Evidential Machine Learning

Thierry Denœux

Université de technologie de Compiègne, Compiègne, France
Institut Universitaire de France, Paris, France

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

7th School on Belief Functions and their Applications
Granada, Spain, October 23, 2025

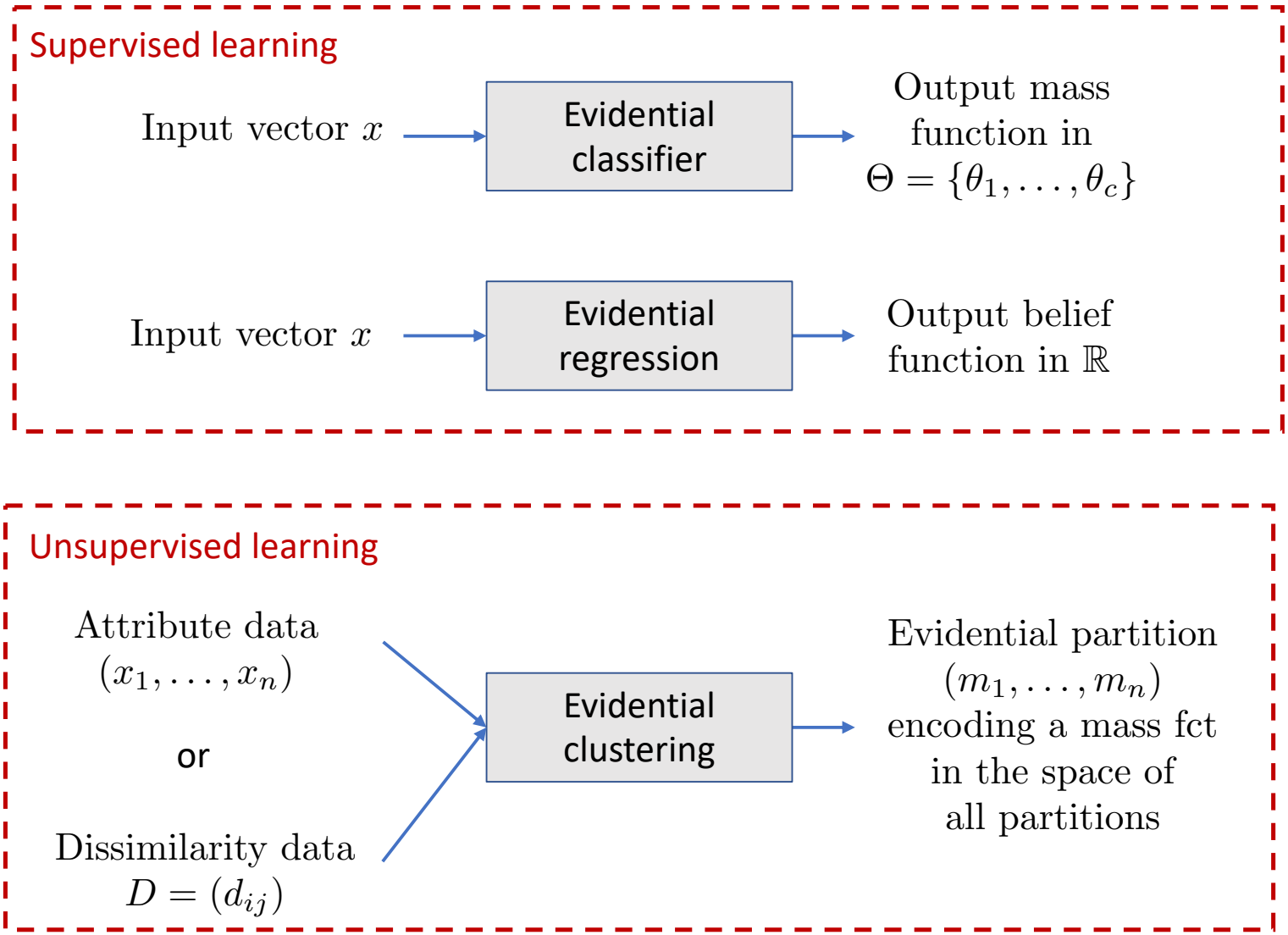
Uncertainty quantification in machine learning

- Uncertainty quantification (UQ): a topical problem in machine learning
  - Prediction uncertainty (supervised learning)
  - Cluster-membership uncertainty (clustering)
- Main sources:
  - Randomness (aleatory uncertainty)
  - Lack of knowledge (epistemic uncertainty)
- Theoretical frameworks:
  - Frequentist statistical framework (prediction regions)
  - Bayesian (additive probabilities)
  - Imprecise probabilities (credal sets, lower previsions)
  - Dempster-Shafer theory (random sets, belief functions)
  - Generalised evidence theory (random fuzzy sets, belief functions)

Evidential Machine Learning

- Arguments for (generalised) evidence theory in ML:
  - Allows for the representation of aleatory and epistemic uncertainties
  - Does not require a precise prior as Bayesian inference (but yields the same results as Bayesian inference if a prior is provided)
  - Well-suited for information fusion (multimodal/multisensor classification, classifier/clusterer combination, etc.)
- Evidential Machine Learning is a branch of ML based on (standard or generalised) evidential reasoning. It quantifies uncertainty using belief functions.

Overview of Evidential ML



Distance-based vs. likelihood-based approaches

- In this talk, we focus on supervised learning.
- Two main approaches:
  1. The distance-based approach computes distances between input vectors and learning vectors or prototypes, seen as pieces of evidence.
     - Classification: evidential k-NN rule, evidential neural network
     - Regression: ENNreg model
  2. The likelihood-based approach assumes a parametric statistical model. The relative likelihood function is considered as a possibility distribution, which allows one to construct a predictive random fuzzy set.

Outline

Outline

Distance-based approach for classification

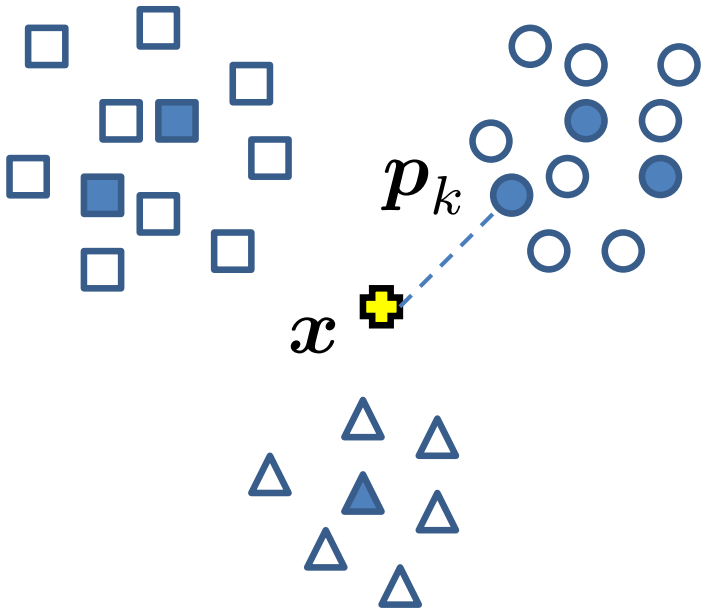
- Two of the first papers applying DS theory to classification:

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

- I will briefly recall the evidential neural network model and describe some recent developments.

