A graph theoretical procedure for clustering binary vectors

Dragoš Cvetković
Faculty of Electrical Engineering
University of Belgrade
11001 Belgrade
Serbia, Yugoslavia
email: ecvetkod@ubbg.etf.bg.ac.yu

ABSTRACT. We report on difficulties in applying traditional clustering procedures to discrete data. We describe a graph theoretical approach in clustering binary vectors where the number of clusters is not given in advance. New clustering procedures are combined from several algorithms and heuristics from graph theory.

1 Introduction

In clustering problems the data are usually represented by vectors from $\mathcal{R}^n$ (see, e.g., [1, 2]). A distance function $d(x, y)$ is assumed to be defined for any $x, y \in \mathcal{R}^n$. Given a set of vectors from $\mathcal{R}^n$, the problem is to partition it into subsets called *clusters* under various conditions. Clustering methods are expected to produce clusters which have the property that vectors from the same cluster in some sense are "closer" to one another than the vectors from different clusters. The number of clusters may, but need not be, given in advance. Sometimes cardinalities of clusters are given or limited by additional conditions.

We consider clustering of discrete data. A typical example of discrete data is provided by binary vectors, i.e. elements of B^n where $B = \{0, 1\}$. In [5] we have considered clustering into a given number of clusters. In this paper the number of clusters is not given in advance.

When standard clustering procedures (see, e.g., [1, 2]) are applied to binary vectors, the resulting clustering usually has a low quality. Among other things, the clustering is highly dependent of the ordering of vectors [4, 5].

To avoid these difficulties it seems reasonable to use specific properties of discrete data and to apply combinatorial, including graph theoretical, tools in handling the problem. We have developed a number of complex graph theoretical procedures for clustering binary vectors [4]. These procedures are described in [5] and in this paper.

Section 2 contains necessary definitions while our clustering procedure is described in Section 3. Section 4 contains several arguments showing the inadequacy of standard clustering procedures in clustering discrete data. The graph theoretical approach is justified in Section 5.

2 Some Definitions

We shall give definitions of some specific graph theory notions. For basic graph theory terminology see, for example, [9].

The number of coordinates in which n-tuples $x, y \in B^n$ differ is called the *Hamming distance* between x and y. A *hypercube* H_n of dimension n is the graph whose vertex set is B^n and two n-tuples are *adjacent* if they are at *Hamming* distance 1.

For a graph G we define its *k-th power* G^k. The graph G^k has the same vertex set as G and vertices x and y are adjacent in G^k if they are at (graph theoretical) distance at most k in G. For $k = 0$ the graph G^k consists of isolated vertices. For $k = 1$ we have $G^k = G$. If X is a subset of the vertex set of a graph G then $G(X)$ denotes the subgraph of G induced by X.

Let $X \subset B^n$ be a set of binary vectors (n-tuples) which is to be clustered. Our procedures for clustering makes use of the graph sequence

$$H_n^0(X), H_n^1(X), H_n^2(X), \ldots, H_n^n(X) \tag{1}$$

which is called the *basic graph sequence* and is denoted by $\mathcal{H}_n(X)$.

Note that two vectors $x, y \in X$ are at the *Hamming* distance k if they are not adjacent in $H_n^{k-1}(X)$ and are adjacent in $H_n^k(X)$. For $i = 1, \ldots, n$ the graph $H_n^i(X)$ has all edges from $H_n^{i-1}(X)$ plus those ones connecting vectors at *Hamming* distance i. $H_n^0(X)$ has only isolated vertices while $H_n^n(X)$ is a complete graph.

Let the vertex set X of a graph G be partitioned into subsets $X_1, X_2, \ldots,$ X_m. A *condensation* of G is a weighted graph on vertices $x_1, x_2, \ldots, x_m$ (called *supervertices*) in which x_i and x_j are connected by an edge if there is at least one edge between X_i and X_j in G. Both supervertices and edges in the condensation carry weights. The weight of the supervertex x_i is equal to $|X_i|$ while the weight of the edge between x_i and x_j is equal to the number of edges between X_i and X_j. We consider a condensation as a multigraph where edge weights are interpreted as edge multiplicities while supervertices as vertices and supervertex weights are ignored.

