Visualization of Single Clusters

Frank Rehm[1], Frank Klawonn[2] and Rudolf Kruse[3]

[1] German Aerospace Center, Braunschweig, Germany
frank.rehm@dlr.de
[2] University of Applied Sciences Braunschweig/Wolfenbüttel, Germany
f.klawonn@fh-wolfenbuettel.de
[3] Otto-von-Guericke-University of Magdeburg, Germany
kruse@iws.cs.uni-magdeburg.de

Abstract. Evaluation of clustering partitions is a crucial step in data processing. A multitude of measures exists, which - unfortunately - give for one data set various results. In this paper we present a visualization technique to visualize single clusters of high-dimensional data. Our method maps single clusters to the plane trying to preserve membership degrees that describe a data point's gradual membership to a certain cluster. The resulting scatter plot illustrates separation of the respecting cluster and the need of additional prototypes as well. Since clusters will be visualized individually, additional prototypes can be added locally where they are needed.

Keywords: Clustering, Visualization, Cluster Validity

1 Introduction

Partitioning data sets is an important task in many domains such as costumer segmentation, organizing textual information or gene expression analysis. The agenda behind this process is knowledge discovery via abstraction over an appropriate data representation. However, despite the availability of powerful analytical methods, the evaluation of resulting models turns out to be non-trivial.

Common prototype-based clustering algorithms, such as k-means or fuzzy c-means, minimize an objective function [3]. As a matter of fact, clustering algorithms always fit the clusters to the data, even if the cluster structure is not adequate for the problem. Thus, the quality of a partition cannot be verified meaningfully by the value of the objective function. Therefore, many validity measures are developed to analyze the adequateness of clustering results [4, 5, 12, 14, 16, 17].

Most of these measures evaluate the partitioning by means of analyzing the fuzzy partition matrix (for fuzzy clustering) or analyzing compactness and separation of clusters considering variance, dispersion, homogeneity or other derivatives drawn from the partitions resulting in a single value, which is of course associated with some loss of information. Primarily, global validity measures cannot give hints which part of the data should be explored more in detail.

VAT, Visual Assessment of Cluster Tendency, is a tool to visualize pairwise dissimilarity information of objects $X = \{x_1, \ldots, x_n\}$ as a square image with n^2 pixels. VAT reorders the data objects so that the image highlights potential cluster structures. As a modification of this, bigVAT allows the visualization for larger data sets [13]. VCV, Visual Cluster Validity, is related to VAT, but takes the inter-datum distances into account that come from partitioning the data set [11].

FUZZSAM, an approach to visualize fuzzy partitions based on Sammon's Mapping is presented in [1]. The proposed tool maps the cluster centres and the data on an arbitrary low dimensional feature space such that the distances between the clusters and the data points will be preserved.

Recently, two visualization methods using the information of a fuzzy clustering partition were presented in [14]. One method arranges the membership degrees to the respective cluster over the distances to the according prototype vector. Each cluster is represented in a single plot. The other method represents a whole fuzzy partition by plotting the highest membership degree to each feature vector over the corresponding second highest membership degree. Such a plot gives an overall impression of a partition.

We propose in this paper a visualization technique to visualize single clusters of high-dimensional data. Our method maps a single cluster to the plane trying to preserve the fuzzy membership degrees that are directly obtained from fuzzy clustering or subsequently derived from crisp partitions. In the following section we briefly recall fuzzy clustering. Section 3 describes the visualization technique. In section 4 we will give some practical details to the implementation. In section 5, we illustrate our method on an artificial data set and on a benchmark data set as well. Finally, we conclude with section 6.