Evidential neural network classifier



- The learning set is summarized by K prototypes
- Each prototype $\boldsymbol{p}_k$ has membership degree $u_{k\ell} \geq 0$ to each class ω_ℓ, with $\sum_{\ell=1}^{c} u_{k\ell} = 1$
- Each prototype $\boldsymbol{p}_k$ is a piece of evidence about the class of $\boldsymbol{x}$; its reliability decreases with its distance to $\boldsymbol{x}$

Propagation equations

- Frame: $\Theta = \{\theta_1, \ldots, \theta_c\}$ for c-class classification.
- Similarity between input $\boldsymbol{x}$ and prototype $\boldsymbol{p}_k$:

$$s_k(\boldsymbol{x}) = \exp(-\gamma_k \|\boldsymbol{x} - \boldsymbol{p}_k\|^2), \quad \gamma_k \geq 0$$

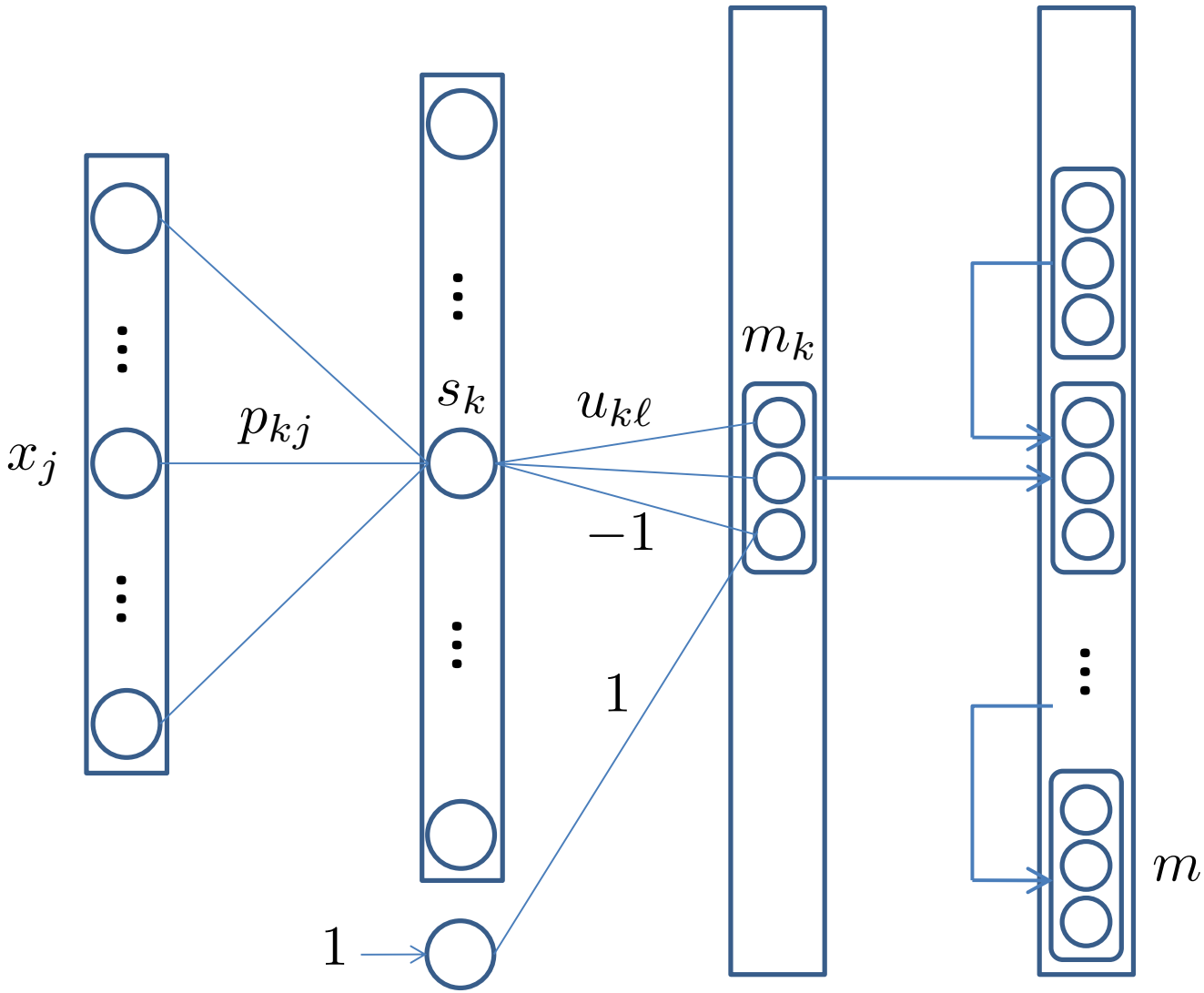
- Mass function induced by prototype $\boldsymbol{p}_k$:

$$m_k(\{\theta_\ell\}) = \alpha_k u_{k\ell} s_k(\boldsymbol{x}), \quad \ell = 1, \ldots, c$$
$$m_k(\Theta) = 1 - \alpha_k s_k(\boldsymbol{x})$$

  with $\alpha_k \in [0, 1]$.
- Property: $\lim_{s_k \to 0} m_k(\Theta) = 1$.
- Combination of the evidence from the K prototypes:

$$m = \bigoplus_{k=1}^{K} m_k$$

Neural network implementation



Learning

- The parameters are the
  - The prototypes $\boldsymbol{p}_k$, $i = k, \ldots, K$ (Kp parameters)
  - The membership degrees $u_{k\ell}$, $k = 1, \ldots, K$, $\ell = 1 \ldots, c$ (Kc parameters)
  - The α_k and γ_k, $k = 1 \ldots, K$ ($2K$ parameters).
- The parameter vector Ψ can be estimated by minimising a loss function such as

$$C_\lambda(\Psi) := \underbrace{\sum_{i=1}^{n}\sum_{\ell=1}^{c}(pl_{i\ell} - y_{i\ell})^2}_{\text{error}} + \lambda \underbrace{\sum_{k=1}^{K}\alpha_k}_{\text{regularization}}$$

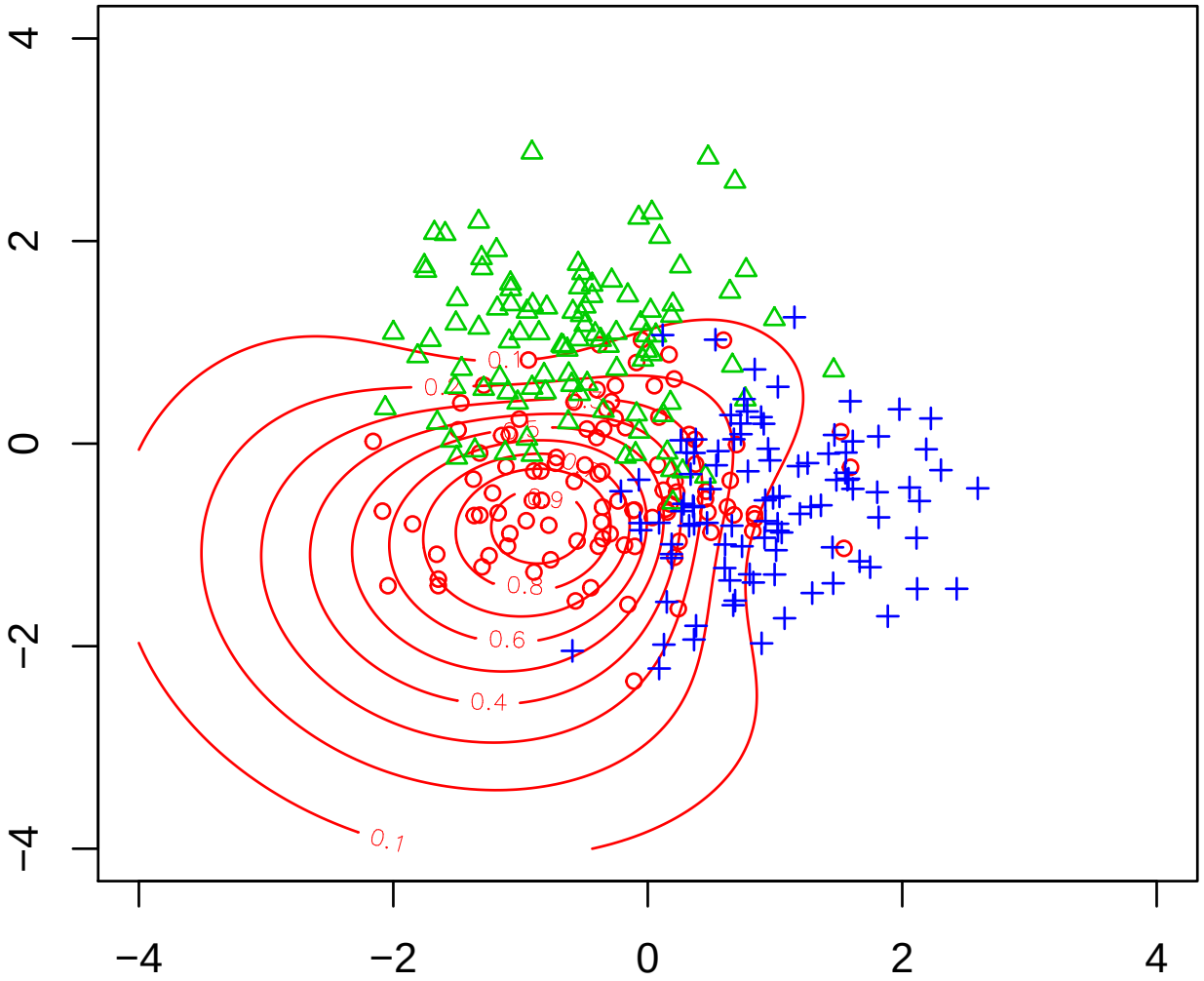
  where $pl_{i\ell}$ is the output plausibility of class θ_ℓ for instance i, $y_{i\ell} = I(y_i = \theta_\ell)$, and λ is a regularisation coefficient (hyperparameter).
- λ can be optimised by cross-validation.

