Supplementary material of “Archetypoids: a new approach to define representative archetypal data”✩

Guillermo Vinuéa, Irene Epifanionb,*, Sandra Alemanyc

aDepartment of Statistics and O.R., University of Valencia, 46100 Burjassot, Spain
bDept. Matemàtiques. Campus del Riu Sec. Universitat Jaume I, 12071 Castelló, Spain
cBiomechanics Institute of Valencia, 46022 Valencia, Spain

Abstract
In this document, we show more details and results from the paper ‘Archetypoids: a new approach to define representative archetypal data’.

1. Outline of the PAM algorithm

PAM algorithm: The goal of $k$-medoids is to minimize the average dissimilarity (cost) of objects to their closest selected object.

1. BUILD phase: look for a good initial set of $k$ objects, the medoids, from the $n$ data points by successive selection of objects.
   (a) The first object is the one for which the sum of the dissimilarities to all other objects is as small as possible (the most centrally located).
   (b) Subsequently, at each step another object is selected. This object is the one which decreases the objective function as much as possible. This step is performed until $k$ objects have been selected.

2. SWAP phase: For each medoid $m$
   (a) For each non-medoid data point $o$
i. Swap $m$ and $o$ and compute the total cost of the configuration.

3. Select the configuration with the lowest cost.

4. Repeat steps 2 to 4 until there is no change in the medoids.

2. Expanded analysis of Example 2 of the paper

We have repeated 100 times the sample generation of Example 2 and have now discarded any points outside the 95% density contour, that is, with Mahalanobis distance $\geq \chi^2_{2}(0.95)$, plotting the archetypes and archetypoids beginning with $cand_{as}$ with $k = 4$ in Fig. 1a, as Cutler and Breiman (1994) did. We can see that like archetypes, the archetypoids cluster around the ends of the major and minor axes. We have overlaid the 100 convex hulls of the 4 archetypoids (for archetypes the figure is nearly identical and is therefore omitted). The image can be seen in Fig. 1b; the whiter it is, the larger the overlap. We have also added the ellipse corresponding to the 95% and 50% probability regions for the bivariate Normal distribution to understand the result better. Fig. 1c shows the centroids (the means obtained by $k$-means with $k = 4$) and the four medoids (the points obtained with PAM) with the same data used for obtaining Fig. 1a. The archetypes and archetypoids are also displayed as in Fig. 1a in order to show that the clustering algorithms choose the prototypes in the middle of the data cloud, not on the boundary of the data.

3. Stability

In order to check the stability (if the solution does not change much when the data are modified slightly), we consider the data in Fig. 3b in the paper leaving out one point. We compute the archetypoids beginning with $cand_{as}$ and medoids with $k = 4$ and we have displayed the number of times that each point appears as an archetypoid or medoid. The results are shown in Fig. 2a and Fig. 2b, respectively. We can see that archetypoids are very stable. Three points are archetypoids 49 times (every time except the one that is left out), and the other point is an archetypoid 48 times. The frequencies of the most repeated medoids are 44, 43, 31 and 28. Although we omit the figures for archetypes, these are also very stable, especially when compared with centroids of $k$-means.
Figure 1: (a) Location of the 4 archetypes (black crosses) and archetypoids (black solid circles) for 100 simulated Bivariate Normal Data; (b) their overlapped convex hulls with contour lines for 95% (green dashed line) and 50% (blue dot-dashed line) probabilities and (c) the location of the 4 centroids (red crosses) and 4 medoids (red solid circles).

Figure 2: Frequencies of the points obtained when one point is left out for the simulated Bivariate Normal data, for (a) 4 archetypoids and (b) 4 medoids, respectively.

4. Simulation when features are unavailable

In this section, we provide two simulation studies of the methodology introduced in Sect. 2.3. of the paper, to check its performance in different
scenarios. In the first, the distance is Euclidean. In the second, we consider a situation similar to that of Sect. 4.2. of the paper, where the distance is not Euclidean, in order to determine which MDS method is more appropriate in this situation.

