

# A new visualization tool for data mining techniques

José M. Martínez-Martínez<sup>1</sup> · Pablo Escandell-Montero<sup>1</sup> · Emilio Soria-Olivas<sup>1</sup> · José D. Martín-Guerrero<sup>1</sup> · Antonio J. Serrano-López<sup>1</sup>

Received: 1 December 2015 / Accepted: 29 December 2015 / Published online: 18 January 2016  
© Springer-Verlag Berlin Heidelberg 2016

**Abstract** Clustering techniques and classification trees are two of the main techniques used in data mining but, at present, there is still a lack of visualization methods for these tools. Many graphs associated with clustering, also with hierarchical clustering, do not give any information about the values of the centroids' attributes and the relationships among them. In classification trees, graphical procedures can also be developed to help simplify their interpretation and to obtain a better understanding, but more visualization methods to support this tool are needed. This paper presents a novel visualization technique called sectors on sectors (SonS), and an extended version called multidimensional sectors on sectors (MDSonS), for improving the interpretation of several data mining algorithms. These methods are applied for visualizing the results of: (a) hierarchical clustering, which makes it possible to extract all the existing relationships among centroids' attributes at any hierarchy level; (b) growing hierarchical self-organizing maps (GHSOM), a variant of the well-known self-organizing maps (SOM), by means of which it is possible to visualize, simultaneously, the data information at each hierarchy level compactly and extract relationships among variables; (c) classification trees, in which the SonS is used for representing the input data information for each class presented in each terminal node of a classification tree providing extra information for a better understanding of the problem. These methods are tested by means of several data sets (real and synthetic). The achieved results show the suitability and usefulness of the proposed approaches.

**Keywords** Data visualization · Clustering visualization · Growing hierarchical self-organizing maps · Classification trees

## 1 Visualization methods for the data mining techniques

There are a number of well-known techniques for visualizing data, such as  $x$ - $y$  plots, line plots, and histograms. These techniques are useful for data exploration, but are usually limited to relatively small and low dimensional data sets. In the last decades, a large number of novel information visualization techniques have been developed, allowing visualizations of multidimensional data sets. Good overviews of the most visual methods can be found in [9,31,34].

Also, many visualization techniques have been developed to support specific data mining tasks, such as classification and clustering. On one hand, in classification, the most popular approaches are algorithms that construct decision trees. Since most algorithms work as black-box approaches, it is often difficult to understand and optimize the decision model. There exists some tools for these tasks, as the decision tree visualizer in SGIs MineSet system<sup>®</sup> that shows an overview of the decision tree together with important parameters such as the attribute value distributions. The system allows an interactive selection of the attributes shown and helps the user understand the decision tree. A more sophisticated approach which also helps in decision tree construction is visual classification, as proposed in [4]. The basic idea is to show each attribute value by a colored pixel and arrange them in bars. The pixels of each attribute bar are sorted separately and the attribute with the purest value distribution is selected as the split attribute of the decision tree. These methods help to optimize the model generation and the classification process, but

✉ José M. Martínez-Martínez  
jose.maria.martinez@uv.es

<sup>1</sup> Electronic Engineering Department, University of Valencia, Av de la Universidad, s/n, 46100 Burjassot, Valencia, Spain

they do not help to extract knowledge and to obtain a better understanding of a classification tree.

On the other hand, results from partitioning cluster analysis can be visualized by projecting the data into a two-dimensional space. Cluster membership is usually represented by different colors and glyphs, or by dividing clusters into several panels of a trellis display [10]. In addition, silhouette plots [29] provide a popular tool for diagnosing the quality of a partition. One remarkable visualization tool is the well-known self-organizing map [18]. SOM is a visualization tool rather than a clustering tool, and it cannot be used to visualize the results of classical clustering algorithms. Moreover, sometimes, the high-dimensional data sets involve some level of hierarchical structure, making the use of the same visualization tools difficult [10,32]. Regarding hierarchical clustering, it is difficult to find methods for visualizing their results. Hierarchical cluster analysis is almost always accompanied by a dendrogram, which is an effective means of representing the sequence of clusterings produced by an agglomerative, or divisive, algorithm [32]. Cutting the dendrogram at a specific level results in clustering. Another visualization tool is the so-called treemap [30]. A treemap works by dividing the display area into a nested sequence of rectangles, whose areas correspond to an attribute of the data set. In some works, treemaps are used to visualize hierarchical clustering [5,21,24,33]. Also, other popular tools as convex cluster hulls or silhouettes are specific to clustering [10]. The dendrogram is an excellent tool to determine the number of clusters in a given hierarchical data set, treemaps are very useful when visualizing the hierarchy, and convex cluster hulls and silhouettes give information about how the centroids partition the input space, and how well each object lies within its cluster, respectively. Nevertheless, these techniques do not provide any information about the values of the attributes in each cluster centroid and the relationships among them. This drawback is solved in the visualization methods proposed in this paper, which are also able to visualize hierarchical structures.

In spite of the lack of the methods for hierarchical clustering visualization, there are techniques for defining hierarchical information structures, that is, structured information, previously stored, in a hierarchical way, e.g., the file system on a computer, the organization of employees, Internet addressing, library cataloging, etc. These techniques are based on hierarchical visualization, but they do not use clustering algorithms since the hierarchy and the clustering are known a priori. Some of these techniques are the classic tree drawing algorithm for ordered binary trees [27], cheops [6], hierarchical edge bundles [15], or reconfigurable disc tree (RDT) [16]. There are also some software or Graphical User Interfaces as hyperbolic browser [3], Information slices [2], Magic Eye View [19], Cone Trees [28], Information Pyramids [1] and 3D Hyperbolic Browser [26], among others.

The point is that these techniques are used when the hierarchy is very deep. Thus, they aim to represent correctly the hierarchy given by the structured information already stored, and not to extract information about the relationship among the attributes, which is the goal of the approaches presented in this paper.

The rest of this paper is organized as follows. The details of the proposed methods are described in Sect. 2. The data sets used to validate the proposed methods are described in Sect. 3. In Sect. 4, the proposed methods are applied to several data mining techniques, such as hierarchical clustering, growing hierarchical self-organizing maps (GHSOM) and classification trees, for visualizing the achieved results in the mentioned data sets. Finally, Sect. 5 summarizes the conclusions of the proposed visualization techniques.

## 2 Methods

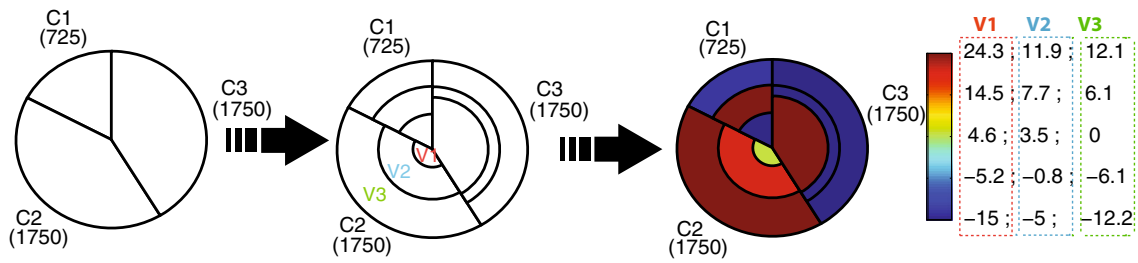
This section is devoted to a detailed description of both the graphics produced by the SonS and MDSONS visualization methods to show up the differences between both methods. Moreover, they are explained in detail to understand how they are interpreted when applied to a data set. Both methods are published as software package tool on the machine learning open source software website.<sup>1</sup>

### 2.1 Sectors on sectors (SonS)

Sectors on Sectors (SonS) is a visualization method that extracts visual information of data groups by representing the number of instances in each group, the value of the centroids of these groups of data and the existing relationships among the several groups and variables. This method is based on the well-known pie chart visualization. Each cluster is represented by a slice of a circle (pie sectors). The arc length of each pie sector is proportional to the number of patterns included in each cluster. By means of new divisions in each pie sector and a color bar with the same number of labels as attributes, the existing relationships among centroids' attributes of the different clusters can be inferred. Figure 1 represents the three steps followed to create the SonS visualization method, which are stated as follows:

1. Division of one circle on several sectors depending on the number of clusters: First of all, the circle is divided into several pie segments or sectors corresponding to each cluster. The arc length of each sector is proportional to the number of patterns included in each cluster. The number of patterns belonging to each cluster is shown within

<sup>1</sup> <http://mloss.org/software/view/469/>.



**Fig. 1** The three steps followed to create the SonS visualization method. From left to right producing as many sectors as clusters; splitting each sector according to the attributes; and color coding to identify real values (color figure online)

parentheses. In this way, the significance of each cluster is easily recognizable (Fig. 1, left).

2. Division of the pie sectors depending on the number and the value of attributes: After the first step, each sector is divided into as many subsectors as variables presented in the problem. The inner part corresponds to the first variable, and going outwards, the next variables appear. Each one of these parts vary in the radius. This radius corresponds to the relative value of each variable, with respect to the sum of all of them.<sup>2</sup> That is, let  $X$  be a centroid corresponding to one cluster, so that,

$$X = \{x_1, x_2, \dots, x_N\}. \tag{1}$$

Then, the radius of each subsector (corresponding to each centroid attribute) is calculated as follows:

$$r_i = \frac{|x_i|}{\sum_{i=1}^N |x_i|}, \quad i = 1 \dots N. \tag{2}$$

In this way, the bigger the radius corresponding to each variable, the higher is the weight of the variable and, therefore, the more relevant the feature. This is a good method to identify the relevance of each variable within each cluster in a straightforward way (Fig. 1, middle).

3. Color coding for identifying the real value of features: Attached to the graph, there is a color bar with the same number of labels as variables (each label for each variable). The mean value of the variables of each class (normally, the centroid) is codified by means of colors.<sup>3</sup> The value of the color for the first feature (inner subsector) is given by the first column label, the second feature

by the second column label and so on. In this way, it is possible to know the exact value of each variable for each cluster centroid (Fig. 1, right).

