SORT 49 (2) July-December 2025, 1-32 DOI:

On generalized Gower distance for mixed-type data: Extensive simulation study and new software tools

Aurea Grané and Fabio Scielzo-Ortiz

Abstract

Data scientists address real-world problems using multivariate and heterogeneous datasets, characterized by multiple variables of different natures. Selecting a suitable distance function between units is crucial, as many statistical techniques and machine learning algorithms depend on this concept. Traditional distances, such as Euclidean or Manhattan, are unsuitable for mixed-type data, and although Gower distance was designed to handle this kind of data, it may lead to suboptimal results in the presence of outlying units or underlying correlation structure. In this work robust distances for mixed-type data are defined and explored, namely robust generalized Gower and robust related metric scaling. A new Python package is developed, which enables to compute these robust proposals as well as classical ones.

MSC: *62H30, 62-04.*

Keywords: *Distances, generalized Gower, multivariate heterogeneous data, outliers, robust Mahalanobis, related metric scaling.*

1. Introduction

Data scientists often face the challenge of clustering datasets of mixed-type, that is, datasets containing both numeric and categorical variables. A common approach is to start by computing classical Gower distance (Gower, 1971) between units, next to obtain a Euclidean configuration, for instance via metric multidimensional scaling (that is, Gower's 1966 principal coordinates, Borg and Groenen, 2005), and finally to apply partitioning algorithms like k-means or k-medians onto the principal coordinates of the

Authors' address: Statistics Deparment, Universidad Carlos III de Madrid, C/ Madrid 126, 28903 Getafe, Spain. E-mails and ORCID code: A. Grané aurea.grane@uc3m.es (ORCID 0000-0003-0980-6409), F. Scielzo-Ortiz fscielzo@pa.uc3m.es.
Corresponding author: A. Grané.
Received: August 2024
Accepted: September 2025

units. Other possibilities skip the Euclidean configuration by directly applying clustering algorithms to classical Gower distance between units. This is the case for k-medoids (Kaufman and Rousseeuw, 1990) or hierarchical methods when the sample size allows it.

In such strategies, a key point is the selection of the metric, which should be able to incorporate the statistical characteristics of the data. For instance, the underlying correlation structure or outlying observations are issues that can distort the true proximity between units and that few metrics are able to consider. This may happen when using Gower distance, which is defined from Gower's similarity coefficient as the simple mean of three partial similarity indices computed from each variable type: a similarity associated with range-normalized Manhattan distance for numerical variables, Jaccard for binary variables and the simple matching coefficient for multiclass ones. Manhattan distance, like all Minkowski distances, implicitly assumes that variables are uncorrelated, and so does the Gower coefficient. Another problem is that the Manhattan distance is not robust to outlying units.

To overcome these drawbacks and inspired by Gower's work, the generalized Gower (G-Gower) distance was defined as the combination of three measures, conveniently standardized and fulfilling the Euclidean requirement (Gower and Legendre, 1986), for numerical, binary and multiclass variables (Grané, Salini and Verdolini, 2021). Indeed, G-Gower appears as a particular case when a more general technique, called related metric scaling (RelMS) (Cuadras and Fortiana, 1995; 1998), is used to tailor a metric. This technique allows combining several distance matrices computed on the same set of individuals into a single one. It has the additional property of discarding redundant information coming from different sources. When all distance matrices to be combined satisfy the Euclidean requirement, so does the final distance matrix (see Albarrán, Alonso and Grané, 2015; Grané and Romera, 2018 for the mathematical proofs).

In this paper, RelMS is used as a strategy to obtain flexible and robust distances for mixed-type data. Several proposals from the least to the greatest complexity are explored and evaluated in the context of clustering. They include three robust Mahalanobis proposals for numerical data, distances associated with Jaccard and Sokal-Michener similarity coefficients for binary data, and for multiclass variables Hamming distance is considered (which is the distance associated to the simple matching coefficient).

The performance of the new robust proposals is evaluated in six mixed-type datasets, four synthetic and two real, with underlying correlation structure and outlier contamination. In each case, k-medoids algorithm is applied to find the clusters, and comparisons with the performance of other classical metrics are provided in terms of classification rate and adjusted Rand index. A total of 34 distances are evaluated. Since some of the datasets are rather complex, metric multidimensional scaling is used to visualize and illustrate the difference between the true and assigned class of the units. A sensitivity analysis on the parameters involved in the robust estimation of the new proposals is provided for each dataset. A study of their computational cost for large and very large datasets can be found in Appendix B. Additionally, in Appendix A a Python package called robust_mixed_dist is presented.

The paper proceeds as follows. In Section 2 we revisit related metric scaling and present the generalized Gower distance as well as several robust proposals. Their performance in the context of clustering and the sensitivity analysis can be found in Section 3. Section 4 contains the main conclusions, and some guidelines on robust_mixed_dist and the study on computational cost can be found in the appendices.

2. Distance proposals for mixed-type data

In this section, a general procedure for combining distance matrices computed on the same set of units is revisited. It was used to obtain distance measures for mixed-type data in Albarrán et al. (2015), Grané and Romera (2018), Grané et al. (2021) and Boj and Grané (2024) in the context of metric multidimensional scaling and distance-based predictive models, where a robust Mahalanobis distance was used for numerical variables and for binary and multiclass data Jaccard and Hamming distances were considered, respectively. In this paper, we explore other robust proposals and provide an extensive simulation study of their performance in the context of clustering.

The strategy to construct a joint distance begins by splitting the dataset according to each variable type (numerical, binary and multiclass), next to compute different distance matrices for each variable type, and finally combine them via related metric scaling (Cuadras and Fortiana, 1995; 1998).

Let $\mathbf{X}$ be an $n \times p$ a data matrix corresponding to the measurements of p mixed-type variables $X_1, \ldots, X_p$ on a sample of n units, and consider sub-matrices $\mathbf{X}_k$ of size $n \times p_k$, $k = 1, 2, 3$, corresponding to each variable type, i.e., numeric, binary and multiclass, with $\sum_{k=1}^{3} p_k = p$. The distance measures considered are:

- Distances for numerical data: Euclidean (ℓ^2 distance), Manhattan (ℓ^1 distance), Canberra, Pearson (standardized ℓ^2 distance), Mahalanobis, robust Mahalanobis (with three variance estimators median absolute deviation, trimmed, winsorized),
- Distances for binary data: Associated with Jaccard coefficient (Jaccard, 1901) and with simple matching Sokal-Michener coefficient (Sokal and Michener, 1958).
- Distances for multiclass data: Hamming (associated to simple matching coefficient).

Most of the above distances are well-known to data scientists and their formulas are considered here. The previous list is not exhaustive, and other distances may be more appropriate depending on the context.

Regarding binary data, two similarity coefficients are considered, Jaccard and Sokal-Michener. It is worth noting that the Jaccard coefficient excludes double-zeros from the similarity assessment, whereas the Sokal-Michener coefficient takes them into account. Thus, the use of Jaccard coefficient is recommended when double-zeros are not informative. Otherwise, the Sokal-Michener coefficient is preferred (see Gower and Legendre,

1986 and Legendre and De Cáceres, 2013 for details and discussion). In any case, the general transformation given in Gower (1966) is considered to obtain a distance from a similarity coefficient. That is, consider the sub-matrix $\mathbf{X}_2$ corresponding to the measurements of p_2 binary variables on the sample of n units. The (squared) distance between units i,r is obtained as

$$\delta^2(\mathbf{x}_{2,i},\mathbf{x}_{2,r}) = s(\mathbf{x}_{2,i},\mathbf{x}_{2,i}) + s(\mathbf{x}_{2,r},\mathbf{x}_{2,r}) - 2\,s(\mathbf{x}_{2,i},\mathbf{x}_{2,r}), \tag{1}$$

where $\mathbf{x}_{2,i}$, $\mathbf{x}_{2,r}$ are $p_2 \times 1$ vectors containing the binary measurements for units i,r, respectively, and $s(\mathbf{x}_{2,i},\mathbf{x}_{2,r})$ is a given similarity coefficient between them.

Regarding numerical data, combinations including Euclidean, Manhattan, Pearson or Canberra distances are not recommended in the presence of an underlying correlation structure or outlying observations. In such cases, robust Mahalanobis proposals are preferred. A robust Mahalanobis distance is obtained by using a robust estimator for the covariance matrix in Mahalanobis distance formula.

In what follows, we focus on a procedure to obtain such a robust estimation, which consists of three steps: estimation of variances, estimation of Pearson's correlation coefficients, and estimation of covariances (see Gnanadesikan, 1997 for the details).

Consider the sub-matrix $\mathbf{X}_1$ corresponding to the measurements of p_1 numerical variables on the sample of n units. The (squared) robust Mahalanobis distance between units i,r is defined as:

$$\delta^2_{Maha}(\mathbf{x}_{1,i},\mathbf{x}_{1,r}) = (\mathbf{x}_{1,i}-\mathbf{x}_{1,r})'\,\mathbf{S}^{*-1}\,(\mathbf{x}_{1,i}-\mathbf{x}_{1,r}) \tag{2}$$

where $\mathbf{x}_{1,i}$ and $\mathbf{x}_{1,r}$ are $p_1 \times 1$ vectors containing the measurements for units i,r, respectively, and $\mathbf{S}^* = (s^*_{jk})_{1\le j,k\le p_1}$ is a robust estimation of the sample covariance matrix of $\mathbf{X}_1$.

In this paper we consider three methods for computing the s^*_{jk}'s, namely median absolute deviation (MAD), trimmed and winsorized. In any case, the first step of the procedure consists in selecting one of these three methods to estimate the variances of the numerical variables (that is, the diagonal elements of $\mathbf{S}^*$):

(1) MAD: $\hat{\sigma}^{*2}(X_j) = MAD(X_j)^2 = [Me\,(|x_{ij} - Me(X_j)| : i = 1,...,n)]^2$, where $Me(\cdot)$ stands for the median.

(2) Trimmed: $\hat{\sigma}^{*2}(X_j) = \hat{\sigma}^2(X_j^\alpha)$, where X_j^α is an α-trimmed version of X_j, that is,

$$X_j^\alpha = \left\{x_{ij} : i \in \{1,\ldots,n\}, x_{ij} \in [Q(\alpha/2,X_j), Q(1-\alpha/2,X_j)]\right\},$$

where $Q(z,X_j)$ is the $z\times 100$ quantile of X_j, $z \in [0,1]$.

(3) Winsorized: $\hat{\sigma}^{*2}(X_j) = \hat{\sigma}^2(X_j^\alpha)$, where X_j^α is an α-winsorized version of X_j, that is, $X_j^\alpha = \left\{h(x) : x \in X_j\right.$, where function h is defined as

$$h(x) = \begin{cases} a(\alpha), & \text{if } x \in A(\alpha), \\ b(\alpha), & \text{if } x \in B(\alpha), \\ x, & \text{if } x \in X_j \text{ and } x \notin A(\alpha), x \notin B(\alpha), \end{cases}$$

where $a(\alpha)$ is the value of X_j that is immediately greater than $Q(\alpha/2, X_j)$, $b(\alpha)$ is the value of X_j that is immediately lower than $Q(1-\alpha/2, X_j)$ and $A(\alpha) = \{x_{ij} : x_{ij} \leq Q(\alpha/2, X_j)\}$, $B(\alpha) = \{x_{ij} : x_{ij} \geq Q(1-\alpha/2, X_j)\}$.

