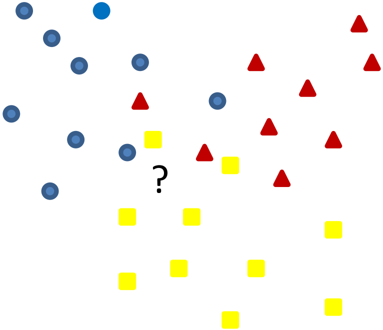
Classification and clustering

Thierry Denœux

Université de Technologie de Compiègne
HEUDIASYC (UMR CNRS 7253)
https://www.hds.utc.fr/~tdenoeux

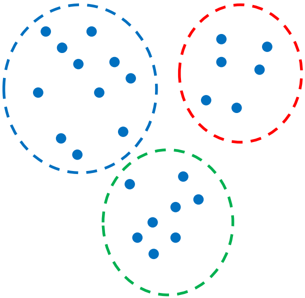
Third School on belief functions and their applications,
Stella Plage, France, October 1st, 2015

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 $\mathbf{x} \in \mathbb{R}^p$;
  - A class label $y \in \Omega$.
- Problem: given a learning set $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, predict the class label of a new instance described by $\mathbf{x}$.

Clustering problem



- n objects described by
  - Attribute vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (attribute data) or
  - Dissimilarities (proximity data)
- Goal: find a meaningful structure in the data set, usually a partition into c subsets, or a more complex mathematical representation (fuzzy partition, etc.)

Why can belief functions be useful?

1. Exploit the high expressiveness of belief functions to
   1. Represent more faithfully the uncertainty of the predictions made by a classifier (for, e.g., combining several classifiers, or providing the user with richer information about the uncertainty of the classification)
   2. Reveal richer information about the data (clustering problems)
2. Represent uncertainty about the data themselves:
   1. Uncertain class labels (partially supervised learning)
   2. Clustering of imprecise/uncertain data

Overview of the main approaches

Classification

1. Classifier fusion: convert the outputs from standard classifiers into belief functions and combine them using, e.g., Dempster's rule (e.g., Quost al., 2011)
2. Develop evidence-theoretic classifiers directly providing belief functions as outputs:
   - Generalized Bayes theorem, extends the Bayesian classifier when class densities and priors are ill-known (Appriou, 1991; Denœux and Smets, 2008)
   - Distance-based classifiers: evidential K-NN rule (Denœux, 1995), evidential neural network classifier (Denœux, 2000)
   - Predictive evidential classifiers (e.g., logistic regression, Xu et al., 2015)

Overview of the main approaches

Clustering

Express uncertainty about the membership of objects to clusters using the notion of credal partition

1. Match degrees of conflict with inter-point distances: EVCLUS algorithm (Denoeux and Masson, 2004)
2. Extend prototype-based clustering methods such as the hard or fuzzy c-means: Evidential c-means (Masson and Denoeux, 2008)
3. Decision-directed clustering using the evidential K-NN classifier: EK-NNclus algorithm (Denoeux et al, 2015)

Overview of the main approaches

Uncertain data

1. In classification, partially supervised data (with uncertain class labels) can be handled using the evidential K-NN classifier (Denoeux, 1995; Denoeux and Zouhal, 2001)
2. More general approach: extend maximum likelihood estimation to uncertain data (e.g., with uncertain class labels and/or attributes) using the Evidential Expectation-Maximization (E^2M) algorithm (Denoeux, 2011; Denoeux, 2012)

Outline

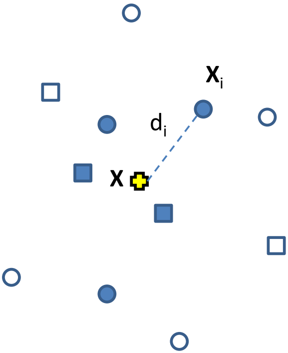
1. Evidential distance-based classifiers
   - Evidential K-NN rule
   - Evidential neural network classifier
2. Learning from uncertain data
   - Motivation
   - Evidential EM algorithm
   - Partially supervised LDA
3. Clustering
   - Credal partition
   - Evidential c-means
   - EK-NNclus

Outline

Outline

1. Evidential distance-based classifiers
   - Evidential K-NN rule
   - Evidential neural network classifier
2. Learning from uncertain data
   - Motivation
   - Evidential EM algorithm
   - Partially supervised LDA
3. Clustering
   - Credal partition
   - Evidential c-means
   - EK-NNclus

Evidential K-NN rule (1/3)



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

Evidential K-NN rule (2/3)

- The evidence of $(\mathbf{x}_i, y_i)$ can be represented by

$$m_i(\{y_i\}) = \varphi(d_i)$$

$$m_i(\Omega) = 1 - \varphi(d_i)$$

where φ is a decreasing function from $[0, +\infty)$ to $[0, 1]$ such that $\lim_{d \to +\infty} \varphi(d) = 0$

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

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

- Function φ can be fixed heuristically or selected among a family $\{\varphi_\theta | \theta \in \Theta\}$ using, e.g., cross-validation

Evidential K-NN rule (3/3)

Decision

- The contour function of the combined mass function m can be written as

$$pl(\omega_k) \propto \prod_{\mathbf{x}_i \in \mathcal{N}_K(\mathbf{x})} (1 - \varphi(d_i))^{1 - s_{ik}}$$

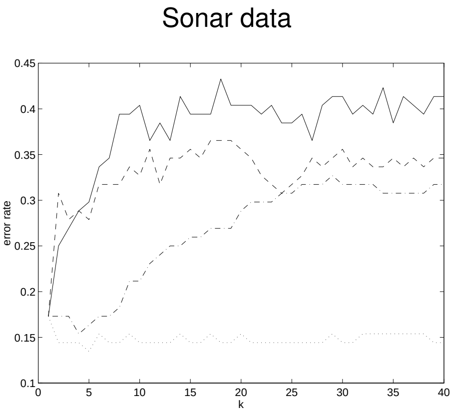
with$s_{ik} = 1$ if $y_i = \omega_k$ and $s_{ik} = 0$ otherwise