### 2.2 Multidimensional sectors on sectors (MDSonS)

The method proposed in this paper, is an improvement of the SonS visualization technique [23], called multidimensional sectors on sectors (MDSonS). The visualization method is different from that proposed in [23] due to the need of accommodating the information provided by multidimensional scaling (MDS) in the new visualization. In MDSonS, each cluster is represented by a circle. The area of each circle is proportional to the number of patterns included in each cluster and the distance among the circles is proportional to the distance among clusters. By means of each slice of a circle (pie sectors) and a color bar with the same number of labels as attributes, the existing relationships among centroids' attributes at any hierarchy level can be extracted.

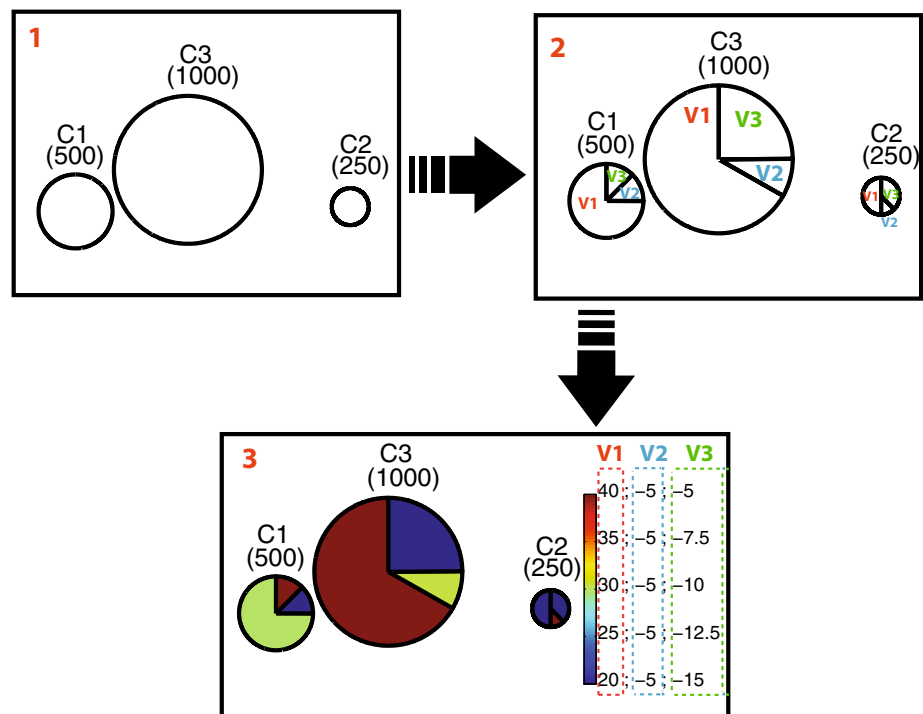
Once the structure of the clustering is decided (number of clusters in each hierarchical level), the visualization graph is produced in three steps for each hierarchy level (see Fig. 2). These three steps should be taken starting from the first hierarchy level and are described as follows:

1. Representation of the different clusters and their size: First of all, as many circles as clusters are drawn. The area of each circle is proportional to the number of patterns included in each cluster and the distance among circles is proportional to the distance among clusters' centroids. The distances among centroids are computed by MDS. MDS produces a representation of the similarity (or dissimilarity) between pairs of objects in a multidimensional space as distances between points of a low-dimensional space [8]. The number of patterns belonging to each cluster is shown within parentheses. In this way, the significance of each cluster and the distance among them are easily recognizable (Fig. 2, top left).
2. Division of the circles depending on the number and the value of attributes: Once the data are divided into clusters and after knowing the size of each one of them, the value

<sup>2</sup> Each variable is scaled between [0, 1] before carrying out the clustering to avoid a biased model. Moreover, the scaling makes the relevance of each variable (represented by the size of the radius) to be independent of its range, that is, the relevance is not higher even though the variable has a higher range. The use of scaled variables guarantees that the radius is relevant to the variable within the cluster.

<sup>3</sup> This is automatically extensible to other measures such as the median, which is a much more adequate prototype measure in the presence of outliers, for instance. Therefore, SonS is not restricted to the use of a particular prototype measure.

**Fig. 2** The three steps followed to create the MDSONS visualization method



of the attributes (or features) for each cluster centroid is analyzed. For this task, each circle, corresponding to each cluster, is divided into several sectors, which correspond to each variable. The first variable is the one that starts with a vertical line at the top middle of the circle, and the rest of the variables appear sequentially counter-clockwise. The arc length of each sector corresponds to the relative value of each variable, with respect to the sum of all of them.<sup>4</sup> In this way, the bigger the arc length of a given variable, the more relevant is the variable. With this method, the relevance of each variable can be identified, within each cluster or within all of them in a straightforward way (Fig. 2, top right).

3. Color coding for identifying the real value of features: Attached to the graph, there is a color bar with the same number of labels as variables (each label for each variable). The first column label gives the value of the first feature, the second column label gives the second feature value and so on. In this way, it is possible to know the exact value of each variable for each cluster centroid (Fig. 2, bottom).

The description for the first hierarchy level can be extended to the rest of the levels. For instance, in the data set analyzed in Fig. 12, for the second hierarchy level it can be observed that from each circle (corresponding to each

cluster) in level 1, a new graph with new values of cluster centroids emerges.

The main advantage of the proposed visualization technique is that it is possible to observe relationships among different variables in the same cluster and relationships among the same variables in different clusters, in the different levels of the hierarchy; but specially, what is remarkable in comparison to SonS is the representation of the distances among clusters' centroids.

### 3 Data sets

This section explains the different data sets used to show the performance of the proposed visualization methods.

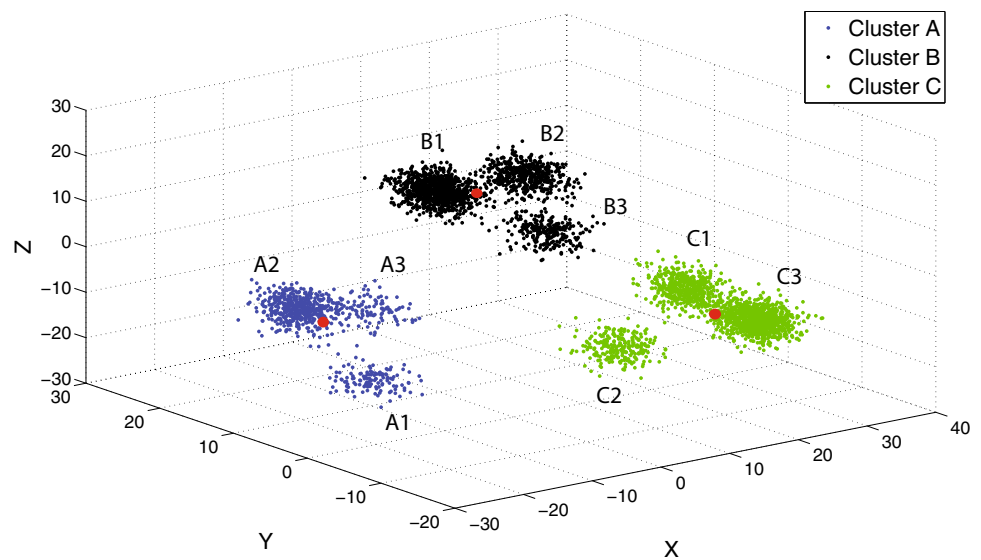
#### 3.1 Synthetic data set

The first data set is a synthetic data set created to show the performance of the proposed visualization method. The data consist of three clouds of points defined by  $X$ ,  $Y$  and  $Z$  coordinates, as shown in Fig. 3. These three clouds of points can be divided into nine, three new clouds for each one of them; thus being a hierarchical structure.

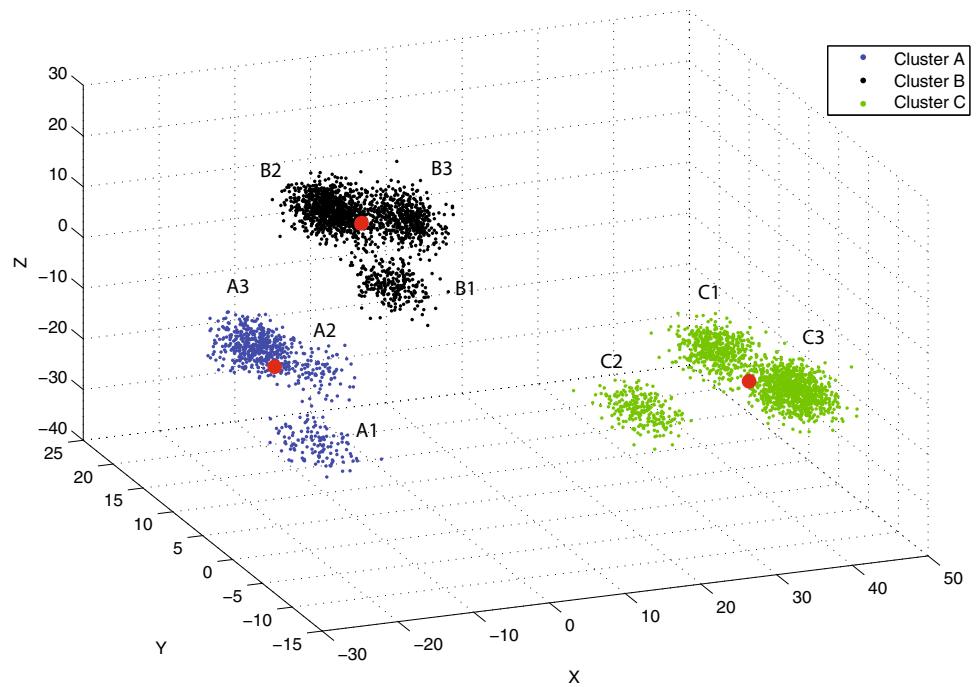
Another variant of this data set was also taken into account (Fig. 4). In this case, the cloud of points corresponding to cluster B was slightly displaced to the left with regard to the previous case. Notice that, while in the first case (Fig. 3) the distances between cluster B and the rest were practically the

<sup>4</sup> As in SonS, each variable is scaled between  $[0, 1]$  before carrying out the clustering to avoid a biased model.