In the second step of the procedure, a robust estimator of Pearson's correlation coefficient between two numerical variables is given. For each pair of variables X_j and X_k, a robust estimation of their Pearson's correlation coefficient is computed as follows:

$$r^*_{jk} = \frac{\hat{\sigma}^{*2}_+ - \hat{\sigma}^{*2}_-}{\hat{\sigma}^{*2}_+ + \hat{\sigma}^{*2}_-},$$

where $\hat{\sigma}^{*2}_+$ and $\hat{\sigma}^{*2}_-$ are robust estimators of the variances of $Z_j + Z_k$ and $Z_j - Z_k$, respectively, with $Z_j = X_j / \sqrt{\hat{\sigma}^{*2}(X_j)}$ and $Z_k = X_k / \sqrt{\hat{\sigma}^{*2}(X_k)}$. Note that the same method to estimate the variances selected in the first step must be used for $\hat{\sigma}^{*2}_+$ and $\hat{\sigma}^{*2}_-$.

In the final step of the procedure, the off-diagonal elements in $\mathbf{S}^*$ are obtained. For each pair of variables X_j and X_k, a robust estimation of their covariance is obtained as:

$$s^*_{jk} = r^*_{jk} \sqrt{\hat{\sigma}^{*2}(X_j)\, \hat{\sigma}^{*2}(X_k)}.$$

In the simulation study, parameter α in trimmed and winsorized methods was set equal to the true proportion of outlying units. Additionally, a sensitivity study on the effect of this parameter on the classification rate is given for each dataset.

Note that formula (2) relies on the fact that $\mathbf{S}^*$ is positive definite. In case $\mathbf{S}^* \geq 0$, then the inverse in formula (2) is substituted by the corresponding Moore-Penrose pseudo-inverse. In case $\mathbf{S}^*$ is not positive semi-definite, that is, in case negative eigenvalues exist, a shrinkage scheme can be applied to its elements to ensure positive definiteness. Some proposals can be found in Devlin et al. (1975). For instance, the Devlin algorithm to obtain positive definite $\mathbf{S}^*$ is based on the following transformation:

$$\mathcal{G}(\mathbf{S}^*) = (g(s^*_{jk}))_{\{j,k=1,\ldots,p\}},$$

where

$$g(s^*_{jk}) = \begin{cases} 0, & \text{if } |s^*_{jk}| \leq z(\varepsilon), \\ z^{-1}(z(s^*_{jk}) + \varepsilon), & \text{if } s^*_{jk} < -z(\varepsilon), \\ z^{-1}(z(s^*_{jk}) - \varepsilon), & \text{if } s^*_{jk} > z(\varepsilon), \end{cases}$$

where ε is a small positive number, for example, $\varepsilon = 0.05$, $z(x) = \arctan h(x) = \frac{1}{2}\log\left(\frac{1+x}{1-x}\right)$, $z^{-1}(x) = \tan h(x)$ and $z(\varepsilon) = z(0.05) \approx 0.05$. The algorithm is applied recursively until $\mathcal{G}(\mathbf{S}^*)$ is positive definite.

In what follows we proceed to describe related metric scaling (RelMS), a multivariate technique introduced by (Cuadras and Fortiana, 1995, 1998), with the aim of combining several distance matrices computed on the same set of individuals in a single one. The method is based on the construction of a joint metric that satisfies several axioms related to the property of identifying and discarding redundant information (see

Albarrán et al., 2015; Grané and Romera, 2018). It is a very general method that allows to combine several sources of information, whenever a distance function can be measured between units. Although in Section 3 we explore it for the combination of numerical, binary and multiclass variables, it can also be applied to combine other kinds of data, such as functional data, time series, images, manifolds, compositional data, etc. Another possibility is to group variables according to different sources of information and combine the resulting distance matrices (Grané et al., 2022). Here we give the general description of the method.

Let $\mathbf{X}$ be an $n \times p$ data matrix corresponding to the measurements of p variables $X_1, \ldots, X_p$ on a sample of n units, and consider that the p variables can be grouped in m different types or sources of information.

1. Split matrix $\mathbf{X}$ into m sub-matrices $\mathbf{X}_k$ of size $n \times p_k$, $k = 1, \ldots, m$, regarding each variable type or source of information.

2. For each sub-matrix $\mathbf{X}_k$ consider a proper distance measure between units, according to the characteristics of the data, δ_k, and compute the corresponding matrix of squared pairwise distances conveniently standardized by its geometric variability (Cuadras and Fortiana, 1995), so that all matrices to be combined are commensurate, that is:

$$\mathbf{\Delta}_k = \frac{1}{V_{\mathbf{\Delta}_k}} \left(\delta_k^2(\mathbf{x}_{k,i}, \mathbf{x}_{k,r}) \right)_{\{1 \leq i,r \leq n\}}, \tag{3}$$

   where $\mathbf{x}_{k,i}, \mathbf{x}_{k,r}$ denote the i-th and r-th rows of matrix $\mathbf{X}_k$, respectively, and

$$V_{\mathbf{\Delta}_k} = \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{r=1}^{n} \delta_k^2(\mathbf{x}_{k,i}, \mathbf{x}_{k,r}).$$

3. For each matrix $\mathbf{\Delta}_k$ compute the corresponding Gram matrix:

$$\mathbf{G}_k = -\frac{1}{2} \mathbf{H} \mathbf{\Delta}_k \mathbf{H},$$

   where $\mathbf{H} = \mathbf{I} - \frac{1}{n} \mathbf{1}\mathbf{1}'$ is the centering matrix, $\mathbf{I}$ is the identity matrix of size $n \times n$ and $\mathbf{1}$ is a $n \times 1$ vector of ones.

4. Check for *Euclideanarity*[1]: Each $\mathbf{G}_k$ must be positive definite. If this is not the case, several transformations can be applied to $\mathbf{\Delta}_k$ so that this requirement is fulfilled. In this paper the additive transformation is applied, but other possibilities may serve for this purpose (see Borg and Groenen, 1986; Gower and Legendre, 2005). For simplicity, we keep the same notation for $\mathbf{G}_k$'s, assuming that they satisfy the Euclidean requirement.

[1] The word "Euclideanarity" was coined by John Gower in Gower and Legendre (1986).

5. Combine all Gram matrices to get the Gram matrix of the joint metric as follows:

$$\mathbf{G} = \sum_{k=1}^{m} \mathbf{G}_k - \frac{1}{m} \sum_{k \neq l} \mathbf{G}_k^{1/2} \mathbf{G}_l^{1/2}, \tag{4}$$

   where $\mathbf{G}_k^{1/2}$ is the square root of $\mathbf{G}_k$, which can be obtained through the singular value decomposition of $\mathbf{G}_k$.

6. The matrix of squared distances of the joint metric is obtained from $\mathbf{G}$ as follows:

$$\mathbf{\Delta} = \mathbf{g}\mathbf{1}' + \mathbf{1}\mathbf{g}' - 2\mathbf{G} = \left(\delta^2(\mathbf{x}_i, \mathbf{x}_r)\right)_{\{1 \leq i,r \leq n\}}, \tag{5}$$

   where $\mathbf{g} = \text{diag}(\mathbf{G})$ is a $n \times 1$ vector containing the diagonal elements of $\mathbf{G}$, and $\mathbf{x}_i, \mathbf{x}_r$ denote the i-th and r-th rows of the data matrix $\mathbf{X}$, respectively.

7. Finally, the distance matrix of the joint metric is $\mathbf{D} = \left(\delta(\mathbf{x}_i, \mathbf{x}_r)\right)_{\{1 \leq i,r \leq n\}}$, that contains the square root of the elements of $\mathbf{\Delta}$.

The first addend of formula (4) mimics classical Gower distance by adding the three metrics, although here the addition is done through the matrices of square distances. The second addend is responsible of discarding redundant information coming from different sources. Note that RelMS can be computationally expensive for large sample sizes (see Appendix B). This is the reason why a simplified version of the above procedure was proposed, called generalized Gower distance (G-Gower).

Inspired by Gower's works, G-Gower (square) distance is defined as the linear combination of the matrices of squared pairwise distances, conveniently standardized by their corresponding geometric variability. That is,

$$\mathbf{\Delta}_{GG} = \sum_{k=1}^{m} \mathbf{\Delta}_k, \tag{6}$$

where each $\mathbf{\Delta}_k$ is defined as in (3) and fulfills the Euclidean requirement. Equivalently, (square) G-Gower can be obtained from the first addend of formula (4).

3. Empirical evaluation

In this section the performance of the distances presented in Section 2 is evaluated in the context of clustering and compared to those of classical Gower and Euclidean distance. The aim of the simulation study is to analyze their performance in the presence of underlying correlation structure and outlier contamination. The simulation involves four synthetic and two real datasets and k-medoids algorithm is used to obtain the partitioning.

In all cases, the true class is known for each unit. Thus, classification rate (proportion of units out of the total that are correctly classified) and adjusted Rand index

(ARI) (Hubert and Arabie, 1985; Rand, 1971) are used to evaluate the goodness of the clustering.

The Rand index was proposed by Rand (1971) as a clustering validation measure. However, as noted by Hubert and Arabie (1985) and Nguyen and Bailey (2009), in practice the Rand index frequently takes values in the $[0.5, 1]$ interval, its reference value (baseline value) can be high and not take a constant value. For these reasons the Rand index is most used in its adjusted version, known as the adjusted Rand index. Considering that there are n units and two partitions of them $\mathcal{C}_1 = \{C_{11}, \ldots, C_{1r}\}$, $\mathcal{C}_2 = \{C_{21}, \ldots, C_{2s}\}$ with r and s clusters, respectively, the adjusted Rand index is defined as

$$ARI = \frac{2\,(ab - cd)}{(a+c)(b+c) + (a+d)(b+d)},$$

where a is the number of pairs of units belonging to the same cluster in both partitions $\mathcal{C}_1$ and $\mathcal{C}_2$. That is, $i, j \in C_{1h}$ and $i, j \in C_{2u}$, for some $h = 1, \ldots, r$ and $u = 1, \ldots, s$; b is the number of pairs of units belonging to different clusters in partitions $\mathcal{C}_1$ and $\mathcal{C}_2$. That is, $i \in C_{1h_1}$ and $j \in C_{1h_2}$ for $h_1 \neq h_2$, $h_1, h_2 = 1, \ldots, r$, and also $i \in C_{2u_1}$ and $j \in C_{2u_2}$, for $u_1 \neq u_2$, $u_1, u_2 = 1, \ldots, s$; c is the number of pairs of units belonging to the same cluster in partition $\mathcal{C}_1$ but to different clusters in partition $\mathcal{C}_2$. That is, $i, j \in C_{1h}$ for $h = 1, \ldots, r$, but $i \in C_{2u_1}$ and $j \in C_{2u_2}$, for $u_1 \neq u_2$, $u_1, u_2 = 1, \ldots, s$; d is the number of pairs of units belonging to different clusters in partition $\mathcal{C}_1$ but to the same cluster in partition $\mathcal{C}_2$. That is, $i \in C_{1h_1}$ and $j \in C_{1h_2}$ for $h_1 \neq h_2$, $h_1, h_2 = 1, \ldots, r$, but $i, j \in C_{2u}$, for $u = 1, \ldots, s$.

