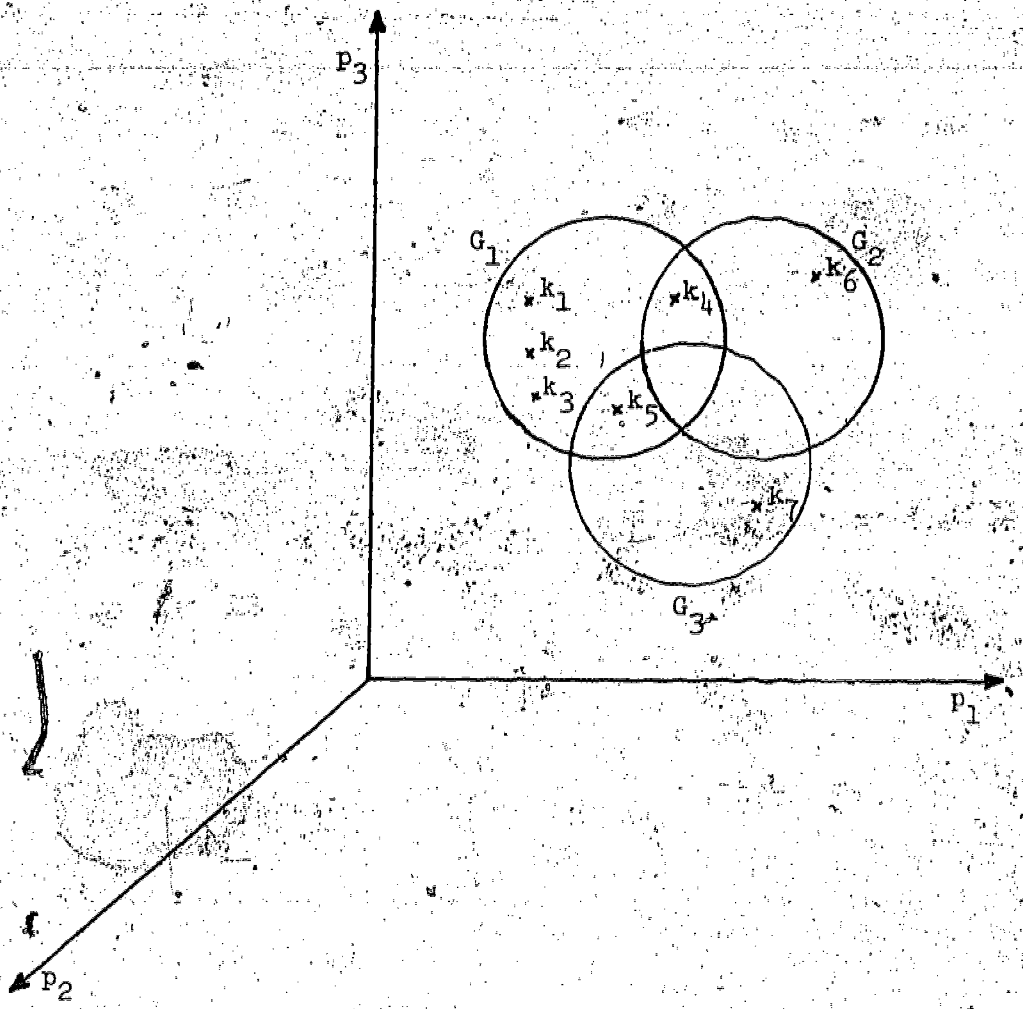


Figure 4.1 Vector Space Representation of a Document

these keywords are indicative of class C_1 because they have a non-default probability component for that class. They can be said to form a cluster of similar words. Assuming that the document does belong to class C_1, k_6 and k_7 are definitely 'noisy' words because they have default probability values for class C_1, and are more indicative of classes C_2 and C_3 respectively. In other words k_1, k_2,

k_3, k_4, and k_5 are more similar to each other than they are to k_6 and k_7. If this fact could be recognized during classification, then k_6 and k_7 could be discarded from consideration while obtaining a primary class.

As another example, instead of having one group of similar keywords, a representation of the given document may be of the form shown in Figure 4.2. Here we notice that two distinct clusters of keywords occur. The first group, denoted by I, has keywords which are exclusively indicative of class C_1. These may be used to obtain a primary class. Group II contains words which have non-default probability components for class C_2, and may be used to obtain a secondary class. Again, intuitively we can see that words in group I are similar to each other and dissimilar from words in group II. A measure capable of isolating such groups of keywords could achieve primary and secondary classification.

4.3 Conventional Distance Measures

The discussion in the previous section has pointed out the need for a similarity or distance measure which will be able to isolate noisy keywords and identify groups of similar keywords. In this section we discuss some of the conventional distance measures that have been used to compute similarity between keywords. It is also shown why these measures are inadequate to handle the two requirements mentioned above, thereby establishing the need for a new distance measure.

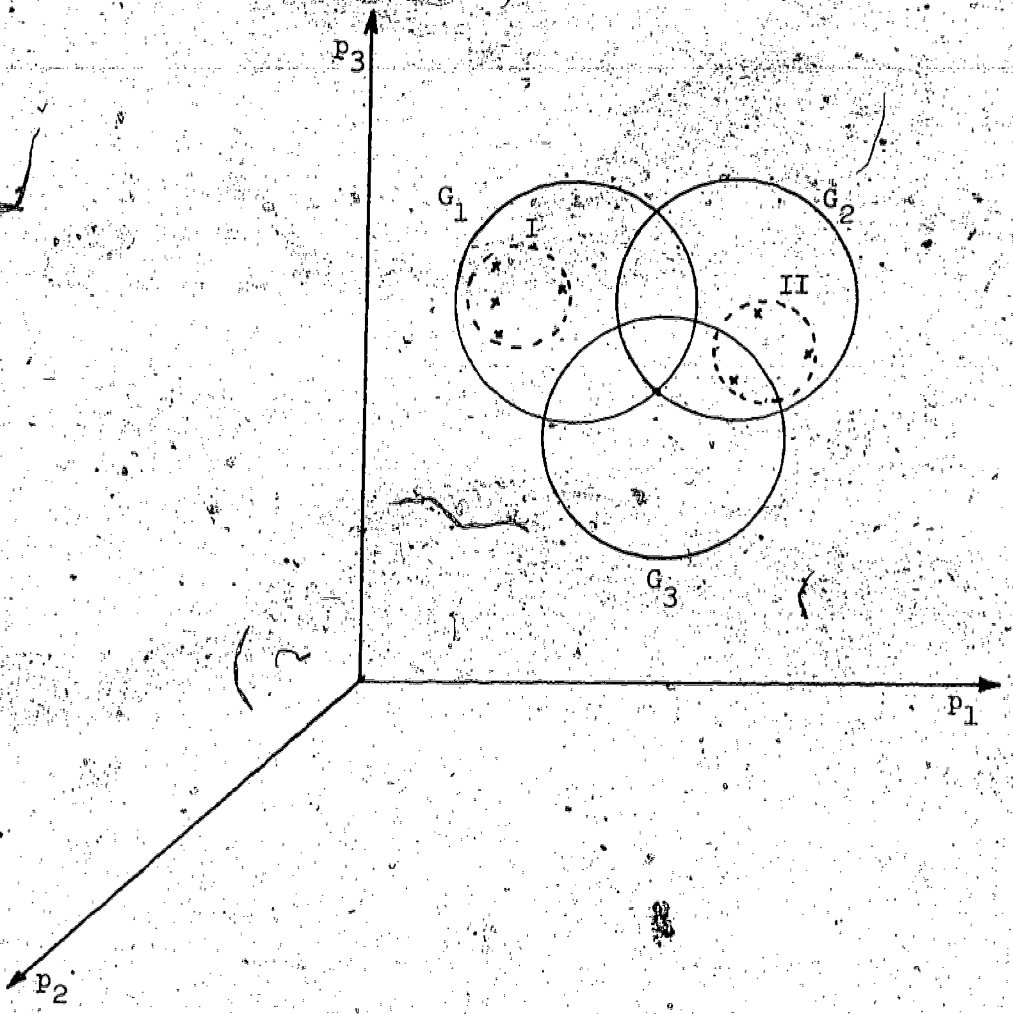


Figure 4.2 Clusters of Keywords

One of the earliest measures used is the vector correlation measure [22], defined for binary vectors. Suppose v and w are t-dimensional binary vectors, then the correlation r_{vw} between the two vectors is given by

$$r_{vw} = \sum_{i=1}^{t} v_i w_i \qquad (4.1)$$

This measure ranges from 0 to t and its size depends on the number of matching non-zero properties in both vectors.

In order to use this measure for our problem, the keyword vectors would have to be converted to binary vectors by defining a threshold value. This might prove to be very wasteful of information. Another disadvantage is that this measure indicates only the similarity and not the dissimilarity between two vectors. For instance, the two vector pairs

$$v = (1, 1, 1, 0, 0, 0, 1, 1)$$

$$w = (1, 1, 1, 0, 0, 0, 0, 0)$$

and

$$v = (1, 1, 1, 0, 0, 0, 0, 0)$$

$$w = (1, 1, 1, 0, 0, 0, 0, 0)$$

have the same similarity value even though they are considerably different.

A more comprehensive measure is the cosine correlation measure [22]. This has been used in many systems and specifically in the SMART retrieval system. This measure has a geometrical interpretation, viz., if v and w are two vectors, then the cosine measure computes the cosine of the angle between these two vectors, i.e.,

$$r_{vw} = \frac{v \cdot w}{||v|| \cdot ||w||} \qquad (4.2)$$

where the numerator is the scalar product of the two vectors and the denominator is the product of their magnitudes. Our problem requires that if two keyword vectors lie within the same class boundary they should be "closer" to each other than two vectors which lie in different classes, even though the angle subtended between them is the same. The cosine measure fails to achieve this.

Another measure that has been generally used to find the distance between points in a vector space is the Euclidean distance measure. Suppose we have two keyword vectors

$$k_1: \ [p_1, p_2, p_3]$$

$$k_2: \ [q_1, q_2, q_3]$$

then the Euclidean distance $D(k_1, k_2)$ between these vectors is given by

$$D(k_1, k_2) = [\sum_{i=1}^{3} (p_i - q_i)^2]^{\frac{1}{2}} \qquad (4.3)$$

Why such a measure is inadequate for our purposes is best illustrated by an example. Suppose we have two keywords k_1 and k_2 with associated vectors

$$k_1: \ [0.7, 0.01, 0.01]$$

$$k_2: \ [0.2, 0.01, 0.01]$$

Then $D(k_1, k_2) = 0.5$. If instead, we had two keywords k_3 and k_4 with vectors

$$k_3: \ [0.01, 0.41, 0.01]$$

$$k_4: \ [0.01, 0.01, 0.31]$$

then again $D(k_3, k_4) = 0.5$.

Suppose now that the a posteriori probabilities, the α_j values, for classes C_1, C_2, C_3 are calculated using keywords k_1, k_2, k_3, and k_4 individually, using equation (3.9) of the previous chapter. We assume that the a priori probabilities of the classes are the same. These α_j values are shown in Table 4.1. We see from the values in Table 4.1 that k_1 and k_2 are both more highly indicative of class C_1 than the other classes. On the other hand k_3 points toward class C_2, and k_4 is clearly indicative of class C_3. Therefore, our requirements of a distance measure demand that k_1 and k_2 be identified as being 'closer' to each other than k_3 and k_4. The Euclidean distance measure fails to do this. The following section defines the Bayesian distance which not only considers the magnitudes of each probability component of a keyword but also information about how a keyword is related to a given class.

4.4 Definition of the Bayesian Distance

Suppose a document is to be classified into one class of a set of classes C_1, C_2, ..., C_t. At a given stage of the sequential process, suppose we have read i keywords k_1, k_2, ..., k_i. Then the a posteriori probabilities of each of these classes is denoted by $P(C_j/k_1, k_2, \ldots, k_i)$, where j varies from 1 to t. For simplicity this will be written as $P(C_j/y)$, where y denotes the occurrence of keywords k_1 to k_i. Then the a posteriori probabilities of all the classes after observation y can be represented by the following vector:

$$P(C/y): \; [p(C_1/y), \ldots, P(C_t/y)].$$

Table 4.1 A Posteriori Probability of Classes C_1, C_2, C_3

KEYWORD	α_1	α_2	α_3
k_1	0.97	0.015	0.015
k_2	0.91	0.045	0.045
k_3	0.025	0.95	0.025
k_4	0.03	0.03	0.94

It can be noted here that $p(C_j/y)$ is simply the α_j computed by equation (3.9) of the previous chapter after keywords k_1 to k_i have been read.

Definition: The Bayesian distance on the probability space of a set of classes after an observation y, i.e., after i keywords have been processed is $D_B(C/y) = [\text{Mag}(i), \text{Dir}(i)]$. Mag(i) and Dir(i) represent the magnitude and direction of D_B respectively, and are given by

$$\text{Mag}(i) = \sum_{j=1}^{t} [P(C_j/y)]^2$$

$$\text{Dir}(i) = r \ni P(C_r/y) \geq P(C_j/y), \quad j = 1, 2, \ldots, t.$$

Dir(i) is the index of the class having the highest a posteriori probability after i keywords have been read.

At this point, before noting the usefulness of the Bayesian distance measure for the purpose of classification, it is worthwhile to explore its relationship with some concepts of interest in classification theory.

4.5 Classification Error

Since we are using the Bayes conditional probability relation to obtain the a posteriori probabilities of the classes at each stage, it is easy to obtain an expression for the classification error. At the time of classification let the a posteriori probabilities be: $\{P(C_1/y), P(C_2/y), \ldots, P(C_t/y)\}$. If C_j is the class chosen, then the error of classification P_E is given by

$$P_E = 1 - P(C_j/y).$$

The error P_E will obviously be a minimum if the class C_j is such that its a posteriori probability is higher than all the others.

Theorem 4.1

If P_E denotes the classification error after observation y, i.e. after i keywords have been read, then

$$P_E \le 1 - Mag(i)$$

Proof: Let us define an index set

$$I = \{1, 2, \ldots, t\}$$

$$Mag(i) = \sum_{j\in I} P(C_j/y)^2 \le [\max_{j\in I}\{P(C_j/y)\}][\sum_{j\in I}[P(C_j/y)].$$

Since $\sum_{j\epsilon I} P(C_j/y) = 1$, we have

$$Mag(i) \leq [\max_{j\epsilon I}\{P(C_j/y)\}] \leq 1 - P_E . \qquad Q.E.D.$$

This shows that the magnitude of the Bayesian distance forms an upper bound on the classification error at any stage of the sequential process.

4.6 Information Theoretic Measure of the Value of a Keyword

Besides the Bayesian distance measure which is being proposed in this research, there exist many other measures in the theory of pattern recognition which are related to the classification error. Some of these may be adapted to the problem of document classification. One such measure is the information theoretic measure of the value of a feature measurement. This can be adapted to the problem of document classification as follows.

Before any keywords are read, the a priori probabilities of the classes are:

$$\{P(C_1), P(C_2), \ldots, P(C_t)\}.$$

The entropy of this probability distribution is given by

$$H = \sum_{i=1}^{t} P(C_i) \log \frac{1}{P(C_i)} .$$

After a keyword has been read and the a posteriori probabilities are calculated using Bayes relationship, the new entropy H' may be calculated as above. Therefore, the reduction in uncertainty in the choice of a right class is a measure of the amount of information, I, given by this keyword. It is given by equation (4.4).

$$I = H - H' . \tag{4.4}$$

Therefore, for a given H the lower the value of H', the greater is the information given by the keyword. It can be shown that for a 2-class problem $P_E \leq \frac{1}{2}H'$. Thus, like the Bayesian distance, the entropy measure also forms an upper bound on the error of classification. However, it can be proven that the magnitude of the Bayesian distance forms a tighter upper bound than the entropy measure.