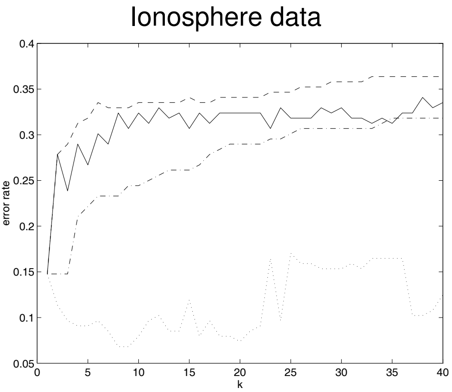
- Its logarithm is

$$\ln pl(\omega_k) = - \sum_{\mathbf{x}_i \in \mathcal{N}_K(\mathbf{x})} s_{ik} \ln (1 - \varphi(d_i)) + C$$

- It can be computed in time proportional to $K|\Omega|$

Performance comparison (UCI database)





Test error rates as a function of *k* for the voting (-), evidential (:), fuzzy (–) and distance-weighted (-.) *k*-NN rules.

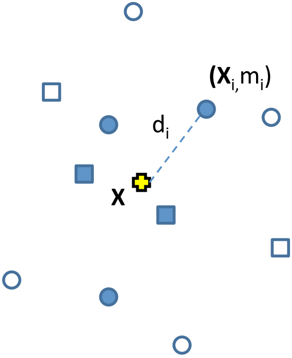
Partially supervised data

- We now consider a learning set of the form

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

  where
  - $\mathbf{x}_i$ 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:

$$m_i'(A) = \varphi(d_i)\, m_i(A), \quad \forall A \subset \Omega.$$

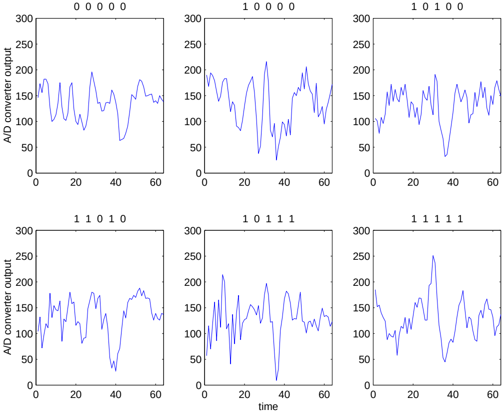
$$m_i'(\Omega) = 1 - \sum_{A \subset \Omega} m_i'(A).$$

- The k mass functions m_i' are combined using Dempster's rule:

$$m = \bigoplus_{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} m_i'.$$

Example: EEG data

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



Results on EEG data

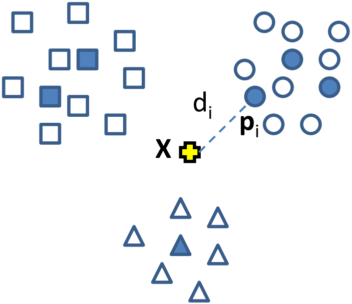
(Denoeux and Zouhal, 2001)

- $c = 2$ classes, $p = 64$
- For each learning instance $\mathbf{x}_i$, 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

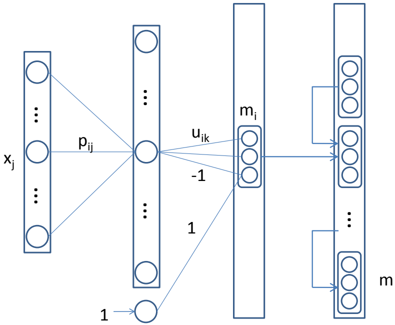
Outline

Evidential neural network classifier



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

Neural network architecture



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

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

- Combination:

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

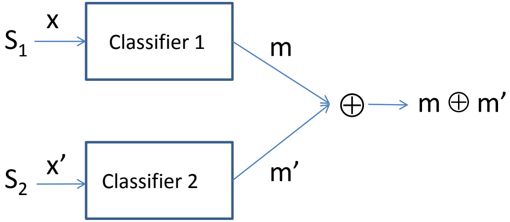
- All parameters are learnt from data by minimizing an error function.

Results on classical data

Vowel data
$c = 11$,
$p = 10$
$n = 568$
test : 462 ex.
(different speakers)

Classifier	test error rate
Multi-layer perceptron (88 units)	0.49
Radial Basis Function (528 units)	0.47
Gaussian node network (528 units)	0.45
Nearest neighbor	0.44
Linear Discriminant Analysis	0.56
Quadratic Discriminant Analysis	0.53
CART	0.56
BRUTO	0.44
MARS (degree=2)	0.42
Evidential NN (33 prototypes)	0.38
Evidential NN (44 prototypes)	0.37
Evidential NN (55 prototypes)	0.37

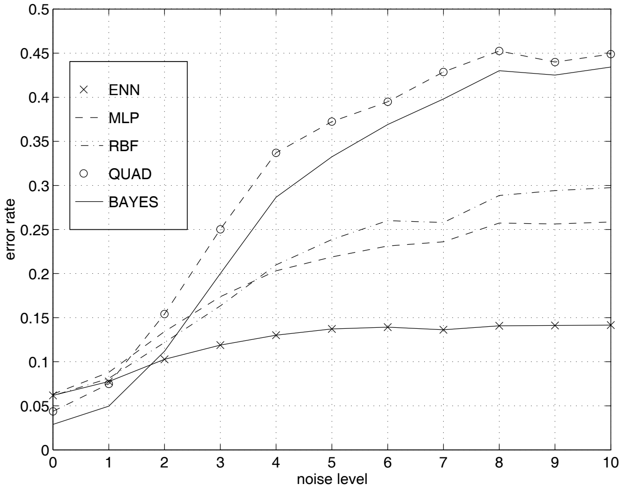
Data fusion example



- $c = 2$ classes
- Learning set ($n = 60$): $\mathbf{x} \in \mathbb{R}^5, \mathbf{x}' \in \mathbb{R}^3$, Gaussian distributions, conditionally independent
- Test set (real operating conditions): $\mathbf{x} \leftarrow \mathbf{x} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.