4.1. First experiment

We carried out a small simulation experiment with three different distributions. We used: 1) the same data as in Fig. 3b of the paper (Bivariate Normal data), 2) we generated 50 samples from a uniform in the square with corners at (1, 1), (-1, 1), (-1, -1), and (1, -1), and 3) 50 samples from two independent exponentials with rate = 1. We computed the Euclidean distance between the rows of each data matrix. Finally, we computed the archetypoids (beginning from cand$_a$) for the raw data, and the archetypoids (beginning from cand$_a$) from the 2D cMDS representation of the distances. They coincide with the combinatorial solutions. As mentioned in the paper, with Euclidean distances we can recover the original information (up to location, rotations and reflections) (Mardia, Kent, and Bibby (1979, Theorem 14.4.1)). Therefore, we obtained the same archetypoids from both configurations for the three samples, as expected. The results are displayed in Figs. 3, 4 and 5. We obtained the same archetypoids beginning with cand$_\beta$ and cand$_{ns}$, except in the case of Fig. 4a. Note that if the distances are Euclidean, the pairwise distances can be converted to centered inner-products, and the RSS can also be expressed in terms of centered inner-products. Thus cMDS may not be necessary in this case.

4.2. Second experiment

We built three binary images with dimensions $24 \times 24$ (576 variables). Each of these images resemble, respectively: an hourglass, a rectangle (a cylinder seen in 2D) and a bell. The choice of these figures to represent women’s trunks in 2D is based on a simplification of the relationship between bust, waist and hip circumferences. The top, central and bottom part of each figure corresponds to the bust, waist and hip, respectively. In the first, the bust and hip circumferences are similar, but the waist circumference is smaller. In the second, the three dimensions are similar, while in the third, the bust and waist circumferences are similar, but the hip circumference is greater. These morphotypes are compatible with the experimental studies carried out by Alemany et al. (2010). These morphotypes are displayed in Fig. 6.
Figure 3: Archetypoids (with the number in the sample) for simulated Bivariate Normal data, with $k = 4$, for (a) the raw data and (b) the 2D cMDS representation of the Euclidean distances.

Figure 4: Archetypoids (with the number in the sample) for simulated Uniform data, with $k = 4$, for (a) the raw data and (b) the 2D cMDS representation of the Euclidean distances.
Figure 5: Archetypoids (with the number in the sample) for simulated Exponential data, with \( k = 4 \), for (a) the raw data and (b) the 2D cMDS representation of the Euclidean distances.

Figure 6: Three binary archetypes resembling women’s morphotypes (as in the paper, white pixel if it belongs to the body, otherwise black).

These morphotypes will be the original archetypes \( \mathbf{A} = [m_1, m_2, m_3] \) from which we will generate data, where \( m_i \) represents each morphotype converted to a 576 dimensional vector, with \( i = 1, 2 \) and 3. Given the archetypes, we generated \( n = 100 \) observations as the binarized version of \( x_i = \mathbf{A}_i \mathbf{h}_i + \mathbf{E}_i \), where \( \mathbf{A}_i \) contains the archetypes after adding salt and pepper noise to them,
h_i is a random vector sampled from a Dirichlet distribution with α = (0.8, 0.8, 0.8), and E_i is a 576 dimensional random vector of Gaussian white noise with a mean of zero and standard deviation of 0.1. The binarized version is obtained by replacing all values above 0.5 with 1 and others with 0. The noise density added to A is 0.05 (the default value used in MATLAB), this affects approximately 5% of the pixels. We decided to add salt and pepper noise to the original archetypes to make the problem more difficult (as in real life). With salt and pepper noise a certain amount of the pixels in the image are changed to either black or white. To ensure that A_i’s are archetypes, we chose α = 0.8, a value near to but less than one. We used a standard deviation of 0.1 for the Gaussian noise, since it is the default value when adding Gaussian white noise to an image in MATLAB. A sample of ten generated observations appears in Fig. 7.