Consider two classes C_1 and C_2. Let P_1 represent the probability of class C_1 at any stage; then $1 - P_1$ is the probability of class C_2. We can then compute the values of classification error, P_E, the Bayesian distance bound, $1 - \text{Mag}(i)$, and the entropy bound, i.e., $\frac{1}{2}H'$, for various values of P_1. Table 4.2 presents these quantities at five different values of P_1 and Figure 4.3 represents them graphically. These upper bounds have been demonstrated for a two class problem. It can be proven that as the number of classes increases, the quality of the Bayesian distance as an approximating function to the classification error does not degrade very much. This is stated as a theorem and proven in Appendix A.

Table 4.2 Upper Bounds on Classification Error

P_1	P_E	1 - Mag(i)	$\frac{1}{2}H'$
0	0	0	0
0.25	0.250	0.307	0.406
0.5	0.500	0.500	0.500
0.75	0.250	0.307	0.406
1	0	0	0

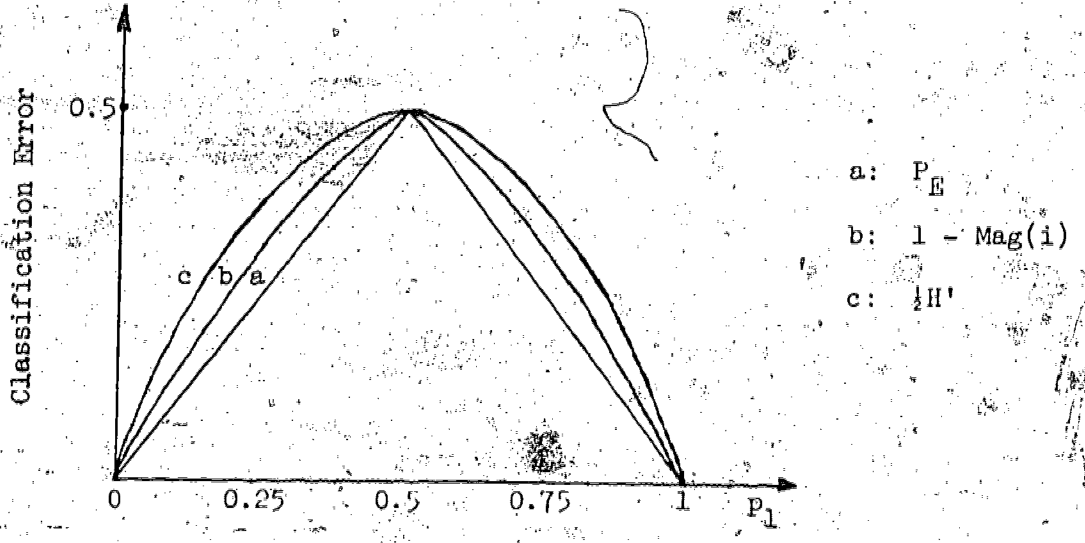


Figure 4.3 Upper Bounds on Classification Error

4.7 Use of Bayesian Distance in Classification

In the previous sections a definition of Bayesian distance has been given and some of its properties with respect to classification theory have been studied. In this section it will be shown why the Bayesian distance is an appropriate tool to handle the problems that have been outlined.

Two of the requirements of a distance measure used for the purposes of classification have been identified as:

(i) the distance measure should be able to isolate noisy keywords, and

(ii) it should be able to identify clusters of similar words or words that are predominantly indicative of one class.

Suppose a keyword k_1 has been read from a document. Then the a posteriori probabilities of each of the classes, $P(C_j)$, is given by the α_j values that are calculated at each stage of the sequential process. Let the set of α_j values be denoted by

$$A = \{\alpha_1, \alpha_2, \ldots, \alpha_t\}.$$

Let I represent the index set {1, 2, ..., t}. If α_1 is the highest α_j value at this point, then the magnitude and direction of the Bayesian distance D_B is given by

$$\text{Mag}(1) = \sum_{j \varepsilon I} (\alpha_j)^2$$

$$\text{Dir}(1) = 1 .$$

We note that Mag(1) is the sum of the squares of a set of numbers which sum to unity. Let $x = \{x_1, x_2, \ldots, x_t\}$ denote such a set. The sum of the squares is then given by

$$S_x = \sum_{j \epsilon I} (x_j)^2 .$$

Let x_1 be such that $x_1 \geq \sum_{j=2}^{t} x_j$, where x_1 is now increased by a quantity δ and the other x_j are decreased in any desired way such that we have a new set of numbers

$$Y = \{y_1, y_2, \ldots, y_t\}$$

where

$$y_1 = x_1 + \delta \quad \text{and} \quad \sum_{j \epsilon I} y_i = 1.$$

Then the new sum of squares is given by

$$S_y = \sum_{j \epsilon I} (y_j)^2 .$$

Then S_y is greater than S_x. A proof of this fact will be given in Appendix B.

Therefore if α_1 is such that

$$\alpha_1 \geq \sum_{j=2}^{t} \alpha_j ,$$

then an increase in α_1 will increase the value of Mag(i) without changing Dir(i). It should be noted here that this is a sufficient condition and not a necessary one.

Since α_1 is the highest α_j value, it implies that probability component p_1 of the vector associated with keyword k_1 is higher than all the other components of the vector. Therefore if the document does indeed belong to class C_1, then k_1 is a good keyword.

Now suppose a keyword k_2 is read which has an associated probability vector as follows

$$k_2: \ [q_1, q_2, \ldots, q_t] .$$

If keyword k_2 is also more highly indicative of class C_1 than the other classes, then the q_1 component will be higher than the other components. The new set of α_j values is given by

$$A' = \{\alpha_1', \alpha_2', \ldots, \alpha_t'\}.$$

The new Bayesian distance is

$$\mathrm{Mag}(2) = \sum_{j \varepsilon I} (\alpha_j')^2$$

$$\mathrm{Dir}(2) = 1.$$

The value of Mag(2) will be greater than Mag(1) if α_1' exceeds α_1. But an increased α_1' means that with the occurrence of keyword k_2 the confidence in class C_1 being the correct class has increased. Therefore k_2 can be considered to be a good keyword. If instead Mag(2) decreases or the direction changes, k_2 can be identified as a noisy word in relation to k_1 and hence isolated. Thus at each stage of the sequential process, depending on the nature of variation of the Bayesian distance magnitude and direction, a keyword can be labeled as either good or noisy. The good keywords can then be analyzed to obtain a primary class, and the noisy keywords, which are noisy with respect to this primary class, can be analyzed to yield a possible secondary class.

The next chapter provides an experimental verification of the claims made in this section. Also the fact that the Bayesian distance magnitude increases with the occurrence of a good keyword will be proved rigorously for a three class problem. Finally it will be shown how the good and noisy keywords can be effectively separated and analyzed to yield primary and secondary classification.

CHAPTER V

APPLICATION OF THE BAYESIAN DISTANCE

In the last chapter the concept of a distance measure on the keyword space was introduced. A specific distance measure, called the Bayesian distance, was defined and some of its properties with respect to classification theory were investigated. In this chapter the use of the Bayesian distance for the purpose of document classification will be studied. It will be shown that by noting the variation of the magnitudes and directions of the Bayesian distance with keywords read, various patterns can be detected which correspond to noisy words and good words. Also, a distinction can be made between ideal and non-ideal documents. Results of the experimental verification of several hypotheses will be presented and in some cases a mathematical justification will be given.

5.1 Variation of Bayesian Distance with Keywords

In order to test the validity of the concept of Bayesian distance and how it will perform in the case of an actual data base, a set of experiments was conducted on the SPIN data base. The purpose of the experiments was to note the variation of the magnitude and direction of the Bayesian distance with each keyword read from a given document.

Although the characteristics and details describing the SPIN data base are described in [28], the basic root classes of SPIN are reproduced here for ease in understanding the results obtained. Each document in SPIN may be classified into one or more of the seven classes listed in Table 5.1. A numerical code, which will be referenced later in this chapter, is indicated in parentheses.

Table 5.1 Spin Root Classes

CLASSIFICATION CODE	DESCRIPTION
B (0)	General
D (1)	High energy and nuclear physics
M (2)	Atoms, molecules, and chemical physics
P (3)	Fluids and plasmas
S (4)	Solid state physics
T (5)	Acoustics, optics, and cross-disciplinary physics
X (6)	Astronomy and astrophysics

Each document was read entirely and a list of all the keywords present was formed. Let this list be rep[illegible] by

$$L = \{W_1, W_2, \ldots, W_n$$

where W_i is the i^{th} keyword occurring in the document. Each keyword W_i is a member of the keyword set K and has an associated probability vector of the form

$$W_i : [p_1^i, p_2^i, \ldots, p_t^i].$$

Let α_j^i represent the a posterior probabilities of the classes after the i^{th} keyword. In Chapter III, methods for their calculation were discussed. The Bayesian distance (D_B) magnitude and direction are calculated using these α_j^i values. Let Mag(i) denote the magnitude and Dir(i) denote the direction of D_B after the i^{th} keyword W_i is processed. Then

$$\text{Mag}(i) = \sum_{j=1}^{t} (\alpha_j^i)^2 \qquad (5.1)$$

and Dir(i) is the index of the highest α_j^i value. Mag(i) and Dir(i) are calculated for each value of i ranging from 1 to n. Before the results of these experiments are reported several terms need to be defined. The definitions of good keyword and noisy keyword, given in Chapter IV, are repeated here for continuity.

Good Keyword: During the process of classification a keyword is considered to be a good keyword if its highest probability component belongs to the class to which the document belongs. In this case the keyword relates the document to the indicated class more highly than to any other class.

Noisy Keyword: During the process of classification a keyword is considered to be a noisy keyword if its highest probability component points to a class other than a class to which the document belongs.

Ideal Document: A document is considered to be ideal if every keyword contained in it is good.

Non-Ideal Document: A document is considered to be non-ideal if there is at least one keyword in it which is noisy.

It should be noted here that these definitions presuppose that the class to which a document belongs is known. During classification, however, the class membership of a document is not known until the end of the sequential process. A keyword that is extracted from a document at each step of this process is tagged as either good or noisy. This decision depends only on the keywords that have preceded this particular keyword, and may change as more of the document is examined. This point is clarified later in the chapter.

Based on these concepts and on the results of the Bayesian distance values, the complete set of documents was divided into a set of ideal documents and a set of non-ideal ones. Sample documents from each set were obtained and examined separately to study the nature of the variation of Bayesian distance with keywords.

5.2 Analysis of Ideal Documents

Twenty documents were selected from the set of ideal documents for detailed study. In all cases the D_B magnitude increases with the number of keywords read until it reaches or closely approaches a value of unity, the direction being constant over the entire range. This phenomenon is illustrated in Figure 5.1. This fact can be utilized to recognize documents which are ideal in nature and classify them into a class designated by the direction of D_B. The fact that such a situation can also occur for non-ideal documents will be discussed later.

The nature of the variation of the magnitude of D_B has been utilized to design a classifier which searches for such a pattern in the Bayesian distances obtained by processing the keywords in a test document. Figure 5.1 illustrates how the D_B curve gradually approaches the value of unity over a range of keywords. The magnitude of D_B at any stage of the sequential process depends on the set of α_j values $\{\alpha_1, \alpha_2, \ldots, \alpha_t\}$ computed at that stage. The only way that the magnitude can assume a value of unity is when one of these α_j is unity and the rest are all zero. Suppose n keywords $\{W_1, W_2, \ldots, W_n\}$ have been read. Without loss of generality, let us assume that each keyword W_i has an associated probability vector of the form $[p_1^i, d, \ldots, d]$, where d is the default probability value. Hence, each of these keywords is strongly indicative of class C_1. If the α_j values are computed using these n keywords then the value of α_1 is given by

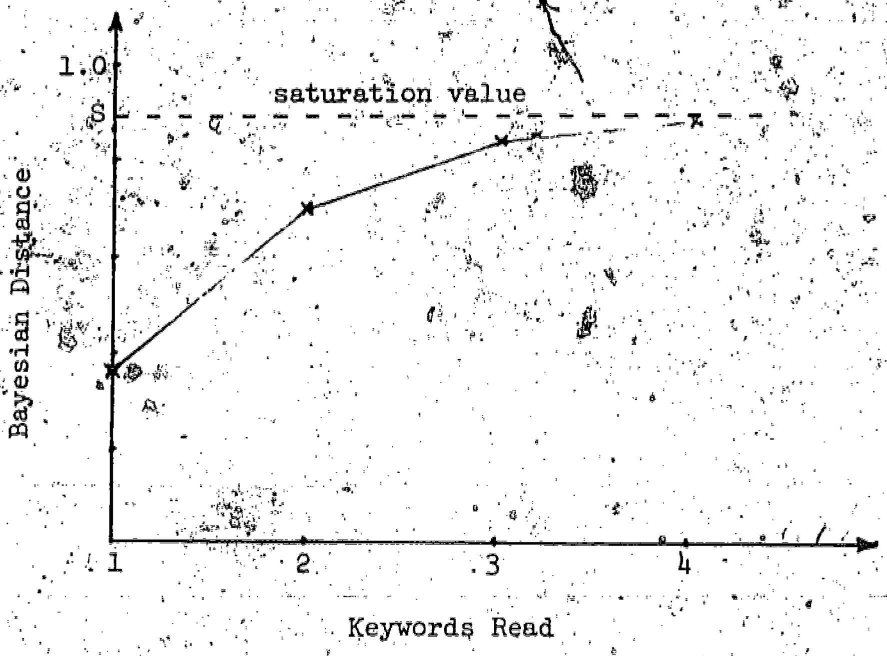


Figure 5.1 Bayesian Distance for Ideal Documents

$$\alpha_1 = \frac{\prod_{i=1}^{n} (p_1^i)}{\prod_{i=1}^{n} (p_1^i) + (t-1)d^n} \qquad (5.2)$$

Each of the other α_j values is given by

$$\alpha_j = \frac{d^n}{\prod_{i=1}^{n} (p_1^i) + (t-1)d^n}, \quad j > 1 . \qquad (5.3)$$

Because $p_1^i > d$, as more and more keywords of the same form are processed, α_1 approaches unity and the other α_j approach zero. Theoretically, however, α_1 never reaches a value of unity. The magnitude of D_B which is given by

$$\text{Mag}(i) = \sum_{j=1}^{t} (\alpha_j)^2,$$

approaches a value of unity as α_1 approaches unity. Therefore Mag(i) can be made as close to unity as desired by processing more such good keywords. In practice this is not feasible because documents have a limited number of keywords. Besides, if a large number of keywords is read ih order to make Bayesian distance magnitude very close to unity, the entire purpose of a sequential technique has been lost.

A practical implementation therefore requires the definition of two parameters. One is the number of keywords that need to be examined before a decision can be made. This parameter will be denoted by Q. The other is a parameter called the saturation value and will be denoted by S. In order that a document may be classified, the magnitude of D_B after Q good keywords are processed should be greater than or equal to S. It was pointed out in Chapter IV that the D_B magnitude forms an upper bound on the error of classification, and so S represents a confidence level. The higher the value of S, the greater is the confidence that the document belongs to the class given by the index of D_B.

