

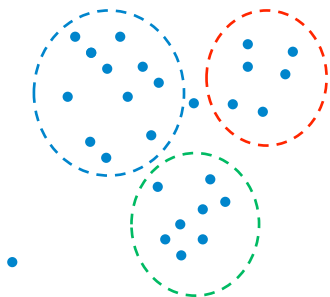
# Evidential clustering

Thierry Denœux

Université de technologie de Compiègne, France  
Institut Universitaire de France  
<https://www.hds.utc.fr/~tdenoeux>

6th School on Belief Functions and their Applications  
Ishikawa, Japan, October 29, 2023

# Clustering



- $n$  objects described by
  - ▶ Attribute vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  (attribute data) or
  - ▶ Dissimilarities (proximity data)
- Goals:
  - 1 Discover groups in the data
  - 2 Assess the uncertainty in group membership

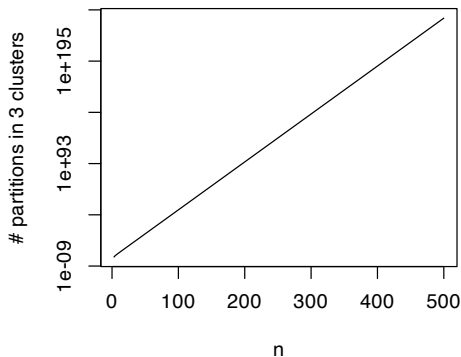
# Frame of discernment

- Assumption: each object belongs to one and only one cluster.
- Consequently, we search for a **partition** of the set  $\mathcal{O}$  of objects, i.e., a family  $P_1, \dots, P_c$  of pairwise disjoint subsets of  $\mathcal{O}$  such that  $\bigcup_{k=1}^c P_k = \mathcal{O}$ .
- The frame of discernment is, thus, one of these sets:
  - ▶ The set  $\mathcal{P}_{[c]}$  of all partitions of  $\mathcal{O}$  into exactly  $c$  clusters (if  $c$  is fixed)
  - ▶ The set  $\mathcal{P}_c = \bigcup_{k=1}^c \mathcal{P}_{[k]}$  of all partitions of  $\mathcal{O}$  into at most  $c$  clusters
  - ▶ The set  $\mathcal{P} = \bigcup_{k=1}^n \mathcal{P}_k$  of all partitions of  $\mathcal{O}$ .
- Problem: these sets are huge!

# Number partitions of $n$ objects into $c$ clusters

The number of partitions of  $n$  objects into  $c$  clusters is the Stirling number of the second kind  $S(n, c)$  given by

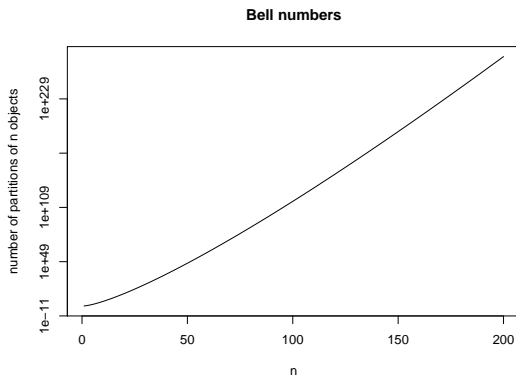
$$S(n, c) = \sum_{i=0}^c \frac{(-1)^{c-i} j^n}{(c-i)! i!}$$



# Number of partitions of $n$ objects

The total number of partitions of  $n$  objects is the Bell number  $B_n$  given by

$$B_n = \sum_{c=1}^n S(n, c)$$



# Defining belief functions in sets of partitions

- Contrary for misleading statements still found in some papers, the theory of belief functions can be implemented in such extremely large spaces, provided the focal sets and the objective of the analysis are carefully defined.
- Different objectives:
  - ▶ Compute the degree of belief and/or plausibility for any subset of partitions (can be very difficult)
  - ▶ Compute the plausibility of any partition
  - ▶ Compute the plausibility of any partition, up to some constant
  - ▶ Find the most plausible partition
- In the following, I will describe two ways to define belief functions on sets of partitions.

# Outline

- 1 **Belief functions in a space of partitions**
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 **Evidential clustering algorithms**
  - EK-NNclus
  - Evidential *c*-means
  - EVCLUS
  - NN-EVCLUS
  - BootClus

# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - EK-NNclus
  - Evidential *c*-means
  - EVCLUS
  - NN-EVCLUS
  - BootClus



# Pairwise mass functions

- The first approach focuses on **pairs of objects**.
- For any  $i < j$ , let  $\Theta_{ij} = \{s_{ij}, \neg s_{ij}\}$ , where  $s_{ij}$  means “objects  $i$  and  $j$  belong to the same group” and  $\neg s_{ij}$  means “objects  $i$  and  $j$  do not belong to the same group”.
- A pairwise mass function is a mass function on  $\Theta_{ij}$  with general form:

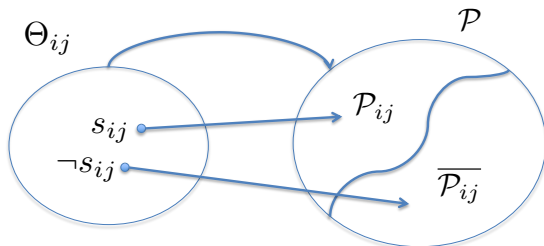
$$m_{ij}(\{s_{ij}\}) = \alpha_{ij}$$

$$m_{ij}(\{\neg s_{ij}\}) = \beta_{ij}$$

$$m_{ij}(\Theta_{ij}) = 1 - \alpha_{ij} - \beta_{ij}$$

- These mass functions can be vacuously extended to the set  $\mathcal{P}$ .

# Vacuous extension of pairwise mass functions



- Let  $\mathcal{P}_{ij}$  denote the set of partitions of the  $n$  objects such that objects  $o_i$  and  $o_j$  are in the same group.
- Each mass function  $m_{ij}$  can be **vacuously extended** to the set  $\mathcal{P}$  of all partitions:

$$\begin{aligned}
 m_{ij}(\{s_{ij}\}) &\longrightarrow \mathcal{P}_{ij} \\
 m_{ij}(\{\neg s_{ij}\}) &\longrightarrow \overline{\mathcal{P}}_{ij} \\
 m_{ij}(\Theta_{ij}) &\longrightarrow \mathcal{P}
 \end{aligned}$$

# Combination of pairwise partitions

- The extended mass functions can then be combined by Dempster's rule to form a mass function  $m^{\mathcal{P}}$  on  $\mathcal{P}$ :

$$m^{\mathcal{P}} = \bigoplus_{i < j} m_{ij}^{\mathcal{P}}$$

- We will only combine the contour functions. The contour function of  $m_{ij}^{\mathcal{P}}$  is

$$\begin{aligned} p_{ij}(P) &= \begin{cases} m_{ij}^{\mathcal{P}}(\mathcal{P}_{ij}) + m_{ij}^{\mathcal{P}}(\mathcal{P}) & \text{if } P \in \mathcal{P}_{ij} \\ m_{ij}^{\mathcal{P}}(\overline{\mathcal{P}_{ij}}) + m_{ij}^{\mathcal{P}}(\mathcal{P}) & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 - \beta_{ij} & \text{if } P \in \mathcal{P}_{ij} \\ 1 - \alpha_{ij} & \text{otherwise} \end{cases} \\ &= (1 - \beta_{ij})^{p_{ij}} (1 - \alpha_{ij})^{1-p_{ij}} \end{aligned}$$

where  $p_{ij} = I(P \in \mathcal{P}_{ij})$ , and  $I(\cdot)$  is the indicator function.

- The contour function of  $m^{\mathcal{P}}$  is

$$pI(P) \propto \prod_{i < j} (1 - \beta_{ij})^{p_{ij}} (1 - \alpha_{ij})^{1-p_{ij}} \quad \text{for any } P \in \mathcal{P}$$

# Finding the most plausible partitions

- The most plausible partition can be found by maximizing

$$\ln pl(P) = \sum_{i < j} p_{ij} \ln(1 - \beta_{ij}) + (1 - p_{ij}) \ln(1 - \alpha_{ij}) + C$$

subject to  $p_{ij} \in \{0, 1\}$  and transitivity constraints

$$\forall i < j < k, \quad p_{ij} + p_{jk} - 1 \leq p_{ik},$$

which is a binary linear programming problem.

- In practice, this problem can be solved exactly only for small  $n$ . A heuristic algorithm (EK-NNclus) will be described later.

[Return to EK-NNclus](#)

# Outline

- 1 **Belief functions in a space of partitions**
  - Orthogonal sums of pairwise belief functions
  - **Credal partitions**
- 2 **Evidential clustering algorithms**
  - EK-NNclus
  - Evidential *c*-means
  - EVCLUS
  - NN-EVCLUS
  - BootClus

# Credal partition

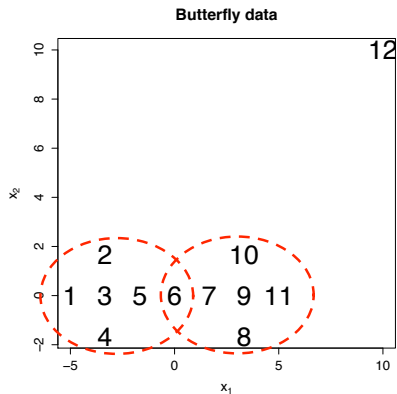
## Definition (Credal partition)

Let  $\mathcal{O} = \{o_1, \dots, o_n\}$  be a set of  $n$  objects and  $\Omega = \{\omega_1, \dots, \omega_c\}$  be a set of  $c$  groups (clusters). A **credal (evidential) partition (CP)** of  $\mathcal{O}$  is an  $n$ -tuple  $M = (m_1, \dots, m_n)$ , where each element  $m_i$  is a mass function on  $\Omega$  describing the uncertain membership of  $o_i$  to one of the  $c$  clusters.

### Remarks:

- The mass functions may be unnormalized, i.e., we may have  $m_i(\emptyset) > 0$ . The quantity  $m_i(\emptyset)$  is then interpreted as our degree of belief that  $o_i$  does not belong to any of the  $c$  clusters (i.e., it is an outlier).
- If all mass functions are certain, i.e., for all  $i$ , there exists  $k$  such that  $m_i(\{\omega_k\}) = 1$ , then the CP specifies a (hard) partition.

