ReClus - A Tool to Explore the Output of Clustering Algorithms

Angel R. Martinez
Naval Surface Warfare Center, Dahlgren, Virginia 22448
Wendy L. Martinez
Office of Naval Research, Arlington, Virginia 22217

KEY WORDS: Rectangle plots, treemaps, agglomerative model-based clustering, *k*-means clustering, cluster visualization, dendrogram.

Abstract

The tool ReClus is introduced. This tool is used to explore the output of clustering algorithms, such as model-based clustering, agglomerative clustering and *k*-means. ReClus follows in the tradition of the treemap displays of Johnson and Shneiderman and rectangle plots of Wills, where rectangles are used to represent clustering results. However, differently from these approaches, ReClus is not limited to agglomerative clustering methods. Additionally, ReClus provides options for the display of class labels or case numbers, as well as color-coded class membership probability.

1. Introduction

In this section, we give a brief explanation of the motivation for a new way of visualizing clusters, especially when the true class membership is known. The goal of this visualization is to provide a way to explore and to assess the results of the clustering algorithm. This novel visualization method is called ReClus. In the next section, we include a description of the visualization methods that predate ReClus. This is followed by an explanation of ReClus, along with several illustrative examples. The visualization methods discussed in this paper were implemented in MATLAB.

The motivation for ReClus came from an experiment in document clustering, where the true topic labels were known [Martinez, 2002]. A new way of encoding text documents called the bigram proximity matrix (BPM) was introduced in that work. The BPM is a non-symmetric matrix that captures the number of word pairs in a document. The BPM is a square matrix whose column and row headings are the alphabetically ordered entries of the lexicon (a listing of unique words in the corpus). Each matrix element in the BPM is the number of times word *i* appears immediately before word *j*. The size of the BPM is determined by the size of the lexicon created from the unique occurrences of the words in the text. It was the goal of Martinez [2002] to show that the BPM representation of the semantic content preserves enough unique features to be semantically separable from BPMs of other thematically unrelated documents.

A corpus of documents called the Topic Detection and Tracking (TDT) Pilot Corpus (Linguistic Data Consortium, Philadelphia, PA) was used in the experiments performed in Martinez [2002]. The TDT corpus is comprised of close to 16,000 newscasts collected from July 1, 1994 to June 30, 1995 from the Reuters newswire service and CNN broadcast news transcripts. A set of 25 events are defined in the TDT. Each of the newscasts is classified as either belonging to the topic (*Yes*), not belonging to the topic (*No*) or only partly belonging (*Brief*). A total of 503 newscasts were chosen, encompassing 16 of the 25 events discussed in the TDT. See Table 1 for a list of topics. The 503 documents chosen contain only the *Yes* or *No* flags. This choice stemmed from the need to demonstrate that the BPM captures enough meaning to make a correct or incorrect topic classification choice.

Table 1. List of 16 Topics

Topic Number	Topic Description	Number of Documents
4	Cessna on the White House	14
5	Clinic Murders (Salvi)	38
6	Comet into Jupiter	45
8	Death of N. Korean Leader	35
9	DNA in OJ Trial	29
11	Hall's Copter in N. Korea	75
12	Humble, TX, Flooding	16
13	Justice-to-be Breyer	8
15	Kobe, Japan Quake	50
16	Lost in Iraq	30
17	NYC Subway Bombing	24
18	Oklahoma City Bombing	76
21	Serbians Down F-16	16
22	Serbs Violate Bihac	19
24	US Air 427 Crash	16
25	WTC Bombing Trial	12

One of the ways this was demonstrated was through the use of model-based clustering [Fraley & Raftery, 2002]. By applying some clustering method to discover topics and then comparing these to the known topic groups, we could ascertain whether the BPMs encode enough semantic information to group documents such that documents in each cluster have similar meaning.