**Fig. 3** Representation of the first synthetic data set variant. The points corresponding to each cluster after the first level (three clusters: A, B, C) are shown in *different colors*. Their centroids are represented with *red dots*. Subclusters at the second level of hierarchy are indicated with a number (1, 2, 3) after the corresponding letter (A, B, C) (color figure online)



**Fig. 4** Representation of the second synthetic data set variant. The points corresponding to each cluster after the first level (three clusters: A, B, C) are shown in *different colors*. Their centroids are represented with *red dots*. Subclusters at the second level of hierarchy are indicated with a number (1, 2, 3) after the corresponding letter (A, B, C) (color figure online)



same, in the second case (Fig. 4) cluster B was significantly closer to cluster A. In this way, the distances among clusters were different; the goal of this variant of the data set is to show the capabilities of MDSONS to represent distances among clusters.

### 3.2 German elections data set

As a real example, a data set of the German parliamentary elections of September 18, 2005 was used. The data, extracted from package *flexclust* of “R” software,<sup>5</sup> consist

of the proportions of “second votes” obtained by the five parties that got elected to the first chamber of the German parliament for each of the 299 electoral districts. The “second votes” are actually more important than the “first votes” because they control the number of seats each party has in parliament. It should be emphasized that the proportions do not sum to unity because parties that did not get elected into parliament were omitted from the data set. Before election day, the German government comprised a coalition of Social Democrats (SPD) and the Green Party (GRUENE); their main opposition consisted of the Conservative Party (Christian Democrats, UNION) and the Liberal Party (FDP). The latter two intended to form a coalition

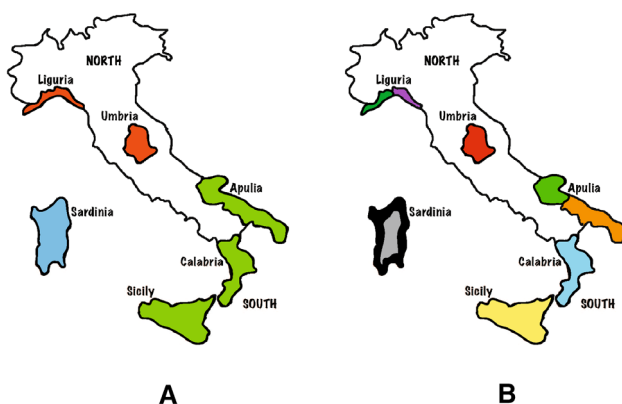
<sup>5</sup> <http://cran.R-project.org>.



after the election if they gained a joint majority, so the two major “sides” during the campaign were SPD+GRUENE versus UNION+FDP. In addition, a new “left-leaning party” (LINKE) canvassed for the first time; this new party contained the descendants of the Communist Party of the former East Germany and some left-wing separatists from the SPD in the former West Germany. This real example has been chosen to show the performance of the presented methods due to the qualitative conclusions that can be drawn from this data set.

### 3.3 Italian olive oil data set

This data set contains information about the percentage composition of fatty acids found in the lipid fraction of Italian olive oils [13]. The data set consists of 572 samples and 10 variables. The training variables are eight fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, eicosenoic) in  $\% \times 100$  (per 10 thousand). The other two variables contain information about the classes. There are two kinds of classes: super-classes that correspond with three regions of Italy: North, South, and the island of Sardinia (see Fig. 5a); and sub-classes corresponding to nine collection areas: three from the Northern region (Umbria, East and West Liguria), four from the South (North and South Apulia, Calabria, and Sicily), and two from the island of Sardinia (inland and coastal Sardinia) (see Fig. 5b). The data set arises from a study to determine the authenticity of olive oil. The goal is to distinguish the oils from different regions and areas in Italy based on their combinations of fatty acids. Just like with the “German election data set”, this real example has been chosen to show the performance of the presented methods due to the qualitative conclusions that can be drawn from this data set.



**Fig. 5** Regions and collection areas of Italy corresponding to the two kinds of classes of the Italian olive oil data set. **a** “Super-classes” or regions of Italy. **b** “Sub-classes” or collections areas of Italy

### 3.4 Iris flower data set

The “Iris flower data set”<sup>6</sup> contains three classes of 50 instances each, where each class refers to a type of iris plant (setosa, versicolor, virginica). One class is linearly separable from the other two; the latter are not linearly separable from each other. The input variables are sepal length, sepal width, petal length, and petal width.

## 4 Results

In this section, the SonS and MSonS visualization methods are applied to different data mining techniques to evaluate their performance using the data sets described in Sect. 3.

### 4.1 SonS applied to hierarchical clustering

In this section, SonS and MDSONS methods are applied to visualize hierarchical clustering. The use of the method, described in Sect. 2.1, can be extended to every hierarchical level found in hierarchical clustering techniques. For the second hierarchy level, for instance, Fig. 7 shows that for each sector (in level 1) emerges a new pie chart with new values of cluster centroids. If in a given data set more than two levels are present, a new pie will emerge from the sectors of the previous level, as occurs when having two levels. It should be emphasized that not always a new pie will emerge from all the sectors, but depends on the selected hierarchy. This method is highly recommended, since it provides a compact visualization of each cluster, making it possible to observe the information of several hierarchy levels simultaneously; thus it is possible to extract information at the different levels of hierarchy.

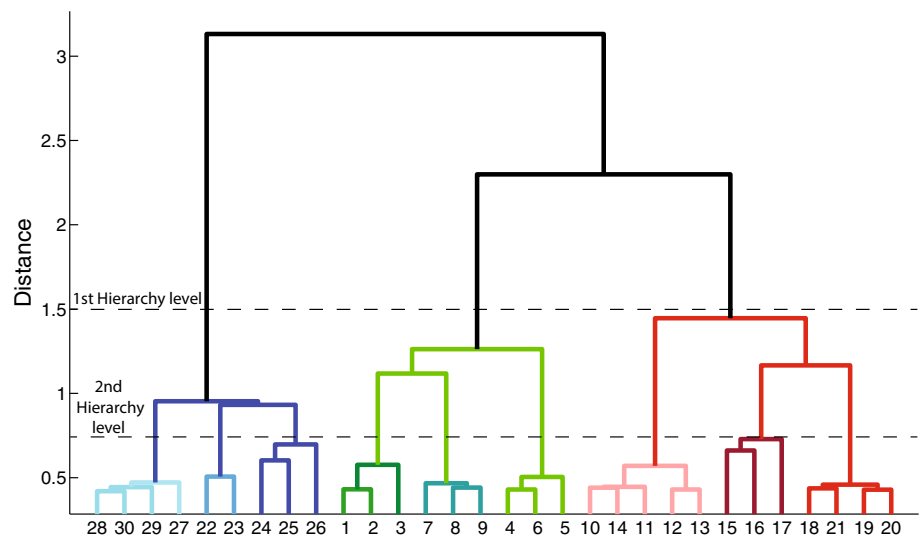
#### 4.1.1 Example 1: synthetic data set

Figure 6 shows the dendrogram corresponding to the data set shown in Fig. 3, which makes it possible to extract the number of clusters visually. In particular, if the dendrogram is analyzed when the distance is 1.75 (higher dashed line), three clusters will be obtained (level 1) which are represented in green, red and blue colors. The second clustering level corresponds to a distance around 0.75 (lower dashed line), in which each former cluster is now divided into three new clusters represented in different shades of the same color as its parent.

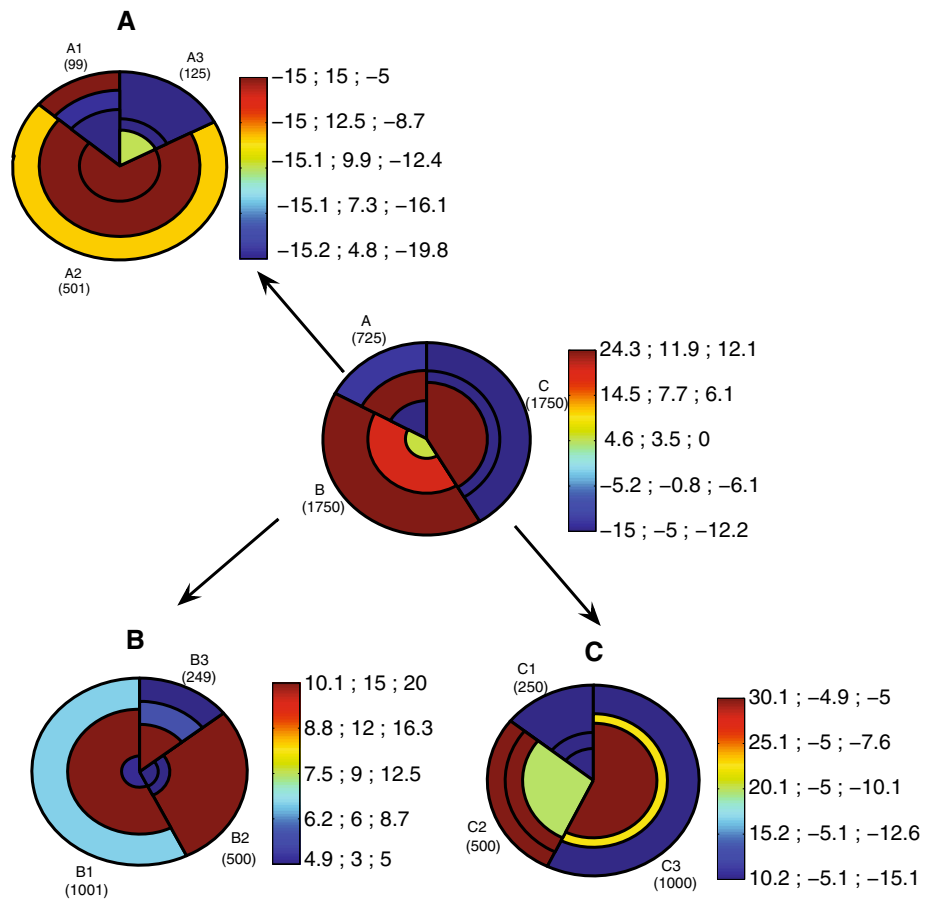
Once the hierarchy is determined and the clusters formed in each level, the clustering is represented with the SonS visualization method (Fig. 7).

<sup>6</sup> <http://archive.ics.uci.edu/ml>.

**Fig. 6** Dendrogram corresponding to the clustering of the first variant of the synthetic data set. Higher dashed line represents the distance of the first hierarchical level. Lower dashed line represents the distance of the second hierarchical level



**Fig. 7** SonS visualization method for the first synthetic data set



The sector corresponding to cluster A, in the first level (center pie chart), has similar radii for the different variables. That means that the centroids' attributes have approximately the same relevance after clustering (in cluster A).

