Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 1381 – 1390

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Exploring data sets for clusters and validating single clusters

Frank Klawonn[a,b,*]

[a]Department of Computer Science, Ostfalia University of Applied Sciences, Salzdahlumer Str. 46/48, D-38302 Wolfenbuettel, Germany
[b]Biostatistics, Helmholtz Centre for Infection Research, Inhoffenstr. 7, D-38124 Braunschweig, Germany

Abstract

Cluster analysis is often used to find clusters and algorithms are designed and tuned to find the "right" clusters. Instead of searching for the "best" clustering algorithm, we argue that a clear concept of what the aim of a cluster analysis is and a better understanding of the data – especially based on visualisations – can be more crucial than the search for the right algorithm. In this paper, we revisit a method called dynamic data assigning assessment clustering that was intended both to asses the inherent cluster structure in a data set as well as to find the clusters. Here we extend this algorithm for better visualisation of possible cluster structures and also to validate single clusters that were found by other algorithms. Although this new approach can help to identify clusters, it is supporting tool and not used as a clustering algorithm itself.

Peer-review under responsibility of KES International

Keywords: Cluster analysis; cluster visualisation;

1. Introduction

Cluster analysis encompasses a collection of unsupervised classification methods that are intended to find structures in data sets in the form of clusters where similar data objects are grouped together into clusters. Although this simple concepts might sound very appealing, the actual purpose of cluster analysis is not very clear and various clustering methods are applied more for exploring the algorithms than to gain insight into the data. We strongly argue that the purpose of clustering should be made very clear in the beginning and that emphasis should be put on understanding the data than just apply a clustering algorithm and take the clusters as granted. We therefore start a short discussion on the general usage of cluster analysis in Section 2. Section 3 provides the necessary background on our concepts for investigating the cluster structure and clusters in data set. How to apply these concepts to a whole data set is described in Section 4 and Section 5 illustrates how single clusters could be validated by our approach. In the final conclusions we emphasise that our approach should be considered as complementary tool to other exploratory methods and should be used in combination with other visualisation techniques.

* Corresponding author. Tel.: +49-5331-939-311000; fax: +49-5331-939-31004.
E-mail address: f.klawonn@ostfalia.de

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.183

2. Clustering as an exploratory data analysis technique

Cluster analysis can almost be seen as an art to find the right ingredients for unfolding the inherent structure in a data set. Many clustering algorithm are designed for an ideal world where the data set consists of well-separated clusters and perhaps a little bit of noise. If a data set really has this structure, cluster analysis is the ideal tool to use. However, real data sets are often not conform to these ideal assumptions.

Cluster analysis can be applied for different purpose and sometimes it is not really made clear in the beginning what the actual purpose is. In our experience the main purposes for applying cluster analysis are the following ones.

- Finding clusters inherent in the data set under the assumptions that these cluster really exist. Most clustering algorithms are designed for this ideal purpose.
- Partitioning the data set into subset. Here it is not very important that the clusters are more or less well-separated. The partition of the data set is needed to reduce the complexity of the data set and handle the clusters separately. It is still important that the homogeneity criterion of cluster analysis is satisfied, i.e. that data objects in the same cluster are similar. But the heterogeneity criterion is dropped. Data objects from different clusters are allowed to be similar.
- Finding single clusters. Whereas standard clustering algorithms usually assume that the data set consists of clusters and a limited amount of noise data, here it is not important to cover a large fraction of the data by clusters. It is sufficient to find one or a few well-separated clusters and the majority of the data might not be assigned to any cluster.
- Data are already at least partly labelled by classes and one wants to verify whether the classes correspond roughly to clusters. Data from genomics and proteomics experiments [1] are a typical scenario for this clustering purpose. If data objects from the same class are scattered over all clusters, this might be an indication that something went wrong during the experiment and they data might be corrupted.
- Semi-supervised classification [2]. This situation is similar to the previous item, data objects are partly labelled by classes. The aim is now to assign the unlabelled data to classes. When the clusters corresponds well to the classes formed by the labelled data, it is reasonable to assign the an unlabelled data object to the class that has a (strong) majority in the corresponding cluster.

