Genie+OWA: Robustifying Hierarchical Clustering with OWA-based Linkages

Anna Cena[a,1], Marek Gagolewski[b,c,a]

[a] *Faculty of Mathematics and Information Science, Warsaw University of Technology*
ul. Koszykowa 75, 00-662 Warsaw, Poland
[b] *School of Information Technology, Deakin University, Geelong, VIC 3220, Australia*
[c] *Systems Research Institute, Polish Academy of Sciences*
ul. Newelska 6, 01-447 Warsaw, Poland

Abstract

We investigate the application of the Ordered Weighted Averaging (OWA) data fusion operator in agglomerative hierarchical clustering. The examined setting generalises the well-known single, complete and average linkage schemes. It allows to embody expert knowledge in the cluster merge process and to provide a much wider range of possible linkages. We analyse various families of weighting functions on numerous benchmark data sets in order to assess their influence on the resulting cluster structure. Moreover, we inspect the correction for the inequality of cluster size distribution – similar to the one in the Genie algorithm. Our results demonstrate that by robustifying the procedure with the Genie correction, we can obtain a significant performance boost in terms of clustering quality. This is particularly beneficial in the case of the linkages based on the closest distances between clusters, including the single linkage and its "smoothed" counterparts. To explain this behaviour, we propose a new linkage process called three-stage OWA which yields further improvements. This way we confirm the intuition that hierarchical cluster analysis should rather take into account a few nearest neighbours of each point, instead of trying to adapt to their non-local neighbourhood.

Keywords: hierarchical clustering, OWA, data fusion, aggregation, Genie.

[*] Corresponding author; Email: A.Cena@mini.pw.edu.pl.

1. Introduction

Cluster analysis has numerous fruitful applications in various fields, such as pattern recognition, image processing, data mining, bioinformatics etc., see, e.g., [7, 8, 22, 23, 30, 33, 37, 40, 41]. It is worth pointing out that this is by far one of the most important unsupervised learning techniques. Generally speaking, clustering aims to automatically discover the "hidden structure" of a given data set in the form of a partition of its elements. Its purpose is to create sets of objects in such a way that entities allocated to one group, called a cluster, are similar, while objects in distinct groups differ as much as possible from each other (with respect to some criteria), see, e.g., [12, 16, 17]).

Hierarchical methods are amongst the most useful clustering procedures. More precisely, agglomerative algorithms (e.g., [3, 13, 26–28]) provide a simple and intuitively appealing way to perform data segmentation without imposing any overly restrictive assumptions on the universe of discourse. At the same time, they generate a quite detailed representation of the underlying structure of a data set in the form of a hierarchy of nested partitions. In this setting, initially each cluster consists of only one data point. Then, in each step of the procedure, the "closest" clusters (with respect to a chosen measure) are merged. In order to evaluate the distance between two groups, some extension of a pairwise dissimilarity measure, called *linkage*, is used. The most commonly applied linkage schemes are typically put under the umbrella of the recursive Lance and Williams formula (see [19, 25]), which provides a general framework that includes the well-known single, complete and average linkages.

However, there exists a lesser-known generalisation of the aforementioned linkage functions – the ones that are constructed by applying the Ordered Weighted Averaging (OWA) [35] data fusion operator (based on a convex combination of order statistics). The use of the OWA-based linkage was initially introduced in [36] and re-invented in [29]. Because of the incorporation of the weights into the cluster merge procedure, OWA-based linkage allows one to interpolate between the extreme cases of the single, average and complete linkages. However, it has not yet been thoroughly evaluated. In particular, we claim that it would be interesting to know whether this approach leads to partitions that reflect the true underlying cluster structure better.

In [5, 6] we performed a preliminary study on this issue, i.e., we established an experimental framework that allows to evaluate the use of various OWA-based linkages. Moreover, we investigated the utilisation of the Genie algorithm-inspired correction [10] for the inequality of the cluster size distribution. The obtained results were promising, especially when the Genie correction was applied. Taking this into account, this paper gives a much more in-depth analysis of this method. What is more, we introduce various modifications and extensions of the OWA-based linkage, including a new three-stage procedure that outperforms state-of-the-art clustering algorithms.

The paper is organised as follows. In Section 2 we recall the OWA-based linkage scheme as well as review different generators of OWA weights. In Section 3 we recall the main ideas behind the Genie clustering algorithm. Next, in Section 4 we evaluate its performance on a comprehensive set of benchmark data, discuss the best performing weighting strategies and investigate the effects of introducing the Genie correction. In Section 5 we introduce a new three-stage OWA-based clustering procedure that improves clustering quality even further. Finally, in Section 6 we conclude the paper.

2. OWA-based linkage

Let us start with a brief summary of some well-known facts concerning hierarchical clustering. We will follow the notation used in [6]. Let $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ be the input data set. Moreover, let $\mathfrak{d} : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ denote a pairwise dissimilarity measure, e.g., the Euclidean distance. The l-partition of $\mathcal{X}$ is defined as $\mathcal{C} = \{C_1, \ldots, C_l\}$, where $\emptyset \neq C_u \subset \mathcal{X}$, $C_u \cap C_v = \emptyset$ for $u \neq v$ and $\bigcup_{u=1}^{l} C_u = \mathcal{X}$. As it was stated above, hierarchical clustering algorithms form a whole hierarchy of nested partitions, i.e., $\mathcal{C}^* = \{\mathcal{C}^{(0)}, \mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(m)}\}$, such that $m < n$, $\mathcal{C}^{(j)} = \{C_1^{(j)}, \ldots, C_{n_j}^{(j)}\}$ is an n_j-partition of $\mathcal{X}$, $n_j > n_{j+1}$ and it holds that $(\forall j = 0, \ldots, m-1)\, (\forall C_u^{(j)} \in \mathcal{C}^{(j)})$ $(\exists C_v^{(j+1)} \in \mathcal{C}^{(j+1)})$ such that $C_u^{(j)} \subseteq C_v^{(j+1)}$ and $(\forall z \neq v)\, C_u^{(j)} \cap C_z^{(j+1)} = \emptyset$. Note that by "cutting" the hierarchy at any chosen level, a well-defined partition for the required number of clusters can be obtained. Therefore, by using hierarchical techniques one can get a better insight into the underlying structure of input data.

Figure 1 presents the most straightforward (naïve) approach to the general agglomerative hierarchical clustering. The procedure goes as follows. Initially each cluster is a singleton, i.e., $\mathcal{C}^{(0)} = \{C_1^{(0)}, \ldots, C_n^{(0)}\}$, $C_i^{(0)} = \{\mathbf{x}^{(i)}\}$,

```
$\mathcal{C}^{(0)} = \{C_1^{(0)}, \dots, C_n^{(0)}\}$, $C_i^{(0)} = \{\mathbf{x}^{(i)}\}$;
for $j = 1, \dots, n-1$ do
    $(u, v) = \operatorname*{arg\,min}_{(u,v), u<v} \mathfrak{d}^*(C_u^{(j-1)}, C_v^{(j-1)})$;
    $C_u^{(j)} = C_u^{(j-1)} \cup C_v^{(j-1)}$;
    $C_i^{(j)} = C_i^{(j-1)}$ for $u \neq i < v$;
    $C_i^{(j)} = C_{i+1}^{(j-1)}$ for $i > v$;
end for
```

Figure 1: A general agglomerative hierarchical clustering algorithm

$i = 1, \dots, n$. Proceeding from the $(j-1)$-th to the j-th step, the algorithm merges clusters $C_u^{(j-1)}$ and $C_v^{(j-1)}$ with $u < v$ such that:

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

where $\mathfrak{d}^* : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to [0, \infty]$ denotes a chosen linkage function. In result we obtain $C_i^{(j)} = C_i^{(j-1)}$ 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 $i > v$.

