Clustered Data Separation via Barycentric Radial Visualization

Adam Russell\(^1\) and Robert Marceau\(^1\) and Franck Kamayou\(^1\)
and Karen Daniels\(^1\) and Georges Grinstein\(^1\)

\(^1\)Computer Science Department, University of Massachusetts Lowell, Lowell, MA, USA

Abstract—This paper addresses visualizing clusters of multi-dimensional data using barycenters as cluster representatives within the RadViz radial visualization technique. RadViz is a composition of two mappings. Where in this two-stage mapping the cluster barycenters are formed is a key decision. Motivated by the nature of the second mapping, we form cluster barycenters at the end of the first stage, rather than at the start of the first stage. In the second stage we must select an appropriate configuration of dimensional representatives (dimensional anchors). Since this problem is intractable we present a heuristic to: 1) separate clusters and 2) move clusters away from the barycenter of the dimensional anchors. The heuristic uses our prior Voronoi quality assessment technique and our recent observation that circular motion of dimensional anchors confines each data image to an annulus. We demonstrate the benefit of our barycentric approach for a variety of clustered datasets.

Keywords: Visualization, clustering

1. Introduction

1.1 Motivation, Contribution and Overview

Data visualization is becoming more prominent as the dimensionality and number of records to be explored becomes large. This paper aims to visualize multi-dimensional clustered data, with scalability in mind, for situations in which cluster barycenters (i.e. averages, centers, or centroids) are cluster representatives. We use this framework:

1) Preprocess data using a clustering technique of choice.
2) Preprocess data to help preserve clustering structure under the chosen visualization mapping.
3) Apply the visualization mapping.

The input to this process is a collection of \( n \) records in Euclidean space \( \mathbb{E}^d \). For step 1, the user selects and applies a clustering technique, and after step 1 the data records are grouped into \( k \) disjoint clusters: \( C = \{C_1, \ldots, C_k\} \). The literature on clustering is extensive; relevant clustering research is summarized in Section 1.2.

Steps 2 and 3 require choosing and applying a visualization mapping to the clustered data. Among the many existing techniques, radial visualizations stand out as often providing useful views of multi-dimensional data [1]. Radial visualization has its origins in the 19th century [2]. In this paper we use RadViz [3], [4], one type of radial visualization, as the primary example. For a collection of records in the data space, RadViz maps each record to a point within a unit disk in a 2D image space. A dimensional anchor (DA) representing each of \( d \) dimensions in the data space is placed on the unit circle. In an interpretation using a spring force analogy with Hooke’s law, the image of each of the records is an “equilibrium” point if we imagine a collection of \( d \) springs with each spring attached to the point at one end and a DA at the other end. Figure 1 shows a RadViz result for 9D data with 5 clusters. A Voronoi diagram [5] structure is overlayed where each DA and the barycenter of the DAs are the Voronoi sites (Section 1.5). In this example only 4 of the 5 clusters are distinguishable, and all data falls within the region associated with the barycenter of the DAs, where cancellation of DA influences can occur. Our goal is to improve on this type of situation by visually separating clusters and trying to move them away from the barycenter of the DAs. RadViz and related work in radial visualizations are discussed in more detail in Section 1.3.

Figure 1: Dimensional anchor placement in RadViz for 1800 records of our 9D synthetic data with 5 clusters (4 visible), with Voronoi regions.

For step 2 in the visualization process, in the RadViz case one would preprocess the data to help prevent losing clustering structure early in the transformation process (see Section 1.3). For scalability reasons, during step 3 we operate on a set of cluster representatives rather than the data images themselves, with one representative for each of the \( k \) disjoint clusters: \( R = \{R_1, \ldots, R_k\} \). RadViz has a 2-stage visualization pipeline, as it is the composition of two mappings [6] (see Section 1.3). The first stage normalizes the data while the second applies DA weights in the image space. A key design question is: where in the pipeline to form the cluster representatives? We show in this paper that the choice makes a difference (Section 2), and that when...
using cluster barycenters as representatives, clustering quality is improved when forming the representatives after the first stage of the transformation and before the second stage rather than forming them in the original data space.