One crucial ingredient of cluster analysis is the distance or similarity measure. This is a discussion on its own and we restrict our considerations here to the Euclidean distance for data in $\mathbb{R}^q$. Of course, data can be or should be transformed properly before applying cluster analysis, for instance by normalising each attribute. It should be noted that our approach mainly relies on membership degrees and is therefore also applicable to other distance measures as they are, for instance, used in the Gustafson-Kessel algorithm [3]. A detailed discussion of distance measures is out of the scope of this paper.

There are too many clustering algorithms that they could be listed here, so that we mention only a small selection here. Very basic algorithms are hierarchical clustering and the k-means algorithm (see for instance [4]). A main difference between them is that hierarchical clustering requires a distance or (dis-)similarity matrix between data objects whereas k-means clustering is explicitly designed for data from $\mathbb{R}^q$. We will mainly focus here on fuzzy clustering. The most basic fuzzy clustering methods is the fuzzy c-means algorithm [5,6]. Gaussian (or other multivariate distribution-based) mixture models are another technique based on a probabilistic model whose parameters need to be identified by the clustering algorithm. A common property of all these algorithms is that – at least in their basic forms – the number clusters must be specified in advance and is not determined automatically. That the number of clusters is known a priori as often an unrealistic assumption and various methods on top these algorithms were designed to help to identify the number of clusters.

Density-based clustering algorithms like DBSCAN [7] or Denclue [8] automatically determine the number of clusters and can even identify noise data. The underlying assumption of these algorithms is that a cluster corresponds to a connected region of high data density. Subspace clustering [9] is specifically designed for high-dimensional data where the so-called concentration of norm phenomenon [10,11] causes problems to use the Euclidean distance directly in the high-dimensional space.

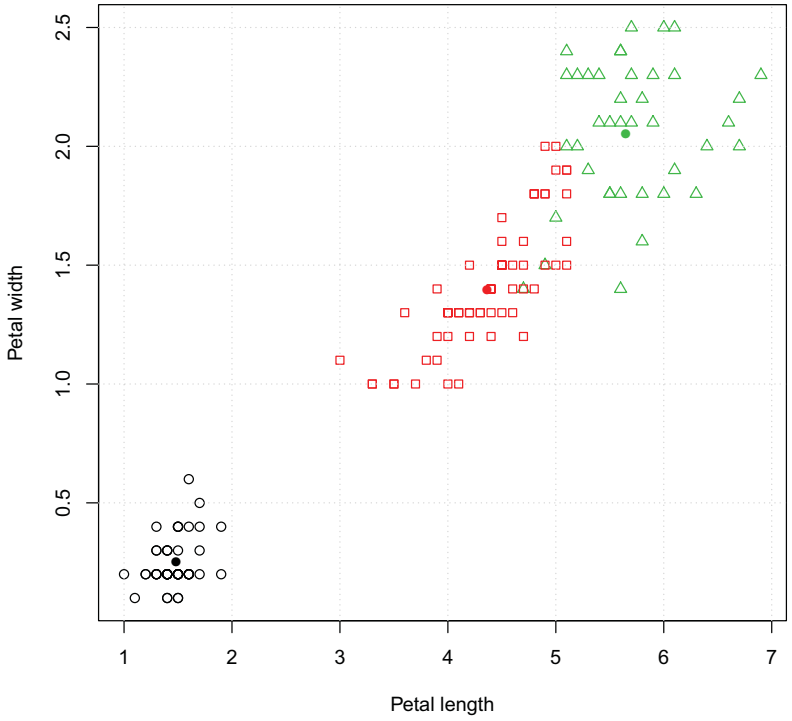

Fig. 1. FCM result for the Iris data set.

There also many cluster validity measure that either evaluate a clustering result as a whole – global measures – or evaluate single clusters, i.e. local measures. Such measures are especially abundant in the area of fuzzy clustering [12,13] because the membership degrees to clusters provide additional information about the validity of clusters. The more ambiguous data are assigned to clusters, the less valid are the clusters.

