Three
AUTOMATIC CLASSIFICATION

Introduction

In this chapter I shall attempt to present a coherent account of classification in such a way that the principles involved will be sufficiently understood for anyone wishing to use classification techniques in IR to do so without too much difficulty. The emphasis will be on their application in document clustering, although many of the ideas are applicable to pattern recognition, automatic medical diagnosis, and keyword clustering.

A formal definition of classification will not be attempted; for our purposes it is sufficient to think of classification as describing the process by which a classificatory system is constructed. The word 'classification' is also used to describe the result of such a process. Although indexing is often thought of (wrongly I think) as 'classification' we specifically exclude this meaning. A further distinction to be made is between 'classification' and 'diagnosis'. Everyday language is very ambiguous on this point:

'How would you classify (identify) this?'

'How are these best classified (grouped)?'

The first example refers to diagnosis whereas the second talks about classification proper. These distinctions have been made before in the literature by Kendall[1] and Jardine and Sibson[2].

In the context of information retrieval, a classification is required for a purpose. Here I follow Macnaughton-Smith[3] who states: 'All classifications, even the most general are carried out for some more or less explicit "special purpose" or set of purposes which should influence the choice of [classification] method and the results obtained.' The purpose may be to group the documents in such a way that retrieval will be faster or alternatively it may be to construct a thesaurus automatically. Whatever the purpose the 'goodness' of the classification can finally only be measured by its performance during retrieval. In this way we can side-step the debate about 'natural' and 'best' classifications and leave it to the philosophers (see for example Hempel[4]).

There are two main areas of application of classification methods in IR:

(1) keyword clustering;

(2) document clustering.

The first area is very well dealt with in a recent book by Sparck Jones[5]. Document clustering, although recommended forcibly by Salton and his co-workers, has had very little impact. One possible reason is that the details of Salton's work on document clustering became submerged under the welter of experiments performed on the SMART system. Another is possibly that as the early enthusiasm for clustering wanted, the realisation dawned that significant experiments in this area required quantities of expensive data and large amounts of computer time.

Good[6] and Fairthorne[7] were amongst the first to recommend that automatic classification might prove useful in document retrieval. A clear statement of what is implied by document clustering was made early on by R. M. Hayes[8]: 'We define the organisation as the grouping together of items (e.g. documents, representations of documents) which are then handled as a unit and lose, to that extent, their individual identities. In other words, classification of a document into a classification slot, to all intents and purposes identifies the document with that slot. Thereafter, it and other documents in the slot are treated as

identical until they are examined individually. It would appear, therefore, that documents are grouped because they are in some sense related to each other; but more basically, they are grouped because they are likely to be *wanted* together, and logical relationship is the means of measuring this likelihood.' In the main, people have achieved the 'logical organisation' in two different ways. Firstly, through direct classification of the documents, and secondly via the intermediate calculation of a measure of closeness between documents. The first approach has proved theoretically to be intractable so that any experimental test results cannot be considered to be reliable. The second approach to classification is fairly well documented now, and above all, there are some forceful arguments recommending it in a particular form. It is this approach which is to be emphasised here.

The efficiency of document clustering has been emphasised by Salton[9], he says: 'Clearly in practice it is not possible to match each analysed document with each analysed search request because the time consumed by such operation would be excessive. Various solutions have been proposed to reduce the number of needed comparisons between information items and requests. A particular promising one generates groups of related documents, using an automatic document matching procedure. A representative document *group vector* is then chosen for each document group, and a search request is initially checked against all the group vectors only. Thereafter, the request is checked against only those individual documents where group vectors show a high score with the request.' Salton believes that although document clustering saves time, it necessarily reduces the effectiveness of a retrieval system. I believe a case has been made showing that on the contrary document clustering has *potential* for improving the effectiveness (Jardine and van Rijsbergen[10]).

Measures of association

Some classification methods are based on a binary relationship between objects. On the basis of this relationship a classification method can construct a system of clusters. The relationship is described variously as 'similarity', 'association' and 'dissimilarity'. Ignoring dissimilarity for the moment as it will be defined mathematically later, the other two terms mean much the same except that 'association' will be reserved for the similarity between objects characterised by discrete-state attributes. The measure of similarity is designed to quantify the likeness between objects so that if one assumes it is possible to group objects in such a way that an object in a group is more like the other members of the group than it is like any object outside the group, then a cluster method enables such a group structure to be discovered.

Informally speaking, a measure of association increases as the number or proportion of shared attribute states increases. Numerous coefficients of association have been described in the literature, see for example Goodman and Kruskal[11, 12], Kuhns[13], Cormack[14] and Sneath and Sokal[15]. Several authors have pointed out that the difference in retrieval performance achieved by different measures of association is insignificant, providing that these are appropriately normalised. Intuitively, one would expect this since most measures incorporate the same information. Lerman[16] has investigated the mathematical relationship between many of the measures and has shown that many are monotone with respect to each other. It follows that a cluster method depending only on the rank-ordering of the association values would given identical clusterings for all these measures.

There are five commonly used measures of association in information retrieval. Since in information retrieval documents and requests are most commonly represented by term or keyword lists, I shall simplify matters by assuming that an object is represented by a set of keywords and that the counting measure $|\ .\ |$ gives the size of the set. We can easily generalise to the case where the keywords have been weighted, by simply choosing an appropriate measure (in the measure-theoretic sense).

The simplest of all association measures is

$|X \cap Y|$ *Simple matching coefficient*

which is the number of shared index terms. This coefficient does not take into account the sizes of X and Y. The following coefficients which have been used in document retrieval take into account the information provided by the sizes of X and Y.

$$2\,\frac{|X \cap Y|}{|X| + |Y|}$$ *Dice's coefficient*

$$\frac{|X \cap Y|}{|X \quad Y|}$$ *Jaccard's coefficient*

$$\frac{|X \cap Y|}{|X|^{1/2} \times |Y|^{1/2}}$$ *Cosine coefficient*

$$\frac{|X \cap Y|}{\min(|X|, |Y|)}$$ *Overlap coefficient*

These may all be considered to be normalised versions of the simple matching coefficient. Failure to normalise leads to counter intuitive results as the following example shows:

If $S_1(X, Y) = |X \cap Y|$ $S_2(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}$

then $|X_1| = 1$ $|Y_1| = 1$ $|X_1 \cap Y_2| = 1$ $S_1 = 1 S_2 = 1$

$|X_2| = 10$ $|Y_2| = 10$ $|X_2 \cap Y_2| = 1$ $S_1 = 1 S_2 = 1 / 10$

$S_1(X_1, Y_1) = S_1(X_2, Y_2)$ which is clearly absurd since X_1 and Y_1 are identical representatives whereas X_2 and Y_2 are radically different. The normalisation for S_2, scales it between) and 1, maximum similarity being indicated by 1.

Doyle[17] hinted at the importance of normalisation in an amusing way: 'One would regard the postulate "All documents are created equal" as being a reasonable foundation for a library description. Therefore one would like to count either documents or things which pertain to documents, such as index tags, being careful of course to deal with the same number of index tags for each document. Obviously, if one decides to describe the library by counting the word tokens of the text as "of equal interest" one will find that documents contribute to the description in proportion to their size, and the postulate "Big documents are more important than little documents" is at odds with "All documents are created equal" '.

I now return to the promised mathematical definition of dissimilarity. The reasons for preferring the 'dissimilarity' point of view are mainly technical and will not be elaborated here. Interested readers can consult Jardine and Sibson[2] on the subject, only note that any dissimilarity function can be transformed into a similarity function by a simple transformation of the form $s = (1 + d)^{-1}$ but the reverse is not always true.

If P is the set of objects to be clustered, a pairwise dissimilarity coefficient D is a function from $P \times P$ to the non-negative real numbers. D, in general, satisfies the following conditions:

$D1$ $D(X, Y) \geq 0$ for all X, Y P

$D2\ \ D(X, X) = 0$ for all $X \quad P$

$D3\ \ D(X, Y) = D(Y, X)$ for all $X, Y \quad P$

Informally, a dissimilarity coefficient is a kind of 'distance' function. In fact, many of the dissimilarity coefficients satisfy the triangle inequality:

$D4\ \ D(X, Y) \leq D(X, Z) + D(Y, Z)$

which may be recognised as the theorem from Euclidean geometry which states that the sum of the lengths of two sides of a triangle is always greater than the length of the third side.