Intuitively, a *linkage* is an extension of a pairwise dissimilarity measure $\mathfrak{d}$ that allows to quantify the dissimilarity between whole sets of points, for which it holds $\mathfrak{d}^*(\{\mathbf{x}^{(a)}\}, \{\mathbf{x}^{(b)}\}) = \mathfrak{d}(\mathbf{x}^{(a)}, \mathbf{x}^{(b)})$ for all a, b. Amongst the noteworthy linkages we find:

- the single linkage:

$$\mathfrak{d}^*_{\text{MIN}}(C_u, C_v) = \min_{\mathbf{u} \in C_u, \mathbf{v} \in C_v} \mathfrak{d}(\mathbf{u}, \mathbf{v}),$$

- the complete linkage:

$$\mathfrak{d}^*_{\text{MAX}}(C_u, C_v) = \max_{\mathbf{u} \in C_u, \mathbf{v} \in C_v} \mathfrak{d}(\mathbf{u}, \mathbf{v}),$$

- the average linkage:

$$\mathfrak{d}^*_{\text{AMean}}(C_u, C_v) = \frac{1}{|C_u||C_v|} \sum_{\mathbf{u} \in C_u, \mathbf{v} \in C_v} \mathfrak{d}(\mathbf{u}, \mathbf{v}).$$

The above linkages can be expressed both in terms of the Lance and Williams formula [19, 25] and OWA operators [35]. Let us now formally introduce the definition of an OWA operator and the corresponding OWA-based linkage [29, 36].

Definition 1. For any vector $\mathbf{w} = (w_1, w_2, \dots, w_z) \in [0, 1]$ such that $\sum_{i=1}^{z} w_i = 1$, the z-ary *ordered weighted averaging* operator $\mathrm{OWA}_{\mathbf{w}} : [0, \infty]^z \to [0, \infty]$, associated with $\mathbf{w}$ is given by:

$$\mathrm{OWA}_{\mathbf{w}}(d_1, d_2, \dots, d_z) = \sum_{i=1}^{z} w_i d_{(i)},$$

where $d_{(i)}$ denotes the i-th greatest value, i.e., $d_{(1)} \geq d_{(2)} \geq \cdots \geq d_{(z)}$.

In the case of a clustering procedure, we shall be interested in conceiving OWA operators as extended aggregation functions, i.e., defined for any number of arguments. Recall that to define a fixed-arity OWA we need z weights. To define an extended OWA, we will follow the convention introduced in [4, 24] and consider a whole *weighting triangle*, $\triangle = (w_{i,z} \in [0, 1],\ z \in \mathbb{N}, i = 1, \dots, z : (\forall z) \sum_{i=1}^{z} w_{i,z} = 1)$, which can be graphically represented as:

$$\begin{array}{cccc}
w_{1,1} & & & \\
w_{1,2} & w_{2,2} & & \\
w_{1,3} & w_{2,3} & w_{3,3} & \\
\vdots & \vdots & \vdots & \ddots
\end{array}$$

Here, the z-th row, whose elements sum to 1, gives the weights used when aggregating z-ary sequences. However, this time, z may vary freely from sequence to sequence the OWA operator is applied on.

Definition 2. An *extended OWA operator*, $\mathrm{OWA}_{\triangle} : \bigcup_{z=1}^{\infty} [0, \infty]^z \to [0, \infty]$, for a given weighting triangle $\triangle = (w_{i,z} \in [0, 1],\ z \in \mathbb{N}, i = 1, \dots, z : (\forall z) \sum_{i=1}^{z} w_{i,z} = 1)$, is defined as:

$$\mathrm{OWA}_{\triangle}(d_1, d_2, \dots, d_z) = \sum_{i=1}^{z} w_{i,z} d_{(i)}$$

for any $d_1, d_2, \dots, d_z \in [0, \infty]$ and any z.

The above allows us to formalise the definition of an OWA operator-based linkage.

Definition 3. The for a given weighting triangle $\triangle$, the OWA$_\triangle$-*based linkage* is given by:

$$\mathfrak{d}^*_{\mathrm{OWA}_\triangle}(C_u, C_v) = \mathrm{OWA}_\triangle(d_1, d_2, \dots, d_z),$$

where $C_u = \{\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(|C_u|)}\}$ and $C_v = \{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(|C_v|)}\}$ are some point sets, $z = |C_u||C_v|$, and

$$d_{i+|C_u|(j-1)} = \mathfrak{d}(\mathbf{u}^{(i)}, \mathbf{v}^{(j)}),$$

for all $i = 1, \dots, |C_u|$ and $j = 1, \dots, |C_v|$ giving the distances between all the pairs of points from the two sets.

It is easily seen that for weights like $w_{i,z} = \frac{1}{z}$ for all i we obtain the average linkage, $w_{z,z} = 1$ and $w_{i,z} = 0$ for $i < z$ gives us the single linkage scheme, and $w_{1,z} = 1$, $w_{i,z} = 0$, $i > 1$ yields the complete linkage.

The crucial question is how to generate new weighting triangles in a systematic manner. In this paper we adopt the setting proposed in [18, 35] (see also [1, 14]) which includes:

(a) $w_{i,z} = \frac{c_i}{\sum_{j=1}^{z} c_j}$, where a sequence $(c_1, c_2, \dots)$ is such that $c_i \geq 0$ for all $i = 1, 2, \dots$ and $c_1 > 0$, see, e.g., [18];

(b) $w_{i,z} = \mathsf{w}\left(\frac{i}{z}\right) - \mathsf{w}\left(\frac{i-1}{z}\right)$, where $\mathsf{w} : [0,1] \to [0,1]$ is a nondecreasing function with $\mathsf{w}(0) = 0$ and $\mathsf{w}(1) = 1$, see, e.g., [35];

(c) $w_{i,z} = \frac{c_{z-i+1}}{\sum_{j=1}^{z} c_j}$, where a sequence $(c_1, c_2, \dots)$ is such that $c_i \geq 0$ for all $i = 1, 2, \dots$ and $c_z > 0$;

(d) $w_{i,z} = \mathsf{w}\left(\frac{z-i+1}{z}\right) - \mathsf{w}\left(\frac{z-i-1}{z}\right)$, where $\mathsf{w} : [0,1] \to [0,1]$ is a nondecreasing function with $\mathsf{w}(1) = 0$ and $\mathsf{w}(0) = 1$.

A number of noteworthy weighting scenarios shall be reviewed in Section 4.

3. Genie correction

In [10] we have introduced the Genie algorithm – a single linkage-based method which robustifies the cluster merge process so as to prevent the formation of size-unbalanced groups and whose reference implementations are included in R package `genie` (`https://cran.r-project.org/web/packages/genie/`) and Python package `genieclust` (`https://pypi.org/project/genieclust/`). Intuitively, if the inequity of the distribution of the cluster sizes at some step of the procedure exceeds a chosen threshold, the merging of the smallest point group is enforced. In order to assess the degree of the cluster size imbalance, the notion of an inequality index G is incorporated. Here, we assume G is the Gini-index.

Definition 4. The Gini-index of a given sample is given by:

$$\mathrm{G}(c_1, \dots, c_l) = \frac{\sum_{i=1}^{l-1} \sum_{j=i+1}^{l} |c_i - c_j|}{(n-1)\sum_{i=1}^{l} c_i}.$$

Note that $\mathrm{G}(c, c, \dots, c) = 0$ (balanced sample) and that G is bounded from above by 1.

Fix a threshold $g \in (0, 1]$. Genie modifies Line 3 of the algorithm presented in Figure 1 as follows. Proceeding from the $(j-1)$-th to the j-th step of the clustering procedure, $j = 1, \dots, n-1$, merge clusters $C_u^{(j-1)}$, $C_v^{(j-1)}$ such that:

1. if $\mathrm{G}(c_1, \dots, c_{n-j+1}) \le g$, where $c_i = |C_i^{(j-1)}|$, apply the standard linkage criterion:
$$(u, v) = \operatorname*{arg\,min}_{u<v} \mathfrak{d}^* \left(C_u^{(j-1)}, C_v^{(j-1)} \right);$$
2. otherwise, enforce the merging of a cluster of the smallest size:
$$(u, v) = \operatorname*{arg\,min}_{\substack{u<v;\\ c_u = c_{(n-j+1)} \text{ or} \\ c_v = c_{(n-j+1)}}} \mathfrak{d}^* \left(C_u^{(j-1)}, C_v^{(j-1)} \right).$$

Please note that for $g = 1.0$ the original Genie algorithm is equivalent to the single linkage method, because it was built under the assumption that $\mathfrak{d}^* = \mathfrak{d}^*_{\mathrm{MIN}}$. It turns out that the performance of the Genie algorithm is significantly better than not only that of the standard hierarchical clustering routines, but also other segmentation algorithms, see [10] and below. The possible application of the Genie correction to other linkages was studied in [11] in case of the nearest centroid-based functions. A preliminary investigation of the potential benefits that this procedure may bring to the OWA-based linkages ($\mathfrak{d}^* = \mathfrak{d}^*_{\mathrm{OWA}}$) was performed in [6]. This investigation shall be significantly extended below.