Selecting a DA configuration is another crucial design question in the RadViz visualization, and is equivalent to applying the second part of the RadViz mapping. We explain in Section 2 how the second part of the RadViz mapping guarantees that, regardless of the chosen DA configuration, in the image space the transformed barycenter clusters remain barycenters of the transformed data clusters. This provides strong justification for forming barycentric cluster representatives after the first stage of the pipeline rather than in the original data space. It also provides motivation for working with cluster representatives that are cluster barycenters rather than other types of cluster representatives (see Section 1.2).

For the remainder of step 3, cluster representatives are used in place of the data to select RadViz DA positions with the goal of preserving clustering structure in the visualization. Because we are working here with cluster representatives, many different DA configurations can quickly be examined, allowing better exploration of the search space and hence better preservation of clustering structure.

As discussed in Section 1.4, deciding where to place the DAs on the RadViz circle is a complicated challenge. In Section 3 this paper offers a heuristic for DA placement that supports 2 cluster visualization goals: 1) separate cluster representatives from each other and 2) move cluster representatives away from the (bary)center of the DAs in the RadViz circle to minimize “cancellation of forces” from opposing DAs. Figure 1 highlights the need for these goals. Our heuristic moves cluster representatives away from the barycenter of the DAs using a quality score (see Section 1.5) and our recent results on how data image points move as DAs move (see Section 1.6). Due to our usage of barycenters as cluster representatives and the importance of the barycenter of the DAs in finding a good DA configuration, we refer to our overall approach as barycentric.

Results are presented in Section 4 for a variety of clustered datasets, including the one from Figure 1. There we experimentally show that forming barycentric cluster representatives after the first stage of the RadViz pipeline is better than forming them in the original data space and we demonstrate the benefit of our DA placement heuristic. In this paper we assume that barycenters are appropriate representatives for our data clusters. Future work will attempt to preserve clustering structure for elongated or very irregularly shaped clusters; some preliminary thoughts on treating this difficult issue appear in our discussion of future work in Section 5.

### 1.2 Clustering

Jain et al. [7], [8] provides an overview of various clustering algorithms. It should be noted that clustering is subjective. These algorithms may be divided into several broad categories. Certain issues which need to be considered across several broad categories of clustering algorithms are discussed first. Although several different similarity measures may be used, we will utilize Euclidean distance between points in the 2-dimensional RadViz circle.

The algorithm may operate in either a divisive or an agglomerative mode. In the divisive mode, all data points are initially placed in a single cluster which is then partitioned until a suitable stopping criterion is met. For the agglomerative mode, each data point is placed in a singleton cluster. Clusters are then merged until a stopping criterion is met.

The algorithms may operate by only considering the distance between two data points along a single dimension (monothetic). Once a partitioning has been performed using one dimension, each cluster is partitioned again, using the distance along another dimension. Alternatively, the distance between data points may consider all dimensions (polythetic).

The membership of a data point in a particular cluster may be either “hard” or “fuzzy.” The algorithm may be either deterministic of stochastic. For very large data sets an incremental algorithm may be desirable. Also of interest for very large data sets is the selection of a smaller set of representative points. Agarwal et al. [9] discuss a “core set” of representatives. Another possible choice of representative is the barycenter (centroid) of a cluster.

The popular k-means partitional algorithm seeks to minimize a squared error (distance from the centroid) function. To be able to employ k-means clustering, the value of k, the number of clusters must be known a priori.

Concepts from Support Vector Machines have been used for Support Vector Clustering [10]. This technique can handle even highly interleaved clusters [11]. Challenges here include selecting an appropriate kernel function and its associated parameter values.

