Hierarchical Clustering via Penalty-Based Aggregation and the Genie Approach

Marek Gagolewski*[1,2], Anna Cena[1], and Maciej Bartoszuk[2]

[1] Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland

Abstract. The paper discusses a generalization of the nearest centroid hierarchical clustering algorithm. A first extension deals with the incorporation of generic distance-based penalty minimizers instead of the classical aggregation by means of centroids. Due to that the presented algorithm can be applied in spaces equipped with an arbitrary dissimilarity measure (images, DNA sequences, etc.). Secondly, a correction preventing the formation of clusters of too highly unbalanced sizes is applied: just like in the recently introduced *Genie* approach, which extends the single linkage scheme, the new method averts a chosen inequity measure (e.g., the Gini-, de Vergottini-, or Bonferroni-index) of cluster sizes from raising above a predefined threshold. Numerous benchmarks indicate that the introduction of such a correction increases the quality of the resulting clusterings significantly.

Keywords: hierarchical clustering, aggregation, centroid, Gini-index, Genie algorithm

1 Introduction

A data analysis technique called clustering or data segmentation (see, e.g., [10]) aims at grouping – in an unsupervised manner – a family of objects into a number of subsets in such a way that items within each cluster are more similar to each other than to members of different clusters. The focus of this paper is on hierarchical clustering procedures, i.e., on algorithms which do not require the number of output clusters to be fixed a priori. Instead, each method of this sort results in a sequence of nested partitions that can be cut at an arbitrary level.

Recently, we proposed a new algorithm, named *Genie* [8]. Its reference implementation has been included in the `genie` package for R [15] see `http://cran.r-project.org/web/packages/genie/`. In short, the method is based on the single linkage criterion: in each iteration, the pair of closest data points from two different clusters is looked up in order to determine which subsets are to be merged. However, if an economic inequity measure (e.g., the Gini-index)

* Corresponding author, email: `gagolews@ibspan.waw.pl`, phone: +48 22 3810 378.

of current cluster sizes raises above a given threshold, a forced merge of low-cardinality clusters occurs so as to prevent creating a few very large clusters and many small ones. Such an approach has many advantages:

– By definition, the *Genie* clustering is more resistant to outliers.
– A study conducted on 29 benchmark sets revealed that the new approach reflects the underlying data structure better than not only when the average, single, complete, and Ward linkages are used, but also when the k-means and BIRCH algorithms are applied.
– It relies on arbitrary dissimilarity measures and thus may be used to cluster not only points in the Euclidean space, but also images, DNA or protein sequences, informetric data, etc., see [6].
– Just like the single linkage, it may be computed based on the minimal spanning tree. A modified, parallelizable Prim-like algorithm (see [14]) can be used so as to guarantee that a chosen dissimilarity measure is computed exactly once for each unique pair of data points. In such a case, its memory use is linear and thus the algorithm can be used to cluster much larger data sets than with the Ward, complete, or average linkage.

The current contribution is concerned with a generalization of the centroid linkage scheme, which merges two clusters based on the proximity of their centroids. We apply, analyze, and test the performance of the two following extensions:

– First of all, we note that – similarly as in the case of the generalized fuzzy (weighted) k-means algorithm [5] – the linkage can take into account arbitrary distance-based penalty minimizers which are related to idempotent aggregation functions on spaces equipped with a dissimilarity measure.
– Secondly, we incorporate the *Genie* correction for cluster sizes in order to increase the quality of the resulting data subdivision schemes.

The paper is set out as follows. The new linkage criterion is introduced in Sec. 2. In Section 3 we test the quality of the resulting clusterings on benchmark data of different kinds. A possible algorithm to employ the discussed linkage criterion is given in Sec. 4. The paper is concluded in Sec. 5.

2 New Linkage Criterion

For some set $\mathcal{X}$, let $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\} \subseteq \mathcal{X}$ be an input data sequence and $\mathfrak{d}$ be a pairwise dissimilarity measure (distance, see [6]), i.e., a function $\mathfrak{d} : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ such that (a) $\mathfrak{d}$ is symmetric, i.e., $\mathfrak{d}(\mathbf{x}, \mathbf{y}) = \mathfrak{d}(\mathbf{y}, \mathbf{x})$ and (b) $(\mathbf{x} = \mathbf{y}) \implies \mathfrak{d}(\mathbf{x}, \mathbf{y}) = 0$ for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Example 1. Numerous practically useful examples of spaces like $(\mathcal{X}, \mathfrak{d})$ can be found very easily. The clustered data sets may consist of points in $\mathbb{R}^d$, character strings (DNA and protein sequences in particular), rankings, graphs, equivalence relations, intervals, fuzzy numbers, citation sequences, images, time series, and so on. For each such $\mathcal{X}$, many popular distances can be utilized, e.g., respectively, the Euclidean, Levenshtein, Kendall, etc. ones, see, e.g., [5,6]. $\boxdot$

