Optimizacijske metode

9. razvrščanje v skupine

Vladimir Batagelj

Univerza v Ljubljani

FMF, matematika

na vrhu

različica: 31. maj 2013 / 09 : 34

Kazalo

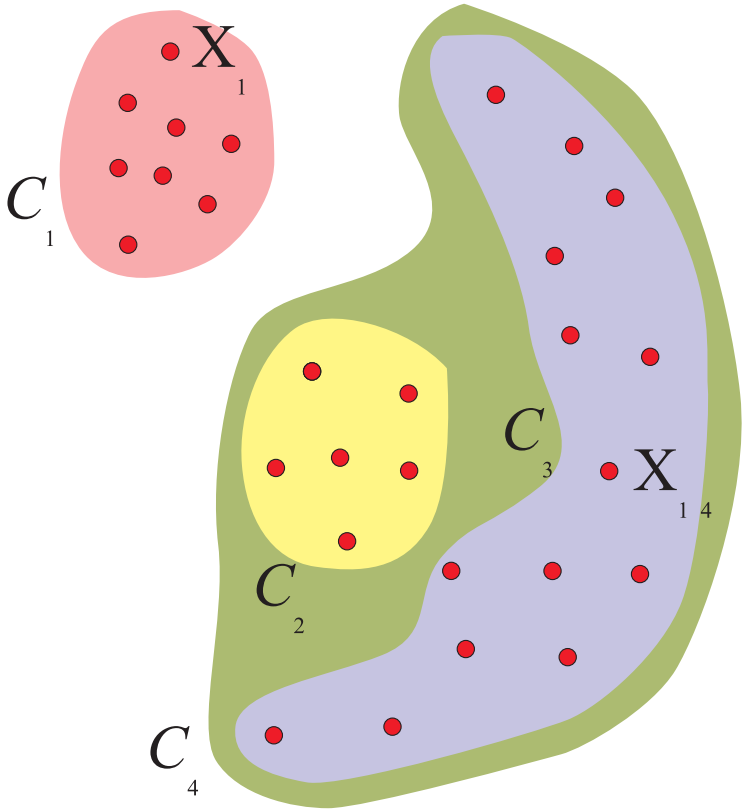
Clustering

- basic notions: cluster, clustering, feasible clustering, criterion function, dissimilarities, clustering as an optimization problem
- different (nonstandard) problems: assignment of students to classes, regionalization; general criterion function; multicriteria problems.
- complexity results about the clustering problem – NP-hardness theorems

Basic notions

Let us start with the formal setting of the clustering problem. We shall use the following notation:

- X – *unit*
- X – *description* of unit X
- $\mathcal{U}$ – *space* of units
- $\mathbf{U}$ – finite *set of units*, $\mathbf{U} \subset \mathcal{U}$
- C – *cluster*, $\emptyset \subset C \subseteq \mathbf{U}$
- $\mathbf{C}$ – *clustering*, $\mathbf{C} = \{C_i\}$
- Φ – set of *feasible clusterings*
- P – *criterion function*, $P : \Phi \rightarrow \mathbb{R}_0^+$



Clustering problem

With these notions we can express the *clustering problem* (Φ, P) 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 $\min(\Phi, P)$.

Units

real or imaginary objects of analysis

WORLD		UNITS		DESCRIPTIONS
$\{\mathrm{X}\}$	$\longleftrightarrow$	X	$\longleftrightarrow$	$[\mathrm{X}]$
	formalization		operationalization	
{ produced cars T }		car T		[seats=4, max-speed= ...]

Usually an unit X is represented by a vector/description $X \equiv [\mathrm{X}] = [x_1, x_2, ..., x_m]$ from the set $[\mathcal{U}]$ of all possible descriptions. $x_i = V_i(\mathrm{X})$ is the value of the i-th of selected properties or *variables* on X. Variables can be measured in different *scale*s: nominal, ordinal, interval, rational, absolute (Roberts, 1976).

There exist other kinds of descriptions of units: symbolic object (Bock, Diday, 2000), list of keywords from a text, chemical formula, vertex in a given graph, digital picture, ...

Clusterings

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 $\mathbf{U}$

$$\bigcup_{i=1}^{k} C_i = \mathbf{U}$$

$$i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

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 classes (clusters) by $P_k(\mathbf{U})$.

Simple criterion functions

Joining the individual units into a cluster C we make a certain "error", we create certain "tension"among them – we denote this quantity by $p(C)$. The *criterion function* $P(\mathbf{C})$ combines these "partial/local errors"into a "global error".

Usually it takes the form:

$$\text{S.} \qquad P(\mathbf{C}) = \sum_{C \in \mathbf{C}} p(C)$$

or

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

which can be unified and generalized in the following way:

Let $(\mathbb{R}, \oplus, e, \leq)$ be an ordered abelian monoid then:

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

For simple criterion functions usually $\min(P_{k+1}\mathbf{U}), P) \leq \min(P_k(\mathbf{U}), P)$ — we fix the value of k and set $\Phi \subseteq P_k(\mathbf{U})$.

Cluster-error function / dissimilarities

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.

To express the cluster-error $p(C)$ we define on the space of units a *dissimilarity* $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}_0^+$ for which we require D1 and D2:

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

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

Usually the dissimilarity d is defined using another dissimilarity $\delta : [\mathcal{U}] \times [\mathcal{U}] \rightarrow \mathbb{R}_0^+$ as

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

Properties of dissimilarities

The dissimilarity d is:

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

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

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

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

D7. *additive*, iff the Buneman's or four-point condition holds $\forall \mathrm{X}, \mathrm{Y}, \mathrm{U}, \mathrm{V} \in \mathcal{U}$:

$$d(\mathrm{X}, \mathrm{Y}) + d(\mathrm{U}, \mathrm{V}) \leq \max(d(\mathrm{X}, \mathrm{U}) + d(\mathrm{Y}, \mathrm{V}), d(\mathrm{X}, \mathrm{V}) + d(\mathrm{Y}, \mathrm{U}))$$

The dissimilarity d is a *distance* iff D4, D5 hold.

Since the description $[\,] : \mathbf{U} \to [\mathbf{U}]$ need not to be injective, d can be indefinite.

Dissimilarities on $\mathbb{R}^m$ / examples 1

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	$\sum_{i=1}^{m}\|x_i - y_i\|$	$[0, \infty)$	$M(1)$
4	rook	$\max_{i=1}^{m}\|x_i - y_i\|$	$[0, \infty)$	$M(\infty)$
5	Minkowski	$\sqrt[p]{\sum_{i=1}^{m}(x_i - y_i)^p}$	$[0, \infty)$	$M(p)$

Dissimilarities on $\mathbb{R}^m$ / examples 2

n	measure	definition	range	note
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.	$\frac{\mathrm{cov}(X, Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}}$	$[1, -1]$	

(Dis)similarities on $\mathbb{B}^m$ / examples

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

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 transform a similarity s from $[1, 0]$ into dissimilarity d on $[0, 1]$ by $d = 1 - s$.

For details see Batagelj, Bren (1995).

(Dis)similarities on $\mathbb{B}^m$ / examples 1

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	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	$[1,0]$

(Dis)similarities on $\mathbb{B}^m$ / examples 2

n	measure	definition	range
9	Q_0	$\frac{bc}{ad}$	$[0,\infty]$
10	Yule (1927), Q	$\frac{ad-bc}{ad+bc}$	$[1,-1]$
11	Pearson, ϕ	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	$[1,-1]$
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)	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$	$[1,-1]$

Dissimilarities between sets

Let $\mathcal{F}$ be a finite family of subsets of the finite set U; $A, B \in \mathcal{F}$ and let $A \oplus B = (A \setminus B) \cup (B \setminus A)$ denotes the symmetric difference between A and B.

The 'standard' dissimilarity between sets is the *Hamming distance*:

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

Usually we normalize it $d_h(A, B) = \frac{1}{M}\,\mathrm{card}(A \oplus B)$. One normalization is $M = \mathrm{card}(U)$; the other $M = m_1 + m_2$, where m_1 and m_2 are the first and the second largest value in $\{\mathrm{card}(X) : X \in \mathcal{F}\}$.

Other dissimilarities

$$d_s(A, B) = \frac{\mathrm{card}(A \oplus B)}{\mathrm{card}(A) + \mathrm{card}(B)} \qquad d_u(A, B) = \frac{\mathrm{card}(A \oplus B)}{\mathrm{card}(A \cup B)}$$

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

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

Problems with dissimilarities

What to do in the case of *mixed units* (with variables measured in different types of scales)?

