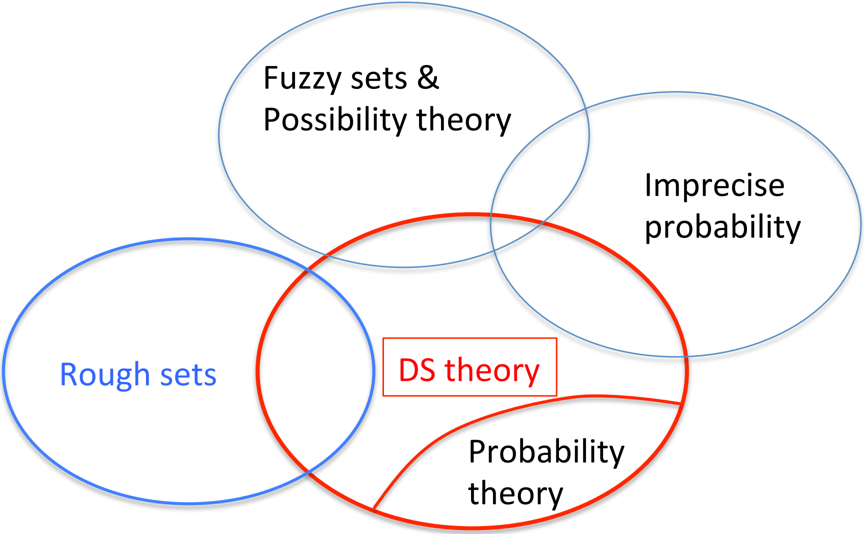
Classification and clustering using Belief functions

Thierry Denœux[1]

[1]Université de Technologie de Compiègne
HEUDIASYC (UMR CNRS 6599)
http://www.hds.utc.fr/~tdenoeux

Tongji University
Shanghai, China
July 7, 2016

Theories of uncertainty



Focus of this talk

- Dempster-Shafer (DS) theory (evidence theory, theory of belief functions):
    - A formal framework for reasoning with partial (uncertain, imprecise) information.
    - Has been applied to statistical inference, expert systems, information fusion, classification, clustering, etc.
- Purpose of these talk:
    - Brief introduction or reminder on DS theory;
    - Review the application of belief functions to classification and clustering.

Outline

1. Dempster-Shafer theory
   - Mass, belief and plausibility functions
   - Dempster's rule
   - Decision analysis
2. Evidential classification
   - Evidential K-NN rule
   - Evidential neural network classifier
   - Decision analysis
3. Application to clustering
   - credal partition
   - Evidential c-means
   - EVCLUS
   - EK-NNclus
   - Handling a large number of clusters

Outline

1. Dempster-Shafer theory
   - Mass, belief and plausibility functions
   - Dempster's rule
   - Decision analysis
2. Evidential classification
   - Evidential K-NN rule
   - Evidential neural network classifier
   - Decision analysis
3. Application to clustering
   - credal partition
   - Evidential c-means
   - EVCLUS
   - EK-NNclus
   - Handling a large number of clusters

Outline

Mass function

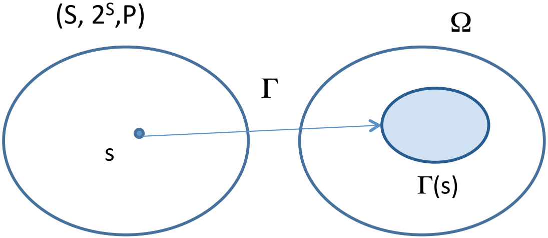
- Let Ω be a finite set called a frame of discernment.
- A mass function is a function $m : 2^{\Omega} \rightarrow [0, 1]$ such that

$$\sum_{A \subseteq \Omega} m(A) = 1.$$

- The subsets A of Ω such that $m(A) \neq 0$ are called the focal sets of m.
- If $m(\emptyset) = 0$, m is said to be normalized (usually assumed).

Source

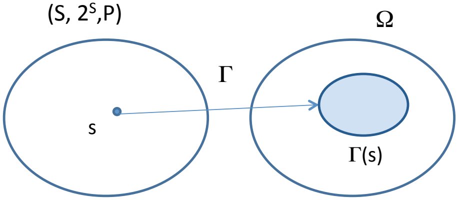
- A mass function is usually induced by a source, defined a 4-tuple $(S, 2^S, P, \Gamma)$, where
  - S is a finite set;
  - P is a probability measure on $(S, 2^S)$;
  - Γ is a multi-valued-mapping from S to 2^Ω.



- Γ carries P from S to 2^Ω: for all $A \subseteq \Omega$,

$$m(A) = P(\{s \in S | \Gamma(s) = A\}).$$

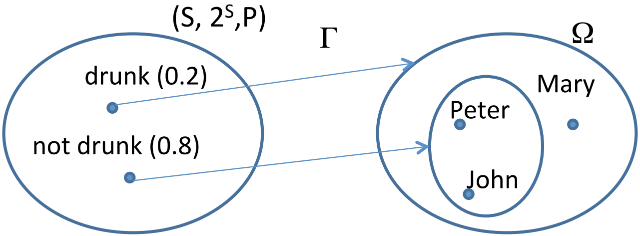
Interpretation



- Ω is a set of possible states of the world, about which we collect some evidence. Let ω be the true state.
- S is a set of interpretations of the evidence.
- If $s \in S$ holds, we know that ω belongs to the subset $\Gamma(s)$ of Ω, and nothing more.
- $m(A)$ is then the probability of knowing only that $\omega \in A$.
- In particular, $m(\Omega)$ is the probability of knowing nothing.

Example

- A murder has been committed. There are three suspects: $\Omega = \{\text{Peter}, \text{John}, \text{Mary}\}$.
- A witness saw the murderer going away, but he is short-sighted and he only saw that it was a man. We know that the witness is drunk 20 % of the time.



- We have $\Gamma(\neg\text{drunk}) = \{\text{Peter}, \text{John}\}$ and $\Gamma(\text{drunk}) = \Omega$, hence

$$m(\{\text{Peter}, \text{John}\}) = 0.8, \quad m(\Omega) = 0.2$$

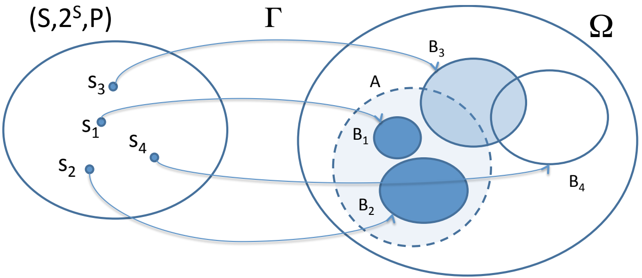
Special cases

- A mass function m is said to be:
  - logical if it has only one focal set; it is then equivalent to a set.
  - Bayesian if all focal sets are singletons; it is equivalent to a probability distribution.
- A mass function can thus be seen as
  - a generalized set, or as
  - a generalized probability distribution.

Belief function

Degrees of support and consistency

- Let m be a normalized mass function on Ω induced by a source $(S, 2^S, P, \Gamma)$.
- Let A be a subset of Ω.
- One may ask:
  1. To what extent does the evidence support the proposition $\omega \in A$?
  2. To what extent is the evidence consistent with this proposition?

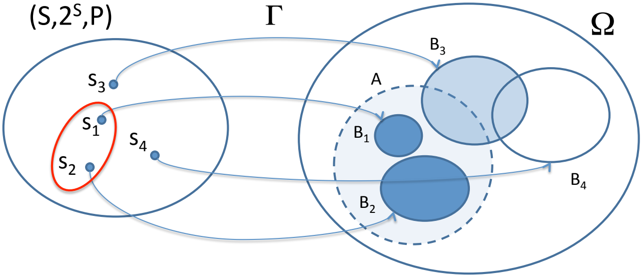


Belief function

Definition and interpretation

- For any $A \subseteq \Omega$, the probability that the evidence implies (supports) the proposition $\omega \in A$ is

$$Bel(A) = P(\{s \in S | \Gamma(s) \subseteq A\}) = \sum_{B \subseteq A} m(B).$$



- The function $Bel : A \to Bel(A)$ is called a belief function.

Belief function

Characterization

- Function $Bel : 2^\Omega \rightarrow [0,1]$ is a completely monotone capacity: it verifies $Bel(\emptyset)=0$, $Bel(\Omega)=1$ and

$$Bel\left(\bigcup_{i=1}^{k} A_i\right) \geq \sum_{\emptyset \neq I \subseteq \{1,\ldots,k\}} (-1)^{|I|+1} Bel\left(\bigcap_{i \in I} A_i\right).$$

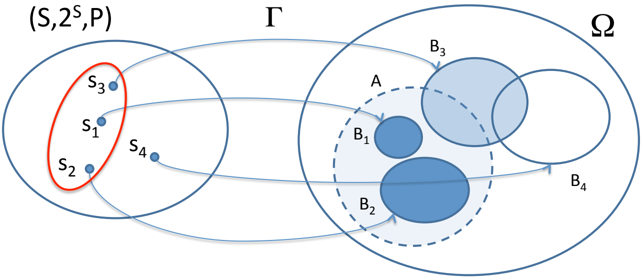
  for any $k \geq 2$ and for any family $A_1, \ldots, A_k$ in 2^Ω.
- Conversely, to any completely monotone capacity *Bel* corresponds a unique mass function m such that:

$$m(A) = \sum_{\emptyset \neq B \subseteq A} (-1)^{|A|-|B|} Bel(B), \quad \forall A \subseteq \Omega.$$

Plausibility function

- The probability that the evidence is consistent with (does not contradict) the proposition $\omega \in A$

$$Pl(A) = P(\{s \in S | \Gamma(s) \cap A \neq \emptyset\}) = 1 - Bel(\overline{A})$$



- The function $Pl : A \rightarrow Pl(A)$ is called a plausibility function.

Special cases

- If m is Bayesian, then $Bel = Pl$ and it is a probability measure.
- If the focal sets of m are nested ($A_1 \subset A_2 \subset \ldots \subset A_n$), m is said to be consonant. Pl is then a possibility measure:

$$Pl(A \cup B) = \max\left(Pl(A), Pl(B)\right)$$

  for all $A, B \subseteq \Omega$ and Bel is the dual necessity measure.
- DS theory thus subsumes both probability theory and possibility theory.

Summary

- A probability measure is precise, in so far as it represents the uncertainty of the proposition $\omega \in A$ by a single number $P(A)$.
- In contrast, a mass function is imprecise (it assigns probabilities to subsets).
- As a result, in DS theory, the uncertainty about a subset A is represented by two numbers $(Bel(A), Pl(A))$, with $Bel(A) \leq Pl(A)$.
- This model has some connections with rough set theory, in which a set is approximated by lower and upper approximations, due to coarseness of a knowledge base.

Outline

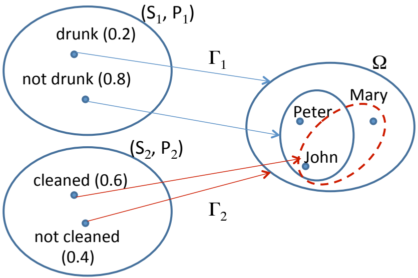
Dempster's rule

Murder example continued

- The first item of evidence gave us: $m_1(\{Peter, John\}) = 0.8$, $m_1(\Omega) = 0.2$.
- New piece of evidence: a blond hair has been found.
- There is a probability 0.6 that the room has been cleaned before the crime: $m_2(\{John, Mary\}) = 0.6$, $m_2(\Omega) = 0.4$.
- How to combine these two pieces of evidence?

Dempster's rule

Justification



- If interpretations $s_1 \in S_1$ and $s_2 \in S_2$ both hold, then $X \in \Gamma_1(s_1) \cap \Gamma_2(s_2)$.
- If the two pieces of evidence are independent, then the probability that s_1 and s_2 both hold is $P_1(\{s_1\})P_2(\{s_2\})$.
- If $\Gamma_1(s_1) \cap \Gamma_2(s_2) = \emptyset$, we know that s_1 and s_2 cannot hold simultaneously.
- The joint probability distribution on $S_1 \times S_2$ must be conditioned to eliminate such pairs.