Several measures that exist for evaluating the “quality” of cluster separation are described by Desgraupe [12]. Of particular interest is the Dunn index. The Dunn index is the quotient of the minimal distance between any two points in distinct clusters over the largest diameter of any cluster. The Dunn index, \( \mathcal{J} \) for a set of clusters \( C_i \) is calculated as:

\[
\begin{align*}
    d_{\text{min}} &= \min_{i \neq j} \min_{p \in C_i, q \in C_j} \|pq\|, \\
    d_{\text{max}} &= \max_{p, q \in C_i} \|pq\|, \\
    \mathcal{J} &= \frac{d_{\text{min}}}{d_{\text{max}}}.
\end{align*}
\]

### 1.3 Radial Visualization

Early examples of radial visualizations are William Playfair’s pie charts and Florence Nightingale’s polar plots [2]. Other radial visualization work includes Circle Segments [13], Star Coordinates [14], [15], and SphereViz [16]. Yi et al. [17] describe a radial visualization that employs “magnets.” Additional RadViz research includes integration of RadViz with Parallel Coordinates by Bertini et al. [18]. Vectorized
RadViz [19], [20], [21] and using RadViz to visualize time series data [22]. Tominski et al. [23] describe several different visualization methods which take high dimensional data and map it to a 2D image space. A thorough survey of radial visualizations is provided by Draper et al. [24]. Diehl et al. empirically evaluate the strengths and weaknesses of radial visualization [25].

The RadViz visualization introduced above can be expressed as follows. Given the $n$ records in Euclidean space $E^d$, where the $i$th record is $v_i$ and that record’s $j$th dimensional value is $v_{i,j}$, we follow the notation from [6], where $\tilde{S}_j$ is the position on the RadViz circle of the $j$th DA and $\tilde{x}_i$ represents the location of the image of $v_i$:

\[ \eta_i = \sum_{j=0}^{d-1} \tilde{x}_i, \quad \tilde{x}_i = \eta_i \sum_{j=0}^{d-1} \tilde{S}_j v_{i,j}. \quad (1) \]

Daniels et al. show that RadViz is a special case of a normalized radial visualization (NRV) formulation. A variety of NRV properties are established there. Previously other authors, such as Nováková [26] and McCarthy et al. [27], had offered informal observations on properties formally stated and proved in Daniels et al. [6].

For the current paper we use 3 especially important aspects of the NRV mapping [6]: 1) it is a composition of two mappings and, 2) for cluster separation preservation some (rotational) preprocessing may be necessary prior to the application of the first mapping, and 3) the second mapping preserves barycentric combinations [28].

The first aspect decomposes the NRV mapping (in matrix form) as: $V N \Psi$, where:

- $V$ is a $1 \times d$ vector of coordinates of a data record $v_i$;
- $N$ is a $d \times d$ diagonal matrix with $\eta_i$ in each diagonal entry and 0 elsewhere;
- $\Psi$ is a $d \times d'$ matrix with each row equal to $\tilde{S}_j$.

$N$ is a perspective projection of the original data from the positive orthant of $E^d$ to $E^d$, with the center of projection at the origin and a projection hyperplane: $\sum_{j=d+1}^{d'} D_j = 1$, where $D_j$ is a variable associated with the $j$th dimension of the data space. See Figure 2 for a $d = 3$ example. In this paper we refer to the clipped projection hyperplane as $\eta$-space. $\Psi$ is an affinity [29], [28] from $E^d$ to $E^d$; for RadViz we have $d' = 2$. (Note: This affinity preserves barycentric coordinates, but is not guaranteed to be 1-1 or invertible since $d' \leq d$.)

The second aspect stems from the definition of $N$. If two or more clusters are intersected by the same ray from the origin, then the intersection points all project to the same point, making them indistinguishable. However, [6] provides preliminary results on rotational preprocessing of the data that helps mitigate this effect. We assume in this paper that the data has not only been clustered, but it has also been rotationally preprocessed. The third aspect of the NRV mapping will be exploited in Section 2.

