Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 60 – 69

Procedia Computer Science

2013 International Conference on Computational Science

Data analysis with intersection graphs

V. M. Vairinhos [a,*], V. Lobo [a], P. Galindo Villardón [b]

[a]Centro de Investigação Naval, Escola Naval, Almada 2810-001, Portugal
[b]Departamento de Estadística, Universidad de Salamanca, Salamanca 37007, España

Abstract

This paper presents a new framework for multivariate data analysis, based on graph theory, using intersection graphs [1]. We have named this approach DAIG – Data Analysis with Intersection Graphs. This new framework represents data vectors as paths on a graph, which has a number of advantages over the classical table representation of data. To do so, each node represents an atom of information, *i.e.* a pair of a variable and a value, associated with the set of observations for which that pair occurs. An edge exists between a pair of nodes whenever the intersection of their respective sets is not empty. We show that this representation of data as an intersection graph allows an easy and intuitive geometric interpretation of data observations, groups of observations, and results of multivariate data analysis techniques such as biplots, principal components, cluster analysis, or multidimensional scaling. These will appear as paths on the graph, relating variables, values and observations. This approach allows for a compact and memory efficient representation of data that contains many missing values or multi-valued attributes. The basic principles and advantages of this approach are presented with an example of its application to a simple toy problem. The main features of this methodology are illustrated with the aid software specifically developed for this purpose.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science

Keywords: Categorical data ; data models ; data structures ; intersection graphs ; multivariate data analysis.

1. Introduction

The widespread use of information technology has led to a dramatic increase in the availability of raw data, which in turn has driven the need to use new and more powerful multivariate data analysis techniques. One of

* Corresponding author . Tel.: +351-918-234-755
E-mail address: valter.vairinhos@sapo.pt

1877-0509
Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.169

the trends is the importance given to analysis of categorical data using graph theory, as can be seen in [2] and [3].

The use of graphs for multivariate data analysis is not new. In [4] random graphs are used to interpret results of multivariate data analysis, in particular cluster analysis. Here, nodes represent objects/observations and the existence of an edge between two nodes depends on the associations detected between those nodes by cluster analysis methods. The vast majority of graph-based data analysis techniques use this type of approach, e.g. [5, 6]. In [7] the concept of Formal Concept Analysis is developed, based on the algebraic concept of lattice, and the associated graphs. Here, variables are related with objects / observations using the concept of formal context. In [8] Association Graphs are used to represent conditional associations among variables. In [9-11] graphical models are developed to represent conditional independence amongst variables using graphs. In those approaches, nodes correspond to variables, and edges are added to relate concepts, in [7] for example, or when there is conditional dependence amongst variables, in [9-11]. A different way to use graphs for data analysis is presented in [2], where bipartite graphs are used. In this approach some nodes are observations while others are variables. Another example of the use of graphs in data analysis is given in [12] where the graphs are used to segment images. In this approach the vertices are data elements (pixels, lines, etc), and edges are the relations amongst these elements. The graphs obtained are clustered with well known algorithms. More recently, [13] developed a graph-based semi-supervised learning algorithm but, once again, the nodes are observations and the edges represent relations amongst them. In none of these approaches are graphs defined using variable-values pairs. In [22] new algorithms to detect cliques in graphs are presented.

The computational advantages and disadvantages of using graph-based approaches to clustering have been discussed in [14, 15], but it must be noted that in those cases the issue is clustering the graphs themselves. That is not the case in this paper: graphs are not clustered, but are the means used for clustering other data.

This paper presents a new framework for multivariate data analysis, based on graph theory that uses intersection graphs to store the data. We have named it DAIG: Data Analysis with Intersection Graphs, and it was first proposed in [16]. This approach to data analysis breaks away from the traditional table/matrix based representation of data, and uses intersection graphs as the basic representation. In this approach, nodes are not observations nor variables, but sets of observations corresponding to pairs of (variable, value). These nodes are named atoms. We will try to show that this approach, which is very different from previous graph-based approaches, can in some cases be more flexible and powerful than tables, and can make good use of the extensive recent work on graph theory.