The adjusted Rand index takes values in $[-0.5, 1]$ and the closer to one, the more similar the compared rankings are. In contrast, the closer to -0.5, the more different the compared rankings. In practice, one of the two cluster configurations (or two classifications) that the Rand index requires (adjusted or not), will be the one defined by a variable that is taken as a grouping response, and the other will be the one defined by a classification algorithm, such as k-medoids. Therefore, in practice it is necessary to have information about a categorical response variable to be able to implement the Rand index as a validation measure for clustering algorithms. In this scenario, the interpretation of the ARI is the following: the closer it is to 1, the more similar the classification made by the algorithm is to the real classification, and the closer it is to -0.5 , the less similar. In fact, in this context, an ARI close to zero indicates that the classification performed by the algorithm is similar to the one that would be obtained with a purely random classification procedure, and when it is negative it indicates that it is even worse.

A total of 34 distances are under study. In the case of G-Gower or RelMS, they are obtained as a combination of three distances, one for each type of data, following formulas (6) or (5), respectively. The distances considered are:

1. Generalized Gower composed by

    - Euclidean (ℓ^2 distance), Manhattan (ℓ^1 distance), Canberra, Pearson (standardized Euclidean), Mahalanobis, robust Mahalanobis (MAD, trimmed, winsozired) for numerical variables,

- Jaccard, Sokal-Michener similarity coefficients for binary variables, transformed to distances according to formula (1),
- Hamming for multiclass variables.

2. Related metric scaling composed by
   - Euclidean (ℓ^2 distance), Manhattan (ℓ^1 distance), Canberra, Pearson (standardized Euclidean), Mahalanobis, robust Mahalanobis (MAD, trimmed, winsozired) for numerical variables,
   - Jaccard, Sokal-Michener similarity coefficients for binary variables, transformed to distances according to formula (1),
   - Hamming for multiclass variables.

Additionally, Euclidean (ℓ^2 distance) and classical Gower distance are considered for comparison of results. Note that applying the Euclidean distance on raw mixed-type data is not recommended at all. The reason why we keep it is because it usually appears as the default distance in many software packages and we want to emphasize the consequences of using such a distance in a wrong context.

Tables 1–4 contain the classification rate and ARI mean values, computed on 100 runs for each scenario and distance considered. Additionally, Figures 1–4 contain metric MDS configurations corresponding to one of the 100 runs that help to illustrate the differences between the true and assigned class of the units. In the simulation study, results concerning the Euclidean distance are shown to illustrate the odd performance when using such a distance in mixed-type data.

3.1. Simulation study

Synthetic datasets of mixed-type data were generated with the `make_blobs` function from the `scikit-learn` Python library. Each dataset is composed by $p_1 = 4$ numerical, $p_2 = 2$ binary, $p_3 = 2$ multiclass variables measured on n units divided in k true classes. Different outlier patterns were added to each dataset.

1. Sample size: $n = 500$,
   Variables: Only 2 of them are informative and 6 with redundant information; Underlying correlation structure: Three pairs of numerical variables are highly correlated; Outliers: 10-12% contamination in three numerical variables.
   True classes: $k = 3$ (balanced).

2. Sample size: $n = 650$,
   Variables: 3 of them are informative and 5 with redundant information; Underlying correlation structure: Two pairs of numerical variables are highly correlated; Outliers: 10% contamination in two numerical variables,
   True classes: $k = 4$ (balanced).

Table 1. *Classification results for synthetic dataset 1.*

Distance	**Classification rate**	**ARI**
RelMS: Mahalanobis-Jaccard-Hamming	0.630	0.273666
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.628	0.278432
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.628	0.277728
G-Gower: Mahalanobis-Jaccard-Hamming	0.628	0.268592
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.628	0.278432
RelMS: Canberra-Jaccard-Hamming	0.622	0.238612
G-Gower: Canberra-Jaccard-Hamming	0.620	0.230389
G-Gower: Canberra-Sokal-Hamming	0.620	0.218362
RelMS: Canberra-Sokal-Hamming	0.612	0.212984
G-Gower: Mahalanobis-Sokal-Hamming	0.608	0.222972
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.598	0.255065
RelMS: Mahalanobis-Sokal-Hamming	0.590	0.195316
RelMS: Pearson-Jaccard-Hamming	0.588	0.190902
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.586	0.188601
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.584	0.185916
classical Gower	0.562	0.192146
G-Gower: Robust Mahalanobis MAD-Jaccard-Hamming	0.554	0.190833
G-Gower: Pearson-Jaccard-Hamming	0.552	0.190902
G-Gower: Pearson-Sokal-Hamming	0.540	0.220737
G-Gower: Robust Mahalanobis MAD-Sokal-Hamming	0.540	0.220737
RelMS: Pearson-Sokal-Hamming	0.538	0.202318
G-Gower: Euclidean-Sokal-Hamming	0.538	0.219598
RelMS: Robust Mahalanobis MAD-Sokal-Hamming	0.538	0.202318
G-Gower: Manhattan-Sokal-Hamming	0.534	0.216939
RelMS: Euclidean-Jaccard-Hamming	0.532	0.175767
G-Gower: Euclidean-Jaccard-Hamming	0.532	0.175767
RelMS: Robust Mahalanobis MAD-Jaccard-Hamming	0.530	0.170310
RelMS: Manhattan-Jaccard-Hamming	0.528	0.168527
G-Gower: Manhattan-Jaccard-Hamming	0.528	0.168527
RelMS: Manhattan-Sokal-Hamming	0.520	0.176607
RelMS: Euclidean-Sokal-Hamming	0.516	0.164035
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.508	0.159580
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.506	0.157313
Euclidean	0.350	0.000090

3. Sample size: $n = 600$,
   Variables: 6 of them are informative and 2 with redundant information; Underlying correlation structure: Three pairs of numerical variables are highly correlated;

Outliers: 7-8% contamination in three numerical variables.
True classes: $k = 4$ (unbalanced).

4. Sample size: $n = 600$,
   Variables: All informative; Uncorrelated and uncontaminated;
   True classes: $k = 4$ (balanced).

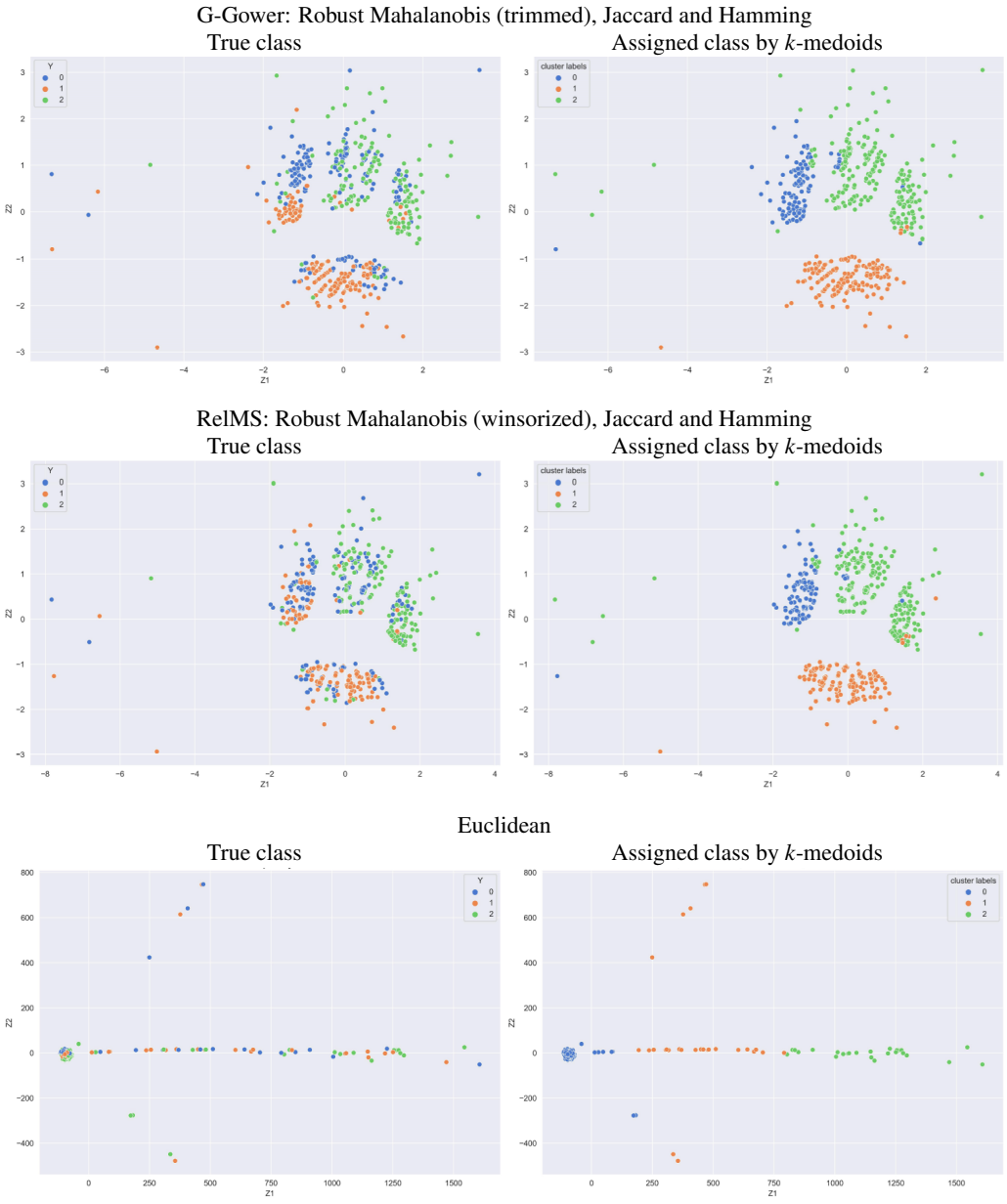

Figure 1. *Clustering visualization. Synthetic dataset 1.*

Table 2. *Classification results for synthetic dataset 2.*

Distance	**Classification rate**	**ARI**
G-Gower: Robust Mahalanobis MAD-Jaccard-Hamming	0.621538	0.329944
G-Gower: Mahalanobis-Sokal-Hamming	0.592308	0.324297
RelMS: Robust Mahalanobis MAD-Jaccard-Hamming	0.590769	0.260597
RelMS: Robust Mahalanobis MAD-Sokal-Hamming	0.567692	0.297498
RelMS: Mahalanobis-Sokal-Hamming	0.567692	0.228790
G-Gower: Canberra-Sokal-Hamming	0.561538	0.221440
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.558462	0.216499
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.558462	0.216499
RelMS: Pearson-Sokal-Hamming	0.552308	0.202442
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.552308	0.214666
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.552308	0.214415
G-Gower: Pearson-Sokal-Hamming	0.550769	0.215299
RelMS: Euclidean-Sokal-Hamming	0.547692	0.244515
G-Gower: Robust Mahalanobis MAD-Sokal-Hamming	0.546154	0.205740
G-Gower: Mahalanobis-Jaccard-Hamming	0.543077	0.226489
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.540000	0.222812
G-Gower: Manhattan-Sokal-Hamming	0.540000	0.232898
G-Gower: Euclidean-Sokal-Hamming	0.540000	0.232898
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.540000	0.222812
RelMS: Pearson-Jaccard-Hamming	0.533846	0.199402
RelMS: Mahalanobis-Jaccard-Hamming	0.526154	0.195419
G-Gower: Pearson-Jaccard-Hamming	0.524615	0.198994
G-Gower: Canberra-Jaccard-Hamming	0.524615	0.194698
RelMS: Euclidean-Jaccard-Hamming	0.503077	0.203928
G-Gower: Manhattan-Jaccard-Hamming	0.500000	0.193327
G-Gower: Euclidean-Jaccard-Hamming	0.500000	0.193327
RelMS: Canberra-Sokal-Hamming	0.496923	0.223695
RelMS: Manhattan-Jaccard-Hamming	0.495385	0.189415
RelMS: Manhattan-Sokal-Hamming	0.492308	0.191534
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.481538	0.155862
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.478462	0.153906
RelMS: Canberra-Jaccard-Hamming	0.460000	0.159691
classical Gower	0.423077	0.125406
Euclidean	0.263077	0.000138

Table 1 contains the classification rate and ARI values for k-medoids algorithm with $k = 3$, concerning synthetic dataset 1. We observe that G-Gower with robust Mahalanobis (trimmed), Jaccard and Hamming reaches a classification rate of 62.3% and an

ARI value of 0.27. Similar values are reached by RelMS with robust Mahalanobis (winsorized), Jaccard and Hamming. These classification rates (and ARI values) are higher than those obtained by classical Gower or Euclidean distance, for which values of 56.2% (ARI 0.19) and 35.0% (ARI $< 10^{-4}$) are attained, respectively.