- conversion to a common scale
- compute the dissimilarities on homogeneous parts and combine them (Gower's dissimilarity)

Fairness of dissimilarity – all variables contribute equally. Approaches: use of normalized variables, analysis of dependencies among variables.

Cluster-error function / examples

Now we can define several 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 $w : \mathcal{U} \to \mathbb{R}^+$ is a *weight* of units, which is extended to clusters by:

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

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

Often $w(\mathrm{X}) = 1$ holds for each $\mathrm{X}\in\mathcal{U}$. 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 (partial) criterion functions to denote *types* of criterion functions. For example:

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

It is easy to prove:

Proposition 1.1 *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* $\mathbf{C} \in P_k(\mathbf{U})$ *:*

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

holds.

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

Sensitive criterion functions

The criterion function $P(\mathbf{C})$, based on the 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 α-*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 1.2 *Every α-sensitive criterion function is also sensitive.*

The proposition 1.1 can be reexpressed as:

Proposition 1.3 *The criterion functions* $\mathrm{SS}, \mathrm{S\overline{S}}, \mathrm{SM}, \mathrm{MS}, \mathrm{M\overline{S}}, \mathrm{MM}$ *are α-sensitive.*

Representatives

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

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

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

Example 1 The representation space need not be the same as the description space. $[\mathbf{U}] \subseteq \mathbb{R}^2$ and $[\mathbf{F}] = \{(a, b, c) : ax + by = c, a^2 + b^2 = 1\}$. □

Example 2 In the case $[\mathbf{U}] \subseteq \mathbb{R}^m$, $[\mathbf{F}] = \mathbb{R}^m$, $d(\mathrm{X}, \mathrm{L}) = d_2^2(\mathrm{X}, \mathrm{L}) = \sum_{i=1}^{m} (x_i - l_i)^2$ there exists a uniquely determined representative – *center of gravity* $\overline{C} = \frac{1}{\mathrm{card}(C)} \sum_{\mathrm{X} \in C} X$. In this case the criterion function SR is called *Ward's criterion function* (Ward, 1963). □

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 $\mathcal{U}$.

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 Batagelj (1988) it is also shown how to replace $\overline{C}$ by a generalized, possibly imaginary (with descriptions not neccessary in the same set as $\mathcal{U}$), central element in the way to preserve the properties characteristic for Ward's clustering problem.

Representatives cluster error

Proposition 1.4 *Let* $p(C)$ *be of type* R *then*

$$\begin{aligned} &a) \quad p(C) + w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C \cup \mathrm{X}}) \leq p(C \cup \mathrm{X}), &\mathrm{X} \notin C \\ &b) \quad p(C \setminus \mathrm{X}) + w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C}) \leq p(C), &\mathrm{X} \in C \end{aligned}$$

Proof: The definition of $\overline{C}$ can be equivalently expressed in the form:

$$\forall \mathrm{L} \in F : p(C) = \sum_{\mathrm{Y} \in C} w(\mathrm{Y}) \cdot d(\mathrm{Y}, \overline{C}) \leq \sum_{\mathrm{Y} \in C} w(\mathrm{Y}) \cdot d(\mathrm{Y}, \mathrm{L})$$

Therefore in case a):

$$\begin{aligned} p(C) &= \sum_{\mathrm{Y} \in C} w(\mathrm{Y}) \cdot d(\mathrm{Y}, \overline{C}) \leq \sum_{\mathrm{Y} \in C} w(\mathrm{Y}) \cdot d(\mathrm{Y}, \overline{C \cup \mathrm{X}}) = \\ &= \sum_{\mathrm{Y} \in C \cup \mathrm{X}} w(\mathrm{Y}) \cdot d(\mathrm{Y}, \overline{C \cup \mathrm{X}}) \; - w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C \cup \mathrm{X}}) = \\ &= p(C \cup \mathrm{X}) - w(\mathrm{X}) \cdot d(\mathrm{X}, \overline{C \cup \mathrm{X}}) \end{aligned}$$

In the similar way we can prove also inequality b). □

Other criterion functions

Several other types of criterion functions were proposed in the literature. A very important class among them are the "*statistical*čriterion functions based on the assumption that the units are sampled from a mixture of multivariate normal distributions (Marriott, 1982) .

General criterion function

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 shall use it in blockmodeling.

Multicriteria clustering

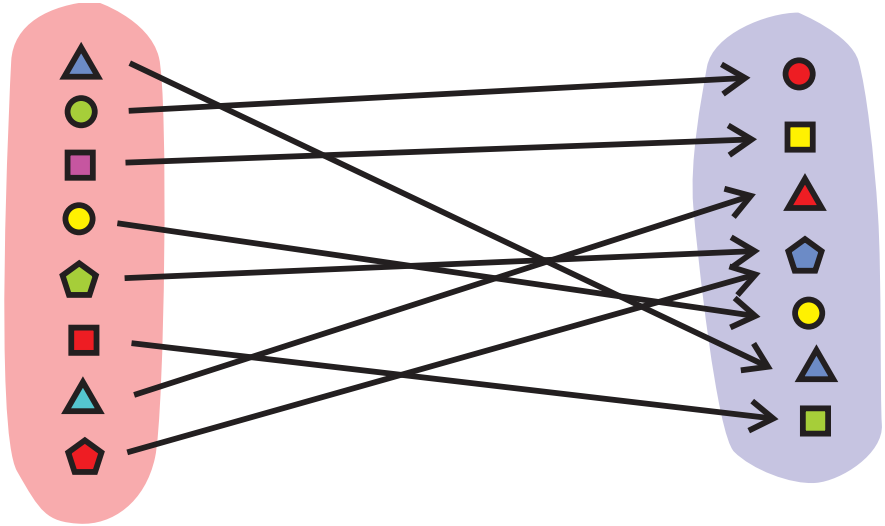
In some problems several criterion functions can be defined $(\Phi, P_1, P_2, \ldots, P_s)$. See Ferligoj, Batagelj (1994).

Example: 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.



Example: Regionalization

The motivation comes from *regionalization* problem: partition given set of territorial units into k connected subgroups of similar units – regions.

Suppose that besides the descriptions of units $[\mathbf{U}]$ they are related also by a binary *relation* $R \subseteq \mathbf{U} \times \mathbf{U}$.

In such a case we have an additional requirement – *relational constraint* on clusterings to be feasible. The set of feasible clusterings can be defined as:

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

If R is nonsymmetric we can define different types of sets of feasible clusterings for the same relation (Ferligoj and Batagelj, 1983).

Complexity of the clustering problem

Because the set of feasible clusterings Φ is finite the clustering problem (Φ, 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. And to get an impression:

$$\begin{aligned} S(20,8) &= 151709326626 79 \\ S(30,11) &= 2150471015606668766 19690 \\ S(n,2) &= 2^{n-1} - 1 \end{aligned}$$

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 (Garey and Johnson, 1979) .

Complexity results

Although there are some polynomial types of clustering problems, for example (P_2, MM) and (P_k, ST), it seems that they are mainly NP-hard.

Brücker (1978) showed that ($\propto$ denotes the polynomial reducibility of problems) :

Theorem 1.5 *Let the criterion function*

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

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

Proof: Select a value P^* such that $P^* > \max_{\mathbf{C}\in P_k(\mathbf{U})} P(\mathbf{C})$, extend, $\mathbf{U}' = \mathbf{U} \cup \{\mathrm{X}^\bullet\}$, the set of units with a new unit $\mathrm{X}^\bullet$, and define the dissimilarities between it and the 'old' units such that $d(\mathrm{X}, \mathrm{X}^\bullet) > P^*/\alpha'$, for $\mathrm{X} \in \mathbf{U}$ and $\alpha' = \alpha^P_{k+1}(\mathbf{U}')$. We get a new clustering problem $(P_{k+1}(\mathbf{U}'), P)$.

