# CHAPTER 3

# CLUSTERING APPROACHES TO NETWORKS

VLADIMIR BATAGELJ[1,2,3]

[1] IMFM, Ljubljana
[2] IAM, University of Primorska, Koper
[3] NRU HSE, Moscow

## 3.1 Introduction

Clustering and classification are two related activities sometimes used as synonyms. In clustering the goal is to identify in a given set of units groups (clusters, classes) of (usually) similar units. In classification a given unit has to be assigned to the corresponding (predefined) group. These two activities are embedded in our language and are therefore basic for most of our daily tasks.

The earliest classification systems were taxonomies of animals and plants: Shen Nung, China, ∼3000 BC and Ebers Papyrus, Egypt, ∼1500 BC. A theoretical framework was proposed by Aristotle (384–322 BC). The taxonomic systems were improved by Linnaeus (1707–1778), Darwin (1809–1882), DNA (1953) and PhyloCode (1998).

The first steps towards "numeric" clustering procedures were done in the first half of 20th century by defining different (dis)similarity measures such as Czekanowski coefficient (1909), coefficient of racial likeness (Pearson, 1926), generalized distance (Mahalanobis, 1936), etc. First methods were proposed inside biometrics and psychometrics by Driver and Kroeber (1932), Forbes (1933), Zubin (1938), Sturtevant (1939) etc. Kruskal's minimum spanning tree algorithm (1956) was predated by unnoticed Borůvka (1926) and Jarnik (1930) algorithms.

The development of cluster analysis started in fifties and resulted in some fundamental books: Sokal and Sneath: *Principles of Numerical Taxonomy* (1963) [52], Jardin and Sibson: *Mathematical Taxonomy* (1971) [37], Benzécri: *L'analyse des données* (1973) [13], Anderberg: *Cluster Analysis for Application* (1973) [2], Hartigan: *Clustering algorithms* (1975) [34] and later Jain and Dubes: *Algorithms for clustering data* (1988) [36] and Kaufman and Rousseeuw: *Finding Groups in Data: An Introduction to Cluster Analysis* (1990) [41].

Two streams of clustering research emerged – inside pattern recognition and data analysis. The data analytic stream was initially attached to psychometric community until 1985 when the IFCS (International Federation of Classification Socities) was established with its own conference and journals *Journal of classification* (published by CSNA from 1984) and *Advances in Data Analysis and Classification* (from 2007). Conference proceedings are published in a Springer series *Studies in classification, data analysis, and knowledge organization*. In ninities the interests of the clustering community extended to data analysis and data science (Hayashi [35]). For details about the development of IFCS see Bock [15]. At the turn of the millenium clustering was somehow absorbed also in data mining as one of its constituents. In social network analysis clustering problem is known as blockmodeling [24].

In this chapter we first present an optimization framework for a general clustering problem. In the second part we discuss clustering of networks and in networks.

## 3.2 Clustering

In data analysis we usually follow the scheme

| real or imaginary world | objects of analysis | data analysis |
|---|---|---|
| CONCEPTS | UNITS | DESCRIPTIONS |
| $\{\mathrm{X}\}$ | $\mathrm{X}$ | $[\mathrm{X}]$ |
| | $\longleftrightarrow$ formalization | $\longleftrightarrow$ operationalization |
| { produced cars of type T } | car of type T | [ seats=4, max-speed= . . . ] |

A *unit* $\mathrm{X} \in \mathscr{U}$ is represented by a vector/*description* $X \equiv [\mathrm{X}] = [x_1, x_2, ..., x_m]$ from the set $[\mathscr{U}]$ of all possible descriptions of units from *space* $\mathscr{U}$. $x_i = V_i(\mathrm{X})$ is the value of the $i$-th of selected properties or *variables* on X. Variables can be measured on different *scale*s: nominal, ordinal, interval, rational, absolute [51]. In concrete analysis the *set of units* of our interset $\mathbf{U} \subset \mathscr{U}$ is (usually) finite, $n = |\mathbf{U}|$.

There exist other kinds of descriptions of units: symbolic object [**?** ], list of keywords from a text, chemical formula, node in a given graph, digital picture, etc.

### 3.2.1 Clustering problem

Let us start with the formal setting of the clustering problem. We shall use the following notation: a nonempty subset of units $C$, $\emptyset \subset C \subseteq \mathbf{U}$, is called a *cluster*. A set of clusters, $\mathbf{C} = \{C_i\}$ , forms a *clustering*. $\Phi$ denotes the set of *feasible clusterings*. A *criterion function*, $P : \Phi \to \mathbb{R}_0^+$, evaluates the quality of a clustering.

With these notions we can express the *clustering problem* $(\Phi, P, \min)$ as follows:

Determine the clustering $\mathbf{C}^\star \in \Phi$ for which

$$P(\mathbf{C}^\star) = \min_{\mathbf{C}\in\Phi} P(\mathbf{C})$$

Since the set of units $\mathbf{U}$ is finite, the set of feasible clusterings is also finite. Therefore the set $\mathrm{Min}(\Phi, P)$ of all solutions of the problem (optimal clusterings) is not empty. (In theory) the set $\mathrm{Min}(\Phi, P)$ can be determined by the complete search. We shall denote the value of criterion function for an optimal clustering by $\min(\Phi, P)$.

Generally the clusters of clustering $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ need not to be pairwise disjoint; yet, the clustering theory and practice mainly deal with clusterings which are the *partition*s of $\mathbf{U}$

$$\bigcup_{i=1}^{k} C_i = \mathbf{U} \qquad \text{and} \qquad i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

Each partition determines an equivalence relation in $\mathbf{U}$, and vice versa. We shall denote the set of all partitions of $\mathbf{U}$ into $k$ clusters (classes) by $P_k(\mathbf{U})$.

### 3.2.2 Criterion functions

The criterion function is usually constructed as follows. Joining the individual units into a cluster $C$ we make a certain "error", we create certain "tension" among them – we denote this quantity by $p(C)$. A *simple* criterion function $P(\mathbf{C})$ combines these "partial/local errors" into a "global error". Usually it takes the form:

S. $P(\mathbf{C}) = \sum_{C\in\mathbf{C}} p(C)$, or

M. $P(\mathbf{C}) = \max_{C\in\mathbf{C}} p(C)$

which can be unified and generalized in the following way: Let $(\mathbb{R}, \oplus, e, \leq)$ be an ordered abelian monoid then:

$\oplus$. $P(\mathbf{C}) = \bigoplus_{C\in\mathbf{C}} p(C)$

The *cluster-error* $p(C)$ has usually the properties:

$$p(C) \geq 0 \qquad \text{and} \qquad \forall \mathrm{X} \in \mathbf{U} : p(\{\mathrm{X}\}) = 0$$

In the continuation we shall assume that these properties of $p(C)$ hold.

Often also

$$p(C_1 \cup C_2) \geq p(C_1) \oplus p(C_2)$$

holds for disjoint clusters, $C_1 \cap C_2 = \emptyset$. In such a case we have for simple criterion functions $\min(P_{k+1}\mathbf{U}), P) \leq \min(P_k(\mathbf{U}), P)$ – we fix the value of $k$ and set $\Phi \subseteq P_k(\mathbf{U})$.

To express the cluster-error $p(C)$ we define on the space of units $\mathscr{U}$ a *dissimilarity* $d : \mathscr{U} \times \mathscr{U} \to \mathbb{R}_0^+$ for which we require:

D1. $\forall \mathrm{X} \in \mathscr{U} : d(\mathrm{X}, \mathrm{X}) = 0$

D2. *symmetric*: $\forall \mathrm{X}, \mathrm{Y} \in \mathscr{U} : d(\mathrm{X}, \mathrm{Y}) = d(\mathrm{Y}, \mathrm{X})$

Table 3.1: Dissimilarities on $\mathbb{R}^m$

| n | measure | definition | range | note |
|---|---|---|---|---|
| 1 | Euclidean | $\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2}$ | $[0,\infty)$ | $M(2)$ |
| 2 | Sq. Euclidean | $\sum_{i=1}^{m}(x_i-y_i)^2$ | $[0,\infty)$ | $M(2)^2$ |
| 3 | Manhattan | $\sum_{i=1}^{m}\lvert x_i-y_i\rvert$ | $[0,\infty)$ | $M(1)$ |
| 4 | rook | $\max_{i=1}^{m}\lvert x_i-y_i\rvert$ | $[0,\infty)$ | $M(\infty)$ |
| 5 | Minkowski | $\sqrt[p]{\sum_{i=1}^{m}\lvert x_i-y_i\rvert^p}$ | $[0,\infty)$ | $M(p)$ |
| 6 | Canberra | $\sum_{i=1}^{m}\frac{\lvert x_i-y_i\rvert}{\lvert x_i+y_i\rvert}$ | $[0,\infty)$ | |
| 7 | Heincke | $\sqrt{\sum_{i=1}^{m}(\frac{\lvert x_i-y_i\rvert}{\lvert x_i+y_i\rvert})^2}$ | $[0,\infty)$ | |
| 8 | Self-balanced | $\sum_{i=1}^{m}\frac{\lvert x_i-y_i\rvert}{\max(x_i,y_i)}$ | $[0,\infty)$ | |
| 9 | Lance-Williams | $\frac{\sum_{i=1}^{m}\lvert x_i-y_i\rvert}{\sum_{i=1}^{m}x_i+y_i}$ | $[0,\infty)$ | |
| 10 | Correlation c. | $\frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var}(X)\operatorname{var}(Y)}}$ | $[1,-1]$ | |

Usually the dissimilarity $d$ is defined using another dissimilarity $\delta : [\mathscr{U}] \times [\mathscr{U}] \to \mathbb{R}_0^+$ defined on unit descriptions as

$$d(\mathrm{X},\mathrm{Y}) = \delta([\mathrm{X}],[\mathrm{Y}])$$

The dissimilarity $d$ is:

D3. *even*: $\forall \mathrm{X},\mathrm{Y} \in \mathscr{U} : (d(\mathrm{X},\mathrm{Y}) = 0 \Rightarrow \forall \mathrm{Z} \in \mathscr{U} : d(\mathrm{X},\mathrm{Z}) = d(\mathrm{Y},\mathrm{Z}))$

D4. *definite*: $\forall \mathrm{X},\mathrm{Y} \in \mathscr{U} : (d(\mathrm{X},\mathrm{Y}) = 0 \Rightarrow \mathrm{X} = \mathrm{Y})$

D5. *metric*: $\forall \mathrm{X},\mathrm{Y},\mathrm{Z} \in \mathscr{U} : d(\mathrm{X},\mathrm{Y}) \leq d(\mathrm{X},\mathrm{Z}) + d(\mathrm{Z},\mathrm{Y})$ – triangle ineaquality

D6. *ultrametric*: $\forall \mathrm{X},\mathrm{Y},\mathrm{Z} \in \mathscr{U} : d(\mathrm{X},\mathrm{Y}) \leq \max(d(\mathrm{X},\mathrm{Z}), d(\mathrm{Z},\mathrm{Y}))$

D7. *additive*, iff the Buneman's or four-point condition holds $\forall \mathrm{X},\mathrm{Y},\mathrm{U},\mathrm{V} \in \mathscr{U}$ :
$d(\mathrm{X},\mathrm{Y}) + d(\mathrm{U},\mathrm{V}) \leq \max(d(\mathrm{X},\mathrm{U}) + d(\mathrm{Y},\mathrm{V}), d(\mathrm{X},\mathrm{V}) + d(\mathrm{Y},\mathrm{U}))$

A dissimilarity $d$ is a *distance* iff D4 and D5 hold. Since the description $[\,] : \mathbf{U} \to [\mathbf{U}]$ does not need to be injective, $d$ can be indefinite. Often a weaker form of definiteness holds:

$$\forall \mathrm{X},\mathrm{Y} \in \mathscr{U} : (d(\mathrm{X},\mathrm{Y}) = 0 \Rightarrow [\mathrm{X}] = [\mathrm{Y}])$$

A dissimilarity $d$ is selected according to the nature of the set of units descriptions $[\mathscr{U}]$ and our analytic goals. Many examples of dissimilarities can be found in [21].

***3.2.2.1 Dissimilarities on*** $\mathbb{R}^m$ In the standard case, $X \in \mathbb{R}^m$, many different dissimilarities were proposed. Some of them are presented in Table 3.1.

***3.2.2.2 (Dis)similarities on*** $\mathbb{B}^m$ Let $\mathbb{B} = \{0,1\}$. For binary vectors $X,Y \in \mathbb{B}^m$ we define $a = XY$, $b = X\overline{Y}$, $c = \overline{X}Y$, $d = \overline{X}\overline{Y}$. It holds $a+b+c+d = m$. The counters $a,b,c,d$ are used to define several resemblances – (dis)similarity measures on binary vectors. See Table 3.2.

In some cases the definition can yield an indefinite expression $\frac{0}{0}$. In such cases we can restrict the use of the measure, or define the values also for indefinite cases. For example, we extend the values of Jaccard coefficient such that $s_4(X,X) = 1$. And for Kulczynski coefficient, we preserve the relation $T = \frac{1}{s_4} - 1$ by

$$s_4 = \begin{cases} 1 & d = m \\ \frac{a}{a+b+c} & \text{otherwise} \end{cases} \qquad s_3^{-1} = T = \begin{cases} 0 & a = 0,\ d = m \\ \infty & a = 0,\ d < m \\ \frac{b+c}{a} & \text{otherwise} \end{cases}$$

We can transform a similarity $s$ from $[1,0]$ into dissimilarity $d$ on $[0,1]$ by $d = 1 - s$. For details see [8].

***3.2.2.3 Dissimilarities between sets*** Let $\mathscr{F}$ be a finite family of subsets of the finite set $U$; $A,B \in \mathscr{F}$ and let $A \oplus B = (A \setminus B) \cup (B \setminus A)$ denote the symmetric difference between $A$ and $B$. The 'standard' dissimilarity between sets is the *Hamming distance*:

$$d_H(A,B) := \mathrm{card}(A \oplus B)$$

Table 3.2: (Dis)similarities on $\mathbb{B}^m$

| n | measure | definition | range |
|---|---|---|---|
| 1 | Russel and Rao (1940) | $\frac{a}{m}$ | $[1,0]$ |
| 2 | Kendall, Sokal-Michener (1958) | $\frac{a+d}{m}$ | $[1,0]$ |
| 3 | Kulczynski (1927), $T^{-1}$ | $\frac{a}{b+c}$ | $[\infty,0]$ |
| 4 | Jaccard (1908) | $\frac{a}{a+b+c}$ | $[1,0]$ |
| 5 | Kulczynski | $\frac{1}{2}(\frac{a}{a+b}+\frac{a}{a+c})$ | $[1,0]$ |
| 6 | Sokal & Sneath (1963), $un_4$ | $\frac{1}{4}(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{d+b}+\frac{d}{d+c})$ | $[1,0]$ |
| 7 | Driver & Kroeber (1932) | $\frac{a}{\sqrt{(a+b)(a+c)}}$ | $[1,0]$ |
| 8 | Sokal & Sneath (1963), $un_5$ | $\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$ | $[1,0]$ |
| 9 | $Q_0$ | $\frac{bc}{ad}$ | $[0,\infty]$ |
| 10 | Yule (1927), $Q$ | $\frac{ad-bc}{ad+bc}$ | $[1,-1]$ |
| 11 | Pearson, $\phi$ | $\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$ | $[1,-1]$ |
| 12 | – bc – | $\frac{4bc}{m^2}$ | $[0,1]$ |
| 13 | Baroni-Urbani, Buser (1976), $S^{**}$ | $\frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}}$ | $[1,0]$ |
| 14 | Braun-Blanquet (1932) | $\frac{a}{\max(a+b,a+c)}$ | $[1,0]$ |
| 15 | Simpson (1943) | $\frac{a}{\min(a+b,a+c)}$ | $[1,0]$ |
| 16 | Michael (1920) | $\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$ | $[1,-1]$ |

Other, normalized to $[0,1]$, dissimilarities between sets are

$$d_s(A,B) = \frac{\text{card}(A \oplus B)}{\text{card}(A) + \text{card}(B)} \qquad d_u(A,B) = \frac{\text{card}(A \oplus B)}{\text{card}(A \cup B)} = 1 - \frac{\text{card}(A \cap B)}{\text{card}(A \cup B)}$$

$$d_m(A,B) = \frac{\max(\text{card}(A \setminus B), \text{card}(B \setminus A))}{\max(\text{card}(A), \text{card}(B))}$$

For all these dissimilarities $d(A,B) = 0$ if $A = B = \emptyset$.