Each hierarchical clustering procedure works in the following way. At the j-th step, $j = 0, \dots, n-1$, there are $n-j$ clusters. It is always true that $\mathcal{C}^{(j)} = \{C_1^{(j)}, \dots, C_{n-j}^{(j)}\}$ is a partition of the input data set. Formally, $C_u^{(j)} \cap C_v^{(j)} = \emptyset$ for $u \neq v$, $C_u^{(j)} \neq \emptyset$, and $\bigcup_{u=1}^{n-j} C_u^{(j)} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$. Initially, the first partition consists solely of singletons, i.e., we have that $C_i^{(0)} = \{\mathbf{x}^{(i)}\}$ for $i = 1, \dots, n$. When proceeding from step $j-1$ to j, a predefined linkage scheme determines which of the two clusters $C_u^{(j-1)}$ and $C_v^{(j-1)}$, $u < v$, are to be merged so as to we get $C_i^{(j)} = C_i^{(j-1)}$ for $u \neq i < v$, $C_u^{(j)} = C_u^{(j-1)} \cup C_v^{(j-1)}$, and $C_i^{(j)} = C_{i+1}^{(j-1)}$ for $i > v$. For instance, the single (minimum) linkage scheme assumes that u and v are such that:

$$\operatorname*{arg\,min}_{(u,v),u<v} \left(\min_{\mathbf{a} \in C_u^{(j-1)}, \mathbf{b} \in C_v^{(j-1)}} \mathfrak{d}(\mathbf{a}, \mathbf{b}) \right),$$

the complete (maximum) linkage is based on:

$$\operatorname*{arg\,min}_{(u,v),u<v} \left(\max_{\mathbf{a} \in C_u^{(j-1)}, \mathbf{b} \in C_v^{(j-1)}} \mathfrak{d}(\mathbf{a}, \mathbf{b}) \right),$$

and the average linkage on:

$$\operatorname*{arg\,min}_{(u,v),u<v} \left(\frac{1}{|C_u^{(j-1)}||C_v^{(j-1)}|} \sum_{\mathbf{a} \in C_u^{(j-1)}, \mathbf{b} \in C_v^{(j-1)}} \mathfrak{d}(\mathbf{a}, \mathbf{b}) \right),$$

see, e.g., [10] for a discussion.

Moreover, assuming that $\mathcal{X} = \mathbb{R}^d$ for some $d \geq 1$ and that $\mathfrak{d}$ is the Euclidean metric, we may consider the *centroid linkage criterion*:

$$\operatorname*{arg\,min}_{(u,v),u<v} \mathfrak{d}\left(\boldsymbol{\mu}(C_u^{(j-1)}), \boldsymbol{\mu}(C_v^{(j-1)}) \right),$$

where $\boldsymbol{\mu}(C)$, $C = \{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}$, denotes the *centroid* of C given by:

$$\boldsymbol{\mu}(\{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}) = \left(\frac{1}{m} \sum_{j=1}^{m} x_1^{(i_j)}, \dots, \frac{1}{m} \sum_{j=1}^{m} x_d^{(i_j)} \right),$$

that is, the componentwise arithmetic mean of points in C. It can easily be shown that in such a setting we have that:

$$\boldsymbol{\mu}(\{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}) = \operatorname*{arg\,min}_{\mathbf{y} \in \mathbb{R}^d} \sqrt{\frac{1}{m} \sum_{j=1}^{m} \mathfrak{d}^2(\mathbf{x}^{(i_j)}, \mathbf{y})}.$$

Just as in the case of the fuzzy k-means algorithm [5], again for an arbitrary $\mathcal{X}$ and $\mathfrak{d}$, we may generalize the above cluster aggregation method as follows:

$$\boldsymbol{\mu}_\varphi(\{\mathbf{x}^{(i_1)},\dots,\mathbf{x}^{(i_m)}\}) = \operatorname*{arg\,min}_{\mathbf{y}\in\mathcal{X}'} \varphi^{-1}\left(\frac{1}{m}\sum_{j=1}^{m}\varphi\left(\mathfrak{d}(\mathbf{x}^{(i_j)},\mathbf{y})\right)\right), \qquad (1)$$

where $\mathcal{X}'\subseteq\mathcal{X}$ and $\varphi:[0,\infty]\to[0,\infty]$ is a strictly increasing continuous function such that $\varphi(0)=0$. In other words, $\boldsymbol{\mu}_\varphi$ determines a minimizer of a distance-based penalty function given via a quasi-arithmetic mean. Let us observe that it is an idempotent fusion function, see [3,7]. Assuming that the solution to (1) exists and is unique, the incorporation of $\boldsymbol{\mu}_\varphi$ leads us to a generalized centroid linkage scheme that can work in arbitrary spaces equipped with a dissimilarity measure.