An example of a dissimilarity coefficient satisfying $D1$ - $D4$ is

$$\frac{|X \ \Delta \ Y|}{|X| \ + \ |Y|}$$

where $(X \ \Delta \ Y) = (X \quad Y) - (X \cap Y)$ is the symmetric different of sets X and Y. It is simply related to Dice's coefficient by

$$1 - \frac{2|X \cap Y|}{|X| + |Y|} = \frac{|X \ \Delta \ Y|}{|X| + |Y|}$$

and is monotone with respect to Jaccard's coefficient subtracted from 1. To complete the picture, I shall express this last DC in a different form. Instead of representing each document by a set of keywords, we represent it by a binary string where the absence or presence of the ith keyword is indicated by a zero or one in the ith position respectively. In that case

$$\frac{\sum x_i \ (1 - y_i) \ + \ \sum y_i \ (1 - x_i)}{\sum x_i \ + \ \sum y_i}$$

where summation is over the total number of different keywords in the document collection.

Salton considered document representatives as binary vectors embedded in an n-dimensional Euclidean space, where n is the total number of index terms.

$$\frac{|X \ \cap \ Y|}{|X|^{1/2} \ \ |Y|^{1/2}}$$

can then be interpreted as the cosine of the angular separation of the two binary vectors X and Y. This readily generalises to the case where X and Y are arbitrary real vectors (i.e. weighted keyword lists) in which case we write

$$\frac{(X, \ Y)}{||X|| \ \ ||Y||}$$

where (X, Y) is the inner product and $||\ .\ ||$ the length of a vector. If the space is Euclidean then for

$X = (x_1, ..., x_n)$ and $Y = (y_1, ..., y_n)$

we get

$$\frac{\sum_{i=1}^{n} x_i y_i}{\left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \left(\sum_{i=1}^{n} y_i^2\right)^{1/2}}$$

Some authors have attempted to base a measure of association on a probabilistic model[18]. They measure the association between two objects by the extent to which their distributions deviate from stochastic independence. This way of measuring association will be of particular importance when in Chapter 5 I discuss how the association between index terms is to be used to improve retrieval effectiveness. There I use the *expected mutual information measure* to measure association. For two discrete probability distributions $P(x_i)$ and $P(x_j)$ it can be defined as follows:

$$I(x_i, x_j) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)\,P(x_j)}$$

When x_i and x_j are independent $P(x_i)P(x_j) = P(x_i,x_j)$ and so $I(x_i,x_j) = 0$. Also $I(x_ix_j) = 0$. Also $I(x_ix_j) = I(x_jx_i)$ which shows that it is symmetric. It also has the nice property of being invariant under one-to-one transformations of the co-ordinates. Other interesting properties of this measure may be found in Osteyee and Good[19]. Rajski[20] shows how $I(x_ix_j)$ may be simply transformed into a distance function on discrete probability distributions. $I(x_ix_j)$ is often interpreted as a measure of the statistical information contained in x_i about x_j (or vice versa). When we apply this function to measure the association between two index terms, say i and j, then x_i and x_j are binary variables. Thus $P(x_i = 1)$ will be the probability of occurrence of the term i and similarly $P(x_i = 0)$ will be the probability of its non-occurrence. The extent to which two index terms i and j are associated is then measured by $I(x_ix_j)$ which measures the extent to which their distributions deviate from stochastic independence.

A function very similar to the expected mutual information measure was suggested by Jardine and Sibson[2] specifically to measure dissimilarity between two *classes* of objects. For example, we may be able to discriminate two classes on the basis of their probability distributions over a simple two-point space {1, 0}. Thus let $P_1(1)$, $P_1(0)$ and $P_2(1)$, $P_2(0)$ be the probability distributions associated with class I and II respectively. Now on the basis of the difference between them we measure the dissimilarity between I and II by what Jardine and Sibson call the *Information Radius*, which is

$$uP_1(1) \log \frac{P_1(1)}{uP_1(1) + vP_2(1)} + vP_2(1) \log \frac{P_2(1)}{uP_1(1) + vP_2(1)} +$$

$$uP_1(0) \log \frac{P_1(0)}{uP_1(0) + vP_2(0)} + vP_2(0) \log \frac{P_1(0)}{uP_1(0) + vP_2(0)}$$

Here u and v are positive weights adding to unit. This function is readly generalised to multi-state, or indeed continuous distribution. It is also easy to shown that under some interpretation the expected mutual information measure is a special case of the information radius. This fact will be of some importance in Chapter 6. To see it we write $P_1(.)$ and $P_2(.)$ as two conditional distributions $P(./w_1)$ and $P(./w_2)$. If we now interpret $u = P(./w_1)$ and $v = P(./w_2)$, that is the prior probability of the conditioning variable in $P(./w_i)$, then on substituting into the expression for the information radius and using the identities.

$$P(x) = P(x/w_1)\,P(w_1) + P(x/w_2)\,P(w_2) \qquad x = 0, 1$$

$$P(x/w_i) = P(x/w_i)\,P(x) \qquad i = 1, 2$$

we recover the expected mutual information measure $I(x,w_i)$.

Classification methods

Let me start with a description of the kind of data for which classification methods are appropriate. The data consists of *objects* and their corresponding descriptions. The objects may be documents, keywords, hand written characters, or species (in the last case the objects themselves are classes as opposed to individuals). The descriptors come under various names depending on their structure:

(1) multi-state attributes (e.g. colour)

(2) binary-state (e.g. keywords)

(3) numerical (e.g. hardness scale, or weighted keywords)

(4) probability distributions.

The fourth category of descriptors is applicable when the objects are classes. For example, the leaf width of a species of plants may be described by a normal distribution of a certain mean and variance. It is in an attempt to summarise and simplify this kind of data that classification methods are used.

Some excellent surveys of classification methods now exist, to name but a few, Ball[21], Cormack[14] and Dorofeyuk[22]. In fact, methods of classification are now so numerous, that Good[23] has found it necessary to give a classification of classification.

Sparck Jones[24] has provided a very clear intuitive break down of classification methods in terms of some general characteristics of the resulting classificatory system. In what follows the primitive notion of 'property' will mean feature of an object. I quote:

(1) Relation between properties and classes

  (a) monothetic

  (b) polythetic

(2) Relation between objects and classes

  (a) exclusive

  (b) overlapping

(3) Relation between classes and classes

  (a) ordered

  (b) unordered

The first category has been explored thoroughly by numerical taxonomists. An early statement of the distinction between monothetic and polythetic is given by Beckner[25]: 'A class is ordinarily defined by reference to a set of properties which are both necessary and sufficient (by stipulation) for membership in the class. It is possible, however, to define a group K in terms of a set G of properties $f_1, f_2, \ldots, f_n$ in a different manner. Suppose we have an aggregate of individuals (we shall not yet call them a class) such that

(1) each one possesses a large (but unspecified) number of the properties in G;

(2) each f in G is possessed by large number of these individuals; and

(3) no *f* in *G* is possessed by every individual in the aggregate.'

The first sentence of Beckner's statement refers to the classical Aristotelian definition of a class, which is now termed *monothetic*. The second part defines polythetic.

To illustrate the basic distinction consider the following example (*Figure 3.1*) of 8 individuals (1-8) and 8 properties (A-H). The possession of a property is indicated by a plus sign. The individuals 1-4 constitute a polythetic group each individual possessing three out of four of the properties A,B,C,D. The other 4 individuals can be split into two monothetic classes {5,6} and {7,8}. The distinction between monothetic and polythetic is a particularly easy one to make providing the properties are of a simple kind, e.g. binary-state attributes. When the properties are more complex the definitions are rather more difficult to apply, and in any case are rather arbitrary.

The distinction between overlapping and exclusive is important both from a theoretical and practical point of view. Many classification methods can be viewed as data-simplification methods. In the process of classification information is discarded so that the members of one class are indistinguishable. It is in an attempt to minimise the amount of information thrown away, or to put it differently, to have a classification which is in some sense 'closest' to the original data, that overlapping classes are allowed.

	A	B	C	D	E	F	G	H
1	+	+	+					
2	+	+		+				
3	+		+	+				
4		+	+	+				
5					+	+	+	
6					+	+	+	
7					+	+		+
8					+	+		+

Figure 3.1. An illustration of the difference between monothetic and polythetic

Unfortunately this plays havoc with the efficiency of implementation for a particular application. A compromise can be adopted in which the classification methods generates overlapping classes in the first instance and is finally 'tidied up' to give exclusive classes.