Results

Test error rates: $\mathbf{x} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$



Outline

Outline

Introductory example

- Let us consider a population in which some disease is present in proportion θ.
- n patients have been selected at random from that population. Let $x_i = 1$ if patient i has the disease, $x_i = 0$ otherwise. Each x_i is a realization of $X_i \sim \mathcal{B}(\theta)$.
- We assume that the x_i's are not observed directly. For each patient i, a physician gives a degree of plausibility $pl_i(1)$ that patient i has the disease and a degree of plausibility $pl_i(0)$ that patient i does not have the disease.
- The observations are uncertain data of the form $pl_1, \ldots, pl_n$.
- How to estimate θ?

Aleatory vs. epistemic uncertainty

In the previous example, uncertainty has two distinct origins:

1. Before a patient has been drawn at random from the population, the uncertainty is due to the variability of the variable of interest in the population. This aleatory uncertainty cannot be reduced.
2. After the random experiment has been performed, the uncertainty is due to lack of knowledge of the state of each particular patient. This epistemic uncertainty could be reduced by carrying out further investigations.

Approach

- Here, we consider statistical estimation problems in which both kinds of uncertainty are present: it will be assumed that each data item x
  - has been generated at random from a population (aleatory uncertainty), but
  - it is ill-known because of imperfect measurement or perception (epistemic uncertainty).
- The proposed model treats these two kinds of uncertainty separately:
  - Aleatory uncertainty will be represented by a parametric statistical model;
  - Epistemic uncertainty will be represented using belief functions.

Real world applications

Uncertain data arise in many applications (but epistemic uncertainty is usually neglected). It may be due to:

- Limitations of the underlying measuring equipment (unreliable sensors, indirect measurements), e.g.: biological sensor for toxicity measurement in water.
- Use of imputation, interpolation or extrapolation techniques, e.g.: clustering of moving objects whose position is measured asynchronously by a sensor network,
- Partial or uncertain responses in surveys or subjective data annotation, e.g.: sensory analysis experiments, data labeling by experts, etc.

Data labeling example

Recognition of facial expressions

joy

surprise

sadness

disgust

anger

fear

Recognition of facial expressions

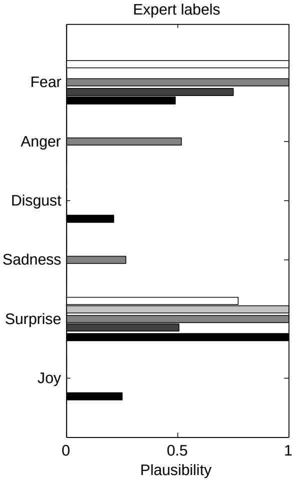
Experiment

- In this kind of problem (object classification in images or videos) the ground truth is usually unknown or difficult to determine with high precision and reliability.
- It is then necessary to have the images subjectively annotated (labeled) by humans.
- How to account for uncertainty in such subjective annotations?
- Experiment:
  - 108 images were labeled by 5 subjects;
  - For each image, subjects were asked to give a degree of plausibility for each of the 6 basic expressions.

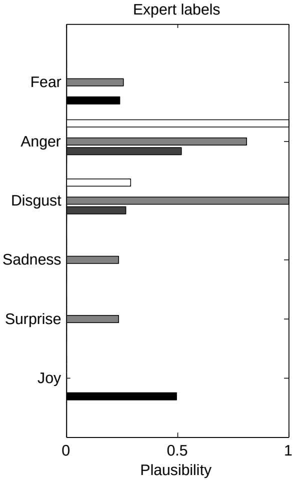
Example 1


Expert labels
Fear
Anger
Disgust
Sadness
Surprise
Joy
0
0.5
1
Plausibility

Example 2


Expert labels
Fear
Anger
Disgust
Sadness
Surprise
Joy
0
0.5
1
Plausibility

Example 3


Expert labels
Fear
Anger
Disgust
Sadness
Surprise
Joy
0
0.5
1
Plausibility

Model

- Complete data: $\mathbf{x} = \{(\mathbf{w}_i, z_i)\}_{i=1}^n$ with
  - $\mathbf{w}_i$: feature vector for image i (pixel gray levels)
  - z_i: class of image i (one the six expressions).
- The feature vectors $\mathbf{w}_i$ are perfectly observed but class labels are only partially known through subjective evaluations.
- How to learn a decision rule from such data?

General approach

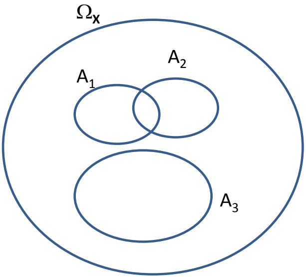
1. Postulate a parametric statistical model $p_{\mathbf{x}}(\mathbf{x};\theta)$ for the complete data;
2. Represent uncertain observed data using belief functions;
3. Estimate θ by minimizing the conflict between the model and the observed data using an extension of the EM algorithm: the evidential EM (E^{2}M) algorithm.

Outline

Model

- Let $\mathbf{X}$ be a (discrete) random vector taking values in $\Omega_{\mathbf{X}}$, with probability mass function $p_{\mathbf{X}}(\cdot; \boldsymbol{\theta})$ depending on an unknown parameter $\boldsymbol{\theta} \in \Theta$.
- Let $\mathbf{x}$ be a realization of $\mathbf{X}$ (complete data).
- We assume that $\mathbf{x}$ is only partially observed, and partial knowledge of $\mathbf{x}$ is described by a mass function m on $\Omega_{\mathbf{X}}$ ("observed" data).
- Problem: estimate $\boldsymbol{\theta}$.

Generalized Likelihood function
Definition



- Assume that m has focal sets $A_1, \ldots, A_r$.
- If we new that $\mathbf{x} \in A_i$, the likelihood would be

$$L(\boldsymbol{\theta}; A_i) = \mathbb{P}_{\mathbf{X}}(A_i; \boldsymbol{\theta}) = \sum_{\mathbf{x} \in A_i} p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\theta}).$$

- General case:

$$L(\boldsymbol{\theta}; m) = \sum_{i=1}^{r} m(A_i) L(\boldsymbol{\theta}; A_i)$$

Generalized Likelihood function

Interpretation

- It can be checked that $L(\theta; m)$ can be written as:

$$L(\theta; m) = \sum_{\mathbf{x} \in \Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}; \theta) pl(\mathbf{x})$$

- $L(\theta; m)$ is equal to one minus the degree of conflict between $p_{\mathbf{X}}(\cdot; \theta)$ and m.
- Consequently, maximizing $L(\theta; m)$ with respect to θ amounts to minimizing the conflict between the parametric model and the uncertain observations.

