Theory of Belief Functions: Application to machine learning and statistical inference

Lecture 2: Decision analysis

Thierry Denœux

Summer 2023

Outline

1. Decision analysis
   - Decision-making under complete ignorance
   - Decision-making with probabilities
   - Decision-making with belief functions

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

Outline

1. Decision analysis
   - Decision-making under complete ignorance
   - Decision-making with probabilities
   - Decision-making with belief functions

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

Example of decision problem under uncertainty

Act (Purchase)	Good Economic Conditions	Poor Economic Conditions
Apartment building	50,000	30,000
Office building	100,000	-40,000
Warehouse	30,000	10,000

Formal framework

Acts, outcomes, states of nature

- A decision problem can be seen as a situation in which a decision-maker (DM) has to choose a course of action (an act) in some set $\mathcal{F} = \{f_1, \ldots, f_n\}$
- An act may have different consequences (outcomes), depending on the state of nature
- Denoting by $\Omega = \{\omega_1, \ldots, \omega_r\}$ the set of states of nature and by $\mathcal{C}$ the set of consequences (or outcomes), an act can be formalized as a mapping f from Ω to $\mathcal{C}$
- In this lecture, the three sets Ω, $\mathcal{C}$ and $\mathcal{F}$ will be assumed to be finite

Formal framework

Utilities

- The desirability of the consequences can often be modeled by a numerical utility function $u : \mathcal{C} \to \mathbb{R}$, which assigns a numerical value to each consequence
- The higher this value, the more desirable is the consequence for the DM
- In some problems, the consequences can be evaluated in terms of monetary value. The utilities can then be defined as the payoffs, or a function thereof
- If the actions are indexed by i and the states of nature by j, we will denote by u_{ij} the quantity $u[f_i(\omega_j)]$
- The $n \times r$ matrix $U = (u_{ij})$ will be called a payoff or utility matrix

Payoff matrix

Act (Purchase)	Good Economic Conditions (ω_1)	Poor Economic Conditions (ω_2)
Apartment building (f_1)	50,000	30,000
Office building (f_2)	100,000	-40,000
Warehouse (f_3)	30,000	10,000

Formal framework

Preferences

- If the true state of nature ω is known, the desirability of an act f can be deduced from that of its consequence $f(\omega)$
- Typically, the state of nature is unknown. Based on partial information, it is usually assumed that the DM can express preferences among acts, which may be represented mathematically by a preference relation $\succcurlyeq$ on $\mathcal{F}$
- This relation is interpreted as follows: given two acts f and g, $f \succcurlyeq g$ means that f is found by the DM to be at least as desirable as g
- We also define
  - The strict preference relation as $f \succ g$ iff $f \succcurlyeq g$ and not($g \succcurlyeq f$) (meaning that f is strictly more desirable than g) and
  - The indifference relation $f \sim g$ iff $f \succcurlyeq g$ and $g \succcurlyeq f$ (meaning that f and g are equally desirable)

Decision problems

- The decision problem can be formalized as building a preference relation among acts, from a utility matrix and some description of uncertainty, and finding the maximal elements of this relation
- Depending on the nature of the available information, different decision problems arise:
  1. Decision-making under ignorance
  2. Decision-making with probabilities
  3. Decision-making with belief functions

Outline

Problem and non-domination principle

- We assume that the DM is totally ignorant of the state of nature: all the information given to the DM is the utility matrix U
- A act f_i is said to be dominated by f_k if the outcomes of f_k are at least as desirable as those of f_i for all states, and strictly more desirable for at least one state

$$\forall j, \; u_{kj} \geq u_{ij} \text{ and } \exists j, \; u_{kj} > u_{ij}$$

- Non-domination principle: an act cannot be chosen if it is dominated by another one

Example of a dominated act

Act (Purchase)	Good Economic Conditions (ω_1)	Poor Economic Conditions (ω_2)
Apartment building (f_1)	50,000	30,000
Office building (f_2)	100,000	-40,000
~~Warehouse (f_3)~~	~~30,000~~	~~10,000~~

Criteria for rational choice

- After all dominated acts have been removed, there remains the problem of ordering them by desirability, and of finding the set of most desirable acts
- Several criteria of “rational choice” have been proposed to derive a preference relation over acts, including:
  1. Maximax criterion
  $$f_i \succeq f_k \text{ iff } \max_j u_{ij} \geq \max_j u_{kj}.$$
  2. Maximin (Wald) criterion
  $$f_i \succeq f_k \text{ iff } \min_j u_{ij} \geq \min_j u_{kj}.$$
  3. Laplace criterion
  $$f_i \succeq f_k \text{ iff } \frac{1}{r}\sum_j u_{ij} \geq \frac{1}{r}\sum_j u_{kj}.$$

Example

Act	ω_1	ω_2	ave	max	min
Apartment (f_1)	50,000	30,000	**40,000**	50,000	**30,000**
Office (f_2)	100,000	-40,000	30,000	**100,000**	-40,000

Hurwicz criterion

- Hurwicz criterion: $f_i \succeq f_k$ iff

$$\alpha \min_j u_{ij} + (1-\alpha) \max_j u_{ij} \geq \alpha \min_j u_{kj} + (1-\alpha) \max_j u_{kj}$$

  where α is a parameter in $[0,1]$, called the pessimism index
- Boils down to
  - the maximax criterion if $\alpha = 0$
  - the maximin criterion if $\alpha = 1$
- α describes the DM's attitude toward ambiguity.
- Formal justification given by Arrow and Hurwicz (1972).