To illustrate what DAIG is, we shall use an example taken from the well known zoo dataset from the UC-Irvine Machine Learning Repository [17], that is composed of 17 characteristics of 101 different animals. Most of these characteristics (15 of them) are binary: hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, tail, domestic, catsize. One of the characteristics, the type, is categorical (mammal, bird, fish, etc). The last characteristic is a quantitative discrete variable (number of legs) that can be treated as categorical since it only has a small number of distinct values.

2. Formalization of DAIG

The main building blocks, or atoms, for our representation of data are sets of observations corresponding to pairs of (variable, value). When the input data is in the form of a table with n rows and p columns, an observation o_i (i = 1…n) is formed by p values of characteristics or variables X_j (j = 1...p), each with a particular value x_{ij}. We can thus represent the observation o_i by p pairs of the form (X_j= x_{ij}), (i=1...n; j=1...p), called atoms. Using these atoms as nodes of a graph, and since these atoms are connected in the sense that they form an observation, we connect them with edges, as can be seen in figure 1.

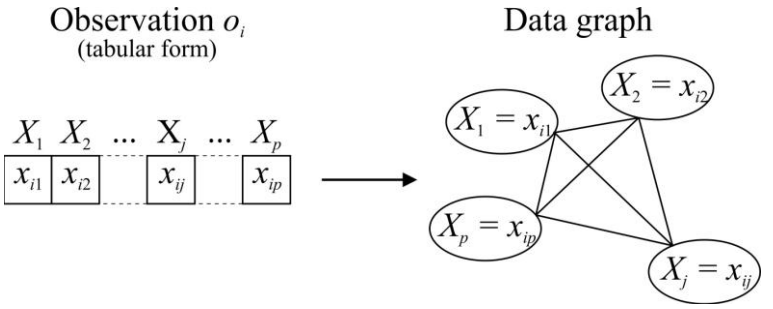


Fig. 1. Tabular and data graph representation of an observation. Each node of the graph is the set of observations for which a specific variable has a specific value. Two nodes are connected by an edge if the corresponding sets have a non empty intersection.

On this data graph, an observation or object is a clique, i.e., a set of nodes/atoms that are mutually connected.

Let us see a practical example. Suppose we observe 3 characteristics of a chicken, and present them in the traditional tabular form. It's representation on a data graph is a 3-clique, as seen in figure 2.

As more observations are made, their representation can be added to the graph. There are various ways to do this, and, depending on what sort of data analysis is desired, we may prefer one or the other. When we have various observations, these will probably share some atoms, i.e., some observations will have the same values for the same variable. This information may be explicitly recorded by each atom (node) associating with it the list of the observations that contain them, as seen in figure 3. This graph is an intersection graph [1], because the edges correspond to non-empty intersections of the sets of observations that form the nodes.

When the observations are in the form (object's identification i, variable X_j, value v_j) – meaning that object i has the value v_j for variable X_j - all that has been said before applies. The only difference is that, now, each observation corresponds to a single atom.

We may or may not associate weights to the edges of this intersection graph, depending on the objective. When necessary, those weights can be defined as a function of sets corresponding to adjacent nodes.

Those weights may be, for example, the cardinality of the intersection of those sets. Another way of defining weighs may be using the function $f(a,b) = |a \cap b|^2 / |a| * |b|$ where a and b are the sets defining the adjacent nodes. This function corresponds to the proximity measure P(a|b) * P(b|a), which can be useful in many analysis [16], and which we named affinity.

Thus, depending of the objective, after adding various observations, we will have:

- An intersection graph, where each node has an associated set of observations. This is the preferred representation because it retains all the information about individual observations, and thus allows all sorts of analysis to be performed.

