# Methods of Network Analysis

# Clustering and Blockmodeling

## 2. Approaches to Clustering

Vladimir Batagelj
University of Ljubljana, Slovenia

University of Konstanz, Algorithms and Data Structures

June 6, 2002, 14-16h, room F 426

# 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 $(\Phi, P)$ 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, $\forall \mathrm{X} \in \Phi : \mathrm{X}S\mathrm{X}$.

They are the 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 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

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

# $\Delta$ for the generalized Ward's method

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 (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 $\Phi$ 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 $\Phi$ 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.

## . . . conditions

**F4.** The predicate $\psi$ 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 2.1** *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 2.2** *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** $\exists C_i, C_j \in \mathbf{C}(k)\colon (i \neq j \wedge \psi(C_i, C_j))$ **repeat**
2.1. $(C_p, C_q) := \mathrm{argmin}\{D(C_i, C_j)\colon i \neq j \wedge \psi(C_i, C_j)\}$;
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 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 | $\alpha_1$ | $\alpha_2$ | $\beta$ | $\gamma$ | $\delta_t$ | $v(C_t)$ |
|---|---|---|---|---|---|---|
| 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 | $\frac{w_{ps}^2}{w_{pqs}^2}$ | $\frac{w_{qs}^2}{w_{pqs}^2}$ | $-\frac{w_s w_{pq}}{w_{pqs}^2}$ | $0$ | $0$ | – |

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

# Hierarchies

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

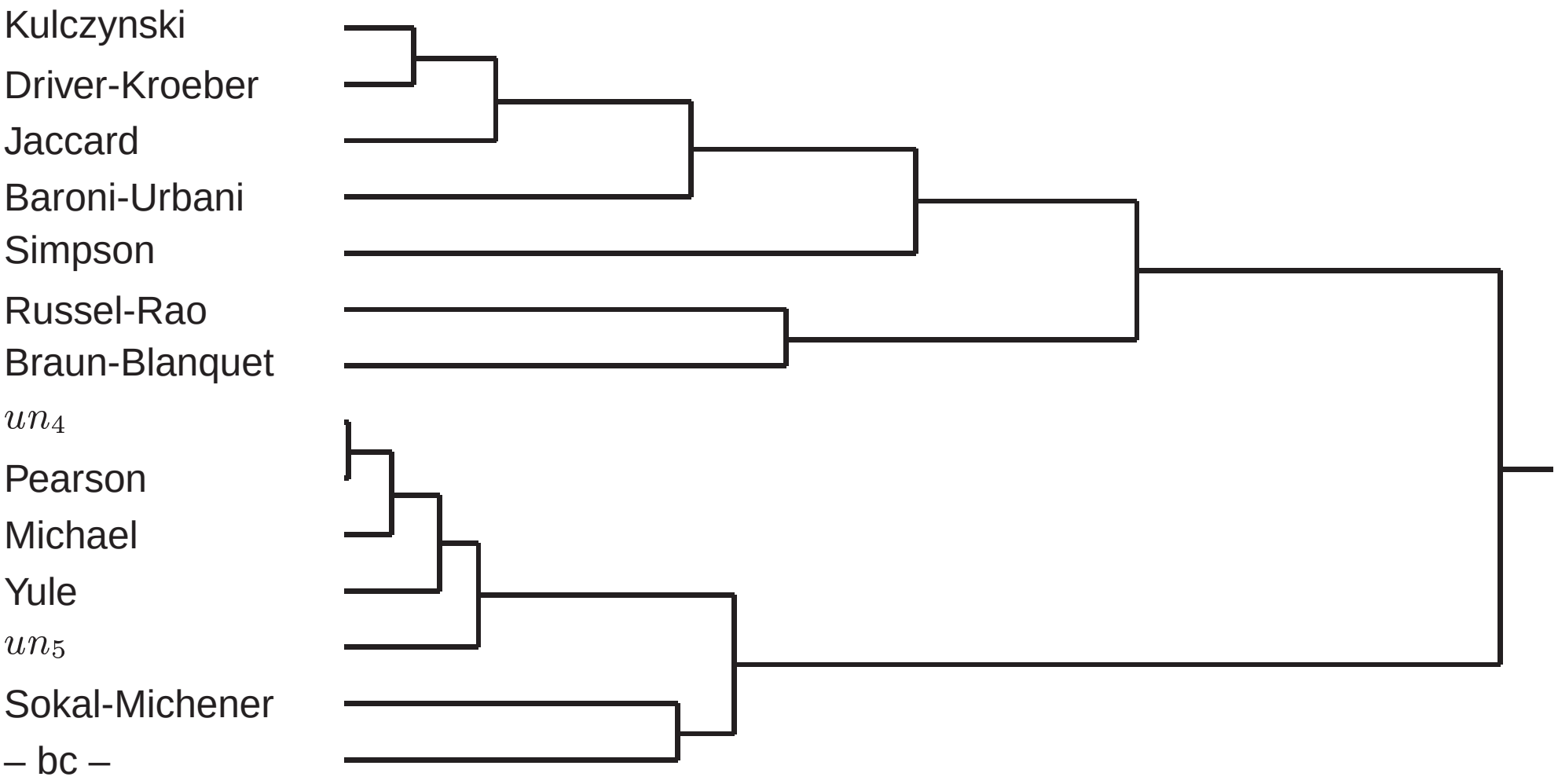
$$\mathbf{C}(A) = \{\overline{B}(\mathrm{X}, r) : \mathrm{X} \in \mathbf{U}, r \in A\} \cup \{\{\mathbf{U}\}\} \cup \{\{\mathrm{X}\} : \mathrm{X} \in \mathbf{U}\}$$

