

# Euclidean Cloud

*from Brigitte Le Roux and Henry Rouanet*

*November 6, 2015*

This vignette follows the treatment in Chapter 3 of [Le Roux and Rouanet's \*Geometric Data Analysis\*](#), with drastically simplified exposition but employing the same example data.

The central object of GDA is a point cloud in a Euclidean space, which encodes a set of statistical observations arranged in a table, or two-dimensional array (for instance, a contingency table in Correspondence Analysis or an Individuals–Variables table in Principal Components Analysis). By convention, the columns of the table correspond to the coordinate axes of the Euclidean space while the rows correspond to the points.

## 3.1 Basic Statistics

Rigorously, a *point cloud* in a Euclidean space  $\mathcal{U}$  consists of a set  $J$  of labels and a mapping  $M^J : J \rightarrow \mathcal{U}$  that takes each label  $j \in J$  to its corresponding point  $M^j \in \mathcal{U}$ . The points are assigned positive weights  $\omega_j$ , which may be assumed unitary when not specified. The most common weights will be absolute frequencies  $n_j$ ,  $n = \sum_{j \in J} n_j$ , and their associated relative frequencies  $f_j = \frac{n_j}{n}$ .

As a running example, we load the *Target example* dataset, used throughout the chapter, whose coordinates are provided in Exercise 3.4:

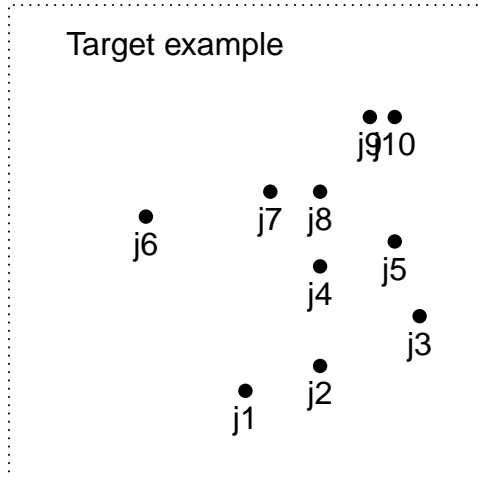
```
data(Target)
print(Target)
```

```
##      x1  x2
## j1    0 -12
## j2    6 -10
## j3   14  -6
## j4    6  -2
## j5   12   0
## j6   -8   2
## j7    2   4
## j8    6   4
## j9   10  10
## j10  12  10
```

```
print(class(Target))
```

```
## [1] "matrix"
```

```
T_freq <- rep(1 / nrow(Target), nrow(Target))
```



The labels consist of  $j_1$  through  $j_{10}$ . Because these data occupy two dimensions, i.e. the labels are mapped into  $\mathcal{U} = \mathbb{R}^2$ , they constitute a *plane cloud*.

### 3.1.1 Mean Point

The *mean point* of the cloud  $M^J$  in  $\mathcal{U}$  is meant to be a proxy point location for the entire cloud. Given an arbitrary point  $P \in \mathcal{U}$ , the mean point is defined as the point arrived to from  $P$  via the sum of the weighted vectors  $f_j \overrightarrow{PM^j}$  from  $P$  to each point  $M^j$ —that is, as the (unique!) point  $G \in \mathcal{U}$  for which

$$\overrightarrow{PG} = \sum_{j \in J} f_j \overrightarrow{PM^j}$$

doesn't depend on the choice of point  $P \in \mathcal{U}$ . The mean point satisfies the *Barycentric property* that

$$\sum_{j \in J} f_j \overrightarrow{GM^j} = \vec{0},$$

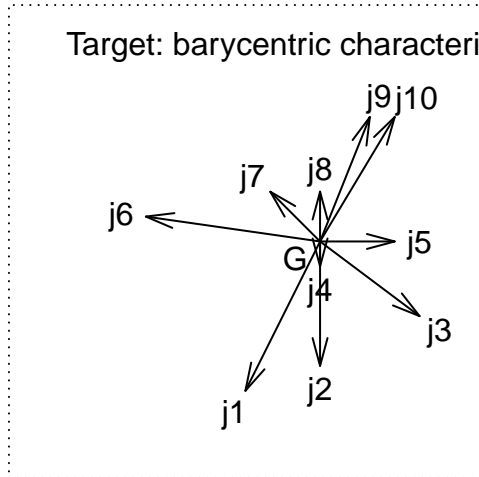
which can be checked by substituting  $P = G$  in the definition.<sup>1</sup>

```
G <- barycenter(Target)
print(G)
```

```
##      [,1] [,2]
## [1,]    6    0
```

---

<sup>1</sup>These and other formulas are given in more general terms in *GDA*, in which the points  $M^j$  are weighted by masses  $\omega_j$ . I may be forced to adopt this generality, but i haven't yet.