Generalized Likelihood function

Case of fuzzy data

- Other interpretation of $L(\theta; m)$:

$$L(\theta; m) = \sum_{\mathbf{x} \in \Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}; \theta) pl(\mathbf{x}) = \mathbb{E}_{\theta}\left[pl(\mathbf{X})\right]$$

- If m is consonant, pl may be interpreted as the membership function of a fuzzy subset of Ω_X: we have fuzzy data.
- $L(\theta; m)$ is then the probability of the fuzzy data, in the sense of Zadeh (1968).

Independence assumptions

- Let us assume that $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{np}$, where each $\mathbf{x}_i$ is a realization from a p-dimensional random vector $\mathbf{X}_i$.
- Independence assumptions:
  1. Stochastic independence of $\mathbf{X}_1, \ldots, \mathbf{X}_n$:
  $$p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\theta}) = \prod_{i=1}^{n} p_{\mathbf{X}_i}(\mathbf{x}_i; \boldsymbol{\theta}), \quad \forall \mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \Omega_{\mathbf{X}}$$
  2. Cognitive independence of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with respect to m:
  $$pl(\mathbf{x}) = \prod_{i=1}^{n} pl_i(\mathbf{x}_i), \quad \forall \mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \Omega_{\mathbf{X}}.$$
- Under these assumptions:
$$\log L(\boldsymbol{\theta}; m) = \sum_{i=1}^{n} \log \mathbb{E}_{\boldsymbol{\theta}} \left[pl_i(\mathbf{X}_i) \right].$$

Evidential EM algorithm

Description

- The generalized log-likelihood function $\log L(\boldsymbol{\theta}; m)$ can be maximized using an iterative algorithm composed of two steps:

  E-step: Compute the expectation of $\log L(\theta; \mathbf{x})$ with respect to $m \oplus p_{\mathbf{X}}(\cdot; \boldsymbol{\theta}^{(q)})$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = \frac{\sum_{\mathbf{x}\in\Omega_X} \log(L(\boldsymbol{\theta}; \mathbf{x})) p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\theta}^{(q)}) pl(\mathbf{x})}{\sum_{\mathbf{x}\in\Omega_X} p_{\mathbf{X}}(\mathbf{x}; \boldsymbol{\theta}^{(q)}) pl(\mathbf{x})}.$$

  M-step: Maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ with respect to $\boldsymbol{\theta}$.

- E- and M-steps are iterated until the increase of $\log L(\boldsymbol{\theta}; m)$ becomes smaller than some threshold.

Evidential EM algorithm

Properties

1. When m is categorical: $m(A) = 1$ for some $A \subseteq \Omega$, then the previous algorithm reduces to the EM algorithm $\rightarrow$ evidential EM (E^{2}M) algorithm.
2. Monotonicity: any sequence $L(\boldsymbol{\theta}^{(q)}; m)$ for $q = 0, 1, 2, \ldots$ of generalized likelihood values obtained using the E^{2}M algorithm is non decreasing, i.e., it verifies

$$L(\boldsymbol{\theta}^{(q+1)}; m) \geq L(\boldsymbol{\theta}^{(q)}; m), \quad \forall q.$$

3. The algorithm only uses the contour function pl, which drastically reduces the complexity of calculations.

Example: uncertain Bernoulli sample

Model and data

- Let us assume that the complete data $\mathbf{x} = (x_1, \ldots, x_n)$ is a realization from an i.i.d. sample $X_1, \ldots, X_n$ from $\mathcal{B}(\theta)$ with $\theta \in [0, 1]$.
- We only have partial information about the x_i's in the form: $pl_1, \ldots, pl_n$, where $pl_i(x)$ is the plausibility that $X_i = x$, $x \in \{0, 1\}$.
- Under the cognitive independence assumption:

$$\log L(\theta; pl_1, \ldots, pl_n) = \sum_{i=1}^{n} \log \left[(1-\theta) pl_i(0) + \theta pl_i(1)\right].$$

E- and M-steps

Complete data log-likelihood:

$$\log L(\theta, \mathbf{x}) = n \log(1-\theta) + \log\left(\frac{\theta}{1-\theta}\right) \sum_{i=1}^{n} x_i.$$

E-step: compute

$$Q(\theta, \theta^{(q)}) = n \log(1-\theta) + \log\left(\frac{\theta}{1-\theta}\right) \sum_{i=1}^{n} \xi_i^{(q)}, \text{ with}$$

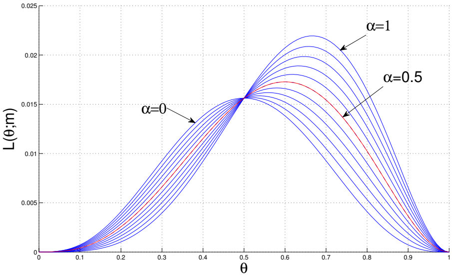
$$\xi_i^{(q)} = \mathbb{E}_{\theta^{(q)}}[X_i | pl_i] = \frac{\theta^{(q)} pl_i(1)}{(1-\theta^{(q)}) pl_i(0) + \theta^{(q)} pl_i(1)}.$$

M-step:

$$\theta^{(q+1)} = \frac{1}{n} \sum_{i=1}^{n} \xi_i^{(q)}.$$

Numerical example

i	1	2	3	4	5	6
$pl_i(0)$	1	1	1	α	0	0
$pl_i(1)$	0	0	0	$1-\alpha$	1	1



$\alpha = 0.5$

q	$\theta^{(q)}$	$L(\theta^{(q)}; pl)$
0	0.3000	6.6150
1	0.5500	16.8455
2	0.5917	17.2676
3	0.5986	17.2797
4	0.5998	17.2800
5	0.6000	17.2800

$\widehat{\theta} = 0.6$

Outline

Object classification

Problem statement

- We consider a population of objects partitioned in g classes.
- Each object is described by d continuous features $\mathbf{W} = (W^1, \ldots, W^d)$ and a class variable Z.
- The goal of discriminant analysis is to learn a decision rule that classifies any object from its feature vector, based on a learning set.