Remark 1. Most commonly, φ is set to be a power fuction, i.e., $\varphi(\delta)=\delta^p$ for some $p\ge 1$. For $\mathcal{X}'=\mathcal{X}$, the power-mean-based penalty minimizer corresponding to $p=1$ is usually called the *1-median*, for $p=2$ – *centroid*, and $p=\infty$ – *1-center*. However, special attention should be paid to whether a chosen fusion function can be computed sufficiently easily. In particular, if $\mathcal{X}=\mathbb{R}^d$, $p=2$, and $\mathfrak{d}$ is the Euclidean distance, then we noted that the solution is the componentwise arithmetic mean. Moreover, if $p=1$ and $\mathfrak{d}$ is the Manhattan distance, then we shall compute the componentwise median. On the other hand, for $p=1$ or $p=\infty$ and $\mathfrak{d}$ being the Euclidean distance, there is no open-form solution (but, e.g., the Weiszfeld algorithm or a quadratic programming task can be applied, see, e.g., [7]). However, e.g., the search for 1-median with respect to the Levenshtein distance on the space of non-trivial character strings yields an NP-complete problem. ⊡

Remark 2. We can also set $\mathcal{X}'=\{\mathbf{x}^{(i_1)},\dots,\mathbf{x}^{(i_m)}\}$ which leads to the concept of a *set exemplar*. In particular, if $\varphi(d)=d$, then the corresponding distance-based penalty minimizer is named *medoid*. The computation of such a fusion function is always relatively easy ($O(m^2)$-time is needed). Yet, we should note that if the set of penalty minimizers is non-unique, some tie breaking rule (e.g., point index-based one) should be additionally introduced. ⊡

In order to increase the clustering quality in the presence of potential outliers (at least if the true underlying cluster structure is not heavily unbalanced), we can also incorporate a correction used in the single-linkage-based Genie [8] algorithm. In order to do so, firstly, let us recall the notion of an inequity index, see [2,4,9].

Definition 1. *For a fixed $n\in\mathbb{N}$, let $\mathcal{G}$ denote the set of all non-increasingly ordered n-tuples with elements in the set of non-negative integers, i.e., $\mathcal{G}=\{(x_1,\dots,x_n)\in\mathbb{N}_0^n : x_1\ge\dots\ge x_n\}$. Then $\mathsf{F}:\mathcal{G}\to[0,1]$ is an inequity index, whenever:*

(a) it is Schur-convex, i.e., for any $\mathbf{x},\mathbf{y}\in\mathcal{G}$ with $\sum_{i=1}^n x_i=\sum_{i=1}^n y_i$, if it holds for all $i=1,\dots,n$ that $\sum_{j=1}^i x_j\le\sum_{j=1}^i y_j$, then $\mathsf{F}(\mathbf{x})\le\mathsf{F}(\mathbf{y})$,

(b) $\inf_{\mathbf{x}\in\mathcal{G}} \mathsf{F}(\mathbf{x}) = 0$,
(c) $\sup_{\mathbf{x}\in\mathcal{G}} \mathsf{F}(\mathbf{x}) = 1$.

Example 2. Noteworthy instances of inequity indices, see [2], include the normalized Gini-index:

$$\mathsf{G}(\mathbf{x}) = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} |x_i - x_j|}{(n-1)\sum_{i=1}^{n} x_i}, \tag{2}$$

the normalized Bonferroni-index:

$$\mathsf{B}(\mathbf{x}) = \frac{n}{n-1}\left(1 - \frac{\sum_{i=1}^{n} \frac{1}{n-i+1}\sum_{j=i}^{n} x_j}{\sum_{i=1}^{n} x_i}\right), \tag{3}$$

or the normalized de Vergottini-index:

$$\mathsf{V}(\mathbf{x}) = \frac{1}{\sum_{i=2}^{n}\frac{1}{i}}\left(\frac{\sum_{i=1}^{n}\frac{1}{i}\sum_{j=1}^{i} x_j}{\sum_{i=1}^{n} x_i} - 1\right). \tag{4}$$

It may be shown that all the indices may be computed in $O(n)$-time given a sorted $\mathbf{x}$. $\boxdot$