An example of an ordered classification is a hierarchy. The classes are ordered by inclusion, e.g. the classes at one level are nested in the classes at the next level. To give a simple example of unordered classification is more difficult. Unordered classes generally crop up in automatic thesaurus construction. The classes sought for a thesaurus are those which satisfy certain homogeneity and isolation conditions but in general cannot be simply

related to each other. (See for example the use and definition of clumps in Needham[26].) For certain applications ordering is irrelevant, whereas for others such as document clustering it is of vital importance. The ordering enables efficient search strategies to be devised.

The discussion about classification has been purposely vague up to this point. Although the break down scheme discussed gives some insight into classification method. Like all categorisations it isolates some ideal types; but any particular instance will often fall between categories or be a member of a large proportion of categories.

Let me know be more specific about current (and past) approaches to classification, particularly in the context of information retrieval.

The cluster hypothesis

Before describing the battery of classification methods that are now used in information retrieval, I should like to discuss the underlying hypothesis for their use in document clustering. This hypothesis may be simply stated as follows: *closely associated documents tend to be relevant to the same requests.* I shall refer to this hypothesis as the *Cluster Hypothesis.*

A basic assumption in retrieval systems is that documents relevant to a request are separated from those which are not relevant, i.e. that the relevant documents are more like one another than they are like non-relevant documents. Whether this is true for a collection can be tested as follows. Compute the association between all pairs of documents:

(a) both of which are relevant to a request, and

(b) one of which is relevant and the other non-relevant.

Summing over a set of requests gives the relative distribution of relevant-relevant (R-R) and relevant-non-relevant (R-N-R) associations of a collection. Plotting the relative frequency against strength of association for two hypothetical collections X and Y we might get distributions as shown in *Figure 3.2.*

From these it is apparent:

(a) that the separation for collection X is good while for Y it is poor; and

(b) that the strength of the association between relevant documents is greater for X than for Y.

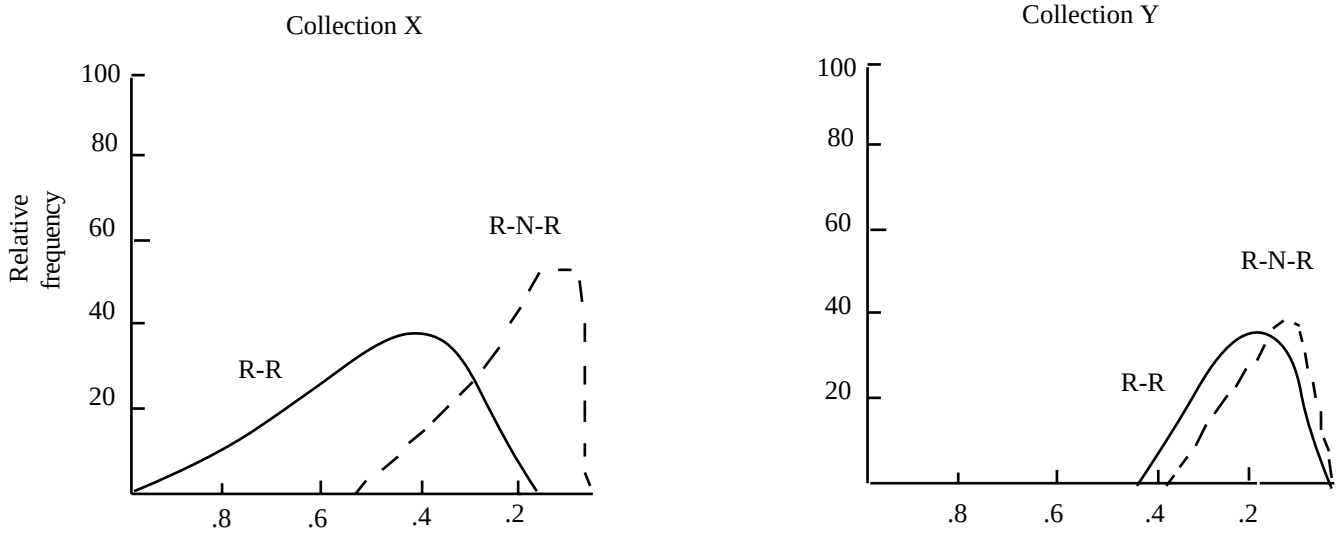

Figure 3.2. R-R is the distribution of relevant-relevant associations, and R-N-R is the distribution of relevant-non-relevant associations.

It is this separation between the distributions that one attempts to exploit in document clustering. It is on the basis of this separation that I would claim that document clustering can lead to more effective retrieval than say a linear search. A linear search ignores the relationship that exists between documents. If the hypothesis is satisfied for a particular collection (some promising results have been published in Jardine and van Rijsbergen[10], and van Rijsbergen and Sparck Jones[27] for three test collections), then it is clear that structuring the collection in such a way that the closely associated documents appear in one class, will not only speed up the retrieval but may also make it more effective, since a class once found will tend to contain only relevant and no non-relevant documents.

I should add that these conclusions can only be verified, finally, by experimental work on a large number of collections. One reason for this is that although it may be possible to structure a document collection so that relevant documents are brought together there is no guarantee that a search strategy will infallibly find the class of documents containing the relevant documents. It is a matter for experimentation whether one can design search strategies which will do the job. So far most experiments in document clustering have been moderately successful but by no means conclusive.

Note that the Cluster Hypothesis refers to given document descriptions. The object of making permanent or temporary changes to a description by such techniques as keyword classifications can therefore be expressed as an attempt to increase the distance between the two distributions R-R and R-N-R. That is, we want to make it more likely that we will retrieve relevant documents and less likely that we will retrieve non-relevant ones.

As can be seen from the above, the Cluster Hypothesis is a convenient way of expressing the aim of such operations as document clustering. Of course, it does not say anything about how the separation is to be exploited.

The use of clustering in information retrieval

There are a number of discussions in print now which cover the use of clustering in IR. The most important of these are by Litofsky[28], Crouch[29], Prywes and Smith[30] and Fritzche[31]. Rather than repeat their chronological treatment here, I shall instead try to isolate the essential features of the various cluster methods.

In choosing a cluster method for use in experimental IR, two, often conflicting, criteria have frequently been used. The first of these, and in my view the most important at this stage of the development of the subject, is the *theoretical soundness* of the method. By this I mean that the method should satisfy certain criteria of adequacy. To list some of the more important of these:

(1) the method produces a clustering which is unlikely to be altered drastically when further objects are incorporated, i.e. it is stable under growth.

(2) the method is stable in the sense that small errors in the description of the objects lead to small changes in the clustering;

(3) the method is independent of the initial ordering of the objects.

These conditions have been adapted from Jardine and Sibson[2]. The point is that any cluster method which does not satisfy these conditions is unlikely to produce any meaningful experimental results. Unfortunately not many cluster methods do satisfy these criteria, probably because algorithms implementing them tend to be less efficient than *ad hoc* clustering algorithms.

The second criterion for choice is the *efficiency* of the clustering process in terms of speed and storage requirements. In some experimental work this has been the overriding consideration. But it seems to me a little early in the day to insist on efficiency even before we know much about the behaviour of clustered files in terms of the effectiveness of retrieval (i.e. the ability to retrieve wanted and hold back unwanted documents.) In any case, many

of the 'good' theoretical methods (ones which are likely to produce meaningful experimental results) can be modified to increase the efficiency of their clustering process.

Efficiency is really a property of the algorithm implementing the cluster method. It is sometimes useful to distinguish the cluster method from its algorithm, but in the context of IR this distinction becomes slightly less than useful since many cluster methods are *defined* by their algorithm, so no explicit mathematical formulation exists.

In the main, two distinct approaches to clustering can be identified:

(1) the clustering is based on a measure of similarity between the objects to be clustered;

(2) the cluster method proceeds directly from the object descriptions.

The most obvious examples of the first approach are the *graph theoretic* methods which define clusters in terms of a graph derived from the measure of similarity. This approach is best explained with an example (see *Figure 3.3*). Consider a set of objects to be clustered. We compute a numerical value for each pair of objects indicating their similarity. A graph corresponding to this set of similarity values is obtained as follows: A threshold value is decided upon, and two objects are considered linked if their similarity value is above the threshold. The cluster definition is simply made in terms of the graphical representation.

A *string* is a connected sequence of objects from some starting point.

A *connected component* is a set of objects such that each object is connected to at least one other member of the set and the set is maximal with respect to this property.