![Sample of ten generated observations.](image)

We computed the distance between each pair of observations with the misclassification error explained in Sect. 4.2 of the paper. We considered a total of seven MDS methods. In particular, three classical multidimensional scaling methods (Hastie et al., 2009; Venables and Ripley, 2002): Classical (Metric) Multidimensional Scaling (cMDS), Kruskal’s Non-metric Multidimensional Scaling (isoMDS) and Sammon’s Non-Linear Mapping (Sammon); and three manifold learning techniques: Isomap (Tenenbaum et al., 2000;
Oksanen et al., 2013), Locally Linear Embedding (LLE) (Roweis and Saul, 2000; Diedrich and Abel, 2012), and the Diffusion Map (DIF) (Nadler et al., 2006; Richards, 2014). We also considered the h-plot (Epifanio (2013)), a recent alternative method, which is especially suitable for nonmetric dissimilarity matrices, even with asymmetric proximity data (Epifanio (2014)). Note that the distance matrix $D$ with elements $d_{ij}$ obtained with the misclassification error is not Euclidean. Let $M$ be a matrix with elements $m_{ij} = -0.5 * d_{ij}^2$, and $B = (I - n^{-1}ee')M(I - n^{-1}ee')$, where $I$ is the $n \times n$ identity matrix and $e$ is the $n \times 1$ vector with all its elements equal to unity. A distance matrix $D$ is Euclidean, only if $B$ is positive semidefinite (Mardia et al., 1979, Theorem 14.2.1). For this distance matrix $D$, the last 38 eigenvalues of $B$ are negative, but they are small (the smallest is -0.1099689927) in relation to the first two eigenvalues (2.1389460120, 0.4942140256).

We calculated the cMDS representation. The goodness-of-fit (GOF) measure $\alpha_2$ proposed by (Mardia et al., 1979, eq. 14.4.8) is 99.39\% for $m = 2$ (94.35\% for $m = 1$). The goodness-of-fit measure for h-plotting (Epifanio (2013)) for $m = 2$ is 99.96\% (93.84\% for $m = 1$). So, we considered $m = 2$ for the MDS methods. The manifold learning methods need several parameters: we used their default values. For the number of neighbors, which has to be specified, we used the function calc$k$ from Diedrich and Abel (2012), which gives 9 neighbors. After computing the MDS representations, we applied the archetypoids algorithm with $k = 3$. We also computed the archetypoids considering the raw data (RAW), with the 576 variables. We computed the misclassification errors between the archetypoids and the original morphotypes in Fig. 6. We also computed the minimum misclassification errors (MM) between the whole data and the original morphotypes, which was the minimum distance that can be expected for those data. The misclassification errors appear in Table 1, while the archetypoids obtained for each method are displayed in Figs. 8 and 9 (we only show the results from cand$a$ since they are identical to those from cand$na$ and cand$b$ for all the methods). As archetypoids do not necessarily appear in the same order as morphotypes, we have permuted the archetypoids obtained in order to return the minimum distance to each morphotype.

The nearest observation to the original morphotype 1 is recovered only with h-plot. The second nearest observation to morphotype 1 is recovered by RAW. The nearest observation to morphotype 2 is not recovered by any method, although the second nearest observation is recovered with RAW, cMDS, Sammon and h-plot. The nearest observation to morphotype 3 is
recovered by RAW and LLE. The second nearest observation to the morphotype 3 is recovered by h-plot. It can be observed that h-plot is able to match morphotypes better than the other MDS methods, and its performance is comparable with that of using raw data. Note that 2 variables are used in computing the archetypoids with h-plot, whereas 576 variables are used with raw data.

This experiment was replicated 50 times, and the mean misclassification errors and their standard deviations (SD) for each morphotype are shown in Table 2. We did not replicate the experiment for Isomap, because for several replications the dissimilarity matrix is fragmented, and therefore only the largest connected group would be analyzed, and its results were the worst by