Outline

Lottery

- Let us now consider the situation where uncertainty about the state of nature is quantified by a probability distribution π on Ω.
- These probabilities can be objective (decision under risk) or subjective.
- An act $f : \Omega \to \mathcal{C}$ induces a probability distribution p_f on the set $\mathcal{C}$ of consequences (assumed to be finite), called a lottery:

$$\forall c \in \mathcal{C}, \quad p_f(c) = \sum_{\{\omega : f(\omega)=c\}} \pi(\omega).$$

Maximum Expected Utility principle

- Given a utility function $u : \mathcal{C} \to \mathbb{R}$, the expected utility for a lottery p is

$$\mathbb{E}_p(u) = \sum_{c \in \mathcal{C}} u(c)p(c).$$

- Maximum Expected Utility (MEU) principle: a lottery p_i is more desirable than a lottery p_k if it has a higher expected utility:

$$p_i \succeq p_k \Leftrightarrow \mathbb{E}_{p_i}(u) \geq \mathbb{E}_{p_k}(u).$$

- The MEU principle was first axiomatized by von Neumann and Morgenstern (1944).

Example

Act	ω_1	ω_2
Apartment (f_1)	50,000	30,000
Office (f_2)	100,000	-40,000

- Assume that there is 60% chance that the economic situation will be poor (ω_2).
- Act f_1 induces the lottery p_1 such that $p_1(50,000) = 0.4$ and $p_1(30,000) = 0.6$. Act f_2 induces the lottery p_2 such that $p_2(100,000) = 0.4$ and $p_2(-40,000) = 0.6$.
- The expected utilities are

$$\mathbb{E}_{p_1}(u) = 50,000 \times 0.4 + 30,000 \times 0.6 = 38,000$$
$$\mathbb{E}_{p_2}(u) = 100,000 \times 0.4 - 40,000 \times 0.6 = 16,000$$

- Act f_1 is thus more desirable according to the maximum expected utility criterion.

Outline

1. Decision analysis
   - Decision-making under complete ignorance
   - Decision-making with probabilities
   - Decision-making with belief functions

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

How belief functions come into the picture

Belief functions become components of a decision problem in any of the following two situations (or both)

1. The DM's subjective beliefs concerning the state of nature are described by a belief function Bel^Ω on Ω
2. The DM is not able to precisely describe the outcomes of some acts under each state of nature

Case 1: uncertainty described by a belief function

- Let m^Ω be a mass function on Ω
- Any act $f : \Omega \rightarrow \mathcal{C}$ carries m^Ω to the set $\mathcal{C}$ of consequences, yielding a mass function $m_f^{\mathcal{C}}$, which quantifies the DM's beliefs about the outcome of act f
- Each mass $m^\Omega(A)$ is transferred to $f(A)$

$$m_f^{\mathcal{C}}(B) = \sum_{\{A \subseteq \Omega : f(A) = B\}} m^\Omega(A)$$

  for any $B \subseteq \mathcal{C}$
- $m_f^{\mathcal{C}}$ is a credibilistic lottery corresponding to act f

Case 2: partial knowledge of outcomes

- In that case, an act may formally be represented by a multi-valued mapping $f : \Omega \rightarrow 2^{\mathcal{C}}$, assigning a set of possible consequences $f(\omega) \subseteq \mathcal{C}$ to each state of nature ω
- Given a probability measure P on Ω, f then induces the following mass function $m_f^{\mathcal{C}}$ on $\mathcal{C}$,

$$m_f^{\mathcal{C}}(B) = \sum_{\{\omega\in\Omega: f(\omega)=B\}} p(\omega)$$

for all $B \subseteq \mathcal{C}$

Example

- Let $\Omega = \{\omega_1, \omega_2, \omega_3\}$ and m^Ω the following mass function

$$\begin{array}{ll} m^\Omega(\{\omega_1,\omega_2\}) = 0.3, & m^\Omega(\{\omega_2,\omega_3\}) = 0.2 \\ m^\Omega(\{\omega_3\}) = 0.4, & m^\Omega(\Omega) = 0.1 \end{array}$$

- Let $\mathcal{C} = \{c_1, c_2, c_3\}$ and f the act

$$f(\omega_1) = \{c_1\}, \quad f(\omega_2) = \{c_1, c_2\}, \quad f(\omega_3) = \{c_2, c_3\}$$