### 3.1.2 Inertia, Sum of Squares, Variance and Contributions

The *inertia* of a cloud  $M^J$  with respect to a point  $P$  is defined, as in other settings, as the sum-squared distance of each point  $M^j$  from  $P$ :

$$In^P M^J = \sum_{j \in J} In_j^P = \sum_{j \in J} \omega_j (PM^j)^2.$$

The inertia for  $\omega_j = n_j$  is the *sum of squares* and that for  $\omega_j = f_j$  is the *mean of squares*. The *variance* of the cloud is its mean of squares from its barycenter, and can be interpreted as the sum of the *absolute contributions* of the individual points in the cloud:

$$Var M^J = \sum_{j \in J} Cta_j = \sum_{j \in J} f_j (GM^j)^2.$$

The \*relative contributions of the points are, then, the fractions  $Ctr_j = \frac{Cta_j}{Var M^J}$ .

```
P <- c(0, 0)
P_in <- inertia(point = P, cloud = Target)
G_in <- inertia(point = G, cloud = Target)
T_var <- inertia(G, cloud = Target,
                weights = T_freq)
print(c(center_inertia = P_in,
        barycenter_inertia = G_in,
        variance = T_var))
```

```
##      center_inertia barycenter_inertia      variance
##                1280                920                92
```

```
Ctas <- T_freq * apply(Target, 1, function(m) {
  sum((m - G) ^ 2)
})
Ctrs <- Ctas / T_var
print(cbind(Ctas, Ctrs))
```

```
##      Ctas      Ctrs
```

```
## j1 18.0 0.195652174
## j2 10.0 0.108695652
## j3 10.0 0.108695652
## j4 0.4 0.004347826
## j5 3.6 0.039130435
## j6 20.0 0.217391304
## j7 3.2 0.034782609
## j8 1.6 0.017391304
## j9 11.6 0.126086957
## j10 13.6 0.147826087
```

The **first Huyghen's theorem** states that the inertia of a cloud  $M^J$  with respect to a point  $P$  can be decomposed into the sum of the variance of the cloud (its inertia with respect to its barycenter  $G$ ) and the squared distance between  $G$  and  $P$ :

$$\sum_{j \in J} f_j (PM^j)^2 = (PG)^2 + \sum_{j \in J} f_j (GM^j)^2.$$

(See the illustration below.) A consequence is the *metric characterization of the barycenter*, which has that the barycenter minimizes the mean of squares of the cloud. That is, across  $P \in \mathcal{U}$ , the quantity  $\sum_{j \in J} f_j (PM^j)^2$  is minimized when  $P = G$ .

```
huyghen_test(point = P, cloud = Target, weights = T_freq)
```

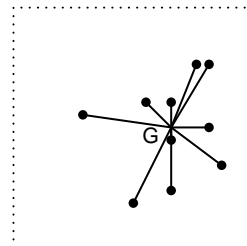
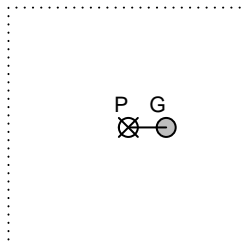
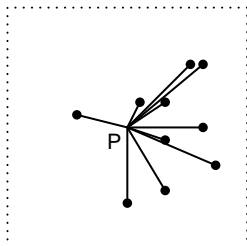
```
## [1] TRUE
```

```
## $lhs
## [1] 128
##
## $rhs
## [1] 128
```

```
huyghen_test(point = G, cloud = Target, weights = T_freq)
```

```
## [1] TRUE
```

```
## $lhs
## [1] 92
##
## $rhs
## [1] 92
```