Also in the first hierarchy level and regarding Cluster C, the most relevant variable is the first one ( $X$  coordinate) since it has the largest radius (see inner subsector), and that matches the information shown in Fig. 3. For this cluster, the first

coordinate shows values around 24, whereas the others are about  $-5$  and  $-12$ , respectively. Analyzing the relationships among the same features in different clusters, in the first hierarchy level it can be observed that the last feature (coordinate  $Z$ ) shows a value of approximately 12 in cluster B and around  $-12$  in cluster C (see in both cases, the last column of labels). Those conclusions match the information shown in Fig. 3.

Summarizing, after applying the proposed method to a synthetic data set, it can be seen that, although the dendrogram is an excellent tool to determine the number of clusters in a given hierarchical data set, Sons provides an additional value by making it possible to visualize relationships between centroids' attributes of all clusters at any hierarchical level.

4.1.2 Example 2: German elections

Figure 8 shows the dendrogram obtained for the “German elections” data set. Again, the number of clusters and the hierarchy can be visually established. Analyzing the dendrogram when the distance is 0.2 (higher dashed line), four different clusters (level 1) are obtained, which are represented in red, black, blue and green colors. The second level can be obtained by cutting when the distance is around 0.14 (lower dashed line). In this way, the red and blue clusters are now divided into two new ones represented in different shades of the same color as its parents.

Each electoral district belongs to one of the 16 German federal states. After carrying out the clustering, the state corresponding to each pattern of the different clusters was analyzed. Therefore, the most predominant states can be found to check if each cluster corresponds to different German areas. The conclusion is that the four clusters (first level in the hierarchy) correspond to the four different regions, namely, West Germany (without Saarland), East Germany (without Berlin and without Bayern) together with Saarland, Bayern and, finally, Berlin represented in Fig. 9 in red, blue, green and black, respectively.

Saarland's behavior (located in the southwest of Germany, at the French border) may attract some attention because they voted in a similar way to the eastern states. This is most likely due to the fact that Oskar Lafontaine, one of the two leaders of LINKE, is a former prime minister of Saarland as pointed

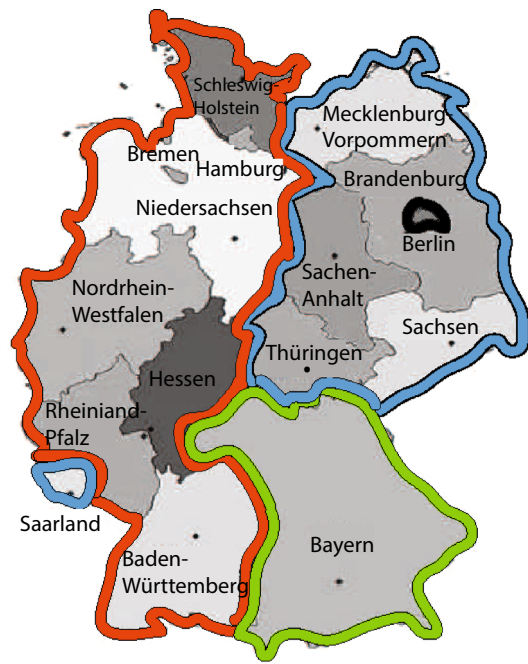
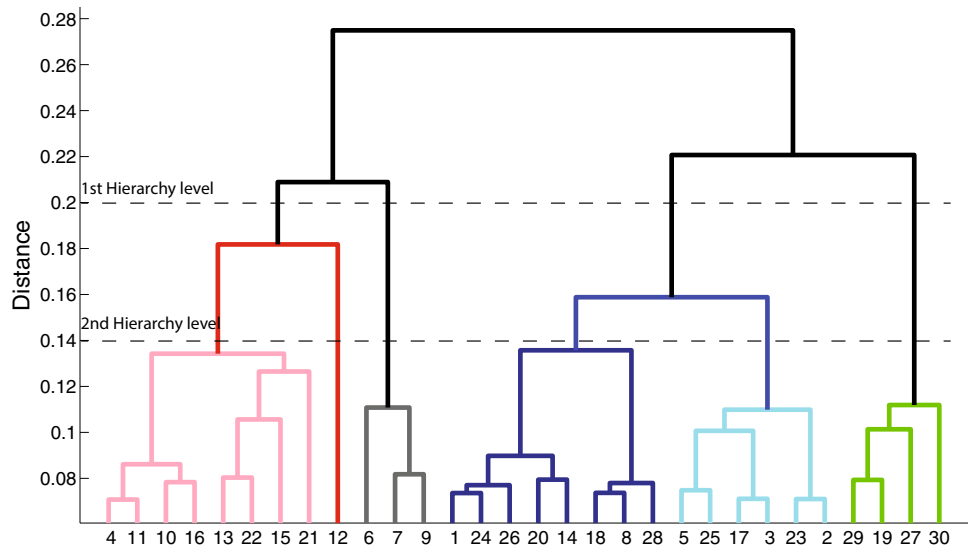


Fig. 9 German map with the 16 different German federal states. The four regions corresponding to the clustering in the first hierarchy level are shown in colors (color figure online)

out in [17]. Another striking state is Berlin, which exhibits very diverse voting behavior and thus spreads over the rest of the clusters except some patterns, which form a different cluster because they are quite far away from other clusters.

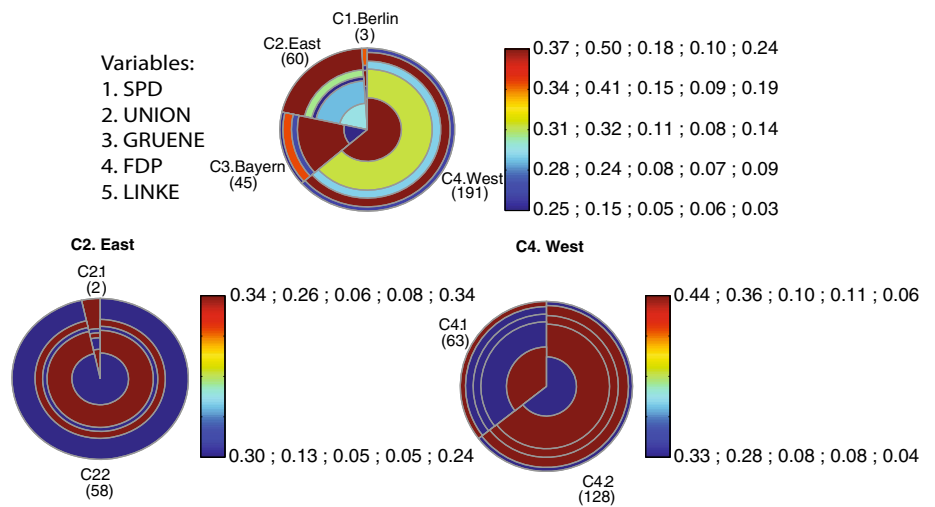
In Fig. 10, the clustering solution for the “German election” data set is represented with the SonS method. Unlike the representation provided by the dendrogram, this visualization is appropriate for this kind of data because a large radius in one subsector, corresponding to one variable, means a large number of votes. In this way, it is easily recognizable

Fig. 8 Dendrogram corresponding to the clustering of the German elections data set. Higher dashed line represents the distance of the first hierarchical level. Lower dashed line represents the distance of the second hierarchical level





**Fig. 10** SonS visualization method for German elections data set



not only which party has the strongest performance, but also the exact value for each party looking at the color bar and its labels.

Focusing on the first level of hierarchy, there are four different clusters corresponding to the geographic areas marked with several colors in Fig. 9. From now on, the red area will be called “West”, the blue one “East”, the green one “Bayern” and the black one “Berlin” as pointed out in Fig. 10. The first cluster represented corresponds to “Berlin”. This cluster has only three patterns, the rest of the patterns of Berlin (nine) spread over the rest of the clusters as mentioned previously. The second cluster corresponds to “East”, in which the parties with the strongest performance are SPD (0.3), UNION (0.25) and LINKE (0.24). Notice that this is the only cluster where LINKE has an important relevance. This makes sense, since LINKE party contained the descendants of the Communist Party of the former East Germany and some left-wing separatists from the SPD in the former West Germany. In the third cluster, corresponding to “Bayern”, the winner party is UNION. According to the fourth cluster, the two main parties are SPD and UNION having a little bit more support SPD than UNION (0.37 and 0.32, respectively) which are in opposite wings.

In the next hierarchy level, each one of cluster 2 (“East”) and cluster 4 (“West”) are divided into two new clusters (C2.1, C2.2, C4.1 and C4.2, respectively). Notice that the first new cluster extracted from the cluster corresponding to “East” (C2.1) has very similar values in its variables as the cluster “Berlin”. That is because the two patterns which are members of this new cluster actually belong to the cluster “Berlin”. The point is that in the first level, the algorithm was not able to distinguish it, because these two patterns were located between the clusters “East” and “Berlin”, but in the second level the difference is more evident; thus showing up the importance of the hierarchical approach. The second new cluster (C2.2) corresponds totally with “East” (blue area in

Fig. 9) where the parties with the strongest performance are SPD, UNION and LINKE, as mentioned before.

According to the division of Cluster 4 (“West”), the two newly formed clusters are similar to each other. In both clusters, the first and second features are the most relevant ones, whereas the last three have a lower significance. Although each variable presents a maximum value for one cluster and a minimum one for the other, actually the difference between the maximum and minimum values is not very significant (as shown in the color bar labels). Therefore, in deeper hierarchy levels, the clustering division is done depending on which party has the strongest performance. That is, the first cluster (C4.1) corresponds to the case when SPD is the most supported party (0.44), and the second cluster (C4.2) corresponds to the case when the party with the biggest support is UNION (0.36). The first new cluster (C4.1) corresponds to the south part of West Germany together with the northern region Schleswig-Holstein, and the second new cluster (C4.2) corresponds to the northern part of West Germany, except Schleswig-Holstein. The region Nordrhein-Westfalen has approximately the same number of patterns in each cluster.

The main conclusions and ideas extracted from this data set have been contrasted with [10,20]. Moreover, new information and new ideas have been extracted by the proposed visualization tool, which could not be obtained with other classical visualization tools.