In Figure 1 metric MDS maps are used to illustrate the k-medoids classification ($k = 3$), for synthetic dataset 1, using G-Gower with robust Mahalanobis (trimmed), Jaccard and Hamming, RelMS with robust Mahalanobis (winsorized), Jaccard and Hamming and Euclidean distance. Units in left panels are colored according to their true class and in right panels, according to their assigned class. The clustering in G-Gower and RelMS panels can be considered rather acceptable, since around half of the units in class 0 are not well identified. On the other hand, the clustering with Euclidean distance is disappointing, where the configuration appears completely distorted due to outlying observations and the underlying correlation structure that this distance is not able to incorporate.

Table 2 contains the classification rate and ARI values for k-medoids algorithm with $k = 4$, regarding synthetic dataset 2. In this case, G-Gower with robust Mahalanobis (MAD), Jaccard and Hamming reaches a classification rate of 62.2% and an ARI value of 0.33, and a 59.1% rate and 0.26 ARI are attained by RelMS with robust Mahalanobis (MAD), Jaccard and Hamming. These classification rates (and ARI values) are higher than those obtained by classical Gower or Euclidean distance, whose values are located at the end of the ranking.

In Figure 2 metric MDS maps are used to illustrate the k-medoids classification ($k = 4$), for synthetic dataset 2, using G-Gower with robust Mahalanobis (MAD), Jaccard and Hamming, RelMS with robust Mahalanobis (MAD), Jaccard and Hamming and classical Gower. Units in left panels are colored according to their true class and in right panels, according to their assigned class. The clustering in G-Gower panels can be considered rather acceptable, since most of the units in classes 0 and 1 are well identified. On the other hand, the clustering with classical Gower is not good since this distance is not able to incorporate the underlying correlation structure as well as the presence of outlying units. Once more, the clustering with Euclidean distance is disappointing.

Table 3 contains the classification rate and ARI values for k-medoids algorithm with $k = 4$, concerning synthetic dataset 3. We observe that G-Gower with robust Mahalanobis (MAD), Sokal and Hamming reaches a classification rate of 88.3% and an ARI value of 0.73. On the other hand, classification rates for Euclidean and classical Gower are of 44.17% (ARI 0.29) and 73.50% (ARI 0.26), respectively.

In Figure 3 metric MDS maps are used to illustrate the k-medoids classification ($k = 4$), for synthetic dataset 3, using G-Gower with robust Mahalanobis MAD, Sokal and Hamming and classical Gower. Units in left panels are colored according to their true class and in right panels, according to their assigned class. In G-Gower panels we can observe that the four clusters are very similar to the true ones. However, this is not the case for classical Gower, where most of the units in class 0 and half of the units in class 3 are not well identified.

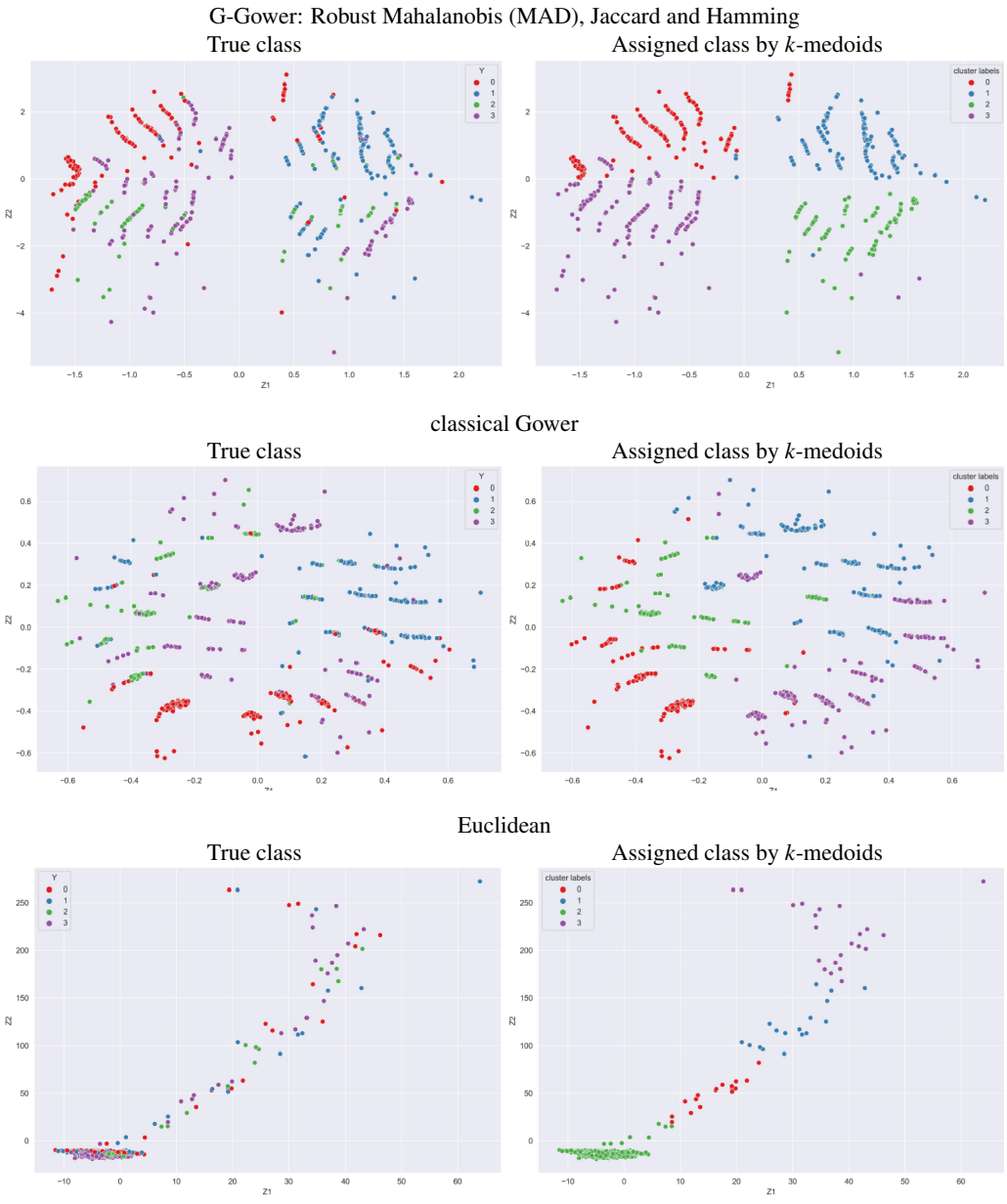

Figure 2. *Clustering visualization. Synthetic dataset 2.*

Table 3. *Classification results for synthetic dataset 3.*

Distance	**Classification rate**	**ARI**
G-Gower: Robust Mahalanobis MAD-Sokal-Hamming	0.883333	0.726101
G-Gower: Pearson-Sokal-Hamming	0.881667	0.723303
G-Gower: Canberra-Jaccard-Hamming	0.880000	0.704662
G-Gower: Canberra-Sokal-Hamming	0.880000	0.703744
RelMS: Canberra-Sokal-Hamming	0.866667	0.683853
RelMS: Canberra-Jaccard-Hamming	0.850000	0.657553
RelMS: Robust Mahalanobis MAD-Sokal-Hamming	0.731667	0.613787
RelMS: Pearson-Sokal-Hamming	0.730000	0.611196
RelMS: Euclidean-Sokal-Hamming	0.728333	0.615708
RelMS: Manhattan-Sokal-Hamming	0.728333	0.615395
G-Gower: Manhattan-Sokal-Hamming	0.723333	0.597885
G-Gower: Euclidean-Sokal-Hamming	0.710000	0.564313
G-Gower: Manhattan-Jaccard-Hamming	0.708333	0.559781
G-Gower: Euclidean-Jaccard-Hamming	0.708333	0.558163
RelMS: Manhattan-Jaccard-Hamming	0.705000	0.556431
RelMS: Euclidean-Jaccard-Hamming	0.700000	0.541697
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.690000	0.587217
G-Gower: Robust Mahalanobis MAD-Jaccard-Hamming	0.686667	0.589241
RelMS: Robust Mahalanobis MAD-Jaccard-Hamming	0.686667	0.589241
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.686667	0.587224
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.686667	0.587224
RelMS: Mahalanobis-Jaccard-Hamming	0.686667	0.587224
G-Gower: Mahalanobis-Jaccard-Hamming	0.685000	0.576739
G-Gower: Pearson-Jaccard-Hamming	0.685000	0.582427
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.683333	0.578341
RelMS: Pearson-Jaccard-Hamming	0.671667	0.547123
classical Gower	0.651667	0.292456
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.605000	0.490818
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.605000	0.491105
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.591667	0.486762
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.586667	0.484619
RelMS: Mahalanobis-Sokal-Hamming	0.583333	0.484672
G-Gower: Mahalanobis-Sokal-Hamming	0.581667	0.486274
Euclidean	0.533333	0.255522