## 3.2 Projected Clouds

The (*orthogonal*) *projection* of a point cloud  $M^J$  in a space  $\mathcal{U}$  onto a subspace  $\mathcal{H}$  is the cloud  $H^J = \{H^j\}_{j \in J}$  consisting of its points' projections, and called the *projected cloud*. An important property of the projected cloud is that its barycenter  $G'$  is the projection of the barycenter  $G$  of  $M^J$ .

```
axis <- rbind(c(0, 0), c(2, 1))
print(affine_projection(cloud = Target, subspace = axis))
```

```
##      [,1] [,2]
## j1 -4.8 -2.4
## j2  0.8  0.4
## j3  8.8  4.4
## j4  4.0  2.0
## j5  9.6  4.8
## j6 -5.6 -2.8
## j7  3.2  1.6
## j8  6.4  3.2
## j9 12.0  6.0
## j10 13.6  6.8
```

```
projection_test(cloud = Target, subspace = axis)
```

```
## [1] TRUE
```

```
## $lhs
##      [,1] [,2]
## [1,]  4.8  2.4
##
## $rhs
##      [,1] [,2]
## [1,]  4.8  2.4
```

### 3.2.1 Variance in a Direction

The variances of the projected clouds of  $M^J$  onto two equidimensional parallel subspaces  $\mathcal{H}$  and  $\mathcal{H}'$  are equal, and are called the *variance in the direction* of  $\mathcal{H}$  (or of  $\mathcal{H}'$ ), here denoted  $\text{Var}_{\mathcal{H}} M^J$ . A special case is the variance in the direction of a line, equal to

$$\text{Var}_{\vec{\alpha}} M^J = \sum_{j \in J} f_j \frac{\langle \overrightarrow{GM}^j \mid \vec{\alpha} \rangle^2}{\|\vec{\alpha}\|^2}$$

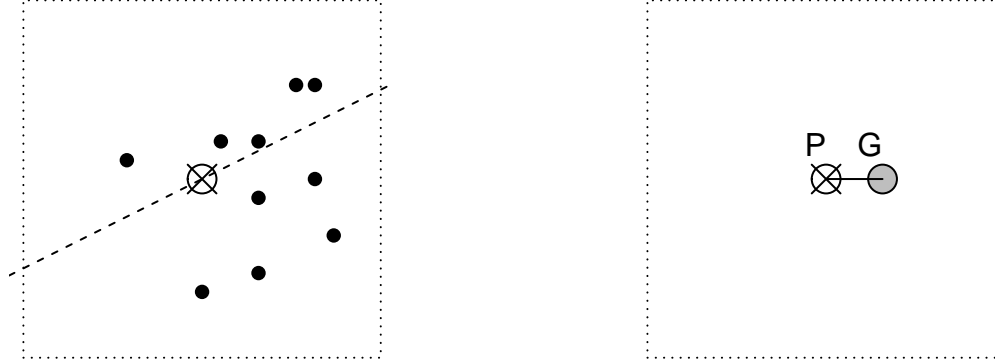
and called the *variance of axis*.

```
axis_var <- direction_variance(cloud = Target, weights = T_freq,
                              subspace = axis)
print(axis_var)
```

```
## [1] 48.8
```

```
print(direction_variance(cloud = Target, weights = T_freq,
                        subspace = affine_decomposition(axis)$linear_subspace,
                        type = "linear"))
```

```
## [1] 48.8
```



### 3.2.2 Residual Square Mean

The first Huyghen's theorem can be extended from “point” variance to directional variance by way of projection: Given a cloud  $M^J$  and a subspace  $\mathcal{H}$ , the *residual square mean*

$$\sum_{j \in J} f_j (M^j H^j)^2$$

of  $M^J$  with respect to  $\mathcal{H}$  is the weighted mean of the squares of the “residuals”  $M^j H^j$ —the distances from the points of  $M^J$  to their projections in  $\mathcal{H}^J$ . The **general Huyghen's theorem** then states that the residual square mean with respect to  $\mathcal{H}$  can be decomposed into the sum of the residual square mean with respect to a same-dimensional parallel space  $\mathcal{H}'$  through the barycenter  $G$  of  $M^J$  and the squared distance from  $G$  to  $\mathcal{H}$ :

$$\sum_{j \in J} f_j (M^j H^j)^2 = \sum_{j \in J} f_j (M^j A^j)^2 + (GG')^2,$$

where  $G'$  is the projection of  $G$  onto  $\mathcal{H}$ .

```
huyghen_general_test(cloud = Target, weights = T_freq, subspace = axis)
```

```
## [1] TRUE
```

```
## $lhs
```

```
## [1] 50.4
```

```
##
```

```
## $rhs
```

```
## [1] 50.4
```