Consider a clustering $\mathbf{C}' \in P_{k+1}(\mathbf{U}')$. There are two possibilities:

a. $\mathrm{X}^\bullet$ forms its own cluster $\mathbf{C}' = \mathbf{C} \cup \{\{\mathrm{X}^\bullet\}\}$, $\mathbf{C} \in P_k(\mathbf{U})$. Then

$$P(\mathbf{C}') = P(\mathbf{C}) \oplus p(\{\mathrm{X}^\bullet\}) = P(\mathbf{C}) \leq \max_{\mathbf{C}\in P_k(\mathbf{U})} P(\mathbf{C}) < P^*$$

b. $\mathrm{X}^\bullet$ belongs to a cluster $C^\bullet$ with $\mathrm{card}(C^\bullet) \geq 2$. Then

$$P(\mathbf{C}') \geq \alpha' \cdot \max_{C\in\mathbf{C}'} \max_{\mathrm{X},\mathrm{Y}\in C} d(\mathrm{X}, \mathrm{Y}) \geq \alpha' \cdot \max_{\mathrm{X},\mathrm{Y}\in C^\bullet} d(\mathrm{X}, \mathrm{Y}) =$$

$$= \alpha' \cdot \max_{\mathrm{X}\in C^\bullet\setminus\{\mathrm{X}^\bullet\}} d(\mathrm{X}, \mathrm{X}^\bullet) > P^*$$

We see that all optimal solutions of the problem $(P_{k+1}(\mathbf{U}'), P)$ have the form **a**. Since in this case $P(\mathbf{C}') = P(\mathbf{C})$

$$\mathbf{C}' \in \mathrm{Min}(P_{k+1}(\mathbf{U}'), P) \Leftrightarrow \mathbf{C} \in \mathrm{Min}(P_k(\mathbf{U}), P)$$

□

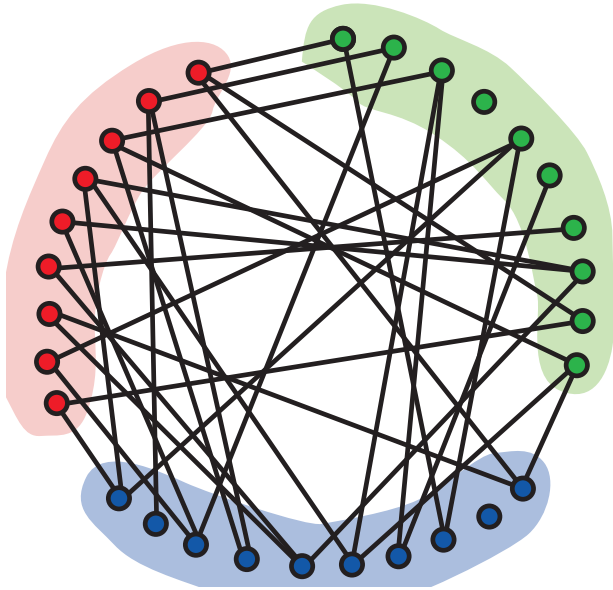
Complexity results 1

Theorem 1.6 *Let the criterion function P be sensitive then*

$$3 - \mathrm{COLOR} \propto (P_3, P)$$

Proof: Let $G = (V, E)$ be a simple undirected graph. We assign to it a clustering problem $(P_3(V), P)$ as follows. We define a dissimilarity d (on which P is based) by

$$d(u,v) = \begin{cases} 1 & (u:v) \in E \\ 0 & (u:v) \notin E \end{cases}$$



Since P is sensitive it holds: the graph G is 3-colorable **iff** $\min(P_3(V), P) = 0$.
Let $\mathbf{C} = \{C_1, C_2, C_3\}$ then $P(\mathbf{C}) = 0$ **iff** $c : V \to \{1,2,3\} : (c(v) = i \Leftrightarrow v \in C_i)$ is a vertex coloring.

□

Complexity results 2

Polynomial	NP-hard	note
(P_2, MM)	(P_3, MM)	Theorem 1.6
	(P_3, SM)	Theorem 1.6
	(P_2, SS)	MAX-CUT $\propto (P_2, \mathrm{SS})$
	$(P_2, \mathrm{S\overline{S}})$	$(P_2, \mathrm{SS}) \propto (P_2, \mathrm{S\overline{S}})$
	(P_2, MS)	PARTITION $\propto (P_2, \mathrm{MS})$
$(\mathbb{R}_2^m, \mathrm{S\overline{S}})$		
$(\mathbb{R}_k^1, \mathrm{S\overline{S}})$		
$(\mathbb{R}_k^1, \mathrm{SM})$		
$(\mathbb{R}_k^1, \mathrm{MM})$		

Note that, by the Theorem 1.5, (P_k, MM), $k > 3$ are also NP-hard ...

Consequences

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.

The most important types of these procedures are:

- local optimization
- hierarchical (agglomerative, divisive and adding)
- leaders and the dynamic clusters method
- graph theory methods

Approaches to Clustering

- local optimization
- dynamic programming
- hierarchical methods; agglomerative methods; Lance-Williams formula; dendrogram; inversions; adding methods
- leaders and the dynamic clusters method
- graph theory (next, 3. lecture);

Local optimization

Often for a given optimization problem (Φ, P) there exist rules which relate to each element of the set Φ some elements of Φ. 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 Φ.
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* (Φ, 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, $\forall \mathrm{X} \in \Phi : \mathrm{X}S\mathrm{X}$.

Local optimization

They are the basis of the *local optimization procedure*

select 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 element of $\mathrm{X}_0 \in \Phi$ repeats moving to an element determined by local transformation which has better value of the criterion function until no such element exists.

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 from one cluster to another

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

- *transposition*: clustering $\mathbf{C}'$ is obtained from $\mathbf{C}$ by interchanging two units 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.

Hints

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 $\mathbf{\Phi}$; otherwise, we can transform using *penalty function method* the problem to an equivalent nonconstrained problem (P_k, Q) 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, ... (Aarts and Lenstra, 1997).

The *initial clustering* $\mathbf{C}_0$ can be given; most often we generate it randomly.
Let $c[s] = u \Leftrightarrow \mathrm{X}_s \in C_u$. Fill the vector c with the desired number of units in each cluster and shuffle it:

for $p := n$ **downto** 2 **do begin** $q := random(1, p)$; $swap(c[p], c[q])$ **end**;

Quick scanning of neighbors

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

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

Δ for the generalized Ward's method

Let us analyze the transition of a unit 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 (Späth, 1977).

Dynamic programming

Suppose that $\mathrm{Min}(\mathbf{\Phi}_k, P) \neq \emptyset$, $k = 1, 2, \ldots$. Denoting $P^*(\mathbf{U}, k) = P(\mathbf{C}_k^*(\mathbf{U}))$ we can derive the generalized *Jensen equality* (Batagelj, Korenjak and Klavžar, 1994):

$$P^*(\mathbf{U}, k) = \begin{cases} p(\mathbf{U}) & \{\mathbf{U}\} \in \mathbf{\Phi}_1 \\ \min\limits_{\substack{\emptyset \subset C \subset \mathbf{U} \\ \exists \mathbf{C} \in \mathbf{\Phi}_{k-1}(\mathbf{U} \setminus C): \mathbf{C} \cup \{C\} \in \mathbf{\Phi}_k(\mathbf{U})}} (P^*(\mathbf{U} \setminus C, k-1) \oplus p(C)) & k > 1 \end{cases}$$

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

Hierarchical methods

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

In the set $\mathbf{\Phi}$ 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 $\mathbf{\Phi} \subseteq P(\mathbf{U})$ then $(\mathbf{\Phi}, \sqsubseteq)$ is a partially ordered set.

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

$\mathbf{C}_1 \sqsubset \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \mathbf{C}_1 \neq \mathbf{C}_2$ – strict inclusion, and

$\mathbf{C}_1 \sqsubset\!\cdot\, \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \neg\exists \mathbf{C} \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2)$ – predecessor.

Conditions on the structure of the set of feasible clusterings

We shall assume that the set of feasible clusterings $\mathbf{\Phi} \subseteq P(\mathbf{U})$ satisfies the following conditions:

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

F2. The feasibility predicate Φ is *local* – it has the form $\Phi(\mathbf{C}) = \bigwedge_{C \in \mathbf{C}} \varphi(C)$ where $\varphi(C)$ is a predicate defined on $\mathcal{P}(\mathbf{U}) \setminus \{\emptyset\}$ (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 $\mathbf{C} \in \mathbf{\Phi}$ consists of 'good' clusters.

F3. The predicate Φ 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.

. . . conditions

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

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

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

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

These conditions provide a framework in which the hierarchical methods can be applied also for constrained clustering problems $\mathbf{\Phi}_k(\mathbf{U}) \subset P_k(\mathbf{U})$.

In the ordinary problem both predicates $\varphi(C)$ and $\psi(C_p, C_q)$ are always true – all conditions F1-F5 are satisfied.

Criterion functions compatible with a dissimilarity between clusters

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

Let $(\mathbb{R}_0^+, \oplus, 0, \leq)$ 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 $\mathbf{\Phi}$ iff for all $C \subseteq \mathbf{U}$ holds:

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

Theorem 1.7 *A* 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*

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

Greedy approximation

Theorem 1.8 *Let P be compatible with D over $\mathbf{\Phi}$, $\oplus$ distributes over* min*, and F1 – F5 hold, then*

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

The equality from theorem 2.1 can also be written in the form

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

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 following agglomerative (binary) procedure for solving the clustering problem.