- To compute $m_f^{\mathcal{C}}$, we transfer the masses as follows

$$\begin{aligned} m^\Omega(\{\omega_1,\omega_2\}) &= 0.3 \rightarrow f(\omega_1) \cup f(\omega_2) = \{c_1, c_2\} \\ m^\Omega(\{\omega_2,\omega_3\}) &= 0.2 \rightarrow f(\omega_2) \cup f(\omega_3) = \{c_1, c_2, c_3\} \\ m^\Omega(\{\omega_3\}) &= 0.4 \rightarrow f(\omega_3) = \{c_2, c_3\} \\ m^\Omega(\Omega) &= 0.1 \rightarrow f(\omega_1) \cup f(\omega_2) \cup f(\omega_3) = \{c_1, c_2, c_3\} \end{aligned}$$

- Finally, we obtain the following mass function on $\mathcal{C}$

$$m^{\mathcal{C}}(\{c_1, c_2\}) = 0.3, \quad m^{\mathcal{C}}(\{c_2, c_3\}) = 0.4, \quad m^{\mathcal{C}}(\mathcal{C}) = 0.3$$

Decision problem

- In the two situations considered above, we can assign to each act f a credibilistic lottery, defined as a mass function on $\mathcal{C}$
- Given a utility function u on $\mathcal{C}$, we then need to extend the MEU model
- Several such extensions will now be reviewed

Upper and lower expectations

- Let m be a mass function on $\mathcal{C}$, and u a utility function $\mathcal{C} \rightarrow \mathbb{R}$
- The lower and upper expectations of u are defined, respectively, as the averages of the minima and the maxima of u within each focal set of m

$$\underline{\mathbb{E}}_m(u) = \sum_{A \subseteq \mathcal{C}} m(A) \min_{c \in A} u(c)$$

$$\overline{\mathbb{E}}_m(u) = \sum_{A \subseteq \mathcal{C}} m(A) \max_{c \in A} u(c)$$

- It is clear that $\underline{\mathbb{E}}_m(u) \leq \overline{\mathbb{E}}_m(u)$, with the inequality becoming an equality when m is Bayesian, in which case the lower and upper expectations collapse to the usual expectation
- If $m = m_A$ is logical with focal set A, then $\underline{\mathbb{E}}_m(u)$ and $\overline{\mathbb{E}}_m(u)$ are, respectively, the minimum and the maximum of u in A

Corresponding decision criteria

- Having defined the notions of lower and upper expectations, we can define two preference relations among credibilistic lotteries as

$$m_1 \underline{\succcurlyeq} m_2 \text{ iff } \underline{\mathbb{E}}_{m_1}(u) \geq \underline{\mathbb{E}}_{m_2}(u)$$

and

$$m_1 \overline{\succcurlyeq} m_2 \text{ iff } \overline{\mathbb{E}}_{m_1}(u) \geq \overline{\mathbb{E}}_{m_2}(u)$$

- Relation $\underline{\succcurlyeq}$ corresponds to a pessimistic (or conservative) attitude of the DM. When m is logical, it corresponds to the maximin criterion
- Symmetrically, $\overline{\succcurlyeq}$ corresponds to an optimistic attitude and extends the maximax criterion
- Both criteria boil down to the MEU criterion when mass functions are Bayesian.

Generalized Hurwicz criterion

- The Hurwicz criterion can be generalized as

$$\begin{aligned}\mathbb{E}_{m,\alpha}(u) &= \sum_{A \subseteq \mathcal{C}} m(A) \left(\alpha \min_{c \in A} u(c) + (1-\alpha) \max_{c \in A} u(c) \right) \\ &= \alpha \underline{\mathbb{E}}_m(u) + (1-\alpha) \overline{\mathbb{E}}(u)\end{aligned}$$

  where $\alpha \in [0,1]$ is a pessimism index
- This criterion was first introduced and justified axiomatically by Jaffray (1988)

Transferable belief model

- A completely different approach to decision-making with belief function was advocated by Smets, as part of the Transferable Belief Model
- Smets defended a two-level mental model
  1. A credal level, where an agent's beliefs are represented by belief functions, and
  2. A pignistic level, where decisions are made by maximizing the EU with respect to a probability measure derived from a belief function
- The rationale for introducing probabilities at the decision level is the avoidance of Dutch books
- Smets argued that the belief-probability transformation T should be linear, i.e., it should verify

$$T(\alpha m_1 + (1-\alpha)m_2) = \alpha T(m_1) + (1-\alpha)T(m_2),$$

for any mass functions m_1 and m_2 and for any $\alpha \in [0, 1]$

Pignistic transformation

- The only linear belief-probability transformation T is the pignistic transformation, with $p_m = T(m)$ given by

$$p_m(c) = \sum_{\{A \subseteq \mathcal{C} : c \in A\}} \frac{m(A)}{|A|}, \quad \forall c \in \mathcal{C}$$

- The expected utility w.r.t. the pignistic probability is

$$\mathbb{E}_p(u) = \sum_{c \in \mathcal{C}} p_m(c) u(c) = \sum_{A \subseteq \mathcal{C}} m(A) \left(\frac{1}{|A|} \sum_{c \in A} u(c) \right)$$