A *maximal complete subgraph* is a subgraph such that each node is connected to every other node in the subgraph and the set is maximal with respect to this property, i.e. if one

further

Objects: {1,2,3,4,5,6}

Similarity matrix

1						
2	.6					
3	.6	.8				
4	.9	.7	.7			
5	.9	.6	.6	.9		
6	.5	.5	.5	.9	.5	
	1	2	3	4	5	6

Threshold: .89

Graph:

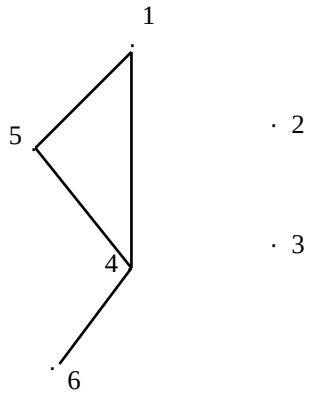

Figure 3.3. A similarity coefficient for 6 objects and the graph that can be derived from it by thresholding.

node were included anywhere the completeness condition would be violated. An example of each is given in *Figure 3.4.* These methods have been used extensively in keyword clustering by Sparck Jones and Jackson[32], Augustson and Minker[33] and Vaswani and Cameron[34].

A large class of *hierarchic* cluster methods is based on the initial measurement of similarity. The most important of these is *single-link* which is the only one to have extensively used in document retrieval. It satisfies all the criteria of adequacy mentioned above. In fact, Jardine and Sibson[2] have shown that under a certain number of reasonable conditions single-link is the only hierarchic method satisfying these important criteria. It will be discussed in some detail in the next section.

A further class of cluster methods based on measurement of similarity is the class of so-called 'clump' methods. They proceed by seeking sets which satisfy certain cohesion and isolation conditions defined in terms of the similarity measure. The computational difficulties of this approach have largely caused it to be abandoned. An attempt to generate a hierarchy of clumps was made by van Rijsbergen[35] but, as expected, the cluster definition was so strict that very few sets could be found to satisfy it.

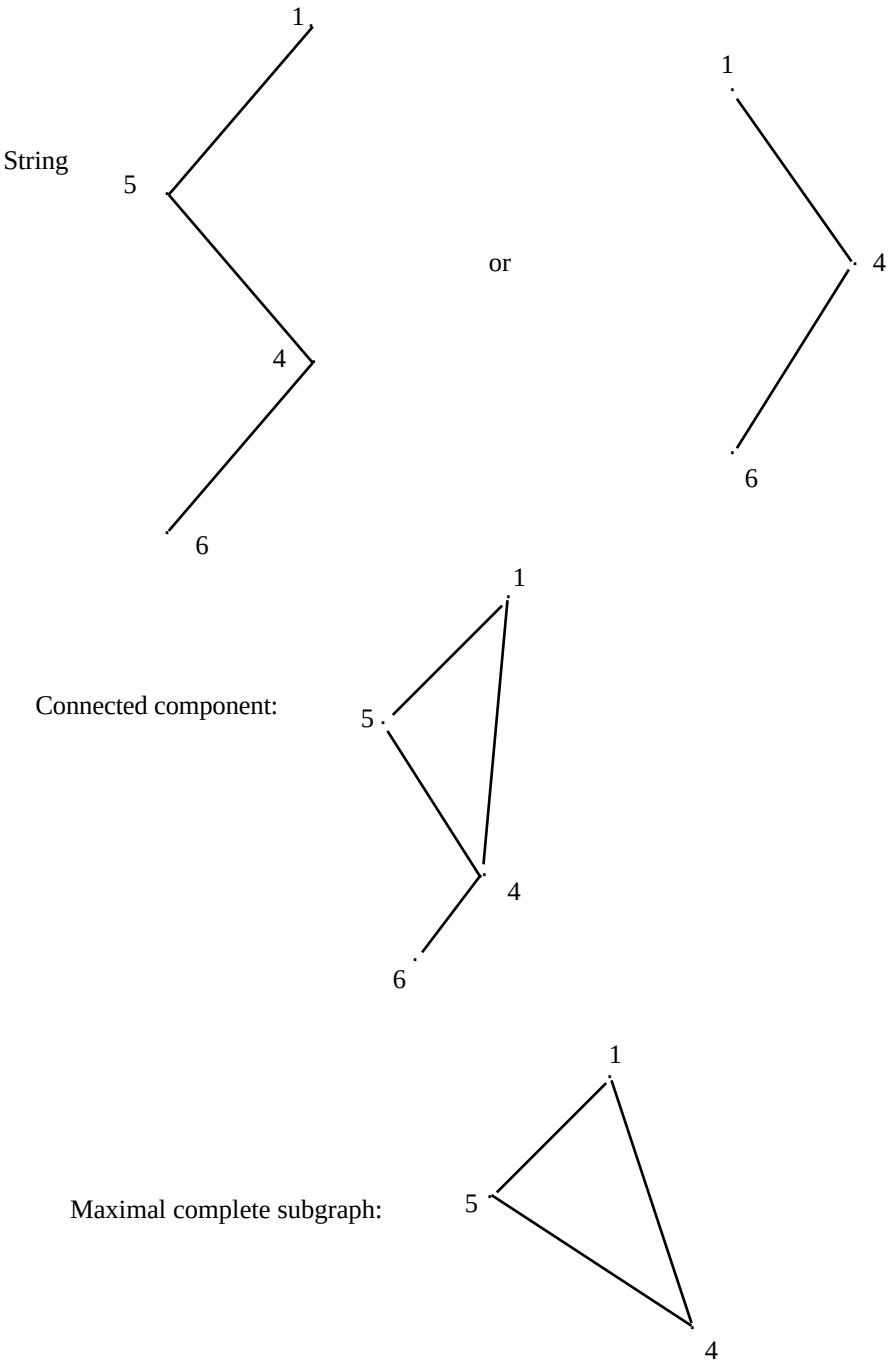

Figure 3.4. Some possible definitions of clusters in terms of subgraphs

Efficiency has been the overriding consideration in the definition of the algorithmically defined cluster methods used in IR. For this reason most of these methods have tended to proceed directly from object description to final classification without an intermediate calculation of a similarity measure. Another distinguishing characteristic of these methods is that they do not seek an underlying structure in the data but attempt to impose a suitable structure on it. This is achieved by restricting the number of clusters and by bounding the size of each cluster.

Rather than give a detailed account of all the heuristic algorithms, I shall instead discuss some of the main types and refer the reader to further developments by citing the appropriate authors. Before proceeding, we need to define some of the concepts used in designing these algorithms.

The most important concept is that of *cluster representative* variously called cluster profile, classification vector, or centroid. It is simply an object which summaries and represents the objects in the cluster. Ideally it should be near to every object in the cluster in some average sense; hence the use of the term centroid. The similarity of the objects to the representative is measured by a *matching function* (sometimes called similarity or correlation function). The algorithms also use a number of *empirically* determined parameters such as:

(1) the number of clusters desired;

(2) a minimum and maximum size for each cluster;

(3) a threshold value on the matching function, below which an object will not be included in a cluster;

(4) the control of overlap between clusters;

(5) an arbitrarily chosen objective function which is optimised.

Almost all of the algorithms are iterative, i.e. the final classification is achieved by iteratively improving an intermediate classification. Although most algorithms have been defined only for one-level classification, they can obviously be extended to multi-level classification by the simple device of considering the clusters at one level as the objects to be classified at the next level.

Probably the most important of this kind of algorithm is Rocchio's clustering algorithm[36] which was developed on the SMART project. It operates in three stages. In the first stage it selects (by some criterion) a number of objects as cluster centres. The remaining objects are then assigned to the centres or to a 'rag-bag' cluster (for the misfits). On the basis of the initial assignment the cluster representatives are computed and all objects are once more assigned to the clusters. The assignment rules are explicitly defined in terms of thresholds on a matching function. The final clusters may overlap (i.e. an object may be assigned to more than one cluster). The second stage is essentially an iterative step to allow the various input parameters to be adjusted so that the resulting classification meets the prior specification of such things as cluster size, etc. more nearly. The third stage is for 'tidying up'. Unassigned objects are forcibly assigned, and overlap between clusters is reduced.

Most of these algorithms aim at reducing the number of passes that have to be made of the file of object descriptions. There are a small number of clustering algorithms which only require one pass of the file of object descriptions. Hence the name 'Single-Pass Algorithm' for some of them. Basically they operate as follows:

(1) the object descriptions are processed serially;

(2) the first object becomes the cluster representative of the first cluster;

(3) each subsequent object is matched against all cluster representatives existing at its processing time;

(4) a given object is assigned to one cluster (or more if overlap is allowed) according to some condition on the matching function;

(5) when an object is assigned to a cluster the representative for that cluster is recomputed;

(6) if an object fails a certain test it becomes the cluster representative of a new cluster.