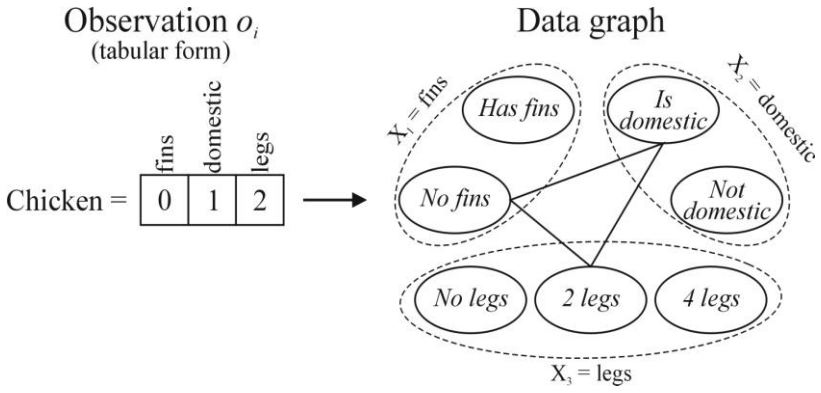


Fig. 2. Tabular and data graph representation of chicken .The concept {Chicken} is represented by the clique {No fins, Two legs, Domestic}.

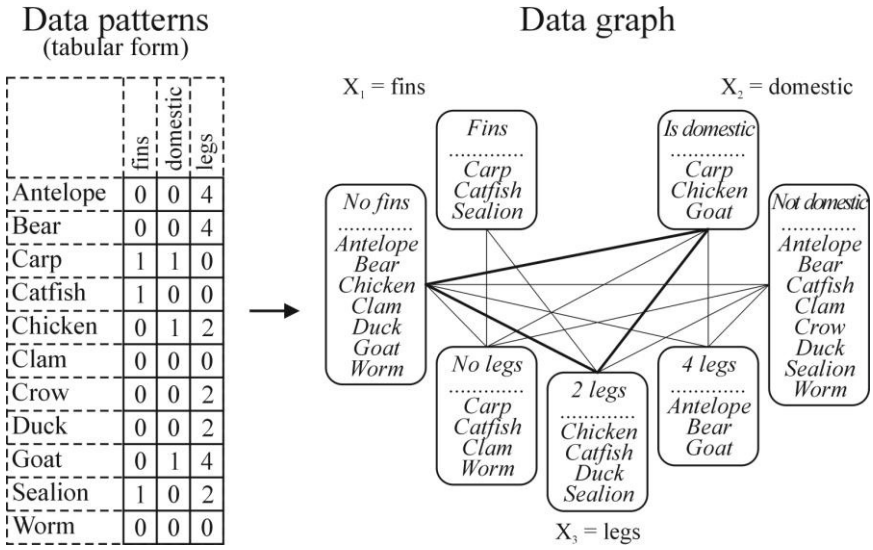


Fig. 3. Data graph of a set of observations.

- A weighted intersection graph, where ach edge has an associated weight. For example, the weight may be the number of co-occurrences of the atoms in the observations | a ∩ b | or any other function as convenient for the objective of the study.

When the input data is in the form of a table with n rows and p columns, the minimum degree of each node is the number of variables minus one. However, it can be less if there are missing values. Generally, the degree will be greater than the number of variables, since different observations will have different values for the other variables. Even if the graph represents a single observation, the degree of a node can be greater than the number

of variables minus one if there are multivalued variables, i.e., there is more than one value for a given variable.

It must be stressed that each node/atom in this intersection graph is a simple concept. The meaning of that concept is represented both by intent, using the pair (Variable, value) as in (Legs, 0), or by extent, using the set of objects for which (Variable = value) as in {Carp, Catfish, Clam, Worm}. In both cases, the concept is the same (zero legged animals). What changes is the expression of its meaning, not the meaning itself.

3. Some useful consequences and techniques

The proposed framework leads to many useful results and allows for efficient implementations of data analysis techniques. One of those implementations can be seen in [16] where this framework has been used to suggest interpretations of concepts discovered with biplots [18, 19]. See 3.6. Each node or path in the intersection graph corresponds to a concept described by intent. For example, the path (Legs = 4) and (Fins = 0) and (Domestic=1) is a concept described by intent as "4 legged domestic animals without fins". If the nodes that form this path have a non-empty intersection, that intersection is the extent (i.e. enumeration) of the corresponding complex concept, which in this case would be {Goat}. In this case, that path is part of a clique. Otherwise, when the nodes along the path have an empty intersection, as in (Fins = 1) and (Legs = 4), the concept has no support in the data. Let us now see some of the useful techniques and results developed for this framework.