Now let F be a fixed inequity index and $g \in (0, 1]$ be some threshold. The Genie-based generalized centroid linkage criterion proceeds as follows. At the j-th step let $c_i = |C_i^{(j)}|$ and denote with $c_{(i)}$ the i-th smallest value in $(c_1, \ldots, c_{n-j})$. Now:

1. if $\mathsf{F}(c_{(n-j)}, \ldots, c_{(1)}) \le g$, then apply the standard generalized centroid linkage criterion:
$$\operatorname*{arg\,min}_{(u,v),u<v} \mathfrak{d}\left(\boldsymbol{\mu}_\varphi(C_u^{(j-1)}), \boldsymbol{\mu}_\varphi(C_v^{(j-1)})\right),$$
2. otherwise, i.e., if $\mathsf{F}(c_{(n-j)}, \ldots, c_{(1)}) > g$, restrict the search domain only to pairs of clusters such that one of them is of the smallest size:
$$\operatorname*{arg\,min}_{\substack{(u,v),u<v,\\ c_u=c_{(1)} \text{ or } c_v=c_{(1)}}} \mathfrak{d}\left(\boldsymbol{\mu}_\varphi(C_u^{(j-1)}), \boldsymbol{\mu}_\varphi(C_v^{(j-1)})\right).$$

Such a linkage scheme prevents drastic increases of the selected inequity measure and guarantees that small clusters are linked to some other ones much earlier. Whatever the choice of F, for $g = 1$ we obtain the ordinary generalized centroid linkage scheme. To recall, the original Genie algorithm [8] minimizes the value of $\min_{\mathbf{a}\in C_u^{(j-1)},\mathbf{b}\in C_v^{(j-1)}} \mathfrak{d}(\mathbf{a}, \mathbf{b})$ instead of $\mathfrak{d}\left(\boldsymbol{\mu}_\varphi(C_u^{(j-1)}), \boldsymbol{\mu}_\varphi(C_v^{(j-1)})\right)$, which for $g = 1$ reduces itself to the single linkage criterion.

Please cite this paper as: Gagolewski M., Cena A., Bartoszuk M., *Hierarchical clustering via penalty-based aggregation and the Genie approach*, In: Torra V. et al. (eds.), *Modeling Decisions for Artificial Intelligence* (*Lecture Notes in Artificial Intelligence* **9880**), Springer, 2016, pp. 191–202, doi:10.1007/978-3-319-45656-0_16.

3 Benchmarks

For testing purposes we use the benchmark data sets already studied in [8]. They are described in more detail and available for download at `http://www.gagolewski.com/resources/data/clustering/`. These include 21 data sets in the Euclidean space: *iris, iris5, s1, s2, s3, s4, a1, a2, a3, g2-2-100, g2-16-100, g2-64-100, unbalance, spiral, D31, R15, flame, jain, Aggregation, Compound, pathbased* as well as 6 non-Euclidean ones: strings over the $\{\mathtt{a}, \mathtt{c}, \mathtt{t}, \mathtt{g}\}$ alphabet (*actg1*, *actg2*, *actg3*, for use with the Levenshtein distance) and 0–1 vectors of fixed lengths (*binstr1*, *binstr2*, *binstr3*, for use with the Hamming distance). *digits2k_pixels* and *digits2k_points* were omitted from the analysis, as the performance of all the clustering algorithms is very weak in their case.

It is worth emphasizing that each data set comes with a vector of reference labels, which can be used to assess the performance of a clustering algorithm. For this purpose, we rely on the well-known notion of the FM-index, which gives the value 1 if a computed clustering (a dendrogram should be cut at an appropriate level) fully agrees with the reference one and 0 if it is totally discordant.

3.1 Choosing Different Inequity Measures

Firstly, let us study the effects of choosing different inequity measures on the original Genie algorithm (in [8], only the Gini-index was considered, but the algorithm's performance was already outstanding).

Figure 1 depicts the average FM-index computed over 21 Euclidean benchmark sets as a function of an inequity index threshold, g. Three measures of inequity are taken into account: the ones by Gini, Bonferroni, and de Vergottini. Additionally, the shaded regions span from the 1st to the 3rd quartiles of the empirical FM-index distributions.

The thresholds yielding the highest average FM-scores are equal to 0.2 in the case of the Bonferroni- and the Gini-index and 0.1 for the de Vergottini-index. Notably, in such cases the empirical FM-index distributions do not significantly differ from each other (as measured by the Wilcoxon (paired) signed rank test, all p-values > 0.1). This suggests that the actual choice of an inequity measure (at least, as far as indices given by (2)–(4) are concerned) is not so important – special attention should rather be paid to the threshold selection.