Object classification

Learning tasks

- Classically, different learning tasks are considered:
  Supervised learning: $\mathcal{L}_s = \{(\mathbf{w}_i, z_i)\}_{i=1}^n$;
  Unsupervised learning: $\mathcal{L}_{ns} = \{\mathbf{w}_i\}_{i=1}^n$;
  Semi-supervised learning: $\mathcal{L}_{ss} = \{(\mathbf{w}_i, z_i)\}_{i=1}^{n_s} \cup \{\mathbf{w}_i\}_{i=n_s}^n$
- Here, we consider partially supervised learning:

$$\mathcal{L}_{ps} = \{(\mathbf{w}_i, m_i)\}_{i=1}^n,$$

  where m_i is a mass function representing partial information about the class of object i.
- This problem can be solved using the E^2M algorithm using a suitable parametric model.

Linear discriminant analysis

- Generative model:
  - Complete data: $\mathbf{x} = \{(\mathbf{w}_i, z_i)\}_{i=1}^n$, assumed to be a realization of an iid random sample $\mathbf{X} = \{(\mathbf{W}_i, Z_i)\}_{i=1}^n$;
  - Given $Z_i = k$, $\mathbf{W}_i$ is multivariate normal with mean $\boldsymbol{\mu}_k$ and common variance matrix Σ.
  - The proportion of class k in the population is π_k.
  - Parameter vector: $\boldsymbol{\theta} = \left(\{\pi_k\}_{k=1}^g, \{\boldsymbol{\mu}_k\}_{k=1}^g, \Sigma\right)$.
- The Bayes rule is approximated by assigning each object to the class k^* that maximizes the estimated posterior probability

$$p(Z = k|\mathbf{w}; \widehat{\boldsymbol{\theta}}) = \frac{\phi(\mathbf{w}; \widehat{\boldsymbol{\mu}}_k, \widehat{\Sigma})\widehat{\pi}_k}{\sum_\ell \phi(\mathbf{w}; \widehat{\boldsymbol{\mu}}_\ell, \widehat{\Sigma})\widehat{\pi}_\ell},$$

where $\widehat{\boldsymbol{\theta}}$ is the MLE of $\boldsymbol{\theta}$.

Observed-data likelihood

- In partially supervised learning, the observed-data log-likelihood has the following expression:

$$\log L(\theta; \mathcal{L}_{ps}) = \sum_{i,k}^{n} pl_{ik} \log\left(\pi_k \phi(\mathbf{w}_i; \boldsymbol{\mu}_k, \Sigma_k)\right),$$

where pl_{ik} is the plausibility that object i belongs to class k.

- Supervised learning is recovered as a special case when:

$$pl_{ik} = z_{ik} = \begin{cases} 1 & \text{if object } i \text{ belongs to class } k; \\ 0 & \text{otherwise.} \end{cases}$$

- Unsupervised learning is recovered when $pl_{ik} = 1$ for all i and k.

E^{2}M algorithm

E-step: Using $p_{\mathbf{X}}(\cdot;\theta^{(q)}) \oplus m$, compute

$$t_{ik}^{(q)} = \mathbb{E}(Z_{ik}|m;\theta^{(q)}) = \frac{\pi_k^{(q)} pl_{ik}\phi(\mathbf{w}_i;\boldsymbol{\mu}_k^{(q)},\Sigma^{(q)})}{\sum_\ell \pi_k^{(q)} pl_{i\ell}\phi(\mathbf{w}_i;\boldsymbol{\mu}_\ell^{(q)},\Sigma^{(q)})}$$

M-step: Update parameter estimates

$$\pi_k^{(q+1)} = \frac{1}{n}\sum_{i=1}^n t_{ik}^{(q)}, \qquad \boldsymbol{\mu}_k^{(q+1)} = \frac{\sum_{i=1}^n t_{ik}^{(q)}\mathbf{w}_i}{\sum_{i=1}^n t_{ik}^{(q)}}.$$

$$\Sigma^{(q+1)} = \frac{1}{n}\sum_{i,k} t_{ik}^{(q)}(\mathbf{w}_i - \boldsymbol{\mu}_k^{(q+1)})(\mathbf{w}_i - \boldsymbol{\mu}_k^{(q+1)})'$$

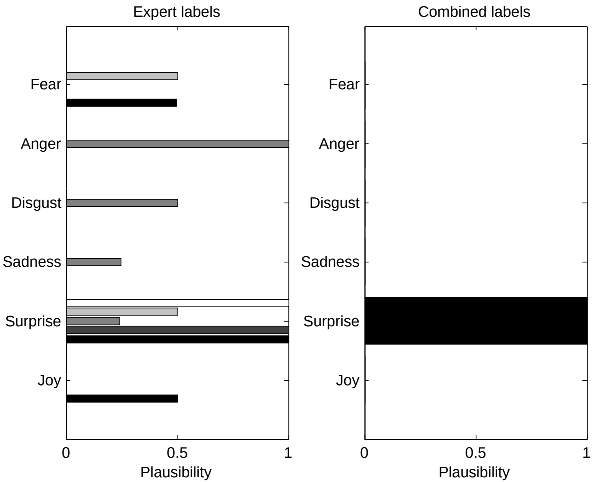
Face recognition problem

Experimental settings

- 216 images of 60 $\times$ 70 pixels, 36 in each class.
- One half for training, the rest for testing.
- A reduced number of features was extracted using Principal component analysis (PCA).
- Each training image was labeled by 5 subjects who gave degrees of plausibility for each image and each class.
- The plausibilities were combined using Dempster's rule (after some discounting to avoid total conflict).