#### 4.2 MDSONS applied to in hierarchical clustering

In this section, the MDSONS method is used to visualize hierarchical clustering, as in the previous section, to highlight the differences between SonS and MDSONS. The use of the method can be extended to every hierarchical level found in hierarchical clustering techniques, as occurred in the SonS case.

#### 4.2.1 Example 1: synthetic data set

Figure 11 shows the corresponding dendrogram that makes it possible to extract the number of clusters visually. In particular, if the dendrogram is analyzed when the distance is 1.5 (higher dashed line), three clusters (level 1) will be obtained, which are represented in blue, green and red colors. The second clustering level corresponds to a distance of around 0.75 (lower dashed line), in which each former cluster is now divided into three new clusters represented in different shades of the same color as its parent.

Once the hierarchy is determined and the clusters formed in each level, the clustering is represented by means of MDSonS (Fig. 12). Figure 12 shows the clustering for the two different hierarchies. The first one is shown in the center of the figure without the frame. From each one of the clusters in the first hierarchy level, three new circles emerge. Those circles enclosed in frames correspond to the second hierarchy level. Focusing on the first hierarchy, three different clusters appear in Fig. 12, one small (cluster A) and two bigger clusters (clusters B and C). It can also be observed that there are two clusters (A and B) that are closer to each other than to the other one (cluster C); this conclusion matches the representation shown in Fig. 4.

Focusing on each cluster, the different variables of cluster A (delimited by the sectors) have a similar area. That means that the centroids' attributes have the same relevance after clustering. Exactly, they take the values  $[-15, 12, -16]$  as it can be observed with the color bar and checked in Fig. 4. Cluster B has a similar area for the second and third variables ( $Y$  and  $Z$  coordinates), and the area of the sector corresponding to the first variable is smaller (less relevant). In particular, they take the values  $[-5, 10, 12]$ . However, in cluster C, the area of the first sector (first feature) is the biggest one by far. As it can be observed with the color bar, the exact values

of the different features are  $[34, -5, -12]$ . All these conclusions agree with the representation of Fig. 4 regarding the relevance of the different variables to define each cluster.

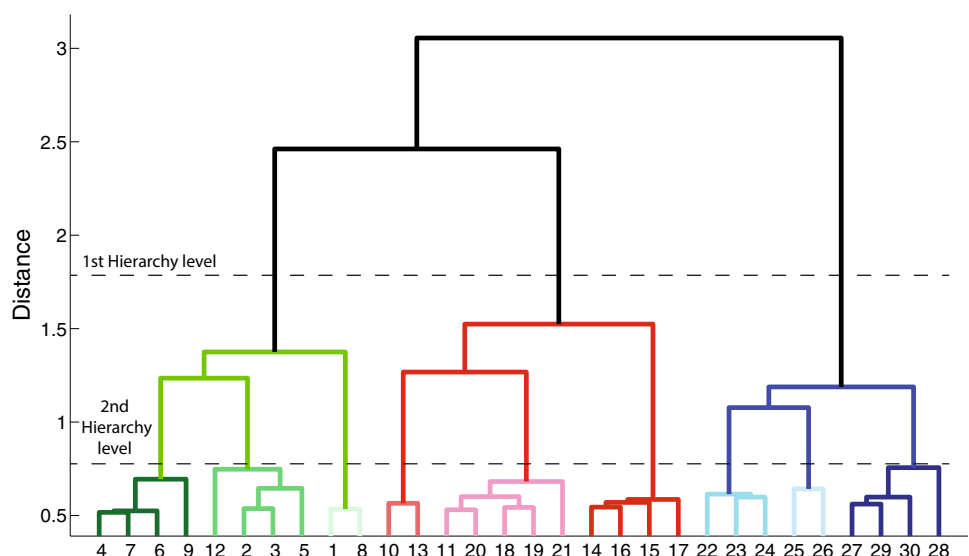
Analyzing relationships among the same features in different clusters, in the first hierarchy level, it can be observed that the first feature ( $X$  coordinate) presents low values in cluster B and A ( $-5$  and  $-15$ , respectively), whereas cluster C presents a high value (34), very different from the other two values. This is the reason why clusters A and B are far away from cluster C. However, to see which feature is the most relevant for distinguishing between cluster A and B, it can be seen that the last feature ( $Z$  coordinate) shows a value of approximately 12 in cluster B and about  $-16$  in cluster A (see in both cases the last column of labels), whereas the other features are quite similar. Thus, as it can be observed in Fig. 4, the cloud of points corresponding to cluster B is higher than the cloud of points corresponding to cluster A (variable  $Z$  corresponds to the height). Notice that, in fact, the third feature can distinguish not only cluster A and B, but can distinguish between cluster B and the other two (high values for cluster B and low values for the rest). To distinguish among subclusters, a similar procedure can be carried out for the next hierarchy level.

#### 4.2.2 Example 2: German elections

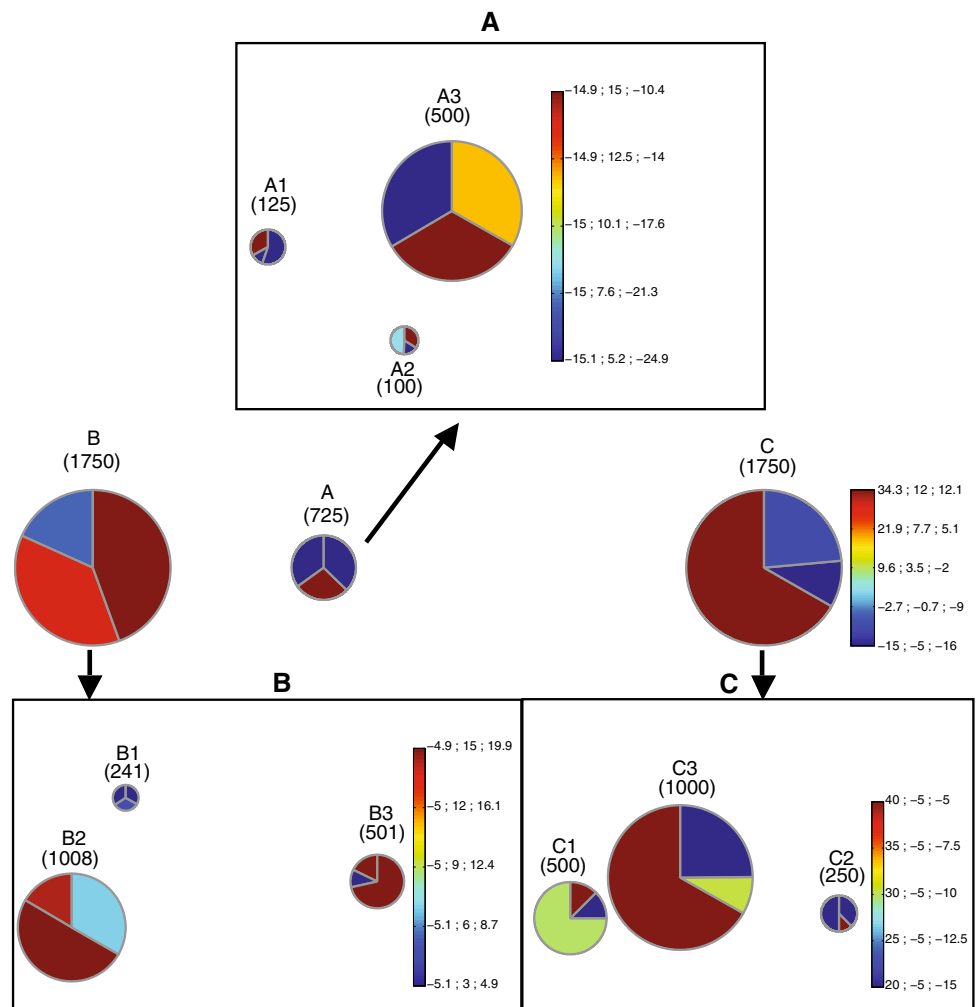
Figure 10 shows the clustering achieved by the proposed method. As in SonS case, this visualization is appropriate for this kind of data because a long arc length corresponding to each variable means a large number of votes.

From Fig. 13, similar conclusions to those seen in Sect. 4.1.2 can be extracted. Summarizing, regarding the cluster “West”, the two main parties are SPD and UNION having a little bit more support SPD than UNION. In cluster “Bayern”, the parties with the biggest support are SPD

**Fig. 11** Dendrogram corresponding to the clustering of the second variant of the synthetic data set. Higher dashed line represents the distance of the first hierarchical level. Lower dashed line represents the distance of the second hierarchical level