For a better understanding of the reasons why it is so, let us focus on the *Aggregation* dataset, which consists of 788 observations. Firstly, we determine the dendrogram using the average linkage method with respect to the Euclidean distance. Next, we cut the dendrogram so as to obtain $2, 3, \ldots, 788$ clusters. Then, for each data set partition obtained in this manner, we compute the three inequity indices for the cluster size distribution. Figure 2 depicts the relationships between values of the three inequity indices. Of course, we observe that each index is not a 1-to-1 function of another one, but the data points are highly correlated (pairwise correlation coefficients – Gini vs Bonferroni: Pearson's $r = 0.98$, Spearman's $\varrho = 0.97$; Gini vs de Vergottini: $r = 0.77$, $\varrho = 0.83$; Bonferroni vs

Please cite this paper as: Gagolewski M., Cena A., Bartoszuk M., *Hierarchical clustering via penalty-based aggregation and the Genie approach*, In: Torra V. et al. (eds.), *Modeling Decisions for Artificial Intelligence* (*Lecture Notes in Artificial Intelligence* **9880**), Springer, 2016, pp. 191–202, doi:10.1007/978-3-319-45656-0_16.

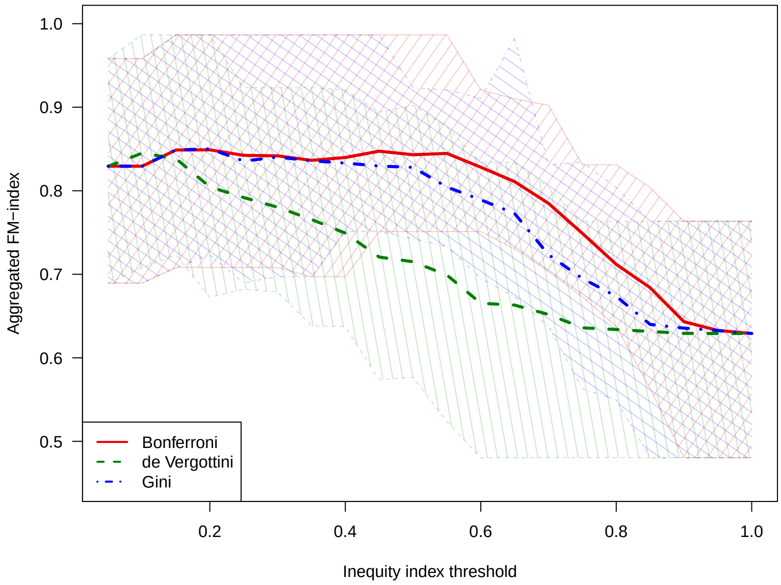

Fig. 1. The original Genie algorithm's performance depending on the choice of an inequity measure. The bold lines represent averaged FM-indices (21 benchmark sets), while the filled areas span from the 1st to the 3rd quartile of the empirical FM-index distribution.

de Vergottini: $r = 0.71$, $\varrho = 0.87$). As for the other data sets similar regularities are detected, we deduce that the actual choice of an inequity index is not as important as choosing the right threshold. However, as far as the current benchmark sets are concerned, from Fig. 1 it seems that such a threshold can be found much more easily in the case of the Gini- or Bonferroni-index than while the de Vergottini-index is in use.

3.2 Choosing Different Penalty Minimizers

Let us now compare the effects of choosing different penalty minimizers. Again, 21 data sets and the Euclidean metric is taken into account. We consider 5 different distance-based penalty minimizers: the *centroid* ($\varphi(\delta) = \delta^2, \mathcal{X}' = \mathbb{R}^d$), *median* ($\varphi(\delta) = \delta, \mathcal{X}' = \mathbb{R}^d$), *medoid* ($\varphi(\delta) = \delta, \mathcal{X}' = \{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}$), *medoid2* ($\varphi(\delta) = \delta^2, \mathcal{X}' = \{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}$), *medoid3* ($\varphi(\delta) = \delta^3, \mathcal{X}' = \{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}$). The three latter fusion functions are instances of set exemplars. Moreover, the Gini-index is used.