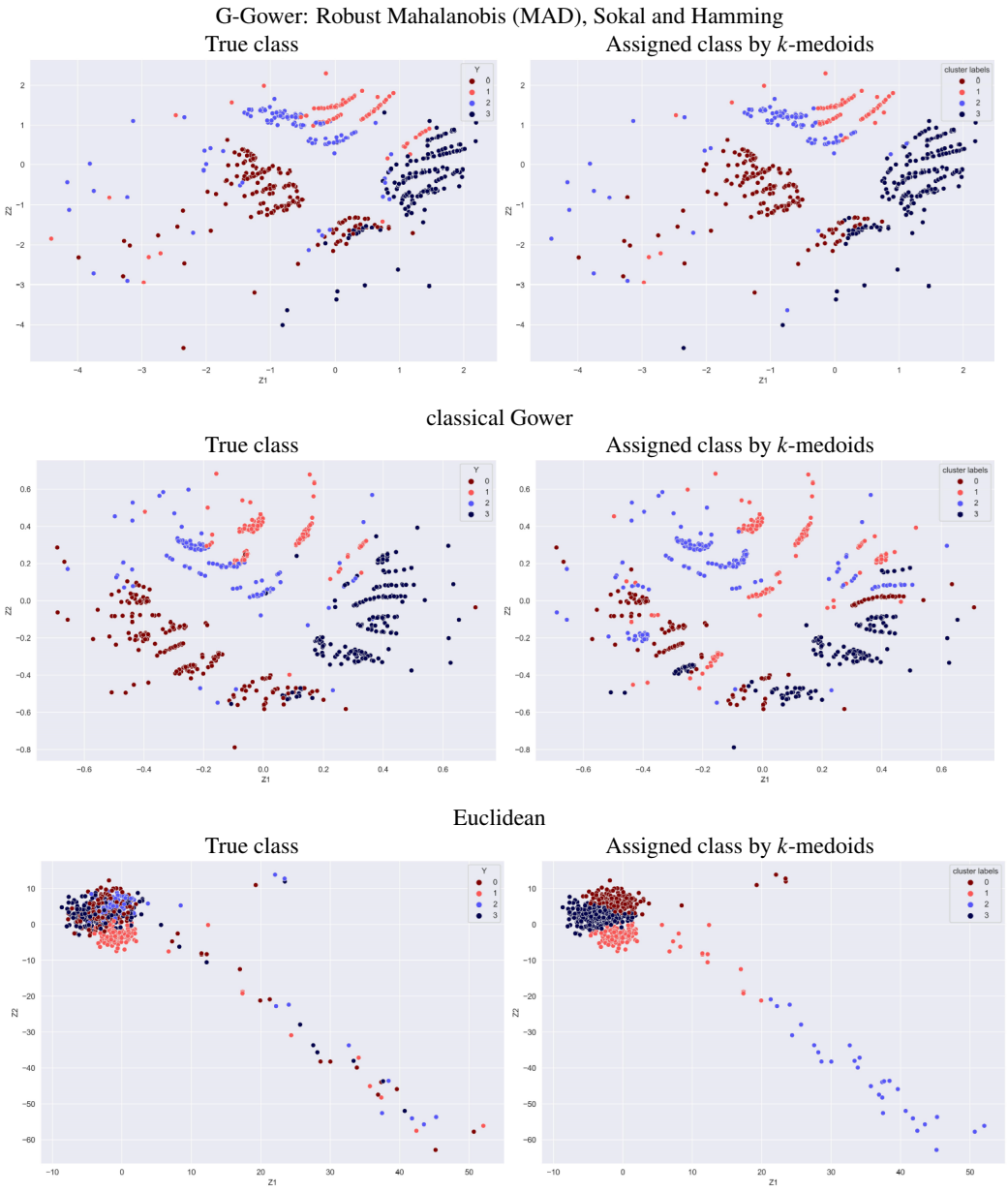

Figure 3. *Clustering visualization. Synthetic dataset 3.*

Table 4 contains the classification rate and ARI values for k-medoids algorithm with $k = 4$, concerning synthetic dataset 4, where variables are uncorrelated and uncontaminated. As expected, classification rates are rather low and most ARI values are close to zero, since data in the four groups lack underlying correlation structure (all variables are uncorrelated and either normally or uniformly distributed). In this case, the best performance is achieved by Euclidean distance, with a classification rate of 55.33% and

an ARI value of 0.1852, followed by G-Gower with Canberra distance for numerical variables. The performance of these methods is illustrated in Figure 4 with metric MDS maps.

Table 4. *Classification results for synthetic dataset 4.*

Distance	**Classification rate**	**ARI**
Euclidean	0.5533	0.1852
G-Gower: Canberra-Sokal-Hamming	0.4900	0.1652
RelMS: Canberra-Jaccard-Hamming	0.4750	0.1278
classical Gower	0.4683	0.1415
RelMS: Pearson-Jaccard-Hamming	0.4300	0.0905
RelMS: Euclidean-Sokal-Hamming	0.4300	0.1140
RelMS: Euclidean-Jaccard-Hamming	0.4283	0.0896
G-Gower: Manhattan-Jaccard-Hamming	0.4283	0.0918
G-Gower: Pearson-Jaccard-Hamming	0.4283	0.0898
G-Gower: Euclidean-Jaccard-Hamming	0.4267	0.0901
RelMS: Canberra-Sokal-Hamming	0.4267	0.1213
RelMS: Pearson-Sokal-Hamming	0.4217	0.1077
RelMS: Robust Mahalanobis mad-Sokal-Hamming	0.4200	0.1039
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.4200	0.0841
RelMS: Manhattan-Jaccard-Hamming	0.4200	0.0826
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.4200	0.0841
RelMS: Mahalanobis-Jaccard-Hamming	0.4200	0.0837
G-Gower: Mahalanobis-Jaccard-Hamming	0.4200	0.0845
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.4183	0.0844
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.4183	0.0845
RelMS: Mahalanobis-Sokal-Hamming	0.4150	0.1022
RelMS: Robust Mahalanobis mad-Jaccard-Hamming	0.4150	0.0808
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.4150	0.1022
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.4150	0.1022
G-Gower: Robust Mahalanobis mad-Jaccard-Hamming	0.4133	0.0816
G-Gower: Euclidean-Sokal-Hamming	0.3967	0.0841
G-Gower: Pearson-Sokal-Hamming	0.3917	0.0807
RelMS: Manhattan-Sokal-Hamming	0.3883	0.0928
G-Gower: Canberra-Jaccard-Hamming	0.3867	0.0528
G-Gower: Manhattan-Sokal-Hamming	0.3817	0.0874
G-Gower: Mahalanobis-Sokal-Hamming	0.3800	0.0882
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.3800	0.0906
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.3783	0.0887
G-Gower: Robust Mahalanobis mad-Sokal-Hamming	0.3783	0.0921

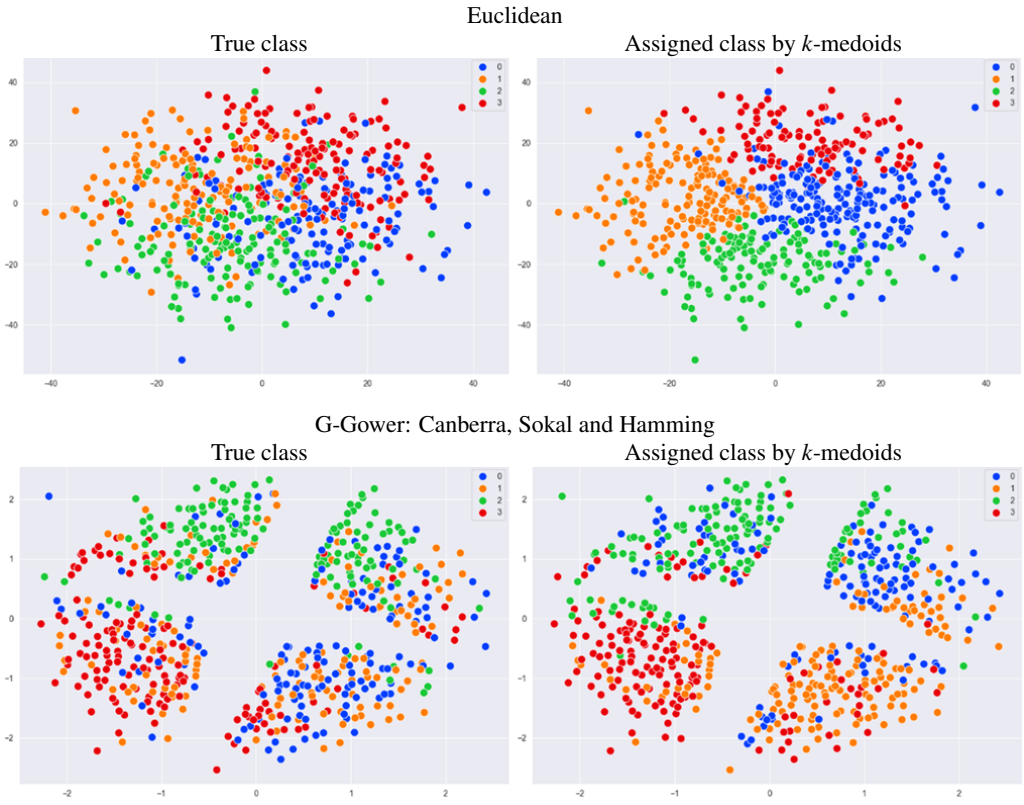

Figure 4. *Clustering visualization. Synthetic dataset 4.*

3.2. Real datasets

3.2.1. Dubai properties dataset

This dataset contains 1905 properties for which 38 characteristics were measured. It is available at `https://www.kaggle.com/datasets/dataregress/dubai-properties-dataset?resource=download`.

The variables considered as predictors are: Latitude and longitude, market price and size (in m^2) as numerical; number of bedrooms (0,1,2,3,4,5) and number of bathrooms (0,1,2,3,4,5,6) as multiclass, and balcony (1=true, 0=false), barbecue are (1=true, 0=false) and private pool (1=true, 0=false) are taken as binary. We decided to consider house quality (1=Low, 0=Medium/High/UltraHigh) as response variable with $k = 2$ classes.

Table 5 contains the classification rate and ARI values for k-medoids algorithm with $k = 2$, where we observe classification rates higher than 85% for some of the robust proposals. In particular, G-Gower with robust Mahalanobis (trimmed, winsorized and MAD), Jaccard and Hamming attain a classification rate of 86.14%, RelMS with robust Mahalanobis (MAD), Jaccard and Hamming reaches the 86.09%, RelMS with robust Mahalanobis (trimmed and winsorized), Jaccard and Hamming reach 85.98% and for

robust Mahalanobis (trimmed, winsorized and MAD), Sokal and Hamming a 85.83% is attained. On the other hand, the classification rate for Euclidean and Gower are 60.58% and 50.97%, respectively.

Table 5. *Classification results for Dubai properties dataset.*

Distance	**Classification rate**	**ARI**
G-Gower: Mahalanobis-Jaccard-Hamming	0.861942	0.505161
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.861417	0.503672
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.861417	0.503672
G-Gower: Robust Mahalanobis MAD-Jaccard-Hamming	0.861417	0.503672
RelMS: Mahalanobis-Jaccard-Hamming	0.860892	0.502544
G-Gower: Canberra-Jaccard-Hamming	0.860892	0.502365
RelMS: Canberra-Jaccard-Hamming	0.860892	0.502365
RelMS: Robust Mahalanobis MAD-Jaccard-Hamming	0.860892	0.502365
RelMS: Pearson-Jaccard-Hamming	0.860892	0.502365
G-Gower: Pearson-Jaccard-Hamming	0.860367	0.500702
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.859843	0.499400
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.859318	0.497922
RelMS: Mahalanobis-Sokal-Hamming	0.858268	0.495329
G-Gower: Robust Mahalanobis MAD-Sokal-Hamming	0.858268	0.495329
RelMS: Pearson-Sokal-Hamming	0.858268	0.495329
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.858268	0.495329
RelMS: Canberra-Sokal-Hamming	0.858268	0.495329
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.858268	0.495329
RelMS: Robust Mahalanobis MAD-Sokal -Hamming	0.858268	0.495329
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming b	0.858268	0.495329
G-Gower: Mahalanobis-Sokal-Hamming	0.858268	0.495329
G-Gower: Pearson-Sokal-Hamming	0.858268	0.495329
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.858268	0.495329
RelMS: Euclidean-Jaccard-Hamming	0.819948	0.394762
RelMS: Manhattan-Jaccard-Hamming	0.819948	0.394762
G-Gower: Manhattan-Jaccard-Hamming	0.817848	0.389618
G-Gower: Euclidean-Jaccard-Hamming	0.817848	0.389618
Euclidean	0.605774	0.000912
G-Gower: Canberra-Sokal-Hamming	0.516010	-0.002348
RelMS: Manhattan-Sokal-Hamming	0.510761	-0.000050
RelMS: Euclidean-Sokal-Hamming	0.510761	-0.000050
classical Gower	0.509711	-0.000083
G-Gower: Euclidean-Sokal-Hamming	0.506562	-0.000263
G-Gower: Manhattan-Sokal-Hamming	0.506562	-0.000263

Similar conclusions can be derived regarding ARI. For example, G-Gower with robust Mahalanobis (trimmed, winsorized and MAD), Jaccard and Hamming reach one of the highest ARIs, as well as RelMS with robust Mahalanobis (MAD), Jaccard and Hamming. On the other hand, values around 0 are obtained by classical Gower and Euclidean distance.