***3.2.2.4 Equivalent resemblances*** Resemblances $r$ and $s$ are *(order) equivalent*, $r \cong s$, iff they induce the same or reverse ordering in the set of unordered pairs of units, i.e., iff

$$\forall \mathrm{X}, \mathrm{Y}, \mathrm{U}, \mathrm{V} \in \mathscr{U} : (r(\mathrm{X},\mathrm{Y}) < r(\mathrm{U},\mathrm{V}) \Leftrightarrow (s(\mathrm{X},\mathrm{Y}) < s(\mathrm{U},\mathrm{V}))$$

or

$$\forall \mathrm{X}, \mathrm{Y}, \mathrm{U}, \mathrm{V} \in \mathscr{U} : (r(\mathrm{X},\mathrm{Y}) < r(\mathrm{U},\mathrm{V}) \Leftrightarrow (s(\mathrm{X},\mathrm{Y}) > s(\mathrm{U},\mathrm{V})).$$

***3.2.2.5 Transformations*** Dissimilarities usually take values in the interval $[0,1]$ or in the interval $[0,\infty]$. They can be transformed one into the other by mappings:

$$\frac{d}{1-d} : [0,1] \to [0,\infty] \quad \text{and} \quad \frac{d}{1+d} : [0,\infty] \to [0,1],$$

or in the case $d_{max} < \infty$ by

$$\frac{d}{d_{max}} : [0, d_{max}] \to [0,1].$$

To transform a distance $d$ into another distance we often use the mappings:

$$\log(1+d), \quad \min(1,d) \quad \text{and} \quad d^r,\ 0 < r < 1.$$

Not all resemblances are dissimilarities. For example, the correlation coefficient has the interval $[1,-1]$ as its range. We can transform it to the interval $[0,1]$ by mappings:

$$\frac{1}{2}(1-d), \quad \sqrt{1-d^2}, \quad 1-|d|, \text{ etc.}$$

When applying these transformations to a measure $d$ we wish that the nice properties of $d$ were preserved. In this respect the following theorems should be mentioned.

**Proposition 3.1** *Let $d$ be a dissimilarity on $\mathscr{U}$ and let a mapping $f\colon d(\mathscr{U} \times \mathscr{U}) \to \mathbb{R}_0^+$ has the property $f(0) = 0$, then $d'(\mathrm{X},\mathrm{Y}) = f(d(\mathrm{X},\mathrm{Y}))$ is also a dissimilarity.*

**Proposition 3.2** *Let $d$ be a distance on $\mathscr{U}$ and let the mapping $f\colon d(\mathscr{U} \times \mathscr{U}) \to \mathbb{R}$ has the properties:*

*(a)* $f(x) = 0 \Leftrightarrow x = 0$,
*(b)* $x < y \Rightarrow f(x) < f(y)$,
*(c)* $f(x+y) \le f(x) + f(y)$,

*then $d'(\mathrm{X},\mathrm{Y}) = f(d(\mathrm{X},\mathrm{Y}))$ is also a distance and $d' \cong d$.*

All concave functions have also the sub-additivity property (c). The following concave functions satisfy the last theorem:

(a) $f(x) = \alpha x,\ \alpha > 0$, (b) $f(x) = \log(1+x),\ x \geq 0$,
(c) $f(x) = \frac{x}{1+x},\ x \geq 0$, (d) $f(x) = \min(1,x)$,
(e) $f(x) = x^{\alpha},\ 0 < \alpha \leq 1$, (f) $f(x) = \arcsin x,\ 0 \leq x \leq 1$.

**Proposition 3.3** *Let $d\colon \mathscr{U} \times \mathscr{U} \to \mathbb{R}$ has the property* D*i, $i = 1,\ldots,7$, then $f(d)$, $f \in$* (a)-(f) *also has this property.*

**Proposition 3.4** *(Joly and Le Calvé, 1986 [38]) For each (nonnegative) dissimilarity measure $d$ there is a unique nonnegative real number $p$, called metric index, such that $d^{\alpha}$ is a metric for all $\alpha \leq p$, and $d^{\alpha}$ is not a metric for all $\alpha > p$.*

Therefore, if a dissimilarity $d$ is not metric, it can be transformed into it using the power transformation.

***3.2.2.6 Problems with dissimilarities*** What to do in the case of *mixed units* with variables measured on different types of scales? Two approaches are usually used:

- conversion to a common type of measurement scale (see Anderberg [2]);
- compute selected dissimilarities on homogeneous parts and combine them. See for example Gower's dissimilarity [32].

In both cases we have to consider the *fairness* of dissimilarity – all variables contribute equally. A partial solution to this problem is to use the normalized variables. We can also consider the dependencies among variables, such as in the Mahalanobis distance [56].

### 3.2.3 Cluster-error function / examples

Now we can define several types of cluster-error functions:

S. $$p(C) = \sum_{\mathrm{X},\mathrm{Y} \in C, \mathrm{X} < \mathrm{Y}} w(\mathrm{X}) \cdot w(\mathrm{Y}) \cdot d(\mathrm{X},\mathrm{Y})$$

$\overline{\mathrm{S}}$. $$p(C) = \frac{1}{w(C)} \sum_{\mathrm{X},\mathrm{Y} \in C, \mathrm{X} < \mathrm{Y}} w(\mathrm{X}) \cdot w(\mathrm{Y}) \cdot d(\mathrm{X},\mathrm{Y})$$

where $w : \mathscr{U} \to \mathbb{R}^{+}$ is a *weight* of units, which is extended to clusters by:

$$w(\{\mathrm{X}\}) = w(\mathrm{X}), \qquad \mathrm{X} \in \mathscr{U}$$

$$w(C_1 \cup C_2) = w(C_1) + w(C_2), \qquad C_1 \cap C_2 = \emptyset$$

Often $w(\mathrm{X}) = 1$ holds for each $\mathrm{X} \in \mathscr{U}$. Then $w(C) = \mathrm{card}(C)$.

M. $$p(C) = \max_{\mathrm{X},\mathrm{Y} \in C} d(\mathrm{X},\mathrm{Y}) = \mathrm{diam}(C)$$ – diameter

T. $$p(C) = \min_{T \text{ is a spanning tree over } C} \sum_{(\mathrm{X}:\mathrm{Y}) \in T} d(\mathrm{X},\mathrm{Y})$$

We shall use the labels in front of the forms of (cluster-) criterion functions to denote *types* of criterion functions. For example:

SM. $$P(\mathbf{C}) = \sum_{C \in \mathbf{C}} \max_{\mathrm{X},\mathrm{Y} \in C} d(\mathrm{X},\mathrm{Y})$$

It is easy to prove:

**Proposition 3.5** *Let* $P \in \{\mathrm{SS}, \mathrm{S\overline{S}}, \mathrm{SM}, \mathrm{MS}, \mathrm{M\overline{S}}, \mathrm{MM}\}$ *then there exists an* $\alpha_k^P(\mathbf{U}) > 0$ *such that for each* $\mathbf{C} \in P_k(\mathbf{U})$ *holds:*

$$P(\mathbf{C}) \geq \alpha_k^P(\mathbf{U}) \cdot \max_{C \in \mathbf{C}} \max_{\mathrm{X},\mathrm{Y} \in C} d(\mathrm{X},\mathrm{Y}).$$

Note that this inequality can be writen also as $P(\mathbf{C}) \geq \alpha_k^P(\mathbf{U}) \cdot \mathrm{MM}(\mathbf{C})$.

The criterion function $P(\mathbf{C})$, based on a dissimilarity $d$, is *sensitive* iff for each feasible clustering $\mathbf{C}$ it holds

$$P(\mathbf{C}) = 0 \Longleftrightarrow \forall C \in \mathbf{C}\ \forall \mathrm{X},\mathrm{Y} \in C : d(\mathrm{X},\mathrm{Y}) = 0$$

and is $\alpha$*-sensitive* iff there exists an $\alpha_k^P(\mathbf{U}) > 0$ such that for each $\mathbf{C} \in P_k(\mathbf{U})$ :

$$P(\mathbf{C}) \geq \alpha_k^P(\mathbf{U}) \cdot \mathrm{MM}(\mathbf{C})$$

**Proposition 3.6** *Every* $\alpha$*-sensitive criterion function is also sensitive.*

Proposition 3.5 can be reexpressed as:

**Proposition 3.7** *The criterion functions* $\mathrm{SS}, \mathrm{S\overline{S}}, \mathrm{SM}, \mathrm{MS}, \mathrm{M\overline{S}}, \mathrm{MM}$ *are* $\alpha$*-sensitive.*

Another form of cluster-error function, which is frequently used in practice, is based on the notion of a leader or representative of the cluster $C$:

R. $p(C) = \min_{\mathrm{L} \in \mathbf{F}} \sum_{\mathrm{X} \in C} w(\mathrm{X}) \cdot d(\mathrm{X},\mathrm{L})$

where $\mathbf{F} \subseteq \mathscr{F}$ is the set of *representatives*. The element $\overline{C} \in \mathbf{F}$, which minimizes the right side expression, is called the *representative* of the cluster $C$. It is not always uniquely determined.

**Proposition 3.8** *Let* $p(C)$ *be of type* R *then*

*a)* $p(C) + w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C \cup \{\mathrm{X}\}}) \leq p(C \cup \{\mathrm{X}\}), \qquad \mathrm{X} \notin C$

*b)* $p(C \setminus \{\mathrm{X}\}) + w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C}) \leq p(C), \qquad \mathrm{X} \in C$

***3.2.3.1 The generalized Ward's criterion function.*** To obtain the *generalized Ward's clustering problem* we, relying on the equality

$$p(C) = \sum_{X \in C} d_2^2(X, \overline{C}) = \frac{1}{2\,\mathrm{card}(C)} \sum_{X,Y \in C} d_2^2(X,Y)$$

replace the expression for $p(C)$ with

$$p(C) = \frac{1}{2w(C)} \sum_{X,Y \in C} w(X) \cdot w(Y) \cdot d(X,Y) = \overline{\mathrm{S}}(C)$$

Note that $d$ can be **any** dissimilarity on $\mathscr{U}$.

From the definition we can easily derive the following equality: If $C_u \cap C_v = \emptyset$ then

$$w(C_u \cup C_v) \cdot p(C_u \cup C_v) = w(C_u) \cdot p(C_u) + w(C_v) \cdot p(C_v) + \sum_{X \in C_u, Y \in C_v} w(X) \cdot w(Y) \cdot d(X,Y)$$

In [5] it is also shown how to replace $\overline{C}$ by a generalized, possibly imaginary (with descriptions not neccessary in the same set as $\mathscr{U}$), central element in the way to preserve the properties characteristic for Ward's clustering problem.

Let $\mathscr{U}^*$ denote the space of units extended with generalized centers. The *generalized center* of cluster $C$ is called an (abstract) element $\overline{C}$ for which the dissimilarity between it and any $U \in \mathscr{U}^*$ is determined by

$$d(U,\overline{C}) = d(\overline{C},U) = \frac{1}{w(C)}\left(\sum_{X\in C} w(X)\cdot d(X,U) - p(C)\right)$$

When for all units $w(X) = 1$, the right part of the definition can be read: the average dissimilarity between unit/center $U$ and cluster $C$ diminished by the average radius of cluster $C$.

Suggestion: For each dissimilarity find its metric index $p$ and in the generalized Huygens theorem use $d$ if $p \geq 1$, otherwise (if $p < 1$) use $d^p$.

For the generalized Ward's criterion function the *generalized Huygens theorem* holds:

**Proposition 3.9**

$$I_T = I_W + I_B$$

*where*

$$I_T = p(\mathbf{U}) = \frac{1}{2w(\mathbf{U})}\sum_{\mathrm{X,Y}\in\mathbf{U}} w(\mathrm{X})\cdot w(\mathrm{Y})\cdot d(\mathrm{X},\mathrm{Y})$$

$$I_W = \sum_{C\in\mathbf{C}} p(C) \qquad \textit{and} \qquad I_B = \sum_{C\in\mathbf{C}} w(C)\cdot d(\overline{C},\overline{\mathbf{U}})$$

For a given set of units $\mathbf{U}$ the value of their "total inertia" $I_T$ is fixed. Therefore minimazing the "standard" criterion function (within inertia) $I_W$ we are also maximazing the function (between inertia) $I_B$ – the traditional definition of clustering problem.

***3.2.3.2 Other criterion functions.*** Several other types of criterion functions were proposed in the literature. A very important class among them are the "statistical" criterion functions based on the assumption that the units are sampled from a mixture of multivariate normal distributions [45] .

Not all clustering problems can be expressed by a simple criterion function. In some applications a *general* criterion function of the form

$$P(\mathbf{C}) = \bigoplus_{(C_1,C_2)\in\mathbf{C}\times\mathbf{C}} q(C_1,C_2), \quad q(C_1,C_2) \geq 0$$

is needed. We use it in the optimization approach to blockmodeling [24].

In some problems several criterion functions can be defined $(\Phi, P_1, P_2, \ldots, P_s)$ and the clustering problem is formulated as *multicriteria clustering* problem [28].

Note that for a criterion function of type SS we have a similar situation as in the generalized Huygens theorem:

**Proposition 3.10**

$$P_T = P_W + P_B$$

*where, denoting* $p(C,D) = \sum_{\mathrm{X}\in C,\mathrm{Y}\in D} d(\mathrm{X},\mathrm{Y})$

$$P_T = p(\mathbf{U},\mathbf{U}), \quad P_W = \sum_{C\in\mathbf{C}} p(C,C) = SS(\mathbf{C}), \quad \textit{and} \quad P_B = \sum_{\substack{C,D\in\mathbf{C}\\ C\neq D}} p(C,D)$$

***3.2.3.3 Partitioning of a generation of pupils into a given number of classes.*** As a kind of nontraditional clustering problem in which the clusters are not characterized as "groups of similar units" let us consider the problem of partitioning of a generation of pupils into a given number of classes so that the classes will consist of (almost) the same number of pupils and that they will have a structure as similar as possible. An appropriate criterion function is

$$P(\mathbf{C}) = \max_{\substack{\{C_1,C_2\}\in\mathbf{C}\times\mathbf{C} \\ \mathrm{card}(C_1)\geq\mathrm{card}(C_2)}} \min_{\substack{f:C_1\to C_2 \\ f \text{ is surjective}}} \max_{\mathrm{X}\in C_1} d(\mathrm{X}, f(\mathrm{X}))$$

where $d(\mathrm{X},\mathrm{Y})$ is a measure of dissimilarity between pupils X and Y.

### 3.2.4 Complexity of the clustering problem

Because the set of feasible clusterings $\Phi$ is finite the clustering problem $(\Phi, P)$ can be solved by the brute force approach inspecting all feasible clusterings. Unfortunately, the number of feasible clusterings grows very quickly with $n$. For example

$$\mathrm{card}(P_k) = S(n,k) = \frac{1}{k!}\sum_{i=0}^{k-1}(-1)^i\binom{k}{i}(k-i)^n, \quad 0 < k \leq n$$

where $S(n,k)$ is a Stirling number of the second kind. For this reason the brute force algorithm is only of theoretical interest.

We shall assume that the reader is familiar with the basic notions of the theory of complexity of algorithms [30]. Although there are some types of clustering problems of polynomial complexity, for example $(P_2, \mathrm{MM})$ and $(P_k, \mathrm{ST})$, it seems that they are mainly NP-hard. Brucker [17] showed that ( $\propto$ denotes the polynomial reducibility of problems [30]) :

**Theorem 3.11** *Let the criterion function*

$$P(\mathbf{C}) = \bigoplus_{C\in\mathbf{C}} p(C)$$

*be $\alpha$-sensitive, then for each problem $(P_k(\mathbf{U}), P)$ there exists a problem $(P_{k+1}(\mathbf{U}'), P)$, such that $(P_k(\mathbf{U}), P) \propto (P_{k+1}(\mathbf{U}'), P)$.*

**Theorem 3.12** *Let the criterion function P be sensitive then 3-COLOR $\propto (P_3, P)$.*

Note that, by Theorem 3.11, $(P_k, \mathrm{MM})$, $k > 3$ are also NP-hard, etc.