Figure 3 depicts the box-and-whisker plots for the FM-score distributions. Please note that the FM-indices may vary depending on the permutation of observations in a data set, because the distance matrix may consist of non-unique elements. Due to that, the median of 10 trials is computed (for different random rearrangements of the input points). For each generalized centroid, we

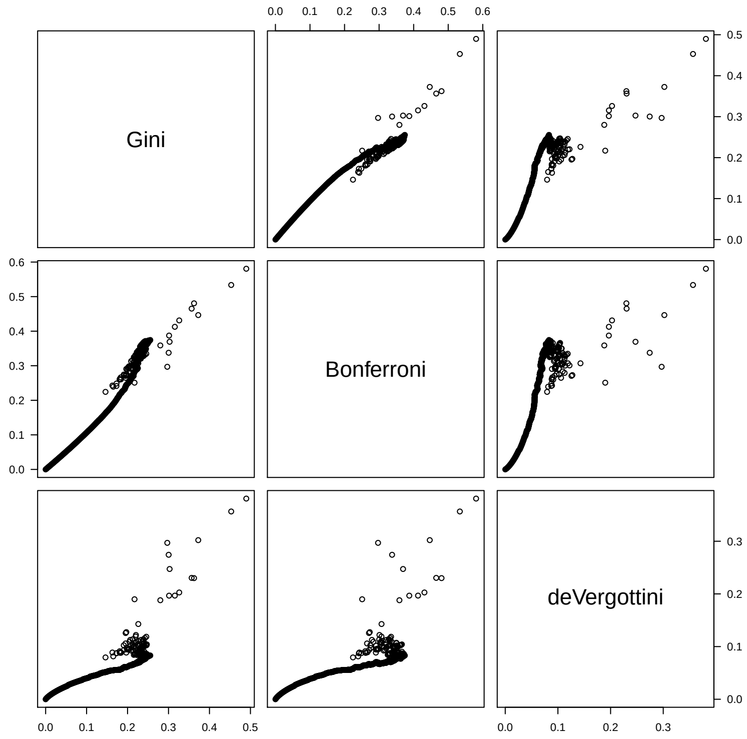

Fig. 2. Pairwise relationships between three inequity indices for the cluster size distributions as a function of the number of clusters in the case of the *Aggregation* data set.

report the results generated by considering two different Gini-index thresholds: 1.0 (no Genie correction applied at all) and the one maximizing the median among the 21 FM-index measurements.

We see that in each case the application of the Genie correction has a positive impact on the median FM-score. Among the considered distance-based penalty minimizers, *medoid2* yields the best results. When the Genie correction is in use, please observe the similarity between the results generated by relying on the 4 other fusion functions, especially between *medoid* and *median* (it should be noted that the latter is much more difficult to compute). The new linkage's performance is comparable with the Ward one. On the other hand, if $g = 1.0$, then the results are much more dependent on the choice of $\boldsymbol{\mu}$.

3.3 Non-Euclidean Benchmark Sets

As an example of the usefulness of the introduced algorithm in non-Euclidean spaces, let us now consider 6 different benchmark sets: *binstr1,2,3* (fixed-length

0-1 strings, the Hamming distance) and *actg1,2,3* (variable-length strings with elements in $\{\mathtt{a}, \mathtt{c}, \mathtt{t}, \mathtt{g}\}$, the Levenshtein distance).

Figure 4 depicts the FM-index distribution in the case of the single, complete, Ward, average, original Genie, and *medoid2*-based ($\varphi(\delta) = \delta^2, \mathcal{X}' = \{\mathbf{x}^{(i_1)}, \dots, \mathbf{x}^{(i_m)}\}$) linkages. First of all, we observe that not only a too large inequity index threshold, but also a too small one may lead to unsatisfying results. Secondly, again, the Genie correction has a positive impact on the aggregated FM-index.

4 An Algorithm to Compute the New Linkage

The pseudocode of an $O(n^3)$-time and $O(n^2)$-space algorithm to compute the introduced type of clustering task is given in Fig. 5 (the cost of computing a selected penalty minimizer and inequity index is not included). The core of the routine is a quite straightforward modification of Anderberg's algorithm [1] as given in [13]. Hence, we omit a detailed discussion on the role of the *minidx*, *mindist*, etc. objects. The applied modifications include the Genie-like correction (step 8.1.2, compare [8]) as well as a generic distance-based penalty minimizer instead of the Lance and Williams formula [11] in step 8.4. Note that the original Genie algorithm [8] runs in $O(n^2)$-time and $O(n)$-space.

5 Conclusion

We have proposed a generalization of the nearest centroid linkage scheme. First of all, generic distance-based penalty minimizers may be taken into account. Due to that, the algorithm can be computed in arbitrary spaces equipped with dissimilarity measures. Secondly, the clustering quality can be improved by using a correction for the inequity of cluster size distribution, as known from the original Genie algorithm.

We noted that the actual choice of an inequity index has no significant impact on the benchmark FM-measures (at least as far as the Gini-, Bonferroni-, and de Vergottini-indices are concerned). Interestingly, if the Genie correction is in use, the choice of a distance-based penalty minimizer is not very important too. If this is not the case, we observed that $\boldsymbol{\mu}$ based on the quadratic mean (*centroid, medoid2*) leads to more favorable results.