Most of these cluster validity measures yield a number indicating how well or bad clusters are. Such a number is good to use in other algorithms for further processing, e.g. trying to determine the number of clusters automatically, but more difficult for a human.

More suitable for humans are visualisation techniques like the silhouette plot, the extension of DBSCAN in the form of OPTICS [14] or special methods in the context of fuzzy clustering [15,16,17].

These visualisation techniques require that the clusters have already been found. Before we can introduce a visualisation technique that can also be applied before clusters are determined, we need to briefly recall some preliminaries from previous work.

3. DDAA revisited

We will mainly base our ideas on concepts from fuzzy clustering and will restrict our considerations to the simplest version, i.e. the fuzzy c-means algorithm (FCM) as it was introduced by Dunn and Bezdek [5,6]. In contrast to the crisp version, the k-means algorithm, FCM is less sensitive to initialisation [18].

FCM clusters a data set $\{x_1, \ldots, x_n\} \subset \mathbb{R}^q$ by finding prototypes or cluster centres v_i for a given number of clusters c and assigning the data objects to clusters with membership degrees u_{ij}. FCM tries to minimise the objective function

$$f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^w d_{ij} \tag{1}$$

under the constraints

$$\sum_{i=1}^{c} u_{ij} = 1 \qquad \text{for all } j = 1, \ldots, n \tag{2}$$

where $u_{ij} \in \{0, 1\}$ indicates whether data vector x_j is assigned to cluster i ($u_{ij} = 1$) or not ($u_{ij} = 0$). $d_{ij} = \| x_j - v_i \|^2$ is the squared Euclidean distance between data vector x_j and cluster prototype v_i. The minimisation of this objective function is carried out by an alternating optimisation schemes that alternatingly updates the prototypes v_i and the membership degree u_{ij}.

Throughout this paper we will use the well-known Iris data set[19] to illustrate our methods. The Iris data set consists of four numerical attributes giving the lengths and widths of the sepal and petal leaves of three types of Iris flowers, setosa, versicolor and virginica. The setosa group froms a cluster that is well-separated from the other two types of Iris flowers whereas the other two correspond to slightly overlapping clusters. Fig. 1 shows a projection of the Iris data set to two dimensions and clustering result based on FCM. A data point is assigned to the cluster with the highest membership degree. Different clusters are marked by different symbols and different colours. The lower left cluster of black circles corresponds to the setosa flowers. The filled circles correspond to the cluster prototypes.

An extension of FCM to cope with noise data was proposed by Davé[20]. Noise clustering uses an additional noise cluster that does not have a prototype but a fixed large distance δ to all data objects. In this way, data objects that are far away from all clusters are assigned to the noise cluster with high membership degree.

Dynamic data assigning assessment clustering (DDAA)[21,22,23] was introduced to assess the cluster structure in a data set and even to identify the clusters, assign the data to the clusters and remove noise data. The basic idea of DDAA is the following. DDAA uses only one cluster and a noise cluster. It starts with a very large noise distance δ, so that all data objects are assigned to the single cluster. Then the noise distance is decreased step by step, resulting in shifting more and more data from the single cluster to the noise cluster until all but one data objects belong to the noise cluster. While data objetcs are shifted to the noise cluster, the prototype of the single cluster also changes its position. Fig. 2 illustrates the movement of the prototype for DDAA applied to Iris data set. The initial position of the prototype is indicated by a square, the final one by a triangle. This visualisation might not be very useful for other data sets, especially when clusters cannot be seen in projections to two dimensions.

But DDAA provides another interesting visualisation shown in Fig. 3 indicated by the black full line (Sum). The x-axis shows the values for the noise distance δ and the y-axis the proportion of data objects assigned to the single cluster. We have not summed up the membership degrees to the single cluster but have summed up the rounded values, i.e. counted the number of data objects having a higher membership degree to the single cluster than to the noise cluster. The reason for this will explained below.