2   Fuzzy Clustering

Generally, fuzzy clustering algorithms partition a data set into several clusters as minimizing an objective function J that describes the sum of weighted distances d_{ij} between c prototypes vectors v_i and n feature vectors x_j of the feature space $\mathbb{R}^p$

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}. \tag{1}$$

Prototype vectors represent the respecting clusters by their location in the cluster's centre. By means of the fuzzifier $m \in (1, \infty]$ one can control how much the clusters overlap. Widely overlapping clusters, which can be obtained with high values for m, will be reflected by almost equal membership degrees u_{ij} to every cluster. Rather crisp partitions can be found with small values for m. Usually, the fuzzifier is set to $m = 2$. In order to avoid the trivial solution assigning no data to any cluster by setting all u_{ij} to zero and avoiding empty clusters, the

following constraints are required:

$$u_{ij} \in [0,1] \quad 1 \le i \le c, \ 1 \le j \le n \tag{2}$$

$$\sum_{i=1}^{c} u_{ij} = 1 \quad 1 \le j \le n \tag{3}$$

$$0 < \sum_{j=1}^{n} u_{ij} < n \quad 1 \le i \le c. \tag{4}$$

When the Euclidian norm

$$d_{ij} = d^2(\boldsymbol{v}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_j - \boldsymbol{v}_i)^T (\boldsymbol{x}_j - \boldsymbol{v}_i)$$

is used as distance measure for distances between prototype vectors v_i and feature vectors x_j, the fuzzy clustering algorithm is called fuzzy c-means algorithm. Other distance measures can by applied resulting in clustering techniques which can adopt different cluster shapes [9, 10]. With the Euclidian distance measure the fuzzy c-means algorithm finds approximately equally sized spherical clusters.

The minimization of the functional (1) represents a nonlinear optimization problem that is usually solved by means of Lagrange multipliers, applying an alternating optimization scheme [2]. This optimization scheme considers alternatingly one of the parameter sets, either the membership degrees

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{1}{m-1}}} \tag{5}$$

or the prototype parameters

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \boldsymbol{x}_j}{\sum_{j=1}^{n} (u_{ij})^m} \tag{6}$$

as fixed, while the other parameter set is optimized according to equations (5) and (6), respectively, until the algorithm finally converges.

3 Visualizing Single Clusters

We propose in this paper to visualize single clusters by projection of the data points onto the plane under the constraint that the membership degrees to clusters are preserved. Note, membership degrees can be obtained directly when using a fuzzy clustering algorithm (e.g. fuzzy c-means), but also when calculating membership degrees after the partitioning, which can be done for any prototype-based clustering algorithm. To achieve the objective of membership preservation, we adopt the noise distance aspect of the noise clustering technique [6].

Noise clustering is based on the introduction of an additional noise cluster that is supposed to contain all feature vectors that are about a certain distance,
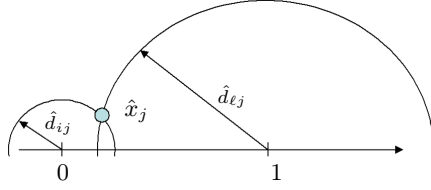
Fig. 1. Placement of $\hat{x}_j$ in the plane

the noise distance δ, away from all other prototype vectors. This means that the prototype v_c for the noise cluster c has no parameters. The clustering scheme differs only in one point from k-means or fuzzy c-means. When calculating the membership degrees the distance of the feature vector x_j to the noise cluster v_c is the fixed constant value $d_{cj} = \delta^2$. The proper specification of δ is discussed in [6, 15].

With the objective to place the cluster in the plane, we need for each data point two coordinates. Note, that the constraint for the projection is not to preserve the distances d_{ij} but the membership degrees u_{ij}. The idea for our visualization is to compute the distances to the cluster prototypes by means of the membership degrees. To achieve this we consider the usual computation of membership degrees as mentioned in equation (5). This provides a very simple connection between membership degrees and distances

$$\frac{u_{ij}}{u_{\ell j}} = \frac{\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}}}{\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{\ell j}}{d_{kj}}\right)^{\frac{1}{m-1}}}} = \left(\frac{d_{\ell j}}{d_{ij}}\right)^{\frac{1}{m-1}} . \tag{7}$$