The complexity results for some types of clustering criterion functions are summarized in Table 3.3.

From these results it follows (it is believed) that no efficient (polynomial) exact algorithm exists for solving the clustering problem. Therefore the procedures should be used which give "good" results, but not necessarily the best, in a reasonable time. In the following section we present some standard approaches for solving clustering problem.

Table 3.3: Complexity of clustering problems

| Polynomial | NP-hard | note |
|---|---|---|
| $(P_2, \mathrm{MM})$ | $(P_3, \mathrm{MM})$ | Theorem 3.12 |
| | $(P_3, \mathrm{SM})$ | Theorem 3.12 |
| | $(P_2, \mathrm{SS})$ | MAX-CUT $\propto (P_2, \mathrm{SS})$ |
| | $(P_2, \mathrm{S\bar{S}})$ | $(P_2, \mathrm{SS}) \propto (P_2, \mathrm{S\bar{S}})$ |
| | $(P_2, \mathrm{MS})$ | PARTITION $\propto (P_2, \mathrm{MS})$ |
| $(\mathbb{R}_2^m, \mathrm{S\bar{S}})$ | | |
| $(\mathbb{R}_k^1, \mathrm{S\bar{S}})$ | | |
| $(\mathbb{R}_k^1, \mathrm{SM})$ | | |
| $(\mathbb{R}_k^1, \mathrm{MM})$ | | |

## 3.3 Approaches to Clustering

### 3.3.1 Local optimization

Often for a given optimization problem $(\Phi, P, \min)$ there exist rules which relate to each element of the set $\Phi$ some elements of $\Phi$. We call them *local transformations*. The elements which can be obtained from a given element are called *neighbor*s – local transformations determine the *neighborhood relation* $S \subseteq \Phi \times \Phi$ in the set $\Phi$. The *neighborhood* of element $\mathrm{X} \in \Phi$ is called the set $S(\mathrm{X}) = \{\mathrm{Y} : \mathrm{X}\, S\, \mathrm{Y}\}$ . The element $\mathrm{X} \in \Phi$ is a *local minimum* for the *neighborhood structure* $(\Phi, S)$ iff

$$\forall \mathrm{Y} \in S(\mathrm{X}) : P(\mathrm{X}) \leq P(\mathrm{Y})$$

In the following we shall assume that $S$ is reflexive, $\forall \mathrm{X} \in \Phi : \mathrm{X}\, S\, \mathrm{X}$.

The relation $S$ is a basis of the *local optimization procedure*

select $\mathrm{X}_0$; $\mathrm{X} := \mathrm{X}_0$;
**while** $\exists \mathrm{Y} \in S(\mathrm{X}) : P(\mathrm{Y}) < P(\mathrm{X})$ **do** $\mathrm{X} := \mathrm{Y}$;

which starting in an initial element $\mathrm{X}_0 \in \Phi$ repeats moving to an element, in its neighborhood determined by local transformation, which has better value of the criterion function until no such element exists. To get good solution we repeat the procedure many times with random initial element $\mathrm{X}_0$ and keep the best solution found.

***3.3.1.1 Clustering neigborhoods.*** Usually the neighborhood relation in local optimization clustering procedures over $P_k(\mathbf{U})$ is determined by the following two transformations:

- *transition*: clustering $\mathbf{C}'$ is obtained from $\mathbf{C}$ by moving a unit $\mathrm{X}_s \in C_u$ from one cluster, $C_u$, to another, $C_v$,

$$\mathbf{C}' = (\mathbf{C} \setminus \{C_u, C_v\}) \cup \{C_u \setminus \{\mathrm{X}_s\}, C_v \cup \{\mathrm{X}_s\}\}$$

- *transposition*: clustering $\mathbf{C}'$ is obtained from $\mathbf{C}$ by interchanging two units, $\mathrm{X}_p \in C_u$ and $\mathrm{X}_q \in C_v$, from different clusters

$$\mathbf{C}' = (\mathbf{C} \setminus \{C_u, C_v\}) \cup \{(C_u \setminus \{\mathrm{X}_p\}) \cup \{\mathrm{X}_q\}, (C_v \setminus \{\mathrm{X}_q\}) \cup \{\mathrm{X}_p\}\}$$

The transpositions preserve the number of units in clusters. The local optimization based on transitions and/or transpositions is usually called the *relocation* method.

Using Proposition 3.8 we can prove the following important property of the minimal solutions of the clustering problem $(P_k, \mathrm{SR}, \min)$:

**Proposition 3.13** *In the locally with respect to transitions minimal clustering for the problem* $(P_k, \mathrm{SR}, \min)$

$$\mathrm{SR}. \qquad P(\mathbf{C}) = \sum_{C \in \mathbf{C}} \sum_{\mathrm{X} \in C} w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C})$$

*each unit is assigned to the nearest representative: Let* $\mathbf{C}^\bullet$ *be locally with respect to transitions minimal clustering then it holds:*

$$\forall C_u \in \mathbf{C}^\bullet \; \forall \mathrm{X} \in C_u \; \forall C_v \in \mathbf{C}^\bullet \setminus \{C_u\} : d(\mathrm{X}, \overline{C}_u) \leq d(\mathrm{X}, \overline{C}_v)$$

Two basic implementation approaches are usually used: *stored data* approach and *stored dissimilarity matrix* approach.

If the constraints are not too stringent, the relocation method can be applied directly on $\Phi$; otherwise, we can transform using *penalty function method* the problem to an equivalent nonconstrained problem $(P_k, Q, \min)$ with $Q(\mathbf{C}) = P(\mathbf{C}) + \alpha K(\mathbf{C})$ where $\alpha > 0$ is a large constant and $K(\mathbf{C}) = 0$, for $\mathbf{C} \in \Phi$, and $K(\mathbf{C}) > 0$ otherwise.

There exist several improvements of the basic relocation algorithm: simulated annealing, tabu search, etc. [1].

***3.3.1.2 Testing*** $P(\mathbf{C}') < P(\mathbf{C})$ is equivalent to $P(\mathbf{C}) - P(\mathbf{C}') > 0$. For the S criterion function

$$\Delta P(\mathbf{C}, \mathbf{C}') = P(\mathbf{C}) - P(\mathbf{C}') = p(C_u) + p(C_v) - p(C'_u) - p(C'_v)$$

Additional simplifications can be done considering relations between $C_u$ and $C'_u$, and between $C_v$ and $C'_v$.

Let us illustrate this on the generalized Ward's method. For this purpose it is useful to introduce the quantity

$$a(C_u, C_v) = \sum_{\mathrm{X} \in C_u, \mathrm{Y} \in C_v} w(\mathrm{X}) \cdot w(\mathrm{Y}) \cdot d(\mathrm{X}, \mathrm{Y})$$

Using the quantity $a(C_u, C_v)$ we can express $p(C)$ in the form $p(C) = \frac{a(C,C)}{2w(C)}$ and the equality mentioned in the introduction of the generalized Ward clustering problem: if $C_u \cap C_v = \emptyset$ then

$$w(C_u \cup C_v) \cdot p(C_u \cup C_v) = w(C_u) \cdot p(C_u) + w(C_v) \cdot p(C_v) + a(C_u, C_v)$$

Let us analyze the transition of a unit $\mathrm{X}_s$ from cluster $C_u$ to cluster $C_v$. We have $C'_u = C_u \setminus \{\mathrm{X}_s\}$ , $C'_v = C_v \cup \{X_s\}$ ,

$$w(C_u) \cdot p(C_u) = w(C'_u) \cdot p(C'_u) + a(\mathrm{X}_s, C'_u) = (w(C_u) - w(\mathrm{X}_s)) \cdot p(C'_u) + a(\mathrm{X}_s, C'_u)$$

and

$$w(C'_v) \cdot p(C'_v) = w(C_v) \cdot p(C_v) + a(\mathrm{X}_s, C_v)$$

From $d(\mathrm{X}_s,\mathrm{X}_s)=0$ it follows $a(\mathrm{X}_s,C_u)=a(\mathrm{X}_s,C'_u)$. Therefore

$$p(C'_u)=\frac{w(C_u)\cdot p(C_u)-a(\mathrm{X}_s,C_u)}{w(C_u)-w(\mathrm{X}_s)} \qquad p(C'_v)=\frac{w(C_v)\cdot p(C_v)+a(\mathrm{X}_s,C_v)}{w(C_v)+w(\mathrm{X}_s)}$$

and finally

$$\Delta P(\mathbf{C},\mathbf{C}')=p(C_u)+p(C_v)-p(C'_u)-p(C'_v)=$$
$$=\frac{w(\mathrm{X}_s)\cdot p(C_v)-a(\mathrm{X}_s,C_v)}{w(C_v)+w(\mathrm{X}_s)}-\frac{w(\mathrm{X}_s)\cdot p(C_u)-a(\mathrm{X}_s,C_u)}{w(C_u)-w(\mathrm{X}_s)}$$

In the case when $d$ is the squared Euclidean distance it is possible to derive also expression for corrections of centers [53].

### 3.3.2 Dynamic programming

Suppose that $\mathrm{Min}(\Phi_k,P)\neq\emptyset$, $k=1,2,\ldots$. Denoting $P^*(\mathbf{U},k)=P(\mathbf{C}^*_k(\mathbf{U}))$ we can derive the generalized *Jensen equality* [10]:

$$P^*(\mathbf{U},k)=\begin{cases} p(\mathbf{U}) & \{\mathbf{U}\}\in\Phi_1 \\ \min\limits_{\substack{\emptyset\subset C\subset\mathbf{U} \\ \exists\mathbf{C}\in\Phi_{k-1}(\mathbf{U}\setminus C):\mathbf{C}\cup\{C\}\in\Phi_k(\mathbf{U})}} (P^*(\mathbf{U}\setminus C,k-1)\oplus p(C)) & k>1 \end{cases}$$

This is a *dynamic programming* (Bellman) equation which, for some special constrained problems, that keep the size of $\Phi_k$ small, allows us to solve the clustering problem by the adapted Fisher's algorithm [10].

### 3.3.3 Hierarchical methods

The set of feasible clusterings $\Phi$ determines the *feasibility predicate* $\underline{\Phi}(\mathbf{C})\equiv\mathbf{C}\in\Phi$ defined on $\mathscr{P}(\mathscr{P}(\mathbf{U})\setminus\{\emptyset\})$; and conversely $\Phi\equiv\{\mathbf{C}\in\mathscr{P}(\mathscr{P}(\mathbf{U})\setminus\{\emptyset\}):\underline{\Phi}(\mathbf{C})\}$.

In the set $\Phi$ the relation of *clustering inclusion* $\sqsubseteq$ can be introduced by

$$\mathbf{C}_1\sqsubseteq\mathbf{C}_2\equiv\forall C_1\in\mathbf{C}_1,C_2\in\mathbf{C}_2:C_1\cap C_2\in\{\emptyset,C_1\}$$

we say also that the clustering $\mathbf{C}_1$ is a *refinement* of the clustering $\mathbf{C}_2$.

It is well known that $(P(\mathbf{U}),\sqsubseteq)$ is a partially ordered set (even more, semimodular lattice). Because any subset of partially ordered set is also partially ordered, we have: Let $\Phi\subseteq P(\mathbf{U})$ then $(\Phi,\sqsubseteq)$ is a partially ordered set.

The clustering inclusion determines two related relations (on $\Phi$):

$\mathbf{C}_1\sqsubset\mathbf{C}_2\equiv\mathbf{C}_1\sqsubseteq\mathbf{C}_2\wedge\mathbf{C}_1\neq\mathbf{C}_2$ – strict inclusion, and
$\mathbf{C}_1\dot{\sqsubset}\mathbf{C}_2\equiv\mathbf{C}_1\sqsubset\mathbf{C}_2\wedge\neg\exists\mathbf{C}\in\Phi:(\mathbf{C}_1\sqsubset\mathbf{C}\wedge\mathbf{C}\sqsubset\mathbf{C}_2)$ – predecessor.

Part of the following text we presented already in Section 9.3 of our book [11]. We include it also here to make the text self contained. We shall assume that the set of feasible clusterings $\Phi\subseteq P(\mathbf{U})$ satisfies the following conditions:

**F1.** $\mathbf{O}\equiv\{\{\mathrm{X}\}:\mathrm{X}\in\mathbf{U}\}\in\Phi$

**F2.** The feasibility predicate $\underline{\Phi}$ is *local* – it has the form $\underline{\Phi}(\mathbf{C})=\bigwedge_{C\in\mathbf{C}}\varphi(C)$ where $\varphi(C)$ is a predicate defined on $\mathscr{P}(\mathbf{U})\setminus\{\emptyset\}$ (clusters). The intuitive meaning of $\varphi(C)$ is: $\varphi(C)\equiv$ the cluster $C$ is 'good'. Therefore, the locality condition can be read: a 'good' clustering $\mathbf{C}\in\Phi$ consists of 'good' clusters.

**F3.** The predicate $\Phi$ has the property of *binary heredity* with respect to the *fusibility* predicate $\psi(C_1, C_2)$, i.e.,

$$C_1 \cap C_2 = \emptyset \wedge \varphi(C_1) \wedge \varphi(C_2) \wedge \psi(C_1, C_2) \Rightarrow \varphi(C_1 \cup C_2)$$

This condition means: in a 'good' clustering, a fusion of two 'fusible' clusters produces a 'good' clustering.

**F4.** The predicate $\psi$ is *compatible* with clustering inclusion $\sqsubseteq$, i.e.,

$$\forall \mathbf{C}_1, \mathbf{C}_2 \in \Phi : (\mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \mathbf{C}_1 \setminus \mathbf{C}_2 = \{C_1, C_2\} \Rightarrow \psi(C_1, C_2) \vee \psi(C_2, C_1))$$

**F5.** The *interpolation* property holds in $\Phi$, i.e., $\forall \mathbf{C}_1, \mathbf{C}_2 \in \Phi$ :

$$(\mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \mathrm{card}(\mathbf{C}_1) > \mathrm{card}(\mathbf{C}_2) + 1 \Rightarrow \exists \mathbf{C} \in \Phi : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2))$$

These conditions provide a framework in which the hierarchical methods can be applied also for constrained clustering problems $\Phi_k(\mathbf{U}) \subset P_k(\mathbf{U})$. In the ordinary problem both predicates $\varphi(C)$ and $\psi(C_p, C_q)$ are always true – all conditions F1-F5 are satisfied.

***3.3.3.1 Greedy approximation.*** We shall call a *dissimilarity between clusters* a function $D : (C_1, C_2) \to \mathbb{R}_0^+$ which is symmetric, i.e., $D(C_1, C_2) = D(C_2, C_1)$.

Let $(\mathbb{R}_0^+, \oplus, e, \leq)$ be an ordered abelian monoid. Then the criterion function $P(\mathbf{C}) = \bigoplus_{C \in \mathbf{C}} p(C)$, $\forall X \in \mathbf{U} : p(\{X\}) = 0$ is *compatible* with dissimilarity $D$ over $\Phi$ iff for all $C \subseteq \mathbf{U}$ holds:

$$\varphi(C) \wedge \mathrm{card}(C) > 1 \Rightarrow p(C) = \min_{(C_1, C_2) : C_2 = C \setminus C_1 \wedge \psi(C_1, C_2)} (p(C_1) \oplus p(C_2) \oplus D(C_1, C_2))$$

**Proposition 3.14** *An* S *criterion function is compatible with dissimilarity $D$ defined by*

$$D(C_p, C_q) = p(C_p \cup C_q) - p(C_p) - p(C_q)$$

*In this case, let* $\mathbf{C}' = \mathbf{C} \setminus \{C_p, C_q\} \cup \{C_p \cup C_q\}$, $C_p, C_q \in \mathbf{C}$, *then*

$$P(\mathbf{C}') = P(\mathbf{C}) + D(C_p, C_q)$$

**Proposition 3.15** *Let $P$ be compatible with $D$ over $\Phi$, $\oplus$ distributes over* min, *and F1 – F5 hold, then*

$$P(\mathbf{C}_k^*) = \min_{\mathbf{C} \in \Phi_k} P(\mathbf{C}) = \min_{\substack{C_1, C_2 \in \mathbf{C} \in \Phi_{k+1} \\ \psi(C_1, C_2)}} (P(\mathbf{C}) \oplus D(C_1, C_2))$$

The equality from Proposition 3.15 can also be written in the form

$$P(\mathbf{C}_k^*) = \min_{\mathbf{C} \in \Phi_{k+1}} (P(\mathbf{C}) \oplus \min_{\substack{C_1, C_2 \in \mathbf{C} \\ \psi(C_1, C_2)}} D(C_1, C_2))$$

from where we can see the following 'greedy' approximation:

$$P(\mathbf{C}_k^*) \approx P(\mathbf{C}_{k+1}^*) \oplus \min_{\substack{C_1, C_2 \in \mathbf{C}_{k+1}^* \\ \psi(C_1, C_2)}} D(C_1, C_2)$$

which is the basis for the agglomerative (binary) procedure for solving the clustering problem.