# Example



## Credal partition

	$\emptyset$	$\{\omega_1\}$	$\{\omega_2\}$	$\{\omega_1, \omega_2\}$
$m_3$	0	1	0	0
$m_5$	0	0.5	0	0.5
$m_6$	0	0	0	1
$m_{12}$	0.9	0	0.1	0

# Relationship with fuzzy clustering

- In fuzzy clustering, each object  $o_i$  has a degree of membership  $u_{ik} \in [0, 1]$  to each cluster  $\omega_k$ , with the constraint  $\sum_{k=1}^c u_{ik} = 1$ .
- Let  $M = (m_1, \dots, m_n)$  be a CP. If all mass functions  $m_i$  are **Bayesian**, then  $M$  defines a fuzzy partition with membership degrees  $u_{ik} = m_i(\{\omega_k\})$ .
- An arbitrary CP  $M = (m_1, \dots, m_n)$  can be **summarized** as a fuzzy partition by transforming each mass function  $m_i$  into a probability distribution using, e.g., the plausibility transformation

$$u_{ik} = \frac{pl_i(\omega_k)}{\sum_{l=1}^c pl_i(\omega_l)}$$



# Relationship with rough clustering

- In rough clustering, each cluster  $\omega_k$  is characterized by two subsets of objects: a **lower approximation**  $\underline{\omega}_k$  and an **upper approximation**  $\bar{\omega}_k$ , with  $\underline{\omega}_k \subseteq \bar{\omega}_k$ . The membership of object  $i$  to cluster  $k$  is described by a pair  $(\underline{u}_{ik}, \bar{u}_{ik}) \in \{0, 1\}^2$ , with  $\underline{u}_{ik} \leq \bar{u}_{ik}$ ,  $\sum_{k=1}^c \underline{u}_{ik} \leq 1$  and  $\sum_{k=1}^c \bar{u}_{ik} \geq 1$ .
- Let  $M = (m_1, \dots, m_n)$  be a CP. If all mass functions  $m_i$  are **logical**, with  $m_i(A_i) = 1$  for some  $A_i \subseteq \Omega$ , then  $M$  defines a hard partition with lower and upper approximations defined as

$$\underline{\omega}_k = \{o_i \in O : A_i = \{\omega_k\}\}, \quad \bar{\omega}_k = \{o_i \in O : \omega_k \in A_i\}.$$

The membership values to the lower and upper approximations of cluster  $\omega_k$  are  $\underline{u}_{ik} = Bel_i(\{\omega_k\})$  and  $\bar{u}_{ik} = Pl_i(\{\omega_k\})$ .

## Relationship with rough clustering (continued)

- An arbitrary CP  $M = (m_1, \dots, m_n)$  can be summarized as a rough partition by approximating each mass function  $m_i$  by a logical mass function  $m'_i$  whose focal set is the focal set of  $m_i$  with the largest mass, i.e.,  $m'_i(A_i) = 1$  with

$$A_i = \arg \max_{A \subseteq \Omega} m_i(A)$$

# Associated mass function on $\mathcal{P}_c$

- Let  $M = (m_1, \dots, m_n)$  be a CP, and let  $\mathcal{F}$  be the set of mappings from  $\mathcal{O}$  to  $\Omega$  (called **labeling functions**). Each mass function  $m_i$  can be extended to  $\mathcal{F}$  as follows:

$$m_i(A) \rightarrow \{f \in \mathcal{F} : f(o_i) \in A\}$$

Let us denote by  $m_i^{\mathcal{F}}$  the vacuous extension of  $m_i$  in  $\mathcal{F}$ .

- Assuming the mass function  $m_1^{\mathcal{F}}, \dots, m_n^{\mathcal{F}}$  to be independent, they can be combined by Dempster's rule. Let  $m^{\mathcal{F}} = \bigoplus_{i=1}^n m_i^{\mathcal{F}}$ . The focal sets of  $m^{\mathcal{F}}$  are of the form

$$\bigcap_{i=1}^n \{f \in \mathcal{F} : f(o_i) \in A_i\} = \{f \in \mathcal{F} : f(o_1) \in A_1, f(o_2) \in A_2, \dots, f(o_n) \in A_n\}$$

where  $A_1, \dots, A_n$  are focal sets of  $m_1, \dots, m_n$ , and the corresponding mass is  $\prod_{i=1}^n m_i(A_i)$ .

## Associated mass function on $\mathcal{P}_c$ (continued)

- Now, let  $\Phi : \mathcal{F} \rightarrow \mathcal{P}_c$  be the many-to-one mapping from  $\mathcal{F}$  to  $\mathcal{P}_c$  that maps each labeling function  $f$  to the corresponding partition  $\Phi(f) \in \mathcal{P}_c$ . We note that  $\mathcal{P}_c$  is a coarsening of  $\mathcal{F}$ .
- Let  $m^{\mathcal{P}_c}$  be the restriction of  $m^{\mathcal{F}}$  in  $\mathcal{P}_c$ . It is obtained by transferring each mass  $m^{\mathcal{F}}(F)$  to the set of all partitions represented by a labeling function  $f$  in  $F$ :

$$m^{\mathcal{F}}(F) \rightarrow \Phi(F) = \bigcup_{f \in F} \Phi(f)$$

- We have shown that a CP  $M = (m_1, \dots, m_n)$  is a compact representation of a mass function  $m^{\mathcal{P}_c}$  in the set  $\mathcal{P}_c$  of partitions of  $\mathcal{O}$  with at most  $c$  clusters.

# Example

- Let  $\mathcal{O} = \{o_1, o_2, o_3\}$ ,  $\Omega = \{\omega_1, \omega_2\}$ , and

$$m_1(\{\omega_1\}) = 0.6, \quad m_1(\Omega) = 0.4$$

$$m_2(\{\omega_1\}) = 0.5, \quad m_2(\Omega) = 0.5$$

$$m_3(\{\omega_2\}) = 0.7, \quad m_3(\Omega) = 0.3$$

- $m^{\mathcal{F}}$  has 8 focal sets:

$$m^{\mathcal{F}}(1, 1, 2) = 0.21, \quad m^{\mathcal{F}}(1, 1, \{1, 2\}) = 0.09$$

$$m^{\mathcal{F}}(1, \{1, 2\}, 2) = 0.21, \quad m^{\mathcal{F}}(1, \{1, 2\}, \{1, 2\}) = 0.09$$

$$m^{\mathcal{F}}(\{1, 2\}, 1, 2) = 0.14, \quad m^{\mathcal{F}}(\{1, 2\}, 1, \{1, 2\}) = 0.06$$

$$m^{\mathcal{F}}(\{1, 2\}, \{1, 2\}, 2) = 0.14, \quad m^{\mathcal{F}}(\{1, 2\}, \{1, 2\}, \{1, 2\}) = 0.06$$

## Example (continued)

- There are 4 partitions of  $\mathcal{O}$  in at most two groups:

$$P_0 = \{\mathcal{O}\}, \quad P_1 = \{\{o_1\}, \{o_2, o_3\}\}$$

$$P_2 = \{\{o_1, o_3\}, \{o_2\}\}, \quad P_3 = \{\{o_1, o_2\}, \{o_3\}\}$$

- We have

$$\Phi(1, 1, 2) = P_3, \quad \Phi(1, 1, \{1, 2\}) = \{P_0, P_3\}$$

$$\Phi(1, \{1, 2\}, 2) = \{P_1, P_3\}, \quad \Phi(1, \{1, 2\}, \{1, 2\}) = P_2$$

$$\Phi(\{1, 2\}, 1, 2) = \{P_2, P_3\}, \quad \Phi(\{1, 2\}, 1, \{1, 2\}) = P_2$$

$$\Phi(\{1, 2\}, \{1, 2\}, 2) = P_2, \quad \Phi(\{1, 2\}, \{1, 2\}, \{1, 2\}) = P_2$$

- Consequently,  $m^{\mathcal{P}_2}$  is

$$m^{\mathcal{P}_2}(\{P_3\}) = 0.21, \quad m^{\mathcal{P}_2}(\{P_0, P_3\}) = 0.09, \quad m^{\mathcal{P}_2}(\{P_1, P_3\}) = 0.21$$

$$m^{\mathcal{P}_2}(\{P_2, P_3\}) = 0.14, \quad m^{\mathcal{P}_2}(P_2) = 0.35$$

# Pairwise mass functions derived from a CP

- Let  $M = (m_1, \dots, m_n)$  be a CP.
- For a pair of objects  $\{o_i, o_j\}$ , we consider the question “Do  $o_i$  and  $o_j$  belong to the same group?” defined on the frame  $\Theta_{ij} = \{s_{ij}, \neg s_{ij}\}$ .
- Let  $S_{ij} = \{(\omega_1, \omega_1), \dots, (\omega_c, \omega_c)\}$ . The mapping  $\rho : 2^{\Theta_{ij}} \rightarrow 2^{\Omega^2}$  such that  $\rho(\{s_{ij}\}) = S_{ij}$ ,  $\rho(\{\neg s_{ij}\}) = \overline{S_{ij}}$  and  $\rho(\Theta_{ij}) = \Omega^2$  is a refining. Consequently,  $\Theta_{ij}$  is a coarsening of  $\Omega^2$ .

$\Omega$	$\omega_1$	$\omega_2$	$\omega_3$	$\omega_4$
$\omega_1$				
$\omega_2$				
$\omega_3$				
$\omega_4$				

Given  $m_i$  and  $m_j$  on  $\Omega$ , a mass function  $m_{ij}$  on  $\Theta_{ij}$  can be computed as follows:

- 1 **Extend**  $m_i$  and  $m_j$  to  $\Omega^2$
- 2 **Combine** the extensions of  $m_i$  and  $m_j$  by the unnormalized Dempster's rule
- 3 Compute the **restriction** of the combined mass function to  $\Theta_{ij}$

# Expression of the pairwise mass function

- We have:

$$m_{ij}(\emptyset) = m_i(\emptyset) + m_j(\emptyset) - m_i(\emptyset)m_j(\emptyset)$$

$$m_{ij}(\{s_{ij}\}) = \sum_{k=1}^c m_i(\{\omega_k\})m_j(\{\omega_k\})$$

$$m_{ij}(\{\neg s_{ij}\}) = \kappa_{ij} - m_{ij}(\emptyset)$$

$$m_{ij}(\Theta_{ij}) = 1 - \kappa_{ij} - \sum_{k=1}^c m_i(\{\omega_k\})m_j(\{\omega_k\}).$$

where  $\kappa_{ij}$  is the degree of conflict between  $m_i$  and  $m_j$ .