3.1. Representing tables in DAIG

Intuitively, the construction of the data structure for DAIG is just like a chemical analysis. An observation is broken down into its constituting atoms. For each variable that characterizes the observation, the existing graph is updated by adding the label of the new observation to the node that corresponds to its value. Edges will be added to the other nodes that compose the observation, if necessary. The edges can always be added later by finding non-empty intersections of the nodes. The weights of those edges can also be computed, since they are a function of the cardinality of those intersections. Adding the edges as the observations are introduced has the advantage of reducing the computing time, since it avoids computing empty intersections.

On this intersection graph, a complex concept with support in data is represented as a non-empty intersection path, and can be seen, intuitively, as a compound molecule composed of distinct atoms/nodes.

3.2. Data objects and cliques

As stated previously, if observations are characterized by p variables, they are p-cliques in the intersection graph. It must be noted that in many real problems there are missing values. In this case, if only $m < p$ variables where measured for an observation, then the corresponding clique will have lower order.

A clique in the graph may correspond to an observation, a set of observations, or no observation at all. A p-clique corresponds to a complete specification of a data observation, and will thus correspond to single datum, a set of undistinguishable ones, or a non-existent object. Whether that observation exists or not can only be determined by computing the intersection of all the nodes involved. If an empty set is obtained, then no observation with all those characteristics was made, meaning that there is no support for the concept in the data.

A m-clique (with $m < p$), will be a generalization, or more general concept, since some variables that characterize the observations are left out. It is important to note that concepts may be defined in such a way that they do not form a clique. For example, the concept "domestic animals with 2 or 4 legs", which corresponds to {chicken, goat}, or the edges (domestic=1) $\wedge$ (legs=2) and (domestic=1) $\wedge$ (legs=4), is not a clique in the graph. In this case, the concept will be represented by a set of cliques.

3.3. Multivalued characteristics

One of the advantages of DAIG is the ease with which it deals with multi-valued characteristics. These types of variables are not very common, but they do occur, and are dealt with in a rather awkward manner by the traditional tabular approach. An example of a possibly multi-valued variable is the nationality of a person. While most people have a single nationality, it is possible to have 2 or more. While this constitutes a problem for some representations, a DAIG will simply add the identifier of the observation to the various nodes that correspond to the different nationalities. The variable set will no longer be an independent set, but for most graph algorithms that is not important.

3.4. Sparse datasets

The proposed DAIG is far more efficient than the traditional table-based representation when there are many missing values in the data. This is a situation that is common in real data, for a number of reasons. A table based representation has to waste space with non-existing values, while the graph-based representation does not: only observed values are stored. Naturally, if desired, the missing values may be coded as special values.

3.5. Representing k-way contingency tables in DAIG

Another big advantage of DAIG is the ease with which multiway contingency tables can be represented. A cell in a p-way contingency table can be seen as the intersection of p atoms corresponding to the p variables, having as absolute frequency the cardinal of that intersection.

It is then easy to see that a cell in a p-way table is represented on an intersection graph as a p-clique. This means that, in DAIG, a contingency table is represented by the set of p-cliques corresponding to its cells. This representation means that a contingency table can be visualized in a parallel coordinates graph [20] by the set of paths that represent its cells on the corresponding intersection graph.

An example of this type of analysis is show in figure 4 where the 17-way contingency table of the 101 animals of the zoo dataset was filtered to show only the edges with a weight of more than 70. The interpretation of this sketch is quite intuitive. If desired, the weighs of the edges can be shown or can be coded as thickness of the edges.