### 1.4 Dimensional Anchor Placement

The focus of much of the other previous theoretical work was the reordering and redistributing of the DAs on the circle [20], [1], [27]. Ankerst showed that the problem of optimally ordering DAs on a circle is NP-complete [30], which motivates our heuristic method for DA placement in Section 3; this appeared in earlier form (for unclustered data) in Russell’s thesis [31]. Prior work on DA placement in RadViz either assumed that the DAs are uniformly distributed around the circle or else assumed that class membership of the data is known. For example, McCarthy et al. [27] cluster the data, extract features, use the $t$-statistic for feature reduction and then use a Class Discrimination Layout technique [20] that uses class structure to divide the circle into equal sized sectors; one for each class.

### 1.5 $Q$-Score

Relevant research on visual clustering metrics is surveyed in Russell [31]. This includes work by Bertini and Santucci [18] and Peng et al. [32]. As introduced in [33] and subsequently explored in [34], [31], the $Q$-score is based on the importance of the barycenter $b_P$ of the DAs:

\[ b_P = \frac{\sum_{j=0}^{d-1} \tilde{S}_j}{d}, \quad (2) \]

where the $P$ subscript refers to the configuration of DA positions. Any data record with all equal dimensional values maps to $b_P$ [31], [35]. Furthermore, records whose standard deviation, computed across dimensional values, is small tend to map close to the barycenter of the DAs [31], [35]. Adding $b_P$ to the set of DAs yields a set of $d + 1$ points that can be interpreted as sites for a Barycentric Voronoi Diagram (BVD), which is a variant of the famous proximity construct known as the Voronoi diagram [5]. This partitions the plane into $d + 1$ regions, one for each site. When the DAs are cocircular, as in most RadViz work, we obtain a Cocircular BVD (CBVD; Figure 1).

We view the region associated with $b_P$ (denote this region by $B_P$) as an undesirable place for data because it can reflect lowly expressed dimensional values or cancellation effects of opposing DAs. For this reason Russell et al. [33] introduced the $Q$ score as a way of measuring visualization quality. First
In Section 3.

DA configuration can be “reverse-engineered” from the point of view of how the data images move as dimensional anchors move along the RadViz circle and the remaining DAs are allowed to move in 360° arcs around the circle the locus of points obtainable for a given $a_i$ is an annulus $a_i$ (see Figure 3 for 3 moving DAs). The center of the annulus depends on the fixed DAs and the inner and outer radii depend on the moving DAs. Furthermore, given any point in $a_i$, it is shown in [31], [35] that a corresponding DA configuration can be “reverse-engineered” from the point within the annulus. These ideas are used within our heuristic in Section 3.

### 1.6 Point Sensitivity

Since one goal of this paper is to determine useful dimensional anchor placement, prior work on how the data images move as dimensional anchors move along the RadViz circle is of particular interest. Preliminary research by Russell [31] and later enhanced in [35] uses a blend of computational geometry and 2D robotic motion planning to show that when some of the DAs are fixed on the RadViz circle and the remaining DAs are allowed to move in 360° arcs around the circle the locus of points obtainable for a given $v_i$ is an annulus $a_i$, (see Figure 3 for 3 moving DAs). The center of the annulus depends on the fixed DAs and the inner and outer radii depend on the moving DAs. Furthermore, given any point in $a_i$, it is shown in [31], [35] that a corresponding DA configuration can be “reverse-engineered” from the point within the annulus. These ideas are used within our heuristic in Section 3.

Figure 3: From [35]: the annular path traced by a single point (indicated by the arrow) when the DAs for Lumbar Lordosis Angle, Pelvic Radius, and Grade of Spondylothesis are moved around the circle. Data points represent records from the Vertebral Column dataset [36] shown in RadViz within the Weave visualization system [37]. Center ≈ (0.27, 0.006) and inner radius = 0.07 and outer radius = 0.43.