### 3.2.3 Fitted and Residual Clouds

When a projected cloud  $A^J$  (in a subspace  $\mathcal{A}$ ) is used to model its preimage  $M^J$ , it is called a *fitted cloud*, and the vectors  $\overrightarrow{A^j M^j}$ , whose lengths are the residuals mentioned above, are called *residual deviations*. This leads to the geometric data–model–error formulation

$$\forall j \in J \quad M^j = A^j + \overrightarrow{A^j M^j},$$

where each  $\overrightarrow{A^j M^j}$  is orthogonal to  $\mathcal{A}$ . Consequently, the cloud  $R^J$  consisting of points  $R^j = G' + \overrightarrow{H^j M^j}$  (where  $G'$ , as above, is the barycenter of  $H^J$ ) depends only on the choice of  $\mathcal{H}$ .  $R^J$  is called the *residual cloud*, and it is constructed so that  $G$  is its barycenter. This yields the orthogonal decomposition

$$\overrightarrow{GM^j} = \overrightarrow{GA^j} + \overrightarrow{GR^j}.$$

It is straightforward to check that the variance of  $R^J$  equals the residual square mean of  $M^J$  with respect to  $\mathcal{A}$ . The distance formula, applied pointwise to the orthogonal decomposition, then yields the *variance decomposition*

$$\text{Var} M^J = \text{Var} A^J + \text{Var} R^J$$

into the cloud's *total variance*, its *fitted variance*, and its *residual variance*.

```
cloud_decomposition(cloud = Target, subspace = axis)
```

```
## $fitted_cloud
##      [,1] [,2]
## j1  -4.8 -2.4
## j2   0.8  0.4
## j3   8.8  4.4
## j4   4.0  2.0
## j5   9.6  4.8
## j6  -5.6 -2.8
## j7   3.2  1.6
## j8   6.4  3.2
## j9  12.0  6.0
## j10 13.6  6.8
##
## $residual_cloud
##      x1    x2
## j1  10.8 -9.6
## j2  11.2 -10.4
## j3  11.2 -10.4
## j4   8.0 -4.0
## j5   8.4 -4.8
## j6   3.6  4.8
## j7   4.8  2.4
## j8   5.6  0.8
## j9   4.0  4.0
## j10  4.4  3.2
```

```
cloud_variance_test(cloud = Target, weights = T_freq, subspace = axis)
```

```
## [1] TRUE

## $lhs
## [1] 92
##
## $rhs
## [1] 92
```

### 3.2.4 Variables Attached to an Axis

The linear formulation in the next section relies on several coordinatization schemes. Begin with an axis  $(G, \vec{\alpha})$ , through the barycenter of  $M^J$  in the direction of  $\vec{\alpha}$ , and the projected cloud  $A^J$  onto this axis. We then define four centered (having mean point zero) variables:

- The *covariant variable*  $\alpha^J = (\alpha^j)_{j \in J}$  has *covariant coordinates*  $\alpha^j = \langle \overrightarrow{GM^j} \mid \vec{\alpha} \rangle$  and variance  $Var A^J \|\vec{\alpha}\|^2$ . The term refers to the fact that these coordinates scale with  $\vec{\alpha}$ .
- The *calibrated variable*  $y^J = (y^j)_{j \in J}$  has *calibrated coordinates*  $y^j = \alpha^j / \|\vec{\alpha}\|$  and variance  $Var A^J$ .
- The *standard variable*  $z^J = (z^j)_{j \in J}$  has *standard coordinates*  $z^j = y^j / SD y^J$  and variance 1.
- The *axial variable*  $t^J = (t^j)_{j \in J}$  has *axial coordinates*  $t^j = \langle \overrightarrow{GM^j} \mid \vec{\alpha} \rangle / \|\vec{\alpha}\|^2$  and variance  $Var t^J = A^J / \|\vec{\alpha}\|^2$ . The  $t^j$  are the projections—GDA calls them the coordinates—of the  $A^j$  onto  $(G, \vec{\alpha})$ .

```
alpha <- affine_decomposition(axis)$linear_subspace
coords <- sapply(c("covariant", "calibrated", "standard", "axial"),
                axis_coordinates, cloud = Target, dir_vec = alpha)
print(coords)
```

```
##      covariant calibrated standard axial
## [1,]      -24 -10.7331263 -1.4575974  -4.8
## [2,]      -10  -4.4721360 -0.6073322  -2.0
## [3,]       10   4.4721360  0.6073322   2.0
## [4,]       -2  -0.8944272 -0.1214664  -0.4
## [5,]       12   5.3665631  0.7287987   2.4
## [6,]      -26 -11.6275535 -1.5790638  -5.2
## [7,]       -4  -1.7888544 -0.2429329  -0.8
## [8,]        4   1.7888544  0.2429329   0.8
## [9,]       18   8.0498447  1.0931980   3.6
## [10,]      22   9.8386991  1.3361309   4.4
```

### 3.2.5 Linear Formalization

The Euclidean space  $\mathcal{U}$  has an underlying vector space  $\mathcal{V}$ . The aforescribed coordinates can be understood in terms of linear mappings (homomorphisms) between  $\mathcal{V}$  and two vector spaces derived from  $J$ : the space  $\mathbb{R}^J$  of variables over  $J$ , and the space  $\mathbb{R}_J$  of measures over  $J$ . The *variable covariant* (Vac) homomorphism sends a vector  $\vec{\alpha} \in \mathcal{V}$  to its covariant coordinates in  $\mathbb{R}^J$ ; the *effect* (Eff) homomorphism sends a measure (given as a vector of weights) in  $\mathbb{R}_J$  to its evaluation on  $M^J$  (the sum of the weighted points in the clouds, viewed as vectors). Both homomorphisms are taken with respect to a given point  $P \in \mathcal{U}$ , which by default takes the barycentric value  $P = G$ .

```
print(vac(cloud = Target, point = P, dir_vec = alpha))
```

```
##      [,1]
## j1     -12
## j2       2
## j3      22
## j4      10
## j5      24
## j6     -14
## j7       8
## j8      16
## j9      30
## j10     34
```



```
print(eff(cloud = Target, point = P, coord_vec = coords[, "covariant"]))
```

```
##      x1  x2
## [1,] 880 680
```

The coordinate-preserving isomorphism  $f_J : \mathbb{R}^J \rightarrow \mathbb{R}_J$ , which associates variables with measures, reveals a duality between  $\text{Vac}$  and  $\text{Eff}$ . Specifically, the composition homomorphism  $\text{Eff}^P \circ f_J$  is the adjoint  $\text{Vac}_P^*$  of  $\text{Vac}_P$ . As these functions are implemented here, `eff` is the adjoint of `vac`. The result then follows directly from the associativity of matrix multiplication.

These maps compose to produce symmetric endomorphisms

$$\text{Som} = \text{Eff}^P \circ f_J \circ \text{Vac}_P : \mathcal{V} \rightarrow \mathcal{V}$$

of the vector space containing  $M^J$  and

$$\text{Tom} = \text{Vac}_P \circ \text{Eff}^P \circ f_J : \mathbb{R}^J \rightarrow \mathbb{R}^J$$

of the coordinate space of  $J$ .

```
print(rbind(alpha,
             som(cloud = Target, point = P, dir_vec = alpha)))
```

```
##      x1  x2
## [1,]  2   1
## [2,] 1600 680
```

```
print(cbind(coords[, "covariant"],
             tom(cloud = Target, point = P, coord_vec = coords[, "covariant"])))
```

```
##      [,1] [,2]
## j1    -24 -8160
## j2    -10 -1520
## j3     10  8240
## j4     -2  3920
## j5     12 10560
## j6    -26 -5680
## j7     -4  4480
## j8      4  8000
## j9     18 15600
## j10    22 17360
```