Let A be the adjacency matrix of a (multi-)graph G and let D be a diagonal matrix with vertex degrees of G on the diagonal. The matrix $C = D - A$ is called the *Kirchhoff* (or *Laplacian* or *admittance*) *matrix* of G. Let $\mu_1, \mu_2, \ldots, \mu_n$ ($\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$) be eigenvalues of C. We have $\mu_n = 0$ and the quantity $a(G) = \mu_{n-1}$ is called the *algebraic connectivity* of G (see [8] or [6], pp. 265–266).

The algebraic connectivity is known to be a very useful parameter for describing the "shape" of a graph (see, e.g., [6], p. 266). Indeed, low algebraic connectivity shows small connectivity and girth and high diameter, although such a statement lacks a precise formulation. In the context of clustering, low algebraic connectivity indicates good clustering properties.

Let Y be a subset of the vertex set X of a graph G. The set of edges connecting vertices from Y with vertices from $X \setminus Y$ is called a *separating set.* A non-trivial graph, in which every separating set has at least k ($k \geq 0$) edges, is called *edge k-connected.* A maximal k-connected induced subgraph of G is called a *k-component* for any $k \geq 1$.

In [10, 11, 12], k-components are recommended as clusters if they do not contain $(k+1)$-components. It was proved in [10, 11] that k-components in a graph are disjoint sets.

In the clustering procedure, which will be described in the next section, the following algorithms, described in the literature, will be used.

Algorithm CP. This is an algorithm for finding components of a graph ([14], pp. 398–405). One starts from a graph without edges when each vertex represents a component. Gradually, we introduce edges of the actual graph thus uniting two components if the edge added links them.

Algorithm KCP. This is an algorithm for finding k-components in a graph. The algorithm has been developed by D.W. Matula [10, 11, 12].

3 A clustering procedure

Let X be a set of binary vectors of dimension n. We shall formulate a procedure for partitioning the set X into clusters. The number of clusters is not given in advance. It is determined from the data by the clustering procedure. The procedure contains several variants for some steps or/and optional steps. It is up to the user to select variants he wishes and to compare results obtained in different ways. The variants can be selected by specifying some parameters either before the procedure starts or interactively during the procedure execution which depends on the implementation.

We consider the basic graph sequence (1) and the process of forming components by Algorithm CP in graphs from this sequence. A component is created in one of graphs $H_n^i(X)$ by uniting two or more components from $H_n^{i-1}(X)$. The *birth time* t of this component is equal to the index

i of the graph $H_n^i(X)$ in which it was created. The particular component can appear in the next several graphs from sequence (1) before it is united with some other components. The number of graphs from sequence (1) in which a component appears is the *life time* v of the component.

Trivially, we have $v \geq 1$. However, we are interested in components whose life times are large. Such components are good candidates for the clusters. The explanation is as follows.

The birth time t of a component C is, for $t > 0$, equal to the maximal Hamming distance between binary vectors represented by the vertices of C. The sum $t+v$ is equal to the minimal Hamming distance between the vectors represented by vertices of C and the remaining vectors. Therefore the component C is the better candidate for a cluster the greater the quotient $(t+v)/t$ is.

Our procedure consists of three parts. In each part some clusters may, but need not, appear.

1. Let us consider components with $v > 1$ and greatest quotient $(t+v)/t$ and among them those with a minimal number of vertices. We select one such component C. Let Y be the vertex set of C and let $Z = X \setminus Y$. We continue the clustering process by analyzing the basic graph sequence $\mathcal{H}_n(Z)$.

   If there are no components with a long life time, we cannot extract clusters in the above sense and we would try to partition the vertex set according to other criteria.

2. Suppose that the life time of any components is equal to 1. Now we are interested in components which have been created by uniting a big number of components from the previous level in a complex manner. The following parameters and their various combinations (which depend on the user) can be used in formulating criteria for the selection of a component for a cluster:

   (i) the number of components from which the components has been created;

   (ii) the number of edges by which the earlier components are interconnected;

   (iii) edge connectivity of the component.

   (iv) the birth time of the component.

   If there are no ways to extract further clusters on the basis of these parameters, then the rest of the vertices is considered as one cluster.