For the purpose of visualization we propose to place the cluster i to be visualized at $(0, 0)$ and to choose a second cluster ℓ at $(1, 0)$. The cluster at $(1, 0)$ is a virtual cluster that contains all feature vectors with the highest membership degree apart from u_{ij}. The intention of this cluster is to collect all feature vectors that are assigned to another cluster than the one we want to visualize. Let us denote the membership degree to the most competing cluster by $u_{\ell j}$. Furthermore, we introduce a noise cluster to cover the clusters apart from i and ℓ. According to the distance of the two chosen cluster prototypes at $(0, 0)$ and $(1, 0)$, we define the noise distance $\delta = 1$. This means we have $u_{noisej} = 1 - u_{ij} - u_{\ell j}$. According to equation (7) this leads to

$$\frac{u_{ij}}{u_{noisej}} = \left(\frac{1}{\hat{d}_{ij}}\right)^{\frac{1}{m-1}} . \tag{8}$$

We denote the distance between cluster i and ℓ on the plane by $\hat{d}_{ij}$ to emphasize the fact that we do not deal with original distances any more but with representative distances with respect to the according membership degrees. Solving

equation (8) for $\hat{d}_{ij}$ we obtain

$$\hat{d}_{ij} = \left(\frac{u_{noisej}}{u_{ij}} \right)^{m-1}. \tag{9}$$

Analogously, we obtain for the second cluster ℓ

$$\hat{d}_{\ell j} = \left(\frac{u_{noisej}}{u_{\ell j}} \right)^{m-1}. \tag{10}$$

This approach enables us to visualize some useful aspects: 1. which feature vectors can be assigned clearly to the cluster i of intrest, 2. if a feature vector cannot be assigned to i, is there another cluster ℓ, where the vector can be assigned to, 3. which feature vectors are near to one or more prototypes apart from i and ℓ, 4. are there feature vectors that cannot be assigned to any cluster clearly.

With equation (9) one can compute the distance of each feature vector x_j to the cluster i, so that it is possible to draw a circle around $(0,0)$ as one hint for the feature vector's position in the plane. With the distance to the other cluster $(1,0)$, one could draw another circle around the cluster centre. The intersection point of these two circles would be the position of the new feature vector in the plane.

Figure 1 illustrates this approach. The small circle that represents the potential coordinates of $\hat{x}_j$, can be drawn with distance $\hat{d}_{ij}$ obtained from equation (9). Analogous, the bigger circle can be drawn with $\hat{d}_{\ell j}$ that we get with equation (10). The intersection point of these two circles represents the feature vector $\hat{x}_j$ in the plane.

4 Implementation Aspects

Apart from the clustering itself, which leads to the membership degrees u_{ij} another parameter affects the transformation, namely m (see equations (8, 9, 10)). A priori, one would set m the same value as for the clustering. But it can be also useful to modify this parameter. Practical tests have shown that in some cases, i.e. when a feature vector is very close to a prototype vector, no intersection point can be obtained in the plane and consequently the membership degrees to the respecting feature vector cannot be preserved exactly.

The rules shown in algorithm (1) handle such cases while trying to preserve membership degrees approximately. Let us denote the transformed data set $\hat{X}$. With $\hat{x}_{2j} = 0$ we define for a feature vector that is very close to one certain prototype vector a position on the x-axis on the plane. The rest of the rule tries to find the proper position for $\hat{x}_j$ on the x-axis balancing the distances to cluster $(0,0)$ and cluster $(1,0)$. If the distance to both clusters is relatively small, say $\max(\hat{d}_{ij}, \hat{d}_{\ell j}) < 1$, then we compute a position between both clusters in relation to $\hat{d}_{ij}$ and $\hat{d}_{\ell j}$. Otherwise, which means one or both clusters are about a distance of 1 or further away from the feature vector we distinguish whether