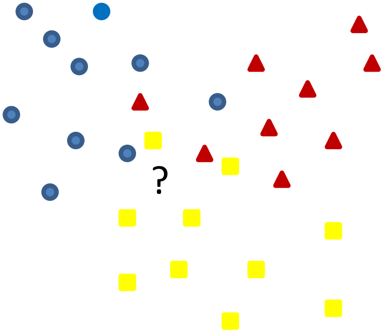
- The maximum pignistic expected utility criterion thus extends the Laplace criterion

Summary

non-probabilized		belief functions	probabilized
maximin	$\longleftrightarrow$	lower expectation	
maximax	$\longleftrightarrow$	upper expectation	
Laplace	$\longleftrightarrow$	pignistic expectation	expected utility
Hurwicz	$\longleftrightarrow$	generalized Hurwicz	

Outline

Classification problem



- A population is 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 $\mathbf{x}$

- The program that maps feature vectors to classes is called a classifier.

Example: expression recognition

joy

surprise

sadness

disgust

anger

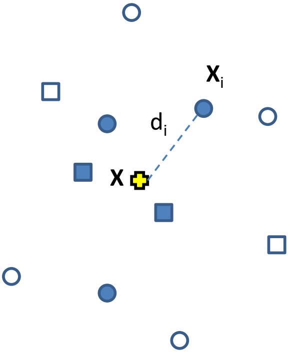
fear

Evidential classifier

- Sometimes, the class cannot be predicted from the feature vector with high certainty.
- Assessing the uncertainty in the classification is an important issue.
- Most traditional classifiers represent uncertainty by computing a conditional probability distribution $P(\cdot|\mathbf{x})$
- An evidential classifier represents classification uncertain using belief functions.
- There are several methods to construct evidential classifiers. We will see two of them:
  - The evidential K-NN classifier
  - The evidential neural network classifier

Outline

Principle



- 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 $\mathbf{x}_i$

Definition

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

$$m_i(\{\omega_k\}) = \varphi_k(d_i)\, y_{ik}, \quad k = 1, \ldots, c$$
$$m_i(\Omega) = 1 - \varphi_k(d_i)$$

  where $y_{ik} = I(y_i = \omega_k)$ and 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 $\mathbf{x}$ is pooled using Dempster's rule of combination

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

- The focal sets of m are the singletons and Ω.
- A decision can be made by selecting the class with the highest plausibility (see below).

Learning

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

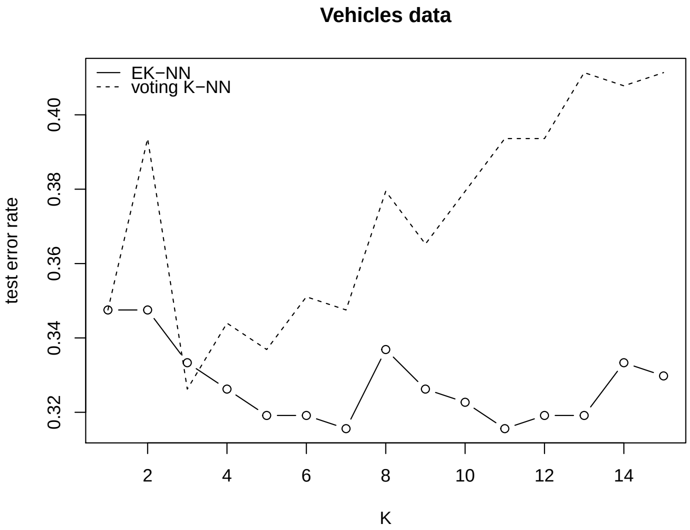
$$C(\gamma) = \sum_{i=1}^{n} \sum_{k=1}^{c} (pl_{(-i)}(\omega_k) - y_{ik})^2,$$

  where $pl_{(-i)}$ is the contour function obtained by classifying $\mathbf{x}_i$ using its K nearest neighbors in the learning set.
- Function $C(\gamma)$ can be minimized by an iterative nonlinear optimization algorithm.

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.

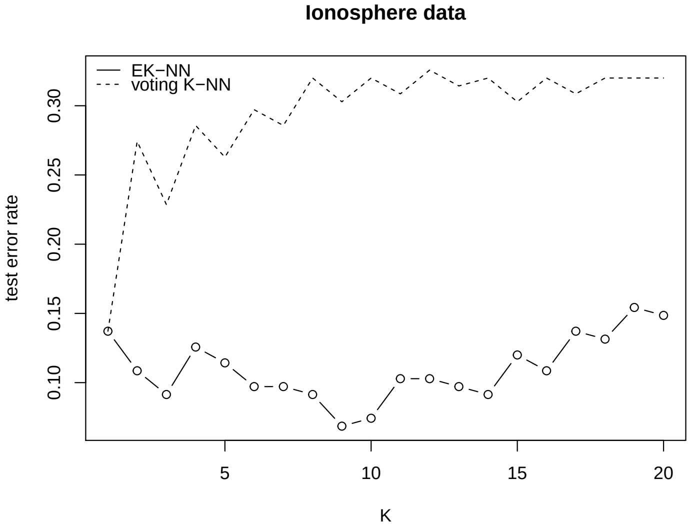
Vehicles datasets: result



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

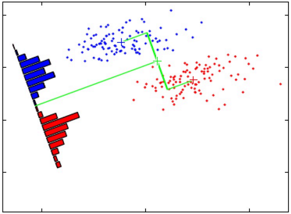


Implementation in R

```
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
```