Agglomerative methods

1. $k := n$; $\mathbf{C}(k) := \{\{\mathrm{X}\} : \mathrm{X} \in \mathbf{U}\}$;
2. **while** $\exists C_i, C_j \in \mathbf{C}(k)$: $(i \neq j \wedge \psi(C_i, C_j))$ **repeat**
2.1. $(C_p, C_q) := \mathrm{argmin}\{D(C_i, C_j) : i \neq j \wedge \psi(C_i, C_j)\}$;
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 $C_s \in \mathbf{C}(k)$
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 Kamvar, Klein, Manning (2002).

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

Some dissimilarities between clusters

We shall use the generalized Ward's c.e.f.

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

and the notion of the *generalized center* $\overline{C}$ of the cluster C, for which the dissimilarity to any cluster or unit U is defined by

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

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

. . . some dissimilarities

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

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

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

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

Weighted increase of variance:

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

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

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

Lance-Williams-Jambu coefficients

method	α_1	α_2	β	γ	δ_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	–

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

Hierarchies

The agglomerative clustering procedure produces a series of feasible clusterings $\mathbf{C}(n)$, $\mathbf{C}(n-1), \ldots, \mathbf{C}(m)$ with $\mathbf{C}(m) \in \operatorname{Max} \mathbf{\Phi}$ (maximal elements for $\sqsubseteq$).

Their union $\mathcal{T} = \bigcup_{k=m}^{n} \mathbf{C}(k)$ is called a *hierarchy* and has the property

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

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

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

c1. $W \subseteq C_{\mathcal{T}}(W)$

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

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

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

Level functions

A mapping $h : \mathcal{T} \rightarrow \mathbb{R}_0^+$ is a *level function* on $\mathcal{T}$ 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 $\mathbf{U}$

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

The converse is also true (see Dieudonne (1960)): Let d be an ultrametric on $\mathbf{U}$. Denote $\overline{B}(\mathrm{X}, r) = \{\mathrm{Y} \in \mathbf{U} : d(\mathrm{X}, \mathrm{Y}) \leq r\}$. Then for any given set $A \subset \mathbb{R}^+$ the set

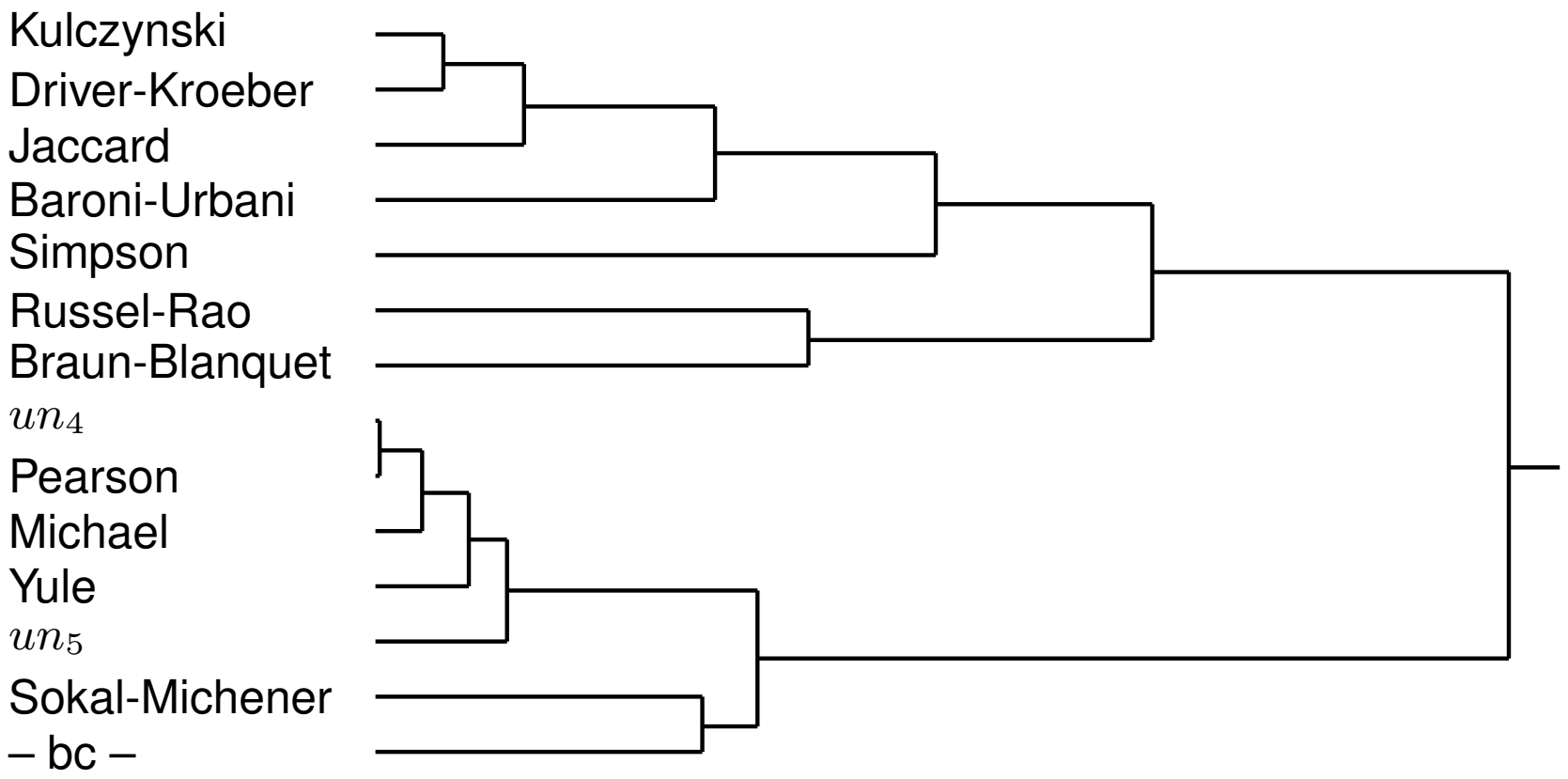
$$\mathbf{C}(A) = \{\overline{B}(\mathrm{X}, r) : \mathrm{X} \in \mathbf{U}, r \in A\} \cup \{\{\mathbf{U}\}\} \cup \{\{\mathrm{X}\} : \mathrm{X} \in \mathbf{U}\}$$

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

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

Association coefficients, Monte Carlo, $m = 15$

CLUSE – maximum $[0.00, 0.33]$



Inversions

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 Ds the *inversion*s, $D(C_p, C_q) > D(C_p \cup C_q, C_s)$, are possible.

Batagelj (1981) showed:

Theorem 1.9 *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 (Bruynooghe, 1977) iff

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

Theorem 1.10 *If a dissimilarity D has the reducibility property then h_D is a level function.*