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

Batagelj (1981) showed:

**Theorem 2.3** *$h_D$ 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 2.4** *If a dissimilarity $D$ has the reducibility property then $h_D$ is a level function.*

## Adding hierarchical methods

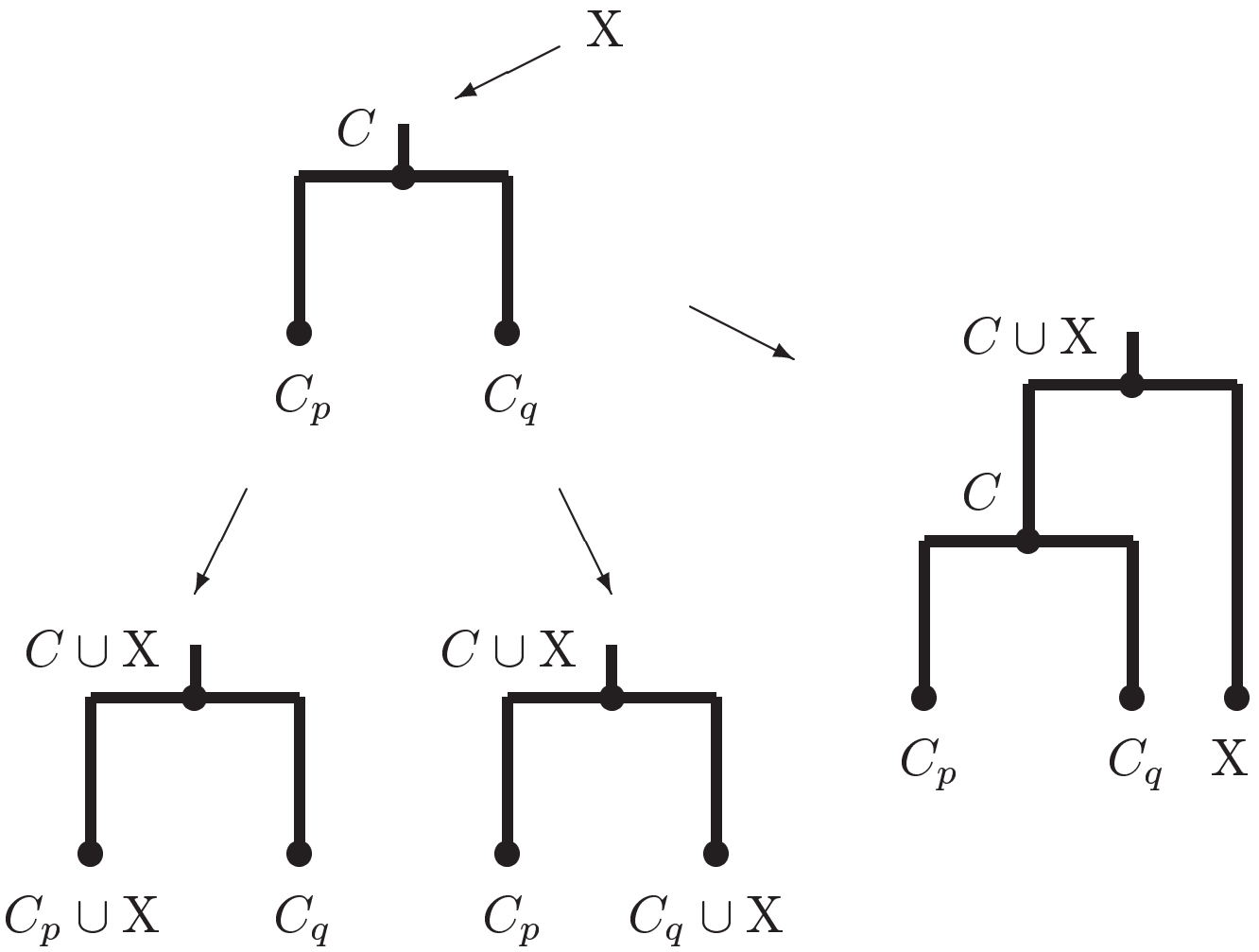
Suppose that we already built a clustering tree $\mathcal{T}$ over the set of units $\mathbf{U}$. To add a new unit $X$ to the tree $\mathcal{T}$ we start in the root and branch down. Assume that we reached the node corresponding to cluster $C$, which was obtained by joining subclusters $C_p$ and $C_q$.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