Model-based clustering is a probability density estimation approach to clustering, where a finite mixture model is first fit to

the data. A finite mixture model assumes the underlying distribution can be modeled as a finite sum of weighted component densities [Everitt & Hand, 1981]. In model-based clustering, the component densities are usually assumed to be multivariate normals. Once the model is obtained (see Fraley and Raftery [2002] for more details on this procedure), we hypothesize that each term in the model corresponds to a cluster and can be used as a template to group observations. In other words, the center of the cluster is given by the mean for the component density, and the shape of the cluster is governed by the covariance matrix. Observations are assigned to clusters based on this model. That is, the probability that the observation belongs to each component density is calculated, and the observation is grouped with the component that has the highest of these probabilities.

The problem then is how to assess the results of the clustering when the true class label is known. The cluster membership 'number' is arbitrary and cannot be mapped to the true class membership labels, in most cases. We might try to visualize the clusters via scatterplots or parallel coordinate plots [Wegman and Carr, 1993]. We would have to encode the discovered clusters using colors or symbols and try to visually compare the same type of plot where the encoding is accomplished using the true class labels. This would be an inexact and tedious process. The ReClus method of visualizing clusters was developed to address this problem. The method is suitable for very high-dimensional data.

2. Antecedents to ReClus

In this section, we present existing methods for visualizing the results from clustering. These include dendrograms, treemaps and rectangle plots. All of these methods are used for the output from agglomerative clustering, so they cannot be applied to model-based clustering or *k*-means clustering. For more information on different clustering methods, see Everitt [1993]. (Agglomerative clustering is a hierarchical approach, where each observation starts as a single cluster. The two closest clusters are merged at each step of the algorithm until all observations are in a single cluster.)

The output of agglomerative clustering can be viewed in a tree or dendrogram. A dendrogram can be shown vertically or horizontally, but it essentially consists of many U-shaped lines that show the hierarchical structure of the clustering algorithm. We show a dendrogram for a simple example of a data set that has 3 known and well-separated clusters. There are $n = 40$ observations in 4 dimensions; a scatterplot matrix of the data set is shown in Figure 1. The MATLAB Statistics Toolbox will produce dendrograms, and an example of one is shown in Figure 2. The vertical axis in a dendrogram represents distance. If we cut the tree at different values along the vertical axis, then we get different partitions or clusters. For example, we can cut the tree in Figure 2 at 4 to obtain 3 clusters. If we use a cutoff of around 3, then we have 5 clusters. The reader should also note that the number of leaf nodes on the horizontal axis is less than 40. Plotting all points in a large data set can lead to overplotting in a dendrogram, degrading its readability and usefulness. In MATLAB, the default is to plot 30 nodes, so some of the nodes shown in the dendrogram correspond to several observations, and the leaf node numbers do not necessarily correspond to an actual observation.

Johnson and Shneiderman [1991] noted that the dendrogram does not efficiently use the existing display space, so they proposed a space-filling display of hierarchical information called treemaps. The original application and motivation for treemaps was to show the directory structure on hard drives, but it is also suitable for showing the results from agglomerative clustering or other information that can be arranged in a tree-like or hierarchical structure.

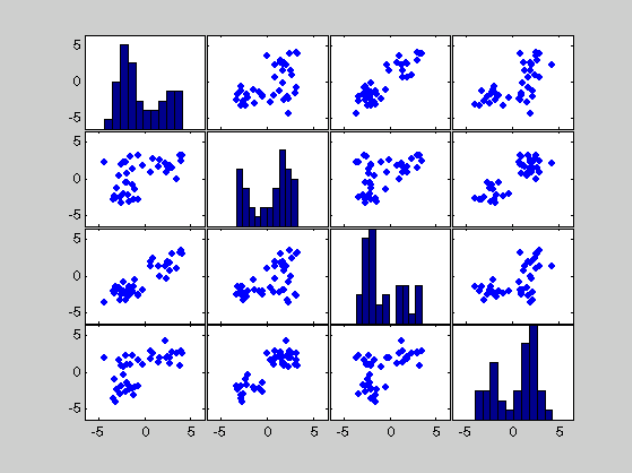

Figure 1. Scatterplot of a simulated data set with 3 clusters and 40 observations.