### 2. Cluster Barycenter Formation

This section has 2 goals: 1) show that forming barycenters in the original data space can produce different results than forming barycenters in the η-space (Section 1.3), and 2) demonstrate that, regardless of how DAs are chosen during the second (Ψ) part of the mapping, transformed cluster representatives (if they are barycenters) are still cluster representatives of the transformed data.

For the first goal we have the following result so that the barycenter in η-space is well-defined:

**Proposition 2.1:** Any convex combination of a set of points in η-space is a point in η-space.

**Proof:** For arbitrary dimension $d$, a convex combination of a set of $q$ points, $x(1), x(2), \ldots, x(q)$ is $c = \sum_{h=1}^{q} h(λ(h)x(h))$, where $0 ≤ λ(h) ≤ 1$ and $\sum_{h=1}^{q} λ(h) = 1$.

Since each $x(h)$ is in the η-space we have:

$$\sum_{j=1}^{d} x_{j}^{(h)} = 1$$

For the convex combination of the points, we have:

$$\sum_{j=1}^{d} c j = \sum_{j=1}^{d} \left(\sum_{h=1}^{q} λ(h)x_{j}^{(h)}\right)$$

$$= \sum_{h=1}^{q} \left(\lambda(h)\sum_{j=1}^{d} x_{j}^{(h)}\right)$$

$$= \sum_{j=1}^{d} \lambda(h).1 = 1.$$

**Corollary 2.1:** The barycenter of a set of points in η-space is a point in η-space.

**Proof:** This follow immediately from Proposition 2.1 since for arbitrary dimension $d$, the barycenter of a set of $q$ points, $x(1), x(2), \ldots, x(q)$ is $b = (b_1, b_2, \ldots, b_d)$ where each $b_j = \frac{1}{q} \sum_{j=1}^{q} x_{j}^{(h)}$. We have $\sum_{j=1}^{d} b_j = 1$ and $\frac{1}{q} ≥ 0$. The barycenter of a convex combination of the $q$ points is a point in η-space.

We present an example (for $d = 2$) below of the difference between a cluster barycenter transformed into η-space versus the barycenter of points that have been transformed into η-space. For $d = 2$ the clipped projection hyperplane is the 2D line segment $L$ connecting point (0,1) and (1,0). So, under the $N$ transformation, each point $p$ in the original data space maps to a point $p’$ on $L$. The point $p’$ on $L$ is the intersection of $L$ with a ray through the origin and $p$. Note that, as mentioned in Section 1.3, all points that lie on the same ray through the origin map to the same point on $L$.

**Proposition 2.2:** The barycenter of the points in the η-space might not coincide with the $N$-transformation of the barycenter of the data points.

**Proof:** Consider the case where $d = 2$ and the data points are (1,3) and (3,1). The barycenter of these two points is (2,2). Applying the $N$-transformation to each of these points, we obtain the points $(\frac{1}{2}, -\frac{1}{2}), (\frac{1}{2}, \frac{1}{2})$, and $(\frac{1}{2}, \frac{1}{2})$ for the barycenter. If the point (3,1) is moved along a line segment...
that passes through the origin and \((3, 1)\), the \(N\)-transformation of this new point will remain \((3, 1)\). The barycenter of the new pair of points will be the midpoint of a line segment connecting \((1, 3)\) and the new point. By moving the new point away from the origin, the \(N\)-transformation of the new barycenter may be moved arbitrarily close to \((\frac{3}{4}, \frac{1}{4})\). □

A key contribution of these results is that they are independent of how the \(\Psi\) mapping is performed because Corollary 2.1 and the example in Proposition 2.2 only refer to the original data space and \(\eta\)-space.