**Fig. 12** MDSONS representation for the synthetic data set



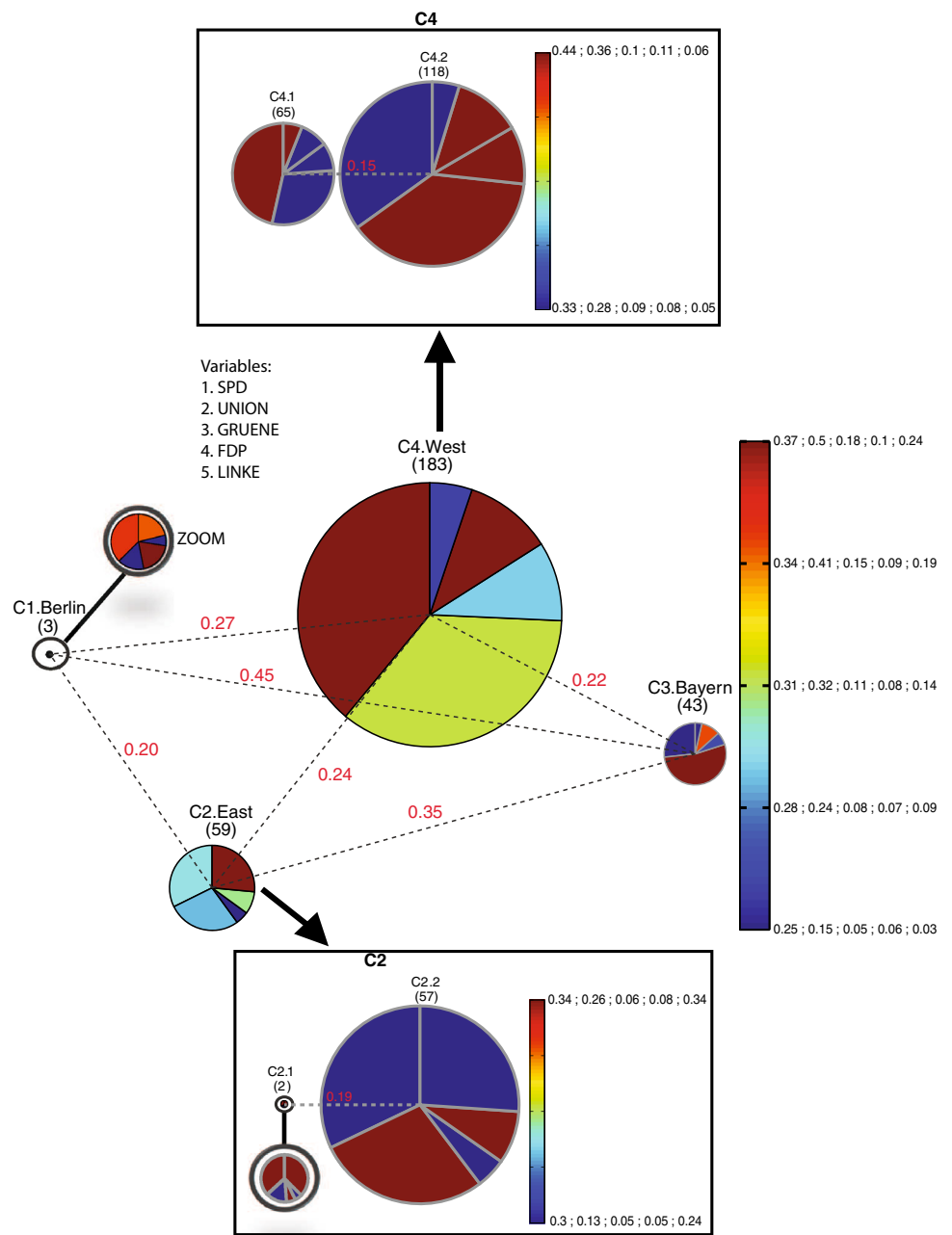
(0.25) and UNION (0.5) having more support than the latter. The cluster “Berlin” has only three patterns; the rest of the Berlin patterns (nine) spread over the rest of the clusters. Thus, these three patterns corresponding to cluster “Berlin” cannot be considered as a global behavioral pattern of Berlin. Actually, these patterns are related to the communist part of Berlin (East). Anyway, it should be pointed out that for these three patterns, the party with the strongest performance is SPD (0.35); and also LINKE has a significant performance (0.19) compared with the clusters commented on previously (see Fig. 13). Finally, in cluster “East”, the parties with the strongest performance are SPD (0.3), UNION (0.25) and LINKE (0.24). As commented previously, LINKE has a significant performance compared with the other clusters.

In addition to conclusions of the features within each cluster, information about the relationship of features in the different clusters can also be extracted. For example, SPD presents the biggest support in cluster “West” (0.37), UNION in cluster “Bayern”, GRUENE in cluster “Berlin” (0.18), FDP in cluster “West” (0.1) and LINKE in cluster “East”(0.24).

Moreover, new conclusions and ideas can be extracted when using MDSONS method, which are related to the information provided by the representation of the distances in this method. This information provides great utility to the method and on certain scenarios may be very important. It also helps to contrast hypotheses. For example, in SonS, it is known by intuition that the clusters Bayern and West are similar, since these two clusters are the only ones that have the largest support in the first two parties, receiving the other three parties much lower support, as mentioned previously. However, representing the distances among clusters, it can be proved that the most similar cluster to “Bayern” is “West”. This assumption can be proved by checking all the distances (red numbers) between the cluster “Bayern” and the other clusters, since the minimum distance appears between the two mentioned clusters.

Moreover, it can also be proved in the hypothesis, extracted from SonS, that clusters “Berlin” and “East” are very similar because these two clusters are the only ones where LINKE has an important relevance. This information, which is not available in the SonS method, provides an

**Fig. 13** MDSONS visualization method for German elections data set



essential aid to the problem understanding. Actually, checking all the distances among all clusters, it can be proved that these are the two most similar (closest) clusters among all. In addition, the result of the method is more intuitive to analyze, and since it presents information in a less compact design, it allows a neat representation of a larger number of variables.

In the next hierarchy level (pies enclosed in frames), also similar conclusions to those seen in Sect. 4.1.2 can be extracted. No more relevant information can be extracted in this hierarchical level apart from the distances between the clusters.

### 4.3 SonS applied to GHSOM algorithm

The self-organizing map (SOM) [18] is one of the most popular visualization tools. The SOM is a neural model that carries out a low-dimensional visualization of patterns defined in N-dimensional data sets. The main advantage of GHSOM over hierarchical clustering is that in the former, the hierarchical structure is found “automatically” (actually, tuning some parameters); and in the latter, the user should visually establish the data structure using a dendrogram. Two of the main limitations of SOM are the static architecture of the model and the difficulty in obtaining hierarchical relation-

ships visually [11]. Since hierarchical models can extract more information from a data set, SOM has been modified in several ways to deal with hierarchical frameworks, with the growing hierarchical self-organizing map (GHSOM) being specially remarkable [11, 12, 25]. The main problem of GHSOM is that it is not possible to visualize simultaneously the data information in each level. In this section, SonS visualization technique is used for visualizing the results obtained from the GHSOM algorithm to circumvent that drawback, since it allows a simultaneous and compact visualization of the different hierarchy levels, and it also enables the extraction of knowledge in terms of relationships among variables. This is not possible using other classical visualizations. From now on, and for this particular application, the several sectors will correspond to neurons instead of clusters.

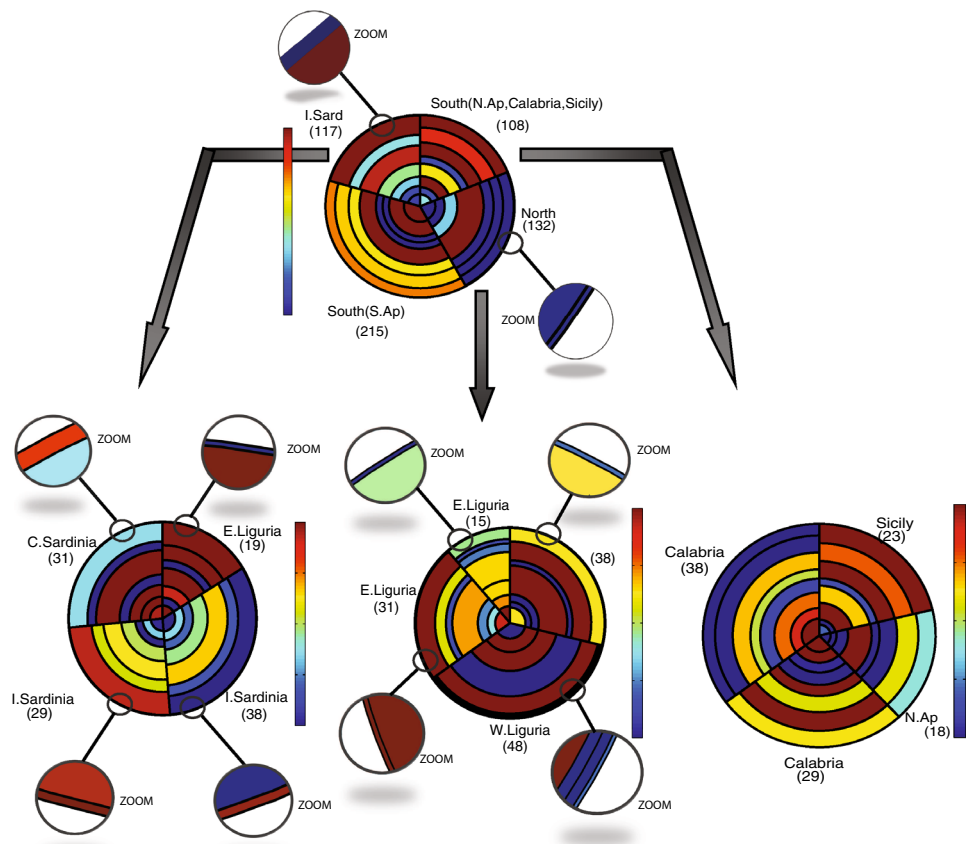
### 4.3.1 Italian olive oils

After training the mentioned data set with the GHSOM algorithm, two hierarchy levels were produced; the first level started the training to four neurons (four sectors in Fig. 14). After this, the predominant region for each neuron, in the first hierarchy level, is checked. For the first hierarchy level (top pie chart, Fig. 14), there is one sector corresponding

to the Island of Sardinia, another to the South (specifically South Apulia), another to the North and finally another corresponding to the South again (specifically, North Apulia, Calabria and Sicily). As it can be observed, the radius of the last variable, for some sectors, is very small. This fact makes difficult the visualization of the value in the mentioned variable. Because of this, a zoom of the image was carried out. The labels of the color bars have been removed for better visualization; only the color bars are shown to indicate a qualitative value. Notice that although a given color may be very similar for two different variables, it does not mean a very similar value for the variables since each variable has its own range of values.

For distinguishing the oils from the different regions, in the first hierarchy level, the most important variable is the eighth (outer subsector in the circle), because it takes a maximum value for one region and a minimum one for the other. If the mentioned variable is high, it shows that the oil belongs to the South and if it is low it belongs to either the North or the Island of Sardinia. For distinguishing between North and Sardinia, the fifth and the seventh variables play a relevant role (high values for Sardinia and low for the North). Summarizing, for the first hierarchical level, a number of rules can be drawn from visual inspection of the generated graph:

**Fig. 14** MDSonS visualization method for Italian olive oil data set





- North if V8 ↑↑
- I. Sardinia if V8 ↓↓; V7 and V5 ↑↑
- South if V8 ↓↓; V7 and V5 ↓↓ .