Algorithm 1	Placement of $\hat{x}_j$ on the x-axis

if no intersection point **then**

    $\hat{x}_{2j} = 0$

    if $\max(d_{ij}, d_{\ell j}) < 1$ **then**

        $\hat{x}_{1j} = d_{ij}/(d_{ij} + d_{\ell j})$

    else

        if $d_{ij} > d_{\ell j}$ **then**

            $\hat{x}_{1j} = 1 + d_{\ell j}$

        else

            $\hat{x}_{1j} = -d_{ij}$

        end if

    end if

end if

cluster $(0,0)$ or cluster $(1,0)$ is nearer. If the distance of x_j to cluster $(0,0)$ is higher than the distance to cluster $(1,0)$ then $\hat{x}_j$ will be placed to the right of cluster $(1,0)$ at $\hat{x}_j = (1 + \hat{d}_{1j}, 0)$. If the distance $\hat{d}_{ij}$ to cluster $(0,0)$ is smaller than the distance $\hat{d}_{\ell j}$ to cluster $(1,0)$ then $\hat{x}_j$ will be placed to the left of cluster $(0,0)$ at $\hat{x}_j = (-\hat{d}_{ij}, 0)$. This concept enables us to place the data point at least accurate relative to the nearest cluster. However, it is not very interesting to know exactly how far away the feature vector is, since the distance is quite large in fact.

With these rules the membership degrees cannot be preserved exactly, but approximated intuitively. Alternatively, one can avoid this kind of approximation by modifying parameter m for the transformation process. Small values $m \to 1$ prevent that no intersection point can be met. Otherwise, one can set higher values for m to force placements on the x-axis. Such transformations may be not that differentiated, but information can be reduced to some essential facts if needed. Generally, data points situated left from 0.5 on the x-axis can be assigned to cluster $(0,0)$, while data points on the other side belong to another cluster.

5   Results

Let us first apply our visualization method to an artificial data set. The cube data set (see figure 2(a)) consists of eight well separated clusters, which are in the corners of an imaginary 3-dimensional cube. A fuzzy c-means partition of the data set with five prototypes is shown in the figure. Of course, eight prototypes would be the best choice to partition the cube data set with. Thus, we can illustrate with this partition which information one can get from the visualization tool.

Figure 2(b) shows the transformation of the cube data set from the perspective of prototype A. Clearly four groups of data points can by observed (circled with a dashed line). The data points in group 1 are those, which can by clearly assigned to prototype A. Data points that are located in group 2 are those, which
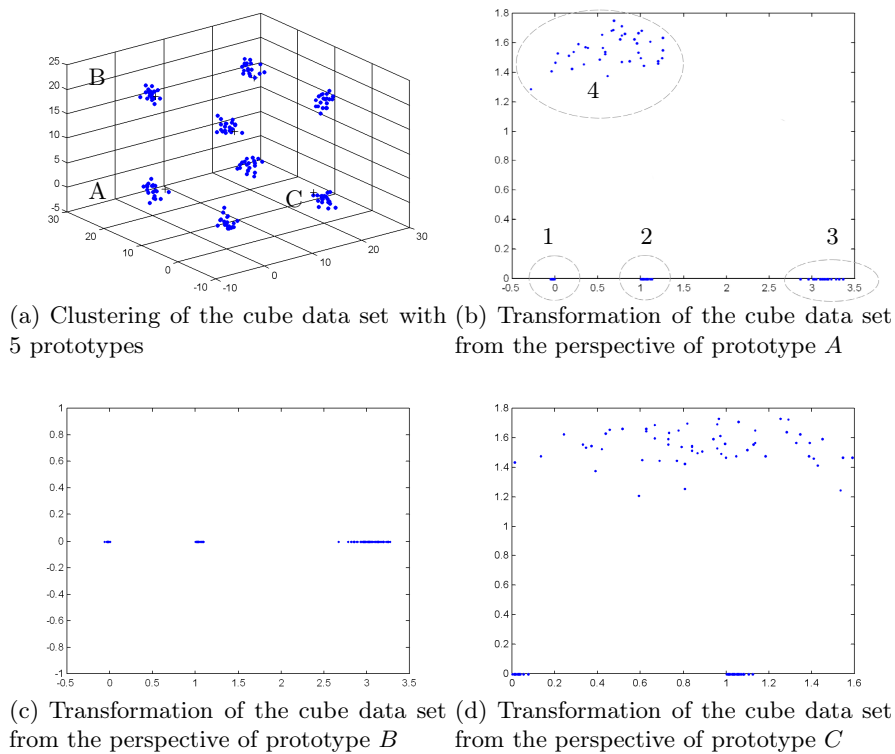
(a) Clustering of the cube data set with 5 prototypes