Table 1: Distances between archetypoids obtained for each method and the original morphotypes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Morphotype 1</th>
<th>Morphotype 2</th>
<th>Morphotype 3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>0.01562500</td>
<td>0.01736111</td>
<td>0.01041667</td>
<td>0.01446759</td>
</tr>
<tr>
<td>RAW</td>
<td>0.01736111</td>
<td>0.01909722</td>
<td>0.01041667</td>
<td>0.01562500</td>
</tr>
<tr>
<td>cMDS</td>
<td>0.03819444</td>
<td>0.01909722</td>
<td>0.02636167</td>
<td>0.02777778</td>
</tr>
<tr>
<td>Sammon</td>
<td>0.03819444</td>
<td>0.01909722</td>
<td>0.04861111</td>
<td>0.03530926</td>
</tr>
<tr>
<td>IsoMDS</td>
<td>0.03819444</td>
<td>0.03472222</td>
<td>0.04861111</td>
<td>0.04050926</td>
</tr>
<tr>
<td>LLE</td>
<td>0.11111111</td>
<td>0.15104167</td>
<td>0.06423611</td>
<td>0.10879630</td>
</tr>
<tr>
<td>DIF</td>
<td>0.03819444</td>
<td>0.03472222</td>
<td>0.04861111</td>
<td>0.04050926</td>
</tr>
<tr>
<td>h-plot</td>
<td>0.01562500</td>
<td>0.01909722</td>
<td>0.01909722</td>
<td>0.01793981</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Morph. 1</th>
<th>Morph. 2</th>
<th>Morph. 3</th>
<th>Total Average (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>0.01597222</td>
<td>0.01548611</td>
<td>0.01517361</td>
<td>0.01554398 (0.00345667)</td>
</tr>
<tr>
<td>RAW</td>
<td>0.01711806</td>
<td>0.01663194</td>
<td>0.01555556</td>
<td>0.01643519 (0.00381990)</td>
</tr>
<tr>
<td>cMDS</td>
<td>0.02996528</td>
<td>0.02680556</td>
<td>0.03038194</td>
<td>0.02905093 (0.00622453)</td>
</tr>
<tr>
<td>Sammon</td>
<td>0.03281250</td>
<td>0.02878472</td>
<td>0.03569444</td>
<td>0.03243056 (0.00742946)</td>
</tr>
<tr>
<td>IsoMDS</td>
<td>0.03906250</td>
<td>0.03166667</td>
<td>0.03517361</td>
<td>0.0341435 (0.0214994)</td>
</tr>
<tr>
<td>LLE</td>
<td>0.02989583</td>
<td>0.03517361</td>
<td>0.03517361</td>
<td>0.03275463 (0.06374894)</td>
</tr>
<tr>
<td>h-plot</td>
<td>0.02069444</td>
<td>0.01961806</td>
<td>0.01934028</td>
<td>0.01988426 (0.00522030)</td>
</tr>
</tbody>
</table>

Table 2: Averages of distances between archetypoids obtained from $c_{n}$ for each method and the original morphotypes and standard deviation between parentheses, when the 2nd experiment is replicated 50 times.
As before, h-plot was able to match morphotypes much better than the other MDS methods, and its performance is comparable with that of using raw data and close to the minimum possible number of misclassification errors (MM).

Based on these results, we used h-plot for the data in Sect. 4.2 of the paper. Note that each binary trunk is a 3D image with dimensions $316 \times 323 \times 383$, corresponding with width, depth and height, respectively. These dimensions give $316 \times 323 \times 383 = 39,092,044$ variables, i.e., nearly 40 million variables. A matrix of 470 women each with 40 million variables is not operational (we cannot compute archetypes with raw data). Therefore,
we computed $D$, and h-plot was used with $m = 4$, since the goodness-of-fit measure for h-plotting (Epifanio (2013)) for $m = 4$ is 99.45% (65.65%, 93.98%, 98.96% for $m = 1, 2$ and 3 respectively). For this distance matrix $D$, the last 277 eigenvalues of $B$ are negative, but they are small (the smallest is $-3.323756\text{e-}05$) in relation to the first four eigenvalues ($7.496575\text{e-}04$, $4.528909\text{e-}04$, $2.295239\text{e-}04$, $2.243783\text{e-}04$).

In conclusion, ADA can be applied after using a MDS method. The problem of which MDS method should be chosen is not exclusive to this procedure (ADA can be used after any MDS method returning a matrix representation). It is a problem found in the MDS literature. We have searched references about comparison of the established MDS methods, or criteria about when to
use one MDS method or another, but we have not found any exhaustive work about this point. We see this as quite a difficult point because it depends a lot on the structure of the dissimilarity matrix and it is not easy to model all the possible structures. We believe it is impossible to tell a priori which the most appropriate method is in general. However, we can give certain guidelines. When we have a metric dissimilarity matrix and the objective is to preserve all the interpoint distances, cMDS is the right method. However, the first dimensions are the common representation (not all the dimensions are usually considered) and they may not contain all the information. The other methods have different objectives. For example, Sammon seeks to preserve the smallest distances, which is appropriate for clustering. IsoMDS seeks to preserve the rank-order of the dissimilarities. Isomap and other manifold learning techniques resemble Sammon in a broad sense, because retaining the neighborhood structure is fundamental since their objective is to recover the representation of a nonlinear manifold. The goal of the h-plot is not to reproduce the interpoint distances precisely, or to put more weight on small distances or neighbors. H-plots seeks to preserve relationships between dissimilarity variables. This aspect is especially relevant when working with nonmetric dissimilarities (although also when working with metric dissimilarities, as not all the information included in the dissimilarity matrix can usually be gathered in the first dimensions), because in this situation the dissimilarities cannot be represented exactly in a Euclidean space, since the matrix is not Euclidean.