In the next hierarchy level, three new GHSonS graphs were found which emerged from the previous sectors corresponding to the Island of Sardinia, North and finally the South, specifically the sector which represented North Apulia, Calabria and Sicily (Fig. 14). To distinguish among the sub-classes in this hierarchy, a similar procedure can be carried out, which consists of checking which variables take maximum values for one region and minimum for others. Thus, an additional advantage of the SonS is that it enables making a feature selection visually, since it is possible to separate the different classes using fewer features than the ones presented in the problem. The conclusions extracted from this data set regarding the second hierarchical level are put forward in [22].

#### 4.4 SonS applied to CART models

Classification tree analysis is one of the main techniques used in data mining [7, 14], but there is still a lack of visualization methods to support this tool. Therefore, graphical procedures should be developed to improve the interpretation of the solutions provided by these models. The sectors on sectors (SonS) visualization method is used to visualize the input space in the terminal nodes of the classification tree. Once the classification tree is built, each one of the subsectors obtained by SonS, corresponding to each variable, varies its radius to represent the relevance of each variable in each cluster; but for the sake of simplicity in the visualization, this step has been omitted.

For classification problems, in which we focus on this section, the goal is to find a tree where the terminal tree nodes are relatively “pure”, i.e., contain observations that (almost) all belong to the same category or class. However,

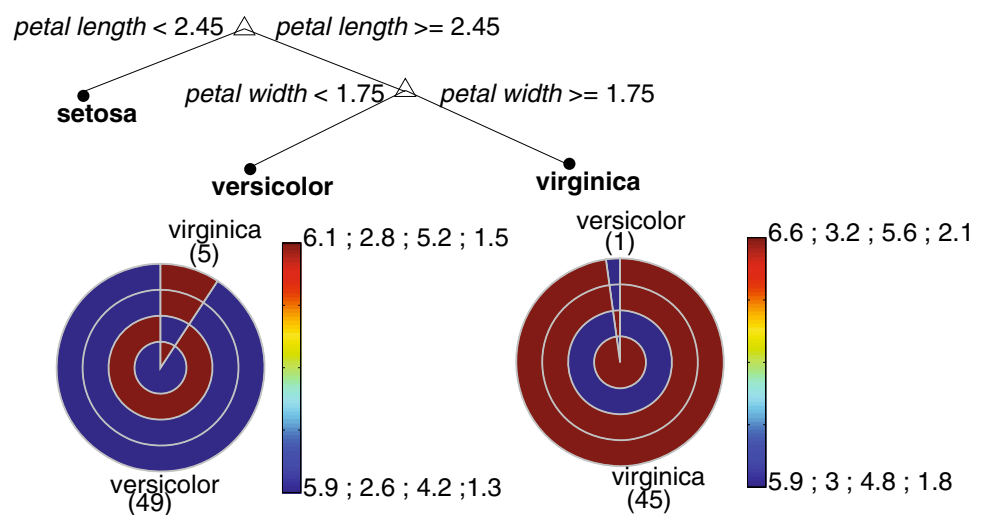
it does not always happen that the terminal nodes are pure. Because of this, we propose a visualization tool in which we can obtain visually the number of patterns belonging to each class presented in each terminal node as well as to extract the maximum information by representing the input data for each class presented in each terminal node. The proposed graphical procedure helps to simplify the interpretation even for complex trees and helps to interpret the different data found in the terminal nodes.

##### 4.4.1 Example 1: Iris flower data set

Figure 15 shows the classification tree obtained for the “Iris flower data set”. In each terminal node, the SonS graph has been drawn unless all the patterns included in the terminal node belong to the same class (as occurs in the terminal node labeled as “Setosa”).

As shown in Fig. 15, the most important variable to separate between setosa class and others is the third variable (petal length). If it takes a value lower than 2.45, it means that the input pattern will belong to setosa, and it will belong to any of the other two classes otherwise. The classification tree indicates that to differentiate between versicolor and virginica classes, the last variable (Petal Width) must be taken into account. If this variable is lower than 1.75, the input pattern will belong to the versicolor class; and if it is greater than or equal to 1.75, the pattern will belong to the virginica class. However, as extracted from the SonS graph, in the terminal node corresponding to versicolor, there are five patterns belonging to virginica class. Looking at the last variable (outer subsector), which distinguishes between the versicolor and virginica classes, along with the last column of the color bar, it can be seen that the sector corresponding to virginica takes a value of 1.5, whereas the versicolor class takes a value of 1.3. Thus, we could say that versicolor class corresponds to a value lower than 1.5 instead of 1.75, as

**Fig. 15** Classification tree obtained for the “Iris flower data set” with the SonS graph in the terminal nodes



the classification tree indicates. In the terminal node corresponding to virginica, it can be observed that just one pattern belonging to versicolor class has been erroneously included.

#### 4.4.2 Example 2: Italian olive oil data set

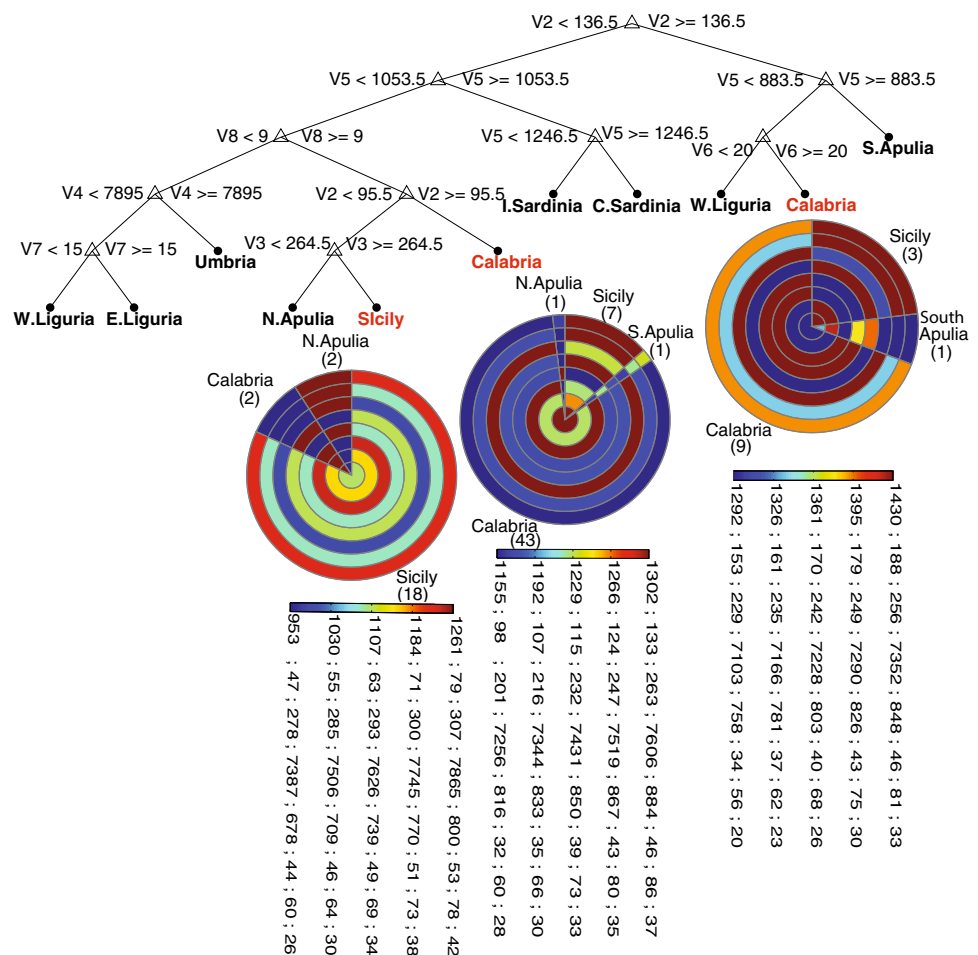
Figure 16 shows the classification tree obtained for the “Italian olive oil data set”. To extract the most significant conclusions, special attention will be paid to those terminal nodes where there is a considerable number of patterns erroneously included (more than 20 %). Therefore, Fig. 16 only shows the SonS graphs that follow this rule. The first SonS graph that attracts attention is that corresponding to the first Calabria terminal node (1st chart starting from the right) because more than the 30 % of the patterns are wrong. This chart has one sector corresponding to Calabria (9 patterns), another one corresponding to Sicily (3 patterns) and finally one corresponding to South Apulia (1 pattern). To distinguish among these groups of patterns, new decision rules must be established. For example, Calabria and Sicily are easily distinguishable by means of the fourth variable because Calabria presents a maximum value (7352), indicated by a deep red color, and Sicily presents a minimum value (7103), indicated

by blue color. Notice that other variables also present maximum values in one of these regions, and minimum values for the other one, but the fourth variable presents the widest range (in relative values) between the maximum and minimum values. Therefore, the procedure to follow is to choose an intermediate value (7227.5) to separate between these two regions. Hence, the new rule is that if the fourth variable has a value lower than 7227.5, the patterns will belong to Sicily; and if it is greater than or equal to 7227.5, the patterns will belong to Calabria. For distinguishing South Apulia from the others regions, a similar procedure can be followed, but since only one pattern is affected, an ad hoc definition of a rule might be pointless.

Another terminal node that presents a large number of patterns erroneously included is that corresponding to the second Calabria terminal node (middle chart). In this case, low values of the eighth and seventh variables separate Calabria from Sicily.

The last terminal node to consider is that corresponding to Sicily (3rd chart starting from the right). In this case, the fifth variable has relevance in distinguishing among the regions in this terminal node. Notice that, if this variable takes low values (blue) the olive oil will belong to North Apulia, if it

**Fig. 16** Classification tree obtained for the “Italian olive oil” data set with the SonS graph in the terminal nodes



takes intermediate values (green) the olive oil will belong to Sicily and finally if it takes high values (red) the olive oil will belong to Calabria. This is the only variable that distinguishes among the three classes included in this terminal node. It is worth mentioning that a deeper tree (which would have less generalization ability) could separate these classes. Our approach allows to extract, visually, this separation as well as gain knowledge about the problem while preserving the generalization capabilities of the tree. Anyway, in this example, the classification tree takes its role quite well because, although it makes mistakes in particular final nodes, the patterns erroneously included actually belong to the same super-class; therefore, the SonS can be seen as an improvement or a fine-tuning.