4. Benchmark data analysis

In this section we identify the OWA weight generation schemes that yield the best overall clustering quality. Note that the papers where the OWA-based linkages were originally introduced [29, 36] did not discuss this issue at all. Additionally here we shall answer the question whether the Genie correction brings any benefits to the clustering procedure.

4.1. Experiment setting

The numerical experiments are aligned within the framework of the so-called "external clustering evaluation". We considered a suite of 51 benchmark data sets that were often used in the literature to assess and compare the performance of various clustering procedures, see [6, 9–11, 15, 32]. The data sets consist of balanced or imbalanced groups of points in $\mathbb{R}^d$ and differ with respect to dimensionality d and the number of input points n, see Table 1. All data sets are available at `https://gitlab.com/cenka/Clustering`.

Moreover, each data set is equipped with a sequence of reference labels, i.e., a vector assigning each point to its true cluster that has been indicated by external experts. We assume that the number of clusters to detect, K, is known in advance, as this parameter is defined by the labels set. Also note that throughout this paper, the pairwise distance function is set to be the squared Euclidean distance. No feature selection/engineering/transformation is applied on data.

In order to measure the degree of concordance between a partition generated by a clustering algorithm and the true (reference, expert-provided) labels, we shall rely on the widely applied notion of the Adjusted Rand-index (AR-index, see [20]).

Definition 5. The AR-index between two K-partitions $\mathcal{C} = \{C_1, \dots, C_K\}$ and $\mathcal{C}' = \{C'_1, \dots, C'_K\}$ of the set $\mathcal{X}$ of cardinality n is defined as:

$$\text{AR-index}(\mathcal{C}, \mathcal{C}') = \frac{\binom{n}{2} \sum_{u=1}^{K} \sum_{v=1}^{K} \binom{m_{u,v}}{2} - \sum_{u=1}^{K} \binom{m_{u,\cdot}}{2} \sum_{v=1}^{K} \binom{m_{\cdot,v}}{2}}{\frac{1}{2}\binom{n}{2}\left(\sum_{u=1}^{K} \binom{m_{u,\cdot}}{2} + \sum_{v=1}^{K} \binom{m_{\cdot,v}}{2}\right) - \sum_{u=1}^{K} \binom{m_{u,\cdot}}{2} \sum_{v=1}^{K} \binom{m_{\cdot,v}}{2}},$$

where $m_{u,v} = |C_u \cap C'_v|$, $m_{u,\cdot} = \sum_{v=1}^{K} m_{u,v}$, and $m_{\cdot,v} = \sum_{u=1}^{K} m_{u,v}$.

Table 1: Basic properties of benchmark data sets considered: the number of points n, data set dimensionality d and the true number of clusters K.

	$\boldsymbol{n}$	$\boldsymbol{d}$	$\boldsymbol{K}$		$\boldsymbol{n}$	$\boldsymbol{d}$	$\boldsymbol{K}$
iris5	105	4	3	iris	150	4	3
flame	240	2	2	pathbased	300	2	3
spiral	312	2	3	jain	373	2	2
Compound	399	2	6	R15	600	2	15
Aggregation	788	2	7	g2-2-100	2048	2	2
g2-16-100	2048	16	2	g2-64-100	2048	64	2
a1	3000	2	20	D31	3100	2	31
s1	5000	2	15	s2	5000	2	15
s3	5000	2	15	s4	5000	2	15
a2	5250	2	35	unbalance	6500	2	8
a3	7500	2	50	Hepta	212	3	7
Lsun	400	2	3	Tetra	400	3	4
Target	770	2	6	TwoDiamonds	800	2	2
Atom	800	3	2	Chainlink	1000	3	2
WingNut	1016	2	2	GolfBall	4002	3	1
EngyTime	4096	2	2	Wine	178	13	3
Norm-density	200	2	2	sonar	208	60	2
Glass	214	9	3	Line-uneven	250	2	2
SwirlDots	250	2	2	SPECT	267	22	2
SwirlDots-outlier	280	2	2	SwirlDots-noisy	300	2	2
haberman	306	3	2	ecoli	336	7	8
Ionosphere	351	34	2	wdbc	569	30	2
BreastCancer	683	9	2	pimaDiabetes-norm	768	8	2
Parabolic	1000	2	2	Ring	1000	2	2
Square	1000	2	2	XOR	1000	2	2
Ring-outliers	1030	2	3	Ring-noisy	1050	2	2
segmentation	2310	19	7				

It is important to note that the AR-index has expected value equal to 0 in the case of two random (uniformly-distributed) partitions and is bounded from above by 1 in the case of a perfect agreement. We should stress that the reference labels are not taken into account during the clustering process, therefore the procedure is fully unsupervised. These are referred to only during the evaluation stage.

4.2. Weighting triangles

As the main focus of this work is on assessing the practical utility of OWA-based linkages, we must consider a very wide range of weighting triangles. Let us note that in [29] only one weighting triangle $\triangle$ was actually used, namely:

$$w_{i,z} = \frac{\varphi(i;\mu_z,\sigma_z)}{\sum_{j=1}^{z}\varphi(j;\mu_z,\sigma_z)},$$

where $\varphi(\cdot;\mu_z,\sigma_z)$ denotes the probability density function of the normal distribution $\mathrm{N}(\mu_z,\sigma_z)$:

$$\varphi(i;\mu_z,\sigma_z) = \frac{1}{\sqrt{2\pi\sigma_z^2}}\exp\left(-\frac{(i-\mu_z)^2}{2\sigma_z^2}\right), \tag{1}$$

$\mu_z = \frac{z+1}{2}$ and $\sigma_z = \sqrt{\frac{1}{z}\sum_{i=1}^{z}(i-\mu_z)^2}$, see [34].

Tables 2 and 3 provide the complete list of the weighting triangles we have studied. We were interested in weights that interpolate around/between the single and complete linkages, sample quartiles, various means and some mixtures of the above. The settings for the weighting triangle generation were as follows:

- σ_z in each use of φ was set to $z/3$ as well as $z/9$;
- p in the trimmed and Winsorised means was set to 0.25;
- (a, b) in the Yager step function was set to $(0, 0.5)$, $(0.5, 1)$ as well as $(0.3, 0.7)$.

Note that the scenarios 1–16 were also included in our preliminary study [6], but were analysed on a much more limited benchmark data sample.

Table 2: $\mathrm{OWA}_{\triangle}$-based linkages studied (part I)

	Alias	**Weighting triangle** $(\triangle = (w_{i,z} \in [0,1],\ z \in \mathbb{N}, i = 1, \dots, z))$
1	AMean (average)	$w_{i,z} = \frac{1}{z}$
2	MIN (single)	$w_{z,z} = 1$ $w_{i,z} = 0$ for $i < z$
3	MAX (complete)	$w_{1,z} = 1$ $w_{i,z} = 0$ for $i > z$
4	Q2 (median)	$w_{(z+1)/2,z} = 1$ for $z = 2k + 1$ $w_{z/2,z} = w_{z/2+1,z} = 0.5$ for $z = 2k$
5	Q1 (first quartile)	Median taken over the lower half of the vector sorted nondecreasingly (without median)
6	Q3 (third quartile)	Median taken over the upper half of the vector sorted nondecreasingly (without median)
7	Norm [34]	$w_{i,z} = \frac{\varphi(i;\mu_z,\sigma_z)}{\sum_{j=1}^{z} \varphi(j;\mu_z,\sigma_z)}$ $\mu_z = \frac{z+1}{2}$, $\sigma_z = \sqrt{\frac{1}{z}\sum_{i=1}^{z}(i-\mu_z)^2}$
8	$\mathit{smooth}\mathrm{MIN}_{\sigma_z}$	$w_{i,z} = \frac{\varphi(i;z,\sigma_z)}{\sum_{j=1}^{z} \varphi(j;z,\sigma_z)}$
9	$\mathit{smooth}\mathrm{MAX}_{\sigma_z}$	$w_{i,z} = \frac{\varphi(i;1,\sigma_z)}{\sum_{j=1}^{z} \varphi(j;1,\sigma_z)}$