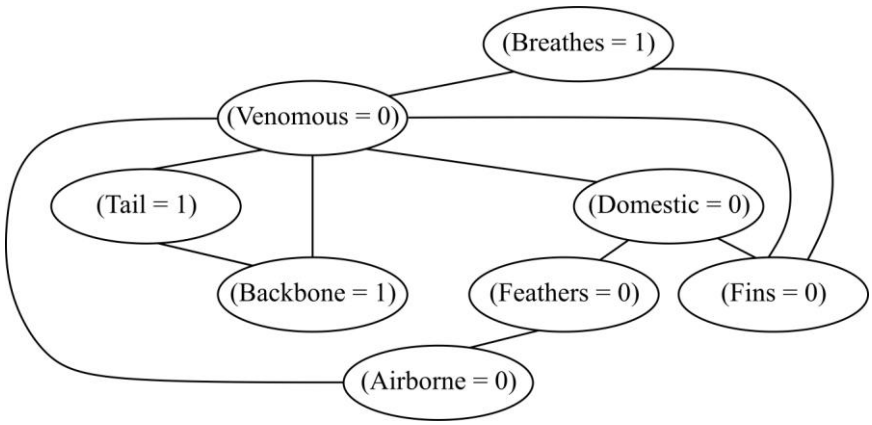


Fig. 4 Sketch of the data graph of the 17 characteristics of 101 animals, filtered to include only edges with more than 70 occurrences.

3.6. Automatic construction of multivariate results descriptions

One important problem of multivariate data analysis, not yet fully addressed in the statistical literature, is the problem of automatic construction of meaningful descriptions of results. When we have multiple analysis, relating the different results is even more difficult.

The results of many multivariate data analysis algorithms are sets of observations whose description is many times complex, and not easily conveyed to users or even researchers. For example, let us assume that a given clustering algorithm gives us a set of thousands of observations forming one of the clusters. It is important to express this set in a short and meaningful way. This can be done giving a statistical summary of the values of the objects that belong to that cluster. However, these kinds of expressions ignore a great deal of information, and many times are not very intuitive to the average user. Using DAIG, this problem consists of identifying the "main" paths along the graph, *i.e.* the paths that have higher weights or cardinality of interception.

For example, let us assume that a cluster analysis of the zoo data includes a cluster defined by the following set: *{Porpoise, Dolphin, Sea lion, Seal, Dog fish, Tuna, Pike, Stingray, Bass, Cat Fish, Chub, Piranha, Herring, Haddock, Sole, Sea horse, Sea Snake, Carp }*. How can we describe this set using, as descriptions, conjunctions of atoms of observed variables? Those conjunctions correspond to intersections of the corresponding atoms and, therefore, to paths in the intersection graph. The problem can be approximately solved as the search for the "best" approximation of the concept to be described using non-empty paths over the intersection graph.

For the present problem a possible solution, supplied by the DAIG software would be the automatically generated description *(Fins = 1) And (Toothed =1)* having a support almost coincident with the set to describe.

4. A computer implementation and application

A data-analysis program has been developed using the DAIG framework. The program was originally developed as part of a PhD thesis [16], and improved versions [21] have been used to analyze maintenance data of ships.

The main purpose of this program – that can be downloaded from http://www.isegi.unl.pt/docentes/ /vlobo/Projectos/projectos.htm – is to illustrate the features of this representation and to show how this approach enables us to easily perform some functions of multivariate data analysis.

The main features of the software are:

- Represent data as an intersection graph using the variables selected by the user.
- Filter the intersection graph leaving only edges with cardinality above some threshold defined by the user.
- Generate the graphical representation of the current intersection graph, based on an interaction with the WinGraphviz library from AT&T Research (www.graphviz.org/Credits.php).
- Compute the distribution of the node's degrees and fit this distribution to a power law.
- Compute the edge weight's distribution and significance.
- Automatically construct conjunctive expressions as best fit approximations – in the sense of affinity – of the meaning of arbitrary concepts given by lists of individuals.
- Construct k-way contingency tables and graphically represent them as intersection graphs.

A screenshot of the opening window is presented in figure 5. From that screen the user may choose 4 options: 1- Open a data file and convert it to DAIG; 2- Perform a statistical analysis of the data; 3- View and analyze the DAIG graph; and 4-View contingency tables.