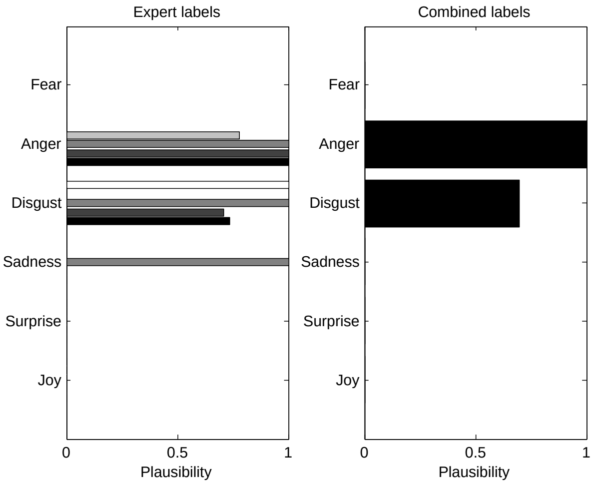
Combined labels

Example 1

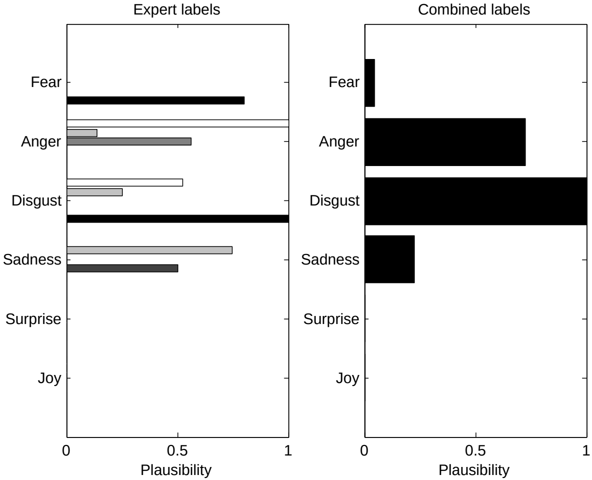


Combined labels

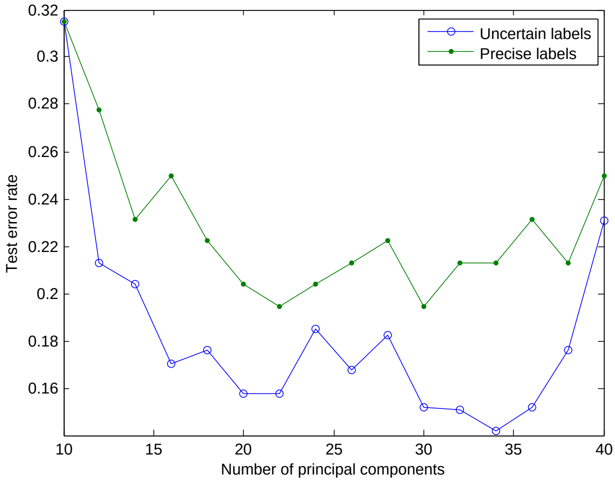
Example 2



Combined labels
Example 3

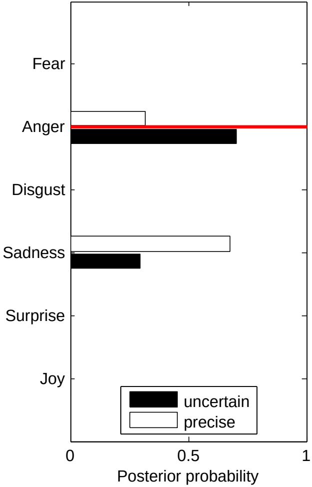


Results


Uncertain labels
Precise labels
Test error rate
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
10
15
20
25
30
35
40
Number of principal components

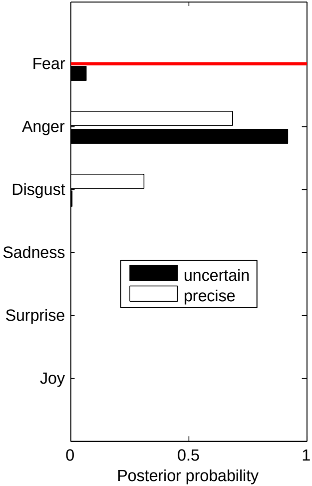
Results
Example 1

Test image 14



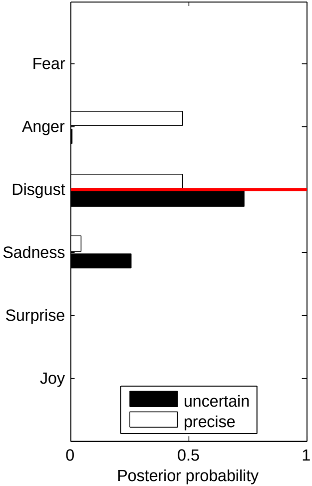
Results
Example 2

Test image 37



Results
Example 3

Test image 48



Summary

- The formalism of belief functions provides a very general setting for representing data uncertainty $\rightarrow$ evidential data
- Maximizing the proposed generalized likelihood amounts to minimizing the conflict between the uncertain data and the parametric model
- This can be achieved using an iterative algorithm (evidential EM algorithm) that reduces to the standard EM algorithm in special cases
- In classification, the method makes it possible to handle uncertainty on class labels (partially supervised learning). Uncertainty on attributes can be handled as well

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

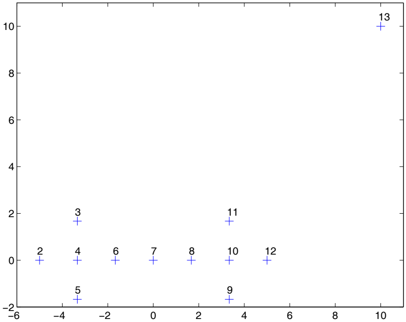
Possibilistic, rough, credal clustering

- 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 $(\underline{u}_{ik}, \overline{u}_{ik}) \in \{0, 1\}^2$ indicating its membership to the lower and upper approximations of cluster k
- Credal clustering: based on Dempster-Shafer (DS) theory (the topic of this talk)