Adding hierarchical methods

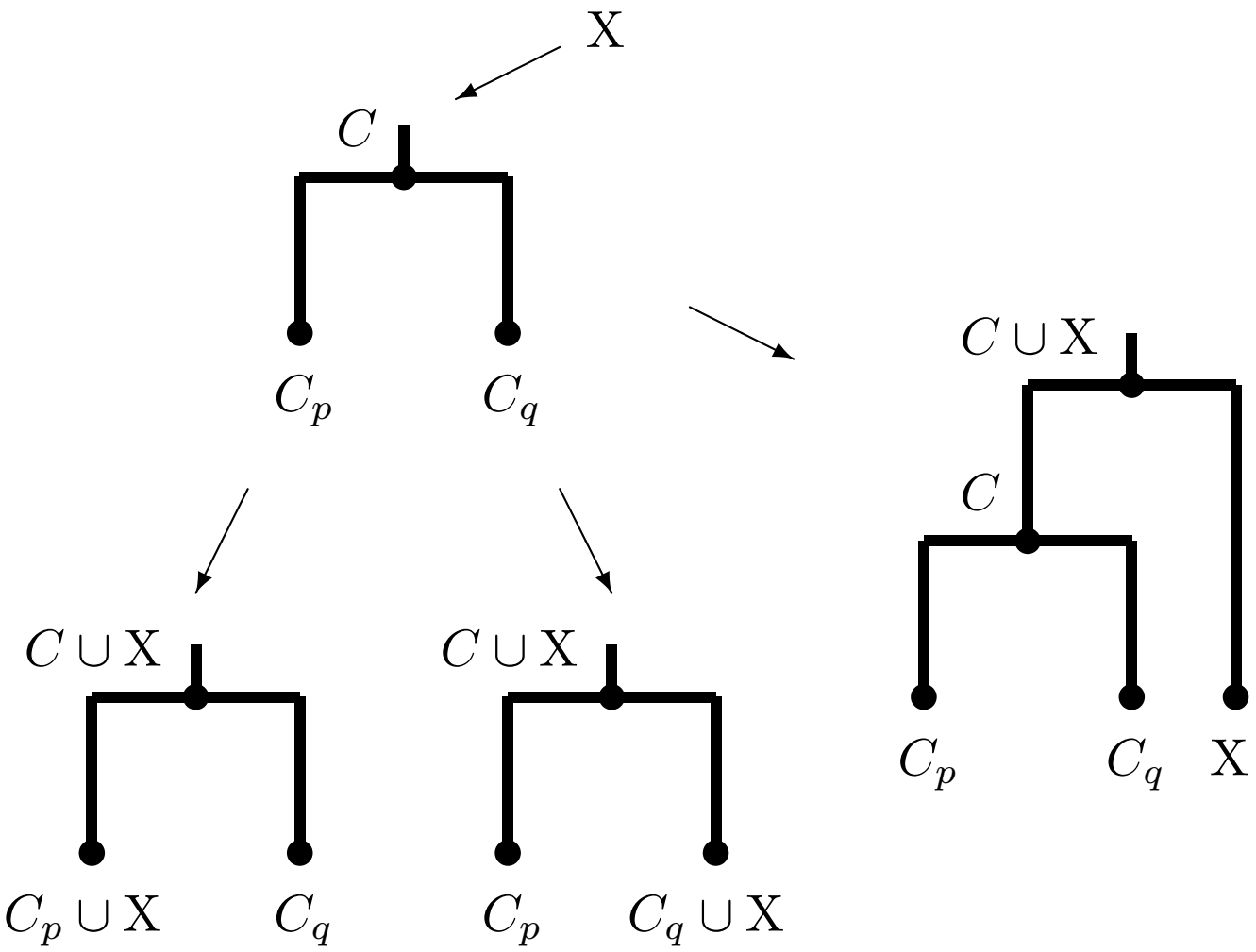
Suppose that we already built a clustering tree $\mathcal{T}$ over the set of units $\mathbf{U}$. To add a new unit X to the tree $\mathcal{T}$ we start in the root and branch down. Assume that we reached the node corresponding to cluster C, which was obtained by joining subclusters C_p and C_q. There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

Consider again the 'greedy approximation' $P(\mathbf{C}^{\bullet}_{k}) = P(\mathbf{C}^{\bullet}_{k+1}) + D(C_p, C_q)$ where $D(C_p, C_q) = \min_{C_u, C_v \in \mathbf{C}^{\bullet}_{k+1}} D(C_u, C_v)$ and $\mathbf{C}^{\bullet}_{i}$ 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 $D(C_p \cup \{X\}, C_q)$, $D(C_q \cup \{X\}, C_p)$ and $D(C_p \cup C_q, \{X\})$.

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.

... adding hierarchical methods



About the minimal solutions of (P_k, SR)

Theorem 1.11 *In the (locally with respect to transitions) minimal clustering for the problem* (P_k, SR)

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

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

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

Proof

Let $\mathbf{C}' = (\mathbf{C}^\bullet \setminus \{C_u, C_v\}) \cup \{C_u \setminus \{\mathrm{X}\}, C_v \cup \{\mathrm{X}\}\}$ be any clustering neighbouring with respect to transitions to the clustering $\mathbf{C}^\bullet$. From the theorem assumptions $P(\mathbf{C}^\bullet) \leq P(\mathbf{C}')$ and the type of criterion function we have:

$$p(C_u) + p(C_v) \leq p(C_u \setminus \mathrm{X}) + p(C_v \cup \mathrm{X})$$

and by proposition 1.4.b: $\leq p(C_u) - w(\mathrm{X}).d(\mathrm{X}, \overline{C}_u) + p(C_v \cup \mathrm{X})$.
Therefore $p(C_v) \leq p(C_v \cup \mathrm{X}) - w(\mathrm{X}).d(\mathrm{X}, \overline{C}_u)$, and

$$\begin{aligned} w(\mathrm{X}).d(\mathrm{X}, \overline{C}_u) &\leq p(C_v \cup \mathrm{X}) - p(C_v) = \\ &= p(C_v \cup \mathrm{X}) - (p(C_v) + w(\mathrm{X}).d(\mathrm{X}, \overline{C}_v)) + w(\mathrm{X}).d(\mathrm{X}, \overline{C}_v) \\ &= w(\mathrm{X}).d(\mathrm{X}, \overline{C}_v) + (p(C_v \cup \mathrm{X}) - \sum_{\mathrm{Y} \in C_v \cup \mathrm{X}} w(\mathrm{Y}).d(\mathrm{Y}, \overline{C}_v)) \end{aligned}$$

By the definition of cluster-error function of type R the second term in the last line is negative. Therefore

$$\leq w(\mathrm{X}).d(\mathrm{X}, \overline{C}_v)$$

Dividing by $w(\mathrm{X}) > 0$ we finally get

$$d(\mathrm{X}, \overline{C}_u) \leq d(\mathrm{X}, \overline{C}_v)$$

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, which is the basis of the ISODATA program (one among the most popular clustering programs) and several recent 'data-mining' methods. In the leaders method the criterion function has the form SR.

The basic scheme of leaders method is simple:

determine $\mathbf{C}_0; \mathbf{C} := \mathbf{C}_0;$
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

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

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

Basic scheme of the dynamic clusters method

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* can be described by the scheme:

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

Properties of DCM

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

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

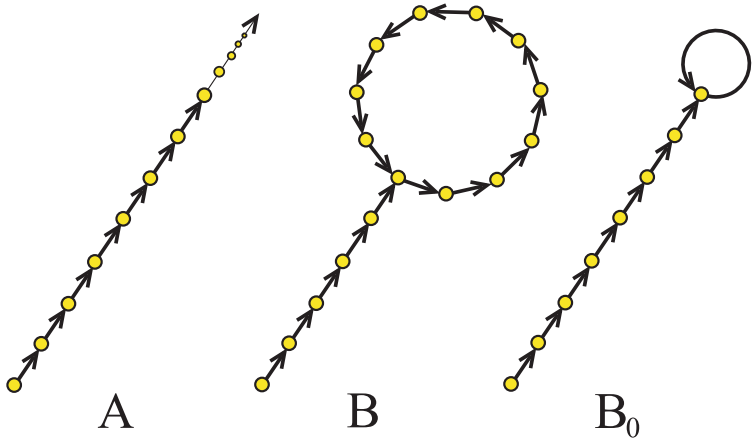
and the sequence of values of the extended criterion function $u_n = W(\mathbf{C}_n, \mathrm{L}_n)$. Let us also denote $u^* = P(\mathbf{C}^*)$. Then it holds:

Theorem 1.12 *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* $\forall n \geq m : u_n = u_m$.

The Theorem 2.6 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.

... types of of DCM sequences

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 $k = m + 1$



The DCM sequence (v_n) is of type B if

- sets Φ and Ψ 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 Φ are finite we expect from a good dynamic clusters procedure to stabilize in finite number of steps – is of type B.

Additional requirement

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 Ψ 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 Φ and Ψ 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.

Theorem 1.13 *If W3 holds and if there exists* $m \in \mathbb{N}$ *such that* $u_{m+1} = u_m$, *then also* $\mathrm{L}_{m+1} = \mathrm{L}_m$.

Simple clustering and representation functions

Usually, in the applications of the DCM, the clustering function takes the form $F : \Psi \rightarrow \Phi$. 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:

Theorem 1.14 *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 $G : \Phi \rightarrow \Psi$?

Theorem 1.15 *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* $\exists m \in \mathbb{N} \forall n \geq m : v_n = v_m$

Original DCM

In the original dynamic clusters method (Diday, 1979) both functions F and G are simple – $F : \Psi \rightarrow \Phi$ and $G : \Phi \rightarrow \Psi$.

We proved, 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.