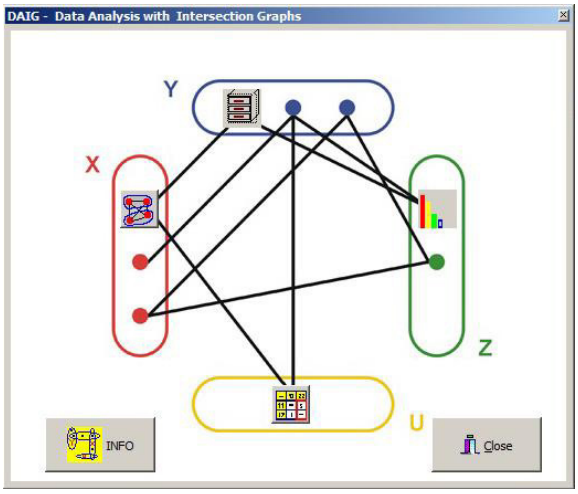
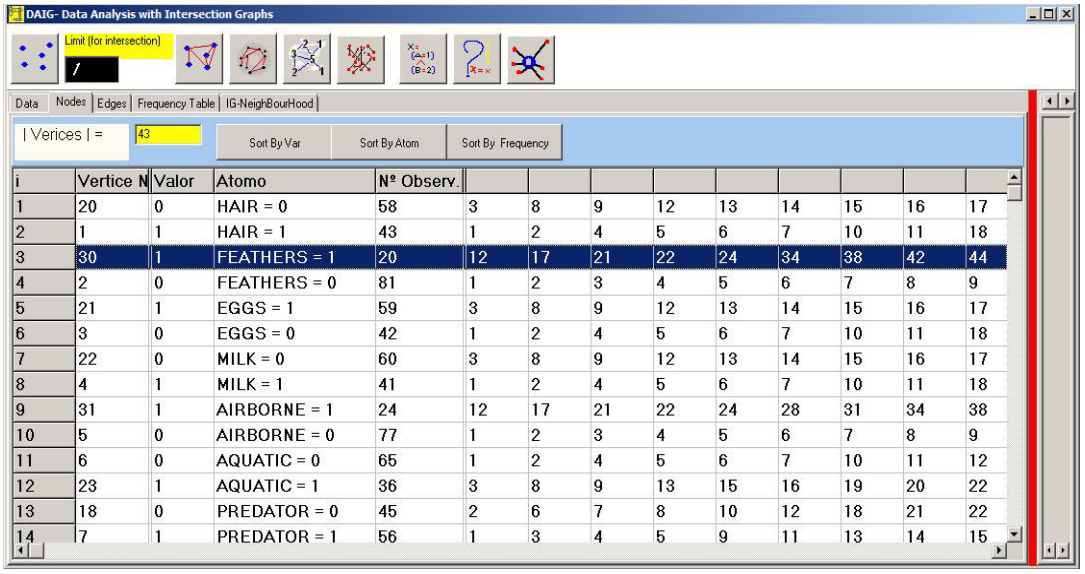


Fig. 5 Screenshot of the opening window of the DAIG software.

In the DAIG graph window (see figures 6 and 7) we may construct and study the intersection graph, the distributions of degrees, edge weights, affinity indexes, and automatically obtain conjunctive expressions as approximations of concepts given by lists of observations.

Figure 6 shows that for the data set Zoo (101 rows and 17 variables) there are 43 atoms some of them are shown. For example, atom Fathers = 1 has a cardinal 20 and the corresponding support set are animal identified by numbers in the set {12, 17, 21, 22, 24, 34, 38, 42, 44, 57, 58, 59, 60, 72, 79, 80, 84, 88, 96, 101}.