3. The clusters obtained in the way described in 1. or 2. can be optionally split further with the following procedure.

Each of the clusters W obtained is represented by the first graph in the graph sequence $H_n^i(W)$ ($i = 0, 1, \ldots, n$) which is connected. We form the condensation of $H_n^i(W)$ with supervertices corresponding to components of $H_n^{i-1}(W)$. Using the ratio α of the algebraic connectivity and the number of vertices we test each cluster (i.e. the corresponding condensation) to determine its suitability for further partitioning. If α is below a value given in advance by the user, we split this cluster using the algorithms KCP for finding k-components. The k-components obtained by KCP are taken as clusters (if they do not contain $(k+1)$-components).

4 Inadequacy of the standard clustering algorithms in clustering discrete data

As stated in [5], both theoretical considerations and experiments on a computer have indicated the inadequacy of standard clustering methods for handling binary vectors. For example, in hierarchical methods (e.g. single or complete linkage) at each step there are usually very many pairs of clusters which are equally good candidates to be united. Hence, we can get very different clusterings depending of the original ordering of vectors, if this ordering determines the choice.

Concerning computer experiments, we used the system PARIS [3] for standard clustering techniques and the system GRAPH [7] as well as some newly developed software [13] for graph theoretical techniques.

The distances between binary vectors (the Hamming distance and other distances) of a given dimension belong to a finite set. If the number of vectors is big we come across many pairs of vectors with the same distance between the vectors in a pair. If we have n binary vectors of dimension m, then the possible values of the Hamming distance between these vectors belong to the set $\{1, 2, \ldots, m\}$. An average number of pairs with the same distance between the vectors from the pair is $\binom{n}{2}/m = n(n-1)/2m$. For example, if we have to cluster 1000 binary vectors of dimension 1000 (i.e. $n = m = 1000$), the average number of pairs with the same distance is equal to $999/2 \approx 500$.

Both hierarchical and non-hierarchical clustering methods encounter difficulties in such a situation.

In hierarchical methods we unite at each step two temporary clusters according to some rules which include the definition of the distance between temporary clusters. The set of possible distances between the clusters is again finite and we can expect a big average number of pairs of clusters at the same distance. Hence, we may have a big choice of equally good alternatives for uniting the temporary clusters, as already mentioned. Hence, the final clustering is highly dependent of the concrete choice in particular steps. Below we quote some examples and theorems supporting these statements.

In non-hierarchical clustering methods, alternatives in step selection can arise in the following ways:

1. The choice of the initial cluster centres depends on a random generator or, equivalently, on the order of input vectors. (Note that this fact is not characteristic only for discrete data; it appears in clustering real vectors, as well).

2. In assigning vectors to clusters according to their distances from the centres of temporary clusters, we often come across the situation that a vector is at the same (minimal) distance from several cluster centres.

3. When determining new cluster centres we have to round the coordinates to 0 or 1. However, it can occur very frequently that the value of a coordinate of a new cluster center is equal exactly to 1/2. Hence, we have two equally valid possibilities of rounding.

The problem of multiple choice of alternatives in each step of clustering procedures, as described in both hierarchical and non-hierarchical clustering methods, is called the *problem of choice.*

The above analysis shows the inadequacy of standard clustering methods when they are applied to discrete data. Namely, we see that in discrete data we get a great number of possible solutions of the clustering problem depending on the decision in each concrete step of the algorithm. This fact sometimes becomes drastic and makes the clustering procedure *unreliable*, i.e., practically, *useless.*

We shall illustrate these ideas with some examples.

We start with the single linkage method, the Hamming distance and the clustering into two clusters. This is perhaps the worst combination. We shall see that under some conditions and limitations we can get as a solution *any* partition of the data into two parts.

We shall prove this statement for the following variant of the single linkage method. Binary vectors, which are to be clustered, are labeled by $1, 2, \ldots, n$. They represent initial clusters. If clusters i and j $(i < j)$ are united, the new cluster gets the number i while cluster labels greater than j become smaller by 1. Clusters at the minimal mutual distance are united. If there are several cluster pairs at a minimal distance, those are united in which the difference of labels is minimal and if there are several such cluster pairs we chose those with minimal labels.