- In particular,

$$p_{ij}(s_{ij}) = 1 - \kappa_{ij}$$



# Example

- CP:

$A$	$\emptyset$	$\{\omega_1\}$	$\{\omega_2\}$	$\{\omega_1, \omega_2\}$
$m_1(A)$	0.3	0.6	0.1	0.0
$m_2(A)$	0.0	0.7	0.1	0.2
$m_3(A)$	0.0	0.1	0.6	0.3

- Pairwise mass functions:

$A$	$\emptyset$	$\{s_{ij}\}$	$\{\neg s_{ij}\}$	$\{s_{ij}, \neg s_{ij}\}$
$m_{12}(A)$	0.30	0.43	0.13	0.14
$m_{13}(A)$	0.30	0.12	0.37	0.21
$m_{23}(A)$	0.00	0.13	0.43	0.44

# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - EK-NNclus
  - Evidential  $c$ -means
  - EVCLUS
  - NN-EVCLUS
  - BootClus

# Evidential clustering algorithm

- In the following, we will see:
  - ▶ An algorithm for finding the most plausible partition, given a belief function on the set of partitions obtained as the orthogonal sum of pairwise mass functions (EK-NNclus).
  - ▶ Four methods for constructing CPs: ECM, EVCLUS, NN-EVCLUS and Bootclus.
- These algorithms are implemented in the R package `evclus`<sup>1</sup>.

---

<sup>1</sup><https://CRAN.R-project.org/package=evclus>

# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - **EK-NNclus**
  - Evidential *c*-means
  - EVCLUS
  - NN-EVCLUS
  - BootClus

# Decision-directed clustering

- **Decision-directed** approach to clustering:
  - ▶ A randomly-initialized classifier is used to label the samples
  - ▶ The classifier is then updated using the labeled samples, and the process is repeated until no changes occur in the labels
- For instance, the  $c$ -means algorithm is based on this principle: here, the nearest-prototype classifier is used to label the samples, and it is updated by taking as prototypes the centers of each cluster.
- Idea: apply this principle using the evidential  $K$ -NN rule<sup>2</sup> as the base classifier.
- The corresponding clustering algorithm is called EK-NNclus<sup>3</sup>.

---

<sup>2</sup>T. Denœux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(05):804-813, 1995.

<sup>3</sup>T. Denœux, O. Kanjanatarakul and S. Sriboonchitta. EK-NNclus: a clustering procedure based on the evidential K-nearest neighbor rule. *Knowledge-Based Systems*, Vol. 88, pages 57–69, 2015.

# EK-NNclus algorithm

## Step 1: preparation

- Let  $D = (d_{ij})$  be a symmetric  $n \times n$  matrix of distances or dissimilarities between the  $n$  objects.
- Given  $K$ , we compute the set  $N_K(i)$  of indices of the  $K$  nearest neighbors of object  $i$ .
- We then transform the distances as

$$\alpha_{ij} = \begin{cases} \varphi(d_{ij}) & \text{if } j \in N_K(i) \\ 0 & \text{otherwise,} \end{cases}$$

for all  $(i, j) \in \{1, \dots, n\}^2$ , where  $\varphi$  is a decreasing mapping from  $\mathbb{R}_+$  to  $[0, 1]$ .

- If computing time is not an issue,  $K$  can be chosen very large, even equal to  $n - 1$ .

# EK-NNclus algorithm

## Step 2: initialization

- To initialize the algorithm, the objects are **labeled randomly** (or using some prior knowledge if available).
- As the number of clusters is usually unknown, it can be set to  $c = n$ , i.e., we initially assume that there are as many clusters as objects and each cluster contains exactly one object.
- If  $n$  is very large, we can give  $c$  a large value, but smaller than  $n$ , and initialize the object labels randomly.
- We define cluster-membership by binary variables  $y_{ik}$  as  $y_{ik} = 1$  if object  $o_i$  belongs to cluster  $k$ , and  $y_{ik} = 0$  otherwise.

# EK-NNclus algorithm

## Step 3: iteration

- An iteration of the algorithm consists in **updating the object labels in some random order, using the EKNN rule.**
- For each object  $o_i$ :

- 1 The evidence from its neighbor  $o_j \in N_K(i)$  is represented by the mass function

$$m_{ij}(\{\omega_k\}) = \alpha_{ij} y_{jk}, \quad k = 1, \dots, c$$

$$m_{ij}(\Omega) = 1 - \alpha_{ij}$$

- 2 The  $K$  mass functions are combined:

$$m_i = \bigoplus_{j \in N_K(i)} m_{ij}$$

- 3 Object  $o_i$  is assigned to the cluster with the highest plausibility, i.e., we update variables  $y_{ik}$  as

$$y_{ik} = \begin{cases} 1 & \text{if } pl_i(\omega_k) = \max_{k'} pl_i(\omega_{k'}) \\ 0 & \text{otherwise} \end{cases}$$

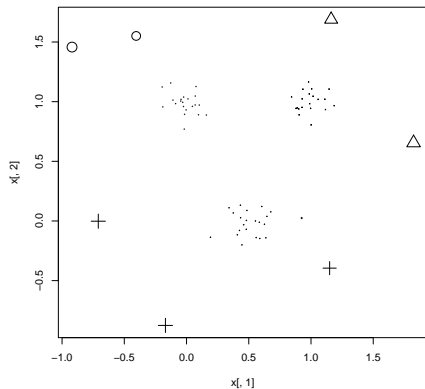
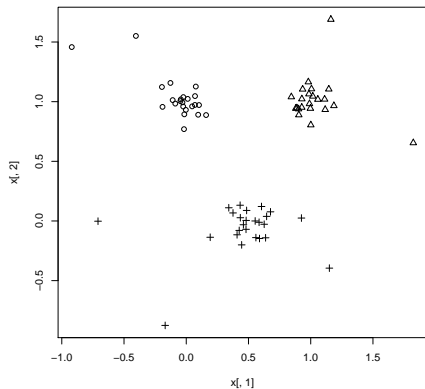


# EK-NNclus algorithm

## Stopping criterion

- If the label of at least one object has been changed during the last iteration, then the objects are randomly re-ordered and a new iteration is started.
- Otherwise, the algorithm is stopped.

# Example



The graph on the left shows an obtained hard partition. In the graph on the right, the size of the symbols is proportional to  $m_i(\Omega)$ . We can see that outliers are clearly identified.

# Convergence

## Theorem (Convergence of EK-NNclus)

*If  $K = n - 1$ , each update of a class label during an iteration of EK-NNclus strictly decreases the following energy function*

$$E(\mathbf{Y}) = -\frac{1}{2} \sum_{k=1}^c \sum_{i=1}^n \sum_{j \neq i} \ln(1 - \alpha_{ij}) y_{ik} y_{jk}$$

*Consequently, EK-NNclus converges to a stable partition in a finite number of iterations.*

Remarks: In practice, convergence is still observed when  $K < n - 1$ .

# Interpretation

- The following equality holds

$$pl(P) = -E(\mathbf{Y}) + C$$

where  $pl(P)$  is the **plausibility of the partition  $P$**  encoded by  $\mathbf{Y}$ , in the orthogonal sum of pairwise mass functions model, with  $\beta_{ij} = 0$ .

▶ [Go back to pairwise mass functions](#)

- EK-NNclus can thus be seen as a greedy search algorithm that **finds a local maximum of the plausibility contour function**, in the space of all partitions of the dataset.

# Experiments

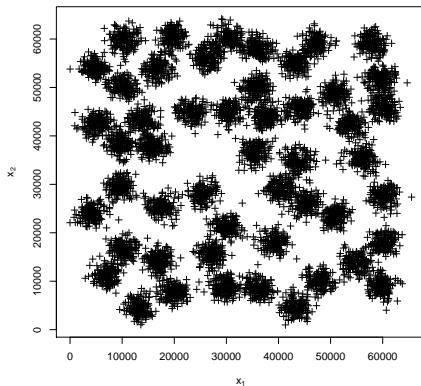
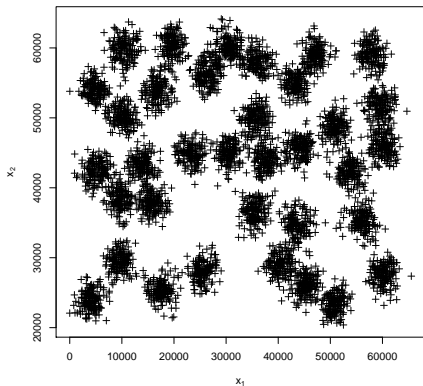
- Settings:

- ▶  $\varphi(d_{ij}) = \exp(-\gamma d_{ij}^2)$ , where  $d_{ij}$  is the Euclidean distance between objects  $i$  and  $j$ .
- ▶ Parameter  $\gamma$  can be fixed to the inverse of the  $q$ -quantile of the set  $\Delta = \{d_{ij}^2, i \in \{1, \dots, n\}, j \in N_K(i)\}$ .

- A-sets: Two-dimensional datasets with 20, 35 and 50 clusters

- ▶ Parameter  $q$  of the EK-NNclus algorithm was fixed to  $q = 0.9$ .
- ▶ The number of neighbors was fixed to  $K = 150$  for dataset A1, and  $K = 200$  for datasets A2 and A3 (rule of thumb:  $K$  should be of the order of two to three times  $\sqrt{n}$ ).
- ▶ Two initialization methods:  $c_0 = n$  initial clusters, and  $c_0 = 1000$  random initial clusters.
- ▶ The EK-NNclus algorithm was run 10 times.

# A-sets



# Results

Dataset	Result	EK-NNclus ( $c_0 = n$ )	EK-NNclus ( $c_0 = 1000$ )	pdfCluster	model-based	model-based (constrained)
A1 $n = 3000$	$c$	20 (0)	20 (0)	17	24	24
	time	32.9 (3.14)	9.8 (0.2)	84.5	31.8	7.88
A2 $n = 5250$	$c$	35 (0)	34 (1)	26	39	39
	time	193 (9.81)	23.8 (0.6)	298	138	36.2
A3 $n = 7500$	$c$	49 (1)	49 (2.5)	34	50	51
	time	358 (8.23)	35.1 (1.09)	629	412	99.4

# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - EK-NNclus
  - **Evidential c-means**
  - EVCLUS
  - NN-EVCLUS
  - BootClus



# Principle

- Problem: generate a credal partition  $M = (m_1, \dots, m_n)$  from **attribute data**  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ ,  $\mathbf{x}_i \in \mathbb{R}^p$ .
- The ECM algorithm<sup>4</sup> generalizes the hard and fuzzy c-means algorithms:
  - ▶ Each cluster is represented by a **prototype**.
  - ▶ **Cyclic coordinate descent** algorithm: optimization of a cost function alternatively with respect to the prototypes and to the credal partition.