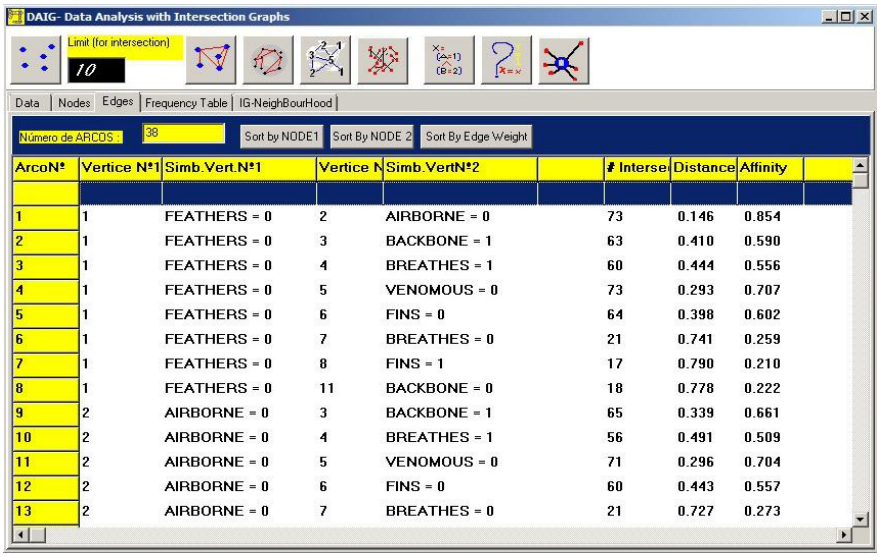


i	Vertice N	Valor	Atomo	Nº Observ.								
1	20	0	HAIR = 0	58	3	8	9	12	13	14	15	16	17
2	1	1	HAIR = 1	43	1	2	4	5	6	7	10	11	18
3	30	1	FEATHERS = 1	20	12	17	21	22	24	34	38	42	44
4	2	0	FEATHERS = 0	81	1	2	3	4	5	6	7	8	9
5	21	1	EGGS = 1	59	3	8	9	12	13	14	15	16	17
6	3	0	EGGS = 0	42	1	2	4	5	6	7	10	11	18
7	22	0	MILK = 0	60	3	8	9	12	13	14	15	16	17
8	4	1	MILK = 1	41	1	2	4	5	6	7	10	11	18
9	31	1	AIRBORNE = 1	24	12	17	21	22	24	28	31	34	38
10	5	0	AIRBORNE = 0	77	1	2	3	4	5	6	7	8	9
11	6	0	AQUATIC = 0	65	1	2	4	5	6	7	10	11	12
12	23	1	AQUATIC = 1	36	3	8	9	13	15	16	19	20	22
13	18	0	PREDATOR = 0	45	2	6	7	8	10	12	18	21	22
14	7	1	PREDATOR = 1	56	1	3	4	5	9	11	13	14	15

Fig. 6 Main DAIG analysis window, showing some atoms corresponding to the variables Hair, Feathers, Eggs, Milk, Airbone, Aquatic and Predator.

Figure 7 shows information about the edges with cardinal over the threshold 10. From this window we can plot the data intersection graph, as seen in figure 4. When plotting, it is useful to filter edges or nodes, which may be done dynamically with this software. In as seen in figure 4, for example, only the edges that have a weight of more than 70, i.e., all pairs of characteristics that co-occur in more than 70 of the 101 animals of the dataset are shown.

The program was written in Borland Delphi using dynamic structures to implement the data graph, and uses the public domain Graphviz library to plot the graphs.



ArcoNº	Vertice Nº1	Simb.Vert.Nº1	Vertice N	Simb.VertNº2	# Interse	Distance	Affinity
1	1	FEATHERS = 0	2	AIRBORNE = 0	73	0.146	0.854
2	1	FEATHERS = 0	3	BACKBONE = 1	63	0.410	0.590
3	1	FEATHERS = 0	4	BREATHES = 1	60	0.444	0.556
4	1	FEATHERS = 0	5	VENOMOUS = 0	73	0.293	0.707
5	1	FEATHERS = 0	6	FINS = 0	64	0.398	0.602
6	1	FEATHERS = 0	7	BREATHES = 0	21	0.741	0.259
7	1	FEATHERS = 0	8	FINS = 1	17	0.790	0.210
8	1	FEATHERS = 0	11	BACKBONE = 0	18	0.778	0.222
9	2	AIRBORNE = 0	3	BACKBONE = 1	65	0.339	0.661
10	2	AIRBORNE = 0	4	BREATHES = 1	56	0.491	0.509
11	2	AIRBORNE = 0	5	VENOMOUS = 0	71	0.296	0.704
12	2	AIRBORNE = 0	6	FINS = 0	60	0.443	0.557
13	2	AIRBORNE = 0	7	BREATHES = 0	21	0.727	0.273

Fig. 7 Main DAIG analysis window, some edges connecting the atoms of variables Feathers and Airborne to other atoms.