#### 3.3.3.2 Agglomerative methods

1. $k := n$; $\mathbf{C}(k) := \{\{\mathrm{X}\} : \mathrm{X} \in \mathbf{U}\}$;
2. **while** $\exists C_i, C_j \in \mathbf{C}(k)\colon (i \neq j \wedge \psi(C_i, C_j))$ **repeat**
2.1. $(C_p, C_q) := \operatorname{argmin}\{D(C_i, C_j)\colon i \neq j \wedge \psi(C_i, C_j)\}$;
2.2. $C := C_p \cup C_q$; $k := k - 1$;
2.3. $\mathbf{C}(k) := \mathbf{C}(k+1) \setminus \{C_p, C_q\} \cup \{C\}$;
2.4. determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$
3. $m := k$

Note that, because it is based on an approximation, this procedure is not an exact procedure for solving the clustering problem.

For another, *probabilistic* view on agglomerative methods see [39].

*Divisive* methods work in the reverse direction. The problem here is how to efficiently find a good split $(C_p, C_q)$ of the cluster $C$.

In derivations of between cluster dissimilarity $D(C_u, C_v)$ for different "classical" agglomerative methods we shall use the generalized Ward's cluster error function $p(C)$ and generalized centers [5].

Minimal: $D^m(C_u, C_v) = \min_{X \in C_u, Y \in C_v} d(X, Y)$

Maximal: $D^M(C_u, C_v) = \max_{X \in C_u, Y \in C_v} d(X, Y)$

Average: $D^a(C_u, C_v) = \dfrac{1}{w(C_u)w(C_v)} \sum_{X \in C_u, Y \in C_v} w(X) \cdot w(Y) \cdot d(X, Y)$

Gower-Bock: $D^G(C_u, C_v) = d(\overline{C}_u, \overline{C}_v) = D^a(C_u, C_v) - \dfrac{p(C_u)}{w(C_u)} - \dfrac{p(C_v)}{w(C_v)}$

Ward: $D^W(C_u, C_v) = \dfrac{w(C_u)w(C_v)}{w(C_u \cup C_v)} D^G(\overline{C}_u, \overline{C}_v)$

Inertia: $D^I(C_u, C_v) = p(C_u \cup C_v)$

Variance: $D^V(C_u, C_v) = \operatorname{var}(C_u \cup C_v) = \dfrac{p(C_u \cup C_v)}{w(C_u \cup C_v)}$

Weighted increase of variance:

$$D^v(C_u, C_v) = \operatorname{var}(C_u \cup C_v) - \frac{w(C_u) \cdot \operatorname{var}(C_u) + w(C_v) \cdot \operatorname{var}(C_v)}{w(C_u \cup C_v)} = \frac{D^W(C_u, C_v)}{w(C_u \cup C_v)}$$

For all of them *Lance-Williams-Jambu formula* holds:

$$\begin{aligned} D(C_p \cup C_q, C_s) \quad = \quad & \alpha_1 D(C_p, C_s) + \alpha_2 D(C_q, C_s) + \beta D(C_p, C_q) + \\ & + \gamma |D(C_p, C_s) - D(C_q, C_s)| + \delta_1 v(C_p) + \delta_2 v(C_q) + \delta_3 v(C_s) \end{aligned}$$

The coefficients $\alpha_1$, $\alpha_2$, $\beta$, $\gamma$ and $\delta$ are given in Table 3.4.

**3.3.3.3 Hierarchies.** The agglomerative clustering procedure produces a series of feasible clusterings $\mathbf{C}(n), \mathbf{C}(n-1), \ldots, \mathbf{C}(m)$ with $\mathbf{C}(m) \in \operatorname{Max} \Phi$ (maximal elements for $\sqsubseteq$). Their union $\mathscr{T} = \bigcup_{k=m}^{n} \mathbf{C}(k)$ is called a *hierarchy* and has the property

$$\forall C_p, C_q \in \mathscr{T} : C_p \cap C_q \in \{\emptyset, C_p, C_q\}$$

The set inclusion $\subseteq$ is a *tree* or *hierarchical* order on $\mathscr{T}$. The hierarchy $\mathscr{T}$ is *complete* iff $\mathbf{U} \in \mathscr{T}$.

Table 3.4: Lance-Williams-Jambu coefficients

| method | $\alpha_1$ | $\alpha_2$ | $\beta$ | $\gamma$ | $\delta_t$ | $v(C_t)$ |
|---|---|---|---|---|---|---|
| minimum | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $-\frac{1}{2}$ | 0 | – |
| maximum | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 | – |
| average | $\frac{w_p}{w_{pq}}$ | $\frac{w_q}{w_{pq}}$ | 0 | 0 | 0 | – |
| Gower-Bock | $\frac{w_p}{w_{pq}}$ | $\frac{w_q}{w_{pq}}$ | $-\frac{w_p w_q}{w_{pq}^2}$ | 0 | 0 | – |
| Ward | $\frac{w_{ps}}{w_{pqs}}$ | $\frac{w_{qs}}{w_{pqs}}$ | $-\frac{w_s}{w_{pqs}}$ | 0 | 0 | – |
| inertia | $\frac{w_{ps}}{w_{pqs}}$ | $\frac{w_{qs}}{w_{pqs}}$ | $\frac{w_{pq}}{w_{pqs}}$ | 0 | $-\frac{w_t}{w_{pqs}}$ | $p(C_t)$ |
| variance | $\frac{w_{ps}^2}{w_{pqs}^2}$ | $\frac{w_{qs}^2}{w_{pqs}^2}$ | $\frac{w_{pq}^2}{w_{pqs}^2}$ | 0 | $-\frac{w_t}{w_{pqs}^2}$ | $p(C_t)$ |
| w.i. variance | $\frac{w_{ps}^2}{w_{pqs}^2}$ | $\frac{w_{qs}^2}{w_{pqs}^2}$ | $-\frac{w_s w_{pq}}{w_{pqs}^2}$ | 0 | 0 | – |

$$w_p = w(C_p),\ w_{pq} = w(C_p \cup C_q),\ w_{pqs} = w(C_p \cup C_q \cup C_s)$$

For $W \subseteq \mathbf{U}$ we define the *smallest cluster* $C_{\mathscr{T}}(W)$ from $\mathscr{T}$ containing $W$ as:

c1. $W \subseteq C_{\mathscr{T}}(W)$
c2. $\forall C \in \mathscr{T} : (W \subseteq C \Rightarrow C_{\mathscr{T}}(W) \subseteq C)$

$C_{\mathscr{T}}$ is a *closure* on $\mathscr{T}$ with a special property

$$\mathrm{Z} \notin C_{\mathscr{T}}(\{\mathrm{X},\mathrm{Y}\}) \Rightarrow C_{\mathscr{T}}(\{\mathrm{X},\mathrm{Y}\}) \subset C_{\mathscr{T}}(\{\mathrm{X},\mathrm{Y},\mathrm{Z}\}) = C_{\mathscr{T}}(\{\mathrm{X},\mathrm{Z}\}) = C_{\mathscr{T}}(\{\mathrm{Y},\mathrm{Z}\})$$

A mapping $h : \mathscr{T} \to \mathbb{R}_0^+$ is a *level function* on $\mathscr{T}$ iff

l1. $\forall \mathrm{X} \in \mathbf{U} : h(\{\mathrm{X}\}) = 0$
l2. $C_p \subseteq C_q \Rightarrow h(C_p) \leq h(C_q)$

A simple example of level function is $h(C) = \mathrm{card}(C) - 1$.

Every hierarchy / level function determines an ultrametric dissimilarity on $\mathbf{U}$

$$\delta(\mathrm{X},\mathrm{Y}) = h(C_{\mathscr{T}}(\{\mathrm{X},\mathrm{Y}\}))$$

The converse is also true (see [23]): Let $d$ be an ultrametric on $\mathbf{U}$. Denote a closed ball in X with radius $r$ with $\overline{B}(\mathrm{X},r) = \{\mathrm{Y} \in \mathbf{U} : d(\mathrm{X},\mathrm{Y}) \leq r\}$. Then for any given set $A \subset \mathbb{R}^+$ the set

$$\mathbf{C}(A) = \{\overline{B}(\mathrm{X},r) : \mathrm{X} \in \mathbf{U}, r \in A\} \cup \{\{\mathbf{U}\}\} \cup \{\{\mathrm{X}\} : \mathrm{X} \in \mathbf{U}\}$$

is a complete hierarchy, and $h(C) = \mathrm{diam}(C)$ is a level function.

The pair $(\mathscr{T},h)$ is called a *dendrogram* or a *clustering tree* because it can be visualized as a tree.

Unfortunately, the function $h_D(C) = D(C_p,C_q)$, $C = C_p \cup C_q$ is not always a level function – for some $D$s the *inversion*s, $D(C_p,C_q) > D(C_p \cup C_q, C_s)$, are possible. Batagelj showed [4]:

**Proposition 3.16** *$h_D$ is a level function for the Lance-Williams procedure ($\alpha_1$, $\alpha_2$, $\beta$, $\gamma$) iff:*

*(i)* $\gamma + \min(\alpha_1, \alpha_2) \geq 0$
*(ii)* $\alpha_1 + \alpha_2 \geq 0$
*(iii)* $\alpha_1 + \alpha_2 + \beta \geq 1$

The dissimilarity $D$ has the *reducibility* property iff

$$D(C_p, C_q) \leq t,\ D(C_p, C_s) \geq t,\ D(C_q, C_s) \geq t \ \Rightarrow\ D(C_p \cup C_q, C_s) \geq t$$

**Proposition 3.17** *(Bruynooghe, 1977) If a dissimilarity $D$ has the reducibility property then $h_D$ is a level function.*

In the book [11] (Subsection 9.3.5) we presented a fast agglomerative clustering procedure based on the nearest neighbor graph for dissimilarities that have reducibility property.

### 3.3.4 Adding hierarchical methods

Suppose that we already built a clustering tree $\mathscr{T}$ over the set of units **U**. To add a new unit X into the tree $\mathscr{T}$ we start in the root and branch down. Assume that we reached the node corresponding to cluster $C$, which was obtained by joining subclusters $C_p$ and $C_q$, $C = C_p \cup C_q$. There are three possibilities: or to add X to $C_p$, or to add X to $C_q$, or to form a new cluster $\{\mathrm{X}\}$. See Figure 3.1.

Consider again the 'greedy approximation'

$$P(\mathbf{C}_k^\bullet) = P(\mathbf{C}_{k+1}^\bullet) + D(C_p, C_q)$$

where $D(C_p, C_q) = \min_{C_u, C_v \in \mathbf{C}_{k+1}^\bullet} D(C_u, C_v)$ and $\mathbf{C}_i^\bullet$ are greedy solutions. Since we wish to minimize the value of criterion $P$ it follows from the greedy relation that we have to select the case corresponding to the maximal among values $D(C_p \cup \{\mathrm{X}\}, C_q)$, $D(C_q \cup \{\mathrm{X}\}, C_p)$ and $D(C_p \cup C_q, \{\mathrm{X}\})$.

This is a basis for the adding clustering method. We start with a tree on the first two units and then successively add to it the remaining units. The unit X is included into all clusters through which we branch it down.

### 3.3.5 Leaders method

In order to support our intuition in further development we shall briefly describe a simple version of dynamic clusters method – the *leaders* or $k$-means method [34; 22] which is a basis of several recent 'data-mining' and 'big data' methods. In the leaders method the criterion function has the form SR. The basic scheme of leaders method is simple:

select $\mathbf{C}_0$; $\mathbf{C} := \mathbf{C}_0$;
**repeat**
  determine for each $C \in \mathbf{C}$ its leader $\overline{C}$;
  the new clustering $\mathbf{C}$ is obtained by assigning each unit to its nearest leader
**until** leaders stabilize

To obtain a 'good' solution and an impression of its quality we can repeat this procedure with different (random) $\mathbf{C}_0$.

The dynamic clusters method is a generalization of the above scheme. Let us denote:

$\Lambda$ – set of *representatives*
$\mathrm{L} \subseteq \Lambda$ – *representation*

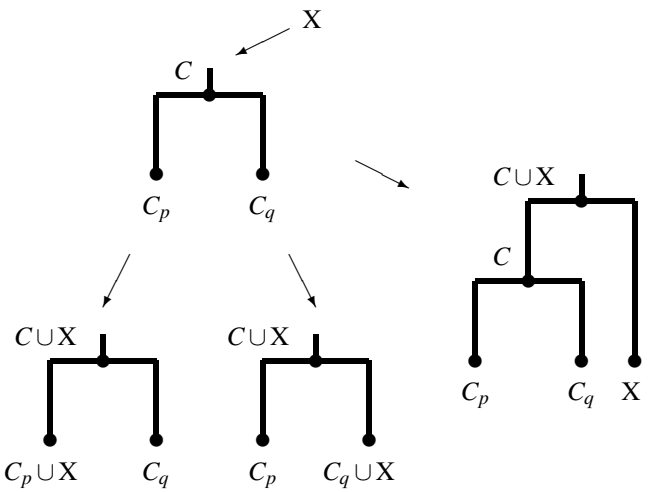



Figure 3.1: Adding hierarchical method

$\Psi$ – set of *feasible representations*
$W : \Phi \times \Psi \to \mathbb{R}_0^+$ – *extended criterion function*
$G : \Phi \times \Psi \to \Psi$ – *representation function*
$F : \Phi \times \Psi \to \Phi$ – *clustering function*

and the following conditions have to be satisfied:

W0. $P(\mathbf{C}) = \min_{\mathrm{L} \in \Psi} W(\mathbf{C}, \mathrm{L})$

the functions $G$ and $F$ tend to improve (diminish) the value of the extended criterion function $W$:

W1. $W(\mathbf{C}, G(\mathbf{C}, \mathrm{L})) \leq W(\mathbf{C}, \mathrm{L})$
W2. $W(F(\mathbf{C}, \mathrm{L}), \mathrm{L}) \leq W(\mathbf{C}, \mathrm{L})$

then the *dynamic clusters method* (DCM) can be described by the scheme:

select $\mathbf{C} := \mathbf{C}_0$; $\mathrm{L} := \mathrm{L}_0$;
**repeat**
  $\mathrm{L} := G(\mathbf{C}, \mathrm{L})$;
  $\mathbf{C} := F(\mathbf{C}, \mathrm{L})$
**until** the clustering $\mathbf{C}$ stabilizes

To this scheme corresponds the sequence $v_n = (\mathbf{C}_n, \mathrm{L}_n), n \in \mathbb{N}$ determined by relations

$$\mathrm{L}_{n+1} = G(\mathbf{C}_n, \mathrm{L}_n) \quad \text{and} \quad \mathbf{C}_{n+1} = F(\mathbf{C}_n, \mathrm{L}_{n+1})$$

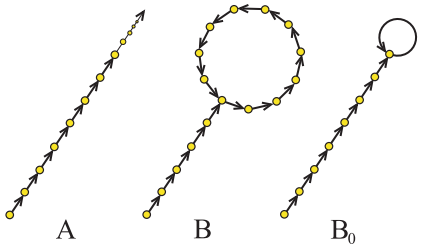
and the sequence of values of the extended criterion function $u_n = W(\mathbf{C}_n, \mathrm{L}_n)$. Let us also denote $u^* = P(\mathbf{C}^*)$. Then it holds:

**Proposition 3.18** *For every* $n \in \mathbb{N}$, $u_{n+1} \leq u_n$, $u^* \leq u_n$, *and if for* $k > m$, $v_k = v_m$ *then* $\forall n \geq m : u_n = u_m$.