---

<sup>4</sup>M.-H. Masson and T. Denœux. ECM: An evidential version of the fuzzy c-means algorithm. *Pattern Recognition*, 41(4):1384–1397, 2008.

# Fuzzy c-means (FCM)

- Minimize

$$J_{\text{FCM}}(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^{\beta} d_{ik}^2$$

with  $d_{ik} = \|\mathbf{x}_i - \mathbf{v}_k\|$  subject to the constraints  $\sum_{k=1}^c u_{ik} = 1$  for all  $i$ .

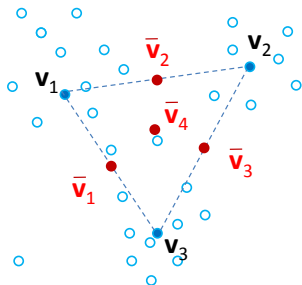
- Alternate optimization algorithm:

$$\mathbf{v}_k = \frac{\sum_{i=1}^n u_{ik}^{\beta} \mathbf{x}_i}{\sum_{i=1}^n u_{ik}^{\beta}}, \quad \forall k$$

$$u_{ik} = \frac{d_{ik}^{-2/(\beta-1)}}{\sum_{\ell=1}^c d_{i\ell}^{-2/(\beta-1)}}, \quad \forall k, i$$

# ECM algorithm

## Principle



- Each cluster  $\omega_k$  represented by a prototype  $\mathbf{v}_k$ .
- Each nonempty set of clusters  $A_j$  represented by a prototype  $\bar{\mathbf{v}}_j$  defined as the center of mass of the  $\mathbf{v}_k$  for all  $\omega_k \in A_j$ .
- Basic ideas:
  - ▶ For each nonempty  $A_j \subseteq \Omega$ ,  $m_{ij} = m_i(A_j)$  should be high if  $\mathbf{x}_i$  is close to  $\bar{\mathbf{v}}_j$ .
  - ▶ The distance to the empty set is defined as a fixed value  $\delta$ .

# ECM algorithm: cost function

- Define the nonempty focal sets  $\mathcal{F} = \{A_1, \dots, A_f\} \subseteq 2^\Omega \setminus \{\emptyset\}$ .
- Minimize

$$J_{\text{ECM}}(M, \mathbf{V}) = \sum_{i=1}^n \sum_{j=1}^f |A_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^n \delta^2 m_{i\emptyset}^\beta$$

subject to the constraints  $\sum_{j=1}^f m_{ij} + m_{i\emptyset} = 1$  for all  $i$ .

- Parameters:
  - ▶  $\alpha$  controls the **specificity** of mass functions (default: 1)
  - ▶  $\beta$  controls the **hardness** of the credal partition (default: 2)
  - ▶  $\delta$  controls the proportion of data considered as **outliers**
- $J_{\text{ECM}}(M, \mathbf{V})$  can be iteratively minimized with respect to  $M$  and to  $\mathbf{V}$ .

# ECM algorithm: update equations

Update of  $M$ :

$$m_{ij} = \frac{c_j^{-\alpha/(\beta-1)} d_{ij}^{-2/(\beta-1)}}{\sum_{k=1}^f c_k^{-\alpha/(\beta-1)} d_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}}$$

for  $i = 1, \dots, n$  and  $j = 1, \dots, f$ , and

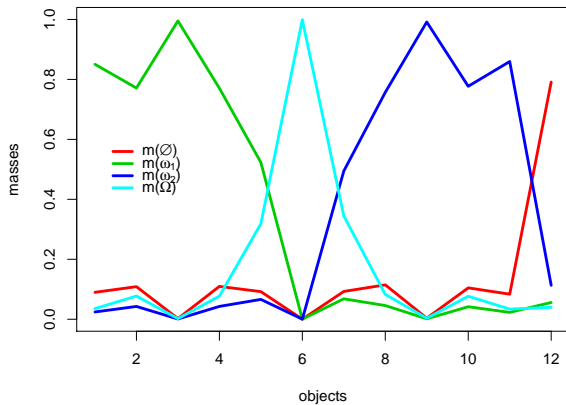
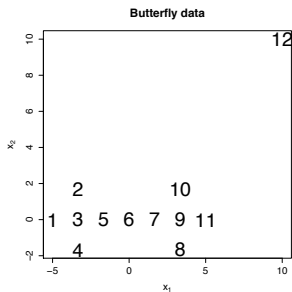
$$m_{i\emptyset} = 1 - \sum_{j=1}^f m_{ij}, \quad i = 1, \dots, n$$

Update of  $\mathbf{V}$ : solve a linear system of the form

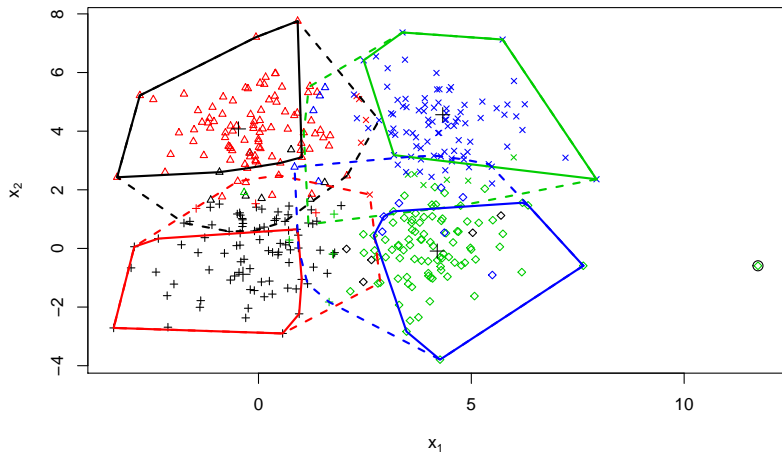
$$\mathbf{H}\mathbf{V} = \mathbf{B},$$

where  $\mathbf{B}$  is a matrix of size  $c \times p$  and  $\mathbf{H}$  a matrix of size  $c \times c$ .

# Butterfly dataset




## 4-class data set



# Constrained Evidential c-means

- In some cases, we may have some **prior knowledge** about the group membership of some objects.
- Such knowledge may take the form of **instance-level constraints** of two kinds:
  - 1 **Must-link** (ML) constraints, which specify that two objects certainly belong to the same cluster
  - 2 **Cannot-link** (CL) constraints, which specify that two objects certainly belong to different clusters
- The CECM algorithm<sup>5</sup> is a variant of ECM that can exploit such constraints.

---

<sup>5</sup>V. Antoine, B. Quost, M.-H. Masson and T. Denœux. CECM: Constrained Evidential C-Means algorithm. *Computational Statistics and Data Analysis*, 56(4):894–914, 2012. 



# Cost function of CECM

- To take into account ML and CL constraints, we can modify the cost function of ECM as

$$J_{\text{CECM}}(M, \mathbf{V}) = (1 - \xi)J_{\text{ECM}}(M, V) + \xi J_{\text{CONST}}(M)$$

with

$$J_{\text{CONST}}(M) = \frac{1}{|\mathcal{M}| + |\mathcal{C}|} \left[ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} pl_{ij}(\neg s_{ij}) + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}} pl_{ij}(s_{ij}) \right]$$

where

- $\mathcal{M}$  and  $\mathcal{C}$  are, respectively, the sets of ML and CL constraints.
- $pl_{ij}(s_{ij})$  and  $pl_{ij}(\neg s_{ij})$  are computed from the pairwise mass function  $m_{ij}$

[▶ Go back to pairwise mass functions](#)

- Minimizing  $J_{\text{CECM}}(M, \mathbf{V})$  w.r.t.  $M$  is a quadratic programming problem.

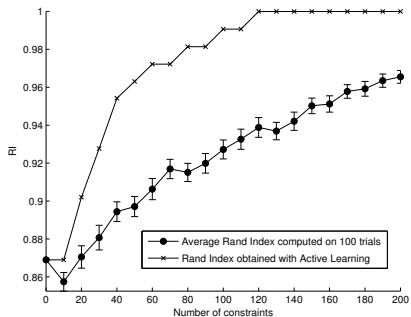
[◀ Return to NN-EVCLUS](#)

# Active learning

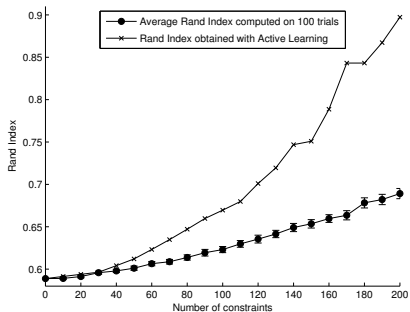
- ML and CL constraints are sometimes given in advance, but they can sometimes be elicited from the user using an **active learning strategy**.
- For instance, we may select pairs of object such that
  - ▶ The first object is classified with **high uncertainty** (e.g., an object such that  $m_i$  has high nonspecificity);
  - ▶ The second object is classified with **low uncertainty** (e.g., an object that is close to a cluster center).
- The user is then provided with this pair of objects, and enters either a ML or a CL constraint.

# Results

## Glass data



## Ionosphere data



# Other variants of ECM

Relational Evidential c-Means (RECM) for (metric) proximity data<sup>6</sup>

Spatial Evidential C-Means (SECM) for image segmentation<sup>7</sup>

Spatial Evidential C-Means with adaptive distance metric for image segmentation<sup>8</sup>

Credal c-means (CCM), a variant of ECM with a different definition of the distance between a vector and a meta-cluster<sup>9</sup>

---

<sup>6</sup>M.-H. Masson and T. Denœux. RECM: Relational Evidential c-means algorithm. *Pattern Recognition Letters* 30:1015–1026, 2009.

<sup>7</sup>B. Lelandais et al. Fusion of multi-tracer PET images for Dose Painting. *Medical Image Analysis*, 18(7):1247–1259, 2014

<sup>8</sup>C. Lian et al. Spatial Evidential Clustering with Adaptive Distance Metric for Tumor Segmentation in FDG-PET Images. *IEEE Trans. on Biomedical Engineering*, 65(1):21–30, 2018.

<sup>9</sup>Z-G Liu et al., Credal c-means clustering method based on belief functions, *Knowledge-Based Systems*, 74:119-132, 2015.


# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - EK-NNclus
  - Evidential *c*-means
  - **EVCLUS**
  - NN-EVCLUS
  - BootClus

# Learning a Credal Partition from proximity data

- Problem: given the dissimilarity matrix  $D = (d_{ij})$ , how to build a “reasonable” credal partition ?
- We need a model that relates cluster membership to dissimilarities.
- The EVCLUS algorithm<sup>10</sup> is based on the following idea: “The more similar two objects, the more plausible it is that they belong to the same group”.
- How to formalize this idea?

---

<sup>10</sup>T. Denœux and M.-H. Masson. EVCLUS: Evidential Clustering of Proximity Data. *IEEE Transactions on Systems, Man and Cybernetics B*, 34(1):95–109, 2004. 

# Formalization

- Let  $m_i$  and  $m_j$  be mass functions regarding the group membership of objects  $o_i$  and  $o_j$ .
- We have seen that the plausibility that objects  $o_i$  and  $o_j$  belong to the same group is

$$pl_{ij}(s_{ij}) = \sum_{A \cap B \neq \emptyset} m_i(A)m_j(B) = 1 - \kappa_{ij}$$

where  $\kappa_{ij}$  = **degree of conflict** between  $m_i$  and  $m_j$ .

- Problem: find a CP  $M = (m_1, \dots, m_n)$  such that **larger degrees of conflict  $\kappa_{ij}$  correspond to larger dissimilarities  $d_{ij}$** .

# Cost function

- Approach: **minimize the discrepancy** between the dissimilarities  $d_{ij}$  and the degrees of conflict  $\kappa_{ij}$ .
- Example of a **cost (stress) function**:

$$J(M) = \sum_{i < j} (\kappa_{ij} - \varphi(d_{ij}))^2$$

where  $\varphi$  is an increasing function from  $[0, +\infty)$  to  $[0, 1]$ , for instance

$$\varphi(d) = 1 - \exp(-\gamma d^2).$$

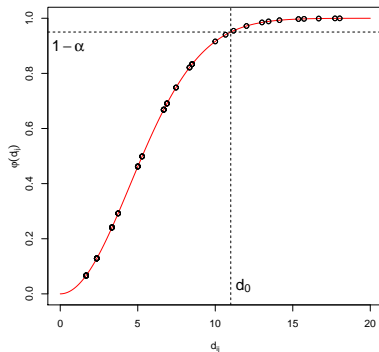
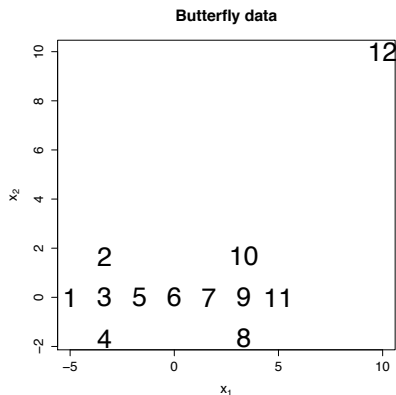
- $\gamma$  can be determined by fixing  $\alpha \in (0, 1)$  and  $d_0$  such that, for any two objects  $(o_i, o_j)$  with  $d_{ij} \geq d_0$ , the plausibility that they belong to the same cluster is at least  $1 - \alpha$ .



# Butterfly example

## Data and dissimilarities

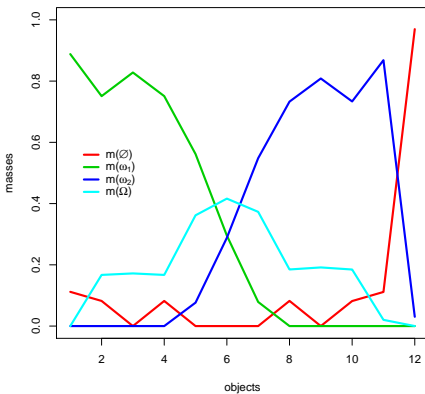
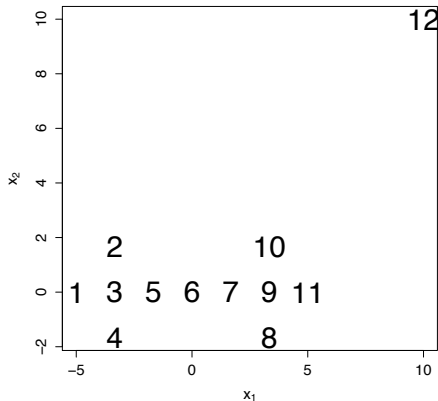
Determination of  $\gamma$  in  $\varphi(d) = 1 - \exp(-\gamma d^2)$ : fix  $\alpha \in (0, 1)$  and  $d_0$  such that, for any two objects  $(o_i, o_j)$  with  $d_{ij} \geq d_0$ , the plausibility that they belong to the same cluster is at least  $1 - \alpha$ .



# Butterfly example

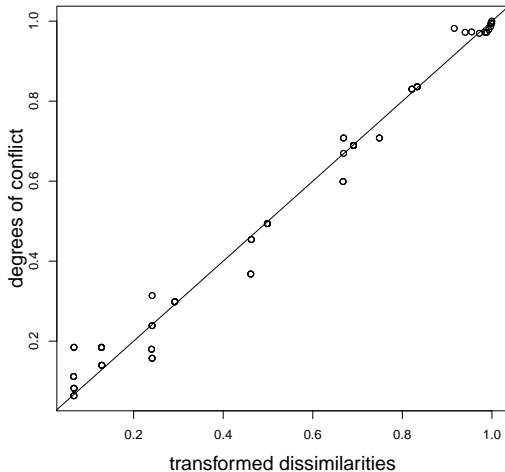
## Credal partition

Butterfly data

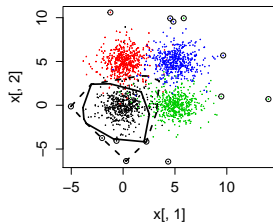
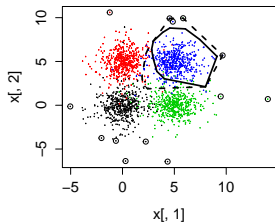
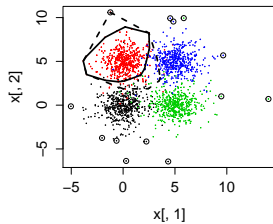
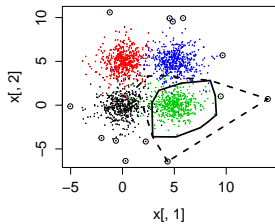


# Butterfly example

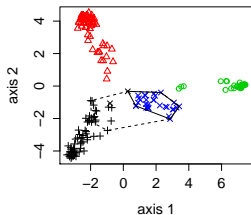
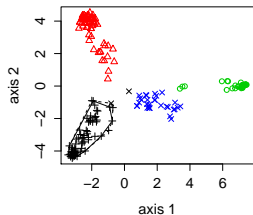
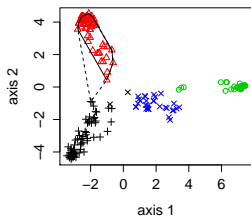
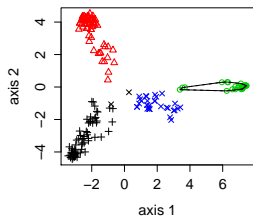
## Shepard diagram



# Example with a four-class dataset (2000 objects)



# Proteins dataset



- Nonmetric dissimilarity matrix derived from the structural comparison of 213 protein sequences.
- Ground truth: 4 classes of globins.
- Only 2 errors.

# Advantages

- Conceptually simple, clear interpretation.
- EVCLUS can handle **nonmetric** dissimilarity data (even expressed on an ordinal scale).
- It was also shown to outperform some of the state-of-the-art relational clustering techniques on a number of datasets.

# Limitations

- Requires to store the whole dissimilarity matrix; the space complexity is thus  $O(n^2)$ , where  $n$  is the number of objects. Restricts application to datasets with  $n \sim 10^2 - 10^3$ .
- Each computation of the gradient requires  $O(f^3 n^2)$  operations, where  $f$  is the number of focal sets of the mass functions. In the worst case,  $f = 2^c$ .
- Two improvements<sup>11</sup>:
  - 1 Sample dissimilarities
  - 2 Carefully select the focal sets

---

<sup>11</sup>T. Denœux, S. Sriboonchitta and O. Kanjanatarakul. Evidential clustering of large dissimilarity data. *Knowledge-Based Systems*, 106:179–195, 2016.

# Sampling dissimilarities

- EVCLUS requires to store the whole dissimilarity matrix: it is inapplicable to large proximity data.
- However, there is usually some **redundancy** in a dissimilarity matrix.
- In particular, if two objects  $o_1$  and  $o_2$  are very similar, then any object  $o_3$  that is dissimilar from  $o_1$  is usually also dissimilar from  $o_2$ .
- Because of such redundancies, it might be possible to compute the differences between degrees of conflict and dissimilarities, for **only a subset of randomly sampled dissimilarities**.



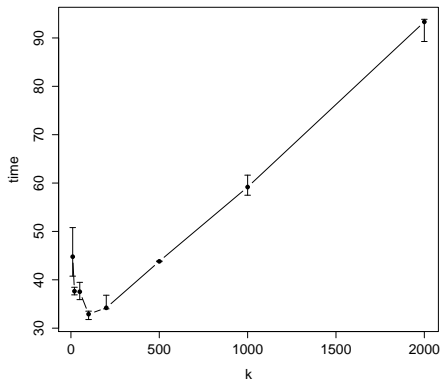
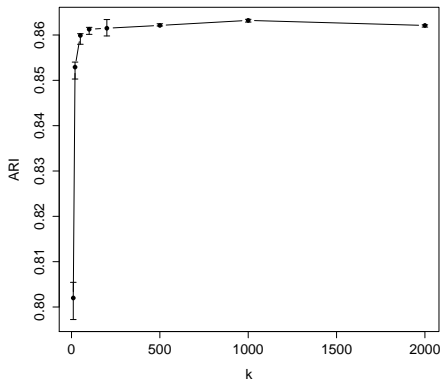
# New stress function

- Let  $j_1(i), \dots, j_k(i)$  be  $k$  integers sampled at random from the set  $\{1, \dots, i-1, i+1, \dots, n\}$ , for  $i = 1, \dots, n$ .
- Let  $J_k$  the following stress criterion,

$$J_k(M) = \sum_{i=1}^n \sum_{r=1}^k (\kappa_{i,j_r(i)} - \varphi(\mathbf{d}_{i,j_r(i)}))^2.$$

- The calculation of  $J_k(M)$  requires only  $O(nk)$  operations.
- If  $k$  can be kept constant as  $n$  increases, then time and space complexities are reduced from quadratic to linear.