These two parameters can be determined experimentally for a given data base by taking samples of ideal documents and analyzing their Bayesian distance patterns. Such experiments have been conducted for the SPIN data base and the results are presented in Table 5.2 for the 20 ideal documents selected for detailed study. Before the results are analyzed, however, another important statistic should be mentioned, i.e., the initial value of the Bayesian distance. Let us denote this by B_I. This initial value B_I reflects the quality of the first keyword and for an ideal document all of whose keywords have predominantly high probability values for a single class, the higher the B_I value the faster will the Bayesian distance reach the saturation threshold. In a practical implementation, the parameter Q can be changed based on the value of B_I. If B_I is low, Q can be increased because more keywords will be required to reach the saturation threshold. This is clarified in the discussion which follows.

The documents in Table 5.2 have been listed in ascending order

Table 5.2 Analysis of Ideal Document

SATURATION VALUE = 0.99

DOCUMENT NUMBER	INITIAL VALUE OF $\|D_B$ (B_I)	NO. OF KEYWORDS REQUIRED FOR SATURATION	SPIN CATEGORY
1	0.353	5	S
2	0.389	3	M
3	0.522	3	M
4	0.522	4	M
5	0.562	3	M
6	0.636	3	S
7	0.648	3	M
8	0.649	[illegible]	M
9	0.650	[illegible]	S
10	0.737	[illegible]	P
11	0.757	3	D
12	0.798	2	T
13	0.923	3	S
14	0.923	2	S
15	0.953	2	S
16	0.967	2	M
17	0.970	3	S
18	0.983	2	T
19	0.985	2	P
20	0.995	1	P

according to the value of B_I. The third column in this table gives the number of keywords that are read before the magnitude of the D_B exceeds or equals the value of the parameter S. For the results given in Table 5.2 a saturation value of 0.99 was assumed. As can be seen from the table, a higher value of B_I requires, in general, that a greater number of keywords be read before saturation is obtained. In other words, the parameter Q should vary according to the value of B_I. In order to study the variation of this parameter, the range of B_I for the ideal documents was divided into four intervals. Experiments were conducted to obtain an average number of keywords required to reach saturation in each of the intervals. The results presented in Table 5.3 show that when [illegible] lower, a higher average number of keywords are read before the saturation value is reached. During classification therefore, B_I can be used to determine the value of the parameter Q.

5.3 Analysis of Non-Ideal Documents

As in the case of ideal documents, 20 non-ideal documents were selected at random and analyzed. These documents are such that each one of them contains at least one noisy keyword. For clarity of presentation, only a sample set of the results obtained is shown in Table 5.4. In the columns of Table 5.4, the first value represents the magnitude of D_B and the second value denotes its direction. The direction corresponds to a numerical index assigned to each of the SPIN categories, which was identified in Table [illegible]. The noisy keywords

Table 5.3 Experiments on Starting Intervals

SATURATION VALUE = 0.99

Intervals for B_I:

I_1: (0.3,0.5)

I_2: (0.5,0.7)

I_3: (0.7,0.9)

I_4: (0.9,1.0)

STARTING INTERVAL	NO. OF DOCUMENTS IN INTERVAL	AVERAGE NO. OF KEYWORDS REQUIRED FOR SATURATION
I_1	2	4.0
I_2	7	3.0
I_3	3	2.3
I_4	8	2.2

Table 5.4 Analysis of Noisy Documents

DOCUMENT	KEYWORDS							SPIN
NUMBER	1	2	3	4	5	6	7	CLASS
1	.880,3	.803,3	.999,3	.999,3				P (3)
2	.502,1	.500,1	.532,0	.976,1	.999,1			D (1)
3	.961,3	.675,3	.888,1	.903,3	.892,3			P (3)
4	.945,1	.481,1	.947,1	.999,1	.853,1	.998,1		D (1)
5	.389,2	.809,4	.430,3	.501,2	.438,4	.986,2	.999,2	M (2)

ote: Noisy keywords are underlined.

are underlined for ease in identification. As can be seen from the data, the occurrence of a noisy keyword is marked by a change in magnitude or direction or both in the Bayesian distance values. The general nature of this variation can be studied by identifying two cases depending upon whether only the magnitude changes or both magnitude and direction change.

This first type is shown in Figure 5.2. In this case the occurrence of a noisy keyword is marked by a decrease in magnitude, the direction remaining the same. After the noisy keyword, the magnitude again increases starting from the following keyword and reaches saturation as in the case of an ideal document. This situation is depicted by documents 1 and 4 in Table 5.4. The second keyword in document i is a noisy keyword and hence the magnitude drops from 0.880 to 0.803. When the third and the fourth keywords are processed, the magnitude increases from 0.803 to 0.999, and can both be identified as good keywords. For document 4, the same phenomenon is observed except that the second and fifth keywords are both noisy.

Another effect caused by the occurrence of a noisy keyword is the change in the direction of the Bayesian distance. Here the noisy keywords are so strongly biased towards a wrong class that a change in direction occurs irrespective of the nature of the change in magnitude. This situation is depicted by documents 2, 3 and 5 in Table 5.4. In document 3 the third keyword is very strongly

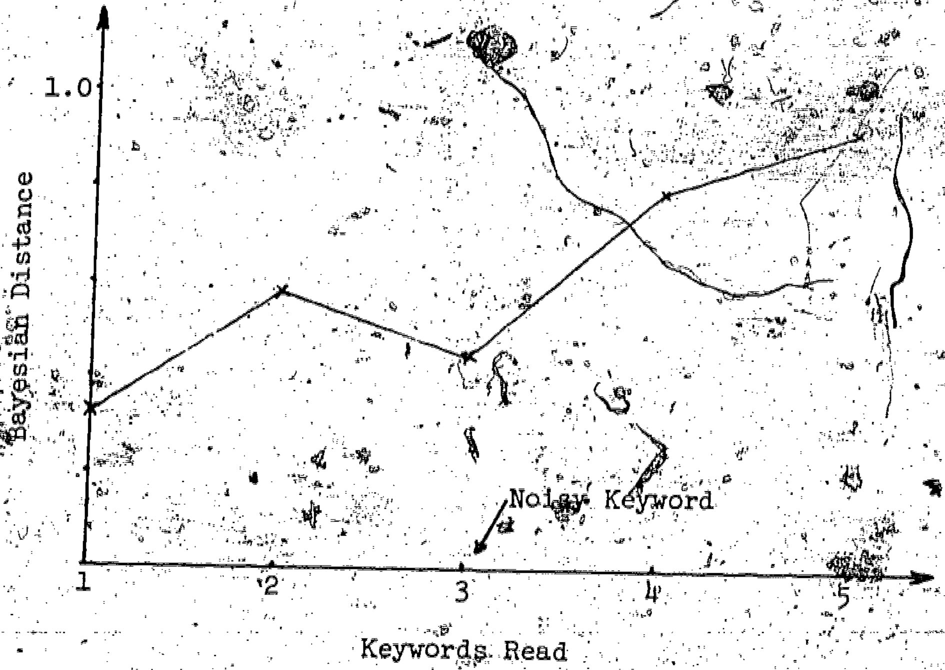


Figure 5.2 Change in Magnitude of Bayesian Distance

indicative of class C_1. As a result even though the first keyword was strongly indicative of class C_3, the direction changes after the third keyword is processed. It can therefore be isolated as a noisy keyword in relation to the first keyword. In document 5, after the second keyword is processed, the direction changes class C_2 to C_4. Hence it is considered to be noisy in relation to the first keyword. The difference between this case and the situation depicted by document 3 is that here the first keyword was not very strongly indicative of class C_2. Hence a change in direction from C_2 to C_4 could have occurred even if the second keyword were not very strongly indicative of class C_4.

A classifier may be designed which will detect such variations in magnitude or direction, and extract the keywords responsible for causing them. These keywords may then be identified as noisy in relation to the class currently under consideration and discarded. An efficient procedure for implementing such a technique will be discussed in the next chapter. The next section will attempt to provide a partly intuitive and partly mathematical explanation of the phenomena observed experimentally in this chapter.

5.4 Keyword Vectors and Their Relationship to Bayesian Distance

As discussed in Chapter IV, information provided by a keyword can be represented by a t-dimensional vector where t is the number of different classes in the classification scheme. Therefore it is fruitful to investigate the relationship between various keyword vector forms and the effect they have on the Bayesian distance. For the sake of simplicity a three-class problem will be considered. Suppose we have read two keywords whose vectors are shown in Figure 5.3.

k_1: $[p_1, p_2, p_3$ [illegible] $[q_1, q_2, q_3]$.

Let $I = \{1,2,3\}$ denote a class index set. Assume that k_1 has p_1 as its largest component giving a high a posteriori probability for class C_1. The magnitude and direction of the Bayesian distance is then given by:

$$\mathrm{Mag}(1) = \frac{\sum_{i\epsilon I} (p_i)^2}{[\sum_{j\epsilon I} p_j]^2} \tag{5.4}$$

$$\mathrm{Dir}(1) = 1, \text{ assuming } p_1 > p_2, p_1 > p_3$$

If we calculate the a posteriori probabilities α_i of the classes C_1, C_2 and C_3 using these two keywords the following expression is obtained:

$$\alpha_i = \frac{p_i q_i}{\sum_{j\epsilon I} p_j q_j}$$

Suppose k_2 has an associated vector which is not strongly indicative of class C_1 as k_1 is. That is, suppose it is not true that $q_1 > q_2$ and $q_1 > q_3$. In such a case the largest value of α_i will depend on the products p_1q_1, p_2q_2 and p_3q_3. If p_1q_1 is larger than both p_2q_2 and p_3q_3, then α_1 will be larger than α_2 and α_3. If now the q_1 component of k_2 is increased, it becomes more indicative of class C_1.

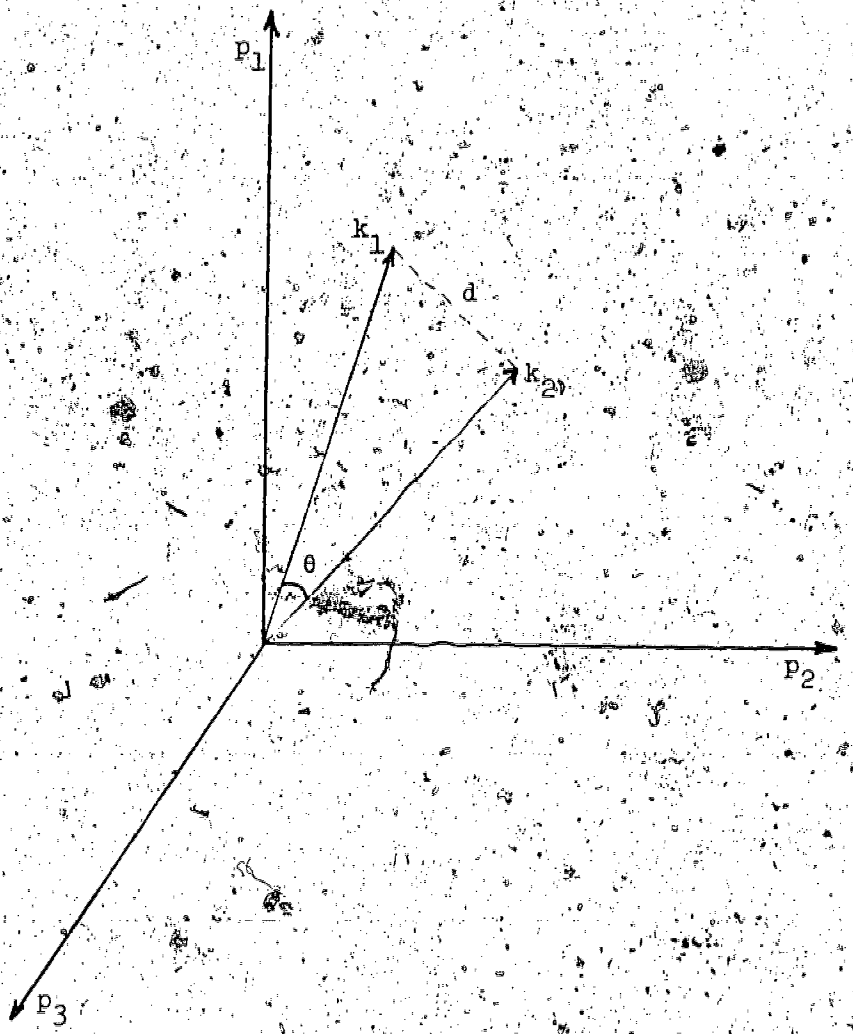


Figure 5.3. Keyword Vector Forms

As q_1 approaches the value of p_1 the Euclidean distance d shown in Figure 5.3 decreases and thus the angle θ also decreases. This causes the value of p_1q_1 to increase while p_2q_2 and p_3q_3 remain the same. As a result α_1 increases relative to α_2 and α_3. Thus k_1 and k_2 together are now more indicative of class C_1. Now consider a case where the angle θ is equal to zero, that is, vectors k_1 and k_2 are coincident. In such a case, the associated vector for k_2 can be represented as a scalar multiple of k_1. Thus $q_1 = \gamma p_1$, $q_2 = \gamma p_2$ and $q_3 = \gamma p_3$. If the a posteriori probabilities of the classes are calculated now using equation (5.5), then we have

$$\alpha_i = \frac{\gamma p_i^2}{\sum_{j \epsilon I} \gamma p_j^2} \qquad (5.6)$$

α_1 will now be greater than α_2 and α_3 because p_1^2 is greater than p_2^2 and p_3^2. That means that if a document were to be classified based on just these two keywords then class C_1 should be chosen. The magnitude of the Bayesian distance should register an increase and the direction should point to class C_1. The new magnitude is given by

$$\mathrm{Mag}(2) = \frac{\sum_{i \epsilon I} \gamma^2 p_i^4}{[\sum_{j \epsilon I} \gamma p_j^2]^2} \qquad (5.7)$$

We note that in both equations (5.6) and (5.7), γ can be cancelled.

Mag(2) in equation (5.7) should be greater than Mag(1) given by equation (5.4). The direction is obviously equal to class C_1

because γp_1^2 is greater than γp_2^2 and γp_3^2. The fact that Mag(2) is greater than Mag(1) is not so obvious. The following theorem proves this fact rigorously.

Theorem 5.1: Given two keywords of the form

$$k_1: \ [p_1, p_2, p_3]$$

$$k_2: \ [\gamma p_1, \gamma p_2, \gamma p_3] \ ,$$

let M_1 and M_2 denote the magnitudes of the Bayesian distance calculated considering keyword k_1 only, and k_1 and k_2 together, respectively; then $M_2 > M_1$, i.e.,

$$\frac{\gamma^4 \sum\limits_{i\varepsilon I} p_i^4}{[\sum\limits_{j\varepsilon I} \gamma^2 p_j^2]^2} > \frac{\sum\limits_{i\varepsilon I} p_i^2}{[\sum\limits_{j\varepsilon I} p_j]^2} \qquad (5.8)$$

Proof: After cancellation of γ, the inequality can be reduced to

$$\frac{\sum\limits_{i\varepsilon I} p_i^4}{[\sum\limits_{j\varepsilon I} p_j^2]^2} > \frac{\sum\limits_{i\varepsilon I} p_i^2}{[\sum\limits_{j\varepsilon I} p_j]^2}$$

Expanding the denominators and omitting the limits of the summations for simplicity we have

$$\frac{\Sigma p_i^4}{\Sigma p_j^4 + 2 \sum\limits_{i>j} p_i^2 p_j^2} > \frac{\Sigma p_i^2}{\Sigma p_j^2 + 2 \sum\limits_{i>j} p_i p_j}$$

Further simplification yields

$$\frac{\sum_{i>j} p_i^2 p_j^2}{\Sigma p_i^4} < \frac{\sum_{i>j} p_i p_j}{\Sigma p_j^2}$$

Expanding both sides we have

$$(p_1^2 + p_2^2 + p_3^2)(p_2^2p_1^2 + p_3^2p_1^2 + p_3^2p_2^2) < (p_1^4 + p_2^4 + p_3^4)(p_2p_1 + p_3p_1 + p_3p_2).$$

Reducing and factorizing we have

$$(p_1 - p_2)p_1p_2(p_2^3 - p_1^3) + (p_2 - p_3)p_3p_2(p_3^3 - p_2^3) + (p_3 - p_1)p_1p_3(p_1^3 - p_3^3)$$
$$+ p_1p_2p_3(3p_1p_2p_3 - p_1^3 - p_2^3 - p_3^3) < 0 .$$

Assuming without loss of generality that $p_1 > p_2 > p_3$, we see that the first three terms of the above inequality are all less than zero. Therefore, we have to show that $Q = p_1^3 + p_2^3 + p_3^3 - 3p_1p_2p_3 < 0$.