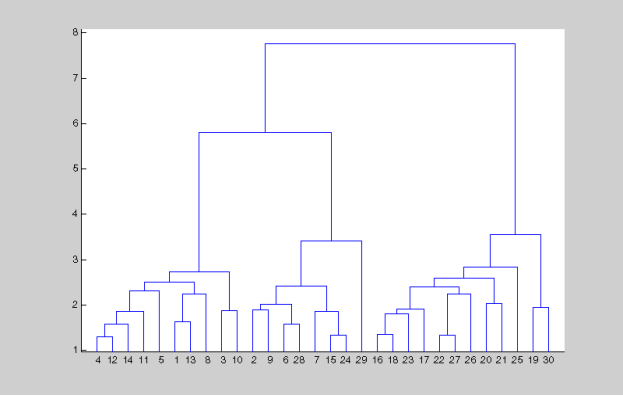

Figure 2. Dendrogram for the data set shown in Figure 1. This dendrogram was based on agglomerative clustering, where the Euclidean distance was used, along with average linkage.

The treemap method displays this information in a series of nested rectangles (or ellipses). The parent rectangle (or root of the tree) is given by the entire display area. The treemap is obtained by recursively subdividing this parent rectangle, where the size of each sub-rectangle is proportional to the size of the node. The rectangles are further subdivided horizontally, vertically, horizontally, etc., until a given leaf configuration is obtained. The area of each rectan-

gle is proportional to an attribute of interest such as directory size or number of observations in the node.

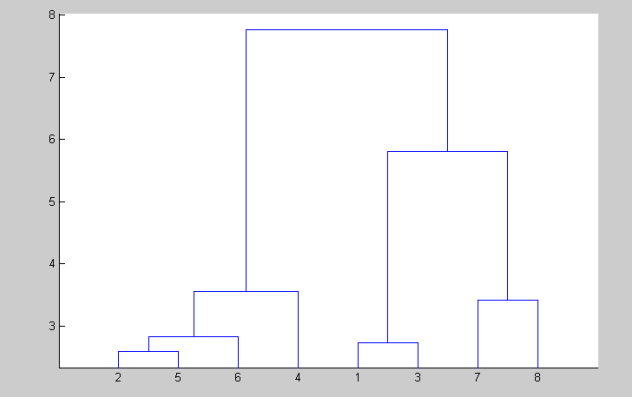

Figure 3. This is a dendrogram of the same data set. We requested 8 leaf nodes for display.

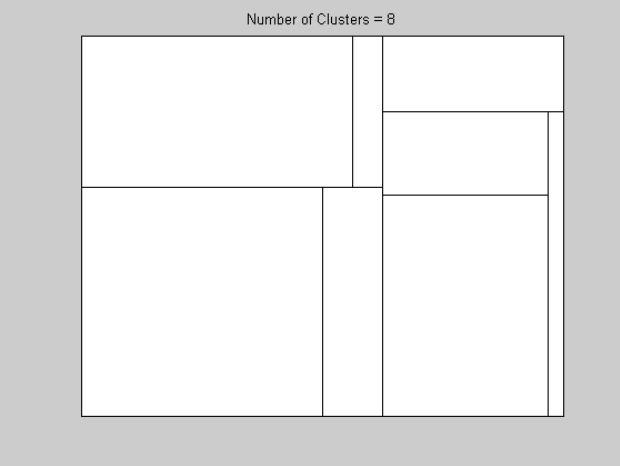

Figure 4. This is a treemap display for the 8 leaf nodes shown in Figure 3.

We show another dendrogram of the same data set in Figure 3, where we requested 8 leaf nodes in the display. The corresponding treemap display is given in Figure 4. The first vertical split cuts the parent rectangle into two pieces: nodes 1, 3, 7, 8 on the left and nodes 2, 5, 6, 4 on the right. It should be noted that our implementation of this in MATLAB will only accommodate the binary splits/merges of the agglomerative clustering. The treemap display can be applied in the more general case where any number of splits can take place at levels of the tree.