# Example with simulated data ( $n = 10,000$ )

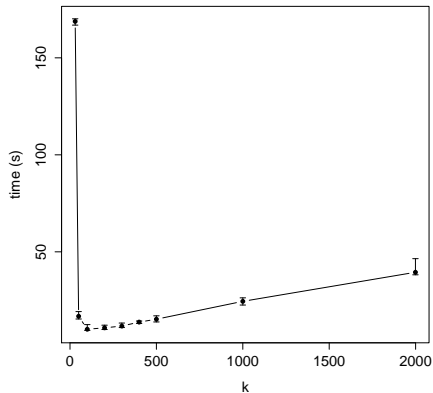
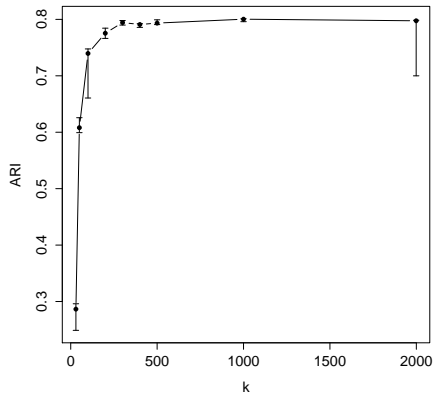


# Zongker Digit dissimilarity data

- Similarities between 2000 handwritten digits in 10 classes, based on deformable template matching.
- $k$ -EVCLUS was run with  $c = 10$  and different following values of  $k$ .
- Parameter  $d_0$  was fixed to the 0.3-quantile of the dissimilarities.
- $k$ -EVCLUS was run 10 times with random initializations.

# Zongker Digit dissimilarity data

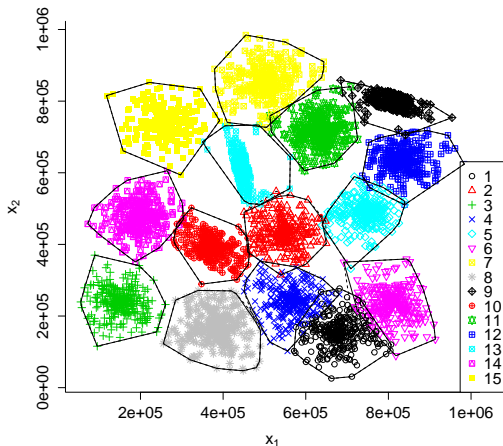
## Results



# Selecting the focal sets

- If no restriction is imposed on the focal sets, the number of parameters to be estimated in evidential clustering **grows exponentially** with the number  $c$  of clusters, which makes it intractable unless  $c$  is small.
- If we allow masses to be assigned to **all pairs of clusters**, the number of focal sets becomes **proportional to  $c^2$** , which is manageable for moderate values of  $c$  (say, until 10), but still impractical for larger  $n$ .
- Idea: assign masses only to **pairs of contiguous clusters**.
- If each cluster has at most  $q$  neighbors, then the number of focal sets is proportional to  $c$ .

# Example



The  $S_2$  dataset ( $n = 5000$ ) and the 15 clusters found by  $k$ -EVCLUS with  $k = 100$

# Method

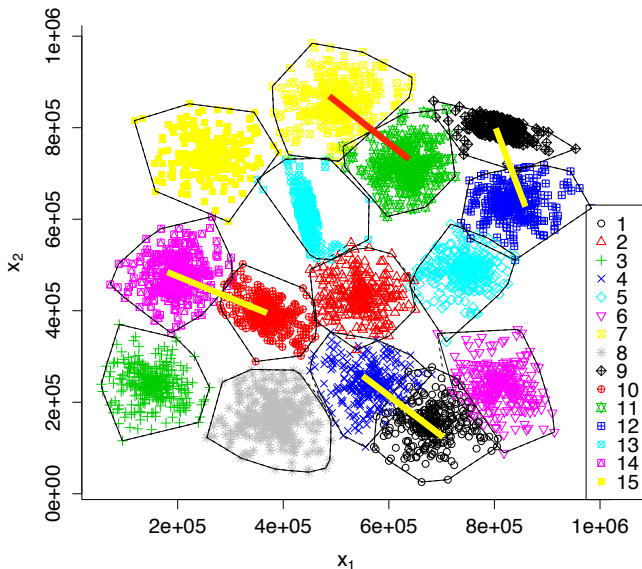
- Step 1:** Run a clustering algorithm (e.g., ECM or EVCLUS) with focal sets of cardinalities 0, 1 and (optionally)  $c$ . A credal partition  $M_0$  is obtained.
- Step 2:** Compute the similarity between each pair of clusters  $(\omega_j, \omega_\ell)$  as

$$S(j, \ell) = \sum_{i=1}^n pl_{ij} pl_{i\ell},$$

where  $pl_{ij}$  and  $pl_{i\ell}$  are the normalized plausibilities that object  $i$  belongs, respectively, to clusters  $j$  and  $\ell$ . Determine the set  $P_K$  of pairs  $\{\omega_j, \omega_\ell\}$  that are **mutual  $q$  nearest neighbors**.

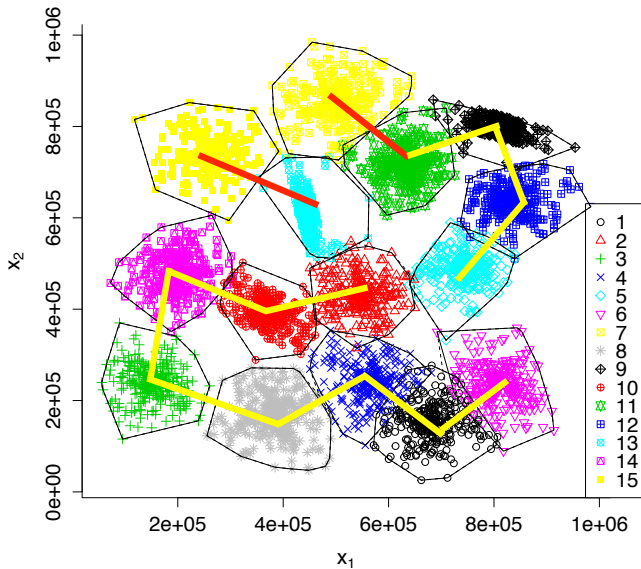
- Step 3:** Run the clustering algorithm again, starting from the previous credal partition  $M_0$ , and adding as focal sets the pairs in  $P_K$ .

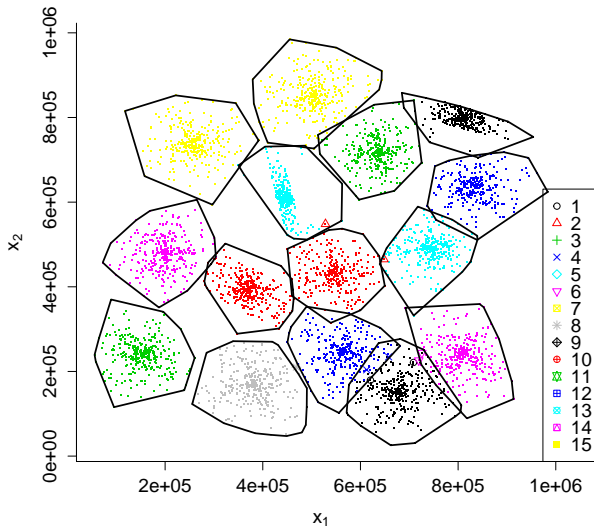
# Pairs of mutual neighbors with $q = 1$

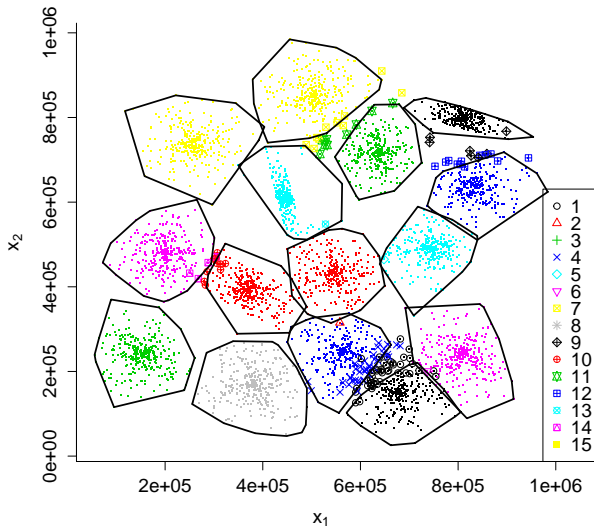




# Pairs of mutual neighbors with $q = 2$



Initial credal partition  $\mathcal{M}_0$ 

Final credal partition ( $q = 1$ )

# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - EK-NNclus
  - Evidential *c*-means
  - EVCLUS
  - **NN-EVCLUS**
  - BootClus

# Motivation

- As opposed to ECM, the EVCLUS algorithm does not build a compact representation of clusters as a collection of prototypes, but it learns an evidential partition of the  $n$  objects directly.
- If each mass function is constrained to have  $f$  focal sets, the number of free parameters is  $n(f - 1)$ : it grows linearly with the number of objects. This characteristic makes EVCLUS **impractical for clustering very large datasets**.
- Also, the algorithm learns an evidential partition of a given dataset, but it does not allow us to **extrapolate beyond the learning set** and make predictions for new objects.
- NN-EVCLUS<sup>12</sup> is a neural network version of EVCLUS that addresses these issues.

---

<sup>12</sup>T. Denœux. NN-EVCLUS: Neural Network-based Evidential Clustering. *Information Sciences*, 572:297–330, 2021.

# Data

- We assume the learning data to consist in
  - ▶ An  $n \times n$  dissimilarity matrix  $\mathbf{D} = (\delta_{ij})$
  - ▶ A collection of  $n$  attribute vectors  $\mathcal{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$
- Most of the time, we get the  $n$  attribute vectors first and compute  $\mathbf{D}$  as, for instance, the matrix of Euclidean distances between vectors  $\mathbf{x}_i$ :

$$\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|.$$

Sometimes, the dissimilarities can be computed using not only the attribute vectors, but also side information such as must-link and cannot-link constraints (more on this later).