Clustering and Networks

- clustering with relational constraint
- transforming data into graphs (neighbors)
- clustering of networks; dissimilarities between graphs (networks)
- clustering of vertices / links; dissimilarities between vertices
- clustering in large networks

Clustering with relational constraint

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

We want to cluster the units according to the similarity of their descriptions, but also considering the relation R – it imposes constraints on the set of feasible clusterings, usually in the following form:

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

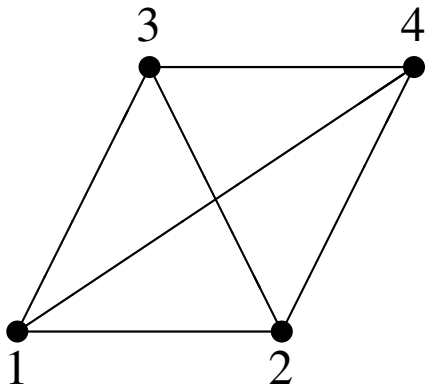
Some types of relational constraints

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

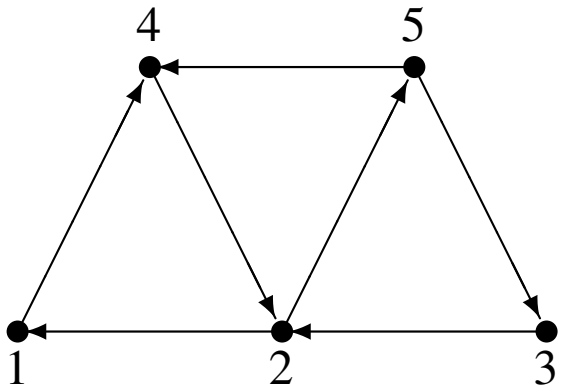
type of clusterings	type of connectedness
$\Phi^1(R)$	weakly connected units
$\Phi^2(R)$	weakly connected units that contain at most one center
$\Phi^3(R)$	strongly connected units
$\Phi^4(R)$	clique
$\Phi^5(R)$	the existence of a trail containing all the units of the cluster

A set of units $L \subseteq C$ is a *center* of cluster C in the clustering of type $\Phi^2(R)$ iff the subgraph induced by L is strongly connected and $R(L) \cap (C \setminus L) = \emptyset$.

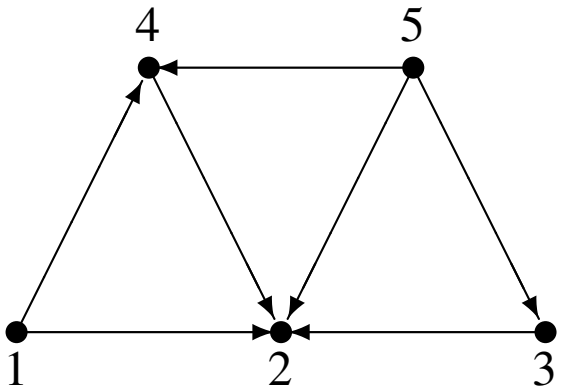
Some graphs of different types



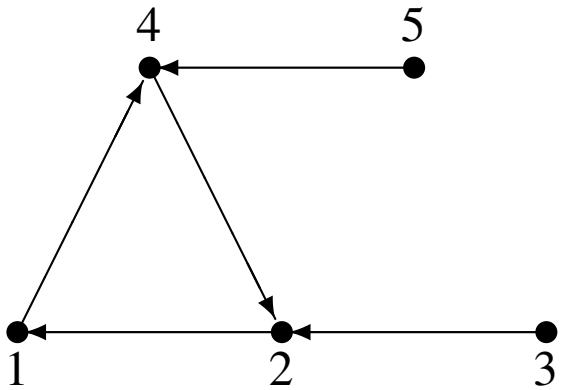
a clique



strongly connected units



weakly connected units



weakly connected units
with a center $\{1, 2, 4\}$

Properties of relational constraints

The sets of feasible clusterings $\Phi^i(R)$ are linked as follows:

$\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$

$\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$

If the relation R is symmetric, then $\Phi^3(R) = \Phi^1(R)$

If the relation R is an equivalence relation, then $\Phi^4(R) = \Phi^1(R)$

Here are also examples of the corresponding fusibility predicates:

$\psi^1(C_1, C_2) \equiv \exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : (\mathrm{X}R\mathrm{Y} \vee \mathrm{Y}R\mathrm{X})$

$\psi^2(C_1, C_2) \equiv (\exists \mathrm{X} \in L_1 \exists \mathrm{Y} \in C_2 : \mathrm{X}R\mathrm{Y}) \vee (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in L_2 : \mathrm{Y}R\mathrm{X})$

$\psi^3(C_1, C_2) \equiv (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : \mathrm{X}R\mathrm{Y}) \wedge (\exists \mathrm{X} \in C_1 \exists \mathrm{Y} \in C_2 : \mathrm{Y}R\mathrm{X})$

$\psi^4(C_1, C_2) \equiv \forall \mathrm{X} \in C_1 \forall \mathrm{Y} \in C_2 : (\mathrm{X}R\mathrm{Y} \wedge \mathrm{Y}R\mathrm{X})$

For ψ^3 the property F5 fails.

We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint (Ferligoj, Batagelj 1983).

Neighborhood Graphs

For a given dissimilarity d on the set of units $\mathbf{U}$ we can define several graphs:

The *k nearest neighbors graph* $\mathbf{G}_N(k) = (\mathbf{U}, A)$

$$(\mathrm{X}, \mathrm{Y}) \in A \Leftrightarrow \mathrm{Y} \text{ is among the } k \text{ closest neighbors of } \mathrm{X}$$

By setting for $a(\mathrm{X}, \mathrm{Y}) \in A$ its value to $w(a) = d(\mathrm{X}, \mathrm{Y})$ we obtain a network.
In the case of equidistant pairs of units we have to decide – or to include them all in the graph, or specify an additional selection rule.

A special case of the k nearest neighbors graph is the *nearest neighbor graph* $\mathbf{G}_N(1)$. We shall denote by $\mathbf{G}^*_{NN}$ the graph with included all equidistant pairs, and by $\mathbf{G}_{NN}$ a graph where a single nearest neighbor is always selected.

The *fixed-radius neighbors graph* $\mathbf{G}_B(r) = (\mathbf{U}, E)$

$$(\mathrm{X} : \mathrm{Y}) \in E \Leftrightarrow d(\mathrm{X}, \mathrm{Y}) \leq r$$

There are several papers on efficient algorithms for determining the neighborhood graphs (Fukunaga, Narendra (1975), Dickerson, Eppstein (1996), Chávez & (1999), Murtagh (1999)). These graphs are a bridge between data and network analysis.

Structure and properties of the nearest neighbor graphs

Let $\mathbf{N} = (\mathbf{U}, A, w)$ be a nearest neighbor network. A pair of units $\mathrm{X}, \mathrm{Y} \in \mathbf{U}$ are *reciprocal nearest neighbors* or RNNs iff $(\mathrm{X}, \mathrm{Y}) \in A$ and $(\mathrm{Y}, \mathrm{X}) \in A$.

Suppose $\mathrm{card}(\mathbf{U}) > 1$. Then in $\mathbf{N}$

- every unit/vertex $\mathrm{X} \in \mathbf{U}$ has the $\mathrm{outdeg}(\mathrm{X}) \geq 1$ — there is no isolated unit;
- along every walk the values of w are not increasing.

using these two observations we can show that in $\mathbf{N}^*_{NN}$:

- all the values of w on a closed walk are the same and all its arcs are reciprocal — all arcs between units in a nontrivial (at least 2 units) strong component are reciprocal;
- every maximal (can not be extended) elementary (no arc is repeated) walk ends in a RNNs pair;
- there exists at least one RNNs pair – corresponding to $\min_{\mathrm{X},\mathrm{Y} \in \mathbf{U}, \mathrm{X} \neq \mathrm{Y}} d(\mathrm{X}, \mathrm{Y})$.

Quick agglomerative clustering algorithms

Any graph $\mathbf{G}_{NN}$ is a subgraph of $\mathbf{G}^*_{NN}$. Its connected components are directed (acyclic) trees with a single RNNs pair in the root.

Based on the nearest neighbor graph very efficient $O(n^2)$ algorithms for agglomerative clustering for methods with the reducibility property can be built.

$chain := [\,]$; $\mathbf{W} := \mathbf{U}$;
while $\mathrm{card}(\mathbf{W}) > 1$ **do begin**
  if $chain = [\,]$ **then** select an arbitrary unit $\mathrm{X} \in \mathbf{W}$ **else** $\mathrm{X} := last(chain)$;
  grow a NN-chain from X until a pair (Y, Z) of RNNs are obtained;
  agglomerate Y and Z:
    $T := \mathrm{Y} \cup \mathrm{Z}$; $\mathbf{W} := \mathbf{W} \setminus \{\mathrm{Y}, \mathrm{Z}\} \cup \{T\}$; compute $D(T, W), W \in \mathbf{W}$
end;

It can be shown that if the clustering method has the reducibility property (minimum, maximum, Ward, ...; but not Bock) then the NN-chain remains a NN-chain also after the agglomeration of the RNNs pair.

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 δ 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 (Trinajstić, 1983).

Fragments count: select a collection of subgraphs (fragments), for example triads, and count the number of appearences of each – *fragments spectrum*.