Extensions to the treemap algorithm include cushion treemaps [van Wijk and Wetering, 1999] and squarified treemaps [Bruls, Huizing and van Wijk, 2000].

Recall that in the dendrogram shown in Figure 2, the user can specify a distance (the value along the vertical axis), and different clusters are obtained depending on what value is specified. To display as a treemap, the user must specify the number of clusters rather than the cutoff point. If the user wants to explore other cluster configurations by specifying a different number of clusters, then the display is re-drawn. In the treemap display, there is no measure of distance associated with the clusters as there is in the dendrogram. A further drawback to the treemap method is the lack of information about the original data, because the rectangles are just given labels or left blank. It would be useful to know what cases are clustered where.

To address some of the issues, Wills [1998] developed the rectangle visualization method based on the treemap display. This method also works with the output of hierarchical (e.g., agglomerative) clustering, but displays the points as glyphs. The layout of the glyphs is determined by the hierarchical structure given by the clustering. The rectangle plots of Wills split the rectangles along the longest side, rather than alternating vertical and horizontal splits as in treemap. They keep splitting until it reaches a leaf node or until the cutoff distance is reached. If a rectangle does not have to be split because it reaches this cutoff point, but there is more than one observation in the rectangle, the algorithm continues to split until it reaches a leaf node. However, it does not draw the rectangles. It uses this information to determine the layout of the points as glyphs, where each point is now in its own rectangle. The advantage to this method is that other configurations (i.e., number of clusters) can be shown without re-displaying the glyphs; only the rectangle boundaries are re-drawn.

The rectangle method of Wills is suitable for linking and brushing applications, where one can highlight an observation in one plot (e.g., a scatterplot) and see the same observation highlighted in another (e.g., a rectangle plot). A disadvantage is that some of the nesting structure seen in treemaps *might* be lost in the rectangle display.

Rectangle plots are shown in Figures 5 and 6. We show the clusters that are obtained when 8 clusters are chosen from the dendrogram shown in Figure 2. Note that all 40 observations are shown here, whereas only 30 leaves are displayed in Figure 2, which is why there is not a one-to-one correspondence between the observations in each display. We can construct a rectangle plot for all 40 observations (i.e., choose to display 40 clusters), and this is given in Figure 6. Notice that the position of the glyphs has not changed from the other rectangle plot; only the rectangle boundaries have been redrawn.

3. ReClus

Another disadvantage of the treemap and rectangle method is that they are both suitable for displaying the results of agglomerative clustering only. In many cases, the analyst might want to use some other clustering method such as model-based clustering or *k*-means and view the results. ReClus is a way to extend the ideas of the rectangle method to display configurations or groups from other clustering methods.

As in the previous methods, ReClus uses the entire display area as the parent rectangle. This is then partitioned into rectangles, where the area is proportional to the number of observations that belong to that cluster. The pseudo-code is given here.

Step 0. Set up the parent rectangle. Note that we will split on the longer side of the rectangle according to the proportion of observations that are in each group.
Step 1. Find all of the points in each cluster and the corresponding proportion.

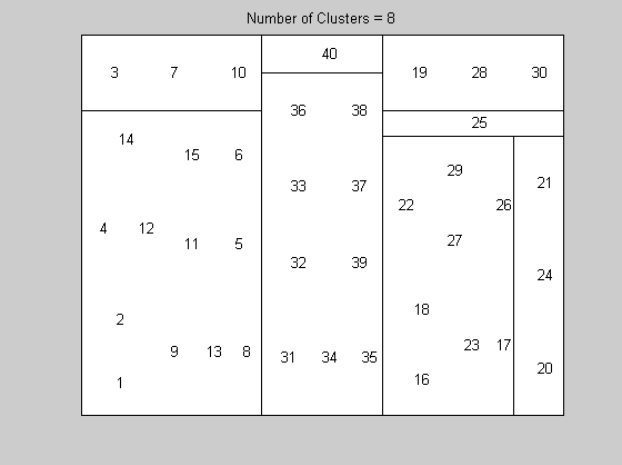

Figure 5. Rectangle plot showing all 40 observations as case numbers and 8 clusters for the results obtained from agglomerative clustering.