One should read the this curve from right to left because the noise distance δ starts with a large value and is decreased step by step. The interesting observation is that this curve has plateau and two phases with a high slope. A plateau means that – although the noise distance decreases – almost no data objects are shifted from the single cluster to the noise cluster. A steep slopes corresponds to losing a large number of data objects to the noise cluster, even if the noise distance is only lower slightly. Such a steep slope can be an indication that a whole cluster is lost on the slope to the noise cluster. And indeed, the steep slope on the right of the Sum curve in Fig. 3 corresponds to the loss of the setosa cluster to the noise cluster.

Using the sum of the original membership degrees instead of the sum of the rounded membership degrees would make the Sum curve less pronounced, the plateau would vanish and the slopes were less steep.

Original DDAA[21,22,23] used the Sum curve and the delta of the Sum curve to identify clusters and noise data automatically. Noise data are those shifted to the noise cluster during a plateau phase. Here we are not interested in a clustering procedure but to use extension of DDAA to identify candidates for clusters and to validate clusters that have been found by other algorithms.

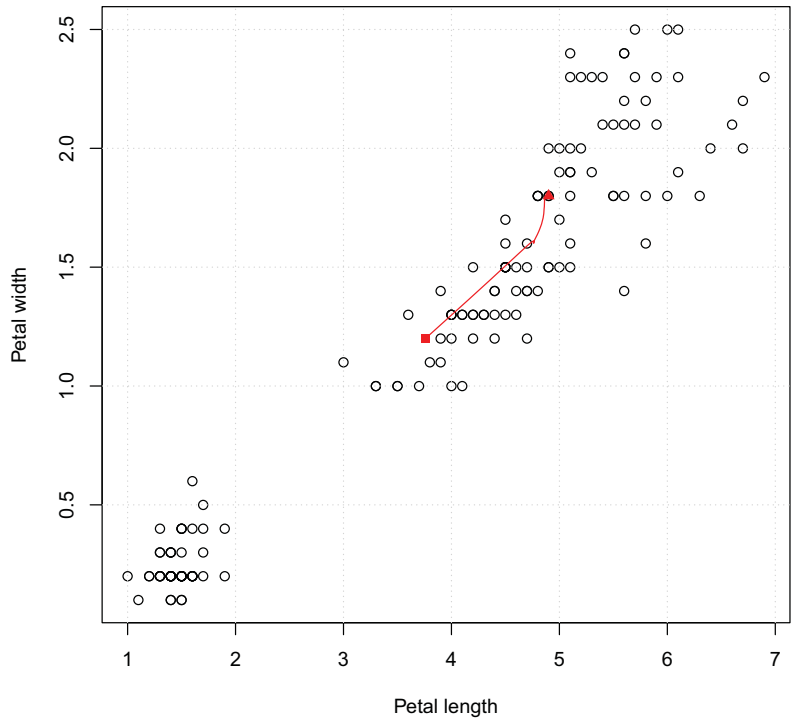

Fig. 2. DDAA prototype movement for the Iris data set.

4. Extending the DDAA measures

Apart from the Sum curve, Fig. 3 shows a number of other curves that can also provide interesting or additional information on the potential clusters in the data set. In this section we explain how these additional curves are defined. The PC and PE curves are based on validity measures used in fuzzy clustering. These validity measures assign good values to a clustering result when the membership degrees tend to be almost non-fuzzy, i.e. when data points are assigned to a cluster either with a membership degree close to one or close to zero. The partition coefficient[6] (PC) is defined by

$$\frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2}{n}.$$

The higher the value of the partition coefficient the better the clustering result. The highest value 1 is obtained, when the fuzzy partition is actually crisp, i.e. $u_{ij} \in \{0, 1\}$. The lowest value $1/c$ is reached, when all data are assigned to all clusters with the same membership degree $1/c$.

The partition entropy[6] (PE)

$$\frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij} \ln(u_{ij})}{n}$$

is inspired by the Shannon entropy. The smaller the value of the partition entropy, the better the clustering result.