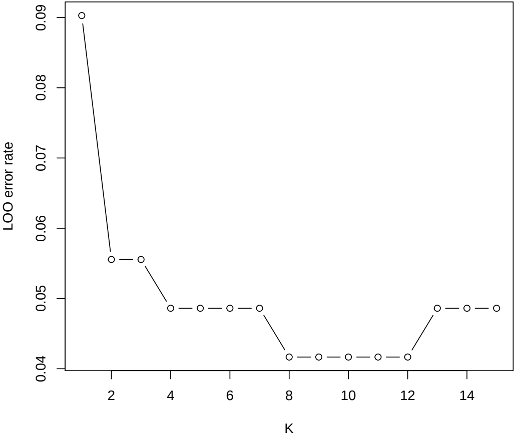
Face data

Projection in a 5D subspace (LDA) → evidential classification → decision



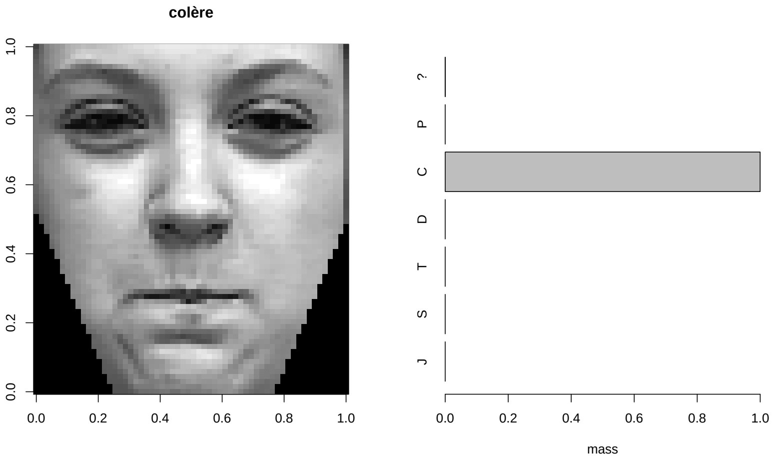
- 216 images 70 $\times$ 60 (36 per expression)
- 144 for learning, 72 for testing
- 5 features extracted by linear discriminant analysis

Face data: training

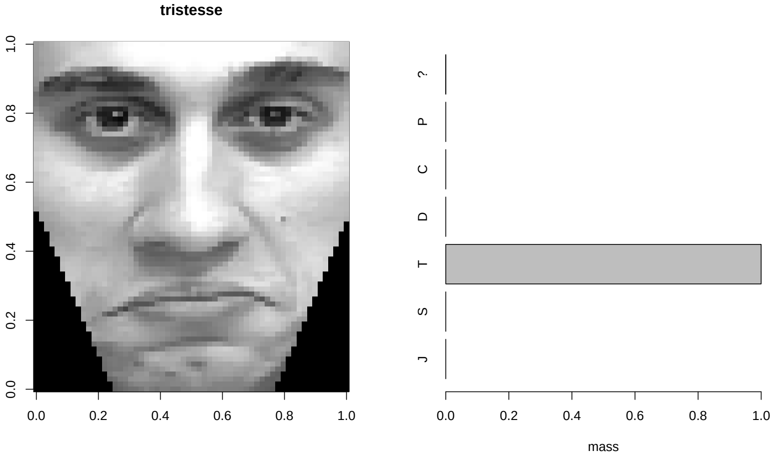



```
> print(val$err)
0.1527778
> table(ytst,val$ypred)

ytst  1  2  3  4  5  6
1  10  0  0  0  0  0
2  0  14  0  0  0  0
3  0  0  11  0  4  0
4  0  1  1  7  0  0
5  0  0  0  0  11  0
6  2  0  1  0  2  8
```