If p_1, p_2 and p_3 are all equal, then Q equals zero. If p_1 is now increased, keeping p_2 and p_3 the same, then the value of Q becomes less than zero. Thus if we can prove that Q has no extremal points as p_1 is increased, the inequality will have been established.

Differentiating and setting to zero we have

$$\frac{\partial Q}{\partial p_1} = 3p_1^2 - 3p_2p_3 = 0 \qquad (5.9)$$

$$\frac{\partial Q}{\partial p_2} = 3p_2^2 - 3p_1p_3 = 0 \qquad (5.10)$$

$$\frac{\partial Q}{\partial p_3} = 3p_3^2 - 3p_1p_2 = 0 \qquad (5.11)$$

Now from equation (5.9) we find that

$$p_1^2 = p_2 p_3$$

i.e.,

$$p_1 = \sqrt{p_2 p_3} \, .$$

This is impossible if $p_1 > p_2$ and $p_1 > p_3$. Therefore Q does not have any extremal points, and the theorem is proven.

A stronger version of Theorem 5.1 will now be considered. It was noted in section 5.2 that when a succession of good keywords occur in a document, the Bayesian distance magnitude increases monotonically and the direction remains constant. That is, if a keyword occurs which has the highest probability component for a class C_i followed by another keyword which has the highest component for the same class C_i, then our confidence in C_i as the correct class to which the document belongs increases. This experimental phenomenon can be explained by the theorem which is stated below.

Theorem 5.2: Given two keywords

$$k_1: \; [p_1, p_2, p_3] \quad \text{where } p_1 > p_2 \geq p_3 \, ,$$

$$k_2: \; [q_1, q_2, q_3] \quad \text{where } q_1 > q_2 \geq q_3 \, .$$

Let M_1 be the magnitude of the Bayesian distance calculated using k_1, and M_2 be the magnitude calculated using k_1 and k_2 respectively. Then $M_2 \geq M_1$, i.e.,

$$\frac{\sum\limits_{i \in I} p_i^2 q_i^2}{[\sum\limits_{j \in I} p_j q_j]^2} \geq \frac{\sum\limits_{i \in I} p_i^2}{[\sum\limits_{j \in I} p_j]^2}$$

The proof of Theorem 5.2 will be presented in Appendix C. The appeal of this result lies in the fact that it is counter-intuitive. Let us denote each term of the summation on the right hand side of the inequality above by a_i, i.e.,

$$a_i = \frac{p_i^2}{[\sum_{j\in I} p_j]^2}, \quad i\in I \qquad (5.12)$$

Similarly let b_i denote each term on the left hand side, i.e.,

$$b_1 = \frac{p_i^2 q_i^2}{[\sum_{j\in I} p_j q_j]^2}, \quad i\in I \qquad (5.13)$$

We note that

$$\sum_{i\in I} a_i = 1 \text{ and}$$

$$\sum_{i\in I} b_i = 1 .$$

M_1 and M_2 therefore, can now be written as

$$M_1 = a_1^2 + a_2^2 + a_3^2 \qquad (5.14)$$

$$M_2 = b_1^2 + b_2^2 + b_3^2 . \qquad (5.15)$$

If the conditions

$$p_1 > p_2 \geq p_3 \text{ and}$$

$$q_1 > q_2 \geq q_3$$

are satisfied then the following inequalities are true:

$$a_1 > a_2 \geq a_3 ,$$

$$b_1 > b_2 \geq b_3 , \text{ and}$$

$$b_1 > a_1 .$$

Since $b_1 > a_1$; $(b_2 + b_3)$ must be less than $(a_2 + a_3)$. The value $(a_2^2 + a_3^2)$ may be increased by making a_2 large and a_3 small. Similarly the sum $(b_2^2 + b_3^2)$ may be decreased by making b_2 equal to b_3. It appears therefore that even though b_1^2 is greater than a_1^2, $(b_2^2 + b_3^2)$ may be made sufficiently smaller than $(a_2^2 + a_3^2)$ such that M_2 may be actually smaller than M_1. The theorem states that such a situation cannot occur. In short this theorem states that for the purposes of classification, if two keywords are more indicative of one class than other classes, then they are good keywords in relation to each other. The inequality above assures us that in such a case the magnitude of the Bayesian distance will increase and the direction will remain constant.

However, it should be noted here that the constraint imposed on keyword k_2 for a given keyword k_1 such that M_2 may be greater than M_1 represents only sufficient conditions. Let k_1, as before, be given by

$$k_1 : [p_1, p_2, p_3] , \quad p_1 > p_2 \geq p_3 .$$

Let k_2 now be given by

$$k_2 : [q_1, q_2, q_3] , \quad q_2 > q_1 \geq q_3$$

We note that k_2 now is not a good keyword in relation to k_1 because the index of its highest probability component is different from that for k_1. Under such circumstances can M_2 be greater than M_1?

If the values of q_1, q_2 and q_3 are such that $p_1q_1 > p_2q_2 \geq p_3q_3$, then the quantities a_1, a_2, a_3, b_1, b_2 and b_3 defined earlier will still satisfy the inequalities

$$a_1 > a_2 \geq a_3 ,$$

$$b_1 > b_2 \geq b_3 , \quad \text{and}$$

$$b_1 > a_1 .$$

Hence depending on the values of a_2, a_3, b_2 and b_3 it is possible that $(b_1^2 + b_2^2 + b_3^2)$ is greater than $(a_1^2 + a_2^2 + a_3^2)$. Thus M_2 is greater than M_1. This can be illustrated better by an example. Let k_1 and k_2 be given by

$$k_1: \quad [0.5, 0.4, 0.1]$$

$$k_2: \quad [0.41, 0.5, 0.00001].$$

Here we see that the index of the highest probability component for k_1 is class C_1 and for k_2 it is C_2. Thus the constraints stated in Theorem 5.2 are no longer satisfied. However, if the Bayesian distances are calculated, we see that Dir(1) = Dir(2) = 1 and

$$M_1 = 0.42 ,$$

$$M_2 = 0.504 .$$

Therefore

$$M_2 > M_1 .$$

It was pointed out earlier that a monotonically increasing pattern in the magnitude of the Bayesian distance and a constant direction could be utilized to recognize documents which are ideal in nature. The discussion above illustrates that in addition to these conditions, the index of the highest probability component for each keyword should be checked to see whether they are all the same. If they are, then the keywords can be considered to be good keywords in relation to each other and the document can be considered to be ideal in nature.

The two theorems stated above explain some of the experimental phenomena observed in the earlier sections of this chapter. The next chapter develops a classification algorithm based on these observed phenomena. Keywords are extracted sequentially from a document and at each stage the magnitude and direction of the Bayesian distance are calculated. Changes in these quantities are observed in order to isolate noisy keywords and identify clusters of similar keywords. If after removal of the noisy keywords, a monotonically increasing pattern is observed in the Bayesian distance magnitudes, as in the case of an ideal document, classification is attempted. Description of how such a method may be implemented is given in the next chapter.

CHAPTER VI

THE REVISED SEQUENTIAL METHOD: PRIMARY CLASSIFICATION

The previous chapter showed how the Bayesian distance measure could be used to detect the presence of noisy keywords and to identify groups of good keywords in a document. The basic sequential algorithm discussed in Chapter III can now be modified to achieve classification by taking into account only the good keywords. This chapter is devoted to the design of such an algorithm. The algorithm works in two phases. In the first phase keywords are extracted sequentially from a document and, based on the Bayesian distance analysis, the total number of keywords read are divided into two groups--the good keywords and the noisy keywords. The first section of this chapter discusses a method which achieves such a separation of the keywords. When the good keywords are such as to meet the Q and S thresholds discussed in the previous chapter, the algorithm analyzes these words to obtain a primary class for the document. In the second phase the noisy keywords are analyzed to see whether these are indicative of another class. If so then the document is classified into a secondary class. This second phase of the algorithm will be discussed in Chapter VII.

6.1 Separation of Good and Noisy Keywords

Each keyword extracted from a document has the form k_j: $[p_1, p_2, \ldots, p_t]$ where p_i denotes the probability $p(k_j/C_i)$. These probability values are obtained from the conditional probability matrix CP discussed in Chapter III. Besides a probability vector of the form shown above, each keyword has also associated with it an index I, representing its location in the document. The first keyword read from a document has an index 1, the second keyword has an index 2, and so on. Each stage of the sequential process now consists of the following.

(i) Q keywords are read from a document where the parameter Q represents a threshold. The D_B magnitude has to increase monotonically over a sequence of Q keywords in order for a document to be classified in a primary class. This parameter was introduced and discussed in Chapter V.

(ii) The indices and the associated probability vectors for each of these Q keywords are stored in an input buffer.

In order to separate the noisy keywords from the good ones, two auxiliary buffers are set up. Each of these buffers are capable of storing the index of a keyword where the index serves as a pointer to an entry in the input buffer. The first auxiliary buffer will be used to store the indices of the good keywords, and will be called the good keyword buffer (GBUF). The second auxiliary buffer will be used to contain the indices of the noisy keywords, and will be referred to as the noisy keyword buffer (NBUF).

Initially let us assume that all the Q keywords read at any stage of the sequential process are good keywords in that they are all indicative of one class. Therefore the indices of these Q keywords are tentatively loaded in the GBUF. Suppose W_1, W_2, ..., W_Q are the Q keywords whose indices are in the GBUF. Each W_i is of course a member of the keyword set $K = \{k_1, k_2, \ldots, k_m\}$. In order to check whether a monotonically increasing pattern in the magnitude of D_B is obtained over these Q keywords, the Bayesian distances are calculated using keyword W_1, then keywords W_1 and W_2, and so on. The form of such a monotonically increasing pattern was shown in Figure 5.1 of the previous chapter. If at any stage a particular keyword causes the magnitude to decrease or the direction of D_B to change, it is tagged as a noisy word. Its index is then removed from the GBUF and put in the NBUF. For example, suppose four keywords W_1, W_2, W_3 and W_4 have been read. The indices 1, 2, 3 and 4 are tentatively loaded into the GBUF. Further suppose that keywords W_1 and W_2 yield a monotonically increasing pattern in the magnitude of the Bayesian distance and a constant direction. It is then assumed that they are good keywords in relation to each other. Now let keyword W_3 be such that it causes a decrease in the magnitude. Then it is tentatively tagged as a noisy keyword and the index 3 is removed from the GBUF and put in the NBUF. The situation at this stage is depicted by Figure 6.1. The process of calculating the Bayesian distance is repeated using keywords W_1, W_2 and W_4 to check whether W_4 is a good or a noisy keyword. The following section describes how the Bayesian distances are calculated and how the magnitudes and directions are tested at each stage to identify the noisy keywords.

6.2 The Bayesian Distance Calculator and Noise Detector

Let Mag(i) and Dir(i) denote the magnitudes and directions respectively, calculated at the end of the i^{th} keyword in the GBUF. This is done by first calculating the α_j values using the first i keywords in the GBUF.

$$\alpha_j = \frac{\sum_{r=1}^{i} p(W_r/C_j)}{\sum_{j=1}^{t} [\prod_{r=1}^{i} p(W_r/C_j)]}, \quad j = 1 \text{ to } t. \qquad (6.1)$$

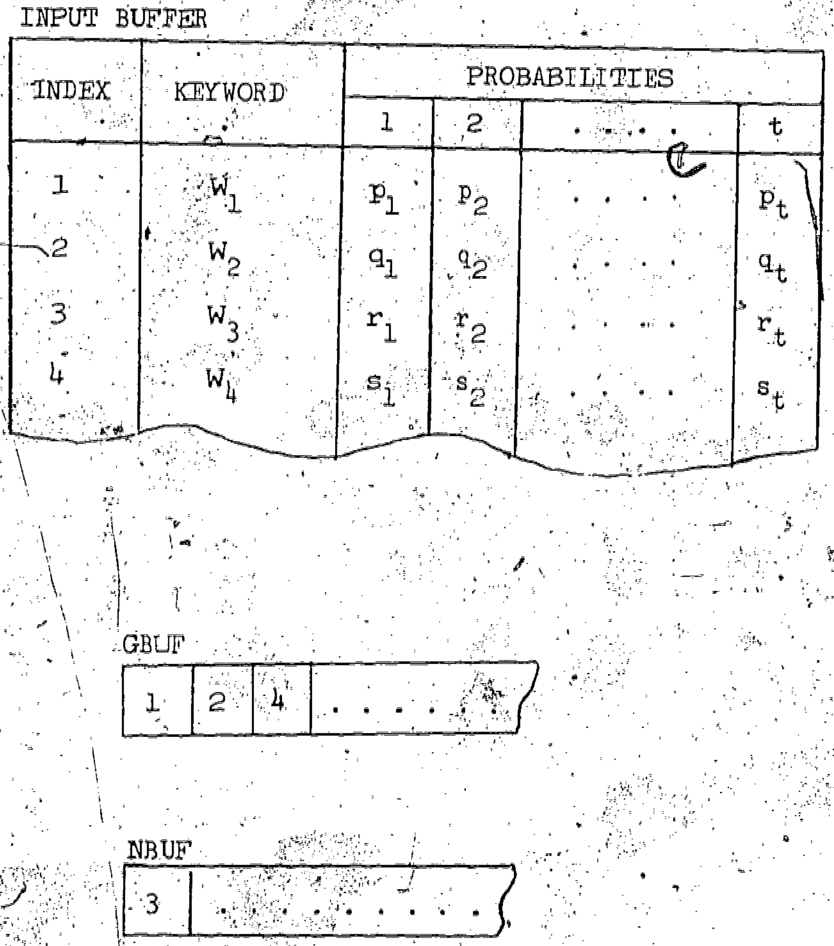
Mag(i) and Dir(i) are then calculated using these values as follows:

$$\text{Mag}(i) = \sum_{j=1}^{t} (\alpha_j)^2 \qquad (6.2)$$

$$\text{Dir}(i) = \text{index of the highest } \alpha_j \text{ value.}$$

For each value of i greater than two a noise detection procedure is implemented. If either of the following two conditions

(i) $\text{Mag}(i) < \text{Mag}(i - 1)$, or



INPUT BUFFER

INDEX	KEYWORD	PROBABILITIES			
		1	2	. . . . .	t
1	W_1	p_1	p_2	. . . .	p_t
2	W_2	q_1	q_2	. . . .	q_t
3	W_3	r_1	r_2	. . . .	r_t
4	W_4	s_1	s_2	. . . .	s_t

GBUF

1	2	4	. . . . .

NBUF

3	. . . .