For the second goal of this section, we recall that Section 1.3 stated (from [6]) that \(\Psi\) preserves barycentric combinations [28]. This implies that, if \(s\) represents a point that has been transformed from the data space via the \(N\) mapping into the \(\eta\)-space, and if we express \(s\) as a barycentric combination of points in the \(\eta\)-space: \(s = \sum_{h=1}^{q} \gamma_h g_h\) for \(\sum_{h=1}^{q} \gamma_h = 1\) and \((g_h \in \eta - \text{space})\), then:

\[
\Psi(s) = \Psi\left(\sum_{h=1}^{q} \gamma_h g_h\right) = \sum_{h=1}^{q} \gamma_h \Psi(g_h).
\] (4)

Now we observe that the barycenter of a set of points is a barycentric combination of that set of points; in this special case all \(\gamma_h\) are equal to \(1/q\). In the clustering context this implies that we can do the following. Apply \(N\) to each cluster \(C_m\) to obtain \(N(C_m)\). Now calculate the barycenter of \(N(C_m)\) and let this serve as the representative \(R_m\) of cluster \(C_m\). With the goal of finding a good configuration of DAs, the preservation of barycentric combinations under \(\Psi\) guarantees that \(\Psi(R_m)\) is the barycenter of \(\Psi(N(C_m))\). So, working with \(\Psi(R_m)\) provides a barycentric representative of cluster \(N\Psi(C_m)\) in the \(\eta\)-image space. This provides execution time scalability, which is especially useful if the total number of data records is large. Working with \(\Psi(R_m)\) for different \(\Psi\) possibilities, we can try many different DA configurations quickly inside of our heuristic, with each \(\Psi\) representing a different DA configuration.

### 3. Annular DA Placement Heuristic

We now combine our point sensitivity results (Section 1.6) with insights from our barycenter investigations (Section 2) to inform a DA placement heuristic. Our annular heuristic proceeds as follows: we first remove records with 0 standard deviation, rank the dimensions by \(t\)-statistic and select the highest ranking half as our moving set of DAs. Thereby we have removed records which are known to map exactly to the barycenter and also selected the dimensions which correspond to DAs which have a larger influence on the representative’s image point position. Our heuristic then iterates over all the cluster representatives, determining a DA configuration using an annulus corresponding to the potential representative positions, and computing \(Q\) at each pass. At the completion of this loop we select the DA configuration corresponding to the highest \(Q\) computed over all iterations.

Finally, we compute the Dunn index as a measure of how well we’ve separated the clusters. We’ve chosen the Dunn index as we feel it matches well to our intuitive sense of what we hope to achieve. The ratio of the minimum of inter-cluster distance to the maximal intra-cluster distance, a high Dunn index indicates denser, well separated clusters.

Note that steps 1, 2, and 17-18 operate on the complete set of original data points (untransformed), whereas inside the for loop we work with \(k\) transformed cluster representatives. To locate an extreme point \(p_m\) on annulus \(a_m\) \((1 \leq m \leq k)\) with respect to the barycenter \(b_p\) we shoot a ray through the annulus center and select the intersection of this ray with \(a_m\’s\) outer boundary that is furthest from \(b_p\). We justify this in Russell [31].

**ANNULAR_HEURISTIC(C.R)**

1: Remove from consideration in \(C\) the low standard deviation points according to Section 1.5.
2: Apply \(t\)-statistic to \(C\) to identify lowly ranked dimensions.
3: Fix lowly ranked dimensions uniformly on circle.
4: \(b_p \leftarrow\) barycenter of fixed dimensional anchors
5: \(\text{best}_DA \leftarrow\) initial uniform configuration of moving DAs
6: \(\text{best}_Q \leftarrow -1\)
7: for \(m = 1\) to \(k\) do
8: Make annulus \(a_m\) for cluster representative \(R_m\).
9: Identify extreme point \(p_m\) on \(a_m\) with respect to \(b_p\).
10: Reverse-engineer DA positions \(A_m\) for \(p_m\).
11: Using \(A_m\), calculate \(Q\)-score \(Q_m\) using CBVD built from \(A_m\).
12: if \(Q_m > \text{best}_Q\) then
13: \(\text{best}_Q \leftarrow Q_m\)
14: \(\text{best}_DA \leftarrow A_m\)
15: end if
16: end for
17: \(I \leftarrow\) Dunn index using \(\text{best}_DA\) and all the data.
18: \(Q \leftarrow\) \(Q\)-score using \(\text{best}_DA\) and all the data.
19: Report \(\text{best}_DA\) and \(Q + I\).