6. Conclusion

An alternate way of looking at data analysis was presented and explained. This approach is based on graph theory, representing data as an intersection graph. This provides an environment for reasoning geometrically about data and concepts, using common intuition. It also provides a powerful mathematical framework, since it allows data analysis to be made using graph theory algorithms.

The main difference between this approach and other graph based data analysis techniques is that nodes are elementary concepts whose meaning is indivisible. This meaning is given by intent as a pair (Variable, value), or by extent as the set of observations for which that variable has that value. The intersection graph is obtained with these sets. A path of length k corresponds to the conjunction of the nodes (elementary concepts) that form it, and is thus a complex concept. Any set of observations with the same values for k variables, is represented by a k-clique. In particular a single observation (or datum) with p characteristics is represented by a p-clique.

This representation has several advantages over other approaches, such as avoiding non-structural missing values, providing an easy representation of multivalued variables, and a geometric representation of k-way contingency tables.

To test the validity of this approach, a prototype program has been developed. This software proved to be useful in practical situations.

References

[1] McKee, T.A., McMorris, F.R., 1999. *Topics in intersection Graph Theory*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

[2] Michailidis, G., Leeuw, J.D., 2001. Data Visualization Through Graph Drawing. *Computational Statistics*, vol. 16, p. 435-450.

[3] Cook, D. J., Holder, L. B., 2006. *Mining Graph Data*. New York: John Wiley.

[4] Godehardt, E., 1988. *Graphs as Strucutral Models*. Friedrich Vieweg & Sohn Verlag.

[5] Matula, D. W., 1977. *Graph Theoretic Techniques for Cluster Analysis Algorithms. Classification and Clustering*. V. Ryzin, Ed.: Academic Press, p. 95-129.

[6] Wu, Z., Leahy, R., 1993. An Optimal Graph Theoretic Approach to Data Clustering: Theory and its Application to Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, p. 1001-1113.

[7] Ganter, B., Wille, R.., 1999. *Formal Concept Analysis*. New York: Springer Verlag.

[8] Agresti, A., 1996. *An Introduction to Categorical Data Analysis*. New York: John Wiley & Sons.

[9] Whittaker, J., 1990. *Graphical Models in Applied Statistics*. John Wiley.

[10] Edwards, D., 1995. *Introduction to Graphical Modeling*. Springer.

[11] Lauritzen, S. L., 1996. *Graphical Models*. Oxford Science Publications.

[12] Amir, A. Lindenbaum, M., 1998. A Generic Grouping Algorithm and its Quantitative Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, p. 168-185.

[13] Culp, M., Michailidis, G., 2008. Graph-Based Semisupervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, p. 174-178.

[14] Wilson, R. C., Hancock, E. R., Luo, B., 2005. Pattern Vectors from Algebraic Graph Theroy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, p. 1112-1124.

[15] Dhillon, I. S., Guan, Y., Kulis, B., 2007. Wheighted Graph Cuts without Eigenvectors: *A Multilevel Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, p. 1944-1957.

[16] Vairinhos, V. M., 2003. *Desarrollo de un Sistema para Minería de Datos Basado en los Metodos Biplot. Salamanca*, Spain: Universidad de Salamanca.

[17] UCI, 2008. *Machine Learning Repository*. University of California at Irvine, vol. 2008.

[18] Gabriel, K. R., 1971. The Biplot Graphic Metrics with Application to Principal Component Analysis. *Biometrika*, vol. 58, p. 453-467.

[19] Galindo, M. P., 1986. Una Alternativa de Representación Simultanea: HJ-Biplot, *Qüestiió*, vol. 10, p. 12-23.

[20] Inselberg, A., Dimsdale, B., 1990. Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. *IEEE Visualization,* p. 361-378.

[21] Vairinhos, V. M., 2004. Biplots PMD-Data Mining Centrada em Biplots. Apresentação de um Protótipo, *in Joclad'2004 - XI Jornadas de Classificação e Análise de Dados*, Lisboa.

[22] Bhamaikar, A.A., Rao, P.R., 2012. Detecting Cliques Using Degree of Connectivity Constraints. *International Journal of Data Mining & Knowledge Management Process*, vol. 2, No. 2, March.