Implementations

- Matlab: `http://www.hds.utc.fr/~tdenoeux/software/belief_NN/belief_NN.zip`
- R package `evclass`, available at `https://cran.r-project.org/web/packages/evclass`

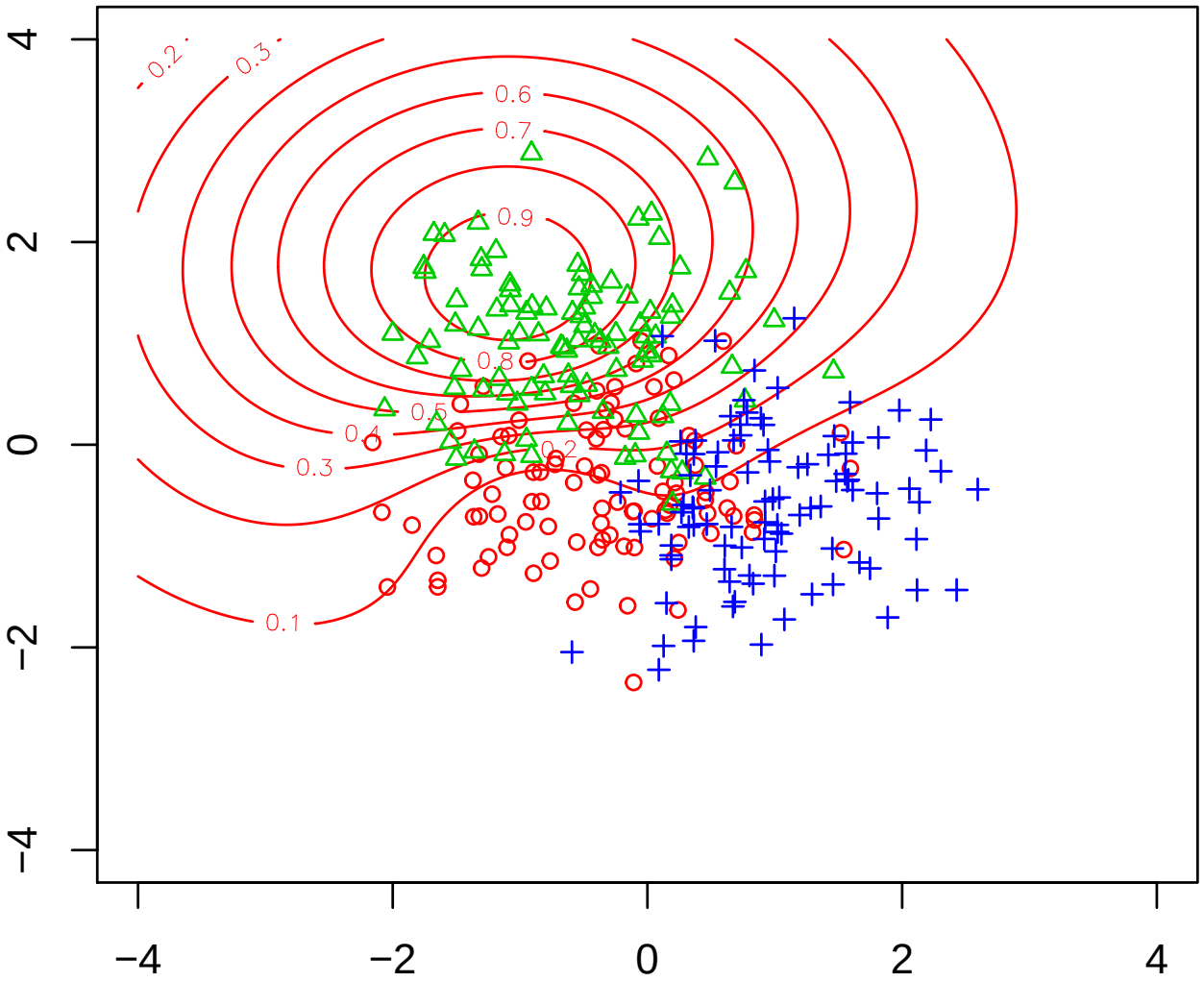
Example

Mass on $\{\theta_1\}$



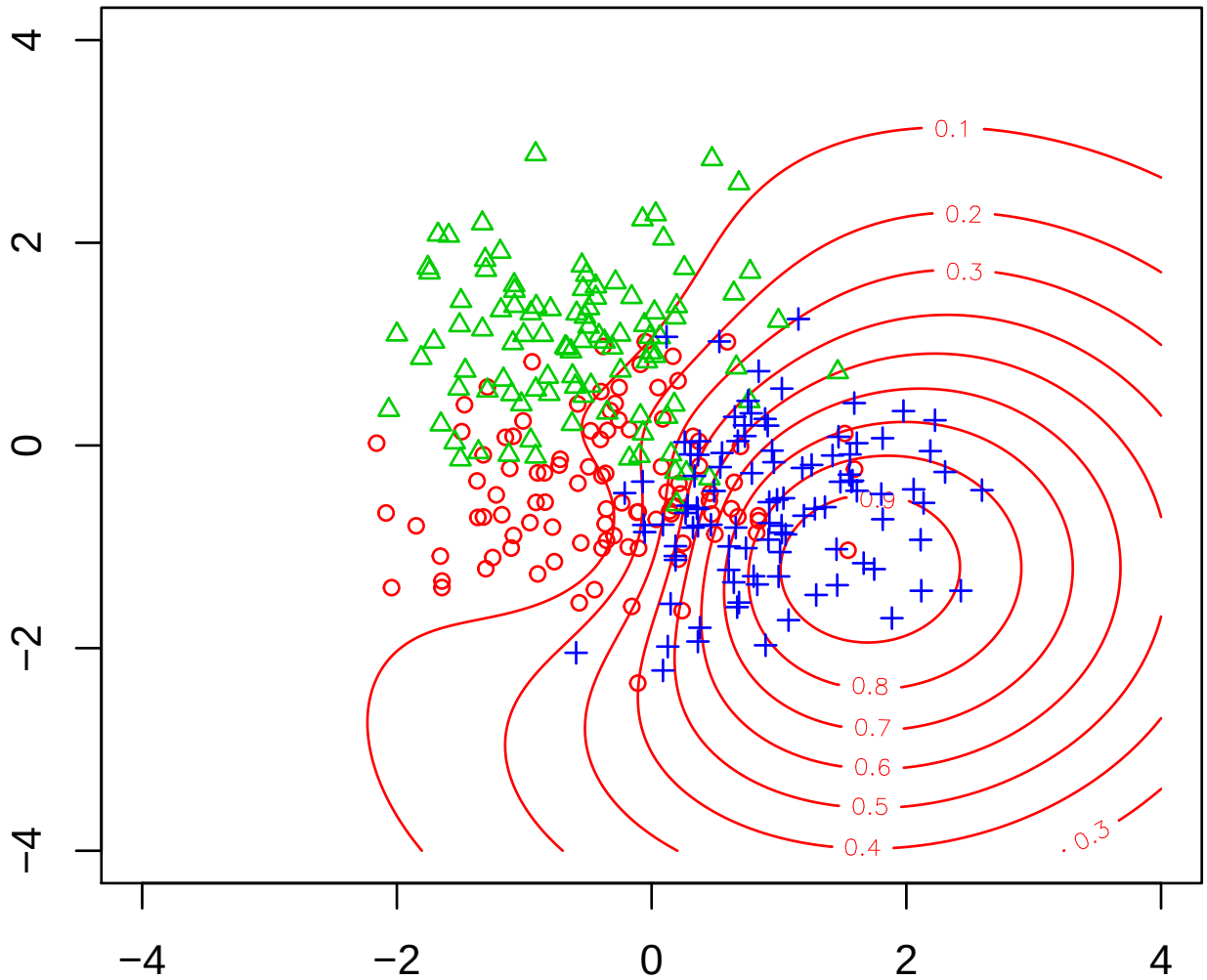
Example

Mass on $\{\theta_2\}$



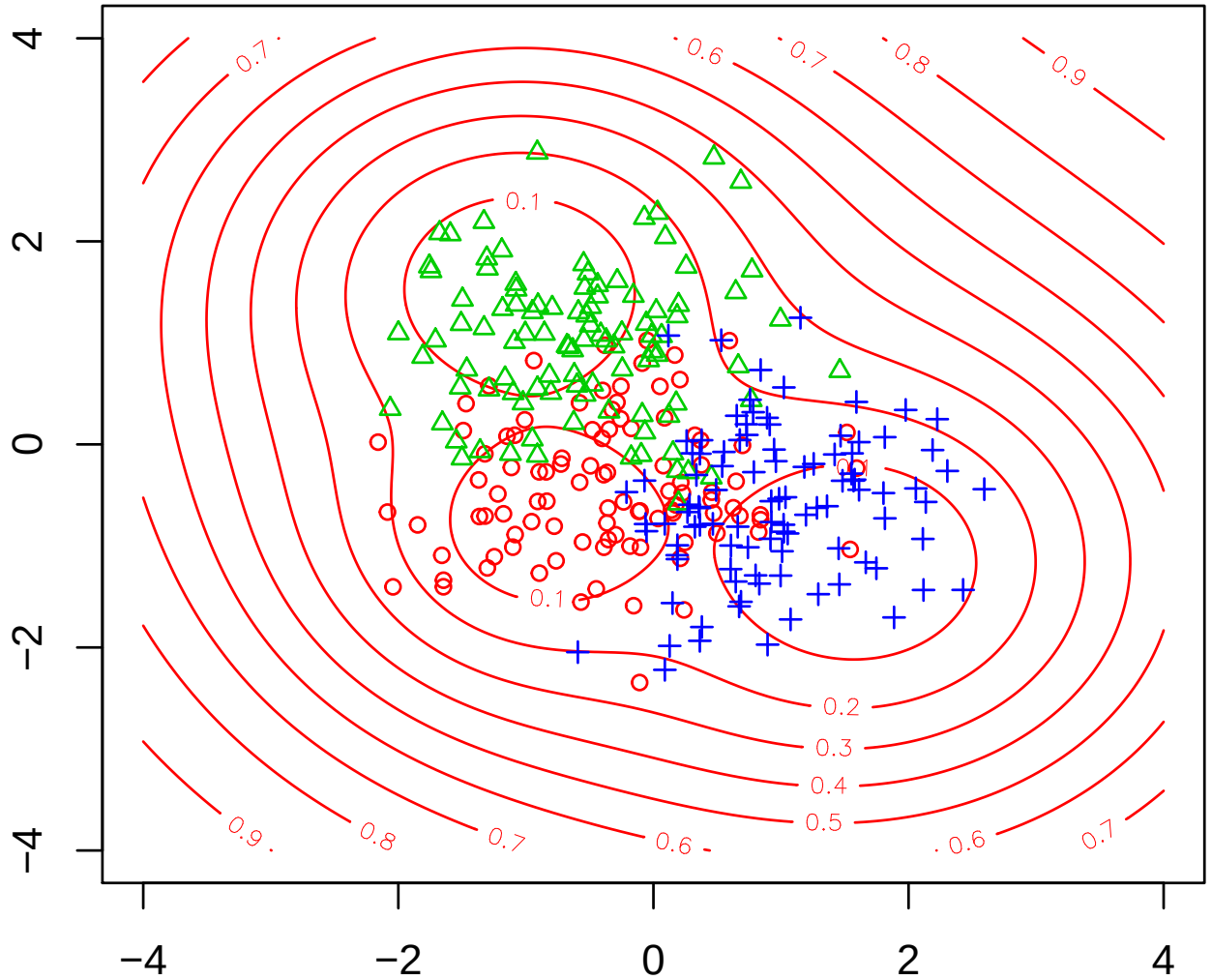
Example

Mass on $\{\theta_3\}$



Example

Mass on Θ



Decision

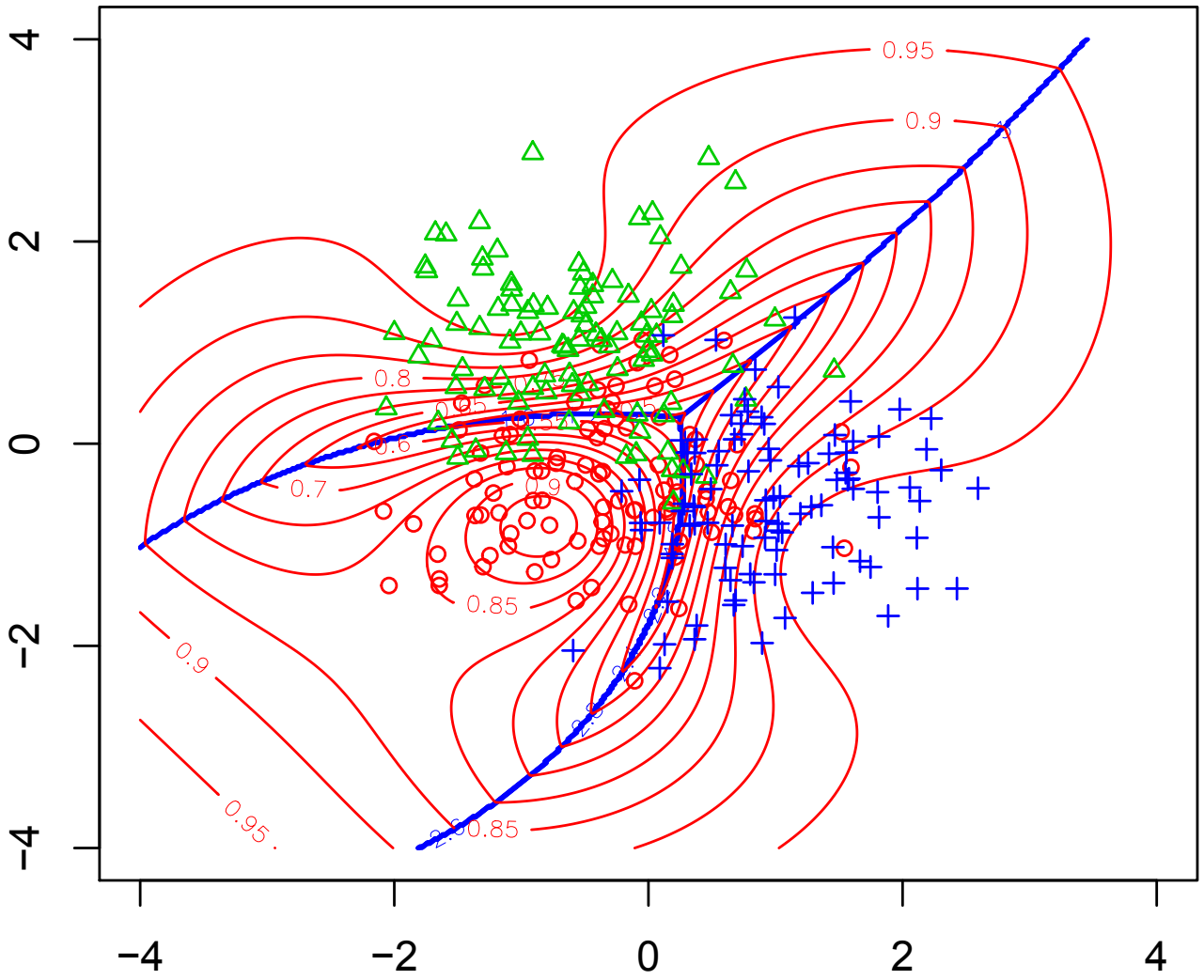
- The focal sets of the combined mass function m are the singletons $\{\theta_k\}$, $k = 1, \ldots, c$ and Θ.
- For decision analysis, we need to define:
  - A set $\mathcal{O}$ of outcomes (correct decision, error, no classification, etc)
  - A set of $\mathcal{G} = \{g_1, \ldots, g_q\}$ of acts, where $g_k : \Theta \rightarrow \mathcal{O}$ (classification in class θ_k, partial classification in set $A \subset \Theta$, rejection, etc.)
  - A utility function $u : \mathcal{O} \rightarrow \mathbb{R}$
- Different decision rules for precise or partial classification can be defined, see

T. Denœux
Decision-Making with Belief Functions: a Review
International Journal of Approximate Reasoning, 109:87–110, 2019