### 4. Results

Here we assess clustering visualization results for our 2 competing ways to form cluster representatives using our heuristic from Section 3, implemented in Perl. For this, we compare \(Q + I\) for calculating barycenters in the data space with the same measure for calculating barycenters in the \(\eta\)-space, where \(Q\) is the best \(Q\)-score (Section 1.5) and \(I\) is the Dunn index for clustering quality (Section 1.2).

In Table 1 the number of dimensions \((d)\), number of clusters \((k)\), number of data records \((n)\) are represented for each dataset. For each dataset \(Q_1 + I_1\) is \(Q + I\) for barycenters calculated in data space and \(Q_2 + I_2\) is \(Q + I\) for barycenters calculated in \(\eta\)-space. Datasets 1-4 are generated by starting with one randomly generated point for each of the desired number of clusters and then randomly perturbing that point...
Table 1: $Q_1 + I_1$ is for barycenters calculated in data space; $Q_2 + I_2$ is for barycenters calculated in $\eta$-space.

<table>
<thead>
<tr>
<th>Dataset #</th>
<th>$d$</th>
<th>$k$</th>
<th>$n$</th>
<th>$Q_1 + I_1$</th>
<th>$Q_2 + I_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>6</td>
<td>1800</td>
<td>-0.282</td>
<td>0.310</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>5</td>
<td>1800</td>
<td>-0.887</td>
<td>0.191</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>6</td>
<td>600</td>
<td>-0.223</td>
<td>0.966</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>8</td>
<td>2048</td>
<td>-0.088</td>
<td>1.014</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2</td>
<td>100310</td>
<td>1.152</td>
<td>1.256</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>5</td>
<td>1800</td>
<td>0.167</td>
<td>0.555</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>6</td>
<td>600</td>
<td>0.009</td>
<td>1.139</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>6</td>
<td>1800</td>
<td>-0.717</td>
<td>0.070</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>1000000</td>
<td>-0.825</td>
<td>-5.23</td>
</tr>
</tbody>
</table>

to generate additional points in each cluster. Dataset 5 was formed by adding to the Vertebral Column dataset (from the UCI machine learning repository [38]) a second cluster which is a 6d hyper-sphere. Datasets 6-9 are similar to datasets 1-4 with some of the dimensions more highly expressed for some of the clusters. For these 9 datasets, the average improvement in $Q + I$ for calculating barycenters in the $\eta$-space over calculating barycenters in the data space is 0.74. Furthermore, for every dataset $Q_2 + I_2 > Q_1 + I_1$.

Figure 4 shows the same dataset (row 2 in Table 1) as Figure 1 in Section 1.1. In Figure 4 the representatives for the heuristic were the cluster barycenters in the $\eta$-space. In Figure 1 the barycenters of the clusters in the data space were used. Note that all 5 clusters are now visible and 2 are almost out of the region for the DA barycenter. Figures 5-8 show promising results for datasets 8 and 9.

5. Conclusion and Future Work

This paper has presented a barycentric approach for visualizing clustered datasets within the context of the RadViz visualization tool. Barycenters represent the clusters and the barycenter of the dimensional anchors is a key part of our dimensional anchor placement heuristic. We show where in the 2-stage RadViz pipeline the cluster barycenters are formed makes a difference, and, guided by the nature of part of the RadViz mapping, we form the barycenters at the end of the first stage.

Section 1.1 alluded to difficulty in preserving cluster separation for elongated or very irregularly shaped clusters.
One way to cope with that would be to assign a set of representatives for each of the clusters. Another intriguing direction for future work might be to use other NRV visualization invariants, such as convexity preservation, to guide the design decisions within the 2-stage visualization process.

References