(b) Transformation of the cube data set from the perspective of prototype A

(c) Transformation of the cube data set from the perspective of prototype B

(d) Transformation of the cube data set from the perspective of prototype C

Fig. 2. An illustrative example

are not assigned to prototype A at all, but to another prototype. Note, a partition that only consists of these both groups is ideal. Group 3 stands for feature vectors, which are not assigned to prototype A and not to any other prototype. Instead, the data points have approximately the same membership degree to two or more prototype vectors (but not to prototype A). Group 4 represents feature vectors that have approximately the same membership degree to prototype A and another prototype.

Figure 2(c) shows the transformation of the cube data set from the perspective of prototype B. At first sight one can notice that group 4 is absent. That means in fact that no other prototype than prototype B has high membership degrees to the data points in group 1. A closer look reveals that the distance of prototype B to some misrepresented data is higher comparing to other prototypes, such as prototype A and C. All other data points that could contribute to group 4 are clearly represented by some prototypes. The transformation of the cube data set from the perspective of prototype C is shown in figure 2(d). Now group 3 is missing in the plot. This becomes evident, because all data points that are underrepresented are directly between prototype C and at least one other
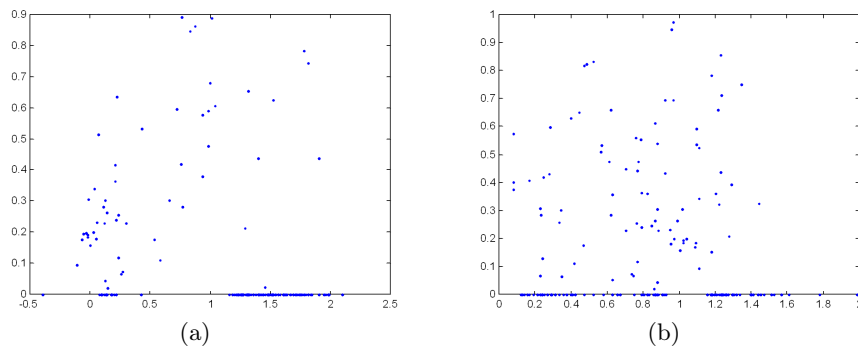
(a)

(b)

Fig. 3. Some transformations of clusters of the wine data set

(a) Transformation with $m = 1.2$
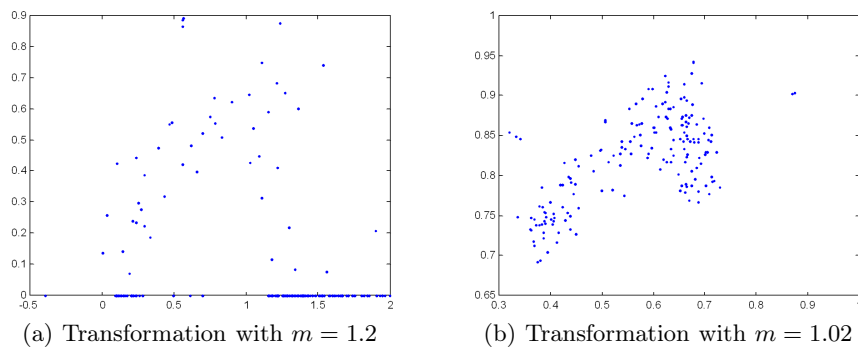
(b) Transformation with $m = 1.02$

Fig. 4. The effect of parameter m

prototype. As we have discussed above, group 3 only occurs when data points have low membership degrees to the regarding prototype and approximately equal membership degrees to two or more other prototypes. Since prototype C is at least as near as other prototypes, group 3 cannot be formed.