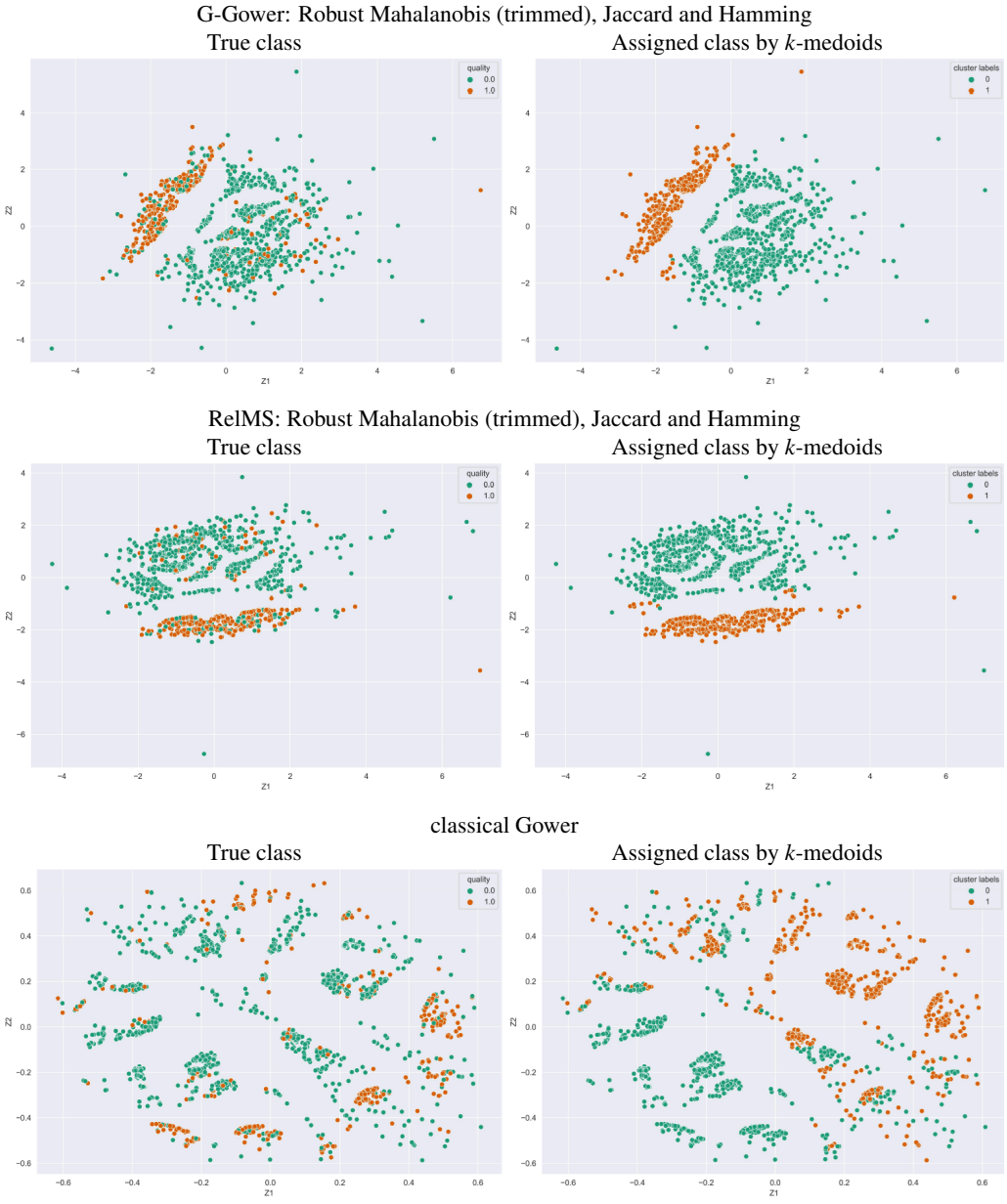

Figure 5. *Clustering visualization. Dubai properties dataset.*

In Figure 5 we use metric MDS maps to visualize the classification found by k-medoids for different metrics ($k = 2$). In particular, we illustrate the results obtained by G-Gower with robust Mahalanobis trimmed, Jaccard and Hamming, RelMS with the same combination of metrics and classical Gower. Units in left panels are colored according to their true class and in right panels, according to their assigned class.

Looking at G-Gower and RelMS panels, we can see that k-medoids is able to identify the class of most of the units, which is coherent with the classification rates obtained, higher than 80%. Focusing on Gower panels, we can see that many units are not well identified, and this is reflected in a classification rate of 51%. Considering that there are only two classes, this means that the classification is practically the same as what would be obtained if the units were classified randomly, following a uniform probability distribution. The ARI value is 0, which is consistent with the low classification rate obtained.

3.2.2. World development indicators dataset

This dataset contains 18 indicators measured 97 countries. It is available at `https://www.kaggle.com/datasets/hn4ever/world-development-indicators-by-countries`. Source: World Bank.

We consider the following predictors. Numerical variables: Access to electricity (% of the population with access to electricity in 2017), life expectancy (number of years a newborn in 2019 would live if the mortality patterns existing at the time of his/her birth remained the same throughout his/her life),insufficient nutrition (% of the population in 2019 whose habitual food consumption is insufficient to provide the levels of dietary energy necessary to maintain a normal active and healthy life), arable land (% of the country's land that is arable in 2014-16), population with less than 3.20$ per day (% of population with less than 3.20$ per day in PPP); Binary variables: Quality of the health system (0=Suitable 1=Inappropriate), Investment in education and health (0=High, 1=Low); multiclass variables: Pollution (0=High, 1=Medium, 2=Low), Insecurity (0=High, 1=Medium, 2=Low). Variable Poverty (0=High, 1=Medium, 2=Low) is taken as response variable with $k = 3$ classes.

Table 6 contains the classification rate and ARI values for k-medoids algorithm with $k = 3$, where we observe classification rates around 65% for some of the robust proposals. In particular, G-Gower with robust Mahalanobis (trimmed and winsorized), Sokal or Jaccard and Hamming attains a 64.95%, RelMS with robust Mahalanobis (trimmed and winsorized), Jaccard and Hamming reaches a 63.92%, as well as G-Gower with robust Mahalanobis (MAD), Sokal or Jaccard and Hamming. On the other hand, the classification rate with Euclidean distance is 46.39% and 60.82% with classical Gower.

Similar results can be observed concerning ARI, where the robust proposals tend to attain ARI values around 0.35-0.37. On the other hand, classical Gower and Euclidean distance obtain values of 0.26 and 0.04, respectively.

Table 6. *Classification results for World development indicators dataset.*

Distance	**Classification rate**	**ARI**
G-Gower: Canberra-Jaccard-Hamming	0.659794	0.405188
G-Gower: Manhattan-Jaccard-Hamming	0.649485	0.315937
RelMS: Canberra-Jaccard-Hamming	0.649485	0.397914
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.649485	0.354318
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.649485	0.375463
G-Gower: Mahalanobis-Sokal-Hamming	0.649485	0.354318
G-Gower: Pearson-Jaccard-Hamming	0.649485	0.315937
G-Gower: Canberra-Sokal-Hamming	0.649485	0.382536
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.649485	0.334477
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.649485	0.315937
RelMS: Mahalanobis-Jaccard-Hamming	0.649485	0.334477
G-Gower: Euclidean-Jaccard-Hamming	0.639175	0.353248
RelMS: Pearson-Jaccard-Hamming	0.639175	0.313709
G-Gower: Pearson-Sokal-Hamming	0.639175	0.375701
RelMS: Euclidean-Jaccard-Hamming	0.639175	0.353248
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.639175	0.334477
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.639175	0.334477
G-Gower: Robust Mahalanobis MAD-Jaccard-Hamming	0.639175	0.307562
G-Gower: Robust Mahalanobis MAD-Sokal-Hamming	0.639175	0.326646
RelMS: Mahalanobis-Sokal-Hamming	0.628866	0.347343
RelMS: Robust Mahalanobis trimmed-Sokal- Hamming	0.628866	0.325988
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.628866	0.325988
RelMS: Robust Mahalanobis MAD-Jaccard-Hamming	0.628866	0.288271
G-Gower: Mahalanobis-Jaccard-Hamming	0.628866	0.319983
RelMS: Robust Mahalanobis MAD-Sokal-Hamming	0.618557	0.305616
RelMS: Canberra-Sokal-Hamming	0.608247	0.255497
classical Gower	0.608247	0.258923
G-Gower: Euclidean-Sokal-Hamming	0.577320	0.208742
G-Gower: Manhattan-Sokal-Hamming	0.577320	0.208742
RelMS: Manhattan-Sokal-Hamming	0.556701	0.170696
RelMS: Pearson-Sokal-Hamming	0.556701	0.170696
RelMS: Euclidean-Sokal-Hamming	0.556701	0.170696
RelMS: Manhattan-Jaccard-Hamming	0.525773	0.096234
Euclidean	0.463918	0.040735

In Figure 6 we illustrate the results of the k-medoids classification for different metrics ($k = 3$) through metric MDS maps. In particular, the results obtained by RelMS with robust Mahalanobis trimmed, Jaccard and Hamming, and Euclidean distance are shown.

Units in left panels are colored according to their true class and in right panels, according to their assigned class. Looking at RelMS panels, we can see that the classification is rather acceptable, since most of the units in class 0 and half of the units in class 1 are well classified. Note that the classification rate obtained is 64.9%, which is around twice a expected classification rate of 33% that would be attained if the units were classified through a uniform random mechanism. Regarding the Euclidean distance panels, we observe that *k*-medoids algorithm is not able to identify the units' class, which is coherent with an ARI of 0.04 and a classification rate of 46%.

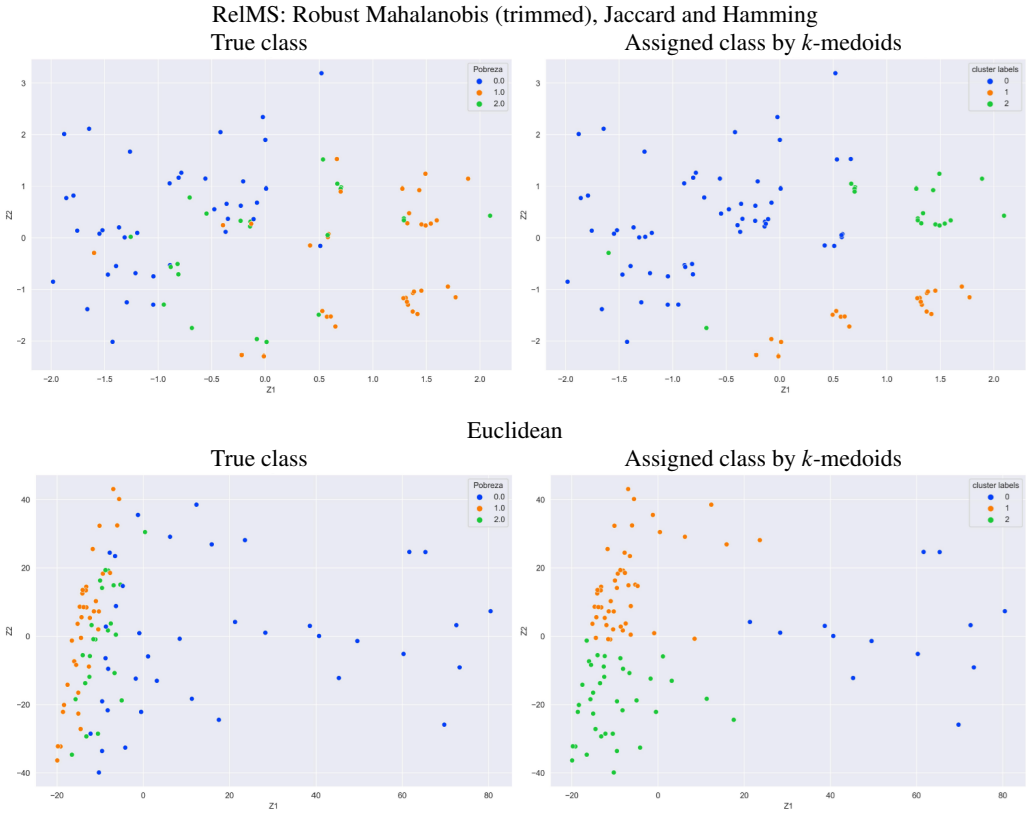

Figure 6. *Clustering visualization. World development indicators dataset.*

3.3. Sensitivity study on the trimming and winsorizing parameter

In this section the sensitivity of parameter α used in trimmed and winsorized versions of the covariance matrix in Mahalanobis distance is studied. Classification rate is used to analyze the performance of the distance for each dataset. Table 7 contains the mean values computed on 100 runs.