Table 3: OWA$_\triangle$-based linkages (part II)

	Alias	**Weighting triangle** $(\triangle = (w_{i,z} \in [0,1],\ z \in \mathbb{N}, i = 1, \ldots, z))$
10	$\mathit{smooth}\mathrm{MIN\&MAX}_{\sigma_z}$	$v_i = \max\Big\{\varphi(i; 1, \sigma_z), \varphi(i; z, \sigma_z)\Big\}$ $w_{i,z} = \frac{v_i}{\sum_{j=1}^{z} v_j}$
11	$\mathit{smooth}\mathrm{Q3}_{\sigma_z}$	$w_{i,z} = \frac{\varphi(i; \frac{1}{4}z, \sigma_z)}{\sum_{j=1}^{z} \varphi(j; \frac{1}{4}z, \sigma_z)}$
12	$\mathit{smooth}\mathrm{Q1}_{\sigma_z}$	$w_{i,z} = \frac{\varphi(i; \frac{3}{4}z, \sigma_z)}{\sum_{j=1}^{z} \varphi(j; \frac{3}{4}z, \sigma_z)}$
13	$\mathit{smooth}\mathrm{Q1\&Q3}_{\sigma_z}$	$v_i = \max\Big\{\varphi(i; \frac{3z}{4}, \sigma_z), \varphi(i; \frac{z}{4}, \sigma_z)\Big\}$ $w_{i,z} = \frac{v_i}{\sum_{i=1}^{z} v_i}$
14	$\mathrm{TriMean}_p$	$w_{i,z} = \frac{1}{z-2k}$ for $i = k+1, \ldots, z-k$ $w_{i,z} = 0$ otherwise, $k = \lfloor pz \rfloor$
15	$\mathrm{WinMean}_p$	$w_{i,z} = \frac{1}{z}$ for $i = k+2, \ldots, z-k-1$ $w_{i,z} = \frac{(k+1)}{z}$ for $i = k+1, z-k$ $w_{i,z} = 0$ otherwise, $k = \lfloor pz \rfloor$
16	Yager-step$_{(a,b)}$ [35]	$w_{i,z} = Q\Big(\frac{i}{z}; a, b\Big) - Q\Big(\frac{i-1}{z}; a, b\Big)$ $Q(x; a, b) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases}$ $0 \leq a < b \leq 1$
17	$\star$ ArMIN$_k$ [38]	$w_{i,z} = \frac{1}{\min\{k,z\}}$ for $i \geq \max\{0, z-k+1\}$, $w_{i,z} = 0$ for $i < \max\{0, z-k+1\}$
18	$\star$ $\mathit{smooth}\mathrm{MIN}_\delta$	$w_{i,z} = \begin{cases} \frac{\varphi(i;z,\delta)}{\sum_{j=1}^{z} \varphi(i;z,\delta)} & \text{for } i \geq 3\delta, \\ 0 & \text{for } i < 3\delta \end{cases}$

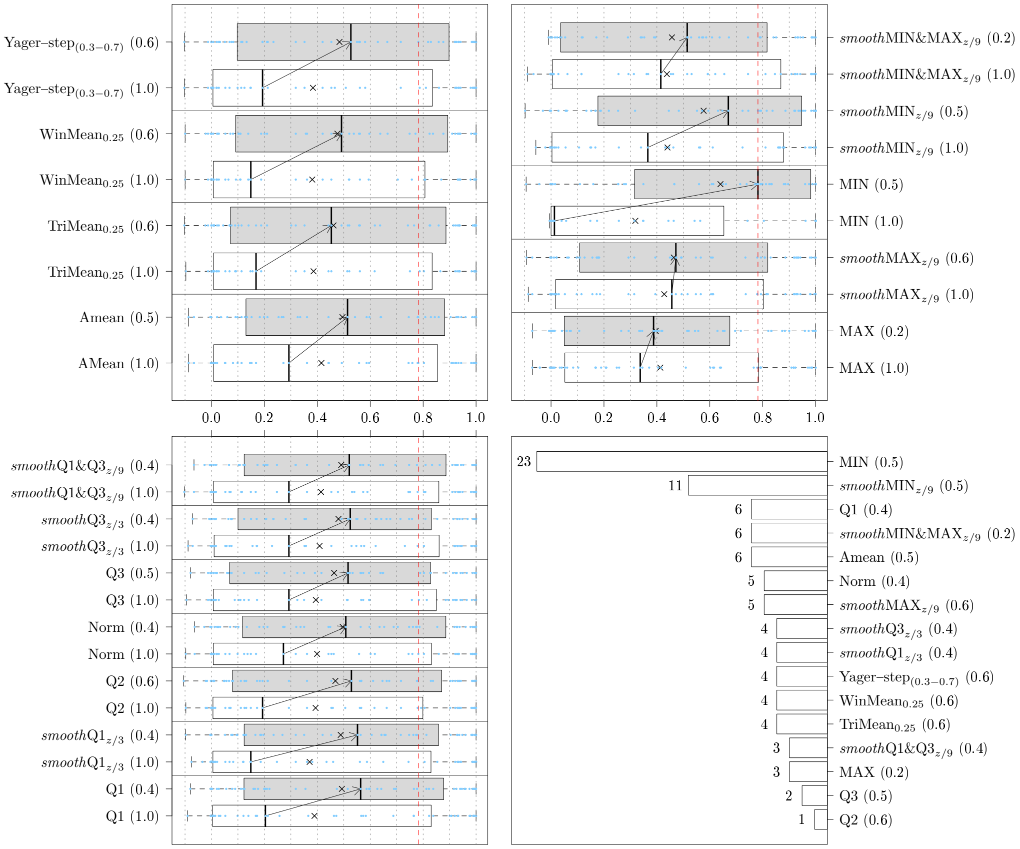

Figure 2: Three groups of box-and-whisker plots for the AR-index distribution for each OWA linkage scenario with ($g < 1.0$; grey) and without ($g = 1.0$; white) the Genie correction. The arrows are included to emphasise the rate of the increment of the median AR-index that is observed when we robustify the procedure with the Genie correction. The bar plot in the bottom-right part represents the total number of benchmark sets for which an indicated OWA-based linkage is the winner (the best threshold g is given in parentheses). The original Genie algorithm (MIN-based) outperforms all the other methods.

4.3. Experiment results

Let us proceed with an empirical validation of the discussed OWA-based linkages. Each box-and-whisker plot in Figure 2 summarises the distribution of the 51 AR-indexes between the set of true (reference) labels and the clustering generated by applying a specific OWA-based linkage. To recall, the higher the AR-index, the more similar the obtained clustering to the expert-provided one. The considered schemes cover scenarios 1–16 grouped into three categories: averages, extreme values (single linkage MIN and complete linkage MAX and their “smoothed“ counterparts) and also the quartiles and the functions interpolating around them. For each scenario, we consider two cases:

- without the Genie correction (denoted with (1.0)) — white boxes,
- with the Gini-index threshold $g \in \{0.1, 0.2, 0.3, \ldots, 0.9\}$ selected so that the median of the AR-index on all data sets was maximised (the optimal g is reported in round brackets) — grey boxes.

The red dotted line corresponds to the highest median AR-index obtained in this experiment.

Lack of Genie correction leads to low clustering quality. First of all, note that when the Genie correction is not applied (i.e., $g = 1.0$), all the results are far from satisfactory – the best median, equal to 0.456, is obtained using $\mathit{smooth}\mathrm{MAX}_{z/9}$. Moreover, it is easily seen that most AR-index distributions are similar to each other – with the exception of the single linkage scheme, which has subpar performance, e.g., the median of the AR-index is as low as 0.013.

To confirm this observation, we have compared the partitions generated by different methods with each other. Figure 3 gives the medians of the AR-indexes computed between label vectors outputted by each pair of linkages (directly, with no reference to the true label set). Indeed, only the single linkage (and, to a much lesser extent, the complete linkage) yields much different partitions.