Figure 6.1 Input Buffer, Good Keyword Buffer and Noisy Keyword Buffer

(ii) $\mathrm{Dir}(i) \neq \mathrm{Dir}(i - 1)$,

is detected then the i^{th} keyword in the GBUF is assumed to be noisy. Its index is put in the NBUF and removed from the GBUF. The index of the next keyword in the input buffer is then loaded into the GBUF and the process is repeated. When the following conditions are found:

(i) $\mathrm{Mag}(1) < \mathrm{Mag}(2) < \ldots < \mathrm{Mag}(Q)$,

(ii) $\mathrm{Mag}(Q) \geq S$ where S is the preset saturation value discussed in section 5.2, and

(iii) $\mathrm{Dir}(1) = \mathrm{Dir}(2) = \ldots = \mathrm{Dir}(Q)$,

then a monotonically increasing pattern in the magnitudes of the Bayesian distances, with the direction remaining constant, is obtained. At this point there are Q keywords in the GBUF which are good keywords in relation to each other and which between them indicate a unique class. This class, whose index is given by Dir(Q), is identified as the primary class for the document. In case one of the three conditions listed above is not satisfied and the document contains no more keywords to be read, then it is termed unclassifiable.

It has been shown that after reading each keyword a decision is made as to whether it should be considered good or noisy depending on the direction and magnitude of the Bayesian distance. Based on this decision the index of the word is either kept in the GBUF or is put in the NBUF. Since a keyword is good or noisy in relation to the other keywords that have preceded it in the document, it is quite possible that at any stage of the sequential process, the NBUF contains the good keywords which point to the correct primary class. This can be better illustrated by an example.

Suppose we have read a series of keywords W_1, W_2, W_3, W_4 and W_5. Then let us assume that the following sequence of actions have been taken:

(i) the index of W_1 is put in the GBUF,

(ii) W_2 is good in relation to W_1 and so its index is put in the GBUF, and

(iii) W_3, W_4 and W_5 are noisy in relation to W_1 and W_2, and so their indices are put in the NBUF.

At this point the GBUF contains the indices of W_1 and W_2, while the NBUF contains the indices of W_3, W_4 and W_5. Suppose W_1 and W_2 are indicative of class C_1 and W_3, W_4 and W_5 are all indicative of class C_j. If now the magnitude of D_B calculated by using W_3, W_4 and W_5 exceeds the value of the magnitude calculated by using W_1 and W_2 in the GBUF, it is highly likely that W_3, W_4 and W_5 are the set of good keywords which point to the correct primary class. If this is the case, then the indices of the words in the GBUF should be interchanged with those in the NBUF. To achieve this, at every stage when the number of words in the NBUF equals that in the GBUF, the Bayesian distances of the words in it are calculated. If a unique direction is obtained and if the magnitude exceeds the magnitude of the Bayesian distance calculated by using the words in the GBUF then the two buffers are interchanged.

The concepts discussed in this section have been implemented as a primary classification algorithm. The next section briefly discusses this algorithm.

6.3 Description of the Primary Classification Algorithm

The first phase of the classification algorithm which obtains a primary class for a test document contains several parts, the most important of which are the primary classifier, the Bayesian distance calculaton and the noise detector. In this section we will briefly discuss these components of the algorithm. The portion of the algorithm which extracts keywords from a document will be described for the sake of completeness. A more comprehensive description will be given in Appendix D. A flowchart of the algorithm is given in Figure 6.2.

6.3.1 The Keyword Extractor

This portion of the algorithm reads one keyword at a time from a given test document and stores the following in an input buffer:

(i) a value i corresponding to the index of the keyword read;

(ii) the keyword; and

(iii) the probability values associated with the keywords.

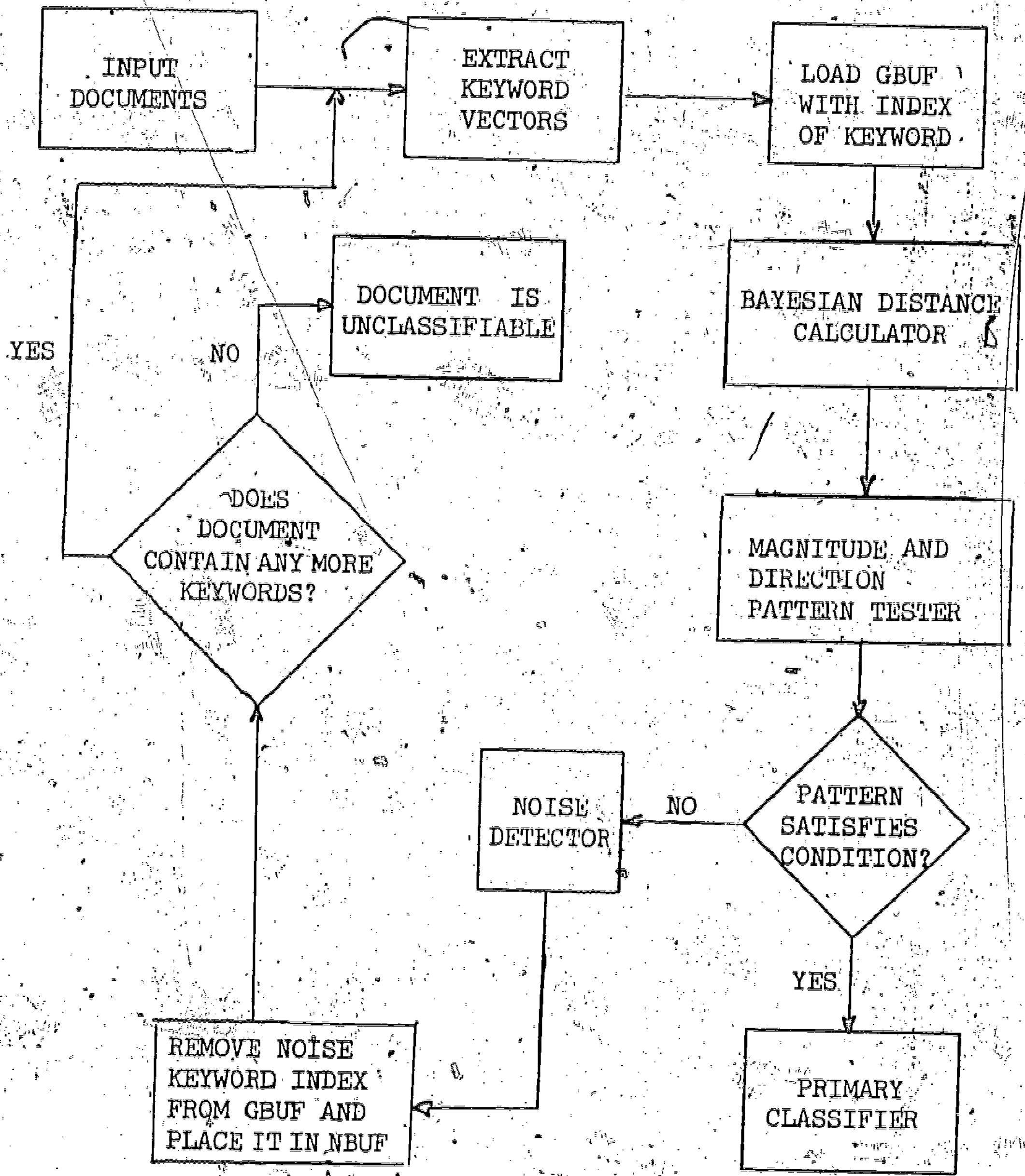

INPUT DOCUMENTS
EXTRACT KEYWORD VECTORS
LOAD GBUF WITH INDEX OF KEYWORD
BAYESIAN DISTANCE CALCULATOR
MAGNITUDE AND DIRECTION PATTERN TESTER
PATTERN SATISFIES CONDITION?
YES
PRIMARY CLASSIFIER
NO
NOISE DETECTOR
REMOVE NOISE KEYWORD INDEX FROM GBUF AND PLACE IT IN NBUF
DOES DOCUMENT CONTAIN ANY MORE KEYWORDS?
YES
NO
DOCUMENT IS UNCLASSIFIABLE

Figure 6.2 Flowchart of the Primary Classification Algorithm

These are stored in a matrix $P(i,j)$, where i represents the index of the keyword and j represents the index of a category.

6.3.2 The Bayesian Distance Calculator

The input to this portion of the algorithm consists of a set of indices S_T, which comprise either the set of indices of the keywords in the GBUF or the set of indices of the keywords in the NBUF. Let this set S_T at any given stage be $S_T = \{i_1, i_2, \ldots, i_r\}$. Then the Bayesian distance calculator computes the magnitudes and directions of D_B considering the keywords W_{i_1}, then W_{i_1} and W_{i_2}, then W_{i_1}, W_{i_2} and W_{i_3}, and so on. The magnitudes and directions $Mag(i_1)$, $Mag(i_2)$, ..., $Mag(i_r)$ and $Dir(i_1)$, $Dir(i_2)$, ..., $Dir(i_r)$ are stored in an array for analysis by the noise detector.

6.3.3 The Noise Detector

Using the array of magnitudes and directions calculated by the Bayesian distance calculator the noise detector checks for a monotonically increasing pattern in the magnitudes. If this is satisfied it checks to see whether the directions are all the same. Suppose in the array of magnitudes and directions one or both of the following two conditions

$$\text{(i) } Mag(i) < Mag(i - 1)$$

$$\text{(ii) } Dir(i) \neq Dir(i - 1)$$

is detected, then the i^{th} keyword is identified as a noisy keyword.

6.3.4 The Primary Classifier

The input to this section of the algorithm consists of the array of magnitudes and directions calculated by the Bayesian distance calculator and the indices of the noisy keywords computed by the noise detector. The parameters Q and S, discussed earlier, govern the operation of this section of the algorithm. The primary classifier performs the following functions.

(i) It loads the set S_T with a set of Q indices from the GBUF and uses the Bayesian distance calculator to compute an array of magnitudes and directions.

(ii) It uses the noise detector to identify a noisy keyword in the set of Q keywords obtained from the GBUF.

(iii) If the noise detector identifies a noisy keyword, it removes the index of this keyword from the GBUF and places it in the NBUF. It then uses the keyword extractor algorithm to obtain an additional keyword. If there is such a keyword its index is loaded into the GBUF. If there are no more keywords in the document then it identifies the document as being unclassifiable.

(iv) Each time the index of a word is loaded into the NBUF it checks to see whether the size of the NBUF equals that of the GBUF. If so, then it checks to see whether, based on the Bayesian distance values, the contents of the NBUF should be interchanged with those of the GBUF. The criterion which governs such an action has been discussed in section 6.3.

(v) If the noise detector does not identify a noisy keyword in the Q keywords present in the GBUF, the primary classifier checks to see whether Mag(Q) ≥ S where S is the saturation value. If so, then the class given by Dir(Q) is identified as the primary class for the document. If Mag(Q) < S, then an additional keyword is read. If the document does not contain any more keywords, the primary classifier still classifies it into the class given by Dir(Q) but records the fact that the S value was not satisfied. This is to signify that the confidence in the primary class obtained for this document is less than those which have satisfied the S threshold.

This algorithm has been implemented to classify approximately 500 documents contained in one of the releases of the SPIN data base. A brief description of this data was given in Chapter III. The following section presents the results that have been obtained.

6.4 Implementation of the Algorithm and Results of Experiments

Since each of the SPIN documents has been preclassified by the American Institute of Physics (AIP), an estimate of how well the algorithm has performed could be easily obtained. This was done by comparing the classification obtained by the algorithm with that given by the AIP.

As pointed out in section 6.3, the algorithm has two parameters Q and S. The parameter S denotes the final value of the Bayesian distance at the point when a document is classified. It represents a measure of confidence in the classification that is obtained. The parameter Q represents the number of keywords over which the Bayesian distance has to satisfy a monotonically increasing pattern in the magnitude, while maintaining a constant direction. Initially S was varied from 0.8 to 1.0 for different values of Q, viz., Q = 2, 3, and 4. In all these cases it was found that if a document satisfied the Q threshold then the final value of the Bayesian distance, i.e., Mag(Q), always resided in the interval 0.9 to 1.0. Variation of S therefore, did not produce any significant change in the results and hence for all subsequent experiments a fixed value of S = 0.9 was used. Before discussing the results of these experiments further, several terms need to be defined.

Let

D = number of documents correctly classified

T = total number of documents in the data base

V = number of documents found unclassifiable.

Then

$$A_1 = \text{accuracy over those classified} = \frac{D}{T - V} \times 100$$

$$A_2 = \text{overall accuracy} = \frac{D}{T} \times 100.$$

The first set of experiments conducted used a fixed value of

Q. For values of Q = 2, 3 and 4 the results are presented in Table 6.1. As can be expected, an increase in Q results in higher values for A_1. This is because as Q increases more keywords are examined before a document is classified. Thus for Q = 2 only two good keywords are needed before a primary class is identified. In many cases this might lead to a precipitous decision. Keywords occurring in the rest of the document may point to an entirely different class which in many cases may be the correct class. An increase in Q avoids such precipitous decisions and hence increases the value of A_1.

However, an increase in Q means a document has to have a greater number of good keywords in order to be classified. Fewer documents are able to satisfy this more stringent criterion. Hence the number of unclassified documents increases and the value of overall accuracy (A_2) decreases.

Some of the unclassified documents are such that they contain a set of good keywords indicative of a unique class. They are identified as unclassifiable by the algorithm because the number of good keywords contained in them is not high enough to satisfy a fixed Q threshold. Therefore in the next set of experiments, the value of Q was dynamically varied during classification. Table 6.2 presents the results of these experiments. If a particular document did not satisfy a given Q threshold then the value of Q was reduced by one and the document was reconsidered for

Table 6.1 Primary Classification with Fixed Q

T = 500

Q	NUMBER OF DOCUMENTS CORRECTLY CLASSIFIED D	NUMBER OF DOCUMENTS INCORRECTLY CLASSIFIED T - (D + V)	NUMBER OF DOCUMENTS FOUND UN-CLASSIFIABLE V	ACCURACY OVER THOSE CLASSIFIED A_1	OVERALL ACCURACY A_2
2	326	116	57	73.9%	65.2%
3	340	62	97	84.6%	69.7%
4	301	51	147	85.5%	61.7%

Table 6.2 Primary Classification with Dynamic Q

T = 500

RANGE OF Q	NUMBER OF DOCUMENTS CORRECTLY CLASSIFIED D	NUMBER OF DOCUMENTS INCORRECTLY CLASSIFIED T - (D + V)	NUMBER OF DOCUMENTS FOUND UN-CLASSIFIABLE V	ACCURACY OVER THOSE CLASSIFIED A_1	OVERALL ACCURACY A_2
1-3	380	105	14	78.4%	77.9%
1-4	386	71	42	84.5%	79.1%

classification under the following conditions:

(i) the document has been read entirely, and

(ii) no noisy keyword has been identified in the entire document.

For example, suppose the starting value of Q is three, and a test document contains only two keywords, both of which are good keywords in relation to each other, i.e., they are both indicative of one class. Then the value of Q would be reduced to two. The primary classifier would then attempt to identify a primary class for the document based on these two keywords.

The results presented in Table 6.2 indicate that some of the previously unclassified documents are now correctly classified. Therefore the value of overall accuracy increases. However, because of the reduction in Q, some of the previously unclassified documents are now incorrectly classified. Therefore the value of A_1 tends to decrease.

6.5 Secondary Classification

The discussion in this chapter has dealt with the design and implementation of an algorithm which obtains primary classes for a set of documents. A section of this algorithm separates the noisy keywords from the good keywords in a given document. The noisy keywords are put in a separate buffer called the NBUF. In effect this means that at the end of primary classification two different groups of keywords are obtained. The good keyword group is analyzed to obtain a primary class. But how about the group of keywords that have been tagged as noisy? If the contents of a document are such that a secondary class could be identified, then it is very likely that the keywords that have been isolated as noisy may actually be indicative of such a class. If this is the case, then the words in the NBUF could be analyzed to obtain a possible secondary class. The next chapter addresses itself to this problem and outlines an algorithm for obtaining secondary classification.