Concerning synthetic datasets, we observe that in dataset 1 (10-12% outlier contamination in three numerical variables) G-Gower with robust Mahalanobis (trimmed),

Table 7. *Classification rates for G-Gower and RelMS for several values of trimming and winsoring parameter α.*

Distance	**5%**	**10%**	**15%**	**20%**	**25%**
Synthetic dataset 1					
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.630	0.628	0.554	0.558	0.556
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.630	0.628	0.604	0.548	0.544
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.626	0.598	0.538	0.546	0.546
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.604	0.628	0.628	0.556	0.554
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.586	0.584	0.580	0.596	0.594
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.582	0.586	0.580	0.592	0.592
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.508	0.506	0.508	0.602	0.604
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.506	0.508	0.504	0.602	0.604
Synthetic dataset 2					
RelMS Robust Mahalanobis trimmed-Jaccard-Hamming	0.491	0.500	0.500	0.502	0.505
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.558	0.558	0.558	0.554	0.555
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.558	0.558	0.558	0.483	0.554
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.552	0.552	0.552	0.548	0.549
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.552	0.552	0.552	0.522	0.548
G-Gower -Robust Mahalanobis trimmed-Jaccard-Hamming	0.540	0.540	0.540	0.522	0.525
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.540	0.540	0.540	0.449	0.522
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.480	0.482	0.480	0.554	0.560
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.477	0.478	0.480	0.442	0.554
Synthetic dataset 3					
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.690	0.687	0.822	0.840	0.840
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.690	0.683	0.792	0.817	0.817
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.688	0.833	0.837	0.843	0.685
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.687	0.815	0.812	0.685	0.687
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.593	0.868	0.878	0.877	0.878
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.592	0.853	0.885	0.883	0.885
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.588	0.590	0.865	0.885	0.883
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.585	0.600	0.857	0.878	0.875
Dubai properties dataset					
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.861	0.861	0.861	0.862	0.861
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.861	0.861	0.861	0.861	0.861
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.860	0.859	0.860	0.861	0.862
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.859	0.860	0.860	0.860	0.860
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.858	0.858	0.858	0.858	0.858
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.858	0.858	0.858	0.858	0.858
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.858	0.858	0.858	0.858	0.858
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.858	0.858	0.858	0.858	0.858
World development indicators dataset					
G-Gower: Robust Mahalanobis trimmed-Jaccard-Hamming	0.649	0.649	0.649	0.649	0.649
G-Gower: Robust Mahalanobis winsorized-Jaccard-Hamming	0.649	0.649	0.649	0.649	0.649
RelMS: Robust Mahalanobis trimmed-Jaccard-Hamming	0.649	0.649	0.639	0.639	0.639
RelMS: Robust Mahalanobis winsorized-Jaccard-Hamming	0.649	0.649	0.639	0.639	0.649
G-Gower: Robust Mahalanobis trimmed-Sokal-Hamming	0.649	0.649	0.649	0.639	0.639
G-Gower: Robust Mahalanobis winsorized-Sokal-Hamming	0.649	0.649	0.649	0.649	0.649
RelMS: Robust Mahalanobis trimmed-Sokal-Hamming	0.629	0.629	0.629	0.629	0.629
RelMS: Robust Mahalanobis winsorized-Sokal-Hamming	0.629	0.629	0.629	0.629	0.629

Jaccard and Hamming and RelMS with robust Mahalanobis (winsorized), Jaccard and Hamming present the highest classification rate (63.0%), which is attained for $\alpha = 0.05$. The second best rate (62.8%) is attained by the same metrics for $\alpha = 0.10$. In dataset 2 (10% outlier contamination in two numerical variables), the best result (55.8%) is obtained for RelMS with robust Mahalanobis (trimmed and winsorized), Sokal and Hamming for $\alpha = 0.05, 0.10, 0.15$. In dataset 3 (7-8% outlier contamination in three numerical variables), G-Gower with robust Mahalanobis (trimmed and winsorized), Sokal and Hamming reach the best results (88.3%-88.5%) for $\alpha = 0.20, 0.25$.

No general conclusions can be derived from the analysis of the synthetic datasets. Although for datasets 1 and 2 the best results are obtained for α values close to the real proportion of outlying units, this is not the case for dataset 3 where hard trimming/winsorizing is needed. Some explanations may be found in the complexity degree of dataset 3, with unbalanced classes and less redundant information than in datasets 1 and 2.

To better analyze the sensitivity of parameter α classification rates are depicted in Figure 7, where we observe that for dataset 1 the classification rate tends to decrease

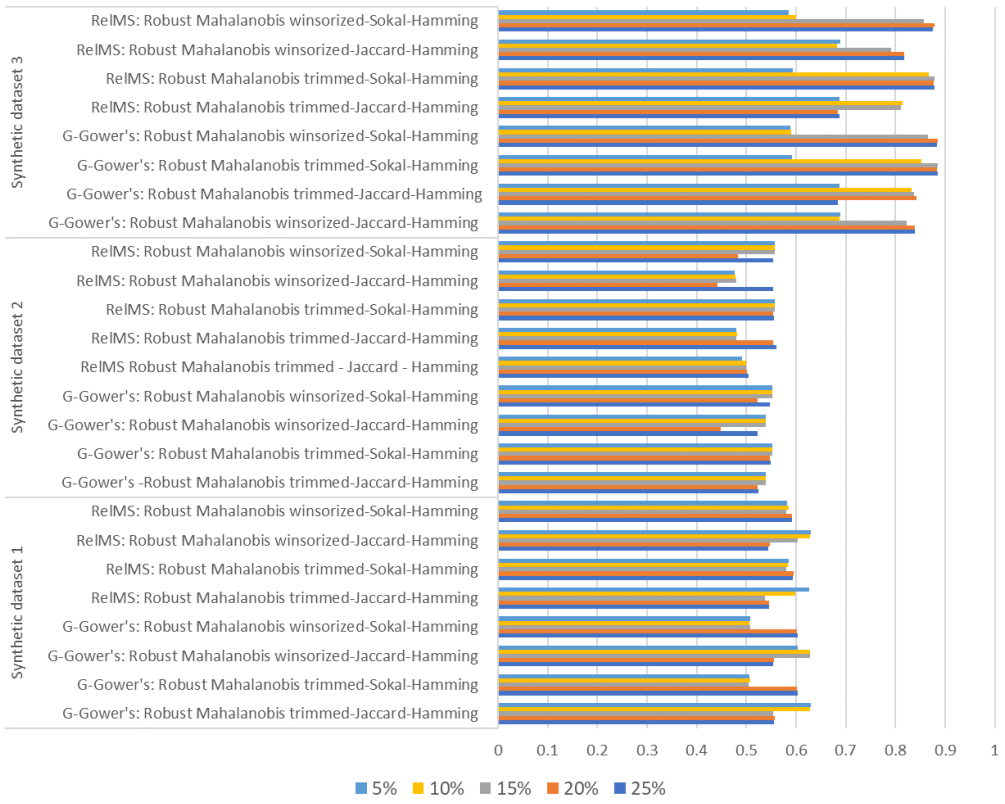

Figure 7. *Classification rates for G-Gower and RelMS for several values of trimming and winsoring parameter α in synthetic datasets.*

with α when Jaccard distance is considered (either in G-Gower or in RelMS), while the opposite happens for Sokal-Michener's. There is not a clear pattern for dataset 2, although the classification rate in G-Gower seems to decrease with α, whereas it stands still or slightly increases when using RelMS. Finally, for dataset 3, the classification rate tends to increase with α in most of the cases considered.

Regarding real datasets, the fluctuation in the classification rate for different values of α is lower than 10^{-2}. In general, for Dubai properties data, the best situation (86.1%) is produced with G-Gower metric with robust Mahalanobis (trimmed and winsorized), Jaccard and Hamming, at any value of α. In the case of World development indicators data, the best results (64.9%) are attained for G-Gower metric with robust Mahalanobis (trimmed and winsorized), Jaccard and Hamming, as well as G-Gower's with robust Mahalanobis (winsorized), Sokal and Hamming at any value of α. The same classification rates are reached for several trimming/winsorizing values of the remaining metrics, except for RelMS with robust Mahalanobis (trimmed and winsorized), Sokal and Hamming, whose classification rate is always equal to 62.9%.

4. Conclusions

In this work, new robust distances for mixed-type data were proposed and studied in the context of clustering. They were obtained as combination of three distances, one for each type of data (numerical, binary and multiclass). As a result, a total of 34 distances were analyzed.

Their performance was evaluated in rather complex synthetic datasets, with underlying correlation structure and outlying units, as well as in two real datasets. Classification rate and adjusted Rand index were used to evaluate the goodness of the clustering obtained with the k-medoids algorithm. Metric multidimensional scaling was used to visualize and illustrate the difference between the true and assigned class of the units. In all scenarios with underlying correlation structure and outlying units, new robust proposals outperformed the classical Gower distance. In the absence of correlation or outlying units, Euclidean distance achieved the best results. However, this is not the usual context in real-world applications, where outliers are highly probable and redundant information is often present in multivariate data. In addition, a sensitivity analysis on the parameters involved in the robust estimation of the new proposals was provided. Some of the robust proposals became computationally unfeasible for sample sizes larger than 30000, with an i7-1365U 1.80 GHz processor, 32.0 GB RAM, where no parallelization was used. The study of their feasibility in larger sample sizes is left for further research.

5. Software availabily

All the distances presented in this paper are implemented in a Python package, called robust_mixed_dist, hosted in `https://pypi.org/project/robust_mixed_dist/`. The package is described in Appendix A and a tutorial is available at `https://fabios`

cielzoortiz.github.io/robust_mixed_dist-docu/intro.html. The robust_mixed_dist package relies on Scipy for an efficient computation of distances. Thus, all distance functions available therein can be easily included in our package. However, the handling of missing data is not considered in Scipy and we leave it for further research.

Funding

The authors acknowledge the support of grants PID2021-123592OB-I00 and TED2021-129316B-I00, funded by MICIU/AEI/10.13039/501100011033, "ERDF A way of making Europe" and "European Union NextGenerationEU/PRTR".