It should be noted that that in both cases the first sum consists only of two summands: one for the single cluster and one for the noise cluster.

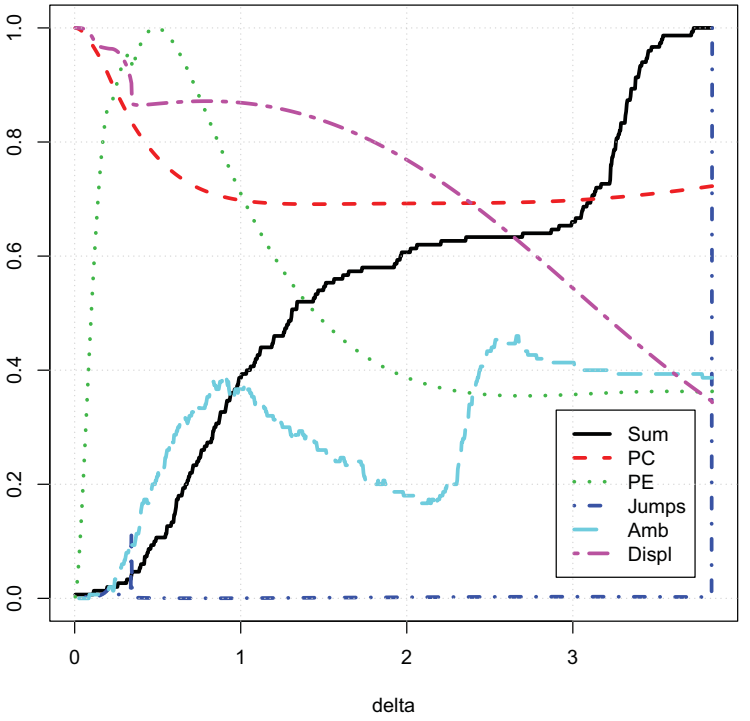

Fig. 3. DDAA result for the Iris data set.

Looking at the PC curve in Fig. 3, one can see that is starts to increase (looking at it again from right to left according to the decreasing noise distance) at the end of the plateau of the Sum curve. This is the point where the cluster formed by versicolor and virginica starts to be shifted to the noise cluster.

The PE curve starts to increase close to the end of the plateau and has its maximum – worst index for a clustering result – at around $\delta = 0.5$. This is when in addition to the full setosa cluster roughly half of the data objects from the versicolor and virginica cluster are lost to the noise cluster.

The Jumps curve in Fig. 3 shows the changes of the position of the protoytpe of the single cluster, normalised by its highest value. Here this curve does not provide interesting information. But we will see in the following section that it can also be of interest.

The Amb curve in Fig. 3 – for ambiguity – shows the proportion of ambiguous membership degrees. Here we have defined a membership degree to be ambiguous if it is between 0.3 and 0.7. The two local maxima of this curve indicate the points where the setosa cluster (right maximum) and the versicolor and virginica cluster are lost to the noise cluster. The local minimum at around $\delta = 2$ occurs when the full setosa cluster is lost to the noise cluster.

Finally, the Displ curve in Fig. 3 – for displacement – shows the displacement of the prototype from its initial starting positions, normalised by its highest value. The plateau of this curve corresponds to the point when the setosa cluster is lost to the noise cluster and the versicolor and virginica cluster starts to be shifted to the noise cluster.

It should be noted that we know the clusters of the Iris data set and have explained the curves based on the known clusters. In a real situation one would look at the behaviour of the curves and draw conclusions on the cluster structure. Deep slopes of the Sum curve correspond to the loss of clusters, maxima of the Amb curve correspond to a point where a cluster is in the middle of being lost to the noise cluster etc.