CHAPTER VII

A TECHNIQUE FOR SECONDARY CLASSIFICATION

The previous chapter has dealt with the design and implementation of a classification algorithm based on the Bayesian distance measure. This algorithm classifies a document into a class which is denoted as the primary class. Besides-obtaining a primary class, an added feature of this technique is that it effects a separation of the good keywords and the noisy keywords into two different groups. At the end of the primary classification the noisy keywords are contained in the noisy keyword buffer (NBUF). These words may be unrelated to each other in that they might not point to any one class, or they may form a coherent cluster in such a way that between them they are indicative of a category which may be identified as a secondary class. This chapter addresses itself to the design of a method that will analyze the words in the NBUF to explore the possibility of classifying a document into a secondary class. It will be pointed out that in many cases the keywords in the NBUF alone are not adequate to obtain such a classification. In these cases keywords are selectively extracted from the good keyword buffer to corroborate the information obtained from the words in the noisy buffer.

7.1 Analysis of Keywords in the Nosiy Buffer

At the end of primary classification the NBUF contains the indices of the keywords which have been identified as noisy keywords by primary classification. Let this set of noisy keywords be denoted by

$$N = \{W_1, W_2, \ldots, W_r\}.$$

The philosophy underlying the analysis of the words in the NBUF is the same as that used for the analysis of the words in the GBUF. The values of the magnitudes and directions of the Bayesian distance are successively calculated using words W_1, then W_1 and W_2, and so on. Let Mag(i) and Dir(i) represent the magnitude and direction respectively, calculated by using the first i keywords in the NBUF. For each value of i the noise detector discussed in section 6.3 of the previous chapter is invoked to check whether a constant direction and a monotonically increasing pattern in the magnitudes is obtained. As in the primary classification algorithm, the number of keywords over which a monotonically increasing pattern has to be satisfied is denoted by Q. If the noise detector detects one of the following situations

$$\text{(i) } \mathrm{Mag}(i - 1) > \mathrm{Mag}(i), \quad \text{or}$$

$$\text{(ii) } \mathrm{Dir}(i - 1) \neq \mathrm{Dir}(i),$$

then the i^{th} keyword in the NBUF is considered to be noisy with respect to the i - 1 keywords preceding it. The index of this keyword is then removed from the NBUF and placed in a miscellaneous buffer referred to as MBUF. The Bayesian distance is again computed for the remaining words in the NBUF to check whether there are any more noisy keywords. At the end of this process, the NBUF contains a set of keywords which are all good with respect to each other in that they are all indicative of one class. Let these be denoted by

$$N' = \{W_1', W_2', \ldots, W_r'\}.$$

If there are Q such good keywords in the NBUF, i.e., if $r \geq Q$ and if $\text{Mag}(Q) \geq S$, then a secondary class is identified as Dir(Q).

One problem that is encountered by this technique is that after the noisy keywords have been eliminated from the NBUF the number of keywords remaining in it may not be sufficient to satisfy the Q threshold. Under such circumstances, this technique would not attempt to obtain a secondary class. This might be very restrictive. One way to circumvent this problem would be to selectively choose keywords from the good keyword buffer to corroborate the information obtained from analyzing the words in the NBUF. The next section outlines a method for doing this.

7.2 Analysis of the Words in the Good Buffer

Let us assume that the keywords in the NBUF do not satisfy the Q threshold but yield a constant direction and a monotonically increasing pattern in the D_B magnitude. If this fixed direction is that of class C_j, then C_j is treated as a potential secondary class. Let us assume that category C_i has been chosen as the primary class. The words in the GBUF are now extracted and, along with their associated probability values obtained from the input buffer, are stored in a matrix as shown in Figure 7.1. Column j in this matrix contains the probability values corresponding to class C_j. Any keyword which has a non-default probability value in this column is a keyword which might be used to provide information about class C_j. Therefore any keyword which has a non-default probability value in column j is isolated from this matrix. Such a keyword will have the form:

$$[p_1, \ldots, p_i, \ldots, p_j, \ldots, p_t]$$

where p_j is greater than the default value. Since this keyword is a good keyword for the primary class, p_i is also greater than the default value. Let the set of these keywords be denoted by $G = \{W_1, W_2, \ldots, W_s\}$. Each of these keywords is indicative of classes C_i and C_j. Since, however, each of these keywords is a good keyword for the primary class, in general the probability component p_i should be greater than the p_j component. That is, all these keywords are probably more indicative of class C_i than of class C_j. A measure of how strongly indicative they are of class C_j can be obtained as follows.

Without loss of generality, suppose just one keyword W_1 in G were used to classify the document. The a posteriori probability

			PRIMARY CLASS		POSSIBLE SECONDARY CLASS	
1	p_1	. . .	p_i	. . .	p_j	p_t
2						
⋮						
n	q_1		q_i		q_j	q_t

Figure 7.1 Keywords in GBUF and their Associated Probability Vectors

of the classes, the α_j values, can be computed by equation (7.1):

$$\alpha_j = \frac{p_j}{\sum_{r=1}^{t} p_r}, \quad j = 1 \text{ to } t. \tag{7.1}$$

Since this is a good keyword for class C_i, α_i is a maximum of the α_j values. A measure of how strongly W_1 relates the document to class C_j can be obtained by computing the ratio X_T where

$$X_T = \frac{\alpha_j}{\alpha_i} = \frac{p_j}{p_i}. \tag{7.2}$$

Thus while choosing keywords from G to corroborate the information given by the keywords in the NBUF, this ratio may be compared with a preset threshold S_T. A keyword is chosen only if the corresponding ratio X_T exceeds S_T. The parameter S_T may be varied to extract keywords from G very selectively in that if S_T is increased the keywords that are chosen will be more highly indicative of class C_j. For a given classification experiment, S_T is fixed at a certain value.

Let $G' = \{W_1', W_2', \ldots, W_n'\}$ be the set of keywords which satisfy the S_T threshold. G' is then merged with the set N' (see section 7.1) of keywords present in the NBUF. Since the primary class C_i has already been chosen, the i^{th} probability component of all these keywords is set to the default value. If there are at least Q keywords in the merged set $G' \cup N'$, then the indices of these keywords are loaded into the NBUF and the procedure outlined in section 7.1 is repeated to determine whether C_j can be chosen as a secondary class. If not, then it is assumed that a secondary class does not exist.

7.3 Results of Experiments

The first set of experiments conducted to obtain secondary classification concentrated exclusively on the words in the noisy buffer after the primary class had been obtained. The method has been outlined in section 7.1. Initially the conditions imposed for secondary classification were as stringent as those for the primary case, i.e., a value of $S = 0.9$ and $Q = 3$ was used. The results, shown in Table 7.1, indicate that very few documents were classified in a secondary class. A reduction of Q from three to two substantially increased this number. Some of the documents that were classified by the algorithm were also assigned a secondary class by the AIP. Table 7.1 indicates that all of these documents were assigned the correct secondary class by the algorithm.

In order to increase the number of documents which could be classified in a secondary class by the algorithm, a second set of experiments was conducted by combining the words in the NBUF with selected words from the GBUF. A method for doing this was discussed in section 7.2. If the NBUF did not have enough keywords to satisfy a threshold of $Q = 3$, then Q was reduced to two by the method discussed in section 6.4. Table 7.2 presents the results obtained by varying the parameter S_T from 0 to 0.8. As the value of S_T increases, fewer documents are classified into a secondary class by the algorithm.

Table 7.1 Secondary Classification Using Words in NBUF

Number of SPIN documents classified in a secondary class by the American Institute of Physics (AIP) = 101

Q	NUMBER CLASSIFIED BY ALGORITHM	NUMBER CLASSIFIED BY A.I.P.	NUMBER CORRECTLY CLASSIFIED
3	7	5	5
2	18	10	10

Table 7.2 Secondary Classification Using Words in NBUF and GBUF

S_T	NUMBER CLASSIFIED BY ALGORITHM	NUMBER CLASSIFIED BY A.I.P.	NUMBER CORRECTLY CLASSIFIED
0	211	49	44
0.2	163	44	39
0.5	117	34	30
0.8	73	26	23

This is because as S_T is increased, keywords that are extracted from the GBUF have to be more indicative of only the potential secondary class in order to be accepted for consideration. Since fewer keywords can satisfy the S_T threshold, there exist a smaller number of documents having an adequate number of keywords to satisfy the Q threshold. We note that Tables 7.1 and 7.2 do not have entries for classification accuracy as in the case of primary classification. This is because determination of accuracy for secondary classification is not as straightforward as it was in the case of primary classification. This problem is discussed in the next section.

7.4 Validation of Results Obtained by the Secondary Classification Algorithm

In order to evaluate the performance of the secondary classification algorithm, it is necessary to compare the results obtained with those given by a standard classification scheme. In the case of primary classification this was done by comparing the results with the classes given by the American Institute of Physics (AIP). In the case of secondary classification this has not been possible because AIP has classified only 103 of the 500 SPIN documents into a secondary class. In many cases the Bayesian distance algorithm assigns a secondary class to a larger number of documents. The situation can be best depicted by a Venn diagram shown in Figure 7.2, where D_a represents the set of documents for which a secondary class has been assigned by the AIP, and D_b represents the set of documents

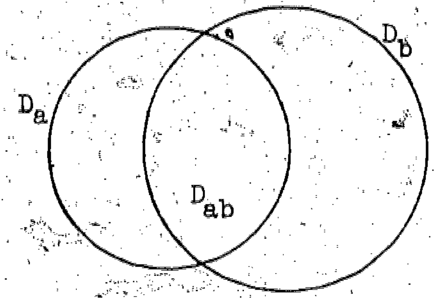


Figure 7.2 Venn Diagram Representation of Secondary Classification

for which a secondary class has been assigned by the Bayesian distance algorithm. For the set of documents D_{ab}, i.e., those that have been assigned a secondary class by both schemes, comparison is straightforward. Such a comparison can be found in the third and fourth columns of Tables 7.1 and 7.2. The third column represents the set D_{ab}, and the fourth column represents the number of documents in D_{ab} that have been correctly classified by the algorithm.

The problem is to obtain a method by which classifications obtained for documents lying in the non-intersection regions of these two sets can be compared. For this purpose the results need to be examined from a different viewpoint. The following two experiments have been conducted to compare the classifications.

Experiment 1

(i) The document sets D_a and D_b are listed along with their primary and secondary classes as shown in lists L_1 and L_2 below.

L_1

D_a	primary class	secondary class
d_{a1}	C_{i1}	C_{j1}
.	.	.
.	.	.
.	.	.
d_{an}	C_{in}	C_{jn}

L_2

D_b	primary class	secondary class
d_{b1}	C_{p1}	C_{q2}
.	.	.
.	.	.
.	.	.
d_{bm}	C_{pm}	C_{qm}

(ii) For each class pair (C_i, C_j), the set of documents that have been assigned to both these classes is obtained from list L_1. Based on the keywords contained in these documents, a standard class correlation matrix, S_a, is computed as follows.

Let

N_i = total number of different keywords contained in documents assigned to class C_i

N_j = total number of different keywords contained in documents assigned to class C_j

N_{ij} = total number of different keywords contained in documents assigned to both classes C_i and C_j

Then

$$S_a(i,j) = \frac{N_{ij}}{N_i + N_j - N_{ij}} \qquad (7.3)$$

(iii) The procedure outlined in (ii) is repeated for list L_2 from which a correlation matrix S_b is obtained.

Since S_a is the class correlation matrix obtained from the AIP classification, and S_b is obtained from the results given by the Bayesian distance algorithm, a comparison of the two matrices should give an indication of how well the Bayesian distance classifier has performed. This comparison is done by computing the mean square difference between the matrices S_a and S_b using equation (7.4). We note that S_a and S_b are symmetric matrices.

$$MSQ = \frac{2}{t(t-1)} \left[\sum_{i=1}^{t-1} \sum_{j=i+1}^{t} [S_a(i,j) - S_b(i,j)]^2 \right]. \qquad (7.4)$$

The smaller the value of the mean square difference, the more nearly equal will be the matrices S_a and S_b. Hence the two classifications will be more similar to each other.

The problem now is to relate the mean square difference between the correlation matrices to classification accuracy. We note that in the case of primary classification, accuracy could be directly obtained by comparing the class indicated by the Bayesian distance method with that assigned by the AIP for each document. This information can be used to relate mean square difference to classification accuracy in the following way.

For each of the experiments conducted for primary classification (see Tables 6.1 and 6.2 of the previous chapter), obtain the class correlation matrix S_B. Then compute the mean square difference between S_a and S_B. Let the range of variation of the mean square differences be N_1 to N_2. Let the range of variation of classification accuracy for these experiments be A_1 to A_2. Then it is conjectured that for a given secondary classification obtained by the Bayesian distance algorithm, if the mean square difference between S_a and S_b

lies within the range N_1 and N_2 then the classification accuracy also lies approximately within the range A_1 and A_2.

The mean square differences for each of the secondary classification obtained for values of $S_T = 0$, 0.2, 0.5 and 0.8 is given in Table 7.3. Similarly the mean square differences for a set of four primary classification experiments with known accuracy of classification is given in Table 7.4. From Table 7.4 we see that as the classification accuracy decreases the mean square difference increases, ranging from 0.00352 to 0.00950. From Table 7.3 it is seen that for values of $S_T = 0.2$, 0.5, 0.8, the mean square differences for secondary classification lie within this range. It is therefore conjectured that the classification accuracy for these experiments lies between 61% to 79%. For $S_T = 0$, it is seen that the mean square difference is much higher and lies outside the range 0.00352 to 0.00950. It is therefore conjectured that the accuracy for this run is probably less than 60%. To validate these conjectures, another experiment was conducted involving manual classification of documents into a secondary class.

Experiment 2

Two individuals well versed in the area of physics were asked to read each of the 500 documents in the SPIN data base. They were apprised of the primary class assigned by the AIP in each case. They were asked to assign a secondary class whenever they felt that such an assignment would be appropriate. The results obtained for each of

Table 7.3 Mean Square Difference of Secondary Classification

S_T	MEAN SQUARE DIFFERENCE BETWEEN S_a AND S_b
0.0	0.01100
0.2	0.00711
0.5	0.00477
0.8	0.00373

Table 7.4 Mean Square Difference for Primary Classification

RUN NO.	CLASSIFICATION ACCURACY	MEAN SQUARE DIFFERENCE BETWEEN S_a AND S_B
1	61.7%	0.00950
2	69.7%	0.00669
3	77.9%	0.00475
4	79.1%	0.00352

the secondary classification experiments were then compared with their classification results to determine classification accuracy. A document was considered to be correctly classified in a secondary class if this class matched that assigned by either of the two individuals. Otherwise it was considered to be incorrectly classified. Classification accuracy was then computed as follows.

Let

D_1 = number of documents classified by the algorithm

D_2 = number of documents correctly classified.

Then

$$\text{Accuracy} = \frac{D_2}{D_1} \times 100 .$$

The results are shown in Table 7.5.

Table 7.5 Accuracy Based on Manual Classification

S_T	TOTAL NUMBER OF DOCUMENTS CLASSIFIED	NUMBER OF DOCUMENTS CORRECTLY CLASSIFIED	ACCURACY
0	211	98	46.4%
0.2	163	101	62.0%
0.5	117	74	63.2%
0.8	73	48	65.8%

These results corroborate the conjectures of Experiment 1 fairly well. It is seen that for the values of S_T = 0.2, 0.5 and 0.8, the classification accuracy is better than 60%. For S_T = 0, the accuracy is substantially lower as had been predicted by Experiment 1.

The results of the above experiments show that the Bayesian distance technique can indeed be used effectively for secondary classification as well as for primary classification.