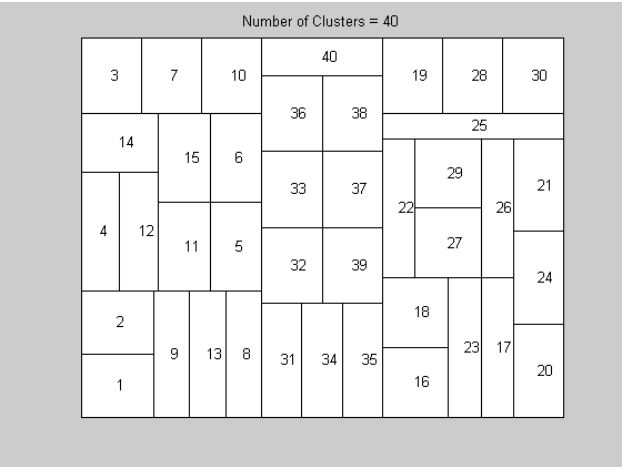

Figure 6. Rectangle plot showing observations in each individual cluster.

Step 2. Order the proportions in ascending order.
Step 3. Partition the proportions into 2 groups. If there are an odd number of clusters, then put more of the clusters into the 'left/lower' group.
Step 4. Based on the total proportion in each group, split the longer side of the parent rectangle. We now have two children. Note that we have to normalize the proportions based on the parent.
Step 5. Repeat steps 3 through 4 until all rectangles represent only one cluster.
Step 6. Find the observations in each cluster and plot, either as the case label or the true class label (if known).

ReClus has several views. The first is to plot the observations using the case label as the glyph. We show this in Figure 7 for clustering based on the agglomerative clustering (in Figure 2). In this case, we specified the number of clusters as 8, and MATLAB (the Statistics Toolbox) provides cluster labels for each of the observations, which is required by the ReClus procedure. If we know the true class labels, then we can show those numbers instead. This will give us a visual picture of how jumbled the clusters are according to the true class information.

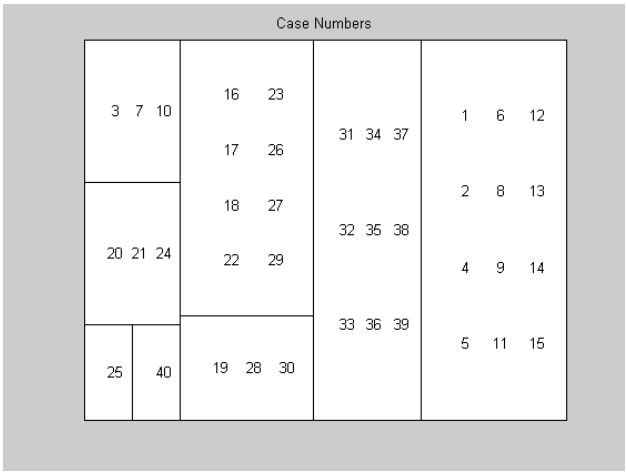

Figure 7. ReClus plot for 8 clusters using the results from the agglomerative clustering. Note that in this case, we are not trying to show the hierarchical relationships between the clusters.

For an example of this type of layout, we return to the problem of document clustering that was discussed in Section 1. In Figure 8, we show the results of using model-based clustering. We see that each rectangle contains a listing of the observations that are clustered together, but they are displayed using the true class label. There is also an optional capability of indicating the probability that an observation belongs to the cluster when we have that available (as we do with model-based clustering). This is indicated by the color of the glyph. To make things somewhat easier to read, we can set a threshold, such that higher probabilities are shown in bold black type. Thus, only the observations that have lower cluster membership probabilities have color. We can see from Figure 8 that those clusters that seem to be very jumbled with observations belonging to several different topics tend to have observations with lower probability of belonging to the cluster.