- If the data consists only in the dissimilarity matrix  $\mathbf{D}$ , we can compute attribute vectors  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$  by applying, e.g., Principal Component Analysis to the dissimilarity matrix.

# Model of mass functions

- We compute a vector representation  $\mathbf{m}$  of a mass function  $m$  as the output of a **multi-layer neural network** with a softmax output layer:

$$\mathbf{m} = g(\mathbf{x}, \theta)$$

where  $\theta$  is the vector of weights.

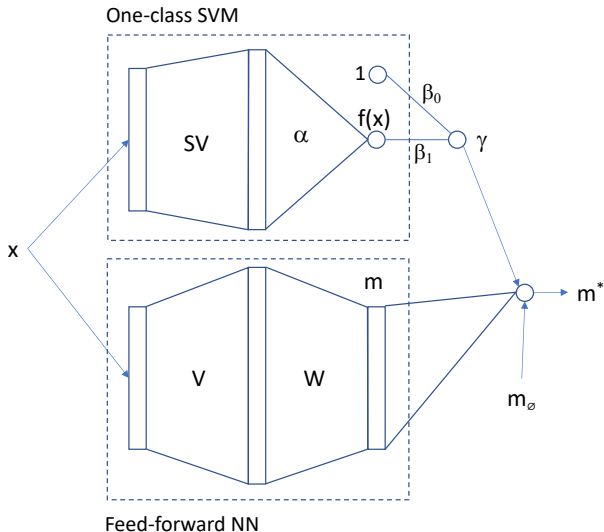
- To account for outliers, we also use a **one-class SVM** with output  $f(\mathbf{x})$ .
- The complete mass function is

$$\mathbf{m}^* = \gamma \mathbf{m} + (1 - \gamma) \mathbf{m}_\emptyset$$

where  $\mathbf{m}_\emptyset$  is the mass vector corresponding to the mass function  $m_\emptyset$  such that  $m_\emptyset(\emptyset) = 1$ , and  $\gamma \in [0, 1]$  is a coefficient defined as

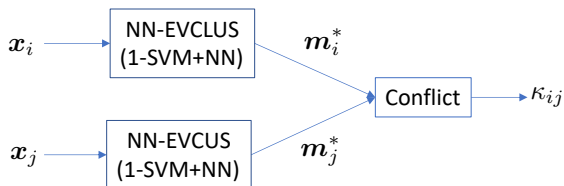
$$\gamma = \frac{\eta}{1 + \eta} \quad \text{with} \quad \eta = \ln [1 + \exp(\beta_0 + \beta_1 f(\mathbf{x}))]$$

# Model of mass functions





# Learning



- Loss function:

$$\mathcal{L}_{ij}(\theta) = (\kappa_{ij}(\theta) - \delta_{ij}^*)^2$$

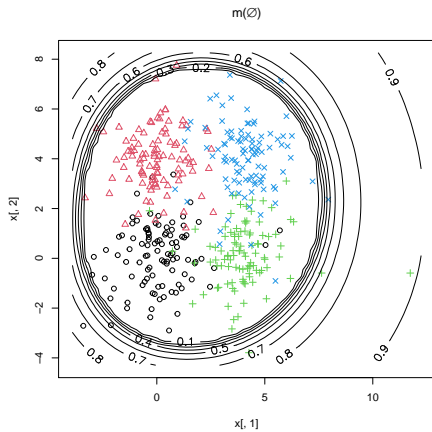
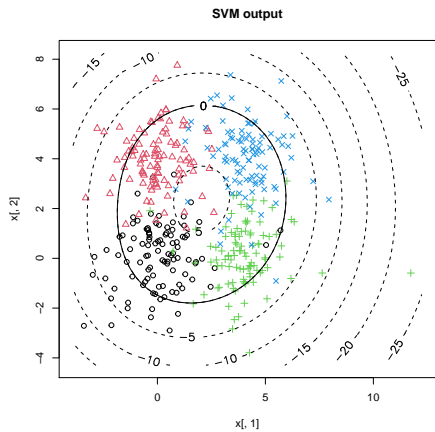
where  $\delta_{ij}^*$  is the transformed dissimilarity between objects  $o_i$  and  $o_j$ .

- The network is trained by minimizing the **regularized average loss**

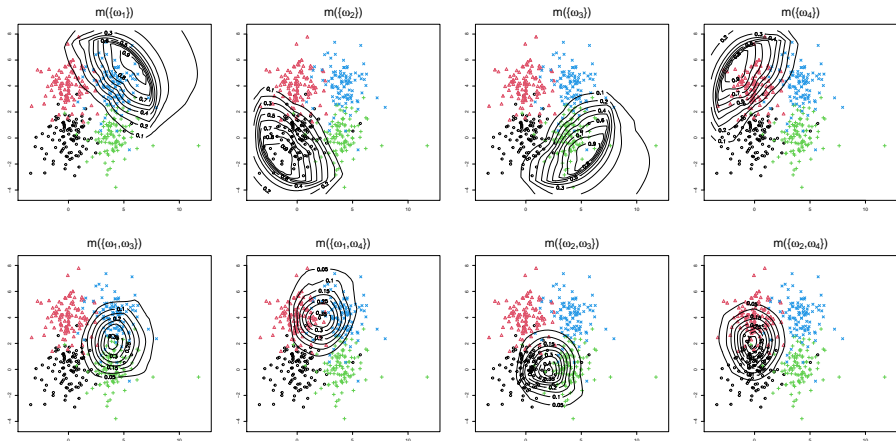
$$\mathcal{L}_R(\theta) = \frac{1}{np} \sum_{i=1}^n \sum_{j \in J(i)} \mathcal{L}_{ij}(\theta) + \frac{\lambda}{2} \left( \frac{1}{n_H(d+1)} \sum_{h,k} v_{hk}^2 + \frac{1}{f(n_H+1)} \sum_{q,h} w_{qh}^2 \right)$$

where  $J(i)$  is a random subset of  $\{1, \dots, n\}$  of cardinality  $p$ , and  $\lambda$  is a regularization hyperparameter.

# Example: one-class SVM output



# Example: output mass functions

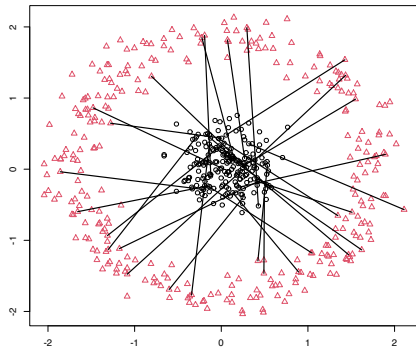
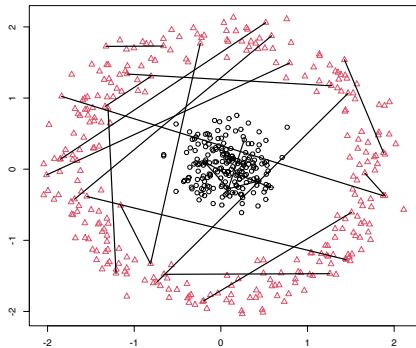


# Exploiting pairwise constraints

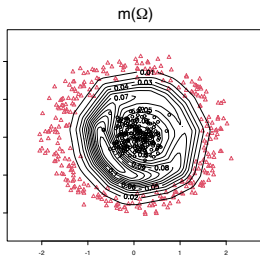
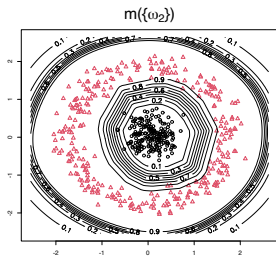
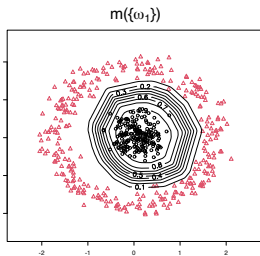
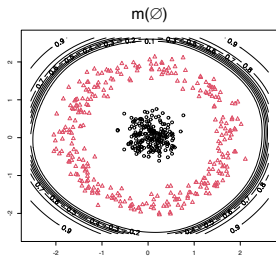
- Like CEVCLUS, NN-EVCLUS can exploit **pairwise (must-link and cannot-link) constraints** by adding penalty terms to the loss function.
- Better results are obtained by using the pairwise constraints for adapting the metric, using feature extraction methods such as (Kernelized) Pairwise Constrained Component Analysis.

▶ [Go back to CECM](#)

# Example: constraints



# Example: mass functions



# Outline

- 1 Belief functions in a space of partitions
  - Orthogonal sums of pairwise belief functions
  - Credal partitions
- 2 Evidential clustering algorithms
  - EK-NNclus
  - Evidential *c*-means
  - EVCLUS
  - NN-EVCLUS
  - **BootClus**

# Basic idea

- **Model-based clustering** allows us to estimate probabilities of cluster membership. The result is a fuzzy partition that describes **first-order uncertainty**.
- To represent **second-order uncertainty** (uncertainty about the probability estimates), we need a more general model.
- The BootClus algorithm<sup>13</sup> exploits the expressiveness of DS theory and generates a CP by **bootstrapping mixture models**.
- As it is built to approximate some confidence intervals, the resulting CP is **frequency-calibrated**.

---

<sup>13</sup>T. Denœux. Calibrated model-based evidential clustering using bootstrapping. *Information Sciences*, 528:17–45, 2020.



# Model

- We assume that the attribute vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$  are an iid random sample from a **mixture distribution** with pdf

$$p(\mathbf{x}; \theta) = \sum_{k=1}^c \pi_k p_k(\mathbf{x}; \theta_k)$$

where each component in the mixture corresponds to a cluster and  $\theta$  is the parameter vector.

- The probability that object  $i$  belongs to cluster  $k$  is

$$\pi_k(\mathbf{x}_i; \theta) = \frac{p_k(\mathbf{x}_i; \theta_k) \pi_k}{\sum_{\ell=1}^c p_\ell(\mathbf{x}_i; \theta_\ell) \pi_\ell}$$

- The probability that two objects  $i$  and  $j$  belong to the same cluster is

$$P_{ij}(\theta) = \sum_{k=1}^c \pi_k(\mathbf{x}_i; \theta) \pi_k(\mathbf{x}_j; \theta)$$

# Estimation

- Given a dataset  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we can compute the MLE  $\hat{\theta}$  of  $\theta$  and the corresponding MLEs  $\pi_k(\mathbf{x}_i; \hat{\theta})$  and  $P_{ij}(\hat{\theta})$ .
- To describe the uncertainty of these estimates, we can use the **bootstrap**.
- Confidence intervals on the pairwise probabilities  $P_{ij}(\theta)$  can easily be obtained by the **bootstrap percentile method**.