Proposition 3.18 states that the sequence $u_n$ is monotonically decreasing and bounded, therefore it is convergent. Note that the limit of $u_n$ is not necessarily $u^*$ – the dynamic clusters method is a local optimization method.

Two types of sequences $v_n$ are possible:

Type A: $\neg\exists k,m \in \mathbb{N}, k > m : v_k = v_m$
Type B: $\exists k,m \in \mathbb{N}, k > m : v_k = v_m$
Type $B_0$: Type B with $k = m+1$




For DCM to be an algorithm the corresponding sequences $v_n$ should be of type $B$. The DCM sequence $(v_n)$ is of type B if

- sets $\Phi$ and $\Psi$ are both finite. For example, when we select a representative of $C$ among its members.
- $\exists\delta > 0 : \forall n \in \mathbb{N} : (v_{n+1} \neq v_n \Rightarrow u_n - u_{n+1} > \delta)$

Because the sets $\mathbf{U}$ and consequently $\Phi$ are finite we expect from a good dynamic clusters procedure to stabilize in finite number of steps – is of type B.

The conditions W0, W1 and W2 are not strong enough to ensure this. We shall try to compensate the possibility that the set of representations $\Psi$ is infinite by the additional requirement:

W3. $W(\mathbf{C}, G(\mathbf{C},\mathrm{L})) = W(\mathbf{C},\mathrm{L}) \Rightarrow \mathrm{L} = G(\mathbf{C},\mathrm{L})$

With this requirement the 'symmetry' between $\Phi$ and $\Psi$ is distroyed. We could reestablish it by the requirement:

W4. $W(F(\mathbf{C},\mathrm{L},\mathrm{L})) = W(\mathbf{C},\mathrm{L}) \Rightarrow \mathbf{C} = F(\mathbf{C},\mathrm{L})$

but it turns out that W4 often fails. For this reason we shall avoid it.

**Proposition 3.19** *If W3 holds and if there exists $m \in \mathbb{N}$ such that $u_{m+1} = u_m$, then also $\mathrm{L}_{m+1} = \mathrm{L}_m$.*

Usually, in the applications of the DCM, the clustering function takes the form $F : \Psi \to \Phi$. In this case the condition W2 simplifies to: $W(F(\mathrm{L}),\mathrm{L}) \leq W(\mathbf{C},\mathrm{L})$ which can be expressed also as $F(\mathrm{L}) \in \mathrm{Min}_{\mathbf{C}\in\Phi} W(\mathbf{C},\mathrm{L})$. For such, *simple* clustering functions it holds:

**Proposition 3.20** *If the clustering function $F$ is simple and if there exists $m \in \mathbb{N}$ such that $\mathrm{L}_{m+1} = \mathrm{L}_m$, then for every $n \geq m : v_n = v_m$.*

What can be said about the case when $G$ is *simple* – has the form $G : \Phi \to \Psi$?

**Proposition 3.21** *If W3 holds and the representation function $G$ is simple then:*

*a.* $G(\mathbf{C}) = \arg\min_{\mathrm{L}\in\Psi} W(\mathbf{C},\mathrm{L})$
*b.* $\exists k,m \in \mathbb{N}, k > m \forall i \in \mathbb{N} : v_{k+i} = v_{m+i}$
*c.* $\exists m \in \mathbb{N} \forall n \geq m : u_n = u_m$
*d.* *if also $F$ is simple then* $\exists m \in \mathbb{N} \forall n \geq m : v_n = v_m$

In the original dynamic clusters method [22] both functions $F$ and $G$ are simple – $F : \Psi \to \Phi$ and $G : \Phi \to \Psi$.

If also W3 holds and the functions $F$ and $G$ are simple, then:

G0. $G(\mathbf{C}) = \text{argmin}_{\mathrm{L}\in\Psi} W(\mathbf{C},\mathrm{L})$

and

F0. $F(\mathrm{L}) \in \text{Min}_{\mathbf{C}\in\Phi} W(\mathbf{C},\mathrm{L})$

In other words, given an extended criterion function $W$, the relations G0 and F0 define an appropriate pair of functions $G$ and $F$ such that the DCM stabilizes in finite number of steps.

### 3.4 Clustering of Graphs and Networks

When the set of units $\mathbf{U}$ consists of graphs (for example chemical molecules) we speak about *clustering of graphs* (networks). For this purpose we can use standard clustering approaches provided that we have an appropriate definition of dissimilarity between graphs.

The first approach is to define a vector description $[\mathbf{G}] = [g_1, g_2, \ldots, g_m]$ of each graph $\mathbf{G}$, and then use some standard dissimilarity $\delta$ on $\mathbb{R}^m$ to compare these vectors $d(\mathbf{G}_1,\mathbf{G}_2) = \delta([\mathbf{G}_1],[\mathbf{G}_2])$. We can get $[\mathbf{G}]$, for example, by:

- **Invariants:** compute the values of selected invariants (indices) on each graph [54].
- **Fragments count:** select a collection of subgraphs (fragments), for example triads, and count the number of appearences of each – a *fragments spectrum* [6; 50].

Let **Gph** be the set of all graphs. An *invariant* of a graph is a mapping $i$: **Gph** $\to \mathbb{R}$ which is constant over isomorphic graphs

$$\mathbf{G} \approx \mathbf{H} \Rightarrow i(\mathbf{G}) = i(\mathbf{H})$$

The number of nodes, the number of arcs, the number of edges, maximum degree $\Delta$, chromatic number $\chi$, etc. are all graph invariants. Invariants have an important role in examining the isomorphism of two graphs. To prove that $\mathbf{G}$ is not isomorphic to $\mathbf{H}$ it is enough to find an invariant $i$ such that $i(\mathbf{G}) \neq i(\mathbf{H})$.

Invariants on *families* of graphs are called *structural properties*: Let $\mathscr{F} \subseteq$ **Gph** be a family of graphs. A property $i$: $\mathscr{F} \to \mathbb{R}$ is *structural* on $\mathscr{F}$ iff

$$\forall \mathbf{G},\mathbf{H} \in \mathscr{F} : (\mathbf{G} \approx \mathbf{H} \Rightarrow i(\mathbf{G}) = i(\mathbf{H}))$$

A collection $\mathscr{I}$ of invariants/structural properties is *complete* iff

$$(\forall i \in \mathscr{I} : i(\mathbf{G}) = i(\mathbf{H})) \Rightarrow \mathbf{G} \approx \mathbf{H}$$

In most cases (families of graphs) there is no efficiently computable complete collection.

Different dissimilarities between strings are based on *transformations*: insert, delete, transpose [44; 40]. For binary trees Robinson considered a dissimilarity based on the transformation of *neighbors exchange over an edge* (see Figure 3.2).

There is a natural generalization of this approach to graphs and other structured objects [6]: Let $\mathscr{T} = \{T_k\}$ be a set of *basic transformations* of units $T_k : \mathscr{U} \to \mathscr{U}$ and $v : \mathscr{T} \times \mathscr{U} \to \mathbb{R}^+$ a value or cost of transformation, which satisfy the conditions:

$$\forall T \in \mathscr{T} : (T : \mathrm{X} \mapsto \mathrm{Y} \Rightarrow \exists S \in \mathscr{T} : (S : \mathrm{Y} \mapsto \mathrm{X} \wedge v(T,\mathrm{X}) = v(S,\mathrm{Y})))$$

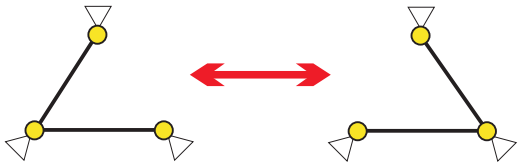


Figure 3.2: Neighbors exchange over an edge

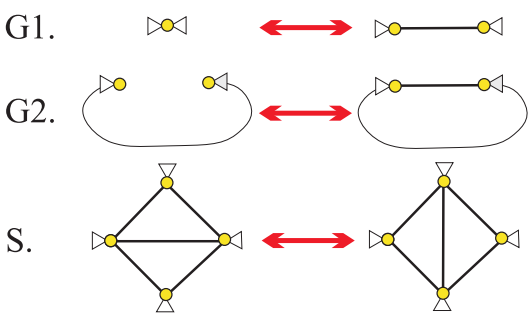



Figure 3.3: Examples of transformations

and $v(\mathrm{id}, X) = 0$.

Suppose that for each pair $\mathrm{X}, \mathrm{Y} \in \mathscr{U}$ there exists a finite sequence $\tau = (T_1, T_2, \ldots, T_t)$ such that: $\tau(\mathrm{X}) = T_t \circ T_{t-1} \circ \ldots \circ T_1(X) = \mathrm{Y}$. Then we can define:

$$d(\mathrm{X}, \mathrm{Y}) = \min_{\tau}(v(\tau(\mathrm{X})) : \tau(\mathrm{X}) = \mathrm{Y})$$

where

$$v(\tau(X)) = \begin{cases} 0 & \tau = \mathrm{id} \\ v(\eta(T(\mathrm{X}))) + v(T, \mathrm{X}) & \tau = \eta \circ T \end{cases}$$

It is easy to verify that so defined dissimilarity $d(\mathrm{X}, \mathrm{Y})$ is a distance.

For example, see Figure 3.3, using the transformations G1 and G2 we can transform any pair of connected simple graphs one to the other. For triangulations of the plane on $n$ nodes S is such a transformation.

## 3.5 Clustering in Graphs and Networks

Since in a graph $\mathbf{G} = (V, L)$ we have two kinds of objects – nodes and links we can speak about *clustering of nodes* and *clustering of links*. Usually we deal with clustering of nodes.

### 3.5.1 Indirect approach

Again we can use the standard clustering methods provided that we have an appropriate definition of dissimilarity between nodes. The usual approach is to define a vector description $[v] = [t_1, t_2, \ldots, t_m]$ of each node $v \in V$, and then use some standard dissimilarity $\delta$ on $\mathbb{R}^m$ to compare these vectors $d(u, v) = \delta([u], [v])$.

We can assign to each node $v$ also different neighborhoods, such as $N(v) = \{u \in V : (v,u) \in L\}$, and other sets. In these cases the dissimilarities between sets are used on them.

For a given graph $\mathbf{G} = (V,L)$ a property $t : V \to \mathbb{R}$ is *structural* iff for every automorphism $\varphi$ of $\mathbf{G}$ it holds

$$\forall v \in V : t(v) = t(\varphi(v))$$

Examples of such properties are

$t(v) =$ degree (number of neighbors) of node $v$
$t(v) =$ number of nodes at distance $d$ from node $v$
$t(v) =$ number of triads of type $x$ at node $v$
$t(v) =$ number of graphlets of type $x$ at node $v$ [50]

For a given graph $\mathbf{G} = (V,L)$ a *property of pairs of nodes* $q : V \times V \to \mathbb{R}$ is *structural* if for every automorphism $\varphi$ of $\mathbf{G}$ it holds

$$\forall u,v \in V : q(u,v) = q(\varphi(u),\varphi(v))$$

Some examples of structural properties of pairs of nodes

$q(u,v) =$ **if** $(u,v) \in L$ **then** 1 **else** 0
$q(u,v) =$ number of common neighbors of units $u$ and $v$
$q(u,v) =$ length of the shortest path from $u$ to $v$

Using a selected property of pairs of nodes $q$ we can describe each node $u$ with a vector

$$[u] = [q(u,v_1), q(u,v_2), \ldots, q(u,v_n), q(v_1,u), \ldots, q(v_n,u)]$$

and again define the dissimilarity between nodes $u,v \in V$ as $d(u,v) = \delta([u],[v])$.

*Corrected* dissimilarities based on properties of pairs of nodes for measuring the similarity between nodes $v_i$ and $v_j$ ($p \geq 0$) should be used [24] such as:
Corrected Manhattan:

$$d_c(p)(v_i,v_j) = \sum_{\substack{s=1 \\ s \neq i,j}}^{n} (|q_{is} - q_{js}| + |q_{si} - q_{sj}|) + p \cdot (|q_{ii} - q_{jj}| + |q_{ij} - q_{ji}|)$$

Corrected Euclidean:

$$d_e(p)(v_i,v_j) = \sqrt{\sum_{\substack{s=1 \\ s \neq i,j}}^{n} ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2) + p \cdot ((q_{ii} - q_{jj})^2 + (q_{ij} - q_{ji})^2)}$$

The corrected dissimilarities with $p = 1$ are usually used.

### 3.5.2 Direct approach – blockmodeling

A partition $\mathbf{C} = \{C_i\}$ splits the set of links (arcs) $L \subseteq V \times V$ into *block*s $B_{ij} = L \cap C_i \times C_j$ – a subgraph of arcs from cluster $C_i$ to cluster$C_j$. In blockmodeling we are trying to find a partition that produces blocks of selected types (complete, empty, regular, etc.), may be with some errors [24]. Usually the relocation method is used for solving the corresponding optimization problems.

In terms of blockmodeling the criterion functions for indirect approach based on dissimilarities are usually expressing the notion of structural equivalence.

### 3.5.3 Graph theory approaches

The basic decomposition of graphs is to (weakly) connected components – partition of nodes (and links); and to (weakly) biconnected components – partition of links. For both very efficient algorithms exist [20]. For directed graphs the fundamental decomposition results can be found in [19].

From a network $\mathbf{N} = (V, L, w)$ we can get for a treshold $t$ a link-cut – a subnetwork $\mathbf{N}(t) = (V, L_t, w)$ where $L_t = \{p \in L : w(p) \geq t\}$. From it we can get a clustering $\mathbf{C}(t)$ with connected components as clusters. For different tresholds these clusterings form a hierarchy. An elaborated version of cuts is provided with *islands* approach [11], Subsection 2.9.1. Islands also form a hierarchy for a selected node property of a given network.

In seventies and eighties Matula studied different types of connectivities in graphs and structures they induce [46]. In most cases the algorithms are too demanding to be used on larger graphs. A nice overview of connectivity algorithms was made by Esfahanian [25].

## 3.6 Agglomerative method for relational constraints

Suppose that the units are described by attribute data $a\colon \mathbf{U} \to [\mathbf{U}]$ and are related by a binary *relation* $R \subseteq \mathbf{U} \times \mathbf{U}$ that determine the *relational data* or *network* $(\mathbf{U}, R, a)$.

We want to cluster the units according to a (dis)similarity of their descriptions, but also considering the relation $R$ which imposes *constraints* on the set of feasible clusterings, usually in the following form:

$$\Phi(R) = \begin{array}{l}\{\mathbf{C} \in P(\mathbf{U}) : \text{each cluster } C \in \mathbf{C} \text{ induces a subgraph } (C, R \cap C \times C) \\ \text{in the graph } (\mathbf{U}, R) \text{ of the required type of connectedness}\}\end{array}$$

and criterion function of type SR:

$$P(\mathbf{C}) = \sum_{C \in \mathbf{C}} p(C), \quad p(C) = \sum_{X \in C} d(\mathrm{X}, T_C)$$

We can define different types of sets of feasible clusterings for the same relation $R$. Some examples of *types of relational constraint* $\Phi^i(R)$ are [27]

| clusterings | type of connectedness |
|---|---|
| $\Phi^1(R)$ | weakly connected units |
| $\Phi^2(R)$ | weakly connected units that contain at most one center |
| $\Phi^3(R)$ | strongly connected units |
| $\Phi^4(R)$ | clique |
| $\Phi^5(R)$ | the existence of a trail containing all the units of the cluster |

A *trail* is a walk in a graph in which all arcs are distinct.