Invariants and structural properties

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

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

The number of vertices, the number of arcs, the number of edges, maximum degree Δ, chromatic number χ, ... are all graph invariants.

Invariants have an important role in examining the isomorphism of two graphs.

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

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

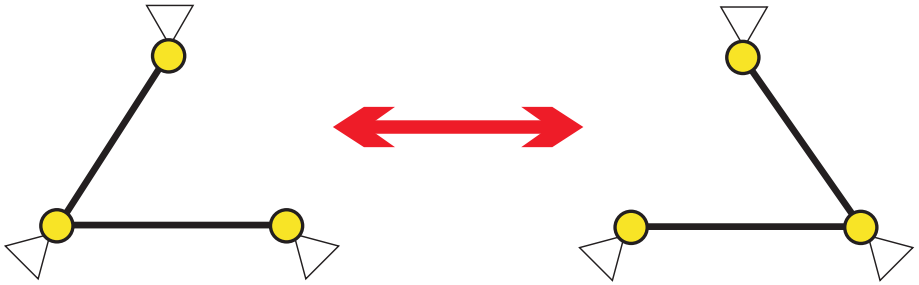
A collection $\mathcal{I}$ of invariants/structural properties is *complete* iff

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

In most cases there is no efficiently computable complete collection.

Transformations

Different dissimilarities between strings are based on transformations: insert, delete, transpose (Levenshtein 1966, Kashyap 1983). For binary trees Robinson considered a dissimilarity based on the transformation of *neighbors exchange over an edge*.



There is a natural generalization of this approach to graphs and other structured objects (Batagelj 1988): Let $\mathcal{T} = \{T_k\}$ be a set of *basic transformations* of units $T_k : \mathcal{U} \to \mathcal{U}$ and $v : \mathcal{T} \times \mathcal{U} \to \mathbb{R}^+$ value of transformation, which satisfy the conditions:

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

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

Transformations based dissimilarity

Suppose that for each pair $\mathrm{X}, \mathrm{Y} \in \mathcal{U}$ 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:

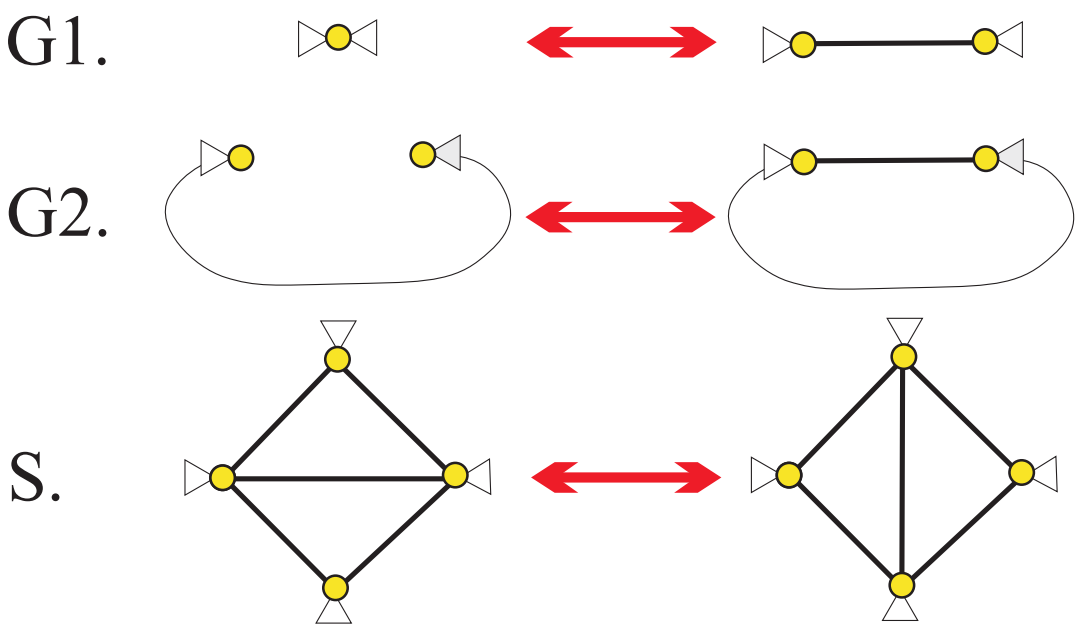
$$d(\mathrm{X}, \mathrm{Y}) = \min_{\tau}(v(\tau(\mathrm{X})) : \tau(\mathrm{X}) = \mathrm{Y})$$

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.

Examples of transformations



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 vertices S is such a transformation.

Clustering in Graphs and Networks

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

Again we can use the standard clustering methods provided that we have an appropriate definition of dissimilarity between vertices.

The usual approach is to define a vector description $[v] = [t_1, t_2, \ldots, t_m]$ of each vertex $v \in V$, and then use some standard dissimilarity δ on $\mathbb{R}^m$ to compare these vectors $d(u, v) = \delta([u], [v])$. For some 'nonstandard' such descriptions see Moody (2001) and Harel, Koren (2001).

We can assign to each vertex v also different neighborhoods

$$N(v) = \{u \in V : (v, u) \in L\}$$

and other sets. In these cases the dissimilarities between sets are used on them.

Properties of vertices

For a given graph $\mathbf{G} = (V, L)$ a property $t : V \to \mathbb{R}$ is *structural* iff for every automorphism φ 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 vertex v
$t(v) =$ number of vertices at distance d from vertex v
$t(v) =$ number of triads of type x at vertex v

Properties of pairs of vertices

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

$q(u, v) =$ **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 vertices q we can describe each vertex 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 vertices $u, v \in V$ as $d(u, v) = \delta([u], [v])$.

Matrix dissimilarities

The following is a list of dissimilarities, used in literature, based on properties of pairs of vertices for measuring the similarity between vertices v_i and v_j $(p \geq 0)$:

Manhattan: $d_m(v_i, v_j) = \sum_{s=1}^{n}(|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$

Euclidean: $d_E(v_i, v_j) = \sqrt{\sum_{s=1}^{n}((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$

Truncated Man.: $d_s(v_i, v_j) = \sum_{\substack{s=1 \\ s \neq i,j}}^{n}(|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$

Truncated Euc.: $d_S(v_i, v_j) = \sqrt{\sum_{\substack{s=1 \\ s \neq i,j}}^{n}((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$

Corrected Man.: $d_c(p)(v_i, v_j) = d_s(v_i, v_j) + p \cdot (|q_{ii} - q_{jj}| + |q_{ij} - q_{ji}|)$

Corrected Euc.: $d_e(p)(v_i, v_j) = \sqrt{d_S(v_i, v_j)^2 + p \cdot ((q_{ii} - q_{jj})^2 + (q_{ij} - q_{ji})^2)}$

Corrected diss.: $d_C(p)(v_i, v_j) = \sqrt{d_c(p)(v_i, v_j)}$

The corrected dissimilarities with $p = 1$ should be used.

Graph theory approaches

The basic decomposition of graphs is to (weakly) connected components – partition of vertices (and links); and to (weakly) biconnected components – partition of links. For both very efficient algorithms exist.

From a network $\mathbf{N} = (V, L, w)$ we can get for a treshold t a layer network $\mathbf{N}(t) = (V, L_t, w)$ 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.

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

For directed graphs the fundamental decomposition results can be found in Harary, Norman, and Cartwright (1965).

Decomposition of directed graphs

Given a simple directed graph $\mathbf{G} = (V, R)$, $R \subseteq V \times V$ we introduce two new relations, $R^\star$ (*transitive and reflexive closure*) and $\overline{R}$ (*transitive closure*), based on R:

$$uR^\star v := \exists k \in \mathbb{N} : uR^k v \qquad \text{and} \qquad u\overline{R}v := \exists k \in \mathbb{N}^+ : uR^k v$$

or equivalently

$$R^\star = \bigcup_{k \in \mathbb{N}} R^k \qquad \text{and} \qquad \overline{R} = \bigcup_{k \in \mathbb{N}^+} R^k$$

Theorem 1.16

a) $uR^k v$ *iff in the graph* $\mathbf{G} = (V, R)$ *there exists a walk of length* k *from* u *to* v.

b) $uR^\star v$ *iff in the graph* $\mathbf{G} = (V, R)$ *there exists a walk from* u *to* v.

c) $u\overline{R}v$ *iff in the graph* $\mathbf{G} = (V, R)$ *there exists a non-null walk from* u *to* v.

Acyclic Relations

A relation $R \in V \times V$ is *acyclic* if and only if

$$\forall v \in V \, \forall k > 0 : \neg v(R \setminus I)^k v$$

i.e. if its graph, except for loops, contains no cycles. This condition can be written also in the form $\overline{(R \setminus I)} \cap I = \emptyset$. We shall denote by $\mathrm{Acy}(V)$ the set of all acyclic relations on V.