In summary, when only the dissimilarity matrix is available, a simulation study can be performed before applying ADA, in order to chose an appropriate MDS method. If the simulation study cannot be carried out, cMDS is a reasonable option if the distance is Euclidean or nearly Euclidean, or otherwise h-plot.

5. The six cockpit dimensions

The description of the variables used in the cockpit design problem is given in Table 3, as in Epifanio et al. (2013).
Table 3: Description of the six variables considered.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thumb Tip Reach</td>
<td>Measure the distance from the wall to the tip of the thumb.</td>
</tr>
<tr>
<td>Buttock-Knee Length</td>
<td>Measure the horizontal distance from the rearmost surface of the right buttock to the forward surface of the right kneecap.</td>
</tr>
<tr>
<td>Popliteal Height Sitting</td>
<td>Measure the vertical distance from the footrest surface to the superior margin of the right kneecap.</td>
</tr>
<tr>
<td>Sitting Height</td>
<td>Measure the vertical distance from the sitting surface to the top of the head.</td>
</tr>
<tr>
<td>Eye Height Sitting</td>
<td>Measure the vertical distance from the sitting surface to the right external canthus (outer “corner” of eye).</td>
</tr>
<tr>
<td>Shoulder Height Sitting</td>
<td>Measure the vertical distance from the sitting surface to the right Acromion - the bony landmark at the tip of the shoulder.</td>
</tr>
</tbody>
</table>

The skeleton of an aircraft pilot together with the six variables is shown in Fig. 10.

Figure 10: Skeleton for an aircraft pilot.

6. Comparison of the computational costs

We have considered the database used in the cockpit design problem, whose sample size is the biggest. For the first twenty individuals of the database, our algorithm provides the results automatically, while the genetic algorithm in the Matlab implementation takes just a few seconds (although
its results do not verify the equations of the archetypoid analysis problem. On the other hand, the branch and bound algorithm in the Matlab implementation takes six hours. The exhaustive search (combinatorial or true solution) takes less than 5 seconds (much less than the branch and bound implementation). SiVM, the greedy algorithm, returns the results automatically. When the first two hundred observations are selected, our algorithm only takes around forty seconds for obtaining the archetypoids from $cand_a$, included the previous computation of the archetypes for the building phase (the time is similar beginning from $cand_{ns}$). SiVM returns again the results automatically. However, the genetic algorithm takes ten minutes (and its results do not verify the equations of the archetypoid problem either), and the exhaustive search takes eight hours. The branch and bound implementation took so much time to compute only that small sample, that it is pointless to use it with this bigger database. As a matter of fact, when the sample size gets larger, the genetic algorithm and the exhaustive search increases its computational time seriously, but our algorithm continues performing well. Thus, our algorithm represents a noticeable reduction in the computation involved. Although SiVM returns the results (individuals with numbers: 189 110 39) automatically, neither of them coincide with the true solution (individuals with numbers: 10 51 85). On the contrary, our algorithm returns the true solution as archetypoids. Note that the number of combinations is around 1 million, but when the number of combinations increases to 17 million in the apparel design problem, the best solution was obtained after 20 days of computation, using exhaustive search.

The speed of our algorithm implementation depends on the efficiency of the convex least squares method as happened in the archetype algorithm implemented in Eugster and Leisch (2009). According to Cutler and Breiman (1994) the penalized non-negative least squares method is quite slow, but can be used if the number of variables is larger than the number of observations.

References


URL http://CRAN.R-project.org/package=lle


URL http://www.jstatsoft.org/


URL http://CRAN.R-project.org/package=vegan

URL http://CRAN.R-project.org/package=diffusionMap