Once again the final classification is dependent on input parameters which can only be determined empirically (and which are likely to be different for different sets of objects) and must be specified in advance.

The simplest version of this kind of algorithm is probably one due to Hill[37]. Subsequently, many variations have been produced mainly the result of changes in the assignment rules and definition of cluster representatives. (See for example Rieber and Marathe[38], Johnson and Lafuente[39] and Etzweiler and Martin[40].)

Related to the single-pass approach is the algorithm of MacQueen[41] which starts with an arbitrary initial partition of the objects. Cluster representatives are computed for the members (sets) of the partition, and objects are reallocated to the nearest cluster representative.

A third type of algorithm is represented by the work of Dattola[42]. His algorithm is based on an earlier algorithm by Doyle. As in the case of MacQueen, it starts with an initial arbitrary partition and set of cluster representatives. The subsequent processing reallocates the objects, some ending up in a 'rag-bag' cluster (cf. Rocchio). After each reallocation the cluster representative is recomputed, but the *new* cluster representative will only replace the old one if the new representative turns out to be nearer in some sense to the objects in the new cluster than the old representative. Dattola's algorithm has been used extensively by Murray[43] for generating hierarchic classifications. Related to Dattola's approach is that due to Crouch[29]. Crouch spends more time obtaining the initial partition (he calls them categories) and the corresponding cluster representatives. The initial phase is termed the 'categorisation stage', which is followed by the 'classification stage'. The second stage proceeds to reallocate objects in the normal way. His work is of some interest because of the extensive comparisons he made between the algorithms of Rocchio, Rieber and Marathe, Bonner (see below) and his own.

One further algorithm that should be mentioned here is that due to Litofsky[28]. His algorithm is designed only to work for objects described by binary state attributes. It uses cluster representatives and matching functions in an entirely different way. The algorithm shuffles objects around in an attempt to minimise the average number of different attributes present in the members of each cluster. The clusters are characterised by sets of attribute values where each set is the set of attributes common to all members of the cluster. The final classification is a hierarchic one. (For further details about this approach see also Lefkovitz[44].)

Finally, the Bonner[45] algorithm should be mentioned. It is a hybrid of the graph-theoretic and heuristic approaches. The initial clusters are specified by graph-theoretic methods (based on an association measure), and then the objects are reallocated according to conditions on the matching function.

The major advantage of the algorithmically defined cluster methods is their speed: order $n \log n$ (where n is the number of objects to be clustered) compared with order n^2 for the methods based on association measures. However, they have disadvantages. The final classification depends on the order in which the objects are input to the cluster algorithm, i.e. it suffers from the defect of order dependence. In additional the effects of errors in the object descriptions are unpredictable.

One obvious omission from the list of cluster methods is the group of mathematically or statistically based methods such as Factor Analysis and Latest Class Analysis. Although both methods were originally used in IR (see Borko and Bernick[46], Baker[47]) they have now largely been superseded by the cluster methods described above.

The method of single-link avoids the disadvantages just mentioned. Its appropriateness for document clustering is discussed here.

Single-link

The dissimilarity coefficient is the basic input to a single-link clustering algorithm. The output is a hierarchy with associated numerical levels called a *dendrogram.* Frequently the hierarchy is represented by a tree structure such that each node represents a cluster. The two representations are shown side by side in *Figure 3.5* for the same set of objects {A,B,C,D,E}. The clusters are: {A,B}, {C}, {D}, {E} at level L_1, {A,B}, {C,D,E} at level L_2, and {A,B,C,D,E} at level L_3. At each level of the hierarchy one can identify a set of classes, and as one moves up the hierarchy the classes at the lower levels are nested in the classes at the higher levels. A mathematical definition of a dendrogram exists, but is of little use, so will be omitted. Interested readers should consult Jardine and Sibson[2].

To give the reader a better feel for a single-link classification, there is a worked example (see Figure 3.6). A DC (dissimilarity coefficient) can be characterised by a set of graphs, one for each value taken by the DC. The different values taken by the DC in the example are L = .1, .2, .3, .4. The graph at each level is given by a set of vertices corresponding to the objects to be clustered, and any two vertices are linked if their dissimilarity is at most equal to the value of the level L. It should be clear that these graphs characterise the DC completely. Given the graphs and their interpretation a DC can be recovered, and vice versa. Graphs at values other than those taken by the DC are simply the same as at the next smallest value actually taken by the DC, for example, compare the graphs at L = .15 and L = .1.

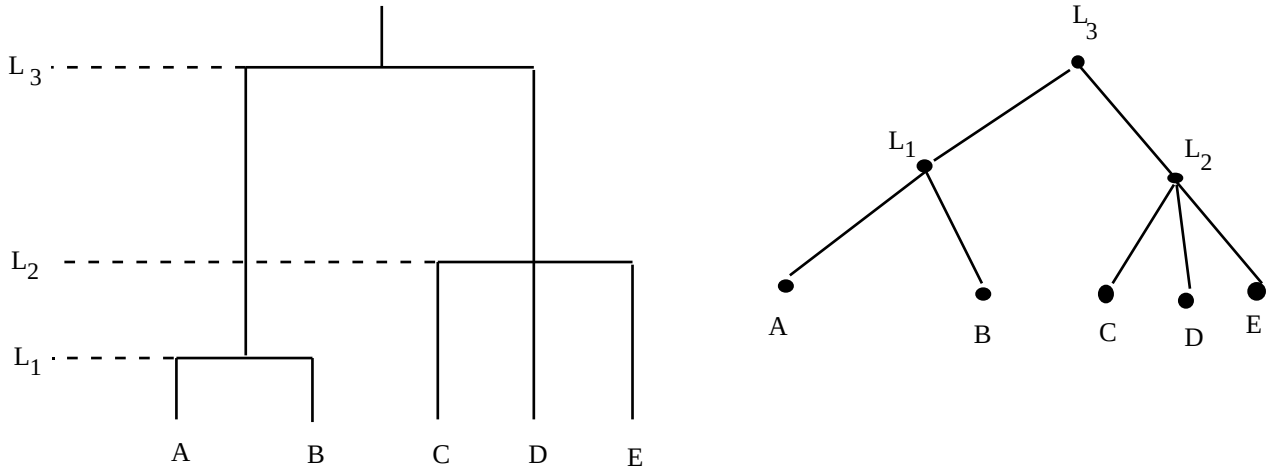

Figure 3.5. A dendrogram with corresponding tree

It is now a simple matter to define single-link in terms of these graphs; at any level a single-link cluster is precisely the set of vertices of a connected component of the graph at that level. In the diagram I have enclosed each cluster with a dotted line. Note that whereas the graphs at any two distinct values taken by the DC will be different, this is not necessarily the case for the

corresponding clusters at those levels. It may be that by increasing the level the links introduced between vertices do not change the total number of connected vertices in a component. For example, the clusters at levels .3 and .4 are the same. The hierarchy is achieved by varying the level from the lowest possible value, increasing it through successive values of the DC until all objects are contained in one cluster. The reason for the name single-link is now apparent: for an object to belong to a cluster it needs to be linked to only one other member of the cluster.

This description immediately leads to an *inefficient* algorithm for the generation of single-link classes. It was demonstrated in the example above. It simply consists of thresholding the DC at increasing levels of dissimilarity. The binary connection matrices are then

calculated at each threshold level, from which the connected components can easily be extracted. This is the basis for many published single-link algorithms. From the point of view of IR, where one is trying to construct a *searchable* tree it is too inefficient (see van Rijsbergen[48] for an appropriate implementation).

The appropriateness of stratified hierarchic cluster methods

There are many other hierarchic cluster methods, to name but a few: complete-link, average-link, etc. For a critique of these methods see Sibson[49]. My concern here is to indicate their appropriateness for document retrieval. It is as well to realise that the kind of retrieval intended is one in which the entire cluster is retrieved without any further subsequent processing of the documents in the cluster. This is in contrast with the methods proposed by Rocchio, Litofsky, and Crouch who use clustering purely to help limit the extent of a linear search.