Figure 3 shows some results on the well known wine data set. The figure shows exemplarily two clusters of a partitioning with four prototypes. The left one is a visualization of a quite compact cluster. Data points left from 0.5 on the x-axis whose component on the y-axis is greater than zero have only small membership degrees to the cluster $(1, 0)$ even if their distance to cluster $(0, 0)$ seams to be far. This is due to the relatively small fuzzifier. The cluster shown in figure 3(b) is much more overlapping other clusters as the points on the x-axis, fairly in the middle between both clusters, indicate. As mentioned above, using small values for m leads to rather sensitive transformations. Even a relative small membership degree to a certain cluster attracts the data point in the transformation. To

smooth this effect it is advisable to decrease m for the transformation or increase m for the clustering if possible.

The effect of decreasing m for the transformation is shown in figure 4. While figure 4(a) shows the transformation of a cluster of the wine data set with $m = 1.2$, the figure 4(b) shows the same cluster transformed with $m = 1.02$. The changeover from cluster $(0, 0)$ to cluster $(1, 0)$, which is the imaginary line at 0.5 through the x-axis, is rather sparse. This fact indicates a compact cluster with only few feature vectors which cannot be assigned that clear to any cluster.

6   Conclusion

We presented in this paper a new method to visualize fuzzy clustering partitions on the plane. The visualization reveals whether a cluster is compact and if there is some data from the perspective of the respective cluster that is not well represented. Our results on two data sets are promising. Subject of future work will be the development of an appropriate evaluation method.

References

 1. Abonyi, J., Babuska, R.: FUZZSAM - visualization of fuzzy clustering results by modified Sammon mapping. Proceedings of the IEEE International Conference on Fuzzy Systems pp. 365–370, 2004.
 2. Bezdek, J.C.: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2, pp. 1–8, 1980.
 3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.
 4. Davies, D.L., Bouldin, W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1, pp. 224–227, 1979.
 5. Dunn, J.C.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics, 4, pp. 95–104, 1974.
 6. Dave, R.N.: Characterization and detection of noise in clustering, Pattern Recognition Letters, 12, pp. 657–664, 1991.
 7. Dave, R.N., Krishnapuram, R.: Robust Clustering Methods: A Unified View, IEEE Transactions on Fuzzy Systems, 5, pp. 270–293, 1997.
 8. Dave, R.N., Sumit, S.: On Generalizing the Noise Clustering Algorithms. In Proceedings of the 7th Fuzzy Systems Association World Congress (IFSA'97), 3, pp. 205–210, 1997.
 9. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, pp. 773–781, 1989.
10. Gustafson, D.E., Kessel, W.C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. Proceedings of the IEEE Conference on Decision and Control, San Diego, pp. 761–766, 1979.
11. Hathaway, R.J., Bezdek, J.C,: Visual cluster validity for prototype generator clustering models. Pattern Recognition Letters, 24(9–10), pp. 1563-1569, 2003.
12. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis, John Wiley & Sons, Chichester, 1999.

13. Huband, J.M., Bezdek, J.C., Hathaway, R.J.: bigVAT:Visual assessment of cluster tendency for large data sets. Pattern Recognition Letters, 38, pp. 1875–1886, 2005.
14. Klawonn, F., Chekhtman, V., Janz, E.: Visual Inspection of Fuzzy Clustering Results. In: Benitez, J., Cordon, O., Hoffmann, F., Roy, R. (eds.): Advances in Soft Computing: Engineering Design and Manufacturing. Springer, London , pp. 65–76, 2003.
15. Rehm, F., Klawonn, F., Kruse, R.: New approaches to noise clustering for detecting outliers. Soft Computing Journal, Springer, (to appear).
16. Rubens, M.: Fuzzy clustering algorithms and their cluster validity. European Journal of Operational Research, 10, pp. 294–301, 1992.
17. Windham, M.P.: Cluster validity for fuzzy clustering algorithms. Fuzzy Sets and Systems, 5, pp. 177–185, 1981.