Outline

Evidential clustering

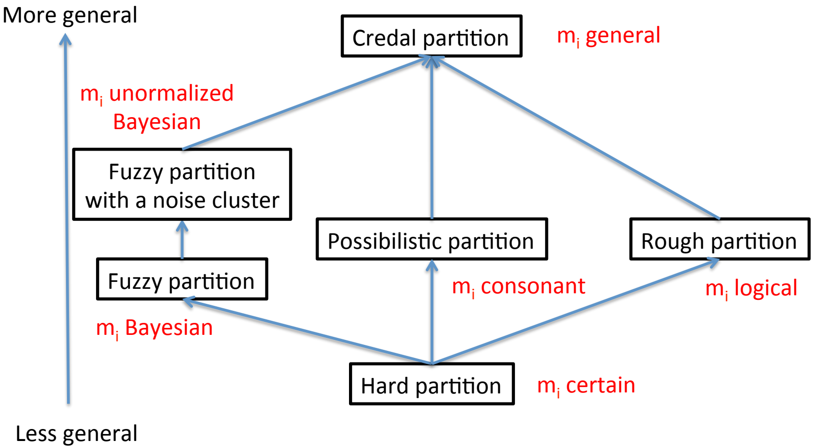
- In Credal clustering, the cluster membership of each object is considered to be uncertain and is described by a (not necessarily normalized) mass function m_i over Ω.
- Example:



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



Algorithms

1. EVCLUS (Denoeux and Masson, 2004):
   - Proximity (possibly non metric) data
   - Multidimensional scaling approach
   - Variant: Constrained EVCLUS (CEVCLUS) (Antoine et al., 2014): EVCLUS with pairwise constraints
2. Evidential *c*-means (ECM): (Masson and Denoeux, 2008):
   - Attribute data
   - HCM, FCM family (alternate optimization of a cost function)
   - Variants
     - Relational Evidential *c*-means (RECM): (Masson and Denoeux, 2009): ECM for proximity data
     - Constrained Evidential *c*-means (CECM) (Antoine et al., 2011): ECM with pairwise constraints
     - Spatial Evidential C-Means (Lelandais et al., 2014): ECM with spatial constraints, for image segmentation
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 $M = (m_1, \ldots, m_n)$ from attribute data $X = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $\mathbf{x}_i \in \mathbb{R}^p$.
- Generalization of hard and fuzzy c-means algorithms:
  - Each class represented by a prototype;
  - Alternate 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 $d_{ik}=||\mathbf{x}_i-\mathbf{v}_k||$ under the constraints $\sum_k u_{ik}=1, \forall i$.

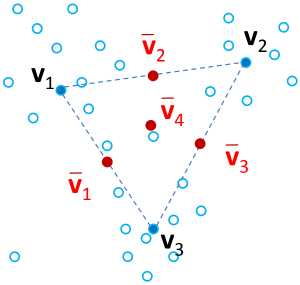
- Alternate optimization algorithm:

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

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

ECM algorithm

Principle



- Each class ω_k represented by a prototype $\mathbf{v}_k$.
- Each non empty set of classes 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 non empty $A_j \in \Omega$, $m_{ij} = m_i(A_j)$ 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 δ.

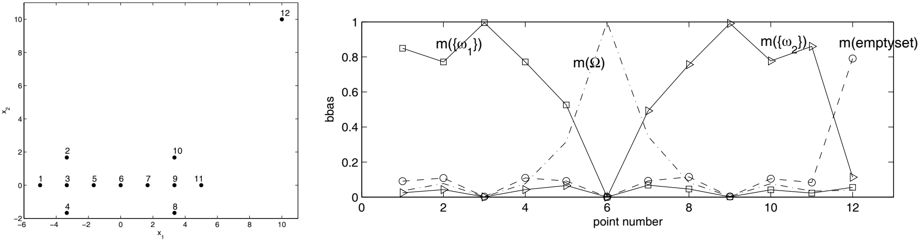
ECM algorithm
Objective criterion

- Criterion to be minimized:

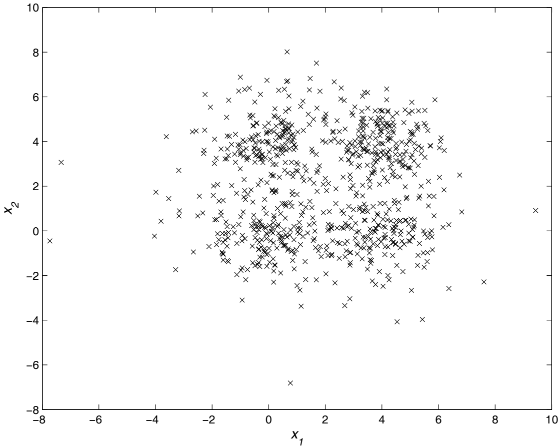
$$J_{\text{ECM}}(M,V)=\sum_{i=1}^{n}\sum_{\{j/A_j\neq\emptyset,A_j\subseteq\Omega\}}|A_j|^{\alpha}m_{ij}^{\beta}d_{ij}^{2}+\sum_{i=1}^{n}\delta^2 m_{i\emptyset}^{\beta},$$

- Parameters:
  - α controls the specificity of mass functions;
  - β controls the hardness of the evidential partition;
  - δ controls the amount of data considered as outliers.
- $J_{\text{ECM}}(M,V)$ can be iteratively minimized with respect to M and V using an alternate optimization scheme.