**Theorem 2.5** *In the (locally with respect to transitions) minimal clustering for the problem* $(P_k, \mathrm{SR})$

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

*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:

| | |
|---|---|
| $\Lambda$ | – set of *representatives* |
| $\mathrm{L} \subseteq \Lambda$ | – *representation* |
| $\Psi$ | – set of *feasible representations* |
| $W : \Phi \times \Psi \rightarrow \mathbb{R}_0^+$ | – *extended criterion function* |
| $G : \Phi \times \Psi \rightarrow \Psi$ | – *representation function* |
| $F : \Phi \times \Psi \rightarrow \Phi$ | – *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

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

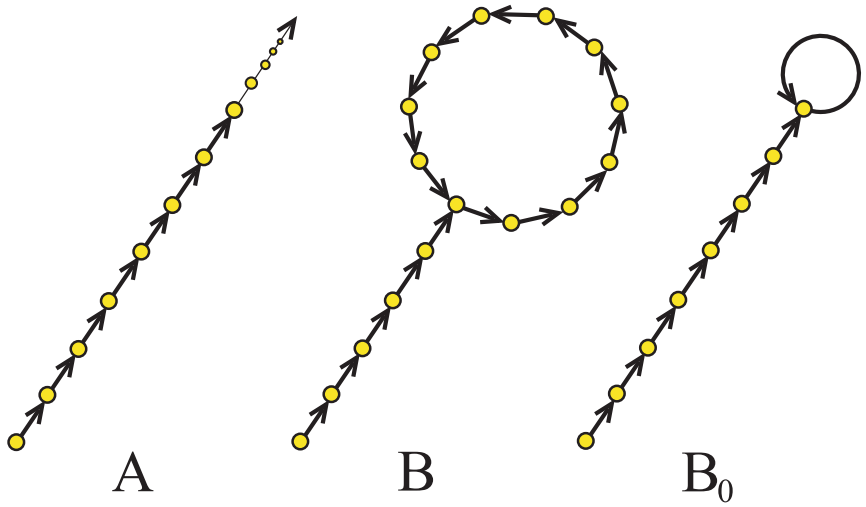
and the sequence of values of the extended criterion function $u_n = W(\mathbf{C}_n, \mathrm{L}_n)$. Let us also denote $u^* = P(\mathbf{C}^*)$. Then it holds:

**Theorem 2.6** *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 $\Phi$ 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.

## 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 $\Psi$ 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.

**Theorem 2.7** *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 2.8** *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 2.9** *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. $G(\mathbf{C}) = \underset{\mathrm{L} \in \Psi}{\operatorname{argmin}} W(\mathbf{C}, \mathrm{L})$

and

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

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

In the next, 3. lecture we shall discuss

- 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