These endomorphisms will be important to performing the central operation of geometric data analysis: identifying the principal directions of a point cloud. In particular, this relies on the relationship between the endomorphism  $\text{Som}$  and the directional variance of  $M^J$ :

$$\text{Var}_{\vec{\alpha}} M^J = \frac{\|\text{Vac}(\vec{\alpha})\|^2}{\|\vec{\alpha}\|^2} = \frac{\langle \text{Som}(\vec{\alpha}) | \vec{\alpha} \rangle}{\|\vec{\alpha}\|^2}$$

This property follows from the variable definitions and from the definition of  $\text{Som}$ , which implies that  $\langle \text{Som}(\vec{\alpha}) | \vec{\alpha} \rangle = \|\text{Vac}(\vec{\alpha})\|^2$ .

### 3.3 Principal Directions of a Cloud

The **problem of principal directions** is to approximate a point cloud (in a vector space of dimension, say,  $L$ ) in a vector space of lower dimension (say,  $L'$ ). This entails identifying successive directions in  $\mathcal{V}$  so that the projected cloud onto each subspace orthogonal to the first  $\ell - 1$  directions has maximum fitted variance in the  $\ell^{\text{th}}$  direction—or, equivalently (by the variance decomposition), so that each projection has minimum residual variance.

#### 3.3.1 Principal Axes

As just observed, questions of directional variance can be recast in terms of Som. Specifically,  $\|\text{Vac}(\vec{\alpha})\|/\|\vec{\alpha}\|$  is maximized in the direction of the first eigenvector of Som—or, if the first eigenvalue is multiple, in any direction of the first eigenspace. This leads to the **principal direction equation** of the direction vectors of the principal lines of  $M^J$  with the eigenvectors of Som:

$$\text{Som}(\vec{\alpha}_\ell) = \lambda_\ell \vec{\alpha}_\ell$$

The proof proceeds inductively on  $L$  and by construction along sequential dimension-one projections, with special attention given to the special case of eigenvalues of multiple order. These cases aside, the proof relies on one key observation: Let  $A_1^J$  be the projected cloud along the first eigenvector of Som, with residual cloud  $R_1^J$ . The orthogonal decomposition in Section 3.2.3 and the linearity of Vac yield the endomorphism decomposition

$$\text{Vac} = \text{Vac}' + \text{Vac}_1,$$

where  $\text{Vac}'(\vec{\alpha}) = (\overrightarrow{\langle GR_1^J \mid \vec{\alpha} \rangle})_{j \in J}$  and  $\text{Vac}_1(\vec{\alpha}) = (\overrightarrow{\langle GA_1^J \mid \vec{\alpha} \rangle})_{j \in J}$ , as well as a similar decomposition of  $\text{Vac}^*$ . These combine to give

$$\text{Som} = \text{Som}' + \text{Som}_1,$$

which, provided the nonzero eigenvectors of Som are orthonormal, partitions these eigenvectors into those of  $\text{Som}'$  ( $\ell - 1$  of them) and those of  $\text{Som}_1$  (one). The single eigenvector of  $\text{Som}_1$  has the greatest eigenvalue,  $\lambda_1$ , of the original family, so the greatest eigenvalue of  $\text{Som}'$  is  $\lambda_2$ . From there the proof continues until all dimensions are exhausted.

Whereas each step identifies a principal vector orthogonal to the space spanned by what will be the remaining principal vectors, *all of the principal vectors (hence the principal directions) are pairwise orthogonal*. Additionally, since the relationship between the residual variance and the first eigenvalue is repeated at each step, the proof provides the **principal breakdown of variance**:

$$\text{Var}M^J = \sum_{\ell=1}^L \lambda_\ell$$

Finally, though the proof involved only variances of one-dimensional directions, the result can be extended to the **hereditary property** of higher-dimensional principal directions: If  $\lambda_{L'} > \lambda_{L'-1}$  (strict inequality) for some  $1 \leq L' \leq L$ , then the **first principal  $L'$ -dimensional direction** contains all first principal lower-dimensional directions. This means that each first principal  $L'$ -dimensional direction is the span of the first  $L'$  principal one-dimensional directions (or, in case of multiple eigenvalues, the first principle lowest-dimensional directions whose dimensions sum to  $L'$ ).

#### 3.3.2 Principal Variables