The set $R(\mathrm{X}) = \{\mathrm{Y} : \mathrm{X}R\mathrm{Y}\}$ is a *set of successors* of unit $X \in \mathbf{U}$ and for a cluster $C \subseteq \mathbf{U}$ $R(C) = \bigcup_{\mathrm{X} \in C} R(\mathrm{X})$. A set of units $L \subseteq C$ is a *center* of a cluster $C$ in the clustering of type $\Phi^2(R)$ iff the subgraph induced by $L$ is strongly connected and $R(L) \cap (C \setminus L) = \emptyset$.

The sets of feasible clusterings $\Phi^i(R)$ are linked as follows: $\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$ and $\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$. If the relation $R$ is symmetric, then $\Phi^3(R) = \Phi^1(R)$. If the relation $R$ is an equivalence relation, then $\Phi^4(R) = \Phi^1(R)$.

The corresponding fusibility predicates are as follows:

$\psi^1(C_1,C_2) \equiv \exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : (\mathrm{X}R\mathrm{Y} \vee \mathrm{Y}R\mathrm{X})$
$\psi^2(C_1,C_2) \equiv (\exists \mathrm{X} \in L_1 \exists \mathrm{Y} \in C_2 : \mathrm{X}R\mathrm{Y}) \vee (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in L_2 : \mathrm{Y}R\mathrm{X})$
$\psi^3(C_1,C_2) \equiv (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : \mathrm{X}R\mathrm{Y}) \wedge (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : \mathrm{Y}R\mathrm{X})$
$\psi^4(C_1,C_2) \equiv \forall \mathrm{X} \in C_1 \forall \mathrm{Y} \in C_2 : (\mathrm{X}R\mathrm{Y} \wedge \mathrm{Y}R\mathrm{X})$
$\psi^5(C_1,C_2) \equiv (\exists \mathrm{X} \in T_1 \exists \mathrm{Y} \in I_2 : \mathrm{X}R\mathrm{Y}) \vee (\exists \mathrm{X} \in I_1 \exists \mathrm{Y} \in T_2 : \mathrm{Y}R\mathrm{X})$

where $I$ denotes initial nodes in a cluster $C$ and $T$ denotes terminal nodes in a cluster $C$. For $\psi^3$ the property F5 fails.

We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint [26; 27; 11]. Here we present only the hierarchical method:

1. $k := n$; $\mathbf{C}(k) := \{\{\mathrm{X}\} : \mathrm{X} \in \mathbf{U}\}$;
2. **while** $\exists C_i, C_j \in \mathbf{C}(k)$: $(i \neq j \wedge \psi(C_i,C_j))$ **repeat**
2.1. $(C_p,C_q) := \operatorname{argmin}\{D(C_i,C_j) : i \neq j \wedge \psi(C_i,C_j)\}$;
2.2. $C := C_p \cup C_q$; $k := k-1$;
2.3. $\mathbf{C}(k) := \mathbf{C}(k+1) \setminus \{C_p,C_q\} \cup \{C\}$;
2.4. determine $D(C,C_s)$ for all $C_s \in \mathbf{C}(k)$
2.5. adjust the relation $R$ as required by the clustering type
3. $m := k$

To get clustering procedures we have to further elaborate the questions how to adjust the relation after joining two clusters and how to update the dissimilarity $D(C,C_s)$ .

In Figures 3.4 and 3.6 four adjusting *strategies* are presented. They are compatible with the corresponding types of constraints: $\Phi^1$ – tolerant, $\Phi^2$ – leader, $\Phi^4$ – strict, and $\Phi^5$ – two.way. In Figure 3.5 an example of application of strategies is presented.

The effects of strategies can be described also as updates of the sets of succecors $R(C)$:

**tolerant**

$$\begin{aligned} R(C_r) &= \{C_r\} \cup R(C_p) \cup R(C_q) \setminus \{C_p,C_q\} \\ R(C_s) &= \{C_r\} \cup R(C_s) \setminus \{C_p,C_q\}, \quad \text{for } s \neq r \wedge \{C_p,C_q\} \cap R(C_s) \neq \emptyset \end{aligned}$$

**strict**

$$R(C_r) = \begin{cases} \{C_r\} \cup R(C_p) \cup R(C_q) \setminus \{C_p,C_q\}, & \text{for } C_q R C_p \\ \{C_r\} \cup R(C_s) \setminus \{C_p,C_q\}), & \text{otherwise} \end{cases}$$

$$R(C_s) = \begin{cases} \{C_s\} \cup R(C_s) \setminus \{C_p,C_q\}), & \text{for } \begin{array}{l} s \neq r \wedge (C_p \in R(C_s) \vee \\ C_q \in R(C_s) \wedge C_q R C_p) \end{array} \\ R(C_s) \setminus \{C_p,C_q\}), & \text{otherwise for } s \neq r \end{cases}$$

**leader**

$$R(C_r) = \begin{cases} \{C_r\} \cup R(C_p) \cup R(C_q) \setminus \{C_p,C_q\}, & \text{for } C_q R C_p \\ \{C_r\} \cup R(C_s) \setminus \{C_p,C_q\}, & \text{otherwise} \end{cases}$$

$$R(C_s) = \begin{cases} \{C_s\} \cup R(C_s) \setminus \{C_p,C_q\}, & \text{for } s \neq r \wedge \{C_p.C_q\} \cap R(C_s) \neq \emptyset \\ R(C_s) \setminus \{C_p,C_q\}, & \text{otherwise for } s \neq r \end{cases}$$

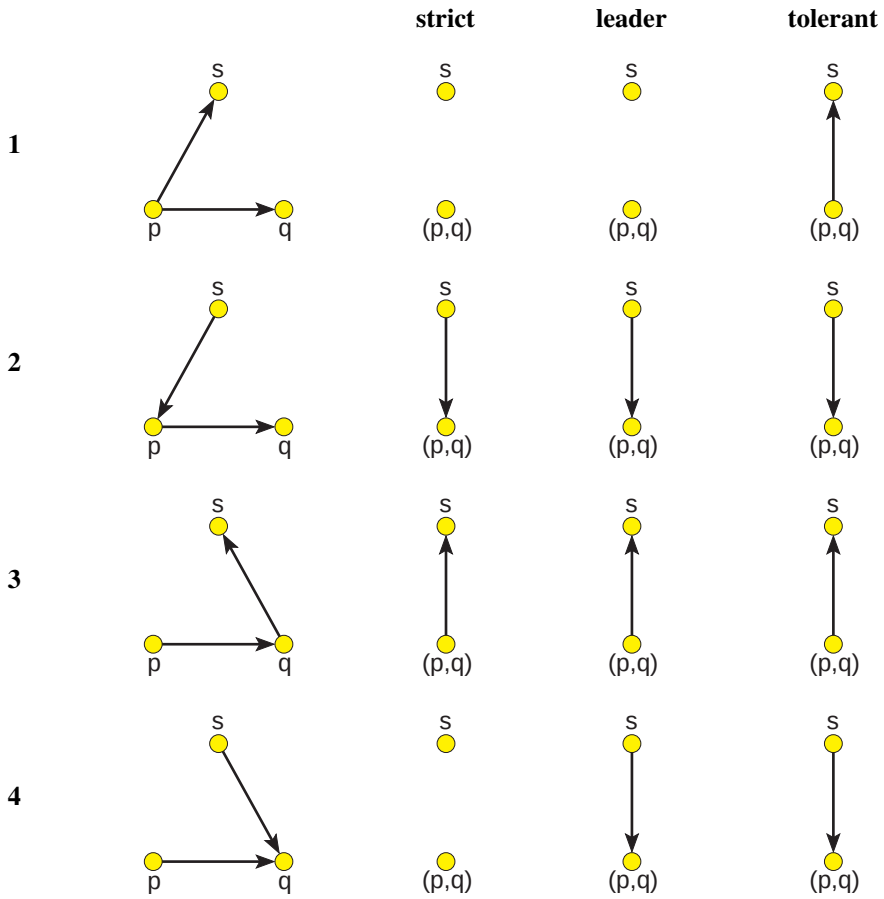



Figure 3.4: Types of relational constraints

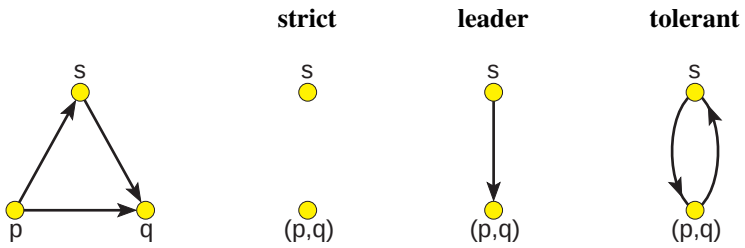



Figure 3.5: A composite example

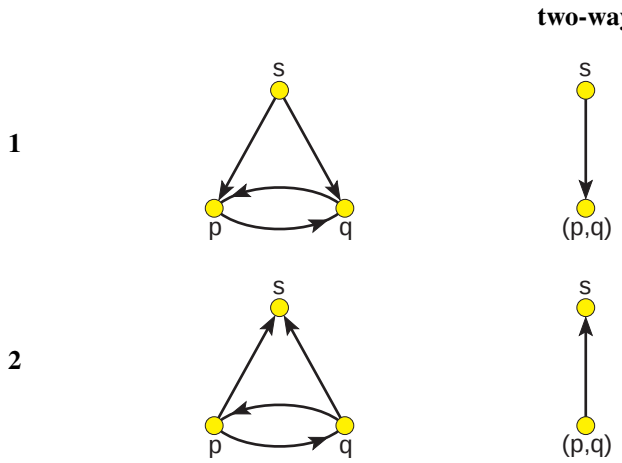



Figure 3.6: The two-way strategy

**two-way**

$$R(C_r) = \{C_r\} \cup (R(C_p) \cap R(C_q)) \setminus \{C_p, C_q\}$$

$$R(C_s) = \begin{cases} \{C_s\} \cup R(C_s) \setminus \{C_p, C_q\}, & \text{for } s \neq r \wedge \{C_p.C_q\} \subseteq R(C_s) \\ R(C_s) \setminus \{C_p, C_q\}, & \text{otherwise for } s \neq r \end{cases}$$

In the original approach [26; 27] a complete dissimilarity matrix is needed. To obtain fast algorithms that can be applied to large data sets we propose to *consider only the dissimilarities between linked units*. For large data sets we assume that the relation $R$ is *sparse*.

The step 2.4. "determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$" in the agglomerative procedure requires the adjustment of dissimilarities – computing the dissimilarities between new cluster $C$ and other remaining clusters. In the case of the relational constraints we can limit the computation only to clusters that are related/linked to $C$.

This can be done efficiently in the following two cases:

- **first approach**: we define a dissimilarity $D(S,T)$ between clusters $S$ and $T$ that allows quick updates (as in Lance-Williams formula)
- **second approach**: to each cluster we assign a representative and we can efficiently compute a representative of merged clusters and a dissimilarity between clusters in terms of their representatives.

The first approach was described already in [11]. Let $(\mathbf{U}, R)$, $R \subseteq \mathbf{U} \times \mathbf{U}$ be a graph and $\emptyset \subset S, T \subset \mathbf{U}$ and $S \cap T = \emptyset$. We call a *block* of relation $R$ for $S$ and $T$ its part $R(S,T) = R \cap S \times T$. The *symmetric closure* of relation $R$ we denote with $\hat{R} = R \cup R^{-1}$. It holds: $\hat{R}(S,T) = \hat{R}(T,S)$.

For all dissimilarities between clusters $D(S,T)$ we set:

$$D(\{s\},\{t\}) = \begin{cases} d(s,t) & s\hat{R}t \\ \infty & \text{otherwise} \end{cases}$$

where $d$ is a selected dissimilarity between units.

**Minimum**

$$D_{\min}(S,T) = \min_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$D_{\min}(S,T_1 \cup T_2) = \min(D_{\min}(S,T_1), D_{\min}(S,T_2))$$

**Maximum**

$$D_{\max}(S,T) = \max_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$D_{\max}(S,T_1 \cup T_2) = \max(D_{\max}(S,T_1), D_{\max}(S,T_2))$$

**Average**

$w : V \to \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in \mathbf{U}$.

$$D_{\mathrm{a}}(S,T) = \frac{1}{w(\hat{R}(S,T))} \sum_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$w(\hat{R}(S,T_1 \cup T_2)) = w(\hat{R}(S,T_1)) + w(\hat{R}(S,T_2))$$

$$D_{\mathrm{a}}(S,T_1 \cup T_2) = \frac{w(\hat{R}(S,T_1))}{w(\hat{R}(S,T_1 \cup T_2))} D_{\mathrm{a}}(S,T_1) + \frac{w(\hat{R}(S,T_2))}{w(\hat{R}(S,T_1 \cup T_2))} D_{\mathrm{a}}(S,T_2)$$

All three disimilarities have the reducibility property. In this case also the *nearest neighbors network* for a given network is preserved after joining the nearest clusters. This allows us to develop a very fast agglomerative hierarchical clustering procedure [48]. It is available in the program **Pajek**. The same approach can be extended also to clustering of links of network [16] by transforming a given network into its line-graph in which the original links become new nodes.

For the second approach we need the representatives of clusters and a dissimilarity between clusters that can be expressed in terms of representatives. For symbolic objects described by discrete distributions (histograms, barcharts) there exist some possibilities [12].

### 3.6.1 Software support

The first approach is implemented for weighted networks (weight is a dissimilarity) in **Pajek** – a program for analysis and visualization of large networks [49]. We also implemented it in R package `cluRC` [7]. An implementation in R of the second approach is still a work in progress.

## 3.7 Examples

To illustrate the hierarchical clustering with relational constraint we present two examples:

- Clustering of US states according to the selected variables into geographically contiguous clusters.
- Clustering of authors from network clustering literature (see Chapter 2) according to their citations into clusters with a single leaders group.

Table 3.5: Averages for Ward's clustering

| | *crime* | *violent* | *smoking* | *drinking* | *diabetes* | *opioid* | *income* |
|---|---|---|---|---|---|---|---|
| $C_1$ | 8.7857 | 496.45 | 0.2251 | 0.1447 | 0.1173 | 10.857 | 44631 |
| $C_2$ | 5.9118 | 427.96 | 0.1826 | 0.1714 | 0.1048 | 13.853 | 53535 |
| $C_3$ | 2.6333 | 239.99 | 0.1755 | 0.2023 | 0.0847 | 10.767 | 55908 |
| $C_4$ | 3.8000 | 300.99 | 0.1521 | 0.1699 | 0.0903 | 23.657 | 69947 |
| $C_5$ | 4.9000 | 273.02 | 0.2645 | 0.1195 | 0.1210 | 33.500 | 43727 |
| *all* | 4.9563 | 354.23 | 0.1856 | 0.1748 | 0.0989 | 14.700 | 54963 |
| $C_1$ | **1.5723** | **1.0924** | 1.1363 | −0.9927 | 1.2826 | −0.4229 | −1.1990 |
| $C_2$ | 0.3923 | 0.5663 | −0.0843 | −0.1123 | 0.4134 | −0.0932 | −0.1657 |
| $C_3$ | **−0.9537** | **−0.8776** | −0.2887 | **0.9094** | **−0.9924** | **−0.4328** | 0.1097 |
| $C_4$ | −0.4747 | −0.4090 | **−0.9605** | −0.1617 | −0.6005 | 0.9856 | **1.7389** |
| $C_5$ | −0.0231 | −0.6239 | **2.2668** | **−1.8260** | **1.5416** | **2.0687** | **−1.3039** |

### 3.7.1 US data 2016

From the site `https://datausa.io/profile/geo/united-states/` we obtained the data about US states in 2016 for the following variables: `crime` – homicide deaths, `violent` – violent crimes, `smoking` – adult smoking prevalence, `drinking` – excessive drinking prevalence, `diabetes` – diabetes prevalence, `opioid` – opioid overdose death rate, and `income` – median household income.

In his book *The Stanford GraphBase* [42] D.E. Knuth provided a description of neighboring relation for the contiguous part of USA `contiguous-usa.dat` (without Alaska and Hawai). Because of missing data we removed also Washington DC.