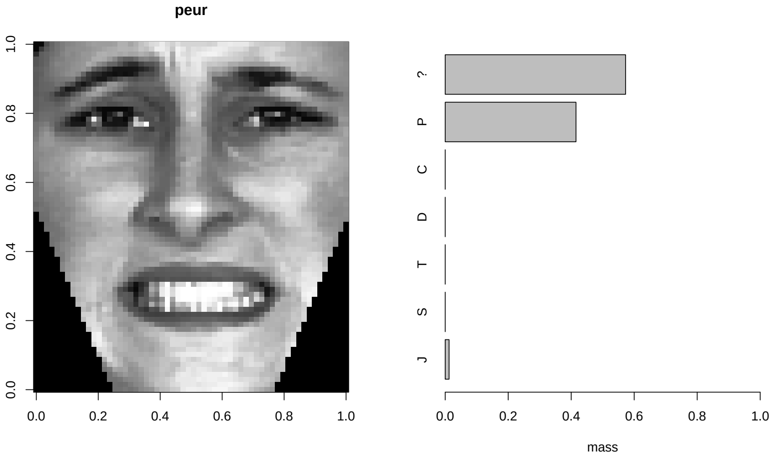
Results



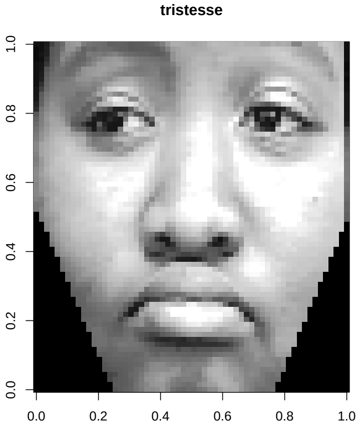
Results

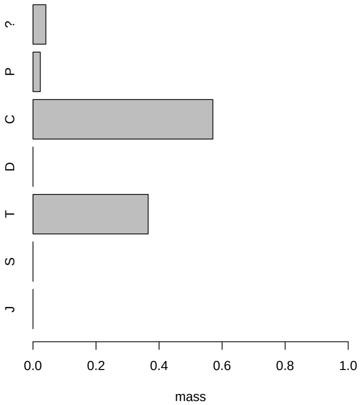


Results

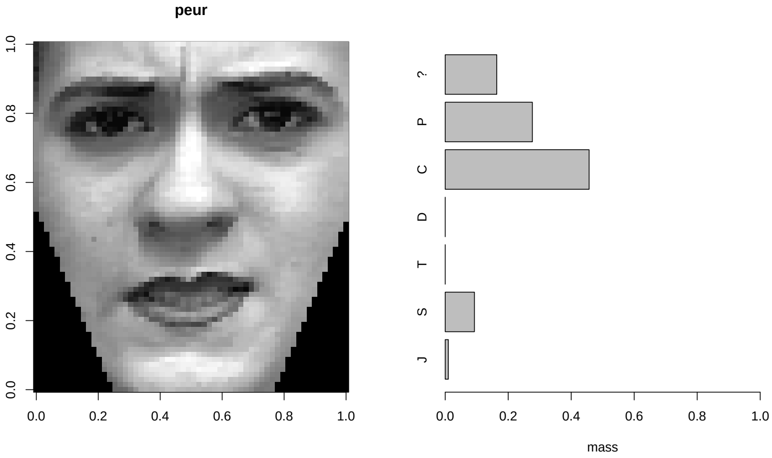


Results

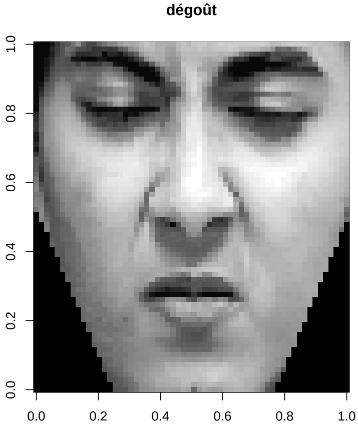

tristesse
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

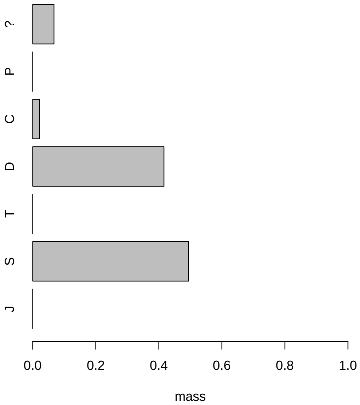

?
P
C
D
T
S
J
0.0
0.2
0.4
0.6
0.8
1.0
mass

Results


peur
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
?
P
C
D
T
S
J
0.0
0.2
0.4
0.6
0.8
1.0
mass

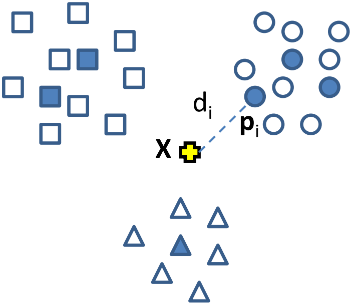
Results





Outline

Principle



- 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 $\mathbf{p}_i$ is a piece of evidence about the class of $\mathbf{x}$, whose reliability decreases with the distance d_i between $\mathbf{x}$ and $\mathbf{p}_i$.

Propagation equations

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

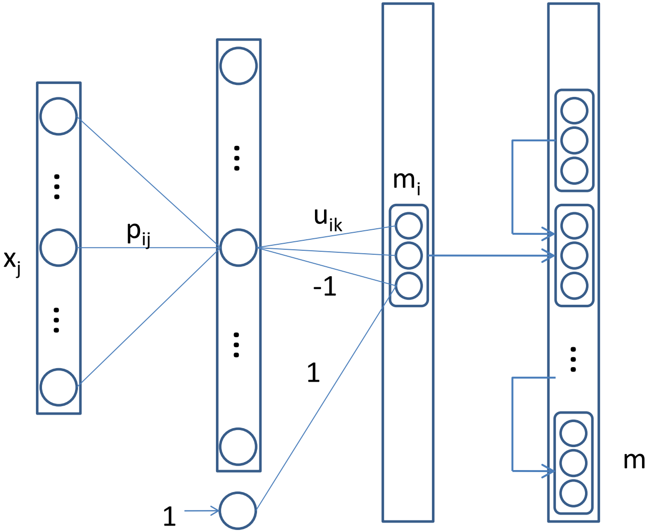
$$m_i(\{\omega_k\}) = \alpha_i u_{ik} \exp(-\gamma_i d_i^2), \quad k = 1, \ldots, c$$
$$m_i(\Omega) = 1 - \alpha_i \exp(-\gamma_i d_i^2)$$