Finally, please note that, just like in the case of the original centroid and Genie linkage criteria, the "heights" at which clusters are merged are not necessarily being output in a nondecreasing order – the so-called reversals (inversions, departures from ultrametricity, see [12]) may occur. Therefore, they should be adjusted somehow when drawing corresponding dendrograms.

Acknowledgments. This study was supported by the National Science Center, Poland, research project 2014/13/D/HS4/01700.

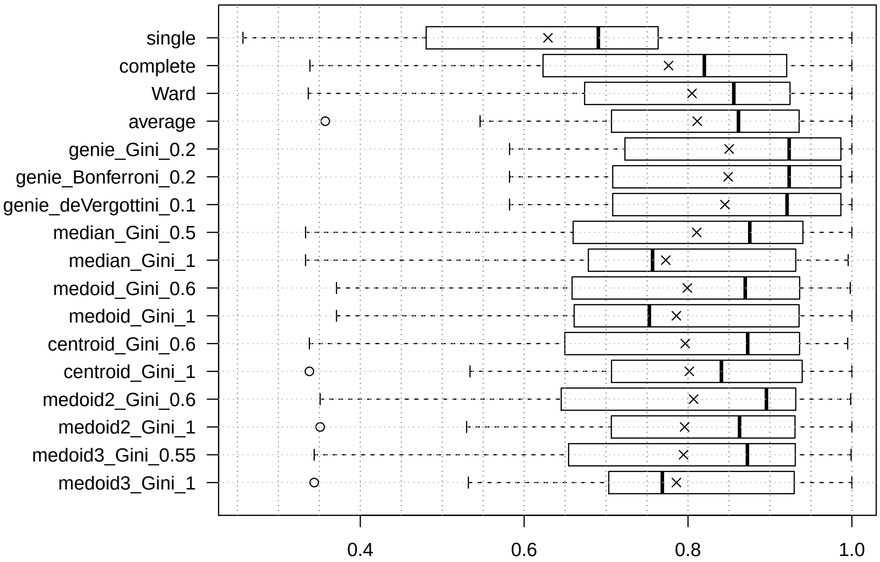

Fig. 3. Box-and-whisker plots representing FM-index distributions computed over 21 Euclidean benchmark sets for different clustering algorithms.