In most cases, we would assume that a mix of 2 or more classes in a rectangle is an undesirable result. However, in the case of our document clustering application, a mix could point to a justifiable confusion. For example, in the rectangles whose cases 8 and 11 are mixed, both sets of documents are about North Korea. Also, topics 17 and 18 are sometimes mixed, and both topics deal with bombing, the Oklahoma City bombing and the NY subway bombing. The same happens a few times with cases 21 and 22; both report on two different aspects of the Serbian conflict.

Also, note the two clusters where class 6 had two pure rectangles filled with its cases. This raises the issue of latent

classes or sub-topics. A reading of the documents involved does show two different foci. The main subject of the set is the crash of fragments of the comet Shoemaker-Levy onto the surface of Jupiter. One group in the set emphasizes background information about the comet as well as the fact that the space shuttle is in orbit ready to observe what is yet to take place. The second group's focus is predominantly on the event already taking place and observations of the phenomenon.

As mentioned earlier classes 8 and 11 appeared mixed in many of the experiments. Topic 8 and topic 11 both deal with North Korea, one regarding the death of Kim Il Sung and the other the crash of the American helicopter in North Korean territory. An additional interesting fact is that most of the time these rectangles contain cases from 8, of which two are mixed with 11, and one (almost purely 11) is slightly mixed with 8. As is the case with class 6, this may imply the existence of latent classes in groups 8 and 11. A quick reading of the newscasts for topic 8 seems to show three major themes discussed over the background of Kim Il Sung's death and the probable succession of his son Kim Jong-il. The three latent topics are: (1) US and North Korea relations and nuclear issues talks; (2) North Korea and South Korea relations; and (3) North Korea's nuclear plants.

4. Summary

To summarize, the treemap and rectangle plots can be used to visualize hierarchical clustering. The ReClus plot is used for other clustering methods, such as model-based clustering, *k*-means or any method where cluster membership has been assigned. Both ReClus and rectangle plots are suitable for linking and brushing because they indicate the actual observations that are grouped together. The area of the rectangles in the treemap display and ReClus are proportional to the size of the clusters. Finally, we note that all of these methods for cluster visualization are not affected by the dimensionality of the data. However, in general, they are not suitable for massive data sets. Software (except for the dendrogram) will be made available on the StatLib website:

http://lib.stat.cmu.edu/

References

Bruls, D.M., C. Huizing, J.J. van Wijk. 2000. *Squarified Treemaps*. In: W. de Leeuw, R. van Liere (eds.), Data Visualization 2000, *Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization*, Springer, p. 33-42.

Everitt, Brian S., 1993. *Cluster Analysis*, Edward Arnold Publishers, New York.

Everitt, Brian S. and D. J. Hand. 1981. *Finite Mixture Distributions*, London: Chapman and Hall.

Fraley, C. and A. Raftery. 2002. "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, 97:611-631.

Martinez, A. R. 2002. *A Framework for the Representation of Semantics*, Ph.D. Dissertation, George Mason University.

Johnson, B. and B. Shneiderman. 1991. 'Treemaps: a space-filling approach to the visualization of hierarchical information structures," *Proceedings of the 2nd International IEEE Visualization Conference*, pp. 284-291.

Shneiderman, B. 1992. 'Tree visualization with tree-maps: 2-D space-filling approach," *ACM Transactions on Graphics*, 11, pp. 92-99.

Wegman, E. J. and D. B. Carr. 1993. "Statistical graphics and visualization," with D. B. Carr, in *Handbook of Statistics 9: Computational Statistics*, (Rao, C. R., ed.), Amsterdam: North Holland, pp. 857-958.

Wijk, J.J. van, H. van de Wetering. 1999. "Cushion Treemaps," in: G. Wills, D. Keim (eds.), *Proceedings IEEE Symposium on Information Visualization (InfoVis'99)*, pp. 73-78.

Wills, G. J. 1998. "An interactive view for hierarchical clustering," *Proceedings IEEE Symposium on Information Visualization*, pp 26-31.