# Bootstrap CIs on pairwise probabilities

**Require:** Dataset  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , model  $p(\cdot; \theta)$ , number of bootstrap samples  $B$ , confidence level  $1 - \alpha$

**for**  $b = 1$  **to**  $B$  **do**

Draw  $\mathbf{x}_{b1}, \dots, \mathbf{x}_{bn}$  from  $\mathbf{x}_1, \dots, \mathbf{x}_n$  with replacement

Compute the MLE  $\hat{\theta}_b$  from  $\mathbf{x}_{b1}, \dots, \mathbf{x}_{bn}$

**for all**  $i < j$  **do**

Compute  $P_{ij}(\hat{\theta}_b)$

**end for**

**end for**

**for all**  $i < j$  **do**

$$P_{ij}^l := \text{Quantile} \left( \left\{ P_{ij}(\hat{\theta}_b) \right\}_{b=1}^B ; \frac{\alpha}{2} \right)$$

$$P_{ij}^u := \text{Quantile} \left( \left\{ P_{ij}(\hat{\theta}_b) \right\}_{b=1}^B ; 1 - \frac{\alpha}{2} \right)$$

**end for**

# Constructing a credal partition

- Given a **normalized CP**  $M = (m_1, \dots, m_n)$ , the belief and plausibility that any two objects  $i$  and  $j$  belong to the same cluster are given by

$$Bel_{ij} = \sum_{k=1}^c m_i(\{\omega_k\})m_j(\{\omega_k\})$$

$$Pl_{ij} = 1 - \kappa_{ij} = \sum_{A \cap B \neq \emptyset} m_i(A)m_j(B)$$

▶ Go back to pairwise mass functions

- Idea: search for a credal partition  $M$  such that the **belief-plausibility intervals**  $[Bel_{ij}, Pl_{ij}]$  approximate the confidence intervals  $[P_{ij}^l, P_{ij}^u]$ .

# Optimization problem and frequentist property

- An approximating CP can be found as the solution of the optimization problem

$$\min_M \sum_{i < j} (Bel_{ij} - P'_{ij})^2 + (P_{ij} - P''_{ij})^2,$$

which can be solved using a grouped coordinate descent procedure (solving a QP problem at each iteration).

- The solution verifies

$$P(Bel_{ij} \leq P_{ij}(\theta) \leq P_{ij}) \approx 1 - \alpha.$$

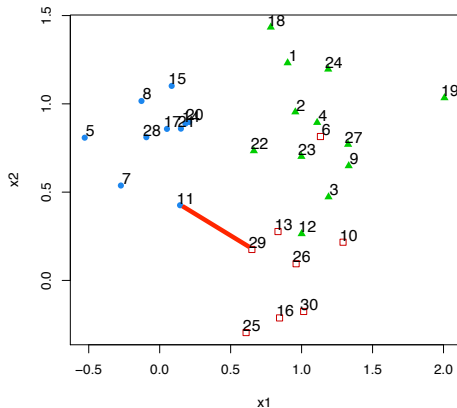
- This corresponds to the definition of a **predictive belief function**<sup>14</sup> at level  $1 - \alpha$ , a special kind of **frequency-calibrated belief function**<sup>15</sup>.

<sup>14</sup>T. Denœux. Constructing Belief Functions from Sample Data Using Multinomial Confidence Regions. *IJAR*, 42(3):228–252, 2006.

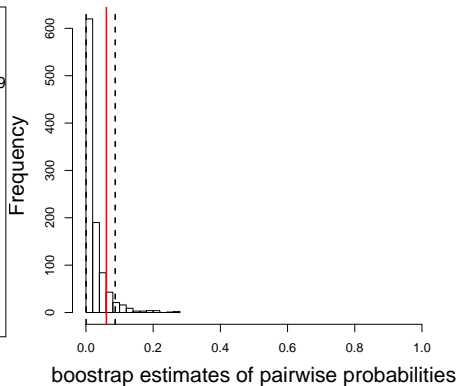
<sup>15</sup>T. Denœux and S. Li. Frequency-Calibrated Belief Functions: Review and New Insights. *IJAR*, 92:232–254, 2018.

# Example

## Bootstrap confidence intervals

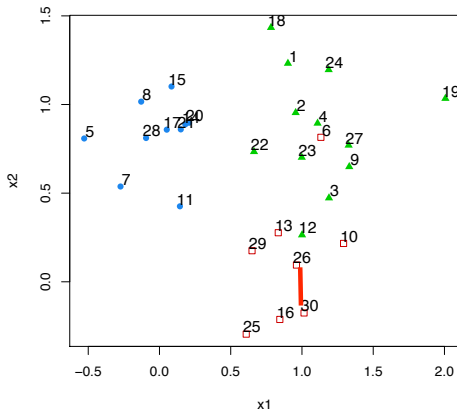


11, 29

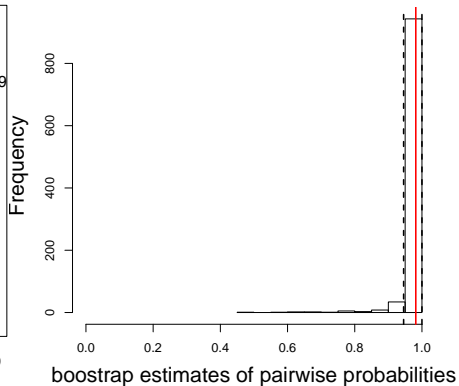


# Example

## Bootstrap confidence intervals

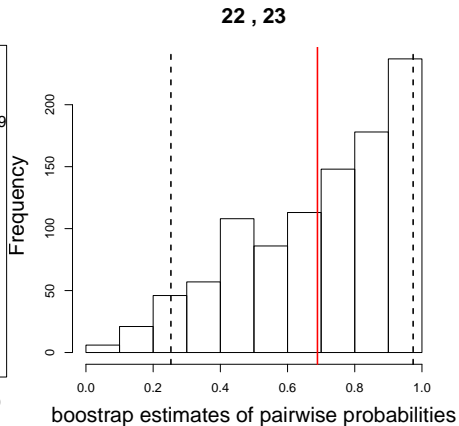
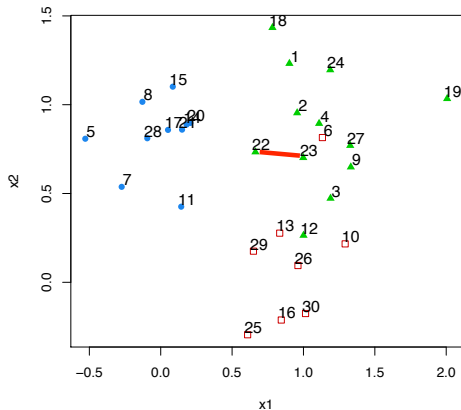


26 , 30



# Example

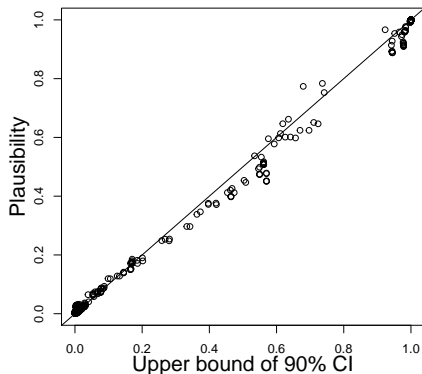
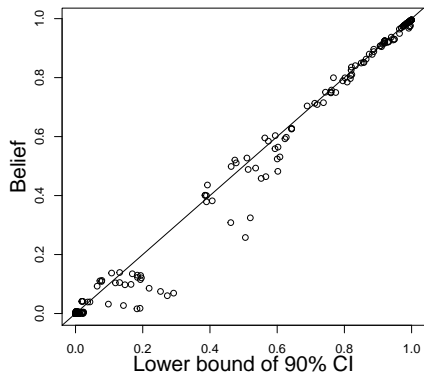
## Bootstrap confidence intervals





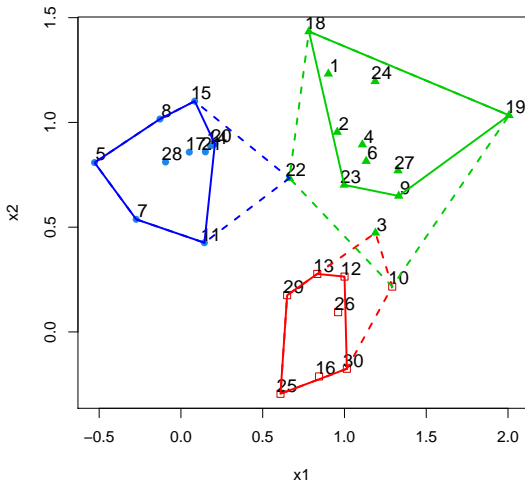
# Example

Approximation of confidence intervals by pairwise belief functions



# Example

Lower/upper approximations of the credal partition



# Summary

- Quantifying clustering uncertainty using DS theory implies defining belief functions on the space of all partitions (or on the space of partitions with at most  $c$  clusters).
- Two useful workable models are (1) orthogonal sums of pairwise mass functions, and (2) CPs.
- In particular, CPs generalize not only hard partitions, but also other “soft” clustering structures (hard, fuzzy, possibilistic, rough partitions).
- Several evidential clustering algorithms have been proposed, most of them generating CPs.

## Summary (continued)

- Some algorithms (ECM, BootClus) handle attribute data only, while others (EK-NNclus, EVCLUS) can also handle (nonmetric) proximity data.
- Datasets with a large number of clusters can be handled by carefully selecting the focal sets.
- The evaluation and comparison of soft clustering algorithms in the belief function framework is a current research topic<sup>161718</sup>.

---

<sup>16</sup>T. Denœux, S. Li and S. Sriboonchitta. Evaluating and Comparing Soft Partitions: an Approach Based on Dempster-Shafer Theory. *IEEE Transactions on Fuzzy Systems*, 26(3):1231–1244, 2018.

<sup>17</sup>A. Campagner, D. Ciucci and T. Denœux. A General Framework for Evaluating and Comparing Soft Clusterings. *Information Sciences*, 623:70–93, 2023.

<sup>18</sup>A. Campagner, D. Ciucci and T. Denœux. A Distributional Framework for Evaluation, Comparison and Uncertainty Quantification in Soft Clustering. *IJAR*, 162:109008, 2023.