Stratified systems of clusters are appropriate because the level of a cluster can be used in retrieval strategies as a parameter analogous to rank position or matching function threshold in a linear search. Retrieval of a cluster which is a good match for a request at a low level in the hierarchy tends to produce high precision but low recall* ; just as a cut-off at a low rank position in a linear search tends to yield high precision but low recall. Similarly, retrieval of a cluster which is a good match for a request at a high level in the hierarchy tends to produce high recall but low precision. *Hierarchic* systems of clusters are appropriate for three reasons. First, very efficient strategies can be devised to search a hierarchic clustering. Secondly, construction of a hierarchic systems is much faster than construction of a non-hierarchic (that is, stratified but overlapping) system of clusters. Thirdly, the storage requirements for a hierarchic structure are considerably less than for a non-hierarchic structure, particularly during the classification phase.

* See introduction for definition.

Dissimilarity matrix:

2	.4			
3	.4	.2		
4	.3	.3	.3	
5	.1	.4	.4	.1
	1	2	3	4

Binary matrices:

2	0			
3	0	0		
4	0	0	0	
5	1	0	0	1
	1	2	3	4

Threshold = .1

2	0			
3	0	1		
4	0	0	0	
5	1	0	0	1
	1	2	3	4

Threshold = .2

2	0			
3	0	1		
4	1	1	1	
5	1	0	0	
	1	2	3	

Threshold = .3

Graphs and clusters:

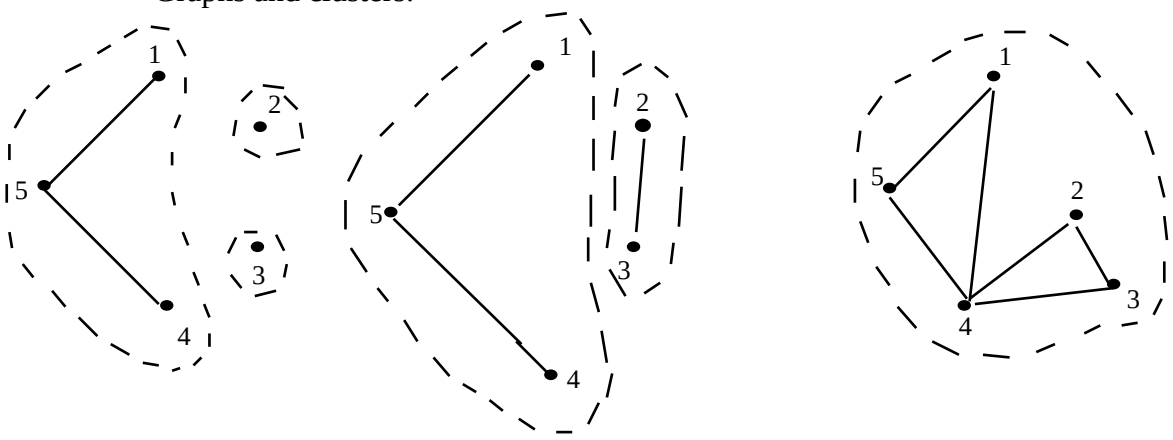

Figure 3.6. To show how single-link clusters may be derived from the dissimilarity coefficient by thresholding it.

Given that hierarchic methods are appropriate for document clustering the question arises: 'Which method?' The answer is that under certain conditions (made precise in Jardine and Sibson[2]) the only acceptable stratified hierarchic cluster method is single-link. Let me immediately qualify this by saying that it applies to a method which operates from a dissimilarity coefficient (or some equivalent variant), and does *not* take into account methods based directly on the object descriptions.

Single-link and the minimum spanning tree

The single-link tree (such as the one shown in *Figure 3.5*) is closely related to another kind of tree: the minimum spanning tree, or MST, also derived from a dissimilarity coefficient (Gower and Ross[50]). This second tree is quite different from the first, the nodes instead of representing clusters represent the individual objects to be clustered. The MST is the tree of minimum length connecting the objects, where by 'length' I mean the sum of the weights of the connecting links in the tree. Similarly we can define a maximum spanning tree as one of maximum length. Whether we are interested in a minimum or maximum spanning tree depends entirely on the application we have in mind. For convenience we will concentrate on the minimum spanning tree since it derives naturally from a dissimilarity coefficient and is more common anyway. (In Chapter 6 we shall have cause to use a maximum spanning tree based on the expected mutual information measure.) Given the minimum spanning tree then the single-link clusters are obtained by deleting links from the MST in order of decreasing length; the connected sets after each deletion are the single-link clusters. The order of deletion and the structure of the MST ensure that the clusters will be nested into a hierarchy.

The MST contains more information than the single-link hierarchy and only indirectly information about the single-link clusters. Thus, although we can derive the single-link hierarchy from it by a simple thresholding process, we cannot reverse this and uniquely derive the MST from the single-link hierarchy. It is interesting to consider in the light of this whether the MST would not be more suitable for document clustering than the single-link hierarchy. Unfortunately, it does not seem possible to update a spanning tree dynamically. To add a new object to a single-link hierarchy is relatively straightforward but to add one to an MST is much more complicated.

The representation of the single-link hierarchy through an MST has proved very useful in connecting single-link with other clustering techniques[51]. For example, Boulton and Wallace[52] have shown, using the MST representation, that under suitable assumptions the single-link hierarchy will minimise their information measure of classification. They see classification as a way of economically describing the original object descriptions, and the best classification is one which does it most economically in an information-theoretic sense. It is interesting that the MST has, independently of their work, been used to reduce storage when storing object descriptions, which amounts to a practical application of their result[53].

Implication of classification methods

It is fairly difficult to talk about the implementation of an automatic classification method without at the same time referring to the file structure representing it inside the computer. Nevertheless, there are a few remarks of importance which can be made.

Just as in many other computational problems, it is possible to trade core storage and computation time. In experimental IR, computation time is likely to be at a premium and a classification process can usually be speeded up by using extra storage.

One important decision to be made in any retrieval system concerns the organisation of storage. Usually part of the file structure will be kept in fast store and the rest on backing store. In experimental IR we are interested in a flexible system and getting experiments done quickly. Therefore, frequently much or all of a classification structure is kept in fast store although this would never be done in an operational system where the document collections are so much bigger.

Another good example of the difference in approach between experimental and operational implementations of a classification is in the permanence of the cluster representatives. In experiments we often want to vary the cluster representatives at search time. In fact, we require that each cluster representative can be quickly specified and implemented at search time. Of course, were we to design an operational classification, the cluster representatives would be constructed once and for all at cluster time.

Probably one of the most important features of a classification implementation is that it should be able to deal with a changing and growing document collection. Adding documents to the classification should not be too difficult. For instance, it should not be necessary to take the document classification 'off the air' for lengthy periods to update it. So, we expect the classification to be designed in such a way that a new batch of documents can be readily inserted without reclassifying the entire set of both old and new documents.

Although many classification algorithms claim this feature, the claim is almost invariably not met. Because of the heuristic nature of many of the algorithms, the updated classification is not the same as it would have been if the increased set had been classified from scratch. In addition, many of the updating strategies mess up the classification to such an extent that it becomes necessary to throw away the classification after a series of updates and reclassify completely.

These comments tend to apply to the $n \log n$ classification methods. Unfortunately, they are usually recommended over the n^2 methods for two reasons. Firstly, because $n \log n$ is considerably less than n^2, and secondly because the time increases only as $\log n$ for the $n \log n$ methods but as n for the n^2 methods. On the face of it these are powerful arguments. However, I think they mislead. If we assume that the $n \log n$ methods cannot be updated without reclassifying each time and that the n^2 methods can (for example, single-link), then the correct comparison is between

$$\sum_{i=1}^{t} n_i \log n_i \qquad \text{and} \qquad N^2$$

where $n_1 < n_2 < \ldots < n_t = N$, and t is the number of updates. In the limit when n is a continuous variable and the sum becomes an integral we are better off with N^2. In the discrete case the comparison depends rather on the size of the updates $n_i - n_{i-1}$. So unless we can design an $n \log n$ dependence as extra documents are added, we may as well stick with the n^2 methods which satisfy the soundness conditions and preserve n^2 dependence during updating.

In any case, if one is willing to forego some of the theoretical adequacy conditions then it is possible to modify the n^2 methods to 'break the n^2 barrier'. One method is to sample from the document collection and construct a *core clustering* using an n^2 method on the sample of the documents. The remainder of the documents can then be fitted into the core clustering by a very fast assignment strategy, similar to a search strategy which has $\log n$ dependence. A second method is to initially do a *coarse* clustering of the document collection and then apply the finer classification method of the n^2 kind to each cluster in turn. So, if there are N documents and we divide into k coarse clusters by a method that has order N time dependence (e.g. Rieber and Marathe's method) then the total cluster time will be of order $N + \sum (N/k)^2$ which will be less than N^2.