A relation $R \in V \times V$ is *strictly acyclic* if and only if

$$\forall v \in V \, \forall k > 0 : \neg v R^k v$$

i.e. if its graph contains no cycles and, also, loops are not allowed. This condition can be written also in form $\overline{R} \cap I = \emptyset$. Each strictly acyclic relation is also acyclic.

Theorem 1.17 *For an acyclic relation $R \in \mathrm{Acy}(V)$ over a finite, nonempty set V there is at least one minimal, $R^{-1}(v) \subseteq \{v\}$, and at least one maximal, $R(v) \subseteq \{v\}$, element.*

Factorization

Suppose that on the set V we have a relation $R \in V \times V$ and an equivalence $\sim$. The equivalence $\sim$ partitions the set V into equivalence classes which form the family $V/\sim$. In $V/\sim$ we can define the *factor* relation $\mathbf{r} = R/\sim$

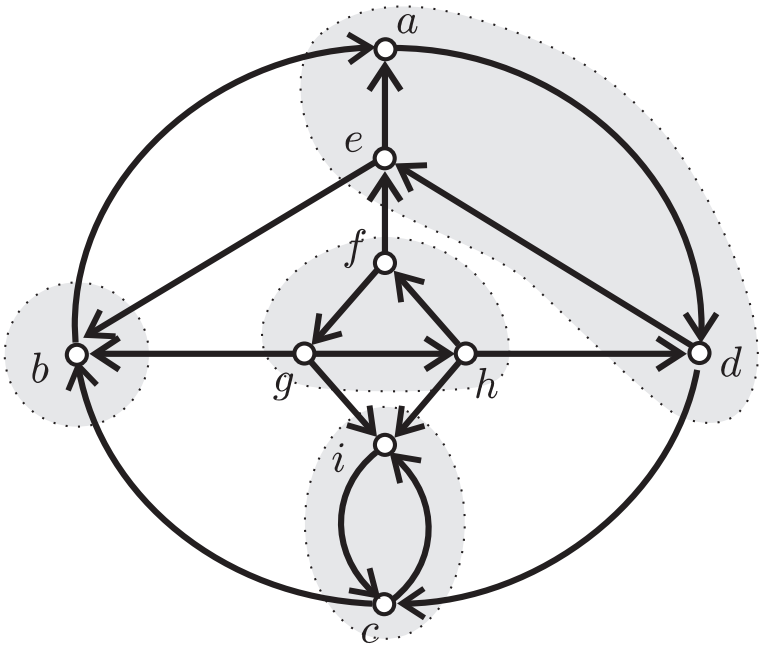
$$\mathbf{r} = R/\sim := \{\exists x \in X \exists y \in Y : xRy\}$$

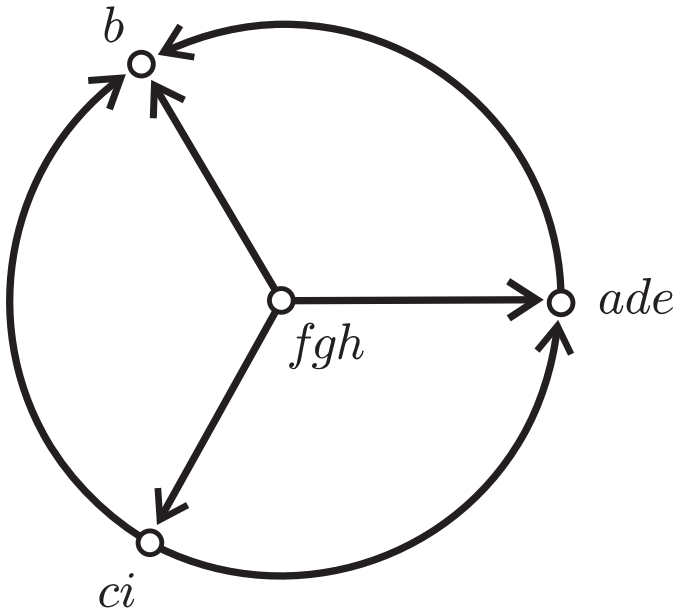
We will see, later, that all blockmodels can be described in these terms. The factor relation is the image of a blockmodel.

For a relation $R \in V \times V$ the strong connectivity relation $R^S = R^\star \cap (R^{-1})^\star$ is an equivalence. It partitions the set V into equivalence classes (strong components) which form a family V/R^S.

Theorem 1.18 *Let $R \in V \times V$. The relation $\sqsubseteq := R/R^S$ is acyclic on V/R^S. If R is a preorder (transitive and reflexive) then $\sqsubseteq$ is a partial order on V/R^S. If R is a tournament (asymmetric and comparable) then $\sqsubseteq$ is a linear order on V/R^S.*

Graph, strong components and factorization





Cores

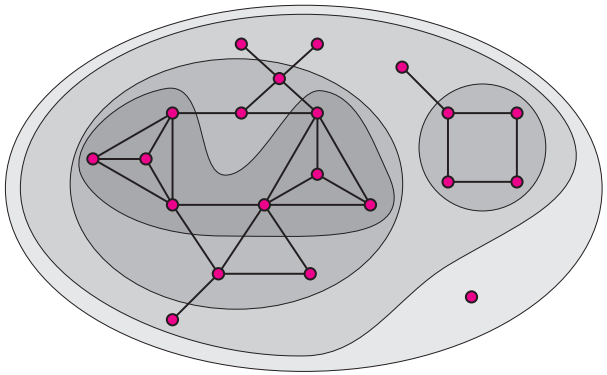
The notion of a core was introduced by Seidman in 1983.

In a given graph $\mathbf{G} = (V, L)$ a subgraph $\mathbf{H}_k = (W, L|W)$ induced by the set W is a k-*core* or a *core of order* k iff $\forall v \in W : \deg_H(v) \geq k$, and $\mathbf{H}_k$ is the maximum subgraph with this property. The core of maximum order is also called the *main* core. The *core number* of vertex v is the highest order of a core that contains this vertex.

The degree $\deg(v)$ can be: in-degree, out-degree, in-degree + out-degree, ... determining different types of cores.

In figure an example of cores decomposition of a given graph is presented. We can see the following properties of cores:

- The cores are nested: $i < j \implies \mathbf{H}_j \subseteq \mathbf{H}_i$
- Cores are not necessarily connected subgraphs.



Determining and using cores

A very efficient $O(m)$ algorithm (Batagelj, Zaveršnik 2002) for determining the cores hierarchy can be built based on the following property:

> If from a given graph $\mathbf{G} = (V, L)$ we recursively delete all vertices, and lines incident with them, of degree less than k, the remaining graph is the k-core.

The notion of cores can be generalized to networks.

Using cores we can identify the densiest parts of a graph. For revealing the internal structure of the main core we can use standard clustering procedures on dissimilarities between vertices. Afterwards we can remove the links of the main core and analyse the residium graph.

Cores can be used also to localize the search for some computationally more demanding substructures.

Short cycles

A subgraph $\mathbf{H} = (V', A')$ of $\mathbf{G} = (V, A)$ is *cyclic k-gonal* if each its vertex and each its edge belong to at least one cycle of length at most k and at least 2 in $\mathbf{H}$.

A sequence $(C_1, C_2, \ldots, C_s)$ of cycles of length at most k (and at least 2) of $\mathbf{G}$ *cyclic k-gonally connects* vertex $u \in V$ to vertex $v \in V$ iff $u \in C_1$ and $v \in C_s$ or $u \in C_s$ and $v \in C_1$ and $V(C_{i-1}) \cap V(C_i) \neq \emptyset$, $i = 2, \ldots s$; such sequence is called a *cyclic k-gonal chain*.

A pair of vertices $u, v \in V$ is *cyclic k-gonally connected* iff $u = v$, or there exists a cyclic k-gonal chain that connects u to v.

Theorem 1.19 *Cyclic k-gonal connectivity is an equivalence relation on the set of vertices V.*

An arc is *cyclic* iff it belongs to some cycle (of any length) in the graph $\mathbf{G}$.

Theorem 1.20 *If in the graph $\mathbf{G}$ for each cyclic arc the length of a shortest cycle that contains it is at most k then the cyclic k-gonal reduction of $\mathbf{G}$ is an acyclic graph.*

Final remarks

The agglomerative methods can be adapted for large sparse networks in sense of relational constraint clustering – we have to compute dissimilarities only between units/vertices connected by a link.

The Sollin's MST algorithm can be very efficiently implemented for large sparse networks.