Genie correction significantly improves the results. It turns out that by introducing the Genie correction we obtain significant improvements in clustering quality for all the considered weighting schemes. The arrows in Figure 2 were introduced to emphasise the increases in values of the medians. Surprisingly, the single linkage MIN with $g = 0.5$ (i.e., the original Genie algorithm) now yields the best results, with median AR-index of 0.782. The second best median (0.670) is obtained for its smoothed version, $\mathit{smooth}\mathrm{MIN}_{z/9}$.

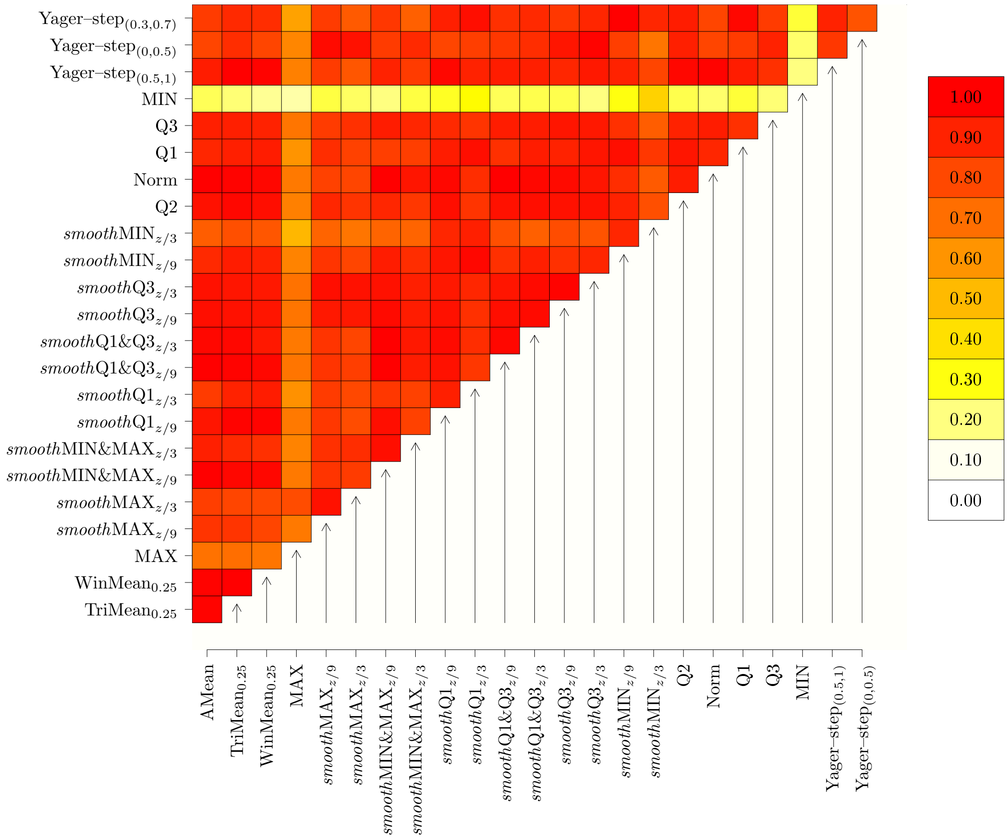

Figure 3: The similarity between the OWA-based linkages in terms of the medians of AR-indexes computed between each pair of label vectors generated by all the weighting scenarios. Here, no Genie correction used, i.e., $g = 1.0$. The higher the aggregated AR-index, the more similar the outputs generated by different linkages to each other. The single linkage (MIN) significantly differs from the other OWA-based weighting schemes.

Nearest neighbours provide more meaningful information. The bar plot in the bottom-right part of Figure 2 presents the total number of times each weighting scenario led to the maximal overall agreement with the reference labels (the comparison between the AR-indexes assumed the indexes are rounded to 3 decimal places). The original Genie algorithm (based on single linkage, MIN) achieves the best agreement with the reference labels on 23 data sets, while *smooth*MIN$_{z/9}$ is the second best (it tops in 11 of the cases). In other words, the most expert-concordant partitions are obtained when using the single linkage and/or its smoothed version together with the Genie correction.

In order to investigate this more thoroughly, let us take into account two additional weighting triangle generation schemes – scenarios 17–18 (denoted with $\star$ in Table 3). Thanks to this we may focus on the weighting triangles that interpolate around the single linkage – the new scenarios are based on few nearest neighbours of each data point. For ArMIN_k and $\mathit{smooth}\mathrm{MIN}_\delta$, the median AR-indexes are maximised for $k = 10$ and $\delta = 2$, respectively.

Figure 6 gives the box plots for the AR-index distributions and the number of cases where each scenario is a winner (for now, the reader is kindly asked to ignore the two presented cases of the OWA^3 scheme which shall be discussed in the next section as well as the results for the other algorithms). It turns out that the distributions are quite similar ($\mathit{smooth}\mathrm{MIN}_2$ yields a better 1st quartile of the AR-index). We have performed the Wilcoxon (paired) signed rank test with the null hypothesis: pairwise differences in the AR-indexes between the 4 best linkages (MIN(0.5), $\mathrm{ArMIN}_{10}(0.5)$, $\mathit{smooth}\mathrm{MIN}_2(0.4)$, and $\mathit{smooth}\mathrm{MIN}_{z/9}(0.5)$) are symmetric around 0. The differences were found to be statistically insignificant (at significance level $\alpha = 0.05$) in each case (the only p-value < 0.1 was obtained for MIN(0.5) vs. $\mathit{smooth}\mathrm{MIN}_2(0.4)$, $p = 0.074$). Hence, we would rather be opting for the use of the original Genie algorithm (MIN(0.5)) as it allows for an efficient implementation based upon a minimum spanning tree of the pairwise similarity matrix, see [10].

Optimal threshold g. Let us also more closely examine the impact the choice of the threshold g has on the clustering performance. Figure 4 depicts the median values of the AR-indexes for all the considered weighting scenarios and thresholds g. Once again we observe a significant increase in clustering quality when the Genie correction is applied on the nearest neighbours-based linkages, especially for MIN, $\mathit{smooth}\mathrm{MIN}_\delta$ and ArMIN_k. The top median AR-index corresponds to $g \in [0.4, 0.5]$. Also note that we have identified that the robustified algorithm is quite stable with respect to small changes of g – modifying the threshold slightly does not change its behaviour drastically.

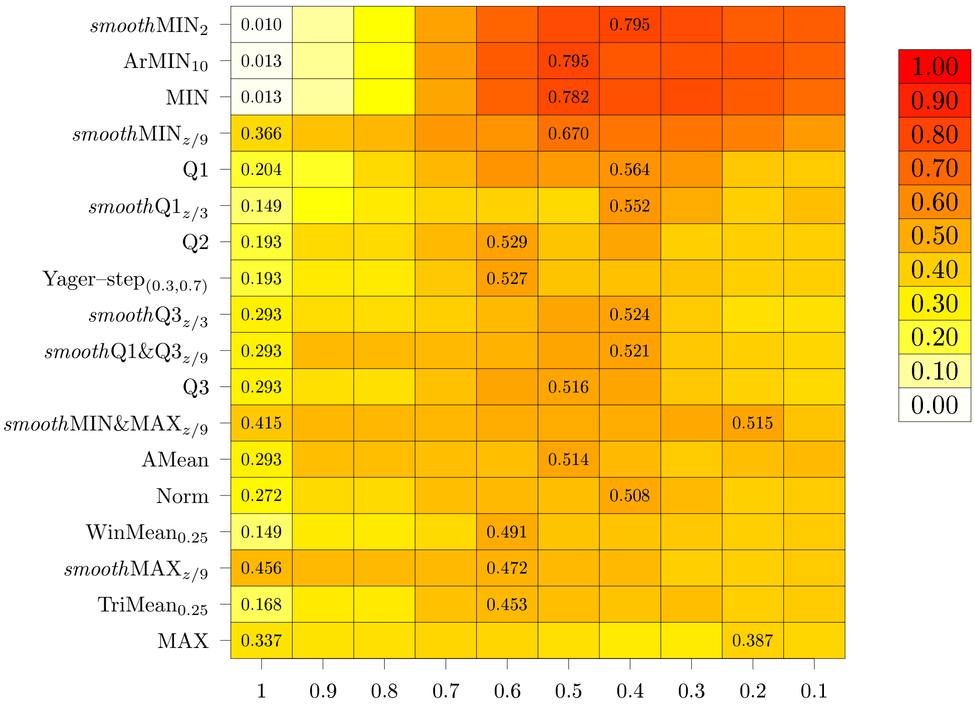

Figure 4: Median of the AR-indexes for different weighting scenarios and the Gini-index thresholds $g \in \{0.1, 0.2, \dots, 1.0\}$.