L. Ma and T. Denœux
Partial Classification in the Belief Function Framework
Knowledge-Based Systems, 214:106742, 2021

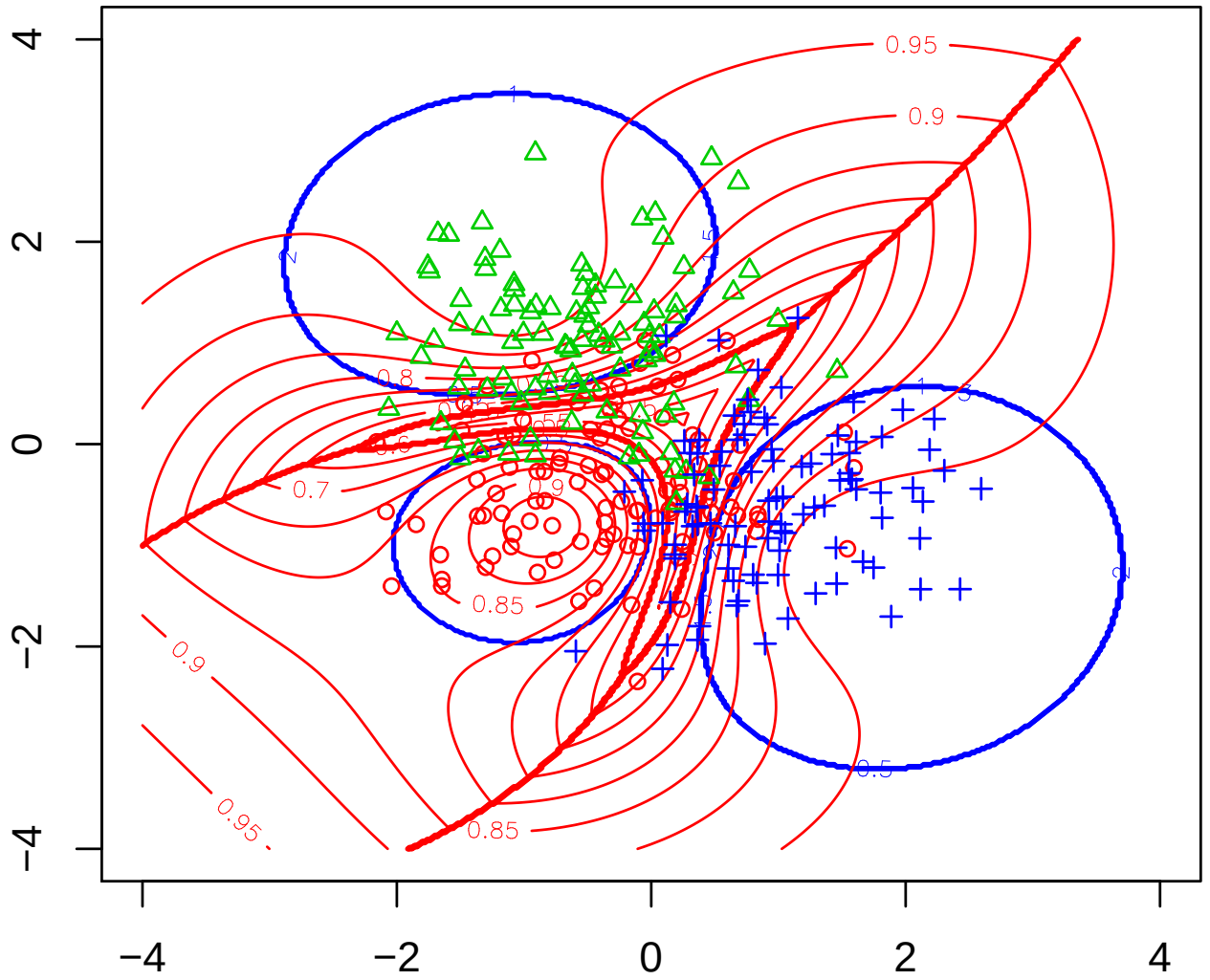
Example

Decision boundaries, optimistic/pessimistic decision rules (no rejection)



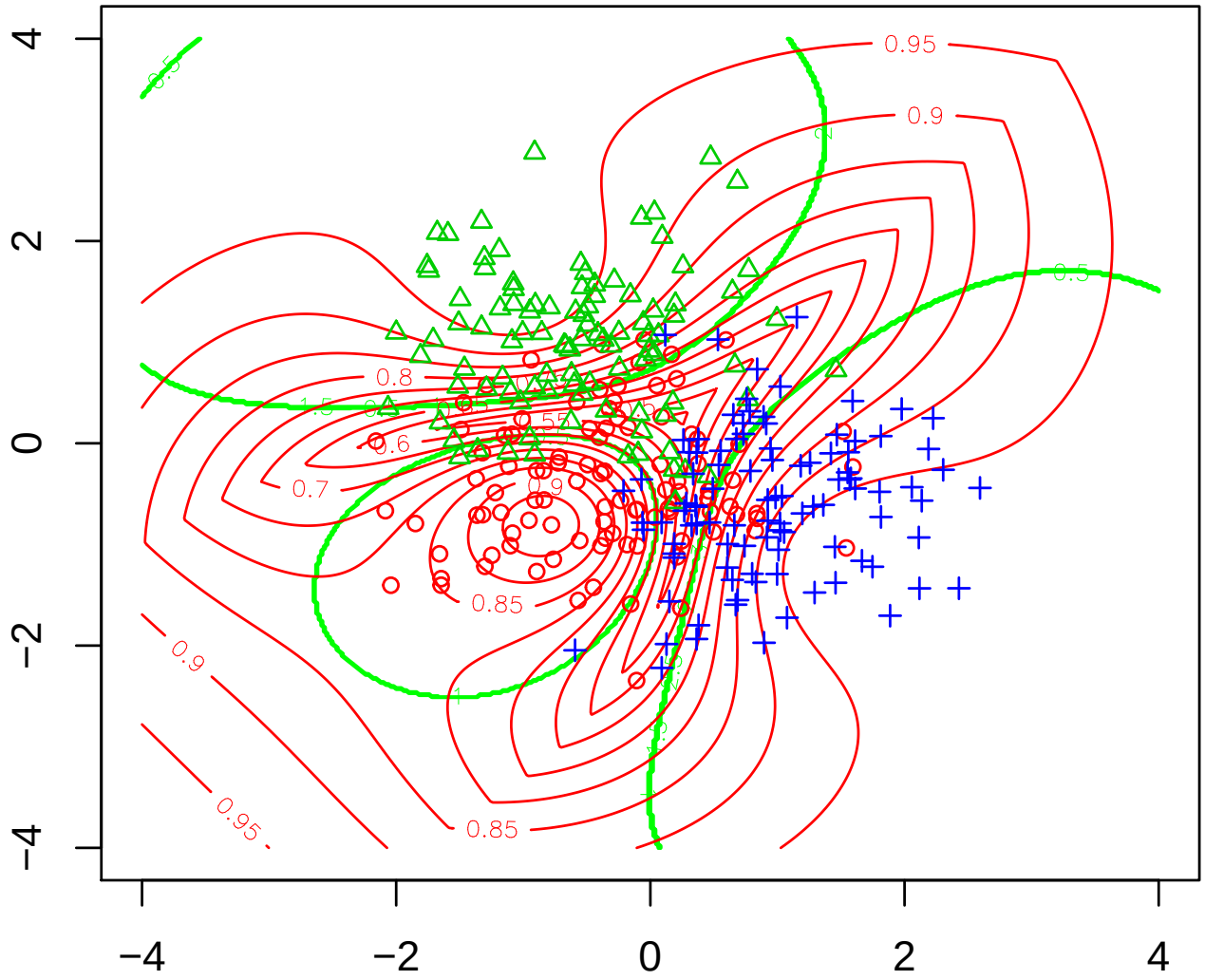
Example

Decision boundaries, optimistic/pessimistic decision rules (with rejection)