We first applied the Ward's hierarchical clustering method on the squared Euclidean dissimilarity between units with standardized variables. On the basis of the corresponding dendrograme (see left top part of Figure 3.7, we decided to consider a clustering into 5 clusters:

$C_1 = \{AL, AR, LA, MS, NM, TN, SC\}$,
$C_2 = \{AZ, CA, DE, FL, GA, IL, IN, KS, MI, MO, NC, NV, NY, OH, OK, PA, TX\}$,
$C_3 = \{CO, IA, ID, ME, MN, MT, ND, NE, OR, SD, WY, RI, WI, WA, VT\}$,
$C_4 = \{CT, MA, MD, NH, NJ, UT, VA\}$,
$C_5 = \{KY, WV\}$.

In the middle left part of Figure 3.7 is presented the dissimilarity matrix reordered according to the obtained clustering.

To interpret the obtained clusters we produced Table 3.5 with averages of each variable over each cluster for raw and standardized units. The interpretation is left to the reader.

In the bottom left part of Figure 3.7 the obtained clustering/partition is represented with node colors on the network of neighboring US states. We can see that the subnetworks induced by clusters are not all connected (forming contiguous regions). For example the subnetwork induced by $C_4$ has 4 components $\{CT, MA, NH\}$, $\{NJ\}$, $\{MD, VA\}$ and $\{UT\}$.

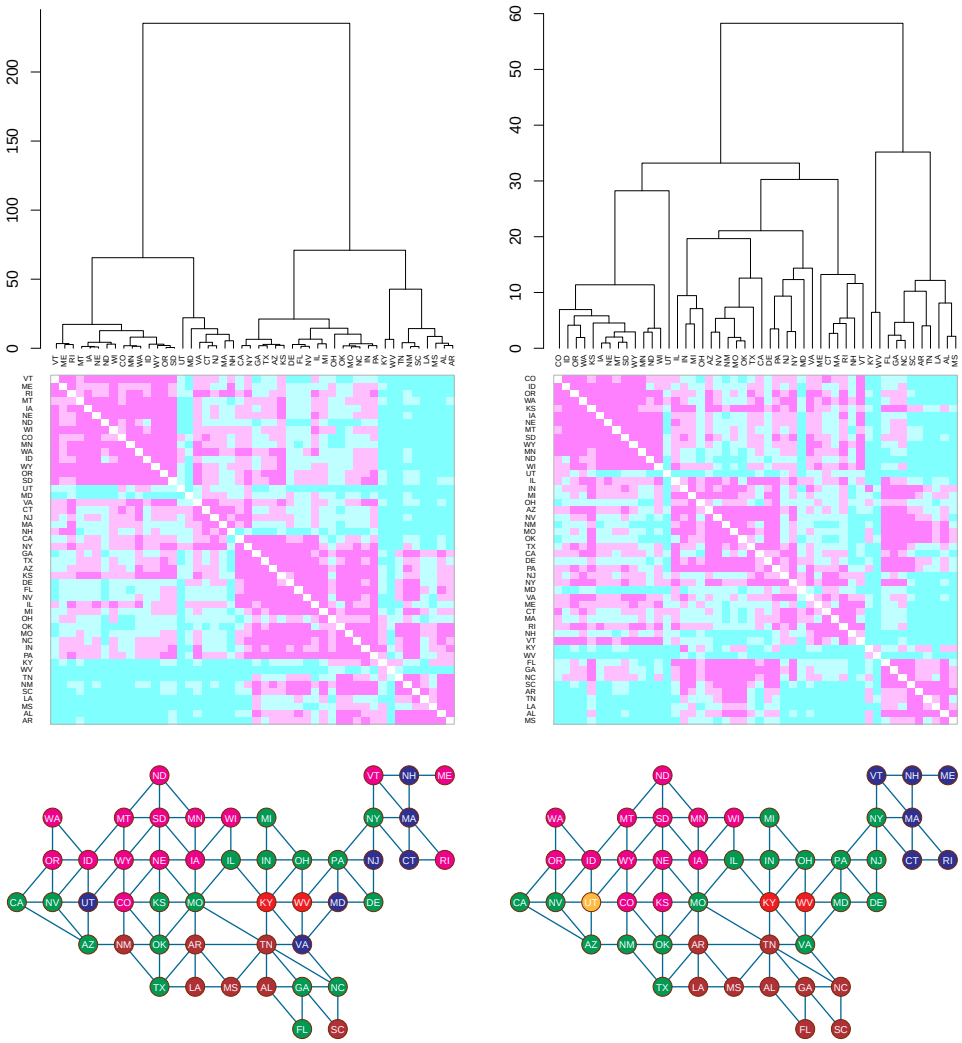



Figure 3.7: Ward clustering (left) and Maximum/Tolerant clustering (right)

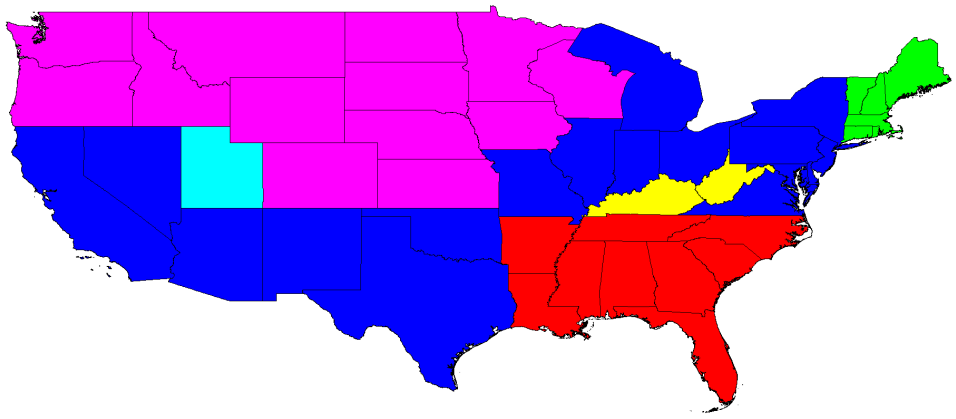


Figure 3.8: Maximum/Tolerant partition on the map

Table 3.6: Averages for Maximum/Tolerant clustering

| | *crime* | *violent* | *smoking* | *drinking* | *diabetes* | *opioid* | *income* |
|---|---|---|---|---|---|---|---|
| $C_1$ | 8.1667 | 462.00 | 0.2140 | 0.1488 | 0.1160 | 10.788 | 46104 |
| $C_2$ | 5.9701 | 425.91 | 0.1804 | 0.1719 | 0.1032 | 15.794 | 57054 |
| $C_3$ | 2.8385 | 265.25 | 0.1765 | 0.2005 | 0.0852 | 7.408 | 55913 |
| $C_4$ | 2.3833 | 234.31 | 0.1660 | 0.1932 | 0.0880 | 26.717 | 62751 |
| $C_5$ | 4.9000 | 273.02 | 0.2645 | 0.1195 | 0.1210 | 33.500 | 43727 |
| $C_6$ | 1.9000 | 204.72 | 0.0970 | 0.1210 | 0.0710 | 16.400 | 62518 |
| *all* | 4.9563 | 354.23 | 0.1856 | 0.1748 | 0.0989 | 14.700 | 54963 |
| $C_1$ | **1.3181** | **0.8278** | 0.8162 | −0.8584 | 1.1929 | −0.4304 | −1.0281 |
| $C_2$ | 0.4165 | 0.5506 | −0.1502 | −0.0928 | 0.3026 | 0.1204 | 0.2427 |
| $C_3$ | −0.8695 | −0.6836 | −0.2620 | **0.8523** | −0.9584 | **−0.8024** | 0.1103 |
| $C_4$ | −1.0564 | −0.9212 | −0.5625 | 0.6087 | −0.7599 | 1.3223 | **0.9039** |
| $C_5$ | −0.0231 | −0.6239 | **2.2668** | **−1.8260** | **1.5416** | **2.0687** | **−1.3039** |
| $C_6$ | **−1.2548** | **−1.1485** | **−2.5445** | −1.7764 | **−1.9456** | 0.1871 | 0.8767 |

Using hierarchical clustering with relational constraint with Maximum/Tolerant strategy we get a clustering that considers a given dissimilarity among units and produces clusters that form contigues regions. On the basis of the dendrograme in the right top part of Figure 3.7 we decided to consider a clustering into 6 clusters:

$C_1 = \{AL, AR, FL, GA, LA, MS, NC, TN, SC\}$,
$C_2 = \{AZ, CA, DE, IL, IN, MD, MI, MO, NJ, NM, NV, NY, OH, OK, PA, VA, TX\}$,
$C_3 = \{CO, IA, ID, KS, MN, MT, ND, NE, OR, SD, WY, WI, WA\}$,
$C_4 = \{CT, MA, ME, NH, RI, VT\}$,
$C_5 = \{KY, WV\}$,
$C_6 = \{UT\}$.

The clusters of obtained clustering/partition induce connected subnetworks, as expected. See right bottom part of Figure 3.7 and Figure 3.8.

The averages of each variable over these clusters for raw and standardized units are given in Table 3.6. In the states of the first cluster $C_1$ is the highest rate of homicide deaths and violent crimes, high adult smoking and diabetes prevalence, and low median household income. The states of the second cluster $C_2$ have all variables around the everage; above average are the homicide deaths and violent crimes. Typical for the cluster $C_3$ is the lowest opioid overdose death rate, the highest excesive drinking and low crime rates. The states of the cluster $C_4$ have the highest income and high excesive drinking and opioid death rate, but low crime, smoking and diabetes. Two states in cluster $C_5$ have the lowest income and excesive drinking, the highest values of smoking, diabetes and opioid death rate, and low crime. Utah, cluster $C_6$, has the lowest values of crime, smoking and diabetes, very low drinking, and high income.

### 3.7.2 Citations among authors from clustering literature

Let us consider again the bibliometric data on the network clustering literature analyzed in Chapter 2. In Section 2.5.3 we analyzed the network **Acite** of citations among authors. Here we will analyze the normalized network of citations among authors $\mathbf{nAcite} = n(\mathbf{WAc})^T * n(\mathbf{CiteC}) * n(\mathbf{WAc})$. Every work has 1 point. They are distributed on arcs of the derived network. The weight $\mathbf{nAcite}[u,v]$ of the arc $(u,v)$ is equal to the fractional share of works co-authored by $u$ that are citing a work co-authored by $v$.

We remove loops (self-citations) and compute weighted indegrees. Let's first look at the largest weighted input degrees – the most cited authors, presented in Table 3.7. Far the most cited ones are Mark Newman and Santo Fortunato. Quite high are also the most important researchers from the field of social network analysis, beginning with Ronald Burt.

In this example we will identify clusters such that the corresponding induced subnetworks are connected and contain a single center – type $\Phi^2$. The **nAcite** weights are similarities, $s \in [\infty, 0]$. To convert them to distances $d$ we can use different transformations. For example $d = \frac{s_{max}}{s} - 1 \in [0, \infty]$ or $d = 1 - \frac{s}{s_{max}} \in [0, 1]$. We selected the second option with $s_{max} = 2.52$. On the obtained network we applied in Pajek the hierarchical clustering with relational constraint procedure with Maximum/Leader strategy and determined the partition of units into clusters of size at most 50. There are 257 such clusters. To reduce their number we decided to consider only clusters with at least 20 units. There are 57 such clusters.

We extracted the corresponding subnetworks of citations among authors for visual inspection. Most of them are (double) star like formed around the most prominent scientists

Table 3.7: The most cited authors / fractional approach

| $i$ | $w_i$ | author | $i$ | $w_i$ | author |
|---|---|---|---|---|---|
| 1 | 329.8886 | NEWMAN_M | 26 | 19.7797 | MALIK_J |
| 2 | 155.4974 | FORTUNAT_S | 27 | 19.7317 | ROSVALL_M |
| 3 | 80.8228 | GIRVAN_M | 28 | 19.2631 | VONLUXBU_U |
| 4 | 51.6716 | BARABASI_A | 29 | 19.1634 | BERGSTRO_C |
| 5 | 45.1972 | BURT_R | 30 | 19.1422 | BARTHELE_M |
| 6 | 42.5944 | ALBERT_R | 31 | 18.6968 | LEFEBVRE_E |
| 7 | 39.6466 | ZACHARY_W | 32 | 18.6552 | GUILLAUM_J |
| 8 | 38.8163 | LANCICHI_A | 33 | 18.6261 | DOREIAN_P |
| 9 | 38.1660 | CLAUSET_A | 34 | 18.3258 | KLEINBER_J |
| 10 | 31.8938 | SCHAEFFE_S | 35 | 18.1618 | BREIGER_R |
| 11 | 31.7021 | STROGATZ_S | 36 | 17.4888 | VICSEK_T |
| 12 | 30.9933 | FREEMAN_L | 37 | 17.4204 | BORGATTI_S |
| 13 | 29.1247 | WASSERMA_S | 38 | 16.9268 | PALLA_G |
| 14 | 29.0661 | MOORE_C | 39 | 16.8126 | OKADA_Y |
| 15 | 26.1896 | FAUST_K | 40 | 16.7620 | BOORMAN_S |
| 16 | 24.8884 | WATTS_D | 41 | 15.8376 | CHUNG_F |
| 17 | 24.7421 | WHITE_H | 42 | 15.8216 | GUIMERA_R |
| 18 | 24.5679 | NEWMARK_N | 43 | 15.7187 | RADICCHI_F |
| 19 | 23.8077 | BLONDEL_V | 44 | 14.9995 | CARLSON_J |
| 20 | 23.0214 | BATAGELJ_V | 45 | 14.9914 | EVERETT_M |
| 21 | 22.6844 | LAMBIOTT_R | 46 | 14.6212 | DUCH_J |
| 22 | 22.5521 | VANDONGE_S | 47 | 14.5231 | AMARAL_L |
| 23 | 20.9136 | ARENAS_A | 48 | 14.4554 | GRANOVET_M |
| 24 | 19.8478 | LESKOVEC_J | 49 | 13.7216 | DERENYI_I |
| 25 | 19.8113 | SHI_J | 50 | 13.7216 | FARKAS_I |