Dempster's rule

Definition

- Let m_1 and m_2 be two mass functions and

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$$

their degree of conflict.

- If $K < 1$, then m_1 and m_2 can be combined as

$$(m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{B \cap C = A} m_1(B) m_2(C), \quad \forall A \neq \emptyset,$$

and $(m_1 \oplus m_2)(\emptyset) = 0$.

Dempster's rule

Properties

- Commutativity, associativity. Neutral element: m_Ω.
- Generalization of intersection: if m_A and m_B are categorical mass functions and $A \cap B \neq \emptyset$, then

$$m_A \oplus m_B = m_{A \cap B}$$

- Generalization of probabilistic conditioning: if m is a Bayesian mass function and m_A is a logical mass function, then $m \oplus m_A$ is a Bayesian mass function corresponding to the conditioning of m by A.
- Notation for conditioning (special case):

$$m \oplus m_A = m(\cdot|A).$$

Outline

Problem formulation

- A decision problem can be formalized by defining:
  - A set of acts $\mathcal{A} = \{a_1, \ldots, a_s\}$;
  - A set of states of the world Ω;
  - A loss function $L : \mathcal{A} \times \Omega \rightarrow \mathbb{R}$, such that $L(a, \omega)$ is the loss incurred if we select act a and the true state is ω.
- Bayesian framework
  - Uncertainty on Ω is described by a probability measure P;
  - Define the risk of each act a as the expected loss if a is selected:

$$R_P(a) = \mathbb{E}_P[L(a, \cdot)] = \sum_{\omega \in \Omega} L(a, \omega) P(\{\omega\}).$$

  - Select an act with minimal risk.
- Extension when uncertainty on Ω is described by a belief function?

Lower and upper expected risk

- Let m be a normalized mass function, and $\mathcal{P}(m)$ its credal set, defined as the set of probability measures on Ω such that

$$Bel(A) \leq P(A) \leq Pl(A), \quad \forall A \subseteq \Omega.$$

- The lower and upper risk of each act a are defined, respectively, as:

$$\underline{R}(a) = \underline{\mathbb{E}}_m[L(a,\cdot)] = \inf_{P \in \mathcal{P}(m)} R_P(a) = \sum_{A \subseteq \Omega} m(A) \min_{\omega \in A} L(a,\omega)$$

$$\overline{R}(a) = \overline{\mathbb{E}}_m[L(a,\cdot)] = \sup_{P \in \mathcal{P}(m)} R_P(a) = \sum_{A \subseteq \Omega} m(A) \max_{\omega \in A} L(a,\omega)$$

Decision strategies

- For each act a we have a risk interval $[\underline{R}(a), \overline{R}(a)]$. How to compare these intervals?
- Three strategies:
  1. a is preferred to a' iff $\underline{R}(a) \leq \underline{R}(a')$ (optimistic strategy)
  2. a is preferred to a' iff $\overline{R}(a) \leq \overline{R}(a')$ (pessimistic strategy)
  3. a is preferred to a' iff $\overline{R}(a) \leq \underline{R}(a')$ (interval dominance);
- The interval dominance strategy yields only a partial preorder:
  - a and a' are not comparable if $\overline{R}(a) > \underline{R}(a')$ and $\overline{R}(a') > \underline{R}(a)$
  - We can consider the set of non dominated acts (the set of acts a such that no act is strictly preferred to a)

Other decision strategies

How to find a compromise between the pessimistic and optimistic strategies? Two approaches:

1. Hurwicz criterion: a is preferred to a' iff $R_\rho(a) \leq R_\rho(a')$ with

$$R_\rho(a) = (1-\rho)\underline{R}(a) + \rho\overline{R}(a).$$

   and $\rho \in [0,1]$ is a pessimism index describing the attitude of the decision maker in the face of ambiguity.

2. Minimize the risk with respect to the pignistic probability measure P_m, defined from m by the probability mass function

$$p_m(\omega) = \sum_{B \ni \omega} \frac{m(B)}{|B|}, \quad \forall \omega \in \Omega.$$

   It can be shown that $P_m \in \mathcal{P}(m)$. Consequently,

$$\underline{R}(a) \leq R_{P_m}(a) \leq \overline{R}(a), \quad \forall a \in \mathcal{A}.$$

Decision making

Example

- Let $m(\{John\}) = 0.48$, $m(\{John, Mary\}) = 0.12$, $m(\{Peter, John\}) = 0.32$, $m(\Omega) = 0.08$.
- We have

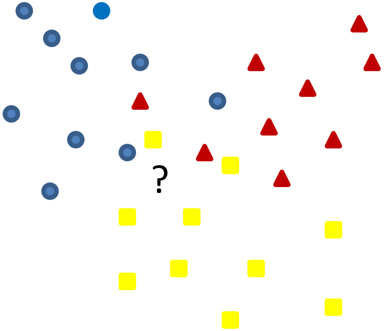
$$p_m(John) = 0.48 + \frac{0.12}{2} + \frac{0.32}{2} + \frac{0.08}{3} \approx 0.73,$$

$$p_m(Peter) = \frac{0.32}{2} + \frac{0.08}{3} \approx 0.19$$

$$p_m(Mary) = \frac{0.12}{2} + \frac{0.08}{3} \approx 0.09$$

Outline

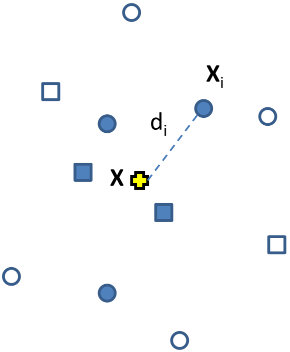
Classification problem



- A population is assumed to be partitioned in c groups or classes
- Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ denote the set of classes
- Each instance is described by
  - A feature vector $\boldsymbol{x} \in \mathbb{R}^p$
  - A class label $y \in \Omega$
- Problem: given a learning set $\mathcal{L} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, predict the class label of a new instance described by $\boldsymbol{x}$

Outline

Principle



- Let $\mathcal{N}_K(\boldsymbol{x}) \subset \mathcal{L}$ denote the set of the K nearest neighbors of $\boldsymbol{x}$ in $\mathcal{L}$, based on some distance measure
- Each $\boldsymbol{x}_i \in \mathcal{N}_K(\boldsymbol{x})$ can be considered as a piece of evidence regarding the class of $\boldsymbol{x}$
- The strength of this evidence decreases with the distance d_i between $\boldsymbol{x}$ and $\boldsymbol{x}_i$

Definition

- If $y_i = \omega_k$, the evidence of $(\boldsymbol{x}_i, y_i)$ can be represented by

$$\begin{aligned} m_i(\{\omega_k\}) &= \varphi_k\left(d_i\right) \\ m_i(\{\omega_\ell\}) &= 0, \quad \forall \ell \neq k \\ m_i(\Omega) &= 1 - \varphi\left(d_i\right) \end{aligned}$$

  where φ_k, $k = 1, \ldots, c$ are decreasing functions from $[0, +\infty)$ to $[0, 1]$ such that $\lim_{d \to +\infty} \varphi_k(d) = 0$

- The evidence of the K nearest neighbors of $\boldsymbol{x}$ is pooled using Dempster's rule of combination

$$m = \bigoplus_{\boldsymbol{x}_i \in \mathcal{N}_K(\boldsymbol{x})} m_i$$

- Decision: any of the decision rules mentioned in the first part.
- With 0-1 losses and no rejection, the optimistic, pessimistic and pignistic rules yield the same decisions.

Learning

- Choice of functions φ_k: for instance, $\varphi_k(d) = \alpha \exp(-\gamma_k d^2)$.
- Parameters $\gamma_1, \ldots, \gamma_c$ can be optimized (see below).
- Parameter $\gamma = (\gamma_1, \ldots, \gamma_c)$ can be learnt from the data by minimizing the following cost function

$$C(\gamma) = \sum_{i=1}^{n} \sum_{k=1}^{c} (pl_{(-i)}(\omega_k) - t_{ik})^2,$$

  where
  - $pl_{(-i)}$ is the contour function obtained by classifying $\boldsymbol{x}_i$ using its K nearest neighbors in the learning set.
  - $t_{ik} = 1$ is $y_i = k$, $t_{ik} = 0$ otherwise.
- Function $C(\gamma)$ can be minimized by an iterative nonlinear optimization algorithm.

Computation of $pl_{(-i)}$

- Contour function from each neighbor $\boldsymbol{x}_j \in \mathcal{N}_K(\boldsymbol{x}_i)$:

$$pl_j(\omega_k) = \begin{cases} 1 & \text{if } y_j = \omega_k \\ 1 - \varphi_k(d_{ij}) & \text{otherwise} \end{cases}, \quad k = 1, \ldots, c$$

- Contour function of the combined mass function

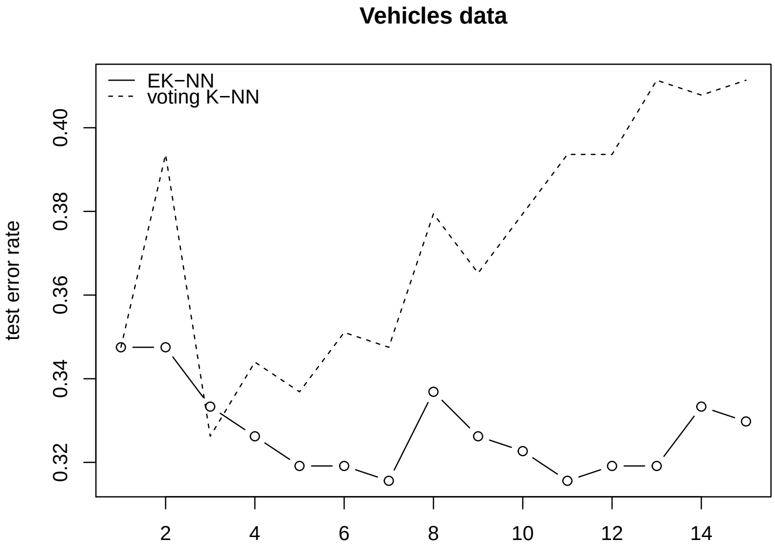
$$pl_{(-i)}(\omega_k) \propto \prod_{\boldsymbol{x}_j \in \mathcal{N}_K(\boldsymbol{x}_i)} (1 - \varphi_k(d_{ij}))^{1-t_{jk}}$$

  where $t_{jk} = 1$ if $y_j = \omega_k$ and $t_{jk} = 0$ otherwise
- It can be computed in time proportional to $K|\Omega|$

Example 1: Vehicles dataset

- The data were used to distinguish 3D objects within a 2-D silhouette of the objects.
- Four classes: bus, Chevrolet van, Saab 9000 and Opel Manta.
- 846 instances, 18 numeric attributes.
- The first 564 objects are training data, the rest are test data.

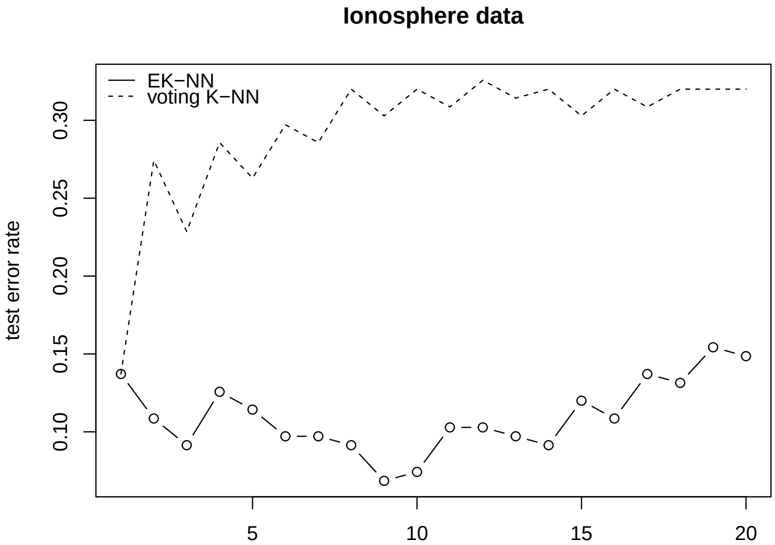
Vehicles datasets: result



Example 2: Ionosphere dataset

- This dataset was collected by a radar system and consists of phased array of 16 high-frequency antennas with a total transmitted power of the order of 6.4 kilowatts.
- The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not.
- There are 351 instances and 34 numeric attributes. The first 175 instances are training data, the rest are test data.