5. Three-stage OWA-based linkage, OWA^3

We have noted that the best clustering quality is obtained by considering the closest neighbours of each data point and applying the Genie correction for the cluster size inequality. In order to verify whether we can improve the results even further, let us now introduce a modification of the OWA-based linkage that – instead of aggregating all the pairwise distances anonymously – first, for each point separately, summarises the distances to their own nearest neighbours and then combines the intermediate aggregates (see Figure 5 for an illustration). This way, we will be able to link the clusters based on, e.g., the averaged distance to all the points' nearest neighbours or the farthest of all the 5th nearest neighbours.

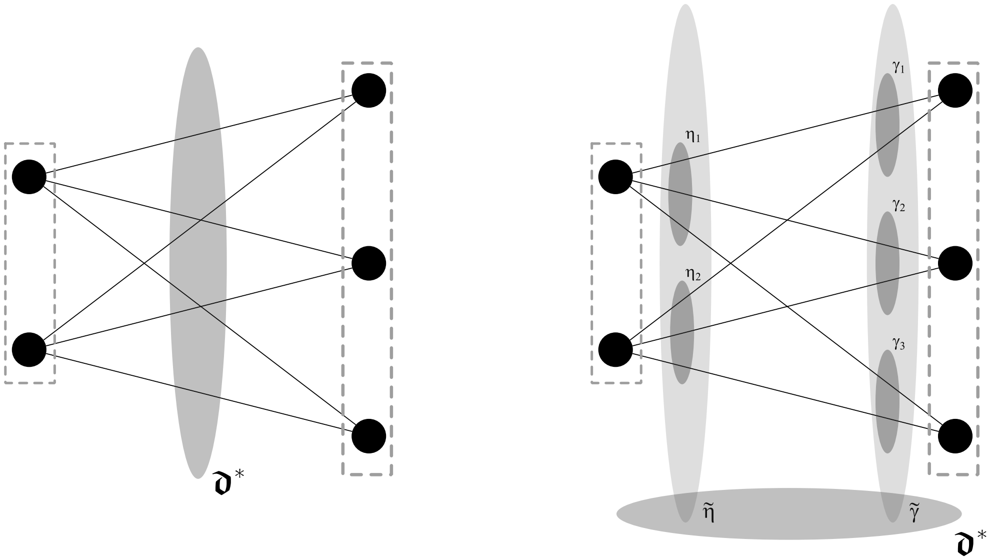

Figure 5: The original OWA-based linkage (left) aggregates all the pairwise distances anonymously. On the other hand, the new three-stage procedure (right) summarises the information on each point's aggregated nearest neighbours data.

5.1. Method definition

Let us formalise the aforementioned idea. The new aggregation process will be divided into three phases:

$$\begin{aligned}\mathfrak{d}^*(C_a^{(j-1)}, C_b^{(j-1)}) = \mathsf{A}^3\Bigg(&\mathsf{A}^2\Bigg(\mathsf{A}^1\Big(\mathfrak{d}(\mathbf{x}^{(l)}, \mathbf{y}^{(i)}) :\ \mathbf{y}^{(i)} \in C_b^{(j-1)}\Big) :\ \mathbf{x}^{(l)} \in C_a^{(j-1)}\Bigg),\\ &\mathsf{A}^2\Bigg(\mathsf{A}^1\Big(\mathfrak{d}(\mathbf{x}^{(l)}, \mathbf{y}^{(i)}) :\ \mathbf{x}^{(l)} \in C_a^{(j-1)}\Big) :\ \mathbf{y}^{(i)} \in C_b^{(j-1)}\Bigg)\Bigg),\end{aligned}$$

where $\mathsf{A}^1 : [0,\infty]^{1,2,\dots} \to [0,\infty]$ and $\mathsf{A}^2 : [0,\infty]^{1,2,\dots} \to [0,\infty]$ are extended OWA functions, and A^3 is a binary OWA operator, i.e., $\mathsf{A}^3 : [0,\infty]^2 \to [0,\infty]$.

The above dissimilarity degree fusion process, from now on called OWA^3, can be re-written as follows. Let $\mathbf{y}^{(i)} \in C_b^{(j-1)}$, $i = 1, 2, \dots, n_b$ and $\mathbf{x}^{(l)} \in C_a^{(j-1)}$, $l = 1, 2, \dots, n_a$, where $n_b = |C_b^{(j-1)}|$ and $n_a = |C_a^{(j-1)}|$.

(1) **Step 1.** For each $\mathbf{x}^{(l)} \in C_a^{(j-1)}$ from the first cluster, function A^1 is used to aggregate all the dissimilarities between $\mathbf{x}^{(l)}$ and the points in the second cluster, i.e.:

$$\forall \mathbf{x}^{(l)} \in C_a^{(j-1)}: \ \eta_l = \mathsf{A}^1\Big(\mathfrak{d}(\mathbf{x}^{(l)}, \mathbf{y}^{(1)}), \dots, \mathfrak{d}(\mathbf{x}^{(l)}, \mathbf{y}^{(c_b)})\Big).$$

The same procedure is applied for each point in the second cluster:

$$\forall \mathbf{y}^{(i)} \in C_b^{(j-1)}: \ \gamma_i = \mathsf{A}^1\Big(\mathfrak{d}(\mathbf{x}^{(1)}, \mathbf{y}^{(i)}), \dots, \mathfrak{d}(\mathbf{x}^{(c_a)}, \mathbf{y}^{(i)})\Big).$$

(2) **Step 2.** Function A^2 is used to summarise the aggregated dissimilarities $\boldsymbol{\eta} = (\eta_1, \dots, \eta_{c_a})$ and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_{c_b})$, i.e.:

$$\tilde{\eta} = \mathsf{A}^2(\eta_1, \dots, \eta_{c_a}),$$

$$\tilde{\gamma} = \mathsf{A}^2(\gamma_1, \dots, \gamma_{c_b}).$$

(3) **Step 3.** Finally, A^3 is applied on $\tilde{\eta}$ and $\tilde{\gamma}$ to obtain the fused inter-cluster distance, i.e.:

$$\mathfrak{d}^*(C_a^{(j-1)}, C_b^{(j-1)}) := \mathsf{A}^3(\tilde{\eta}, \tilde{\gamma}).$$

It is easily seen that the above scheme includes the single linkage (all three aggregation functions set to MIN), average (AMean) and complete (Max) linkages.

5.2. Experiment results

Taking into account the results obtained in previous section, let us perform some experiments concerning the new three stage OWA-based linkage. The functions A^1 and A^2 will be chosen amongst the nearest neighbours-based OWA operators, namely, ArMIN_k, $\mathit{smooth}\mathrm{MIN}_\delta$, $\mathit{smooth}\mathrm{MIN}_{\sigma_z}$ and MIN. In this case we consider various combinations of their underlying parameters. Then the binary A^3 function will be either AMean, MAX or MIN.

Table 4 gives the median AR-indexes for parameter sets maximising this measure. Again, we note that the Genie correction provides a significant boost in the observed clustering quality. The best results are observed for the OWA triples ($\mathit{smooth}\mathrm{MIN}_{z/10}$, ArMIN_5, MIN) and (ArMIN_5, $\mathit{smooth}\mathrm{MIN}_{z/10}$, MIN). These scenarios are denoted as OWA^3_{17} and OWA^3_{19}, respectively.

Figure 6 presents the box-and-whiskers plots for the AR-index distributions for each "winning" strategy. We note that the new three-stage procedure outperforms the other methods.

Table 4: Medians of the AR-indexes for the new three-stage OWA-based linkages with (g) and without (1.0) the Genie correction.