Another advantage of this method is that it could also be used to build shallow classification trees. That means that, it may be no longer necessary to produce very deep trees because the same conclusions can be extracted visually (starting in previous nodes). That is, if the nodes of the tree are removed at some level, it will be possible to establish the rules visually without needing to build deep trees. Moreover, the SonS graphs could be used in other nodes (not only in terminal nodes) to obtain visual information about how the classification tree evolves.

Another interesting use of the original SonS method, in classification trees, could be to carry out a clustering algorithm with the data included in each terminal node and visualize the result. In this case, visual information about the different clusters obtained in each terminal node would be extracted.

## 5 Conclusions

This paper presents a novel visualization technique called sectors on sectors (SonS), and a modified version called multidimensional sectors on sectors (MDSonS), for several data mining algorithms. The MDSonS method makes use of multidimensional scaling to solve a drawback of SonS, namely, the lack of representing distances between pairs of clusters. The performance of these visualization tools has been shown by means of real and synthetic data sets, demonstrating its applicability.

Firstly, SonS and MDSonS methods have shown to be very useful tools when visualizing hierarchical clustering, since it is possible to infer relationships among features, clusters and different levels of the hierarchy. However, MDSonS entails a new improvement over the sectors on sectors (SonS) method, which consists of carrying out a multidimensional scaling (MDS) of the centroids and drawing each pie chart, corresponding to each cluster, in the location provided by MDS. MDS provides centroid coordinates in 2D, taking into account the distances among all clusters.

Secondly, SonS applied in growing hierarchical self-organizing maps (GHSOM) has demonstrated to be a useful alternative visualization tool for this algorithm, since it allows a simultaneous and compact visualization of the different hierarchy levels that is not provided by the classical GHSOM. It is also a useful tool when visualizing hierarchical data, since it is possible to infer relationships among features, neurons and different levels of the hierarchy, demonstrating its capacity for extracting information. This fact is complicated or not possible in classical visualization.

Finally, SonS applied in CART models helps to extract knowledge and to obtain a better understanding even for complex trees, since it represents the input data information, for the classes associated in each terminal node (although the approach can also be applied to non-terminal nodes), of a classification tree. This method is capable of providing visual information of the patterns belonging to a terminal node in the decision tree, so that it will be possible to extract information about the values of their variables and information about the patterns erroneously included. Therefore, new decision rules can be established visually to distinguish them. Other advantages and uses of the SonS applied in CART models are described as follows:

- Another advantage of this method is that it could also be used to build shallow classification trees; deep trees might not be necessary because the same conclusions can be extracted visually (starting in previous nodes). That is, if the nodes of the tree are removed at some level, it will be possible to establish the rules visually without needing to build deep trees.
- As previously mentioned, the SonS graphs could be used in other nodes (not only in terminal nodes) to obtain visual information about how the classification tree evolves.
- Another interesting use of the original SonS method, in classification trees, could be to carry out a clustering algorithm with the data included in each terminal node and visualize the result. In this case, visual information about the different clusters obtained in each terminal node would be extracted.

As far as the authors know, there are no previous works addressing the issue of hierarchical clustering visualization in terms of obtaining information about the values of clusters' centroids and relationship with the hierarchical arrangement provided by the clustering algorithm; nor there is literature about methods capable of providing visual information of the patterns belonging to a terminal node in the decision tree, to the author's knowledge. Therefore, the work represents a novelty and an important advance in the field of data visualization and knowledge extraction, since the performance of the presented visualization methods has been shown by

means of different examples (synthetics and real) demonstrating its applicability in several data mining techniques.

Our ongoing research is focused on including more information about the centroids, such as the standard deviation, covariance matrix or about the variances of the two principal components of each cluster. In this way, information about the clusters' shape would be added to the visualization method. Moreover, we are working on other clustering algorithms (using Mahalanobis metric).

Another possible way of improving the proposed visualization methods is to come up with strategies that may improve its scalability in terms of the number of attributes, that is, to improve the performance of the presented methods when dealing with a large number of features. At this stage, SonS method is able to represent about eight or ten attributes as maximum to have a good interpretability. A number of attributes larger than this would make it difficult to interpret the method. Regarding MDSONS, this method is easier to interpret for an untrained reader, due to the fact that SonS graph is somewhat overlaid with sectors and sub-sectors. Therefore, MDSONS method is able to represent more features than the SonS with an acceptable interpretability. The way to improve the scalability of both methods lies in applying manifolds or dimensionality reduction techniques in such data sets with a significantly higher number of dimensions along with the proposed method to tackle the mentioned improvement. Regarding the scalability in terms of the number of patterns, there is no limitation on using this method in large-scale data sets, since they represent the centroids provided by the used algorithm. Therefore, the limitation would lie in the machine learning or data mining algorithm used rather than in the visualization method.

## References

- Andrews, K.: Visual exploration of large hierarchies with information pyramids. In: Information visualisation, international conference on, p. 793 (2002)
- Andrews, K., Heidegger, H.: Information slices: visualising and exploring large hierarchies using cascading, semi-circular discs. In: Information visualization, 1998. Proceedings IEEE Symposium on (1998)
- Andrews, K., Kasanicka, J.: A comparative study of four hierarchy browsers using the hierarchical visualisation testing environment (hvte). In: Information visualization, 2007. IV '07. 11th international conference, pp. 81–86 (2007)
- Ankerst, M., Ester, M., Kriegel, H.P.: Towards an effective cooperation of the user and the computer for classification. In: KDD, pp. 179–188 (2000)
- Baehrecke, E., Dang, N., Babaria, K., Shneiderman, B.: Visualization and analysis of microarray and gene ontology data with treemaps. *BMC Bioinform.* **5**(1), 84 (2004)
- Beaudoin, L., Parent, M.A., Vroomen, L.: Cheops: a compact explorer for complex hierarchies. In: Visualization '96. Proceedings., pp. 87–92 (1996)
- Berthold, M., Hands D.J. (eds): Intelligent data analysis, 2nd edn. Springer (2002)
- Borg, I., Groenen, P.J.: Modern multidimensional scaling, theory and applications. Springer (2005)
- Card, S.K., Mackinlay, J.D., Shneiderman, B. (eds.): Readings in information visualization: using vision to think. Morgan Kaufmann Publishers Inc., San Francisco (1999)
- Chen, C.H., Hardle, W., Unwin, A.: Handbook of data visualization (Springer Handbooks of Computational Statistics). Springer-Verlag TELOS, Santa Clara (2008)
- Dittenbach, M., Merkl, D., Rauber, A.: The growing hierarchical self-organizing map. In: neural networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS international joint conference on, vol. 6, pp. 15–19 (2000)
- Dittenbach, M., Rauber, A., Merkl, D.: Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing* **48**(1), 199–216(18) (2002)
- Forina, M., Armanino, C., Lanteri, S., Tiscornia, E.: Classification of olive oils from their fatty acid composition. In: Food Research and Data Analysis. Applied Science Publishers, London, pp. 189–214 (1983)
- Hastie, T., Tibshirani, R., Friedman, J.H.: The elements of statistical learning, corrected edn. Springer (2003)
- Holten, D.: Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *Vis Comput Graph IEEE Trans* **12**(5), 741–748 (2006)
- Jeong, C.S., Pang, A.: Reconfigurable disc trees for visualizing large hierarchical information space. In: Information visualization, 1998. Proceedings. IEEE Symposium on, vol. 149, pp. 19–25 (1998)
- Kesselman, M., Krieger, J., Joseph, W.: Introduction to Comparative politics: political challenges and changing agendas. Wadsworth (2009)
- Kohonen, T.: Self-organizing maps, 3rd edn. Springer, Berlin Heidelberg (2001)
- Kreuseler, M., Schumann, H.: Information visualization using a new focus+context technique in combination with dynamic clustering of information space. In: NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM international conference on Information and knowledge management, pp. 1–5. ACM, New York (1999)
- Leisch, F.: Neighborhood graphs, stripes and shadow plots for cluster visualization. *Stat. Comput.* (2009)
- Makanju, A., Brooks, S., Zincir-Heywood, A., Milios, E.: Logview: visualizing event log clusters. In: Privacy, security and trust, 2008. PST '08. Sixth annual conference on, pp. 99–108 (2008)
- Martínez-Martínez, J.M., Escandell-Montero, P., Soria-Olivas, E., Martín-Guerrero, J.D., Gómez-Sanchis, J., Vila-Francés, J.: Growing hierarchical sectors on sectors. In: European symposium on artificial neural networks, computational intelligence and machine learning, 2011. ESANN '11, pp. 239–244 (2011)
- Martínez-Martínez, J.M., Escandell-Montero, P., Soria-Olivas, E., Martín-Guerrero, J.D., Martínez-Sober, M., Gómez-Sanchis, J.: Sectors on sectors (sons): a new hierarchical clustering visualization tool. In: Computational intelligence and data mining, 2001. CIDM '11. IEEE Symposium on, pp. 304–309 (2011)
- McConnell, P., Johnson, K., Lin, S.: Applications of Tree-Maps to hierarchical biological data. *Bioinformatics* **18**(9), 1278 (2002)
- Miikkulainen, R.: Script recognition with hierarchical feature maps. *Connect. Sci.* **2**, 83–101 (1990)
- Munzner, T.: H3: laying out large directed graphs in 3d hyperbolic space. In: Information visualization, 1997. Proceedings., IEEE symposium on, pp. 2–10 (1997)
- Reingold, E., Tilford, J.: Tidier drawings of trees. *Softw Eng IEEE Trans* **SE-7**(2), 223–228 (1981)

28. Robertson, G.G., Mackinlay, J.D., Card, S.K.: Cone trees: animated 3d visualizations of hierarchical information. In: CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 189–194. ACM, New York (1991)
29. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**(1), 53–65 (1987)
30. Shneiderman, B.: Tree visualization with tree-maps: a 2-d space-filling approach. *ACM Trans. Graph.* **11**, 92–99 (1991)
31. Spence, R.: Information visualization. ACM Press Bks, Addison-Wesley (2001)
32. Theodoridis, S., Koutroumbas, K.: Pattern recognition, 4th edn. Academic Press (2008)
33. Thomas, J.J., Tajudin, D.A.: Visualizing the examination timetabling data using clustering method and treemaps. In: Proceedings of the 2nd IMT-GT regional conference on mathematics, statistics and applications (2006)
34. Ware, C.: Information visualization: perception for design. Interactive Technologies. Elsevier Science and Technology (2012)