Ionosphere datasets: result



Implementation in R

```
library("evclass")

data("ionosphere")
xapp<-ionosphere$x[1:176,]
yapp<-ionosphere$y[1:176]
xtst<-ionosphere$x[177:351,]
ytst<-ionosphere$y[177:351]

opt<-EkNNfit(xapp,yapp,K=10)
class<-EkNNval(xapp,yapp,xtst,K=10,ytst,opt$param)

> class$err
0.07428571
> table(ytst,class$ypred)
ytst 1 2
1 106 6
2 7 56
```

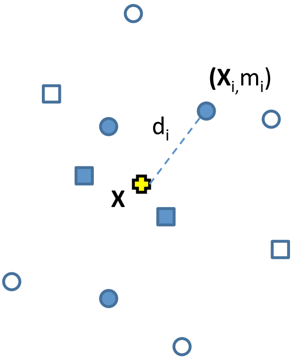
Partially supervised data

- We now consider a learning set of the form

$$\mathcal{L} = \{(\boldsymbol{x}_i, m_i), i = 1, \ldots, n\}$$

  where
  - $\boldsymbol{x}_i$ is the attribute vector for instance i, and
  - m_i is a mass function representing uncertain expert knowledge about the class y_i of instance i
- Special cases:
  - $m_i(\{\omega_k\}) = 1$ for all i: supervised learning
  - $m_i(\Omega) = 1$ for all i: unsupervised learning

Evidential k-NN rule for partially supervised data



- Each mass function m_i is discounted (weakened) with a rate depending on the distance d_i

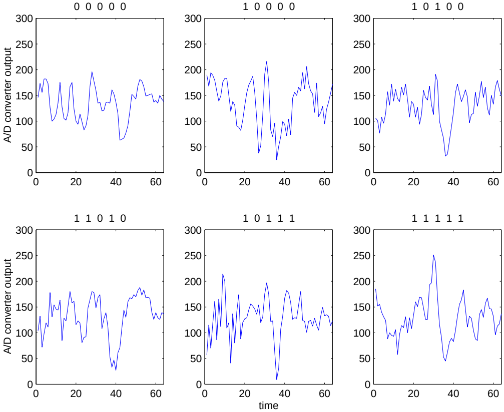
$$m'_i(A) = \varphi(d_i)\, m_i(A), \quad \forall A \subset \Omega$$
$$m'_i(\Omega) = 1 - \sum_{A \subset \Omega} m'_i(A)$$

- The K mass functions m'_i are combined using Dempster's rule

$$m = \bigoplus_{\boldsymbol{x}_i \in \mathcal{N}_K(\boldsymbol{x})} m'_i$$

Example: EEG data

EEG signals encoded as 64-D patterns, 50 % positive (K-complexes), 50 % negative (delta waves), 5 experts.



Results on EEG data

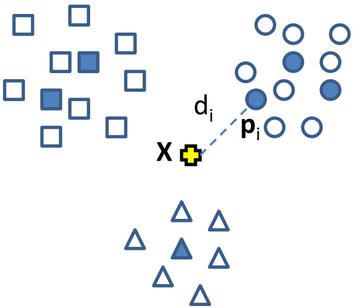
(Denoeux and Zouhal, 2001)

- $c = 2$ classes, $p = 64$
- For each learning instance $\boldsymbol{x}_i$, the expert opinions were modeled as a mass function m_i.
- $n = 200$ learning patterns, 300 test patterns

K	K-NN	w K-NN	Ev. K-NN (crisp labels)	Ev. K-NN (uncert. labels)
9	0.30	0.30	0.31	0.27
11	0.29	0.30	0.29	0.26
13	0.31	0.30	0.31	0.26

Outline

Principle



- The learning set is summarized by r prototypes.
- Each prototype $\boldsymbol{p}_i$ has membership degree u_{ik} to each class ω_k, with $\sum_{k=1}^{c} u_{ik} = 1$.
- Each prototype $\boldsymbol{p}_i$ is a piece of evidence about the class of $\boldsymbol{x}$, whose reliability decreases with the distance d_i between $\boldsymbol{x}$ and $\boldsymbol{p}_i$.

Propagation equations

- Mass function induced by prototype $\boldsymbol{p}_i$:

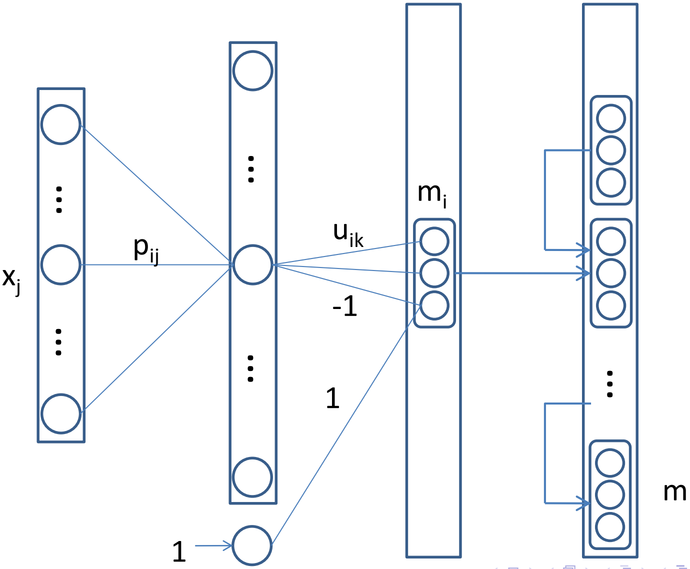
$$
\begin{aligned}
m_i(\{\omega_k\}) &= \alpha_i u_{ik} \exp(-\gamma_i d_i^2), \quad k = 1, \ldots, c \\
m_i(\Omega) &= 1 - \alpha_i \exp(-\gamma_i d_i^2)
\end{aligned}
$$

- Combination:

$$
m = \bigoplus_{i=1}^{r} m_i
$$

- The computation of m_i requires $O(rp)$ arithmetic operations (where p denotes the number of inputs), and the combination can be performed in $O(rc)$ operations. Hence, the overall complexity is $O(r(p+c))$ operations to compute the output for one input pattern.
- The combined mass function m has as focal sets the singletons $\{\omega_k\}$, $k = 1, \ldots, c$ and Ω.

Neural network implementation



Learning

- The parameters are the
  - The prototypes $\boldsymbol{p}_i$, $i = 1, \ldots, r$ (rp parameters)
  - The membership degrees u_{ik}, $i = 1, \ldots, r$, $k = 1 \ldots, c$ (rc parameters)
  - The α_i and γ_i, $i = 1 \ldots, r$ ($2r$ parameters).
- Let $\boldsymbol{\theta}$ denote the vector of all parameters. It can be estimated by minimizing a cost function such as

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{n} (pl_{ik} - t_{ik})^2 + \mu \sum_{i=1}^{r} \alpha_i$$

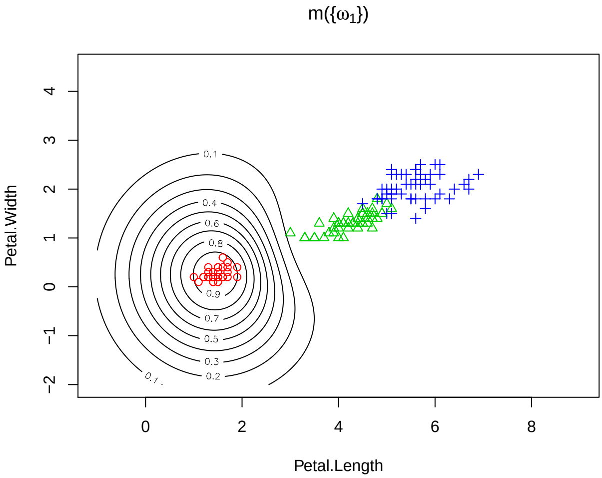
  where pl_{ik} is the output plausibility for instance i and class k, $t_{ik} = 1$ if $y_i = k$ and $t_{ik} = 0$ otherwise, and μ is a regularization coefficient (hyperparameter).
- The hyperparameter μ can be optimized by cross-validation.

Implementation in R

```
library("evclass")

data(glass)
xtr<-glass$x[1:89,]
ytr<-glass$y[1:89]
xtst<-glass$x[90:185,]
ytst<-glass$y[90:185]

param0<-proDSinit(xtr,ytr,nproto=7)
fit<-proDSfit(x=xtr,y=ytr,param=param0)
val<-proDSval(xtst,fit$param,ytst)

> print(val$err)
0.3333333 > table(ytst,val$ypred)
ytst 1 2 3 4
1 30 6 4 0
2 6 27 1 3
3 4 3 1 0
4 0 5 0 6
```