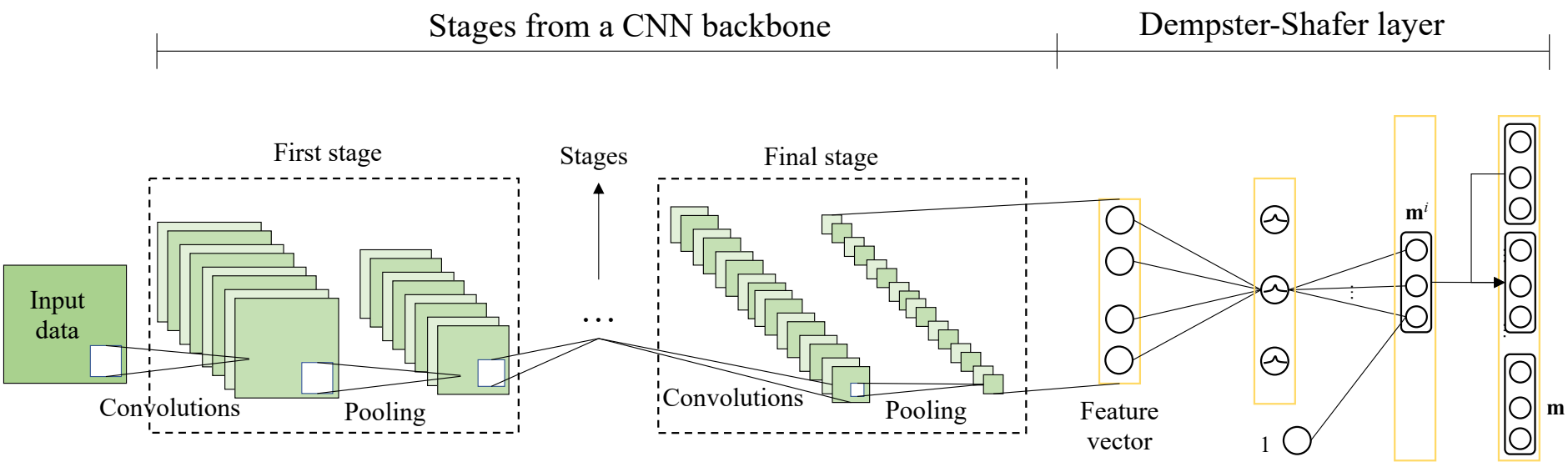


Example

Decision boundaries, Hurwicz criterion, $\rho = 0.5$ (with rejection)

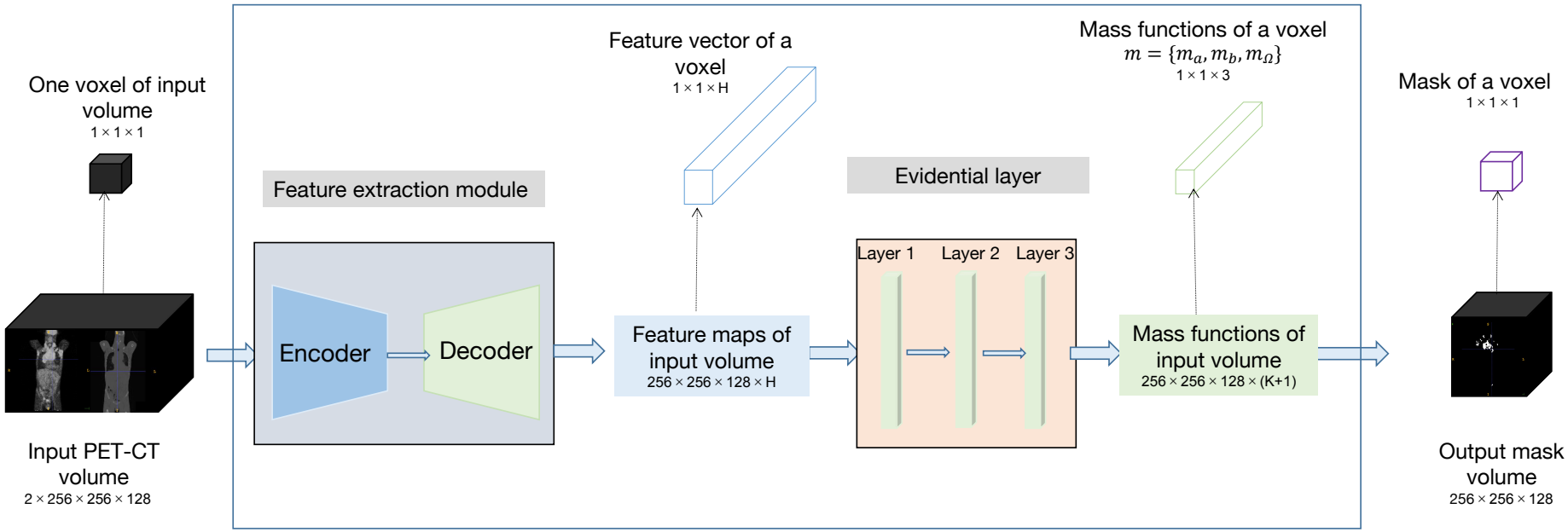


Deep evidential classifier



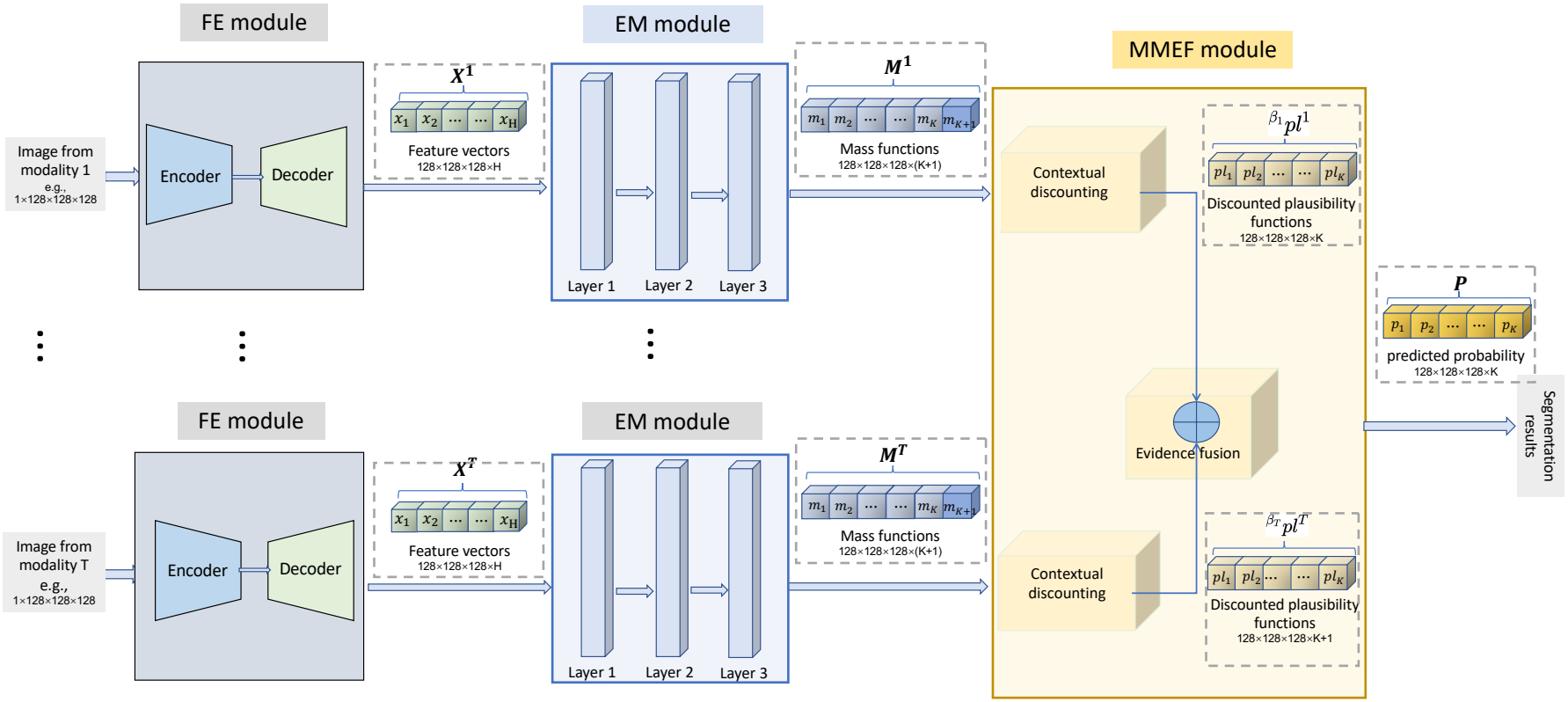
Z. Tong, Ph. Xu and T. Denœux
An evidential classifier based on Dempster-Shafer theory and deep learning
Neurocomputing 450:275–293, 2021

Application to medical image segmentation



L. Huang, S. Ruan, P. Decazes and T. Denœux
Lymphoma segmentation from 3D PET-CT images using a deep evidential network
International Journal of Approximate Reasoning 149:39-60, 2022

Application to multimodal medical image segmentation



L. Huang, S. Ruan, P. Decazes and T. Denœux
Deep evidential fusion with uncertainty quantification and reliability learning for multimodal medical image segmentation
Information Fusion, 113:102648, 2025

Outline

The ENNreg model

- We now consider a regression problem: the task is to predict a continuous random response Y from p input variables $\boldsymbol{X} = (X_1, \ldots, X_p)$, based on a learning set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.
- ENNreg is a neural network model[1] that quantifies uncertainty about the response Y given input vector $\boldsymbol{X} = \boldsymbol{x}$ by a GRFN $\widetilde{Y}(\boldsymbol{x})$ with associated belief function $Bel_{\widetilde{Y}(\boldsymbol{x})}$.
- As the ENN model for classification, ENNreg is based on prototypes. The prototypes are treated as independent pieces of evidence about the response, combined by the product-intersection rule.