Theorem. *Let G be a connected graph. Let $\{X_1, X_2\}$ be a partition of the vertex set X such that subgraphs G_1 and G_2, induced by X_1 and X_2 respectively, are connected. Suppose there exists a vertex x in X_2 which is* *adjacent to no vertex from X_1. Let the single linkage method, as described above, be applied to X in order to get a partition into two clusters. The length of a path is the number of edges in the path. The distance between two vertices x and y of X is defined to be the length of the shortest path joining x and y. Then there exists a labelling of X such that the single linkage method yields the partition $\{X_1, X_2\}$.*

Proof: Since graph G_1 is connected, we can label its vertices in such a way that each vertex, except for the first one, is adjacent to a vertex with a smaller label. Such a labelling of G_1 is extended by the labelling of G_2. Let x be the next vertex. Further, we assign labels to vertices of G_2 in such a way that each vertex is adjacent to a vertex of G_2 with a smaller label. This is possible since G_2 is connected. Now it is easy to see that the single linkage yields the partition $\{X_1, X_2\}$.

This completes the proof.

Suppose now that we modify the single linkage method in such a way that always clusters with minimal indices (among those with a minimal distance) are united. When clustering vertices of a connected graph into two clusters, we would get a cluster containing all but one vertex and the other cluster with a single vertex.

The complete linkage method is a little better than the single linkage. Nevertheless, the problem of choice appears also in complete linkage. Consider, as an example, the graph in Figure 1 (we assume that the distance between vertices is the usual graph theoretical distance as given in the above theorem) which has a natural bipartition (one hexagon belongs to one cluster, the other hexagon goes to the other cluster). However, if we label the vertices as in Figure 1, the complete linkage will yield the bipartition shown in Figure 1 by a broken line. The numbers associated to edges show the order in which the complete linkage method includes them. It is not difficult to find the labelling yielding the two hexagons as clusters.

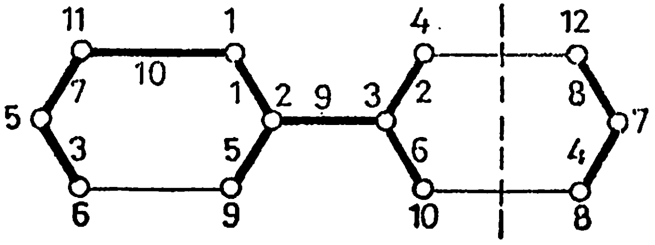

Figure 1

Another example in which the complete linkage method yields several different solutions is shown in Figure 2.

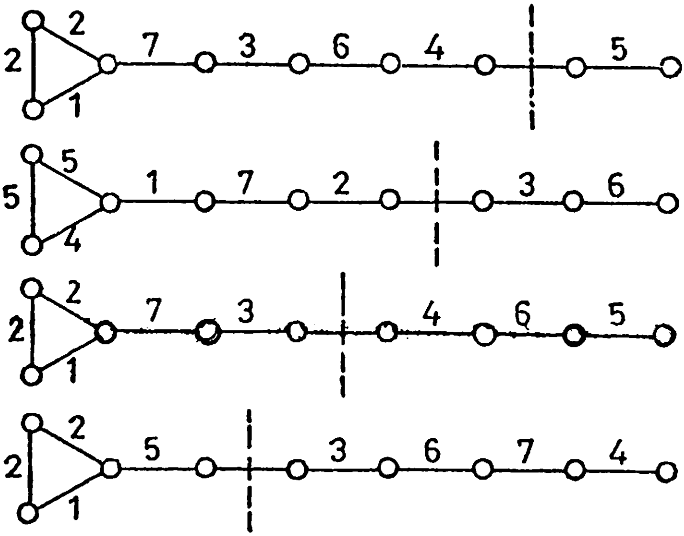

Figure 2

Beside these theoretical explanations, we used the system PARIS [3] for making experiments with standard clustering techniques which are implemented within this system. All standard algorithms showed a sensitivity to the order of binary vectors.

5 Graph theoretical approach

The insufficiency of standard clustering algorithms in clustering discrete data could be perhaps eliminated by further development. However, a successful solution of the problem of choice requires the consideration of all alternatives in each step (i.e. backtracking) or the introduction of additional criteria for closing a single alternative or a narrow class of alternatives. Criteria for the choice of alternatives obviously should depend of global properties of the set of binary vectors under clustering and not only on local properties such as the distance between temporary clusters. Global properties in question should not depend on the order of binary vectors i.e. they must be invariant under a permutation of vectors.