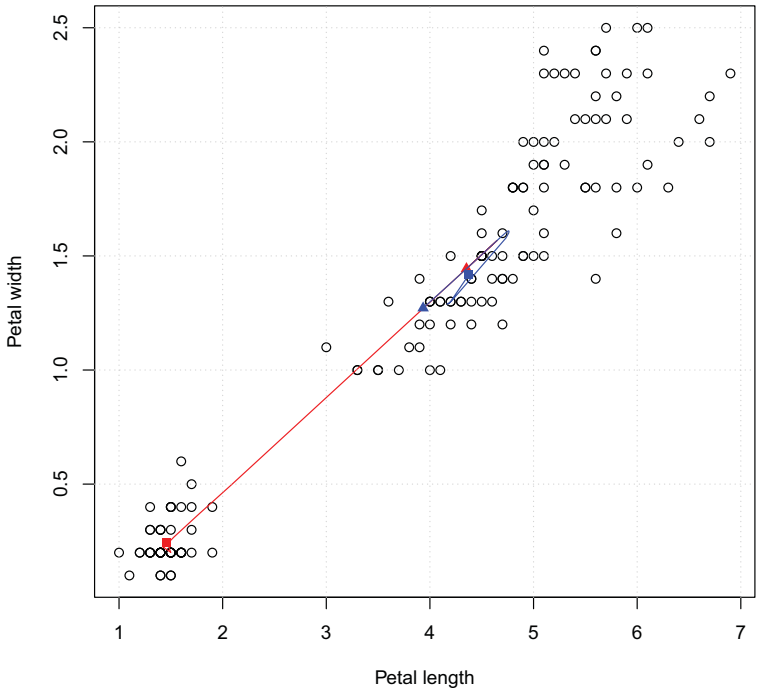

Fig. 4. DDAA prototype movements for using the centres of the left and the middle cluster as starting point, respectively.

5. Validating single clusters

We have seen in the previous section that the inspection of the development of various measures during the change of the noise distance δ can give potential insight to the cluster structures. Now we are interested in validating a cluster that has been identified already, for instance by FCM. For this purpose we reverse the DDAA process in the previous section. We start with a noise distance of (almost) zero and the prototype as the centre of the cluster that we want to validate. Then we increase the noise distance step by step until we have included all data objects of the cluster to be validated are now removed from the noise cluster.

As an example, we carry this out for the setosa cluster and the cluster in the middle of Fig. 1, indicated by the squares. Fig. fig:ddaac12 shows how the two prototypes move around when the noise distance is increased. The inital point is again the square, the terminal point the triangle.

Fig. 5 shows the curves described in the previous section when we start with setosa cluster. Note that the curves should now be read from left to right since we increase the noise distance step by step. For or all curves, we can observe a very significant behaviour at $\delta = 2$. This the point where the full setosa cluster has been shifted from the noise cluster to the single cluster and data from the versicolor and virginica cluster start to be shifted to the single cluster. What we observe here is that the setosa cluster is well separated from the rest of the data.

Fig. 6 shows the curves when we start with cluster in the middle of Fig. 1. What we can see here is that this cluster is not as well defined as the setosa cluster. Some curves indicate that something is "happening" at a noise distance of $\delta = 3$. But this is the point where the full versicolor and virginica cluster has been shifted from the noise cluster to the single cluster and the setosa cluster starts to be shifted to the single cluster. But the cluster we are trying to evaluate consists actually mainly only out of the versicolor flowers, i.e. half of the versicolor and virginica cluster.