[1] T. Denœux. Quantifying Prediction Uncertainty in Regression using Random Fuzzy Sets: the ENNreg model. *IEEE Transactions on Fuzzy Systems*, 31(10):3690–3699, 2023.

Propagation equations (1/2)

- Let $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K$ denote K prototypes in the p-dimensional input space.
- Similarity between input vector $\boldsymbol{x}$ and prototype $\boldsymbol{p}_k$:

$$s_k(\boldsymbol{x}) = \exp(-\gamma_k^2 \|\boldsymbol{x} - \boldsymbol{p}_k\|^2)$$

where $\gamma_k > 0$ is a scale parameter.

- The evidence from prototype $\boldsymbol{p}_k$ is represented by a GRFN

$$\widetilde{Y}_k(\boldsymbol{x}) \sim \widetilde{N}(\mu_k(\boldsymbol{x}), \sigma_k^2, s_k(\boldsymbol{x}) h_k)$$

where σ_k^2 and h_k are variance and precision parameters, and

$$\mu_k(\boldsymbol{x}) = \boldsymbol{\beta}_k^T \boldsymbol{x} + \beta_{k0} \quad \text{with} \quad \boldsymbol{\beta}_k \in \mathbb{R}^p, \beta_{k0} \in \mathbb{R}$$

Propagation equations (2/2)

- The output $\widetilde{Y}(\boldsymbol{x})$ for input $\boldsymbol{x}$ is computed as

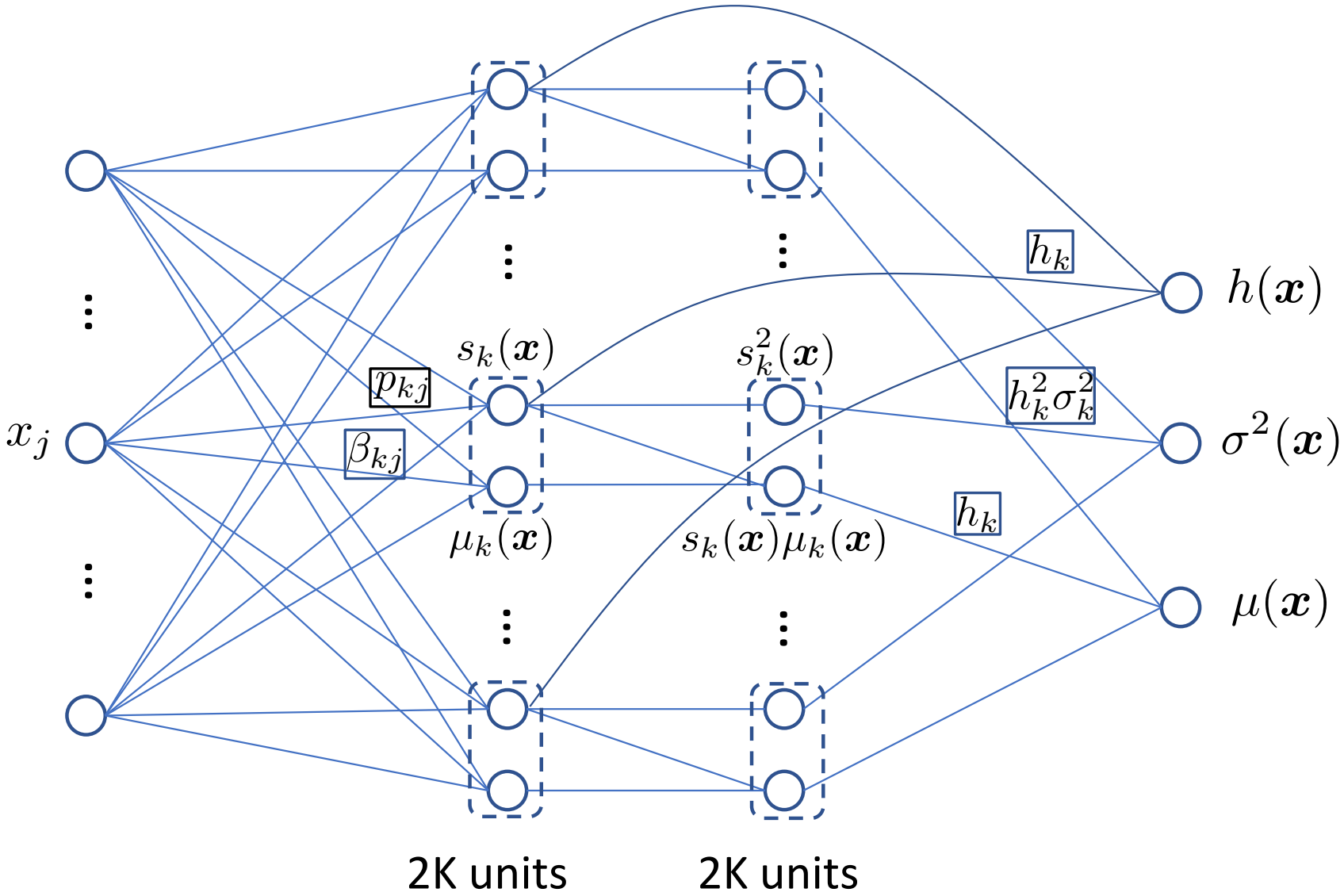
$$\widetilde{Y}(\boldsymbol{x}) = \widetilde{Y}_1(\boldsymbol{x}) \widetilde{\oplus} \ldots \widetilde{\oplus} \widetilde{Y}_K(\boldsymbol{x})$$

  where $\widetilde{\oplus}$ is an approximation of the product-intersection rule (assuming $\widetilde{\boldsymbol{\mu}} \approx \boldsymbol{\mu}$ and $\widetilde{\boldsymbol{\Sigma}} \approx \boldsymbol{\Sigma}$).
- We have $\widetilde{Y}(\boldsymbol{x}) \sim \widetilde{N}\left(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}), h(\boldsymbol{x})\right)$, with

$$\mu(\boldsymbol{x}) = \frac{\sum_{k=1}^{K} s_k(\boldsymbol{x}) h_k \mu_k(\boldsymbol{x})}{\sum_{k=1}^{K} s_k(\boldsymbol{x}) h_k}$$

$$\sigma^2(\boldsymbol{x}) = \frac{\sum_{k=1}^{K} s_k^2(\boldsymbol{x}) h_k^2 \sigma_k^2}{\left(\sum_{k=1}^{K} s_k(\boldsymbol{x}) h_k\right)^2} \quad \text{and} \quad h(\boldsymbol{x}) = \sum_{k=1}^{K} s_k(\boldsymbol{x}) h_k$$

Neural network architecture



Negative log-likelihood loss (probabilistic forecasts)

- In the case of a probabilistic forecast with pdf $\widehat{f}$, we typically measure the prediction error (or loss) by the negative log-likelihood

$$\mathcal{L}(y, \widehat{f}) = -\ln \widehat{f}(y)$$

- We actually never observe a real number y with infinite precision, but an interval $[y]_\epsilon = [y - \epsilon, y + \epsilon]$ centered at y. The probability of that interval is

$$\widehat{P}([y]_\epsilon) = \widehat{F}(y + \epsilon) - \widehat{F}(y - \epsilon) \approx 2\widehat{f}(y)\epsilon$$

  So, $\mathcal{L}(y, \widehat{f}) = -\ln \widehat{P}([y]_\epsilon) + \text{cst}$.
- Generalisation to the case of prediction in the form of a belief function?