	A^1	A^2	A^3	**Med** (1.0)	**Med** (g)
OWA_1^3	ArMIN_5	ArMIN_{10}	AMean	0.040	0.832 (0.5)
OWA_2^3	$smooth\mathrm{MIN}_{z/50}$	$smooth\mathrm{MIN}_{z/15}$	AMean	0.343	0.856 (0.6)
OWA_3^3	$smooth\mathrm{MIN}_{15}$	$smooth\mathrm{MIN}_2$	AMean	0.043	0.784 (0.5)
OWA_4^3	$smooth\mathrm{MIN}_{z/15}$	MIN	AMean	0.377	0.847 (0.6)
OWA_5^3	$smooth\mathrm{MIN}_5$	MIN	AMean	0.057	0.806 (0.4)
OWA_6^3	ArMIN_{15}	MIN	AMean	0.042	0.779 (0.5)
OWA_7^3	$smooth\mathrm{MIN}_{z/15}$	ArMIN_{10}	AMean	0.148	0.864 (0.6)
OWA_8^3	$smooth\mathrm{MIN}_5$	ArMIN_5	AMean	0.057	0.806 (0.5)
OWA_9^3	ArMIN_2	$smooth\mathrm{MIN}_{z/20}$	AMean	0.148	0.830 (0.2)
OWA_{10}^3	ArMIN_5	$smooth\mathrm{MIN}_5$	AMean	0.040	0.814 (0.5)
OWA_{11}^3	ArMIN_5	ArMIN_2	MIN	0.097	0.818 (0.2)
OWA_{12}^3	$smooth\mathrm{MIN}_{z/20}$	$smooth\mathrm{MIN}_{z/50}$	MIN	0.349	0.885 (0.5)
OWA_{13}^3	$smooth\mathrm{MIN}_5$	$smooth\mathrm{MIN}_5$	MIN	0.040	0.785 (0.5)
OWA_{14}^3	$smooth\mathrm{MIN}_{z/15}$	MIN	MIN	0.349	0.848 (0.6)
OWA_{15}^3	$smooth\mathrm{MIN}_5$	MIN	MIN	0.057	0.779 (0.2)
OWA_{16}^3	ArMIN_{10}	MIN	MIN	0.040	0.779 (0.5)
OWA_{17}^3	$smooth\mathrm{MIN}_{z/10}$	ArMIN_5	MIN	0.416	0.896 (0.6)
OWA_{18}^3	$smooth\mathrm{MIN}_5$	ArMIN_{10}	MIN	0.040	0.798 (0.5)
OWA_{19}^3	ArMIN_5	$smooth\mathrm{MIN}_{z/10}$	MIN	0.306	0.896 (0.6)
OWA_{20}^3	ArMIN_5	$smooth\mathrm{MIN}_5$	MIN	0.057	0.819 (0.5)
OWA_{21}^3	ArMIN_5	ArMIN_5	MAX	0.018	0.783 (0.5)
OWA_{22}^3	$smooth\mathrm{MIN}_{z/20}$	$smooth\mathrm{MIN}_{z/50}$	MAX	0.217	0.853 (0.6)
OWA_{23}^3	$smooth\mathrm{MIN}_5$	$smooth\mathrm{MIN}_5$	MAX	0.040	0.787 (0.5)
OWA_{24}^3	$smooth\mathrm{MIN}_{z/50}$	MIN	MAX	0.057	0.791 (0.5)
OWA_{25}^3	$smooth\mathrm{MIN}_{10}$	MIN	MAX	0.043	0.774 (0.5)
OWA_{26}^3	ArMIN_2	MIN	MAX	0.040	0.779 (0.2)
OWA_{27}^3	$smooth\mathrm{MIN}_{z/20}$	ArMIN_2	MAX	0.148	0.834 (0.4)
OWA_{28}^3	$smooth\mathrm{MIN}_2$	ArMIN_2	MAX	0.124	0.825 (0.5)
OWA_{29}^3	ArMIN_5	$smooth\mathrm{MIN}_{z/15}$	MAX	0.059	0.787 (0.4)
OWA_{30}^3	ArMIN_5	$smooth\mathrm{MIN}_5$	MAX	0.018	0.814 (0.5)

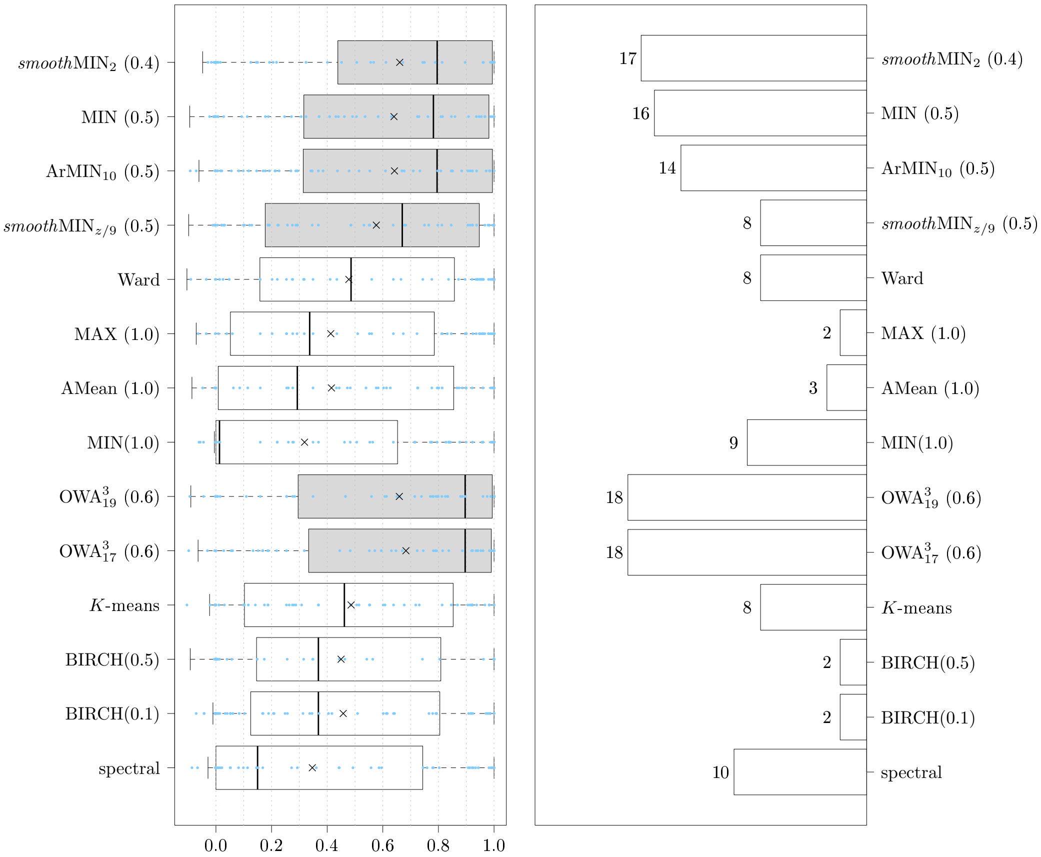

Figure 6: Box plots of the AR-index distributions for the best weighting triangles and thresholds g in the standard OWA-based linkage as well as its new, three stage version (denoted with OWA^3_{17} and OWA^3_{19}) and some other state-of-the art clustering algorithms. The bar plot on the right side represents the total number of benchmark sets for which an indicated weighting scenario and threshold g is the winner.

5.3. A comparison with other state-of-the art clustering algorithms

Figure 6 also gives the AR-indexes for the clustering algorithms available in the scikit-learn package (which allow specifying the number of clusters K in advance) for Python, see `http://scikit-learn.org`:

- K-means – the classical algorithm [21] as implemented in `sklearn.cluster.KMeans()`,
- Spectral clustering – see [31], `sklearn.cluster.SpectralClustering()` with default parameters,
- BIRCH – proposed in [39], `sklearn.cluster.Birch(threshold=t)` with $t = 0.1$ or $t = 0.5$.

Yet, these methods are left far behind the new algorithms robustified by means of the Genie correction.

Note that despite the fact that the OWA triples ($smooth\mathrm{MIN}_{z/10}$, ArMIN_5, MIN) and (ArMIN_5, $smooth\mathrm{MIN}_{z/10}, \mathrm{MIN}$) yield the best results, it is the original Genie algorithm (MIN-based) that allows for the most efficient computer implementation – it can be computed in $O(n\sqrt{n})$ time provided that a minimum spanning tree of the pairwise distance graph (which can be found by performing not more than $n(n-1)/2$ pairwise distance computations) is given.