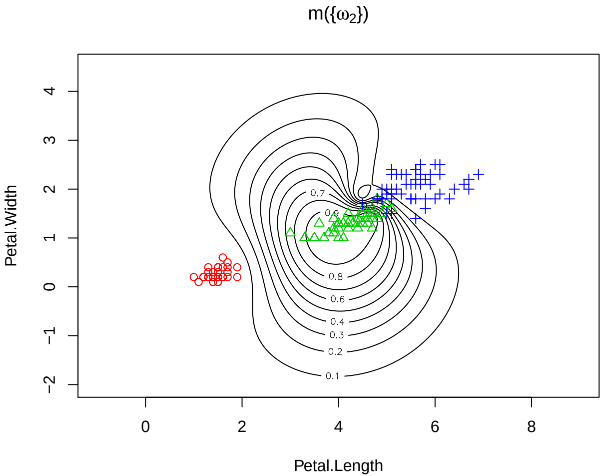
Results on the Iris data

Mass on $\{\omega_1\}$



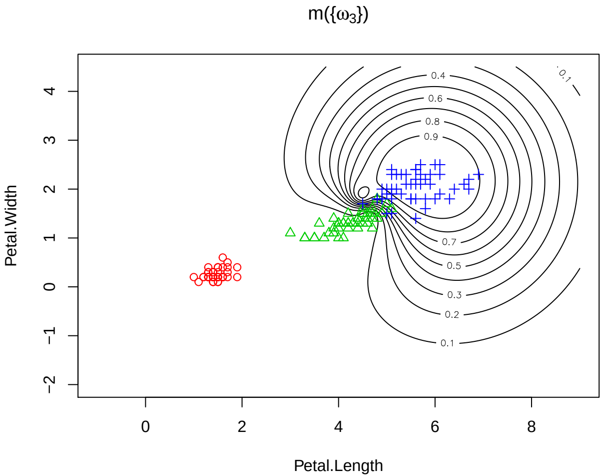
Results on the Iris data

Mass on $\{\omega_2\}$



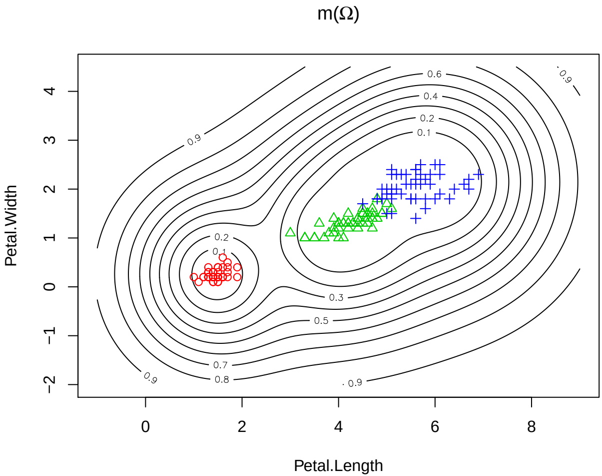
Results on the Iris data

Mass on $\{\omega_3\}$



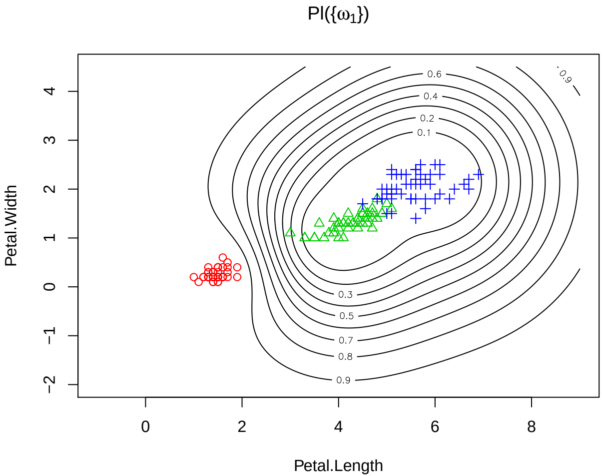
Results on the Iris data

Mass on Ω



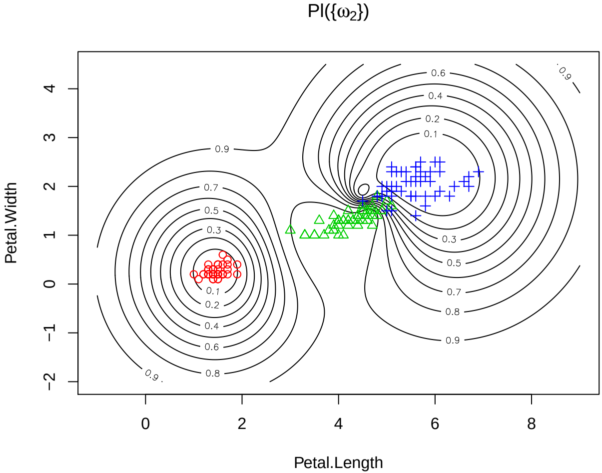
Results on the Iris data

Plausibility of $\{\omega_1\}$



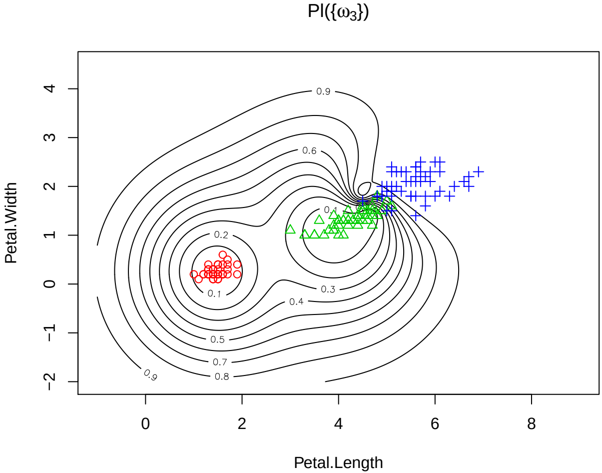
Results on the Iris data

Plausibility of $\{\omega_2\}$



Results on the Iris data

Plausibility of $\{\omega_3\}$



Outline

Simple decision setting

- To formalize the decision problem, we need to define:
  - The acts
  - The loss matrix
- For instance, let the acts be
  - $a_k =$ assignment to class ω_k, $k = 1, \ldots, c$
- And the loss matrix (for $c = 3$)

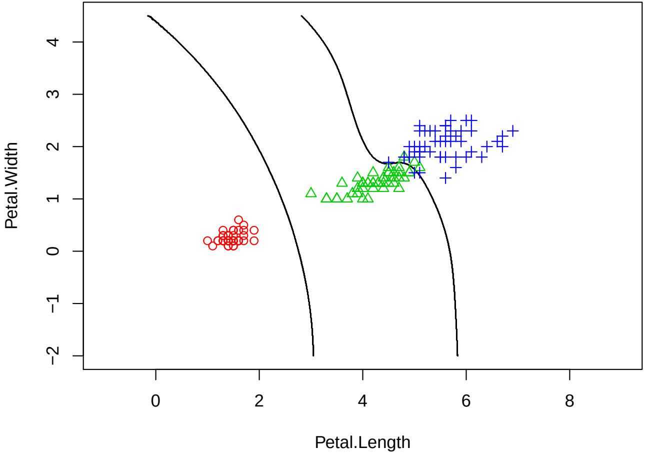
	a_1	a_2	a_3
ω_1	0	1	1
ω_2	1	0	1
ω_3	1	1	0

- $\underline{R}(a_i) = 1 - Pl(\{\omega_i\})$ and $\overline{R}(a_i) = 1 - Bel(\{\omega_i\})$.
- The optimistic, pessimistic and pignistic decision rules yield the same result

Implementation in R

```
param0<-proDSinit(x,y,6)
fit<-proDSfit(x,y,param0)

val<-proDSval(xtst,fit$param)
L<-1-diag(c)
D<-decision(val$m,L=L,rule='upper')
```

Decision regions (Iris data)



Decision with rejection

- Let the acts now be
  - a_k = assignment to class ω_k, $k = 1, \ldots, c$
  - a_0 = rejection
- And the loss matrix (for $c = 3$)

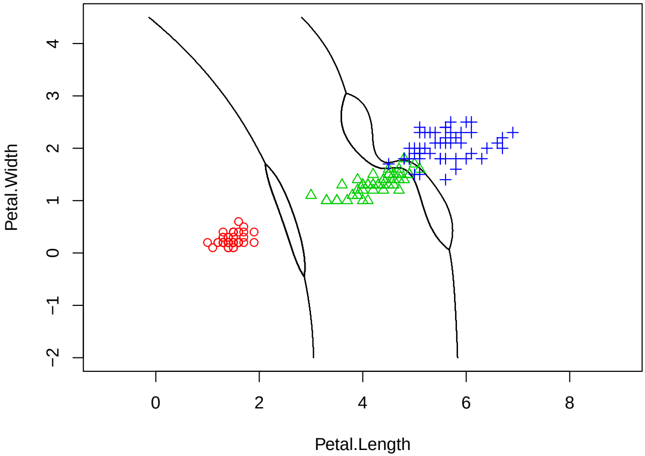
	a_1	a_2	a_3	a_0
ω_1	0	1	1	λ_0
ω_2	1	0	1	λ_0
ω_3	1	1	0	λ_0

Implementation in R

```
param0<-proDSinit(x,y,6)
fit<-proDSfit(x,y,param0)

val<-proDSval(xtst,fit$param)
L<-cbind(1-diag(c),rep(0.3,c))
D1<-decision(val$m,L=L,rule='upper')
D2<-decision(val$m,L=L,rule='lower')
D3<-decision(val$m,L=L,rule='pignistic')
D4<-decision(val$m,L=L,rule='hurwicz',rho=0.5)
```

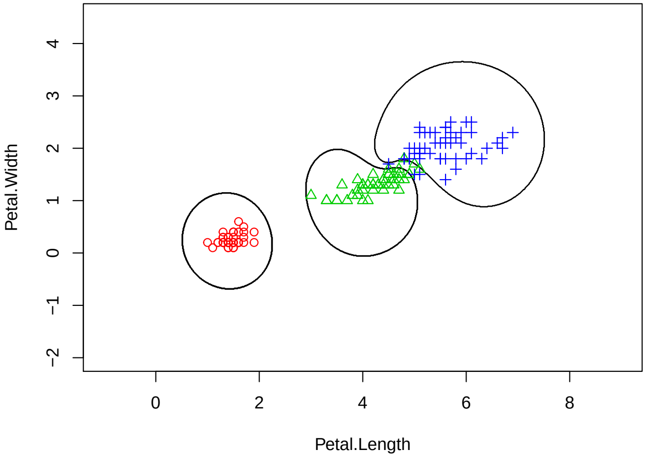
Decision regions (Iris data)

Lower risk



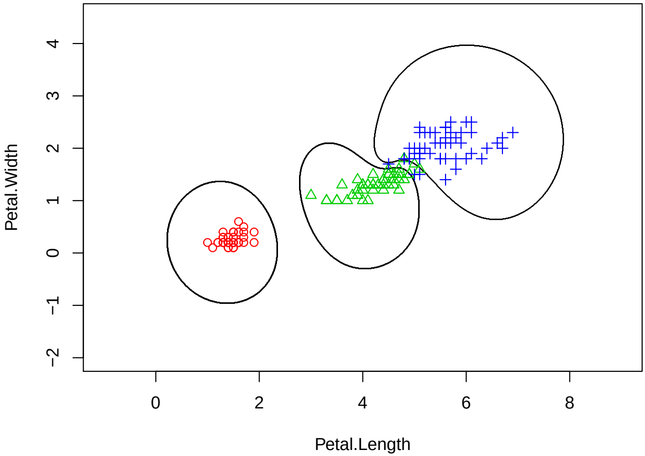
Decision regions (Iris data)

Upper risk



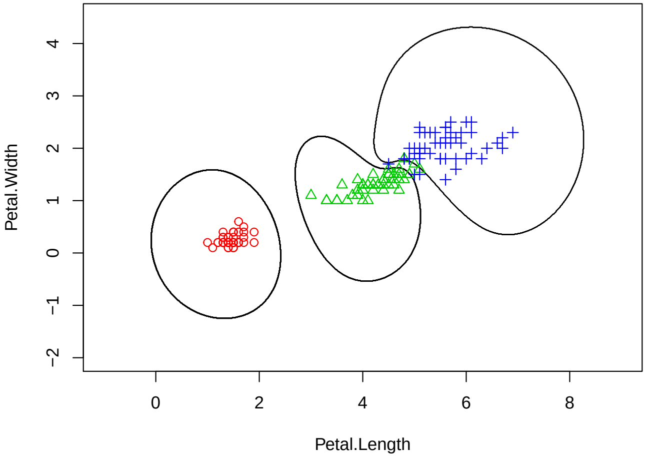
Decision regions (Iris data)

Pignistic risk



Decision regions (Iris data)

Hurwicz strategy ($\rho = 0.5$)



Decision with rejection and novelty detection

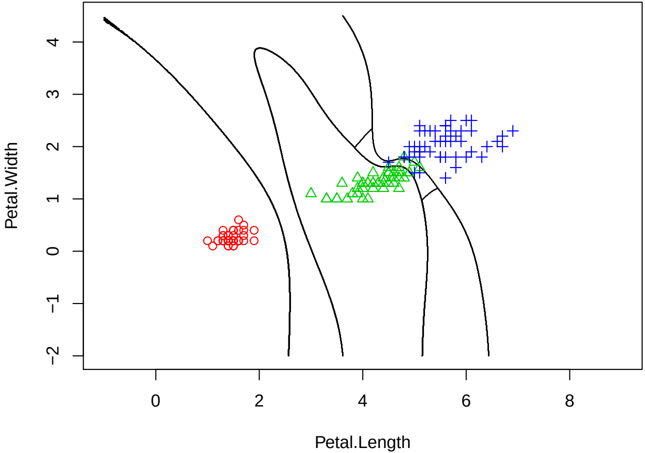
- Assume that there exists an unknown class ω_u, not represented in the learning set
- Let the acts now be
  - a_k = assignment to class ω_k, $k = 1, \ldots, c$
  - a_u = assignment to class ω_u
  - a_0 = rejection
- And the loss matrix

	a_1	a_2	a_3	a_0	a_u
ω_1	0	1	1	λ_0	λ_u
ω_2	1	0	1	λ_0	λ_u
ω_3	1	1	0	λ_0	λ_u
ω_u	1	1	1	λ_0	0