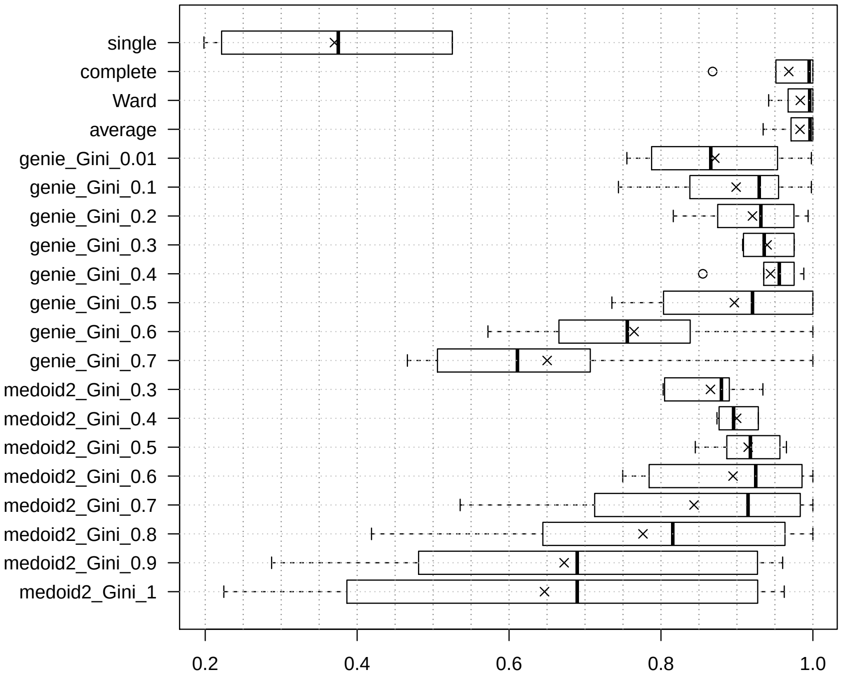

Fig. 4. Box-and-whisker plots representing FM-index distributions computed over 6 non-Euclidean benchmark sets for different clustering algorithms.

```
0. Input: $\mathbf{x}^{(1)},\dots,\mathbf{x}^{(n)}$ – $n$ objects, $g \in (0,1]$ – inequity index threshold,
   $\mathfrak{d}$ – a dissimilarity measure;
1. $ds$ = DisjointSets($\{1\},\{2\},\dots,\{n\}$);
2. $S = \{1,\dots,n\}$;
3. $L$ = List($\emptyset$);                                              /* output list */
4. Let $\mu$ be an array such that $\mu[i] = \mathbf{x}^{(i)}$ for all $i \in S$;    /* by idempotence */
5. Let $d$ be a matrix such that $d[i,j] = \mathfrak{d}(\mu[i],\mu[j])$;
6. for $i$ in $S \setminus \{n\}$:
   6.1. $minidx[i] = \arg\min_{j=i+1,\dots,n} d[i,j]$;
   6.2. $mindist[i] = d[i, minidx[i]]$;
7. $pq$ = MinPriorityQueue($minidx$,$mindist$);              /* add $n-1$ point pairs */
8. for $i = 1,2,\dots,n-1$:
   8.1. if $ds$.compute_inequity() $\le g$:                       /* e.g., the Gini-index */
      8.1.1. $(a,b,\delta)$ = $pq$.pop();                 /* the triple with the smallest $\delta$ */
      else:
      8.1.2. $(a,b,\delta)$ = $pq$.pop_conditional (                 /* Genie-like correction */
                 $(a,b,\delta)$: $ds$.size($a$) = $ds$.min_size() or $ds$.size($b$) = $ds$.min_size() );
   8.2. $L$.append($a,b,\delta$);                                 /* update the output list */
   8.3. $ds$.link($a$, $b$);                               /* extend cluster $b$ by $a$'s members */
   8.4. $\mu[b]$ = computePenaltyMinimizer$_{\mathfrak{d}}$($ds$.getClusterMembers($b$));
   8.5. $S = S \setminus \{a\}$;
   8.6. for $j$ in $S \setminus \{b\}$:
      8.6.1. $d[j,b] = d[b,j] = \mathfrak{d}(\mu[b],\, \mu[j])$;
   8.7. for $j$ in $S$ such that $j < a$ and $minidx[j] = a$:
      8.7.1. $minidx[j] = b$;
   8.8. for $j$ in $S$ such that $j < b$ and $d[j,b] < mindist[j]$:
      8.8.1. $minidx[j] = b$;
      8.8.2. $mindist[j] = d[j,b]$;
      8.8.3. $pq$.update($j$,$minidx[j]$, $mindist[j]$); /* update existing $(j,\cdot,\cdot)$ triple */
   8.9. $minidx[b] = \arg\min_{j=b+1,\dots,n} d[b,j]$;
   8.a. $mindist[b] = d[b,minidx[b]]$;
   8.b. $pq$.update($b$,$minidx[b]$, $mindist[b]$);        /* update existing $(b,\cdot,\cdot)$ triple */
9. return L;
```

Fig. 5. A pseudocode for the introduced clustering algorithm.

References

1. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
2. Aristondo, O., García-Lapresta, J., Lasso de la Vega, C., Marques Pereira, R.: Classical inequality indices, welfare and illfare functions, and the dual decomposition. Fuzzy Sets and Systems 228, 114–136 (2013)
3. Beliakov, G., Bustince, H., Calvo, T.: A Practical Guide to Averaging Functions. Springer (2016)

4. Bortot, S., Marques Pereira, R.: On a new poverty measure constructed from the exponential mean. In: Proc. IFSA/EUSFLAT'15, pp. 333–340. Atlantis Press (2015)
5. Cena, A., Gagolewski, M.: Fuzzy k-minpen clustering and k-nearest-minpen classification procedures incorporating generic distance-based penalty minimizers. In: Proc. IPMU'16, Part II (Communications in Computer and Information Science 611). Springer (2016), in press
6. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer (2013)
7. Gagolewski, M.: Data Fusion: Theory, Methods, and Applications. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland (2015)
8. Gagolewski, M., Bartoszuk, M., Cena, A.: Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. Information Sciences 363, 8–23 (2016)
9. García-Lapresta, J., Lasso de la Vega, C., Marques Pereira, R., Urrutia, A.: A new class of fuzzy poverty measures. In: Proc. of IFSA/EUSFLAT'15, pp. 1140–1146. Atlantis Press (2015)
10. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2013)
11. Lance, G.N., Williams, W.T.: A general theory of classificatory sorting strategies. The Computer Journal 9(4), 373–380 (1967)
12. Legendre, P., Legendre, L.: Numerical Ecology. Elsevier Science BV, Amsterdam (2003)
13. Müllner, D.: Modern hierarchical, agglomerative clustering algorithms. ArXiv:1109.2378 [stat.ML] (2011)
14. Olson, C.F.: Parallel algorithms for hierarchical clustering. Parallel Computing 21, 1313–1325 (1995)
15. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2016), http://www.R-project.org