Butterfly dataset

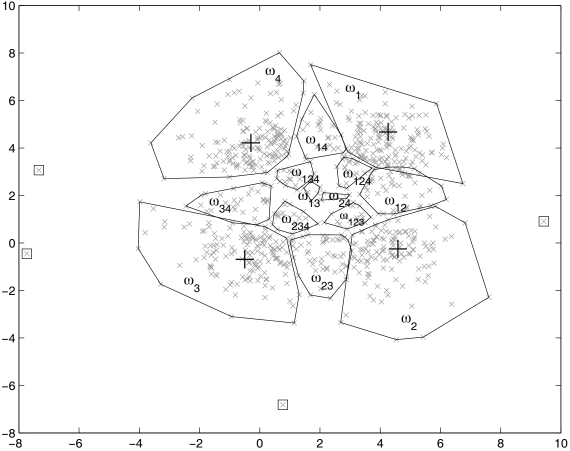


4-class data set

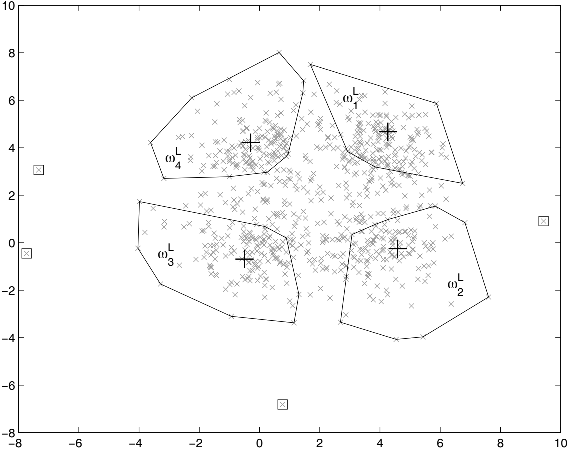


4-class data set

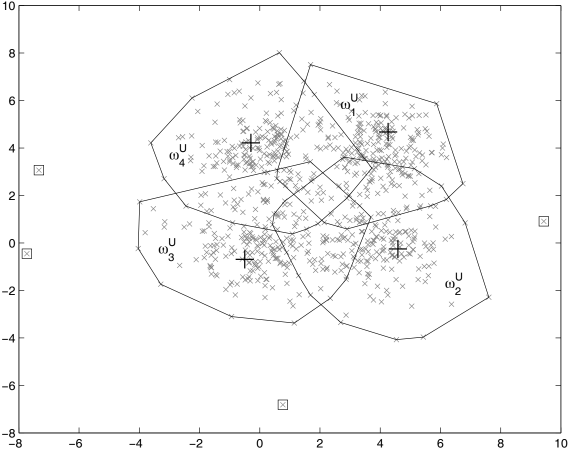
Hard credal partition



4-class data set
Lower approximation



4-class data set
Upper approximation



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
- 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 as the base classifier

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 compute

$$\alpha_{ij} = \begin{cases} \varphi(d_{ij}) & \text{if } j \in N_K(i) \\ 0 & \text{otherwise,} \end{cases}$$
$$v_{ij} = -\ln(1 - \alpha_{ij})$$

  for all $(i, j) \in \{1, \ldots, n\}^2$
- 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
- As before, we define cluster-membership binary variables s_{ik} as $s_{ik} = 1$ is object o_i belongs to cluster k, and $s_{ik} = 0$ 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, we compute the logarithms of the plausibilities of belonging to each cluster (up to an additive constant) as

$$u_{ik} = \sum_{j \in N_K(i)} v_{ij} s_{jk}, \quad k = 1, \ldots, c$$

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

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

- 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, we move to the last step described below, and the algorithm is stopped

EK-NNclus algorithm

Step 4: Computation of the credal partition

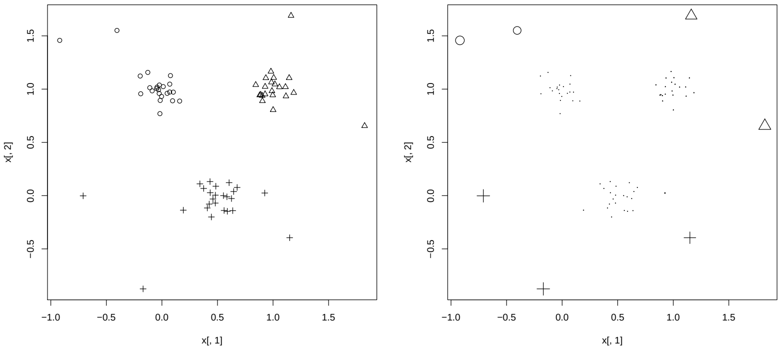
After the algorithm has converged, we can compute the final mass functions

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

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

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

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

$$E(S) = -\frac{1}{2}\sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j\neq i} v_{ij} s_{ik} s_{jk}$$

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

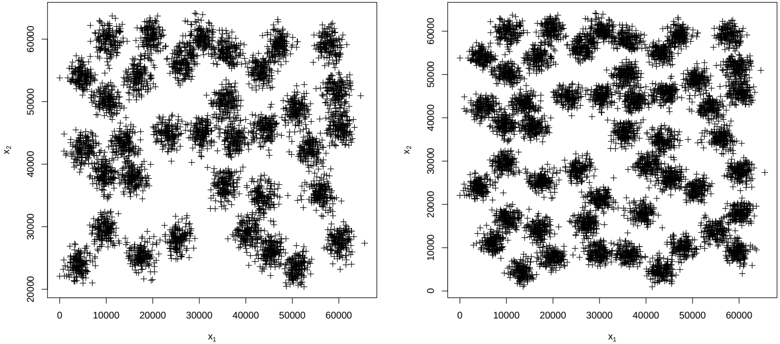
$$pl(R) = -E(S) + C$$

  where $pl(R)$ is the plausibility of the partition encoded by S
- 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
  - Parameter γ was fixed to the inverse of the q-quantile of the set $\Delta = \{d_{ij}^2, i \in \{1, \ldots, n\}, j \in N_K(i)\}$
- 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	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

Summary

- The theory of belief function has great potential for solving challenging machine learning problems:
  - 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)
- Recent developments make it possible to address problems in very large frames (multilabel classification, clustering, preference learning, etc.)

References I

cf. http://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. Denoeux.
Maximum likelihood estimation from Uncertain Data in the Belief Function Framework.
IEEE Transactions on Knowledge and Data Engineering, Vol. 25, Issue 1, pages 119-130, 2013.

References II

cf. http://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.

- V. Antoine, B. Quost, M.-H. Masson and T. Denoeux.
  CECM: Constrained Evidential C-Means algorithm.
  Computational Statistics and Data Analysis, 56(4):894-914, 2012.

- 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. 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.