Acknowledgments

The authors thank anonymous reviewers and AE for their comments, leading to an improved version of the manuscript. Special thanks to Prof. Pierre Legendre for his constructive and detailed comments, suggestions, and references.

References

Albarrán, I., Alonso, P. and Grané, A. (2015). Profile identification via weighted related metric scaling: an application to dependent spanish children. *Journal of the Royal Statistical Society - Series A. Statistics in Society*, 178:1–26.

Boj, E. and Grané, A. (2024). The robustification of distance-based linear models: some proposals. *Socio-Economic Planning Sciences*, 95:11992.

Borg, I. and Groenen, P. (2005). *Modern multidimensional scaling: theory and applications*. Springer, New York.

Cuadras, C.M. and Fortiana, J. (1995). A continuous metric scaling solution for a random variable. *Journal of Multivariate Analysis*, 52:1–14.

Cuadras, C.M. and Fortiana, J. (1998). Visualizing categorical data with related metric scaling. In: *Visualization of Categorical Data*. Ed. by J. Blasius and M. Greenacre. London: Academic Press, pages 365–376.

Devlin, S.J., Gnanadesikan, R. and Kettenring, J.R. (1975). Robust estimation and outlier detection with correlation coefficients. *Biometrika*, 62(3):531–545.

Gnanadesikan, R. (1997). *Methods for Statistical Data Analysis of Multivariate Observations*. John Wiley and Sons, London.

Gower, J.C. (1966). Some distance properties of latent root and vector methods used in multivariate analysisilarity coefficients. *Biometrika*, 53:325–338.

Gower, J.C. (1971). A general coefficient of similarity and some of its properties. *Biometrika*, 27:857–874.

Gower, J.C. and Legendre, P. (1986). Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3:5–48.

Grané, A., Manzi, G. and Salini, S. (2022). Dynamic mixed data analysis and visualization. *Entropy*, 24:1399.

Grané, A. and Romera, R. (2018). On visualizing mixed-type data: A joint metric approach to profile construction and outlier detection. *Sociological Methods and Research*, 47(2):207–39.

Grané, A., Salini, S. and Verdolini, E. (2021). Robust multivariate analysis for mixed-type data: Novel algorithm and its practical application in socio-economic research. *Socio-Economic Planning Sciences*, 73:100907.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Société vaudoise des sciences naturelles*, 37:547–579.

Kaufman, L. and Rousseeuw, P. (1990). *Finding groups in data: an introduction to cluster analysis*. Wiley, Boca Raton.

Legendre, P. and De Cáceres, M. (2013). Beta diversity as the variance of community data: dissimilarity coefficients and partitioning. *Ecology Letters*, 16:951–963.

Nguyen X.V., Epps, J. and Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, 1073–108.

Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 53:846–850.

Sokal, R.R. and Michener, C.D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438.

A. The Python package

The robust_mixed_dist package is a new Python tool for computing classical statistical distances between units. The distance functions implemented are: Euclidean (ℓ^2 distance), Minkowski (family of ℓ^p distances), Canberra, Pearson (standardized Euclidean), Mahalanobis, robust Mahalanobis, Gower, generalized Gower (G-Gower) and related metric scaling (RelMS). A total of 41 statistical distances can be calculated, including those proposed in this paper. Additionally, since robust_mixed_dist relies on Scipy Python library, all distances included there can be easily included. Forn example, this is the case for Chi-square distance, Chebyschev distance, Jensen-Shannon distance, among others, as well as many similarity coefficients available at Scipy.

The package is hosted in PyPI (Python Packages Index), the official repository of Python packages. More information about robust_mixed_dist can be found in `https://pypi.org/project/robust_mixed_dist/`.

In what follows we provide a small demonstration of how to use robust_mixed_dist. For more details, a more extensive tutorial is available at `https://fabioscielzoortiz.github.io/robust_mixed_dist-docu/intro.html`.

Example of use

Installing:

```
pip install robust-mixed-dist
```

Loading modules:

```
from robust_mixed_dist.mixed import GGowerDistMatrix
from robust_mixed_dist.mixed import RelMSDistMatrix
from robust_mixed_dist.quantitative import robust_maha_dist_matrix, S_robust
```

Computing some distances for a real Madrid houses dataset, which can be found at https://www.kaggle.com/datasets/mirbektoktogaraev/madrid-real-estate-market. In this brief tutorial only the following variables of that dataset were considered:

- Quantitative: sq mt built, number of rooms, number of bathrooms, number of floors, buy price.
- Binary: is renewal needed, has lift, is exterior, has parking.
- multiclass: energy certificate, house type.

Robust Mahalanobis (MAD):

```
S_robust_ = S_robust(X=madrid_houses_df, method='MAD',
                     epsilon=0.05, n_iters=20)
robust_maha_dist_matrix(madrid_houses_df, S_robust=S_robust_)
```

```
array([[ 0.        ,  6.47092419,  7.01983235, ...,  4.96377088,
         5.69177645,  3.68021705],
       [ 6.47092419,  0.        ,  3.03471006, ..., 10.43356417,
        10.12781147,  5.95613137],
       [ 7.01983235,  3.03471006,  0.        , ..., 11.35024985,
        10.9171085 ,  6.21243845],
       ...,
       [ 4.96377088, 10.43356417, 11.35024985, ...,  0.        ,
         3.65216542,  7.11373136],
       [ 5.69177645, 10.12781147, 10.9171085 , ...,  3.65216542,
         0.        ,  7.86440327],
       [ 3.68021705,  5.95613137,  6.21243845, ...,  7.11373136,
         7.86440327,  0.        ]])
```

G-Gower: Robust Mahalanobis (trimmed), jaccard, Hamming (matching):

```
GG_init = GGowerDistMatrix(p1=5, p2=4, p3=2,
           d1='robust_mahalanobis', d2='jaccard', d3='hamming',
           method='trimmed', alpha=0.05, epsilon=0.05, n_iters=20,
           fast_VG=False)
D_GG = GG_init.compute(X=madrid_houses_df)
D_GG
```

```
array([[0.       , 2.21885363, 1.93318704, ..., 1.93891555, 3.12986955,
        2.26834878],
       [2.21885363, 0.       , 1.22309875, ..., 2.38689136, 2.63841547,
        2.01262089],
       [1.93318704, 1.22309875, 0.       , ..., 2.3585878 , 2.50589448,
        1.63422771],
       ...,
       [1.93891555, 2.38689136, 2.3585878 , ..., 0.       , 2.89514966,
        1.7665964 ],
       [3.12986955, 2.63841547, 2.50589448, ..., 2.89514966, 0.       ,
        3.02408907],
       [2.26834878, 2.01262089, 1.63422771, ..., 1.7665964 , 3.02408907,
        0.       ]])
```

RelMS: Robust Mahalanobis (winsorized), Jaccard, Hamming (matching):

```
RelMS_init = RelMSDistMatrix(p1=5, p2=4, p3=2,
                d1='robust_mahalanobis', d2='jaccard', d3='hamming',
                method='winsorized', epsilon=0.05, alpha=0.05,
                n_iters=20)
D_RelMS = RelMS_init.compute(X=madrid_houses_df.head(1000),
                             Gs_PSD_trans=True)
D_RelMS
```

```
array([[ 0.       , 10.29131982, 10.20148541, ..., 10.25180831,
        10.1963865 , 10.23458336],
       [10.29131994,  0.       , 10.13419898, ..., 10.10288394,
        10.08333674, 10.0731428 ],
       [10.20148539, 10.134199  ,  0.       , ..., 10.14619431,
        10.03394396, 10.11537897],
       ...,
       [10.25180831, 10.10288394, 10.14619431, ...,  0.       ,
        10.05982432, 10.00909222],
       [10.1963865 , 10.08333674, 10.03394396, ..., 10.05982432,
         0.       , 10.03008924],
       [10.23458336, 10.0731428 , 10.11537897, ..., 10.00909221,
        10.03008923,  0.       ]])
```

B. Computational cost

In this section a computational experiment is derived to illustrate the computational cost of G-Gower and RelMS metrics.

For such purpose eight synthetic datasets were generated using `make_blobs` function from `scikit-learn` Python library. The datasets have increasing sample size, going from $n = 3{,}000$ to $n = 70{,}000$, and same characteristics like $p_1 = 4$ numerical, $p_2 = 2$ binary and $p_3 = 2$ multiclass variables and number of true classes $k = 3$.

For each dataset the computation time for G-Gower (robust Mahalanobis trimmed-Jaccard-Hamming), RelMS (robust Mahalanobis trimmed-Jaccard-Hamming) and robust Mahalanobis (trimmed) distances was collected. The experiment was carried out with a DESKTOP-I1Q2RCC device, 13th Gen Intel(R) Core(TM) i7-1365U 1.80 GHz processor, with Installed RAM of 32.0 GB (31.6 GB usable) and 64-bit operating system, x64-based processor. Results are shown in Figure 8.

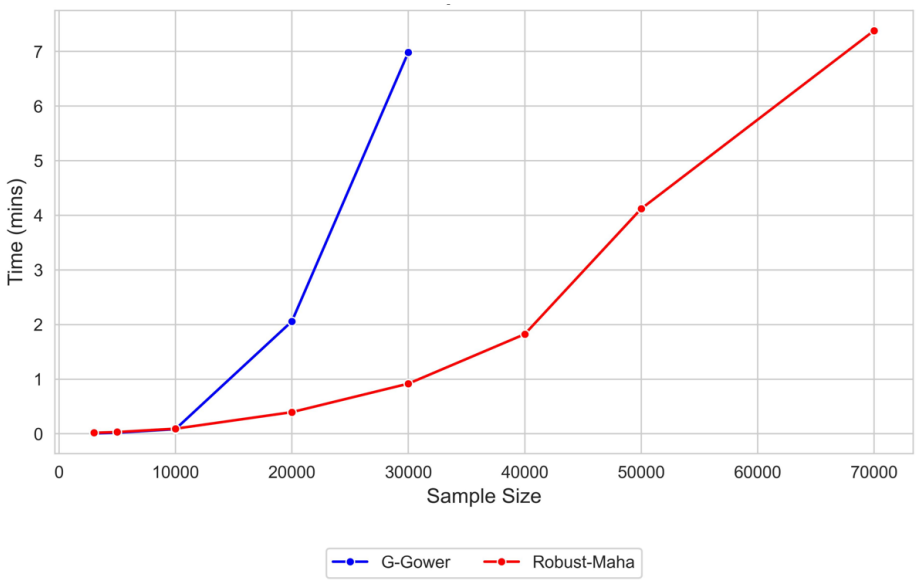

Figure 8. *Computational time (in minutes) for G-Gower and robust Mahalanobis.*

The computationally cheapest distance is robust Mahalanobis, since it takes less than one minute for $n = 30,000$ and below, between one and four minutes for sizes between $n = 30,000$ and $n = 50,000$, and for the largest size ($n = 70,000$) it takes seven minutes, approx. Its calculation is feasible for all sample sizes tested, and reasonably practical for all of them as well. G-Gower takes few seconds for sample sizes up to $n = 10,000$, no more than two minutes for sample sizes between $n = 10,000$ and $n = 20,000$, between 2 and 7 minutes for sizes between $n = 20,000$ and $n = 30,000$, and becomes unfeasible for larger sample sizes. Computational time for RelMS goes from 2.37 minutes for $n = 3,000$ to 534.07 minutes for $n = 20,000$, this is the reason why they are not included in Figure 8. Sample sizes $n \geq 30,000$ are unfeasible for RelMS.