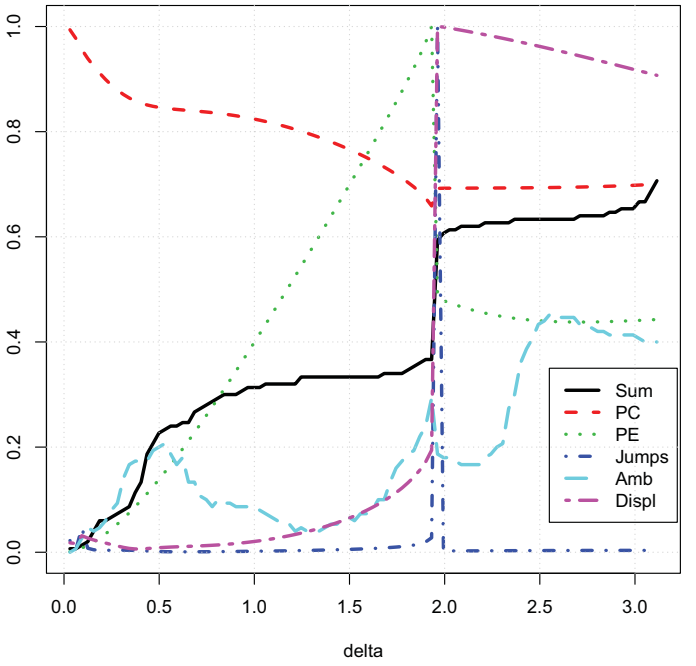

Fig. 5. DDAA starting from the lower left cluster.

The speaks in favour of joining the two flower groups versicolor and virginica to one cluster in terms of clustering, i.e. of unsupervised classification where we completely ignore the class information given by the species of the Iris flowers. Most visualisations also indicate that might be a suitable from the viewpoint of clustering.

6. Conclusions

We have proposed various curves to better understand the possible cluster structure in a data set and to validate single clusters. It should be emphasised that we consider these visualisation techniques as complementary to other approaches and always recommend to look at a data set from different angles.

We have restricted our considerations to simple FCM. Of course, one could easily apply our visualisations also to extensions of FCM like the Gustafson-Kessel algorithm[3], which is capable to fit to ellipsoidal cluster shapes or to the polynomial fuzzifier[24,25] that avoids certain disadvantages of the fuzzifier in FCM, especially when it is applied to higher-dimensional data where FCM tends to fail completely[26].

It should be noted that further experiments are needed to completely evaluate our visualisations and that there are more validity measures that could be included in our visualisation.

References

1. Kerr, G., Ruskin, H., Crane, M.. Techniques for clustering gene expression data. *Computers in Biology and Medicine* 2008;**38**(3):383–393.
2. Bair, E.. Semi-supervised clustering methods. *Wiley interdisciplinary reviews Computational statistics* 2013;**5**:349–361.
3. Gustafson, D., Kessel, W.. Fuzzy clustering with a fuzzy covariance matrix. In: *IEEE CDC*. San Diego; 1979, p. 761–766.

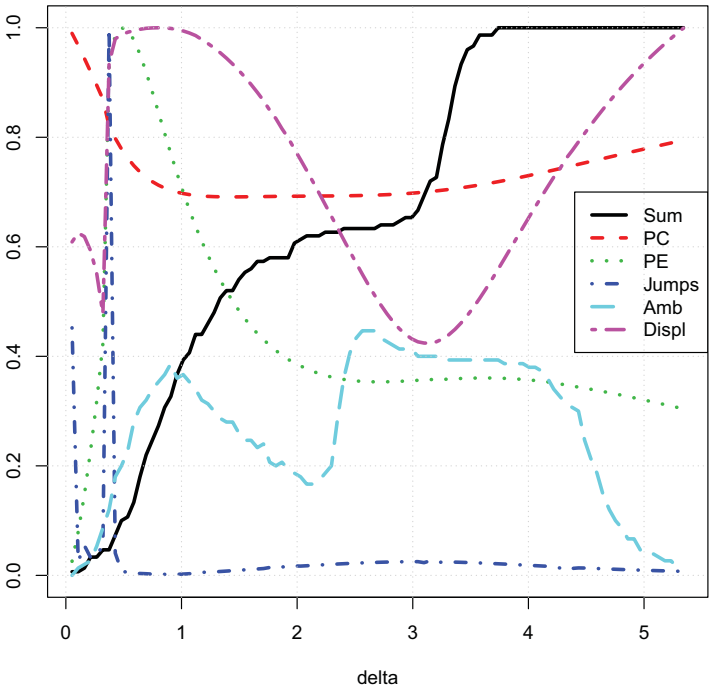

Fig. 6. DDAA starting from the cluster in the middle.