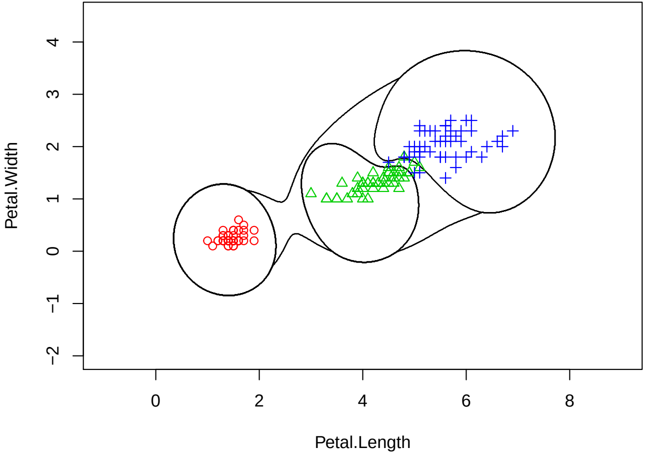
Implementation in R

```
param0<-proDSinit(x,y,6)
fit<-proDSfit(x,y,param0)

val<-proDSval(xtst,fit$param)
L<-cbind(1-diag(c),rep(0.3,c),rep(0.32,c))
L<-rbind(L,c(1,1,1,0.3,0))
D1<-decision(val$m,L=L,rule='lower')
D2<-decision(val$m,L=L,rule='pignistic')
D3<-decision(val$m,L=L,rule='hurwicz',rho=0.5)
```

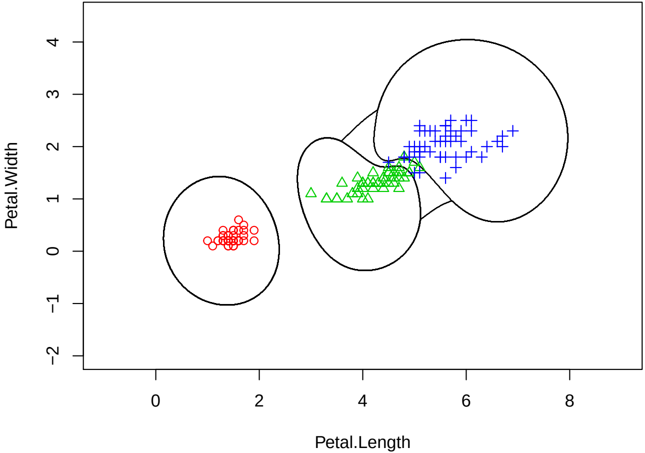
Decision regions (Iris data)



Decision regions (Iris data)



Decision regions (Iris data)



References on classification I

cf. https://www.hds.utc.fr/~tdenoeux

T. Denœux.
A k-nearest neighbor classification rule based on Dempster-Shafer theory.
IEEE Transactions on SMC, 25(05):804–813, 1995.

T. Denœux.
A neural network classifier based on Dempster-Shafer theory.
IEEE transactions on SMC A, 30(2):131–150, 2000.

T. Denœux.
Analysis of evidence-theoretic decision rules for pattern classification.
Pattern Recognition, 30(7):1095–1107, 1997.

C. Lian, S. Ruan and T. Denœux.
An evidential classifier based on feature selection and two-step classification strategy.
Pattern Recognition, 48:2318–2327, 2015.

References on classification II

cf. https://www.hds.utc.fr/~tdenoeux

C. Lian, S. Ruan and T. Denœux.
Dissimilarity metric learning in the belief function framework.
IEEE Transactions on Fuzzy Systems (to appear), 2016.

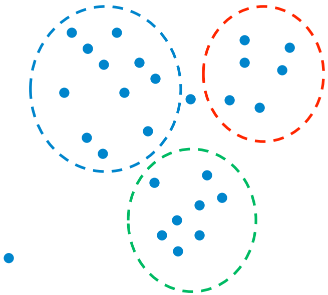
Outline

1. Dempster-Shafer theory
   - Mass, belief and plausibility functions
   - Dempster's rule
   - Decision analysis

2. Evidential classification
   - Evidential K-NN rule
   - Evidential neural network classifier
   - Decision analysis

3. **Application to clustering**
   - credal partition
   - Evidential c-means
   - EVCLUS
   - EK-NNclus
   - Handling a large number of clusters

Clustering



- n objects described by
  - Attribute vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ (attribute data) or
  - Dissimilarities (proximity data).
- Goal: find a meaningful structure in the data set, usually a partition into c crisp or fuzzy subsets.
- Belief functions may allow us to express richer information about the data structure.

Outline

Clustering concepts

Hard and fuzzy clustering

- Hard clustering: each object belongs to one and only one group. Group membership is expressed by binary variables u_{ik} such that $u_{ik} = 1$ if object i belongs to group k and $u_{ik} = 0$ otherwise
- Fuzzy clustering: each object has a degree of membership $u_{ik} \in [0, 1]$ to each group, with $\sum_{k=1}^{c} u_{ik} = 1$
- Fuzzy clustering with noise cluster: each object has a degree of membership $u_{ik} \in [0, 1]$ to each group and a degree of membership $u_{i*} \in [0, 1]$ to a noise cluster, with $\sum_{k=1}^{c} u_{ik} + u_{i*} = 1$

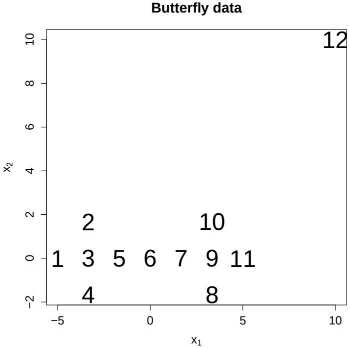
Clustering concepts

Possibilistic, rough, credal clustering

- Possibilistic clustering: the condition $\sum_{k=1}^{c} u_{ik} = 1$ is relaxed. Each number u_{ik} can be interpreted as a degree of possibility that object i belongs to cluster k
- Rough clustering: the membership of object i to cluster k is described by a pair $(\underline{u}_{ik}, \overline{u}_{ik}) \in \{0,1\}^2$, with $\underline{u}_{ik} \leq \overline{u}_{ik}$, indicating its membership to the lower and upper approximations of cluster k
- Evidential clustering: based on Dempster-Shafer (DS) theory (the topic of this talk)

Evidential clustering

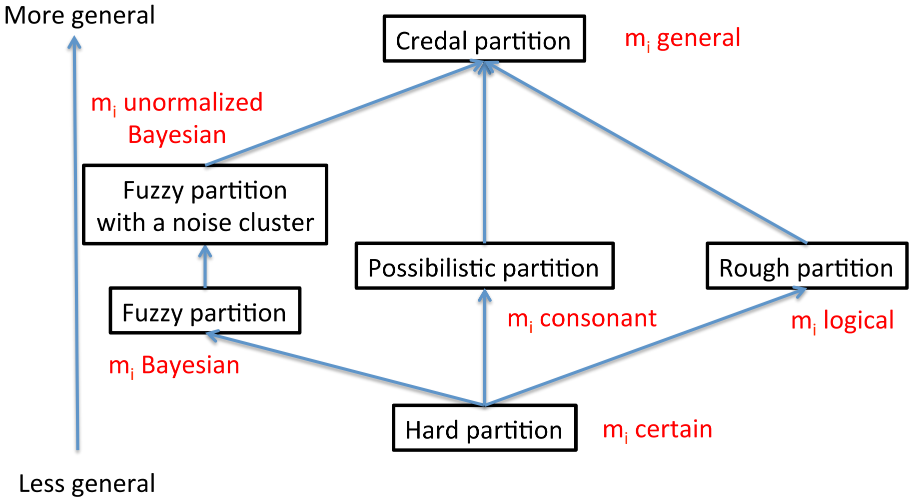
- In evidential clustering, the cluster membership of each object is considered to be uncertain and is described by a (not necessarily normalized) mass function m_i over Ω
- The n-tuple $\mathcal{M} = (m_1, \ldots, m_n)$ is called a credal partition
- Example:



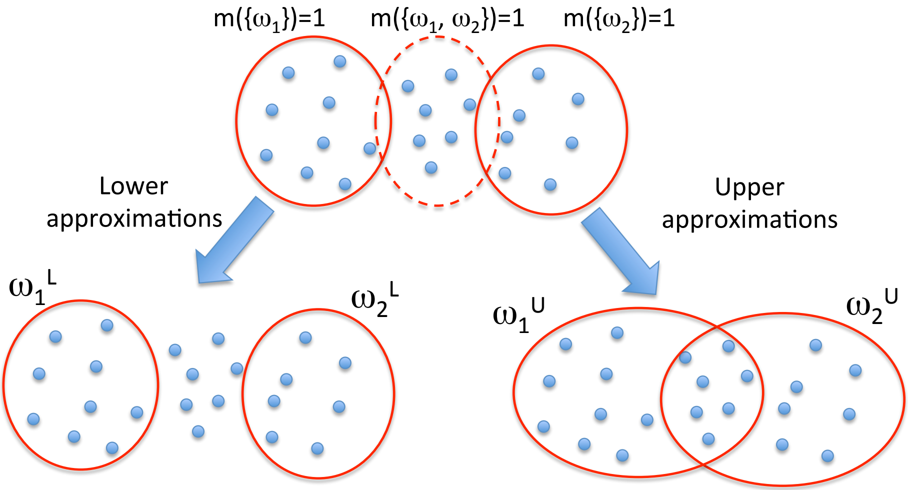
Credal partition

	$\emptyset$	$\{\omega_1\}$	$\{\omega_2\}$	$\{\omega_1, \omega_2\}$
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

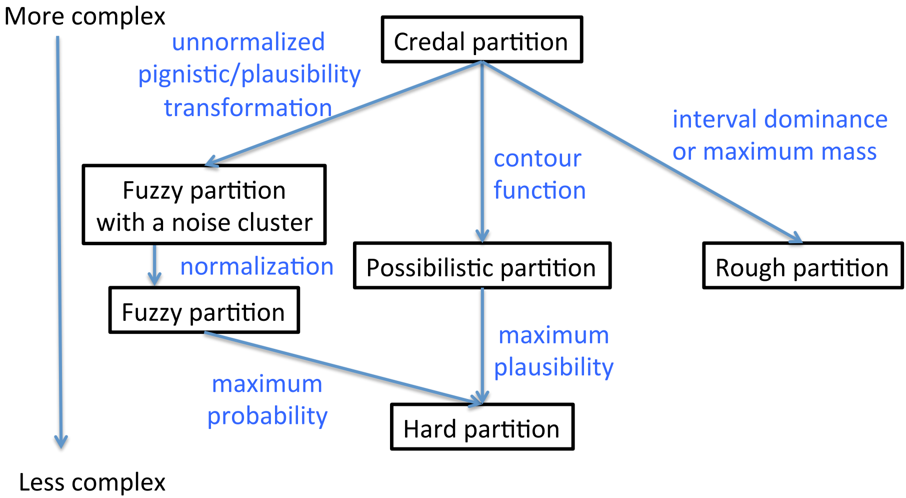
Relationship with other clustering structures



Rough clustering as a special case



Summarization of a credal partition



Algorithms

1. Evidential *c*-means (ECM): (Masson and Denoeux, 2008):
    - Attribute data,
    - HCM, FCM family (alternate optimization of a cost function).
2. EVCLUS (Denoeux and Masson, 2004; Denoeux et al., 2016):
    - Proximity (possibly non metric) data,
    - Multidimensional scaling approach.
3. EK-NNclus (Denoeux et al, 2015)
    - Attribute or proximity data
    - Decision-directed clustering algorithm based on the evidential K-NN classifier

Outline

Principle

- Problem: generate a credal partition $\mathcal{M} = (m_1, \ldots, m_n)$ from attribute data $X = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$, $\boldsymbol{x}_i \in \mathbb{R}^p$.
- Generalization of hard and fuzzy c-means algorithms:
  - Each cluster is represented by a prototype;
  - Cyclic coordinate descent algorithm: optimization of a cost function with respect to the prototypes and to the credal partition.