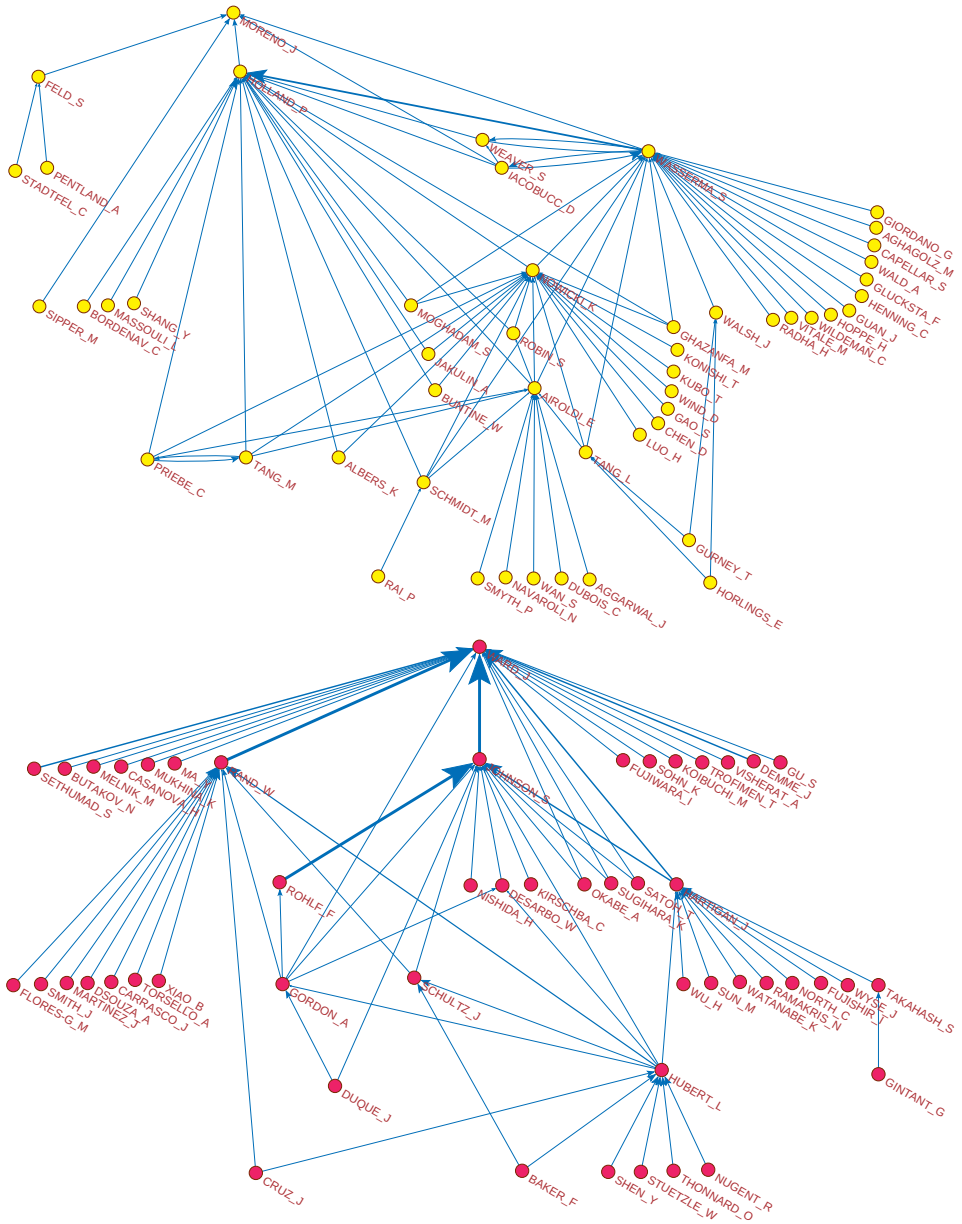



Figure 3.9: Subnetworks Wasserman and Ward

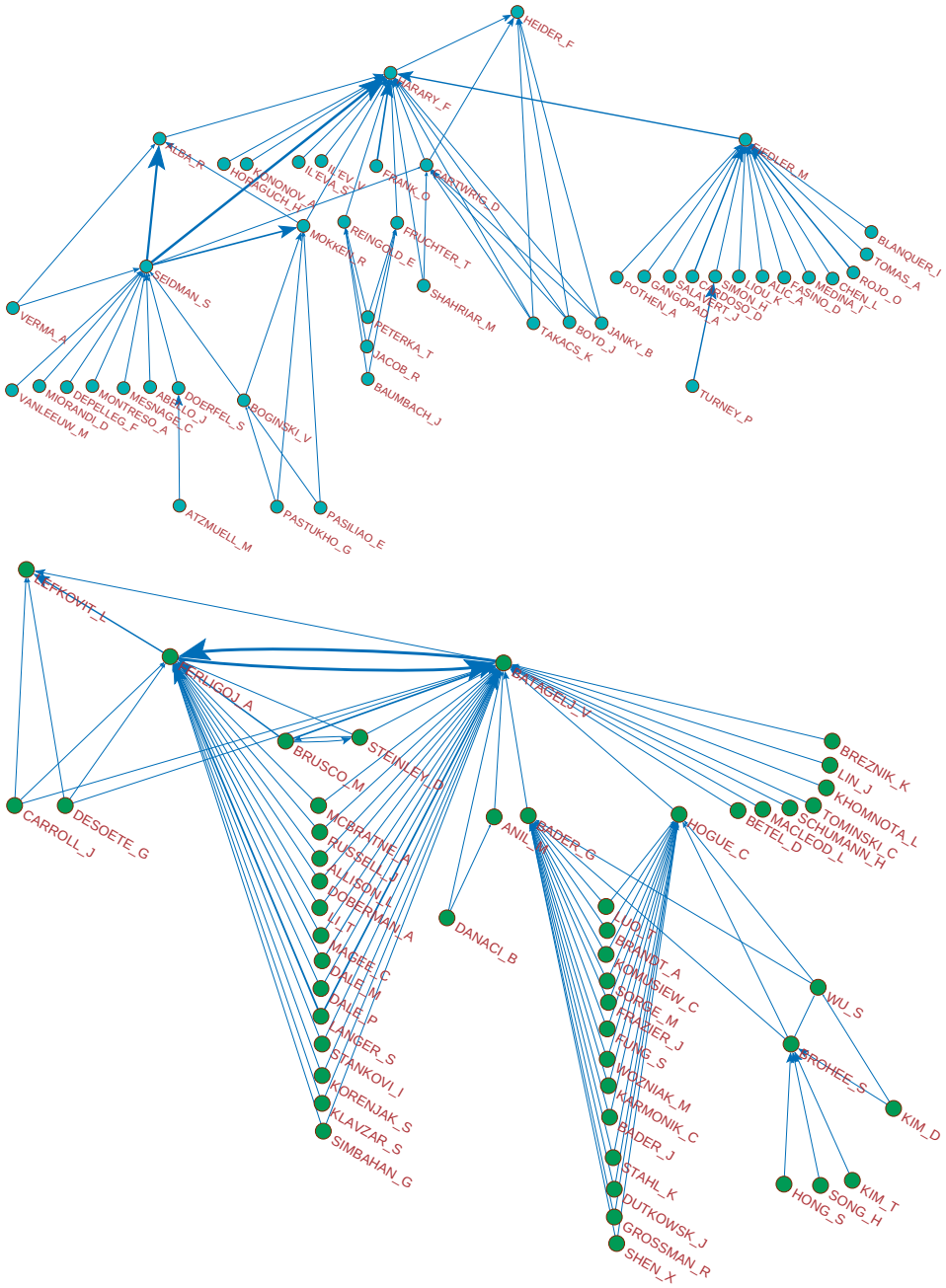



Figure 3.10: Subnetworks Harary and Batagelj + Ferligoj

in the field: Albert R + Barabási A, Bergstrom C + Rosvall M, Bezdek J, Blei D, Blondel V, Bonacich P + Kleinberg J, Breiger R, Burt R + Doreian P, Chung F + von Luxburg U, Clauset A, Dietrich J + Maede B, Fortunato S, Freeman L, Ghosh J, Girvan M, Goldberg D, Jaccard P, Jain A, Johnson D, Jordan M, Kaufman L, Knuth D, Leskovec J, Mac Queen J, Newman M, Newmark N, Okada Y, Palla G + Viscek T, Prescott W, Schaeffer S, Scott J, Sporus O, Stein C, Strehl A, Strogatz S, Van Dongen S, and some "cliques" of co-authors with attachments. We visually selected 12 clusters (Adamic L, Batagelj V + Ferligoj A, Bollobas B, Burt R + Doreian P, Faust K + Watts D, Fiedler M + Harary F, Granovetter M, Mizruchi M, Murtagh F, Nowicki K + Wasserman S, Robins G, Ward J, White H + Zachary W) with more interesting network structure for detailed inspection.

Most of the subnetworks of clusters for the Leader strategy have almost acyclic structure. This has to be considered also in their visualization. Because of the limited space we present here only subnetworks induced by four among the selected clusters.

The central author in the first selected subnetwork is Wasserman S. He forms a strong component with Iacobucci D and Weaver S. The leader of this subnetwork is the founder of SNA (sociometrics) Moreno J [47]. Other important authors are Holland P (with Leinhardt S the "father" of statistical approaches to SNA), Nowicki K and Airoldi E. The subnetwork is about statistical modeling of networks.

One of the most often used clustering methods is the Ward's method [55]. Ward J is the leader of the second subnetwork. It contains also other founders of clustering methods Johnson S and Rohlf F, authors of fundamental books Hartigan J [34] and Gordon A [31], and a theoretician Hubert L. The subnetwork is about cluster analysis.

The central author in the third subnetwork is Harary F, the author of the fundamental book on graph theory [33]. He is accompanied with other founders of graph-theoretic approches to network analysis: Heider S (signed networks), Alba (cliques), Cartwright (structure of directed networks), Seidman S (cores) and Fiedler (eigen values/vectors).

Central to the fourth subnetwork is a strong component Batagelj V and Ferligoj A. They are citing the leader Lefkovitz L. It contains also a strong component of authors Brusco M and Stainley D, working on efficient implementations of clustering algorithms, and several authors citing the paper [3] of Bader G and Hogue C describing the MCODE algorithm. The subnetwork is primarily about clustering with relational constraint.

## 3.8 Conclusion

In this chapter an attempt is made to present the "classical" approaches and results on clustering problem and show ways how to adapt them for clustering of/in networks. Most of the chapters in this monograph are essentially proposing different clustering criterion functions and some of them also new methods for obtaining the solutions. As already mentioned, most of criterion functions are based on structural equivalence. One of the challenges for future research is to develop efficient algorithms for other types of equivalences for large networks.

## Bibliography


1. E. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

2. M. R. Anderberg. *Cluster Analysis for Application*. Academic Press, New York, 1973.

3. G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(2):1–27, 2003.

4. V. Batagelj. Note on ultrametric hierarchical clustering algorithms. *Psychometrika*, 46(3):351—-352, 1981.

5. V. Batagelj. Generalized Ward and related clustering problems. In H. Bock, editor, *Classification and Related Methods of Data Analysis*, pages 67–74. North-Holland, Amsterdam, 1988.

6. V. Batagelj. Similarity measures between structured objects. In A. Graovac, editor, *MATH/CHEM/COMP 1988: proceedings of an International Course and Conference on the Interfaces between Mathematics, Chemistry, and Computer Science, Dubrovnik, Yugoslavia, 20-25 June 1988*, Studies in physical and theoretical chemistry, pages 25–40. Elsevier, 1989.

7. V. Batagelj. ***clurc*** – R package for clustering with relational constraint. 2017. URL `https://github.com/bavla/cluRC`.

8. V. Batagelj and M. Bren. Comparing resemblance measures. *Journal of Classification*, 12:73–90, 1995.

9. V. Batagelj and A. Ferligoj. Clustering relational data. In W. Gaul, O. Opitz, and M. Schader, editors, *Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 3–15. Springer, Berlin, Heidelberg, 2000.

10. V. Batagelj, S. Korenjak-Černe, and S. Klavžar. Dynamic programming and convex clustering. *Algorithmica*, 11(2):93–103, 1994.

11. V. Batagelj, P. Doreian, A. Ferligoj, and N. Kejžar. *Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution*. Wiley Series in Computational and Quantitative Social Science Series. Wiley, 2014.

12. V. Batagelj, N. Kejžar, and S. Korenjak-Černe. Clustering of modal valued symbolic data. *arXiv preprint arXiv:1507.06683*, 2015.

13. J. Benzécri and L. Bellier. *L'analyse des données: La Taxinomie*, volume 1 of *L'analyse des données*. Dunod, 1973.

14. L. Billard and E. Diday. *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. Wiley Series in Computational Statistics. Wiley, 2012.

15. H.-H. Bock. A history of the international federation of classification societies. 2006. URL `https://ifcs.boku.ac.at/site/lib/exe/fetch.php?media=pdfs:ifcs_history.pdf`. This is a slightly modified, translated and updated version of Chapter 9 of the book H.-H. Bock, P. Ihm (eds.): 25 Jahre Gesellschaft für

Klassifikation: Klassifikation und Datenanalyse im Wandel der Zeit. Shaker Verlag, Aachen 2001, 184 pp., ISBN 3-8265-9778-8.

16. J. Bodlaj and V. Batagelj. Hierarchical link clustering algorithm in networks. *Physical Review E*, 91(6):062814, 2015.

17. P. Brucker. On the complexity of clustering problems. In R. Henn, B. Korte, and W. Oettli, editors, *Optimization and Operations Research*, volume 157 of *Lecture Notes in Economics and Mathematical Systems*, pages 45–54. Springer, Berlin, Heidelberg, 1978.

18. M. Bruynooghe. Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. *Statistique et Analyse des Données*, (3):24–42, 1977.

19. D. Cartwright and F. Harary. Structural balance: A generalization of Heider's theory. *Psychological Review*, 63:277–293, 1956.

20. T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction To Algorithms*. MIT Press, Cambridge, 2 edition, 2001.

21. M. M. Deza and E. Deza. *Encyclopedia of distances*. Springer, 2009.

22. E. Diday. *Optimisation en classification automatique, Tome 1., 2.* INRIA, Rocquencourt, 1979. (in French).

23. J. Dieudonné. *Foundations of modern analysis*. Academic Press, New York, 1960.

24. P. Doreian, V. Batagelj, and A. Ferligoj. *Generalized Blockmodeling*. Cambridge University Press, Cambridge, 2005.

25. A.-H. Esfahanian. On the evolution of connectivity algorithms. In L. W. Beineke and R. J. Wilson, editors, *Topics in structural graph theory*, volume 147 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, New York, 2013.

26. A. Ferligoj and V. Batagelj. Clustering with relational constraint. *Psychometrika*, 47 (4):413–426, 1982.

27. A. Ferligoj and V. Batagelj. Some types of clustering with relational constraints. *Psychometrika*, 48(4):541–552, 1983.

28. A. Ferligoj and V. Batagelj. Direct multicriteria clustering algorithms. *Journal of Classification*, 9:43–61, 1992.

29. G. Gan, C. Ma, and J. Wu. *Data Clustering – Theory, Algorithms, and Applications*. SIAM, Philadelphia, 2007.

30. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1979.

31. A. D. Gordon. *Classification, 2nd Edition*, volume 82 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, Boca Raton, 1999. ISBN 978-1584880134.

32. J. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27:857–874, 1971.

33. F. Harary. *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

34. J. A. Hartigan. *Clustering algorithms*. Wiley-Interscience, New York, 1975.

35. C. Hayashi. Chikio Hayashi and Data Science – What is data science? *Student*, 2(1): 44–51, 1997.

36. A. Jain and R. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988.

37. N. Jardine, P. Jardine, and R. Sibson. *Mathematical Taxonomy*. Wiley series in probability and mathematical statistics. Wiley, 1971.

38. S. Joly and G. L. Calve. Étude des puissances d'une distance. *Statistique et analyse des données*, 11(3):30–50, 1986.

39. S. D. Kamvar, D. Klein, and C. D. Manning. Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 283–290, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

40. R. Kashyap and B. Oommen. A common basis for similarity measures involving two strings. *International Journal of Computer Mathematics*, 13(1):17–40, 1983.

41. L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. A Wiley-Interscience publication. Wiley, 1990.

42. D. Knuth. *The Stanford GraphBase, A Platform for Combinatorial Computing*. ACM Press, New York, 1993.

43. S. Korenjak-Černe, N. Kejžar, and V. Batagelj. A weighted clustering of population pyramids for the world's countries, 1996, 2001, 2006. *Population studies*, 69(1):105–120, 2015.

44. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965. English translation in Soviet Physics Doklady, 10(8):707-710, 1966.

45. F. H. C. Marriott. Optimization methods of cluster analysis. *Biometrika*, 69(2):417–421, 1982.

46. D. W. Matula. Graph theoretic techniques for cluster analysis algorithms. In J. Van Ryzin, editor, *Classification and clustering: proceedings of an advanced seminar conducted by the Mathematics Research Center, the University of Wisconsin-Madison, May 3-5, 1976*, pages 95–130. Academic Press, 1977.

47. J. Moreno. *Who shall survive? : a new approach to the problem of Human Interrelations*, volume 58 of *Nervous and mental disease monograph series*. Nervous and Mental Disease Publ., Washington, 1934.

48. F. Murtagh. *Multidimensional clustering algorithms*, volume 4. Physika Verlag, Vienna, 1985.

49. W. D. Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek, 3rd edition*. Cambridge University Press, New York, NY, USA, 2018.

50. N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

51. F. S. Roberts. *Discrete mathematical models, with applications to social, biological, and environmental problems*. Prentice-Hall, Englewood Cliffs, N.J., 1976.

52. R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. Books in biology. W. H. Freeman, 1963.

53. H. Späth. *Cluster-Analyse-Algorithmen: zur Objektklassifizierung und Datenreduktion*. Datenverarbeitung: Oldenbourg. Oldenbourg R. Verlag GmbH, 1977.

54. R. Todeschini and V. Consonni. *Molecular Descriptors for Chemoinformatics*. John Wiley & Sons, New York, 2. edition, 2009.

55. J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

56. Wikipedia. Mahalanobis distance. 2018. URL `https://en.wikipedia.org/wiki/Mahalanobis_distance`.