- Combination:

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

- 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 $\mathbf{p}_i$, $i = 1, \ldots, r$ (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

$$C(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{c} (pl_{ik} - y_{ik})^2}_{\text{error}} + \mu \underbrace{\sum_{i=1}^{r} \alpha_i}_{\text{regularization}}$$

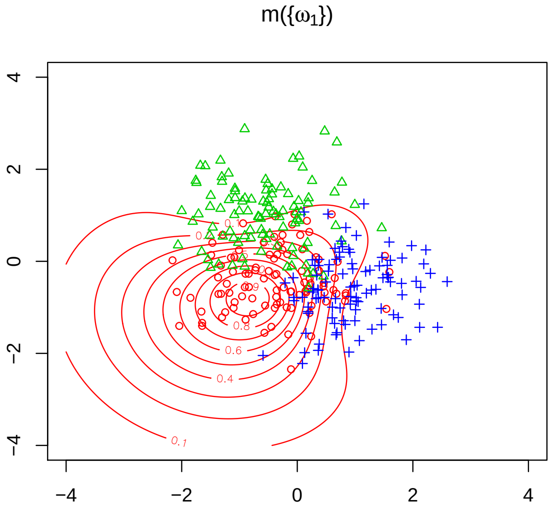
  where pl_{ik} is the output plausibility for instance i and class k, and μ is a regularization coefficient (hyperparameter).
- The hyperparameter μ can be optimized by cross-validation.