Fuzzy *c*-means (FCM)

- Minimize

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

with $d_{ik}=||\boldsymbol{x}_i-\boldsymbol{v}_k||$ under the constraints $\sum_k u_{ik}=1, \forall i$.

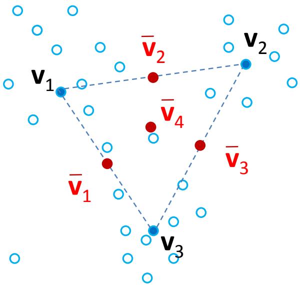
- Alternate optimization algorithm:

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

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

ECM algorithm

Principle



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

ECM algorithm: objective criterion

- Criterion to be minimized:

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

  subject to

$$\sum_{\{j/A_j\subseteq\Omega,A_j\neq\emptyset\}} m_{ij}+m_{i\emptyset}=1,\quad \forall i\in\{1,\ldots,n\},$$

- Parameters:
  - α controls the specificity of mass functions (default: 1)
  - β controls the hardness of the credal partition (default: 2)
  - δ controls the proportion of data considered as outliers
- $J_{\text{ECM}}(M,V)$ can be iteratively minimized with respect to M and V using a cyclic coordinate descent algorithm.

ECM algorithm: update equations

- Optimization of $J_{\text{ECM}}(M, V)$ w.r.t. M for fixed V:

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

for $i = 1, \ldots, n$ and for all j such that $A_j \neq \emptyset$, and

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}, \quad i = 1, \ldots, n$$

- Optimization of $J_{\text{ECM}}(M, V)$ w.r.t. V for fixed M: solving a system of the form

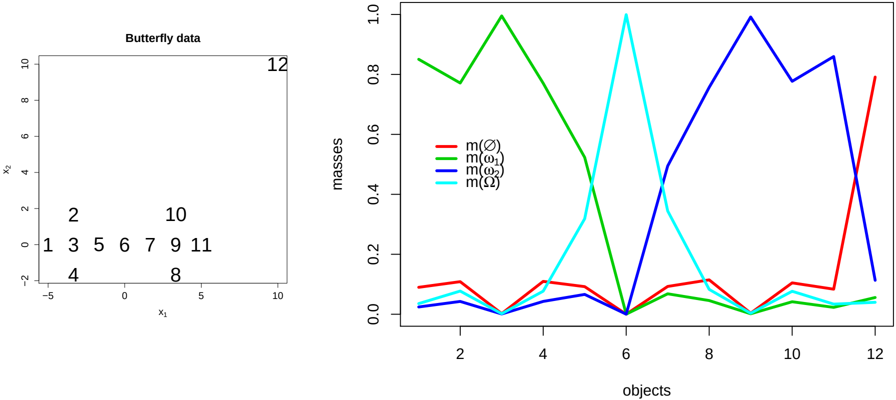
$$HV = B,$$

where B is the matrix of size $c \times p$ and H the matrix of size $c \times c$

Implementation in R

```
library(evclust)
data('butterfly')
n<-nrow(butterfly)

clus<-ecm(butterfly[,1:2],c=2,delta=sqrt(20))
```

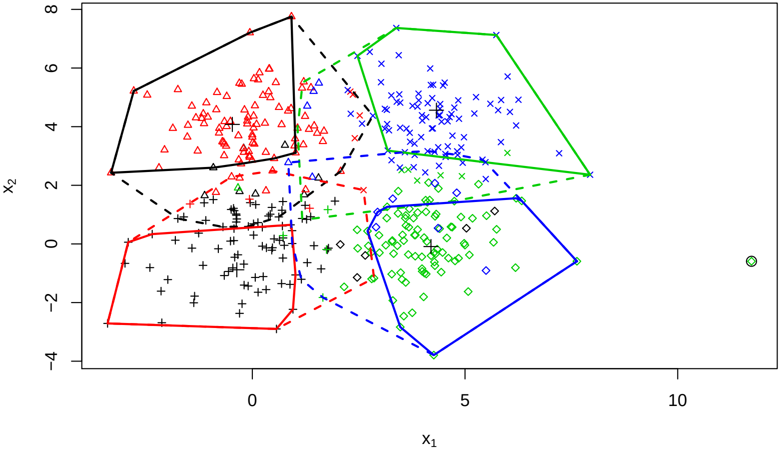
Butterfly dataset



Four-class dataset

```
data("fourclass")
clus<-ecm(fourclass[,1:2],c=4,type='pairs',delta=5)

plot(clus,X=fourclass[,1:2],ytrue=fourclass[,3],Outliers = TRUE,
approx=2)
```

4-class data set



Determining the number of groups

- If a proper number of groups is chosen, the prototypes will cover the clusters and most of the mass will be allocated to singletons of Ω.
- On the contrary, if c is too small or too high, the mass will be distributed to subsets with higher cardinality or to $\emptyset$.
- Nonspecificity of a mass function:

$$N(m) \triangleq \sum_{A \in 2^\Omega \setminus \emptyset} m(A) \log_2 |A| + m(\emptyset) \log_2 |\Omega|$$

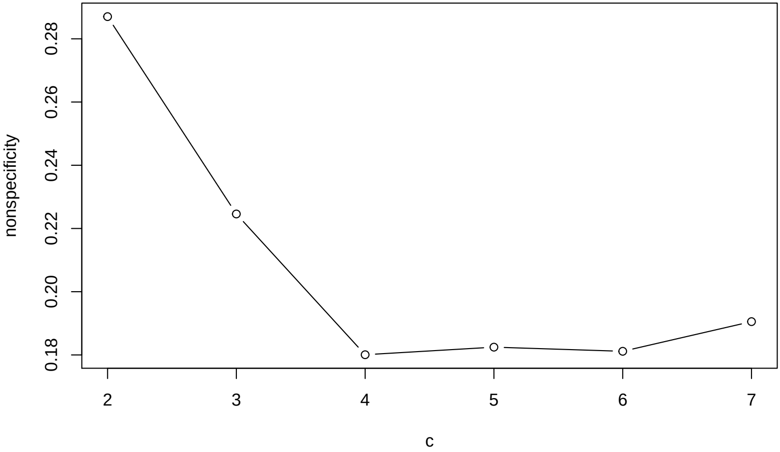
- Proposed validity index of a credal partition:

$$N^*(c) \triangleq \frac{1}{n \log_2(c)} \sum_{i=1}^{n} \left[\sum_{A \in 2^\Omega \setminus \emptyset} m_i(A) \log_2 |A| + m_i(\emptyset) \log_2(c) \right]$$

Example (Four-class dataset)

```
C<-2:7
N<-rep(0,length(C))
for(k in 1:length(C)){

clus<-ecm(fourclass[,1:2],c=C[k],type='pairs',alpha=2,
delta=5,disp=FALSE)

N[k]<-clus$N
}
plot(C,N,type='b',xlab='c',ylab='nonspecificity')
```

Results



Outline

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.
- Basic idea: "The more similar two objects, the more plausible it is that they belong to the same group".
- How to formalize this idea?

Formalization

- Let m_i and m_j be mass functions regarding the group membership of objects o_i and o_j.
- The plausibility of the proposition S_{ij}: "objects o_i and o_j belong to the same group" can be shown to be equal to:

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

  where κ_{ij} = degree of conflict between m_i and m_j.
- Problem: find a credal partition $\mathcal{M} = (m_1, \ldots, m_n)$ such that larger degrees of conflict κ_{ij} correspond to larger dissimilarities d_{ij}.

Cost function

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

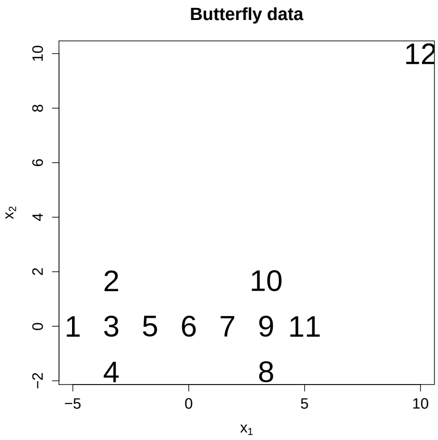
$$J(\mathcal{M}) = \eta \sum_{i<j} (\kappa_{ij} - \varphi(d_{ij}))^2$$

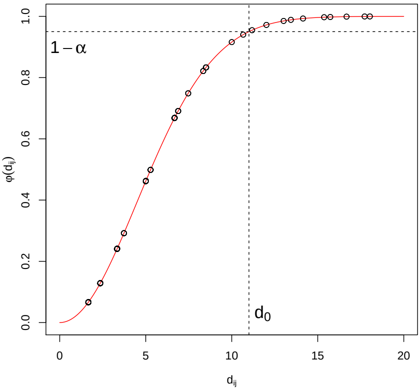
where

- $\eta = \left(\sum_{i<j} \varphi(d_{ij})^2\right)^{-1}$ is a normalizing constant, and
- φ is an increasing function from $[0, +\infty)$ to $[0, 1]$.
- For instance: $\varphi(d) = 1 - \exp(-\gamma d^2)$

Butterfly example

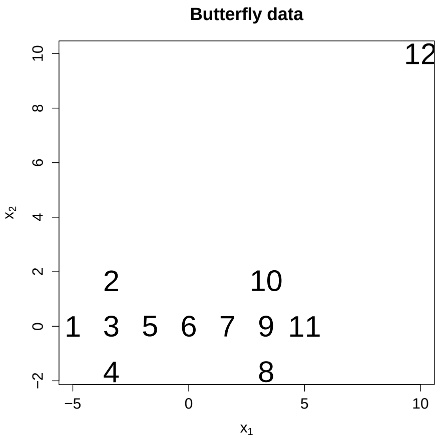
Data and dissimilarities

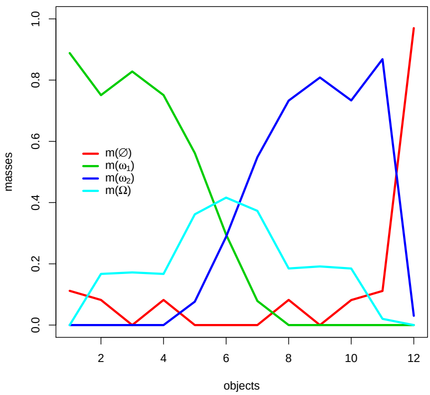




Butterfly example

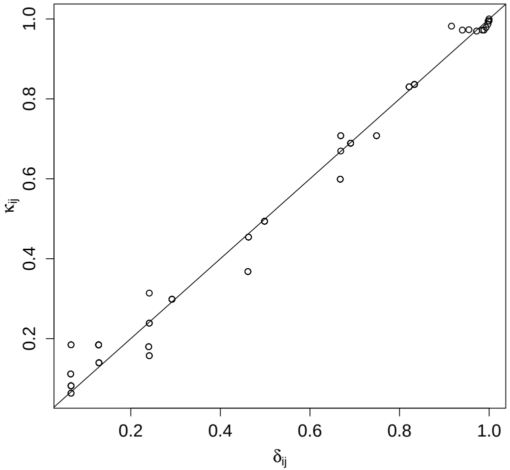
Credal partition





Butterfly example

Shepard diagram



Optimization algorithm

- How to minimize $J(\mathcal{M})$? Two methods:
  1. Using a gradient or quasi-Newton algorithm (slow).
  2. Using a cyclic coordinate descent algorithm minimizing $J(\mathcal{M})$ with respect to each m_i at a time.
- The latter approach exploits the particular approach of the problem (a quadratic programming problem is solved at each step), and it is thus much more efficient.
- This algorithm is called Iterative Row-wise Quadratic Programming (IRQP).