4. Duda, R., Hart, P.. *Pattern Classification and Scene Analysis*. New York: Wiley; 1973.
5. Dunn, J.. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Cybernetics and Systems* 1973; **3**(3):32–57.
6. Bezdek, J.. *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press; 1981.
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press; 1996, p. 226–231.
8. Hinneburg, A., Gabriel, H.H.. Denclue 2.0: Fast clustering based on kernel density estimation. In: *Proceedings of the 7th International Symposium on Intelligent Data Analysis*. 2007, p. 70–80.
9. Kriegel, H.P., Kröger, P., Zimek, A.. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans Knowl Discov Data* 2009;**3**(1):1–58. URL: `http://doi.acm.org/10.1145/1497577.1497578`. doi:10.1145/1497577.1497578.
10. Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U.. When is "nearest neighbor" meaningful? In: *ICDT*. 1999, p. 217–235.
11. Durrant, R.J., Kabán, A.. When is 'nearest neighbour' meaningful: A converse theorem and implications. *J Complexity* 2009;**25**(4):385–397.
12. Bezdek, J., Keller, J., Krishnapuram, R., Pal, N.. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Boston: Kluwer; 1999.
13. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.. *Fuzzy Cluster Analysis*. Chichester: Wiley; 1999.
14. Ankerst, M., Breunig, M., Kriegel, H.P., Sander, J.. Optics: Ordering points to identify the clustering structure. In: *ACM SIGMOD international conference on Management of data*. ACM Press; 1999, p. 49–60.
15. Klawonn, F., Chekhtman, V., Janz, E.. Visual inspection of fuzzy clustering results. In: Benitez, J., Cordón, O., Hoffmann, F., Roy, R., editors. *Advances in Soft Computing: Engineering Design and Manufacturing*. London: Springer; 2003, p. 65–76.
16. Rueda, L., Zhang, Y.. Geometric visualization of clusters obtained from fuzzy clustering algorithms. *Pattern Recognition* 2006;**39**(8):1415 – 1429.
17. Sharko, J., Grinstein, G.. Visualizing fuzzy clusters using RadViz. In: *2009 13th International Conference Information Visualisation*. 2009, p. 307–316.
18. Jayaram, B., Klawonn, F.. Can fuzzy clustering avoid local minima and undesired partitions? In: Moewes, C., Nürnberger, A., editors. *Computational Intelligence in Intelligent Data Analysis*. Berlin: Springer; 2012, p. 31–44.
19. Anderson, E.. The species problem in Iris. *Annals of the Missouri Botanical Garden* 2013;**23**:457–509.

20. Davé, R.N.. Characterization and detection of noise in clustering. *Pattern Recognition Letters* 1991;**12**:406–414.
21. Georgieva, O., Klawonn, F.. Evolving clustering via the dynamic data assigning assessment algorithm. In: *Proc. 2006 International Symposium on Evolving Fuzzy Systems*. IEEE; 2006, p. 95–100.
22. Georgieva, O., Klawonn, F.. Cluster analysis via the dynamic data assigning assessment algorithm. *Information Technologies and Control* 2006;**2**:14–21.
23. Georgieva, O., Klawonn, F.. Dynamic data assigning assessment clustering of streaming data. *Applied Soft Computing* 2008;**8**:1305–1313.
24. Klawonn, F., Höppner, F.. What is fuzzy about fuzzy clustering? understanding and improving the concept of the fuzzifier. In: Berthold, M.R., Lenz, H.J., Bradley, E., Kruse, R., Borgelt, C., editors. *Advances in Intelligent Data Analysis*; vol. V. Berlin: Springer; 2003, p. 254–264.
25. Winkler, R., Klawonn, F., Kruse, R.. Fuzzy clustering with polynomial fuzzifier in connection with m-estimators. *Applied and Computational Mathematics* 2011;**10**:146–163.
26. Winkler, R., Klawonn, F., Kruse, R.. Fuzzy c-means in high dimensional spaces. *Fuzzy System Applications* 2011;**1**:1–17.