6. Conclusions

First of all, the investigation carried out in this paper shows that the use of the OWA linkages allows to obtain high quality partitions within the hierarchical clustering framework. However, this is true only when the Genie correction for the inequality of cluster sizes is applied. In each case the Genie correction leads to a significant improvement in clustering quality. We confirmed the intuition that agglomerative algorithms should rather take into account a few nearest neighbours of each of the points under consideration, instead of trying to adapt to their non-local context.

Secondly, in terms of the median agreement with reference labels, the new three-stage OWA-based linkage yields the best results. Nevertheless, the number of parameters required by this procedure is high and thus tuning them up might be difficult. On the other hand, the original Genie (single linkage-based) algorithm, due to its simplicity, has a very efficient computer implementation (see R package `genie`, `https://cran.r-project.org/web/packages/genie/`, and Python package `genieclust`, `https://pypi.org/project/genieclust/`). Yet, we note that all the near-neighbour approaches can benefit from using spatial search data structures such as kd-trees or vp-trees in the case of low-dimensional data sets or approximate algorithms in high-dimensional ones.

Finally, our analysis was based on the Euclidean distance between the points. For future research, it could be interesting to incorporate other metrics (also featuring variable weighting and selection), for example OWA-based ones that have already been investigated in the context of k-means clustering in [2].

Acknowledgments

Marek Gagolewski's research was supported by the Czech Science Foundation through the project of No. 18-06915S.

References

[1] G. Beliakov, S. James, Stability of weighted penalty-based aggregation functions, Fuzzy Sets and Systems 226 (2013) 1–18.

[2] G. Beliakov, S. James, G. Li, Learning Choquet-integral-based metrics for semisupervised clustering, IEEE Transactions on Fuzzy Systems 19 (2011) 562–574.

[3] R. Cai, Z. Zhang, A.K. Thung, C. Dai, Z. Hao, A general framework of hierarchical clustering and its applications, Information Sciences 272 (2014) 29–48.

[4] T. Calvo, G. Mayor, J. Torrens, J. Suner, M. Mas, M. Carbonell, Generation of weighting triangles associated with aggregation functions, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 8 (2000) 417–451.

[5] A. Cena, Adaptive hierarchical clustering algorithms based on data aggregation methods, Ph.D. thesis, Systems Research Institute, Polish Academy of Sciences, 2018. In Polish.

[6] A. Cena, M. Gagolewski, OWA-based linkage and the Genie correction for hierarchical clustering, in: Proc. FUZZ-IEEE'17, IEEE, 2017. No. 8015652.

[7] Y. Djenouri, A. Belhadi, P. Fournier-Viger, J.C.W. Lin, Fast and effective cluster-based information retrieval using frequent closed itemsets, Information Sciences 453 (2018) 154–167.

[8] C. Euán, H. Ombao, J. Ortega, The hierarchical spectral merger algorithm: A new time series clustering procedure, Journal of Classification 35 (2018) 71–99.

[9] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Applied Intelligence 48 (2018) 4743–4759.

[10] M. Gagolewski, M. Bartoszuk, A. Cena, Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm, Information Sciences 363 (2016) 8–23.

[11] M. Gagolewski, A. Cena, M. Bartoszuk, Hierarchical clustering via penalty-based aggregation and the Genie approach, in: V. Torra, Y. Narukowa, G. Navarro-Arribas, C. Yanez (Eds.), Modeling Decisions for Artificial Intelligence (Lecture Notes in Artificial Intelligence 9880), Springer, 2016, pp. 191–202.

[12] G. Gan, C. Ma, J. Wu, Data Clustering: Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability, Philadelphia, Alexandria, 2007.

[13] R.J. Gil-Garcia, J.M. Badia-Contelles, A. Pons-Porrata, A general framework for agglomerative hierarchical clustering algorithms, in: 18th International Conference on Pattern Recognition (ICPR'06), volume 2, 2006, pp. 569–572.

[14] D. Gomez, K. Rojas, J. Montero, J. Rodriguez, G. Beliakov, Consistency and stability in aggregation operators: An application to missing data problems, International Journal of Computational Intelligence Systems 7 (2014) 595–604.

[15] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering: A comparative experimental study, Fuzzy Sets and Systems 161 (2010) 522–543.

[16] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: Data mining, inference, and prediction, Springer, 2009.

[17] A.K. Jain, Richard C. Dubes, Algorithms for clustering data, Prentice Hall, 1988.

[18] B. Jamison, S. Orey, W. Pruitt, Convergence of weighted averages of independent random variables, Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 4 (1965) 40–44.

[19] G. Lance, W. Williams, A general theory of classification sorting strategies: 1. Hierarchical systems, Computer Journal (1967) 373–380.

[20] H. Lawrence, A. Phipps, Comparing partitions, Journal of Classification 2 (1985) 193–218.

[21] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., volume 1, University of California Press, Berkeley, 1967, pp. 281–297.

[22] J. Majewska, S. Truskolaski, Cluster-mapping procedure for tourism regions based on geostatistics and fuzzy clustering: Example of Polish districts, Current Issues in Tourism (2018) 1–21.

[23] D. Małyszko, S.T. Wierzchoń, Standard and genetic k-means clustering techniques in image segmentation, in: 6th International Conference on Computer Information Systems and Industrial Management Applications (CISIM'07), IEEE, 2007, pp. 299–304.

[24] G. Mayor, T. Calvo, On extended aggregation functions, in: Proc. IFSA 1997, volume 1, Academia, Prague, 1997, pp. 281–285.

[25] G.W. Milligan, Ultrametric hierarchical clustering algorithms, Psychometrika 44 (1979) 343–346.

[26] D. Müllner, Modern hierarchical, agglomerative clustering algorithms, ArXiv:1109.2378 [stat.ML] (2011).

[27] D. Müllner, fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python, Journal of Statistical Software 53 (2013) 1–18.

[28] F. Murtagh, A survey of recent advances in hierarchical clustering algorithms, The Computer Journal 26 (1983) 354–359.

[29] E. Nasıbov, C. Kandemır-Cavas, OWA-based linkage method in hierarchical clustering: Application on phylogenetic trees, Expert Systems with Applications 38 (2011) 12684–12690.

[30] O. Şeref, Y.J. Fan, E. Borenstein, W.A. Chaovalitwongse, Information-theoretic feature selection with discrete k-median clustering, Annals of Operations Research 263 (2018) 93–118.

[31] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 888–905.

[32] A. Ultsch, Clustering with SOM: U*C, in: Workshop on Self-Organizing Maps, WSOM 2005, 2005, pp. 75–82.

[33] D. Vu, S. Georgievska, S. Szoke, A. Kuzniar, V. Rober, fMLC: Fast multi-level clustering and visualization of large molecular datasets, Bioinformatics 34 (2018) 1577–1579.

[34] Z. Xu, An overview of methods for determining OWA weights, International Journal of Intelligent Systems 20 (2005) 843–865.

[35] R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Transactions on Systems, Man, and Cybernetics 18 (1988) 183–190.

[36] R.R. Yager, Intelligent control of the hierarchical agglomerative clustering process, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 30 (2000) 835–845.

[37] H. Yahyaoui, H.S. Own, Unsupervised clustering of service performance behaviors, Information Sciences 422 (2018) 558–571.

[38] P. Yildirim, D. Birant, K-linkage: A new agglomerative approach for hierarchical clustering, Advances in Electrical and Computer Engineering 17 (2017) 77–88.

[39] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: An efficient data clustering method for large databases, in: Proc. ACM SIGMOD International Conference on Management of Data – SIGMOD ’96, pp. 103–114.

[40] A. Zhou, Y. Wang, J. Zhang, Objective extraction via fuzzy clustering in evolutionary many-objective optimization, Information Sciences 509 (2020) 343–355.

[41] J. Zhu, Z. Jiang, G.D. Evangelidis, C. Zhang, S. Pang, Z. Li, Efficient registration of multi-view point sets by k-means clustering, Information Sciences 488 (2019) 205–218.

Please cite this paper as:
A. Cena, M. Gagolewski, Genie+OWA: Robustifying hierarchical clustering with OWA-based linkages, Information Sciences 520, 324–336, 2020, doi:10.1016/j.ins.2020.02.025