IRQP algorithm

Vector representation of the cost function

- The stress function can be written as

$$J(\mathcal{M}) = \eta \sum_{i<j} (\boldsymbol{m}_i^T \boldsymbol{C} \boldsymbol{m}_j - \delta_{ij})^2.$$

  where
  - $\delta_{ij} = \varphi(d_{ij})$ are the scaled dissimilarities
  - $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ are vectors encoding mass functions m_i and m_j
  - $\boldsymbol{C}$ is a square matrix, with general term $C_{k\ell} = 1$ if $F_k \cap F_\ell = \emptyset$ and $C_{k\ell} = 0$ otherwise.
- Fixing all mass functions except m_i, the stress function becomes quadratic. Minimizing J w.r.t. $\boldsymbol{m}_i$ is a linearly constrained positive least-squares problem, which can be solved using efficient algorithms.
- By iteratively updating each m_i, the algorithm converges to a local minimum of the cost function.

Reducing the number of parameters

- If the mass functions have a general form, the number of parameters to estimate of $n(2^c - 1)$. It grows exponentially with c.
- To reduce the complexity, focal sets can be reduced to $\{\omega_k\}_{k=1}^c$, $\emptyset$, and Ω.
- A more sophisticated strategy will be described later.

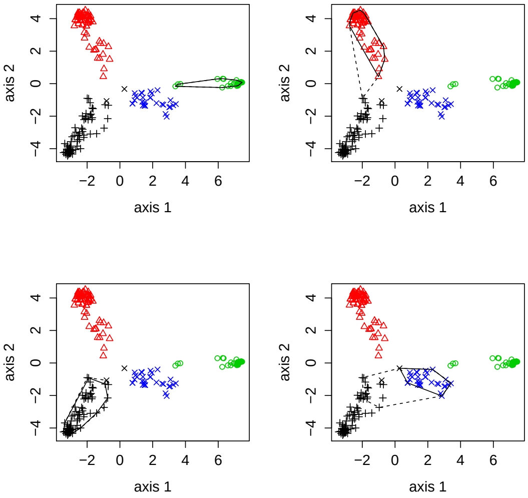
Proteins example

- Dissimilarity matrix derived from the structural comparison of 213 protein sequences.
- Each of these proteins is known to belong to one of four classes of globins: hemoglobin-α (HA), hemoglobin-β (HB), myoglobin (M) and heterogeneous globins (G).
- The next figure displays a two-dimensional MDS configuration of the data with the true partition, as well as the clustering result obtained by EVCLUS, with $c = 4$ and $d_0 = \max_{i,j} d_{ij}$.

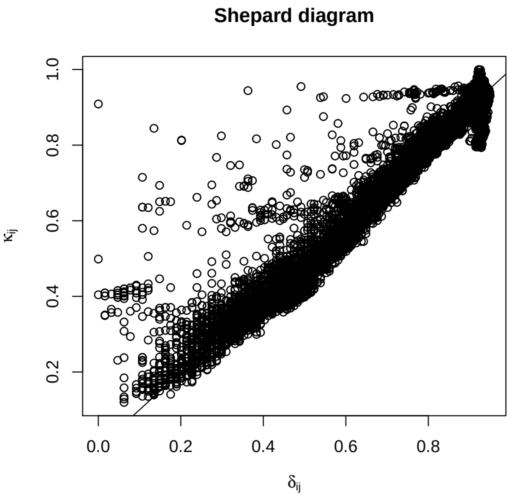
Implementation in R

```
library(evclust)
data(protein)

clus <- kevclus(D=protein$D,c=4,type='simple',d0=max(protein$D))

z<- cmdscale(protein$D,k=2)

plot(clus,X=z,mfrow=c(2,2),ytrue=protein$y,
Outliers=FALSE,approx=1)
```

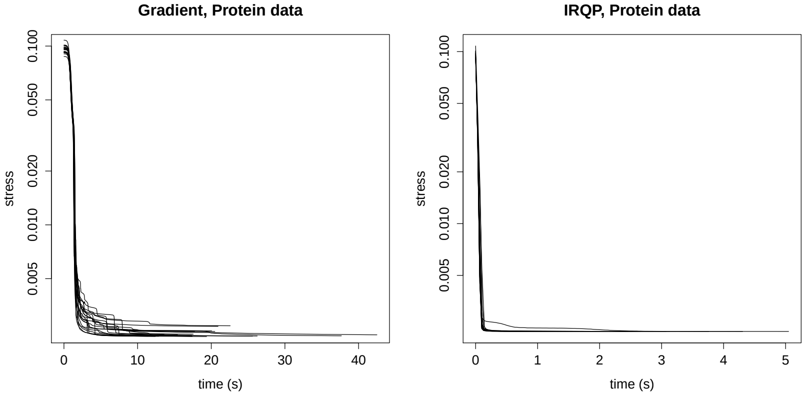
Proteins example: partition



Proteins example: Shepard diagram

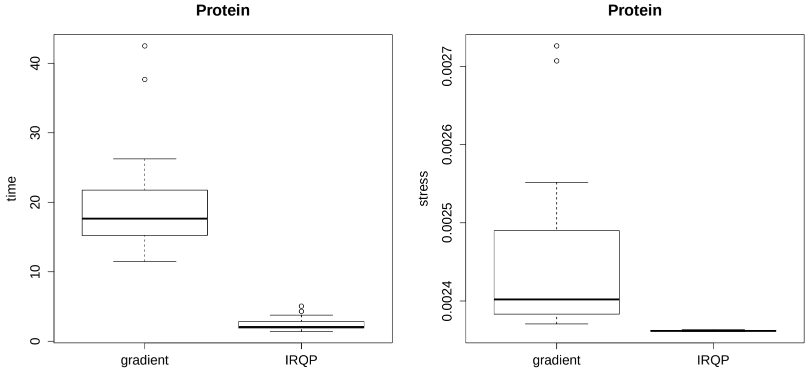


Proteins example: learning curves



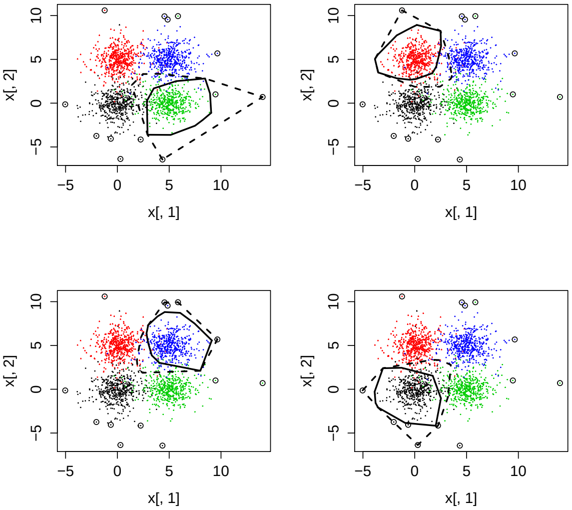
Stress vs. time (in seconds) for 20 runs of the Gradient (a) and IRQP (b) algorithms on the Protein data. Note the different scales on the *x*-axes.

Proteins example: learning curves



Boxplots of computing time (a) and stress value at convergence (b) for 20 runs of the Gradient and IRQP algorithms on the Protein data.

Example with a four-class dataset (2000 objects)



Handling large datasets

- EVCLUS requires to store the whole dissimilarity matrix: it inapplicable to large dissimilarity data.
- Idea: compute the differences between degrees of conflict and dissimilarities, for only a subset of randomly sampled dissimilarities.
- Let $j_1(i), \ldots, j_k(i)$ be k integers sampled at random from the set $\{1, \ldots, i-1, i+1, \ldots, n\}$, for $i = 1, \ldots, n$. Let J_k the following stress criterion,

$$J_k(\mathcal{M}) = \eta \sum_{i=1}^{n} \sum_{r=1}^{k} (\kappa_{i,j_r(i)} - \delta_{i,j_r(i)})^2,$$

- The calculation of $J_k(\mathcal{M})$ requires only $O(nk)$ operations.
- If k can be kept constant as n increases, or, at least, if k increases slower than linearly with n, then significant gains in computing time and storage requirement could be achieved.

Zongker Digit dissimilarity data

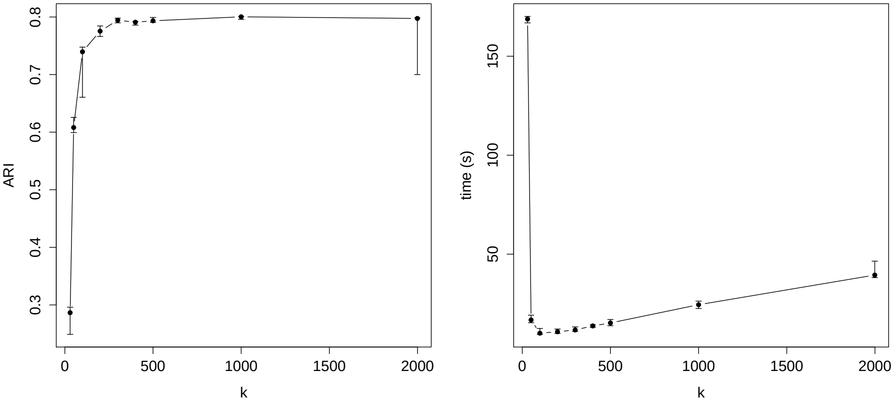
- Similarities between 2000 handwritten digits in 10 classes, based on deformable template matching.
- As the dissimilarity matrix was initially non symmetric, we symmetrized it by the transformation $d_{ij} \leftarrow (d_{ij} + d_{ji})/2$.
- The k-EVCLUS algorithm was run with $c = 10$ and the following values of k: 30, 50,100, 200, 300, 400, 500, 1000 and 1999. Parameter d_0 was fixed to the 0.3-quantile of the dissimilarities. For each value of k, k-EVCLUS was run 10 times with random initializations.

Implementation in R

```
load('zongker.RData')

n<-nrow(zongker$D)
k=200
D<-matrix(0,n,k)
J<-matrix(0,n,k)
for(i in 1:n){
ii<-sample((1:n)[-i],k)
J[i,]<-ii
D[i,]<-zongker$D[i,ii]
}

clus<-kevclus(D=D,J=J,c=10,type='simple',d0=quantile(D,0.3))

library(mclust)
adjustedRandIndex(zongker$y,clus$y.pl)
```

Zongker Digit dissimilarity data

Results



Outline

Decision-directed clustering

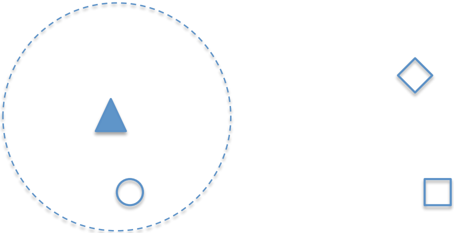
- Decision-directed approach to clustering:
  - Prior knowledge is used to design a classifier, which is used to label the samples
  - The classifier is then updated, and the process is repeated until no changes occur in the labels
- The c-means algorithm is based on this principle: here, the nearest-prototype 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 as the base classifier

Example

Toy dataset

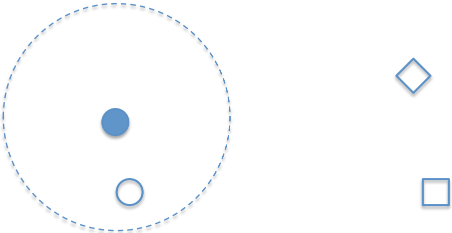
Example

Iteration 1



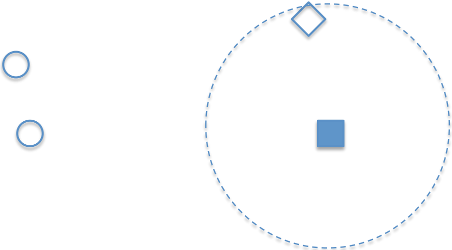
Example

Iteration 1 (continued)



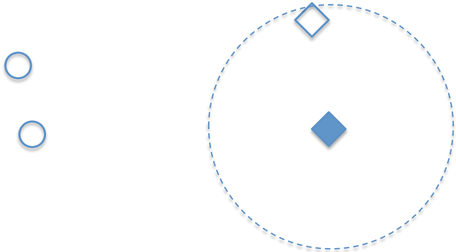
Example

Iteration 2



Example

Iteration 2 (continued)



Example

Result

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, compute the set $N_K(i)$ of indices of the K nearest neighbors of each object i.
- 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 binary variables u_{ik} as $u_{ik} = 1$ is object o_i belongs to cluster k, and $u_{ik} = 0$ 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
- We classify each object o_i it using the EK-NN rule. The plausibility that object o_i belongs to class k is

$$pl_{ik} \propto \prod_{j \in N_K(i)} \left(1 - \varphi(d_{ij})\right)^{1-u_{jk}}$$

  with $\varphi(d_{ij}) = \exp(-\gamma d_{ij}^p)$, $p = 1$ or $p = 2$.
- Its logarithm is (up to an additive constant)

$$\begin{aligned} s_{ik} &= -\sum_{j \in N_K(i)} \ln(1 - \varphi(d_{ij}))u_{jk} \\ &= \sum_{j \in N_K(i)} w_{ij} u_{jk} \end{aligned}$$

  with $w_{ij} = -\ln(1 - \varphi(d_{ij}))$.