Since the set of binary vectors is very suitably presented by a graph and since the theory of graphs considers exactly those properties of graphs which are invariant under permutations (relabelling) of vertices, it turns out that the theory of graphs is just the mathematical tool which is suitable for problems of clustering discrete structures. Clustering results obtained by the theory of graphs are, in principle, unique, i.e. not sensitive to the relabelling of vertices. For example, a good graph theoretical procedure should always cluster the data presented in Figure 1 into two hexagons (when the clusters are required) independently of the labelling of vertices (vectors). In this way, by introducing the theory of graphs we solve the problem of choice.

Nevertheless, in some cases graph theoretical techniques can yield several solutions. However, this is not caused by the insufficiency of these techniques; it is the result of some characteristics of particular clustering problems. We quote some cases in which we do not get unique solutions:

1. The data set can have some symmetry properties. This is reflected in the richness of the automorphism group of the correspondary graph. If a solution of the clustering problem is optimal in some sense, then another optimal solution is obtained by an automorphic mapping. (For example, any bipartition with the given cardinalitics of the parts of a complete graph is optimal, in this case by any optimality criterion).

2. Even if the data set lacks any symmetry, there might exist several optimal solutions.

3. If a graph theoretically formulated clustering problem is solved by a heuristic, the suboptimal solution obtained can depend on the mechanism of the heuristic (for example, on a random number generator or on some intervention of the user).

Concerning 3., note that the standard clustering method can be realized as heuristics but, for the case of discrete data, as bad ones having in view the problem of choice.

A direct comparison of the efficiency of the standard and graph theoretical clustering procedures in clustering discrete data is not suitable since these two groups of procedures have different goals. While the standard clustering procedures ignore the problem of choice (or treat it inadequately), graph theoretical clustering procedures solve this problem by imposing additional optimality criteria. Therefore one can expect higher running times in the second group of procedures but, of course, with better solutions. However, well selected graph theoretical procedures can compete in the time complexity with the standard procedures. Without going into details, most of the standard procedures have the running times of the order of the third power of the cardinality of the data set. The same complexity has, for example, the procedure for clustering binary vectors into a given number of clusters, described in [5].

References

[1] D. Acketa, *Selected topics in pattern recognition with applications*, (Serbian), Mathematical Institute, Novi Sad, 1986.

[2] M.R. Anderberg, *Cluster analysis for applications*, Academic Press, New York, 1973.

[3] L.J. Buturović, *PARIS—an interactive system for the analysis and recognition of patterns*, (Serbian), University of Belgrade, Faculty of Electrical Engineering, Belgrade, 1988.

[4] D. Cvetković, *Combinatorial algorithms and heuristics for clustering points of a hypercube, I, II, III, IV*, (Serbian), University of Belgrade, Faculty of Electrical Engineering, Belgrade, pp. 32, 39, 53, 54, unpublished report, 1988.

[5] D. Cvetković, Graph theoretical procedures in clustering discrete data, *Univ. Beograd, Publ. Elektroteh. Fak., Ser. Mat.*, **3** (1992), 21–26.

[6] D. Cvetković, M. Doob, H. Sachs, *Spectra of graphs — Theory and application*, Academic Press, New York, 1980.

[7] D. Cvetković, I. Pevac, Man-machine theorem proving in graph theory, *Artificial Intelligence* **35** No. 1 (1988), 1–23.

[8] M. Fiedler, Algebraic connectivity of graphs, *Czechoslovak Math. J.* **23(98)** (1973), 298–305.

[9] F. Harary, *Graph theory*, Reading, 1969.

[10] D.W. Matula, The cohesive strength of graphs, The many facets of graph theory, *Proc. Conf.* held at Western Michigan Univ., Kalamazoo, MI, 1968, eds. G. Chartrand, S.F. Kapoor, Springer-Verlag, Berlin-Heidelberg-New York, 1969, 215–221.

[11] D.W. Matula, *k*-components, clusters, and scalings in graphs, *SIAM J. Appl. Math.* **22** No. 1 (1972), 459–480.

[12] D.W. Matula, Graph theoretic techniques for cluster analysis algorithms, *Classification and Clustering*, ed. J. Van Ryzin, Academic Press, New York, 1977, 95–129.

[13] S. Petrović, *Algorithms and heuristics for clustering vertices of a graph with applications to pattern recognition*. (Serbian), Master Thesis, University of Belgrade, Faculty of Electrical Engineering, Belgrade, 1989.

[14] R. Sedgewick, *Algorithms*, Addison-Wesley, Reading, 1983.