CHAPTER VIII

CONCLUSIONS AND FUTURE RESEARCH

This research has concentrated on the design of an efficient method for automatic sequential classification of documents. It has been pointed out that one of the main advantages of such a technique is that a document need not be examined in its entirety before a decision regarding its class membership can be made.

The basis of the research has been a sequential classification algorithm developed by Fried and implemented for various data bases by White and coworkers. In this technique keywords are extracted sequentially from a document and at each stage a statistical prediction technique is used to determine whether or not the document can be classified. If not, then the document is examined further. The process is continued until a definite decision can be reached. It has been shown that this basic sequential technique is vulnerable to the occurrence of noisy keywords and is not sophisticated enough to assign a document to more than one class systematically. Therefore the major part of this research has dealt with the development of a modified sequential algorithm which is able to isolate noisy keywords during classification and to identify clusters of similar keywords. These keyword clusters are then analyzed separately to obtain primary and secondary classes for a document.

The basis of the modified sequential technique is a t-dimensional vector space representation of the keywords contained in a document. It has been shown that using such a representation the relationship between the keywords can be observed very systematically by defining a distance measure on this vector space. This distance measure, called the Bayesian distance, consists of two components, a magnitude and direction. This research has shown, both experimentally and mathematically, that when a series of keywords, all of which are indicative of a unique class, is processed, the magnitude of the Bayesian distance increases monotonically and the direction remains constant. If, however, a noisy keyword occurs, the magnitude decreases or the direction changes. This interesting phenomenon has been utilized to effectively separate the keywords contained in a document into two groups--the good keywords and the noisy keywords. The good keywords, all of which are in general indicative of a unique class, are then analyzed to identify a primary class for the document. But how about the group of noisy keywords?

This research has shown that the noisy keywords, which were identified as being noisy with respect to the primary class, can be utilized to explore the possibility of assigning a secondary class to the document. The noisy keywords are analyzed using the Bayesian distance measure to ascertain whether they relate the document to an additional class. If this is so then this additional class is identified as the secondary class.

Results obtained by applying this automatic sequential algorithm on a portion of the SPIN data base have shown that the technique can be quite successful in identifying primary and secondary classes for a document. An accuracy of about 80% has been obtained for primary classification. For secondary classification the accuracy has been better than 60%. Experimental evidence obtained by using class correlation techniques and manual methods of classification has corroborated this result. Considering the fact that the SPIN data base contains only abstracts most of which have a very limited number of keywords, these results appear to be quite encouraging. It is expected that if full documents had been used instead of abstracts, then the secondary classifier would be able to examine a substantially greater number of keywords before assigning secondary classes, and hence would probably yield better results.

Several related areas of research can be identified at this stage. In the area of automatic document classification, the problem of keyword selection is of paramount importance. Manual and semi-automatic methods have generally been used to select keywords for a given data base. It would be very advantageous if there were an automatic method that could identify bad or inappropriate keywords in a set which is initially chosen manually or semi-automatically. We have shown that the Bayesian distance measure is an effective device to detect the occurrence of noisy keywords in a document during classification. Some of these noisy keywords are eventually used to obtain a secondary class. However, those that do not give any information about either a primary class or a secondary class can be isolated as bad keywords and eliminated from the original list of keywords. Hopefully, this would result in a keyword set which is more representative of the data base.

Another interesting problem would be to study the effect of the order in which the keywords are processed to obtain primary and secondary classification. It has been shown in this research that if all the keywords extracted from a document are indicative of a unique class, then the order in which they are processed does not matter. However, if noisy keywords are present in a document, then in some cases the order of processing the keywords may have detrimental effects on the final decision. This aspect needs further development and better mathematical characterization.

Finally, a third problem, which is much broader in scope, can be identified. It was pointed out in the introduction that the main purpose of automatic document classification is to aid the process of information retrieval from a data base. How can the Bayesian distance technique be utilized for this purpose? If the user queries could also be processed by means of this technique to obtain an indication of the various subject areas which might be relevant, then the classification and retrieval aspects of an information system could be integrated. Since, however, queries are much shorter in length than documents or abstracts, the Bayesian distance technique would probably have to be modified considerably in order that the queries may be analyzed effectively to achieve desirable retrieval performance.

APPENDIX A

BAYESIAN DISTANCE AND CLASSIFICATION ERROR

In Chapter IV it was shown that for a two class problem the magnitude of the Bayesian distance forms a tight upper bound on the classification error. In this appendix it will be shown that as the number of classes increases, the quality of the Bayesian distance as an approximating function to the classification error does not degrade very much.

Theorem:

For an t-class problem after an observation y, i.e., after i keywords have been read, the upper bound on classification error P_E, given by $1 - \text{Mag}(i)$, does not exceed the value of P_E by more than $\frac{t-1}{4t}$, i.e., if $I = \{1, 2, \ldots, t\}$ then

$$[1 - \text{Mag}(i)] - P_E \le \frac{t-1}{4t}$$

or

$$[\max_{r \varepsilon I}\{p(C_r/y)\}] - \sum_{s \varepsilon I} [p(C_s/y)]^2 \le \frac{t-1}{4t}$$

Proof:

Let

$$[\max_{r \varepsilon I}\{p(C_r/y)\}] = p$$

If the inequality is satisfied for the minimum value of $\sum_{r\epsilon I} [p(C_r/y)]^2$ then the theorem is proven. Without loss of generality let us assume

$$p(C_1/y) = p.$$

Then the minimum of $\sum_{r\epsilon I} [p(C_r/y)]^2$ is achieved when the remaining a posteriori probabilities are equal, i.e., when $p(C_r/y) = \frac{1-p}{t-1}$, $r = 2, 3, \ldots, t$.

Therefore,

$$\max_{r\epsilon I}\{p(C_r/y)\} - \sum_{s\epsilon I} [p(C_s/y)]^2$$

$$= p - [p^2 + (t-1) \cdot (\frac{1-p}{t-1})^2]$$

$$= p - p^2 - \frac{1}{t-1} + \frac{2}{t-1} \cdot p - \frac{1}{t-1} \cdot p^2$$

$$= Q$$

Q achieves its maximum for some p; therefore taking the derivative of Q with respect to p and setting it equal to zero, we have

$$\frac{dQ}{dp} = 1 - 2p + \frac{2}{t-1} - \frac{2}{t-1} \cdot p = 0$$

or

$$p = \frac{t+1}{2t}$$

Substituting this in Q we have

$$Q = \frac{t-1}{4t}.$$

Since Q is equal to this in the worst case, in general therefore

$$Q \leq \frac{t-1}{4t} \qquad \text{Q.E.D.}$$

Essentially, this theorem says that as the number of classes increases, the upper bound approaches a constant value. In terms of document classification, this theorem stipulates that the higher the final value of the Bayesian distance at the time of classification, the more probable it is that the document will be correctly classified.

APPENDIX B

BAYESIAN DISTANCE AND α_j VALUES

In Chapter IV it was claimed that if we have a set of α_j values

$$[\alpha_1, \alpha_2, \ldots, \alpha_t]$$

such that

$$\alpha_1 \geq \sum_{i=2}^{t} \alpha_i \ ,$$

and

$$\alpha_i = \alpha_{i+1}, \ i = 2 \text{ to } t-1;$$

then an increase in α_1 will increase the magnitude of the Bayesian distance. In this appendix this claim will be validated.

Let M_1 be the D_B magnitude obtained from $[\alpha_1, \alpha_2, \ldots, \alpha_t]$. Then

$$M_1 = \sum_{i=1}^{t} (\alpha_i)^2 \qquad \text{(B.1)}$$

Since $\alpha_i = \alpha_{i+1}$, $i = 2$ to $t-1$, we can write equation (B.1) as

$$M_1 = (\alpha_1)^2 + \frac{(1-\alpha_1)^2}{t-1}$$

Let

$$a = (\alpha_1)^2$$

$$b = \frac{(1-\alpha_1)^2}{t-1}$$

then

$$M_1 = a + b \qquad (B.2)$$

Now, if α_1 is increased by a quantity δ we have the new set of α_j values

$$[\alpha_1', \alpha_2', \ldots, \alpha_t'] ,$$

where

$$\alpha_1' = \alpha_1 + \delta .$$

The value of the new D_B magnitude, M_2, calculated by using this set of α_j values will be a minimum if the α_2' through α_t' are all equal. Then the minimum value of M_2 is given by

$$M_2 = (\alpha_1 + \delta)^2 + \frac{(1-\alpha_1-\delta)^2}{t-1}$$

$$= a' + b'$$

Therefore,

$$M_2 - M_1 = (a'-a) + (b'-b)$$

$$= 2\delta[\alpha_1 - \frac{(1-\alpha_1)}{t-1}] + 2\delta^2$$

Since

$$\alpha_1 \geq \sum_{i=2}^{t} \alpha_i ,$$

it is true that

$$\alpha_1 \geq (1 - \alpha_1) .$$

Therefore

$$M_2 - M_1 > 0 .$$

Since the minimum value of M_2 has been shown to be greater than M_1 the claim has been validated.

APPENDIX C

BAYESIAN DISTANCE AND KEYWORD VECTORS

In this appendix, Theorem 5.2 stated and discussed in Chapter V will be proven.

Theorem 5.2: Given two keywords

k_i: (p_1, p_2, p_3) where $p_1 > p_2 \geq p_3$

k_j: (q_1, q_2, q_3) where $q_1 > q_2 \geq q_3$

let M_1 be the magnitude of the Bayesian distance calculated using k_i, and M_2 be the magnitude calculated using k_i and k_j. Then $M_1 \geq M_2$, i.e.,

$$\frac{\sum_{r=1}^{3} p_r^2 q_r^2}{\left[\sum_{r=1}^{3} p_r q_r\right]^2} \geq \frac{\sum_{r=1}^{3} p_r^2}{\left[\sum_{r=1}^{3} p_r\right]^2} \tag{C.1}$$

Proof: Expanding the left hand side of the inequality (C.1) we have

$$F = \frac{p_1^2 q_1^2 + p_2^2 q_2^2 + p_3^2 q_3^2}{[p_1 q_1 + p_2 q_2 + p_3 q_3]^2} \tag{C.2}$$

Assume that the values of p_1, p_2 and p_3 are fixed. If it can be shown that the minimum value of F obtained by varying q_1, q_2 and q_3 under the given constraints is greater than or equal to the right hand side of (C.1), then the inequality is proven. Taking the partial derivatives of F with respect to q_1, q_2 and q_3, simplifying and setting to zero we obtain:

$$\frac{\partial F}{\partial q_1} = p_2q_2(p_1q_1 - p_2q_2) + p_3q_3(p_1q_1 - p_3q_3) = 0 \qquad (C.3)$$

$$\frac{\partial F}{\partial q_2} = p_1q_1(p_2q_2 - p_1q_1) + p_3q_3(p_2q_2 - p_3q_3) = 0 \qquad (C.4)$$

$$\frac{\partial F}{\partial q_3} = p_1q_1(p_3q_3 - p_1q_1) + p_2q_2(p_3q_3 - p_2q_2) = 0 \qquad (C.5)$$

From equations (C.3), (C.4) and (C.5) we see that a minimum occurs when $p_1q_1 = p_2q_2 = p_3q_3$, but that contradicts the assumptions that

$$p_1q_1 > p_2q_2 \text{ and } p_1q_1 > p_3q_3$$

Therefore we have to check if there are any other minimum points of F.

Solving equation (C.4) for p_2q_2 we have

$$p_2q_2 = \frac{p_1^2q_1^2 + p_3^2q_3^2}{p_1q_1 + p_3q_3} \qquad (C.6)$$

Substituting that in equation (C.5) we have

$$\begin{aligned} &(p_1q_1)(p_3q_3)(p_1^2q_1^2 + 2p_1q_1p_3q_3 + p_3^2q_3^2) \\ &- (p_1^2q_1^2)(p_1^2q_1^2 + 2p_1q_1p_3q_3 + p_3^2q_3^2) \\ &+ (p_3q_3)(p_1^2q_1^2 + p_3^2q_3^2)(p_1q_1 + p_3q_3) \\ &- (p_1^4q_1^4 + 2p_1^2q_1^2p_3^2q_3^2 + p_3^4q_3^4) = 0 \qquad (C.7) \end{aligned}$$

Let $x = p_1q_1$, $y = p_3q_3$; then equation (C.7) reduces to

$$x^3y + yx^3 - y^4 - x^4 = 0 \qquad \text{(C.8)}$$

$y = x$ is a solution to equation (C.8), but this implies $p_1q_1 = p_3q_3$. Since this is not possible let us identify the other roots of equation (C.8). Dividing it by $(y - x)$ we obtain

$$y^3 + xy^2 + x^2y - x^3 = 0 \qquad \text{(C.9)}$$

From Descartes' rule of signs it is seen that equation (C.9) has no more than one positive real root and no more than two negative real roots.

Substituting $y = z - \frac{x}{3}$ in equation (C.9) we obtain a reduced cubic in z.

$$(z - \frac{x}{3})^3 + x(z - \frac{x}{3})^2 + x^2(z - \frac{x}{3}) - x^3 = 0 \qquad \text{(C.10)}$$

Equation (C.10) simplifies to

$$z^3 + \frac{2zx^2}{3} - \frac{34}{27}x^3 = 0 \qquad \text{(C.11)}$$

This is of the form

$$y^3 + ay + b = 0$$

and can be solved by Cardan's method.

Substitute $z = u + v$. z is a root if

$$u^3 + v^3 = -b \qquad \text{(C.12)}$$

and

$$uv = \frac{-a}{3} \qquad \text{(C.13)}$$

From equation (C.13) we have

$$v = -\frac{a}{3u}$$

Substituting this in equation (C.12) we have

$$(u^3)^2 + b(u^3) - \frac{a^3}{27} = 0 \qquad \text{(C.14)}$$

If u_1 is a cube root of $\frac{1}{2}\{-b + (b^2 - \frac{4a^3}{27})^{\frac{1}{2}}\}$ then the three roots of equation (C.11) are

$$z_1 = u_1 + v_1$$

$$z_2 = wu_1 + w^2 v_1$$

$$z_3 = w^2 u_1 + wv_1$$

$$\text{where} \quad v_1 = -\frac{a}{3u_1} \quad \text{and} \quad w = -\frac{1}{2} + \frac{1}{2}(3i)^{\frac{1}{2}}$$

This is of the form

$$z = -\{\frac{b}{2} + (R)^{\frac{1}{2}}\}^{\frac{1}{3}} + \{-\frac{b}{2} - (R)^{\frac{1}{2}}\}^{\frac{1}{3}}$$

where

$$R = (\frac{b}{2})^2 + \frac{a^3}{27}$$

If these roots are real and distinct then R is less than zero.

For our problem

$$a = \frac{2x^3}{3} \quad \text{and} \quad b = -(\frac{34}{27})x^3$$

Therefore

$$R = (\frac{34}{54})^2 x^6 + \frac{1}{27}(\frac{8}{27})x^6 > 0$$

From this we have

$$R = \frac{29.7x^3}{27}$$

Therefore the positive real root is given by

$$z = [\frac{17}{27}x^3 + \frac{17.234}{27}x^3]^{\frac{1}{3}} + [\frac{17}{27}x^3 - \frac{17.234}{27}x^3]^{\frac{1}{3}}$$

Therefore

$$z = \frac{2.631\,x^3}{3}$$

Since $y = z - \frac{x}{3}$ we have

$$y = 0.54369\,x$$

i.e., $$p_3q_3 = (0.54369)p_1q_1 \qquad (C.15)$$

From equation (C.6) we obtain

$$p_2q_2 = (0.83929)p_1q_1 \qquad (C.16)$$

The minimum value of F is therefore given by

$$F_{min} = \frac{(p_1q_1)^2}{(p_1q_1)^2} \cdot \frac{1 + (0.54369)^2 + (0.83929)^2}{[1 + 0.54369 + 0.83929]^2}$$

which reduces to

$$F_{min} = 0.35220.$$

This is independent of (p_1q_1).