Another comment to be made about $n \log n$ methods is that although they have this time dependence in theory, examination of a number of the algorithms implementing them shows that they actually have an n^2 dependence (e.g. Rocchio's algorithm). Furthermore, most $n \log n$ methods have only been tested on single-level classifications and it is doubtful whether they would be able to preserve their $n \log n$ dependence if they were used to generate hierarchic classifications (Senko[54]).

In experiments where we are often dealing with only a few thousand documents, we may find that the proportionality constant in the $n \log n$ method is so large that the actual time taken for clustering is greater than that for an n^2 method. Croft[55] recently found this when he compared the efficiency of SNOB (Boulton and Wallace[56]), an $n \log n$ cluster method, with single-link. In fact, it is possible to implement single-link in such a way that the generation of the similarity values is overlapped in real time with the cluster generation process.

The implementation of classification algorithms for use in IR is by necessity different from implementations in other fields such as for example numerical taxonomy. The major differences arise from differences in the scale and in the use to which a classification structure is to be put.

In the case of scale, the size of the problem in IR is invariably such that for cluster methods based on similarity matrices it becomes impossible to store the entire similarity matrix, let alone allow random access to its elements. If we are to have a reasonably useful cluster method based on similarity matrices we must be able to generate the similarity matrix in small sections, use each section to update the classification structure immediately after it has been generated and then throw it away. The importance of this fact was recognised by Needham[57]. van Rijsbergen[48] has described an implementation of single-link which satisfies this requirement.

When a classification is to be used in IR, it affects the design of the algorithm to the extent that a classification will be represented by a file structure which is

(1) easily updated;

(2) easily searched; and

(3) reasonably compact.

Only (3) needs some further comment. It is inevitable that parts of the storage used to contain a classification will become redundant during an updating phase. This being so it is of some importance to be able to reuse this storage, and if the redundant storage becomes excessive to be able to process the file structure in such a way that it will subsequently reside in one contiguous part of core. This 'compactness' is particularly important during experiments in which the file structure is read into core before being accessed.

Conclusion

Let me briefly summarise the logical structure of this chapter. It started very generally with a descriptive look at automatic classification and its uses. It then discussed association measures which form the basis of an important class of classification methods. Next came a breakdown of classification methods. This was followed by a statement of the hypothesis underlying the use of automatic classification in document clustering. It went on to examine in some detail the use of classification methods in IR leading up to recommendation of single-link for document clustering. Finally we made some practical points about implementation.

This chapter ended on a rather practical note. We continue in this vein in the next chapter where we discuss file structures. These are important if we are to appreciate how it is that we can get dictionaries, document clustering, search strategies, and such like to work inside a computer.

Bibliographic remarks

In recent years a vast literature on automatic classification has been generated. One reason for this is that applications for these techniques have been found in such diverse fields as Biology, Pattern Recognition, and Information Retrieval. The best introduction to the field is still provided by Sneath and Sokal[15] (a much revised and supplemented version of their earlier book) which looks at automatic classification in the context of numerical taxonomy. Second to this, I would recommend a collection of papers edited by Cole[58].

A book and a report on cluster analysis with a computational emphasis are Anderberg[59] and Wishart[60] respectively. Both given listings of Fortran programs for various cluster methods. Other books with a numerical taxonomy emphasis are Everitt[61], Hartigan[62] and Clifford and Stephenson[63]. A recent book with a strong statistical flavour is Van Ryzin[64].

Two papers worth singling out are Sibson[65] and Fisher and Van Ness[66]. The first gives a very lucid account of the foundations of cluster methods based on dissimilarity measures. The second does a detailed comparison of some of the more well-known cluster methods (including single-link) in terms of such conditions on the clusters as connectivity and convexity.

Much of the early work in document clustering was done on the SMART project. An excellent idea of its achievement in this area may be got by reading ISR-10 (Rocchio[36]), ISR-19 (Kerchner[67]), ISR-20 (Murray[43]), and Dattola[68]. Each has been predominantly concerned with document clustering.

There are a number of areas in IR where automatic classification is used which have not been touched on in this chapter. Probably the most important of these is the use of 'Fuzzy Sets' which is an approach to clustering pioneered by Zadeh[69]. Its relationship with the measurement of similarity is explicated in Zadeh[70]. More recently it has been applied in document clustering by Negoita[71], Chan[72] and Radecki[73].

One further interesting area of application of clustering techniques is in the clustering of citation graphs. A measure of closeness is defined between *journals* as a function of the frequency with which they cite one another. Groups of closely related journals can thus be isolated (Disiss[74]). Related to this is the work of Preparata and Chien[75] who study citation patterns between *documents* so that mutually cited documents can be stored as closely together as possible. The early work of Ivie[76] was similarly motivated in that he proposed to collect feedback information from users showing which pairs of documents were frequently was then taken as proportional to the strength of association, and documents more closely associated were made more readily accessible than those less closely associated.

Finally, the reader may be interested in pursuing the use of cluster methods in pattern recognition since some of the ideas developed there are applicable to IR. Both Duda and Hart[77] and Watanabe[78] devote a chapter to clustering in the context of pattern recognition.

References

1. KENDALL, M.G., In Multivariate Analysis (Edited by P.R. Krishnaiah), Academic Press, London and New York, 165-184 (1966).

2. JARDINE, N. and SIBSON, R., Mathematical Taxonomy, Wiley, London and New York (1971).

3. MACNAUGHTON-SMITH, P., Some Statistical and Other Numerical Techniques for Classifying Individuals, Studies in the causes of delinquency and the treatment of offenders. Report No. 6, HMSO, London (1965).

4. HEMPEL, C.G., Aspects of Scientific Explanation and Other Essays in the Philosophy of Science, The Free Press, New York, 137-154 (1965).

5. SPARCK JONES, K., Automatic Keyword Classification for Information Retrieval, Butterworths, London (1971).

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

7. FAIRTHORNE, R.A., 'The mathematics of classification'. Towards Information Retrieval, Butterworths, London, 1-10 (1961).

8. HAYES, R.M., 'Mathematical models in information retrieval'. In Natural Language and the Computer (Edited by P.L. Garvin), McGraw-Hill, New York, 287 (1963).

9. SALTON, G., Automatic Information Organization and Retrieval, McGraw-Hill, New York, 18 (1968).

10. JARDINE, N. and van RIJSBERGEN, C.J., 'The use of hierarchic clustering in information retrieval', Information Storage and Retrieval, 7, 217-240 (1971).

11. GOODMAN, L. and KRUSKAL, W., 'Measures of association for cross-classifications', Journal of the American Statistical Association, 49, 732-764 (1954).

12. GOODMAN, L. and KRUSKAL, W., 'Measures of association for cross-classifications II: Further discussions and references', Journal of the American Statistical Association, 54, 123-163 (1959).

13. KUHNS, J.L., 'The continuum of coefficients of association'. In Statistical Association Methods for Mechanised Documentation, (Edited by Stevens et al.) National Bureau of Standards, Washington, 33-39 (1965).

14. CORMACK, R.M., 'A review of classification', Journal of the Royal Statistical Society, Series A, 134, 321-353 (1971).

15. SNEATH, P.H.A. and SOKAL, R.R., Numerical Taxonomy: The Principles and Practice of Numerical Classification, W.H. Freeman and Company, San Francisco (1973).

16. LERMAN, I.C., Les Bases de la Classification Automatique, Gauthier-Villars, Paris (1970).

17. DOYLE, L.B., 'The microstatistics of text'. Information Storage and Retrieval, 1, 189-214 (1963).

18. MARON, M.E. and KUHNS, J.L., 'On relevance, probabilistic indexing and information retrieval', Journal of the ACM, 7, 216-244 (1960).

19. OSTEYEE, D.B. and GOOD, I.J., Information, Weight of Evidence, the Singularity between Probability Measures and Signal Detection, Spring Verlag, Berlin (1974).

20. RAJSKI, C., 'A metric space of discrete probability distributions', Information and Control, 4, 371-377 (1961).

21. BALL, G.H., 'Data-analysis in the social sciences: What about the details?', Proceedings of the Fall Joint Computer Conference, 27, 533-559 (1966).

22. DOROFEYUK, A.A., 'Automatic Classification Algorithms (Review)', Automation and Remote Control, 32, 1928-1958 (1971).