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
```

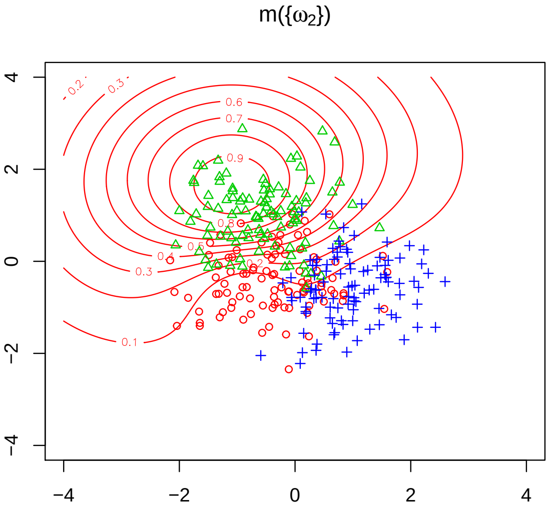
Example

Mass on $\{\omega_1\}$



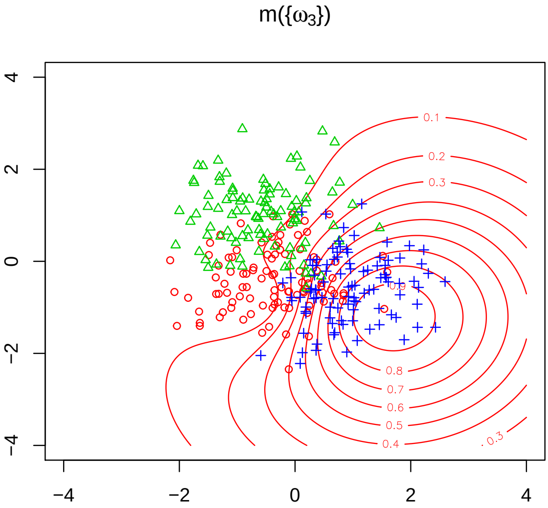
Example

Mass on $\{\omega_2\}$



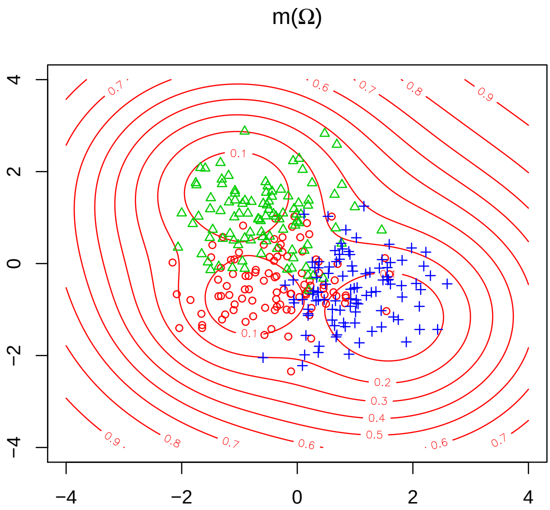
Example

Mass on $\{\omega_3\}$



Example

Mass on Ω



Outline

Simple decision setting

We have seen that, to formalize the decision problem, we need to define:

1. The set of consequences
2. The set of acts
3. The utility function

Simple decision setting

- Let
  - $\mathcal{C} = \{\text{correct}, \text{error}\}$
  - $\mathcal{F} = \{f_1, \ldots, f_c\}$ with f_k = assignment to class ω_k,

$$f_k(\omega_k) = \text{correct}, \quad f_k(\omega_\ell) = \text{error}, \ \forall \ell \neq k$$

  - $u(\text{correct}) = 1$, $u(\text{error}) = 0$
- In classification, we more often use the notion of loss, to be minimized. Here, the loss function can be defined as

$$\lambda(\text{correct}) = 0, \quad \lambda(\text{error}) = 1.$$

The expected loss is called the risk.

Simple decision setting (continued)

- Given a mass function m on Ω, act f_k induces the following mass m_k on $\mathcal{C}$:

$$m_k(\{\text{correct}\}) = m(\{\omega_k\}) = Bel(\{\omega_k\})$$
$$m_k(\{\text{error}\}) = \sum_{\omega_k \notin A} m(A) = 1 - Pl(\{\omega_k\})$$
$$m_k(\mathcal{C}) = \sum_{\omega_k \in A, |A|>1} m(A)$$

- The lower and upper risk are

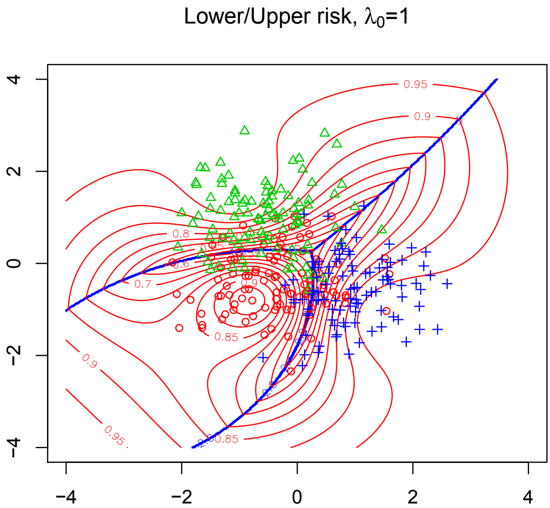
$$\begin{aligned}\underline{\mathbb{E}}_{m_k}(\lambda) &= m_k(\{\text{correct}\}) \times 0 + m_k(\{\text{error}\}) \times 1 + m_k(\mathcal{C}) \times 0 \\ &= 1 - Pl(\{\omega_k\}) \\ \overline{\mathbb{E}}_{m_k}(\lambda) &= m_k(\{\text{correct}\}) \times 0 + m_k(\{\text{error}\}) \times 1 + m_k(\mathcal{C}) \times 1 \\ &= 1 - Bel(\{\omega_k\})\end{aligned}$$

- When the focal sets of m are $\{\omega_k\}$, $k = 1, \ldots, c$ and Ω, the different decision rules (optimistic, pessimistic, Hurwicz, pignistic) are equivalent.

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')
```

Example


Lower/Upper risk, λ_0=1
0.95
0.9
0.8
0.7
0.6
0.85
0.9
0.95
0.85
4
2
0
−2
−4
−4
−2
0
2
4

Decision with rejection

- Let us now assume
  - $\mathcal{C} = \{\text{correct}, \text{error}, \text{reject}\}$
  - $\mathcal{F} = \{f_0, f_1, \ldots, f_c\}$, where f_0 denotes rejection,

$$f_0(\omega_k) = \text{reject}, \quad \forall k$$

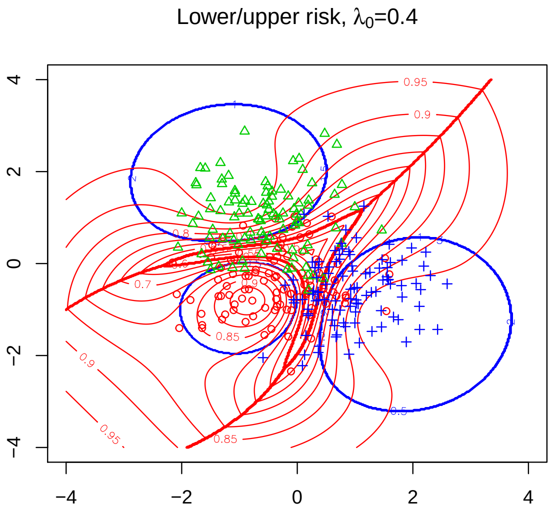
    and f_k = assignment to class ω_k, as before.
  - $\lambda(\text{correct}) = 0$, $\lambda(\text{error}) = 1$, $\lambda(\text{reject}) = \lambda_0$
- We can carry out the analysis as before. In this case, the different decision rules generally lead to different decisions.

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)
```

Example

Lower/upper risk, $\lambda_0 = 0.4$



Example

Hurwicz strategy ($\alpha = 0.5$), $\lambda_0 = 0.4$