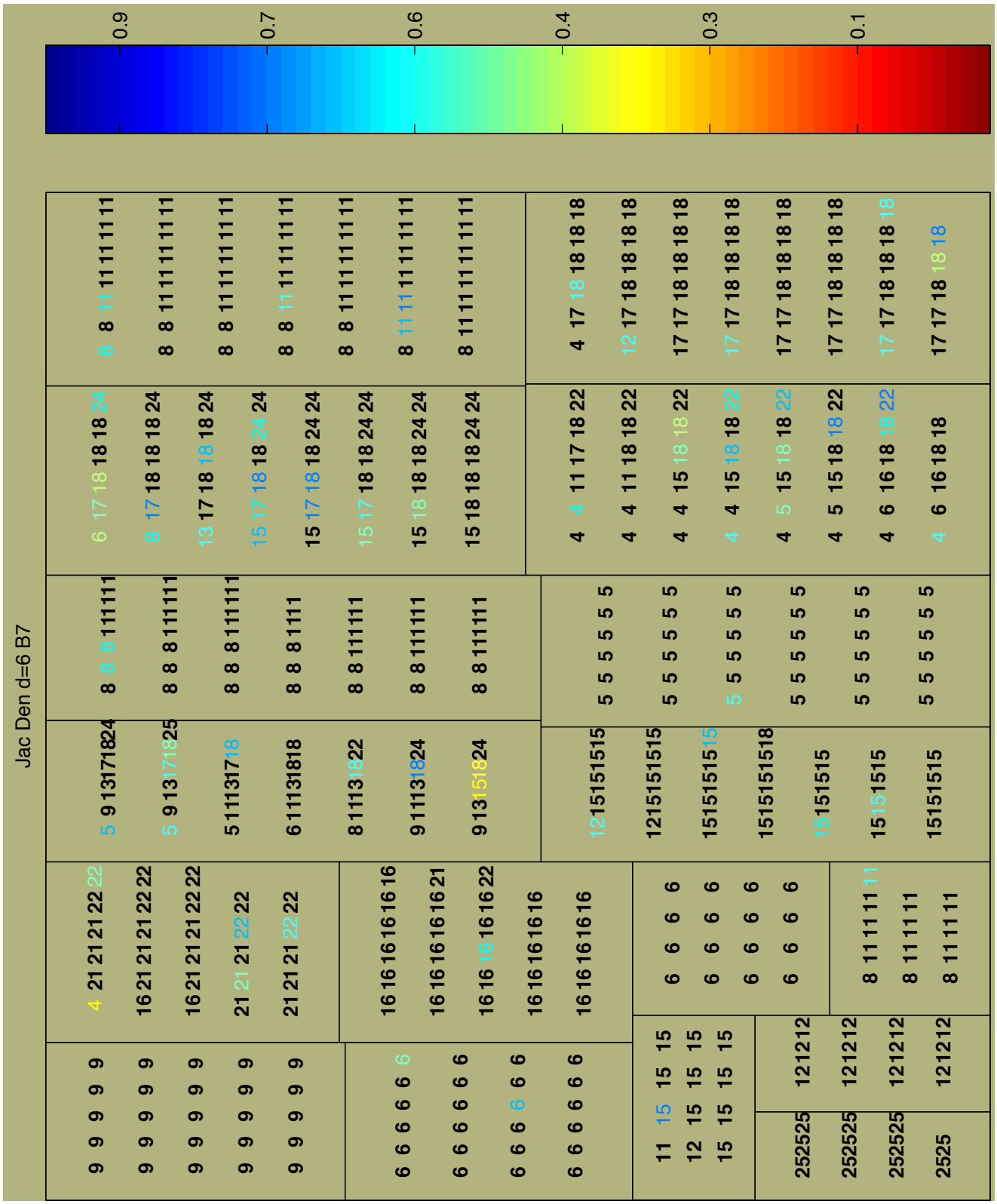

Figure 8. ReClus plot showing probability that the observation belongs to the cluster (by the color) and the true topic label.