23. GOOD, I.J., 'Categorization of classification' In Mathematics and Computer Science in Biology and Medicine, HMSO, London, 115-125 (1965).

24. SPARCK JONES, K., 'Some thoughts on classification for retrieval', Journal of Documentation, 26, 89-101 (1970).

25. BECKNER, M., The Biological Way of Thought, Columbia University Press, New York, 22 (1959).

26. NEEDHAM, R.M., 'The application of digital computers to classification and grouping', Ph.D. Thesis, University of Cambridge (1961).

27. van RIJSBERGEN, C.J. and SPARCK JONES, K., 'A test for the separation of relevant and non-relevant documents in experimental retrieval collections', Journal of Documentation, 29, 251-257 (1973).

28. LITOFSKY, B., 'Utility of automatic classification systems for information storage and retrieval', Ph.D. Thesis, University of Pennsylvania (1969).

29. CROUCH, D., 'A clustering algorithm for large and dynamic document collections', Ph.D. Thesis, Southern Methodist University (1972).

30. PRYWES, N.S. and SMITH, D.P., 'Organization of Information', Annual Review of Information Science and Technology, 7, 103-158 (1972).

31. FRITZCHE, M., 'Automatic clustering techniques in information retrieval' Diplomarbeit, Institut für Informatik der Universität Stuttgart (1973).

32. SPARCK JONES, K. and JACKSON, D.M., 'The use of automatically-obtained keyword classifications for information retrieval', Information Storage and Retrieval, 5, 175-201 (1970).

33. AUGUSTSON, J.G. and MINKER, J., 'An analysis of some graph-theoretic cluster techniques', Journal of the ACM, 17, 571-588 (1970).

34. VASWANI, P.K.T. and CAMERON, J.B., The National Physical Laboratory Experiments in Statistical Word Associations and their use in Document Indexing and Retrieval, Publication 42, National Physical Laboratory, Division of Computer Science (1970).

35. van RIJSBEGEN, C.J., 'A clustering algorithm', Computer Journal, 13, 113-115 (1970).

36. ROCCHIO, J.J., 'Document retrieval systems - optimization and evaluation', Ph.D. Thesis, Harvard University, Report ISR-10 to National Science Foundation, Harvard Computation Laboratory (1966).

37. HILL, D.R., 'A vector clustering technique', In Mechanised Information Storage, Retrieval and Dissemination, (Edited by Samuelson), North-Holland, Amsterdam (1968).

38. RIEBER, S. and MARATHE, U.P., 'The single pass clustering method', In Report ISR-16 to the National Science Foundation, Cornell University, Department of Computer Science (1969).

39. JOHNSON, D.B. and LAFUENTE, J.M., 'A controlled single pass classification algorithm with application to multilevel clustering', In Report ISR-18 to the National Science Foundation and the National Library of Medicine (1970).

40. ETZWEILER, L. and MARTIN, C., 'Binary cluster division and its application to a modified single pass clustering algorithm', In Report No. ISR-21 to the National Library of Medicine (1972).

41. MacQUEEN, J., 'Some methods for classification and analysis of multivariate observations', In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1965, University of California Press, 281-297 (1967).

42. DATTOLA, R.T., 'A fast algorithm for automatic classification', In Report ISR-14 to the National Science Foundation, Section V, Cornell University, Department of Computer Science (1968).

43. MURRAY, D.M. 'Document retrieval based on clustered files', Ph.D. Thesis, Cornell University, Report ISR-20 to the National Science Foundation and to the National Library of Medicine (1972).

44. LEFKOVITZ, D., File Structures for On-line Systems, Spartan Books, New York (1969).

45. BONNER, R.E., 'On some clustering techniques', IBM Journal of Research and Development, 8, 22-32 (1964).

46. BORKO, H. and BERNICK, M., 'Automatic document classification', Journal of the ACM, 10, 151-162 (1963).

47. BAKER, F.B., 'Information retrieval based upon latest class analysis', Journal of the ACM, 9, 512-521 (1962).

48. van RIJSBERGEN, C.J., 'An algorithm for information structuring and retrieval', The Computer Journal, 14, 407-412 (1971).

49. SIBSON, R., 'Some observations of a paper by Lance and Williams', The Computer Journal, 14, 156-157 (1971).

50. GOWER, J.C. and ROSS, G.J.S., 'Minimum spanning trees and single-linkage cluster analysis', Applied Statistics, 18, 54-64 (1969).

51. ROHLF, J., 'Graphs implied by the Jardine-Sibson Overlapping clustering methods, B_k', Journal of the American Statistical Association, 69, 705-710 (1974).

52. BOULTON, D.M. and WALLACE, C.S., 'An information measure for single link classification', The Computer Journal, 18, 236-238 (1975).

53. KANG, A.N.C., LEE, R.C.T., CHANG, C-L. and CHANG, S-K., "Storage reduction through minimal spanning trees and spanning forests', IEEE Transactions on Computers, C-26, 425-434 (1977).

54. SENKO, M.E., 'File organization and management information systems', Annual Review of Information Science and Technology, 4, 111-137 (1969).

55. CROFT, W.B., 'Document clustering', M.Sc. Thesis, Department of Computer Science, Monash University, Australia (1975).

56. BOULTON, D.M. and WALLACE, C.S., 'A program for numerical classification', The Computer Journal, 13, 63-69 (1970).

57. NEEDHAM, R.M., 'Problems of scale in automatic classification', In Statistical Association methods for mechanised documentation (Abstract) (Edited by M.E. Stevens et al.), National Bureau of Standards, Washington (1965).

58. COLE, a.j., Numerical Taxonomy, Academic Press, New York (1969).

59. ANDERBERG, M.R. Cluster Analysis for Applications, Academic Press, London and New York (1973).

60. WISHART, D., FORTRAN II Program for 8 Methods of Cluster Analysis (CLUSTAN I) Computer Contribution 38 State Geological Survey. The University of Kansas, Lawrence, Kansas, U.S.A. (1969).

61. EVERITT, B., Cluster Analysis, Heineman Educational Books, London (1974).

62. HARTIGAN, J.A., Clustering Algorithms, Wiley, New York and London (1975).

63. CLIFFORD, H.T. and STEPHENSON, W., An Introduction to Numerical Classification, Academic Press, New York (1975).

64. VAN RYZIN, J., Classification and Clustering, Academic Press, New York (1977).

65. SIBSON, R., 'Order invariant methods for data analysis', Journal of the Royal Statistical Society, Series B, 34, No. 3, 311-349 (1972).

66. FISHER, L. and VAN NESS, J.W., 'Admissible clustering procedures', Biometrika, 58, 91-104 (1971).

67. KERCHNER, M.D., 'Dynamic document processing in clustered collections', Ph.D. Thesis, Cornell University. Report ISR-19 to National Science Foundation and to the National Library of Medicine (1971).

68. DATTOLA, R.T., Automatic classification in document retrieval systems, Ph.D. Thesis, Cornell University (1973).

69. ZADEH, L.A., 'Fuzzy sets', Information and Control, 8, 338-353 (1965).

70. ZADEH, L.A., 'Similarity relations and fuzzy orderings', Information Sciences, 5, 279-286 (1973).

71. NEGOITA, C.V., 'On the application of the fuzzy sets separation theorem for automatic classification in information retrieval systems', Information Sciences, 5, 279-286 (1973).

72. CHAN, F.K., 'Document classification through use of fuzzy relations and determination of significant features', M.Sc. Thesis, Department of Computer Science, University of Alberta, Canada (1973).

73. RADECKI, T., 'Mathematical model of time-effective information retrieval system based on the theory of fuzzy sets', Information Processing and Management, 13, 109-116 (1977).

74. DISISS, 'Design of information systems in the social sciences. Clustering of journal titles according to citation data: preparatory work, design; data collection and preliminary analysis.' Bath, Bath University Library, Working Paper No. 11 (1973).

75. PREPARATA, F.F. and CHIEN, R.T., 'On clustering techniques of citation graphs', Report R-349, Co-ordinated Science Laboratory, University of Illinois, Urbana, Illinois (1967).

76. IVIE, E.L., 'Search procedures based on measures of relatedness between documents', Ph.D. Thesis, M.I.T., Report MAC-TR-29 (1966).

77. DUDA, R.O. and HART, P.E., Pattern Classification and Science Analysis, Wiley, New York (1973).

78. WATANABE, S., Knowing and Guessing, Wiley, New York (1969).