From equations (C.15) and (C.16) we obtain

$$q_2 = (0.83929)\left(\frac{p_1}{p_2}\right)q_1 \qquad \text{(C.17)}$$

$$q_3 = (0.54369)\left(\frac{p_1}{p_3}\right)q_1 \qquad \text{(C.18)}$$

The choice of p_1 and q_1 can be arbitrary, but the constraints require that

$$q_2 < q_1$$

and

$$q_3 < q_1$$

which imply (from equations (C.17) and (C.18))

$$p_2 > 0.83929\, p_1 \qquad \text{(C.19)}$$

$$p_3 > 0.54369\, p_1 \qquad \text{(C.20)}$$

At this point it is appropriate to recapitulate the original problem.

$$M_1 = \frac{p_1^2 + p_2^2 + p_3^2}{(p_1 + p_2 + p_3)^2} \qquad \text{(C.21)}$$

$$M_2 = \frac{p_1^2q_1^2 + p_2^2q_2^2 + p_3^2q_3^2}{(p_1q_1 + p_2q_2 + p_3q_3)^2} \qquad \text{(C.22)}$$

We form the function G given by

$$G = [M_1 - M_2] \qquad \text{(C.23)}$$

G will be maximized under several inequality constraints discussed later in the section. In order to obtain the extreme points of G, these will be replaced by equalities in each case. If it can be shown that the maximum value of G is less than or equal to zero then the inequality (C.1) will have been proven. M_1 is independent of q_1, q_2 and q_3, so M_2 has to be minimized with respect to q_1, q_2 and q_3. It has been shown that this is achieved when equations (C.17) and (C.18) are satisfied. But conditions $q_2 < q_1$ and $q_3 < q_1$ require extra constraints. These constraints are given by inequalities (C.19) and (C.20). Therefore under these constraints the maximum value of M_1 is obtained when

$$p_2 = 0.83929\, p_1 \qquad (C.24)$$

$$p_3 = 0.54369\, p_1 \qquad (C.25)$$

Thus the maximum value of G when the constraints given by inequalities (C.19) and (C.20) are satisfied is

$$\begin{aligned} G &= [M_1 - M_2] \\ &= [0.35220 - 0.35220] \\ &= 0 \end{aligned}$$

Therefore inequality (C.1) holds, i.e., the theorem is proven under these constraints.

Now, it has to be shown that when constraints (C.19) and (C.20) do not hold, inequality (C.1) is still valid. Three cases have to be considered.

(I) Constraint $p_2 > 0.83929\, p_1$ is not satisfied, but $p_3 > 0.54369\, p_1$ is satisfied.

(II) Constraint $p_3 > 0.54369\, p_1$ is not satisfied, but $p_2 > 0.83929\, p_1$ is satisfied.

(III) Neither of the two constraints is satisfied.

Case (I)

In this case in order to minimize M_2, q_2 can be made as large as possible keeping in mind that

$$q_2 < q_1$$

has to be satisfied. Therefore choose $q_2 = q_1$; so only q_3 can vary. It has been shown (see equation (C.5))

$$\frac{\partial F}{\partial q_3} = 0$$

implies

$$p_1 p_3 q_1 q_3 + p_2 p_3 q_2 q_3 - p_1^2 q_1^2 - p_2^2 q_2^2 = 0 \qquad (C.26)$$

Since $q_2 = q_1$ from equation (C.26) we have

$$q_3 = \frac{(p_1^2 + p_2^2)}{(p_1 + p_2)} \cdot \frac{q_1}{p_3}$$

If the constraint $q_3 < q_1$ is to be satisfied then

$$\frac{(p_1^2 + p_2^2)}{(p_1 + p_2)} \cdot \frac{1}{p_3} < 1 \qquad (C.27)$$

This means

$$p_1^2 + p_2^2 < p_1 p_3 + p_2 p_3$$

Since $p_1 > p_2 \geq p_3$, inequality (C.28) is never satisfied. Therefore we must choose $q_3 = q_2 = q_1$ in order to minimize M_2. Hence

$$M_2 = M_1$$

and

$$G = 0.$$

Therefore the theorem is true in this case.

Case (II)

Following the same arguments given in Case (I) we choose

$$q_3 = q_1 .$$

Then is F is minimized with respect to q_2 under the constraint $q_2 < q_1$ we obtain

$$\frac{(p_1^2 + p_3^2)}{(p_1 + p_3)} \cdot \frac{1}{p_2} < 1$$

$$\text{i.e.,} \quad p_1(p_1 - p_2) < p_3(p_2 - p_3) \qquad \text{(C.29)}$$

Since $p_1 > p_2 \geq p_3$, it is possible that inequality (C.29) may be satisfied. Thus two subcases may be identified.

Case II (a)

If $\frac{(p_1^2 + p_3^2)}{(p_1 + p_3)} \cdot \frac{1}{p_2} < 1$, then to minimize F choose

$$q_3 = q_1$$

and

$$q_2 = \frac{(q_1^2 + p_3^2)}{(p_1 + p_3)} \cdot \frac{q_1}{p_2} .$$

Then M_2 is given by

$$M_2 = \frac{p_1^2 + \dfrac{(p_1^2 + p_3^2)^2}{(p_1 + p_3)^2} + p_3^2}{[p_1 + \dfrac{p_1^2 + p_2^2}{p_1 + p_3} + p_3]^2} .$$

We recall that

$$M_1 = \frac{p_1^2 + p_2^2 + p_3^2}{(p_1 + p_2 + p_3)^2} .$$

Since

$$\frac{p_1^2 + p_3^2}{p_1 + p_3} < p_2 , \quad M_2 < M_1$$

Therefore the theorem is also valid in this case.

Case II (b)

If $\dfrac{(p_1^2 + p_3^2)}{(p_1 + p_3)} \cdot \dfrac{1}{p_2} > 1$

then we have to choose $q_1 = q_2 = q_3$ so that F may be minimum. Then we have

$$M_2 = M_1$$

and

$$G = 0.$$

Therefore the theorem holds in this case.

Case (III)

Since neither constraints (C.19) or (C.20) is satisfied q_2 and q_3 can be made as large as possible under the constraint

$$q_1 > q_2 \geq q_3$$

Therefore for minimum F we have

$$q_1 = q_2 = q_3 .$$

Hence

$$-M_2 = M_1$$

and

$$G = 0 .$$

Therefore the theorem has been shown to hold for all cases.

APPENDIX D

THE CLASSIFICATION ALGORITHM

The classification algorithm consists of the keyword extractor module, the primary classifier, the Bayesian distance calculator and the noise detector. In this appendix we will outline each of these modules so that this algorithm may be implemented. The following entities need to be discussed for better understanding of the algorithm.

Q and S are parameters discussed in Chapter V;

S_G and S_N are sets containing the indices of the words in the GBUF and the NBUF respectively;

S_T is a set of indices of keywords for which the Bayesian distances have to be calculated; during any call to the Bayesian distance calculator module, it is equal to either S_G or S_N;

Mag[1:n] and Dir[1:n] are arrays of magnitudes and directions calculated by the Bayesian distance calculator.

Keyword Extractor

This module reads one keyword at a time and stores the following in an input buffer:

a) a value i corresponding to the index of the keyword read;

b) the keyword; and

c) the probability values associated with the keywords. These are stored in the matrix P(i,j), where i is the index of the keyword and j = 1 to t correspond to the t categories.

Primary Classifier

This module analyzes the words in the GBUF to obtain a primary class. The input consists of values for Q and S and the output is a numeric code corresponding to the SPIN category which has been assigned as the primary class.

```
1. $S_G = \emptyset$; $S_N = \emptyset$;
2. i = 1;
3. call keyword extractor; i = i + 1; if i < Q then go
   step 2;
4. $S_G = S_G \cup \{1, 2, \ldots, Q\}$;
5. load indices 1 through Q in GBUF;
6. $S_T = S_G$; $D = |S_T|$; call Bayesian Distance Calculator:
7. for j = 1 to D - 1, check to see whether Mag(j)<Mag(j+1)
   and Dir(j)=Dir(j+1). If not then call Noise Detector;
   else go to step 15;
8. j = index of noisy word; put j in the NBUF;
   $S_G = S_G - \{j\}$; $S_N = S_N \cup \{j\}$; A = last element in $S_G$;
```

```
    B = last element in $S_N$;
9.  if $|S_G| = |S_N|$ then $S_T = S_G$; call Bayesian Distance Calculator; else go to step 6;
10. $X = Mag(A)$;
11. $S_T = S_N$; call Bayesian Distance Calculator;
12. $Y = Mag(b)$; if $Y > X$ then interchange the elements of the GBUF and the NBUF and the elements of $S_G$ and $S_N$ respectively;
13. call Keyword Extractor; if document has no more keywords then identify it as unclassifiable;
14. load index of new keyword in the GBUF;
    $S_G = S_G \cup \{new\ index\}$; go to step 6;
15. if $Mag(D) \geq S$ then primary class = Dir(D); else go to step 13;
16. stop.
```

Bayesian Distance Calculator

Given a set S_T of indices of keywords, this module calculates the magnitudes and directions of D_B, and stores them in the arrays, Mag[1:n] and Dir[1:n] respectively. The input to this module is a matrix P[n,t] containing the probability values associated with the keywords whose indices are in S_T.

```
1.  $N = |S_T|$;
2.  $i = 1$;
```

```
3. if $i = 1$ then for $j = 1$ to $t$, set
      $PROB(i,j) = P(i,j)$; else for $j = 1$ to $t$, set
      $PROB(i,j) = PROB(i-1,j) * P(i,j)$;
4. $PSTAR = \sum_{j=1}^{t} PROB(i,j)$;
5. for $j = 1$ to $t$ compute
      $AL(j) = \frac{PROB(i,j)}{PSTAR}$;
6. $Mag(i) = \sum_{j=1}^{t} AL^2(j)$;
7. $Dir(i)$ = index of the largest value of $AL(j)$;
8. $i = j + 1$; if $i > N$ then return; else go to step 3.
```

Noise Detector

This module compares the magnitudes and directions of the keywords in the GBUF and detects a noisy word. It returns the index of this word.

```
1. $N = |S_T|$;
2. for $i = 1$ to $N - 1$ check whether $Mag(i) > Mag(i + 1)$.
      If so then return $(i + 1)$ as the index of the noisy
      keyword;
3. for $i = 1$ to $N - 1$, check whether $Dir(i) = Dir(i + 1)$.
      If not then return $(i + 1)$ as the index of the noisy
      keyword.
```

BIBLIOGRAPHY

1. Bonner, R. E., "On Some Clustering Techniques," IBM Journal of Research and Development, Vol. 8, No. 1, January 1964, pp. 22-32.

2. Borko, H., "Research in Document Classification and File Organization," Report No. SP - 1423, System Development Corporation, Santa Monica, Calif., 1963.

3. Borko, H. and Bernick, M., "Automatic Document Classification," Journal of the ACM, Vol. 10, No. 2, April 1963, pp. 151-162.

4. Cleverdon, C., "The Cranfield Tests on Index Language Devices," ASLIB Proc. Vol. 19, No. 6, June 1967.

5. Dattola, R. T., "A Fast Algorithm for Automatic Classification," Information Storage and Retrieval, Scientific Report No. ISR - 14, Cornell U., Ithaca, New York, October 1968.

6. Day, A. C., "Full Table Quadratic Searching for Scatter Storage," Communications of the ACM, Vol. 13, No. 8, August 1970, pp. 481-482.

7. Devijver, P. A., "On a New Class of Bounds on Bayes Risk in Multihypothesis Pattern Recognition," IEEE Transactions on Computers, Vol. C-23, No. 1, January 1974, pp. 70-80.

8. Doyle, L. B., "Is Automatic Classification a Reasonable Application of Statistical Analysis of Text?" Journal of the ACM, Vol. 12, No. 4, October 1965, pp. 473-489.

9. Doyle, L. B., "Breaking the Cost Barrier in Automatic Classification," Report No. SP - 2516, System Development Corporation, Santa Monica, Calif., July 1966.

10. Edmundson, H. P. and Wyllys, R. E., "Automatic Abstracting and Indexing - Survey and Recommendations," Communications of the ACM, Vol. 4, No. 5, May 1961, pp. 226-234.

11. Fried, J. B., et. al., "Index Simulation Feasibility and Automatic Document Classification," Technical Report No. 68-4, Computer and Information Science Research Center, The Ohio State University, October 1968.

12. Fu, K. S., Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968.

13. Heaps, H. S., "A Theory of Relevance for Automatic Document Classification," Information and Control, Vol. 22, No. 3, April 1973, pp. 268-278.

14. Hillman, D. J., "Mathematical Classification Techniques for Non-Static Document Collections with Particular Refer nce to the Problem of Relevance," International Study Conference on Classification Research, P. Atherton (ed.), Munksgaard, Copenhagen, 1965.

15. Jackson, D. M., "Classification, Relevance, and Information Retrieval," Advances in Computers, Vol. 11, Academic Press, New York, 1971.

16. Jardine, N. and Sibson, R., Mathematical Taxonomy, John Wiley and Sons, Ltd., London, 1971.

17. Johnson, S. C., "Hierarchical Clustering Schemes," Psychometrica, Vol. 32, No. 3, September 1967, pp. 241-254.

18. Luhn, H. P., "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," IBM Journal of Research and Development, Vol. 1, No. 4, October 1957, pp. 309-317.

19. Maron, M. E., "Automatic Indexing: An Experimental Inquiry," Journal of the ACM, Vol. 8, No. 3, July 1961, pp. 404-417.

20. Maron, M. E. and Kuhns, J. L., "On Relevance, Probabilistic Indexing and Information Retrieval," Journal of the ACM, Vol. 7, No. 3, July 1960, pp. 216-244.

21. Rocchio, J. J., "Document Retrieval Systems - Optimization and Evaluation," Information Storage and Retrieval, Scientific Report No. ISR - 10, Cornell University, Ithaca, New York, 1966.

22. Salton, G., Automatic Information Organization and Retrieval, McGraw Hill, New York, 1968.

23. Salton, G., The SMART Retrieval System - Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, N.J., 1971.

24. Sokal, R. R. and Sneath, P. H. A., Principles of Numerical Taxonomy, W. H. Freeman, San Francisco, 1963.

25. Tanimoto, T., "An Elementary Mathematical Theory of Classification and Prediction," IBM Corporation, 1958.

26. Van Rigsbergen, C. J., "An Algorithm for Information Structuring Retrieval," Computer Journal, Vol. 14, No. 4, 1971, pp. 407-412.

27. Wald, A., *Sequential Analysis*, John Wiley and Sons, New York, 1947.

28. White, L. J., et al., "A Sequential Method for Automatic Document Classification," Technical Report No. CISRC 75-5, The Ohio State University, September 1975.

29. Williams, J. H., "A Discriminant Method for Automatically Classifying Documents," Proceedings of the Fall Joint Computer Conference, Vol. 24, 1963, pp. 161-166.

30. Williams, J. H., "Computer Classification of Documents," *Mechanized Information Storage, Retrieval and Dissemination*, North Holland, 1968, pp. 235-245.

31. Yu, C. T., "A Clustering Algorithm Based on User Queries," Journal of the ASIS, Vol. 25, No. 4, July - August 1974, pp. 218-226.

32. Yu, C. T., "Theory of Indexing and Classification," Report No. TR 73-181, Department of Computer Science, Cornell University, August 1973.