EK-NNclus algorithm

Step 3: iteration (continued)

- We then assign object o_i to the cluster with the highest plausibility, i.e., we update the variables u_{ik} as

$$u_{ik} = \begin{cases} 1 & \text{if } s_{ik} = \max_{k'} s_{ik'} \\ 0 & \text{otherwise} \end{cases}$$

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

EK-NNclus algorithm

Step 4: Computation of the credal partition

After the algorithm has converged, we can compute the final mass functions

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

for $i = 1, \ldots, n$, where each m_{ij} is the following mass function,

$$\begin{aligned} m_{ij}(\{\omega_k\}) &= u_{jk}\varphi(d_{ij}), \quad k = 1, \ldots, c \\ m_{ij}(\Omega) &= 1 - \varphi(d_{ij}) \end{aligned}$$

EK-NNclus algorithm

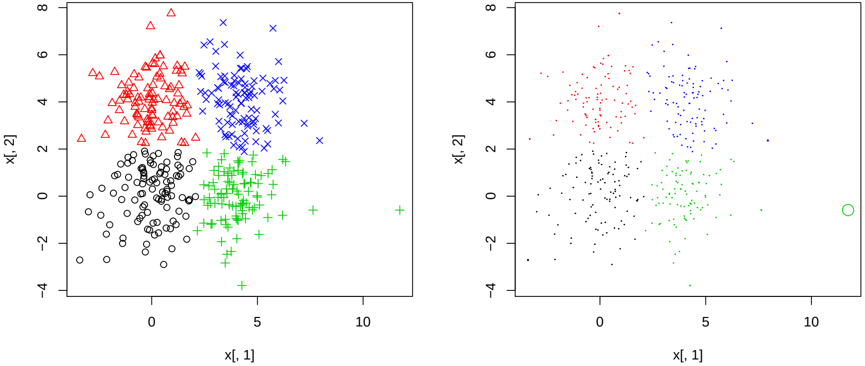
Parameter tuning

- Number K of neighbors: two to three times $\sqrt{n}$
- γ: fixed to the inverse of the q-quantile of the distances d_{ij}^p between an object and its K NN
- Typically, with $q \geq 0.5$

Ek-NNclus in R

```
data(fourclass)
x<-fourclass[,1:2]
n<-nrow(x)
y0<-1:n
clus<-EkNNclus(x, D, K=50, y0, ntrials = 1, q = 0.5, p = 1)

plot(x[,1],x[,2],pch=clus$y.pl,col=clus$y.pl)

c<-ncol(clus$mass)-1
plot(x[,1],x[,2],pch=clus$y,col=clus$y.pl,
cex=0.1+2*clus$mass[,c+1])
```

Example



Properties

- The EK-NNclus algorithm can be implemented exactly in a competitive Hopfield neural network model
- The neural network converges a stable state corresponding to a local minimum of the following energy function

$$E(U) = -\frac{1}{2}\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j\neq i} w_{ij} u_{ik} u_{jk}$$

  where $U = (u_{ik})$ denotes the $n \times c$ matrix of 0s and 1s encoding the neuron states
- The following relation holds

$$pl(R) = -E(U) + C$$

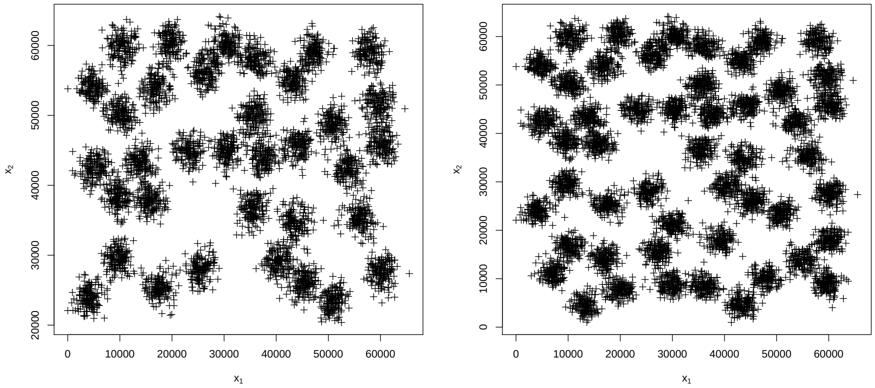
  where $pl(R)$ is the plausibility of the partition encoded by U
- The EK-NNclus algorithm thus searches for the most plausible partition, in the (huge) space of all partitions of the dataset!

Experiments

- Settings:
  - $\varphi(d_{ij}) = \exp(-\gamma d_{ij}^2)$, where d_{ij} is the Euclidean distance between objects i and j
  - $q = 0.9$
  - Number K of neighbors: two to three times $\sqrt{n}$
  - Initialization methods: $c_0 = n$ initial clusters, or $c_0 = 1000$ random initial clusters
- Datasets[1]
  1. A-sets: Two-dimensional datasets with $n \in \{3000, 5250, 7000\}$ objects and $c \in \{20, 35, 50\}$ clusters
  2. DIM-sets: $n = 1024$ objects and 16 Gaussian clusters in 256, 512 and 1024 dimensions

[1] From `http://cs.joensuu.fi/sipu/datasets`

A-sets



Results with the A-sets

- Number of neighbors: $K = 150$ for dataset A1, and $K = 200$ for datasets A2 and A3
- The EK-NNclus algorithm was run 10 times

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

Results with the DIM-sets

- Number of neighbors: $K = 50$
- The EK-NNclus algorithm was run 10 times with $c_0 = n$

Dataset	Result	EK-NNclus	c-means	`pdfCluster`	model-based (constrained)
dim256	c	16 (0)	16 (fixed)	5	16
	ARI	1.0 (0)	0.94	0.23	1
	time	1.4 (0.058)	2.76	11.30	116
dim512	c	16 (0)	16(fixed)	9	16
	ARI	1 (0)	0.94	0.5	1
	time	1.4 (0.11)	13.27	10.9	467
dim1024	c	16 (0)	16 (fixed)	8	18
	ARI	1 (0)	0.94	0.28	0.998
	time	1.4 (0.14)	36.38	11.13	23

Outline

1. Dempster-Shafer theory
   - Mass, belief and plausibility functions
   - Dempster's rule
   - Decision analysis
2. Evidential classification
   - Evidential K-NN rule
   - Evidential neural network classifier
   - Decision analysis
3. Application to clustering
   - credal partition
   - Evidential c-means
   - EVCLUS
   - EK-NNclus
   - Handling a large number of clusters

Need to limit the number of 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.

Method

1. In the first step, a clustering algorithm (ECM, EVCLUS, EK-NNclus) is run in the basic configuration, with focal sets of cardinalities 0, 1 and (optionally) c. A credal partition $\mathcal{M}_0$ is obtained.
2. The similarity between each pair of clusters (ω_j, ω_ℓ) is computed as

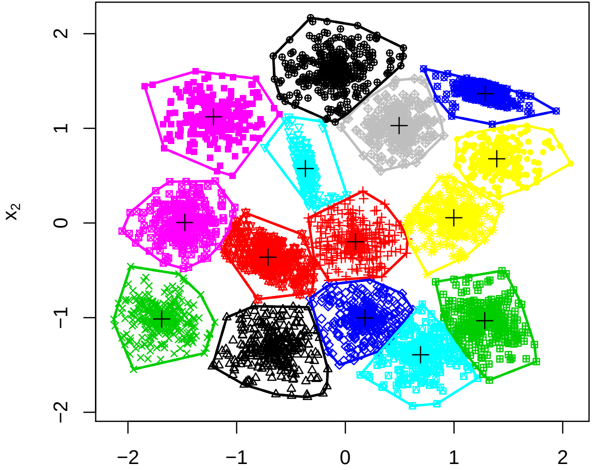
$$S(j,\ell) = \sum_{i=1}^{n} pl_{ij} pl_{i\ell},$$

   where pl_{ij} and $pl_{i\ell}$ are the normalized plausibilities that object i belongs, respectively, to clusters j and ℓ. We then determine the set P_K of pairs $\{\omega_j, \omega_\ell\}$ that are mutual K nearest neighbors.
3. The clustering algorithm is run again, starting from the previous credal partition $\mathcal{M}_0$, and adding as focal sets the pairs in P_K.

Example in R: step 1

```
data(s2)
clus<-ecm(x=s2,c=15,type='simple',Omega=FALSE,delta=1,disp=FALSE)
plot(x=clus,X=s2,Outliers = TRUE)
```

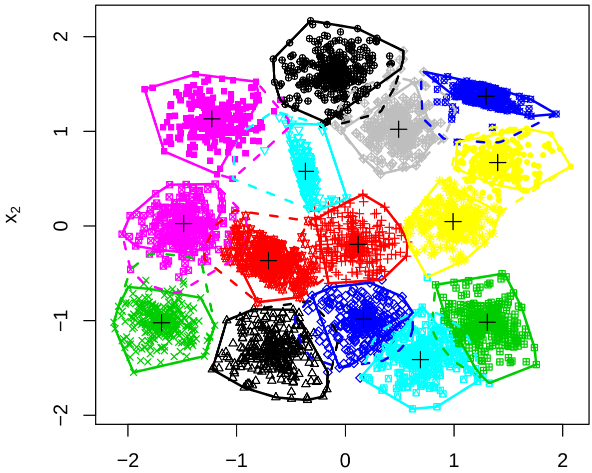
Result after Step 1



Example in R: steps 2 and 3

```
P<-createPairs(clus,k=2)

clus1<-ecm(x=s2,c=15,type='pairs',Omega=FALSE,pairs=P$pairs,
g0=clus$g,delta=1,disp=FALSE)

plot(x=clus1,X=s2,Outliers = TRUE,approx=2)
```

Final result



Summary

- The theory of belief function has great potential in data analysis and challenging machine learning:
  - Classification (supervised learning)
  - Clustering (unsupervised learning) problems
- Belief functions allow us to:
  - Learn from weak information (partially supervised learning, imprecise and uncertain data)
  - Express uncertainty on the outputs of a learning system (e.g., credal partition)
  - Combine the outputs from several learning systems (ensemble classification and clustering), or combine data with expert knowledge (constrained clustering)
- R packages `evclass` and `evclust` available from CRAN at `https://cran.r-project.org/web/packages`

References on clustering I

cf. https://www.hds.utc.fr/~tdenoeux

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

- M.-H. Masson and T. Denœux.
  RECM: Relational Evidential c-means algorithm.
  Pattern Recognition Letters, 30:1015-1026, 2009.

- B. Lelandais, S. Ruan, T. Denoeux, P. Vera, I. Gardin.
  Fusion of multi-tracer PET images for Dose Painting.
  Medical Image Analysis, 18(7):1247-1259, 2014.

- T. Denœux and M.-H. Masson.
  EVCLUS: Evidential Clustering of Proximity Data.
  IEEE Transactions on SMC B, 34(1):95-109, 2004.

References on clustering II

cf. https://www.hds.utc.fr/~tdenoeux

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

T. Denoeux, 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.