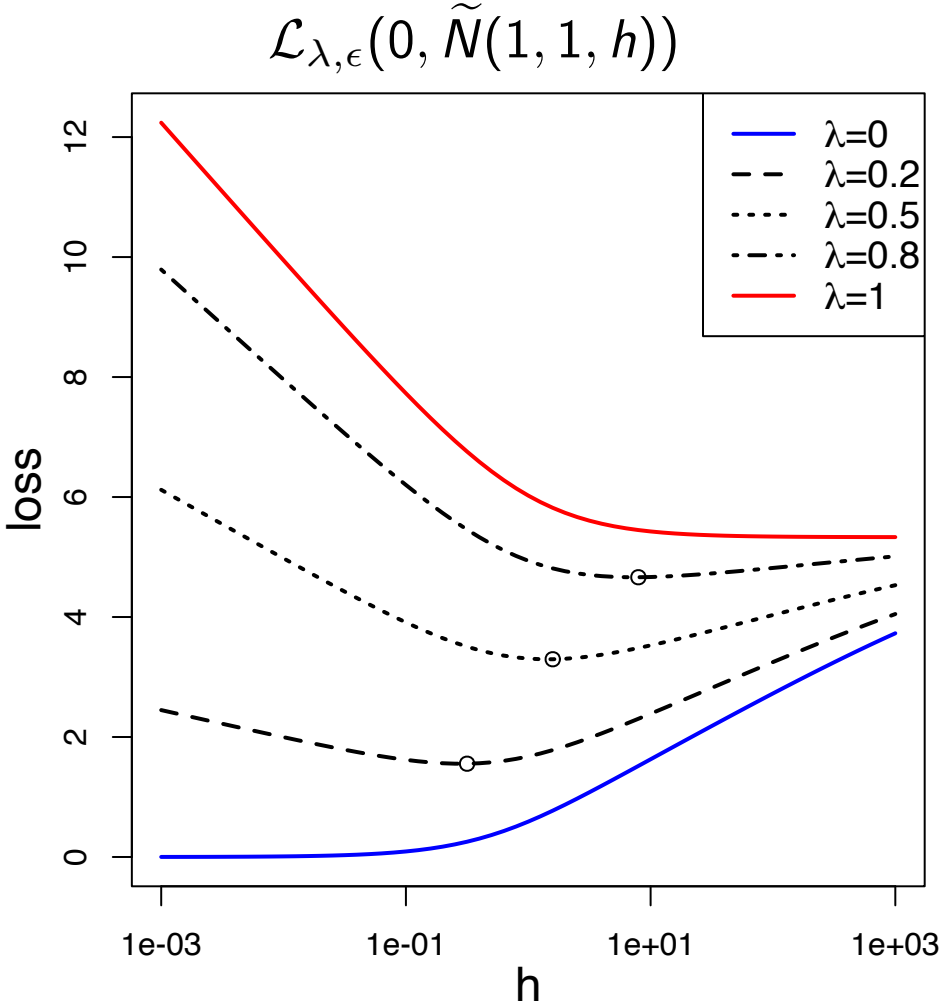
Extension

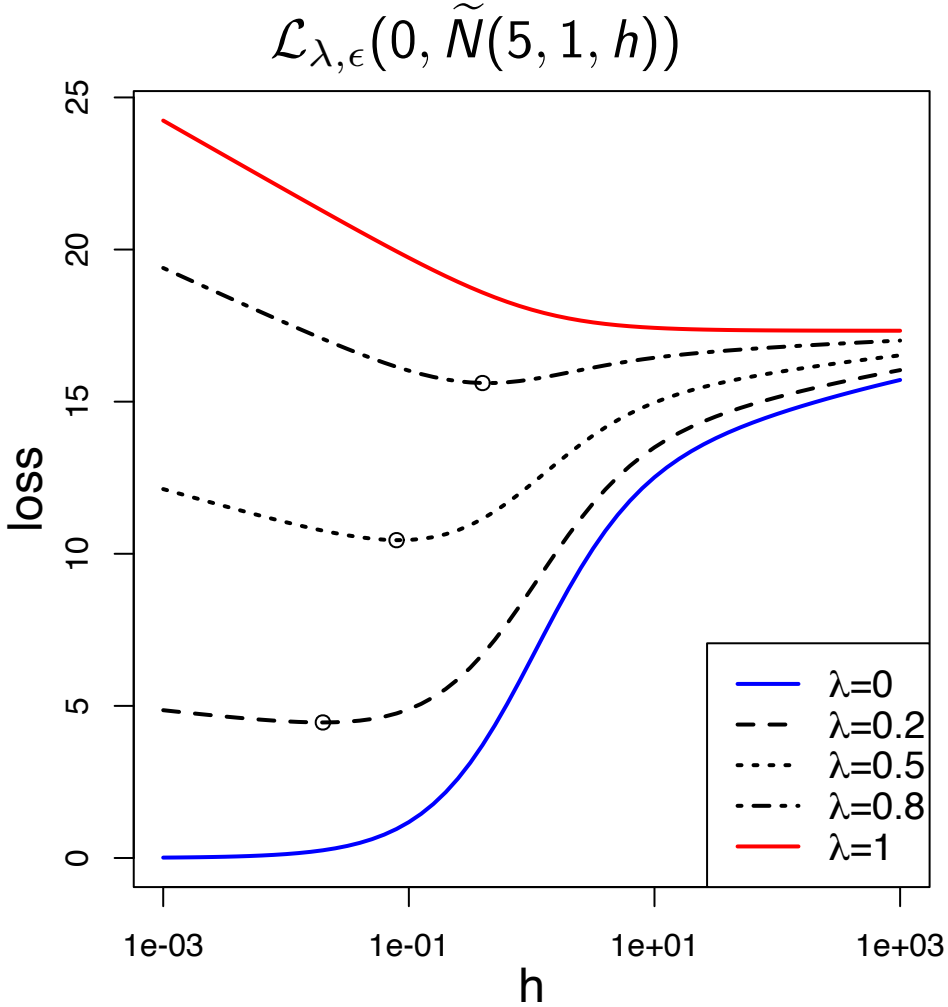
- $\mathcal{L}_\epsilon(y, \widetilde{Y}) = -\ln Bel_{\widetilde{Y}}([y]_\epsilon)$ does not work (does not reward imprecision).
- $\mathcal{L}_\epsilon(y, \widetilde{Y}) = -\ln Pl_{\widetilde{Y}}([y]_\epsilon)$ also does not work (minimized when $\widetilde{Y}$ is vacuous).
- Proposal:

$$\mathcal{L}_{\lambda,\epsilon}(y, \widetilde{Y}) = -\lambda \ln Bel_{\widetilde{Y}}([y]_\epsilon) - (1-\lambda) \ln Pl_{\widetilde{Y}}([y]_\epsilon)$$

  with $\lambda \in [0, 1]$ and $\epsilon > 0$.
- Smaller values of λ correspond to more cautious predictions.

Influence of λ





Training

- The network is trained by minimising the regularised average loss

$$C^{(R)}_{\lambda,\epsilon,\xi,\rho}(\Psi) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\lambda,\epsilon}(y_i, \widetilde{Y}(\boldsymbol{x}_i;\Psi))}_{C_{\lambda,\epsilon}(\Psi)} + \underbrace{\frac{\xi}{K}\sum_{k=1}^{K} h_k}_{R_1(\Psi)} + \underbrace{\frac{\rho}{K}\sum_{k=1}^{K} \gamma_k^2}_{R_2(\Psi)}$$

  where
  - $R_1(\Psi)$ has the effect of reducing the number of prototypes used for the prediction (setting $h_k = 0$ amounts to discarding prototype k)
  - $R_2(\Psi)$ shrinks the solution towards a linear model (setting $\gamma_k = 0$ for all k yields a linear model).
- Heuristics: $\lambda = 0.9$, $\epsilon = 0.01\widehat{\sigma}_Y$, ξ and ρ tuned using a validation set or cross-validation.

Calibration

- For any $\alpha \in (0, 1]$, we define an α-level belief prediction interval (BPI) as an interval $\mathcal{B}_\alpha(\boldsymbol{x})$ centered at $\mu(\boldsymbol{x})$, such that $Bel_{\widetilde{Y}(\boldsymbol{x})}(\mathcal{B}_\alpha(\boldsymbol{x})) = \alpha$.
- The predictions are said to be (weakly) calibrated if, for all $\alpha \in (0, 1]$, α-level BPIs have a coverage probability at least equal to α, i.e,

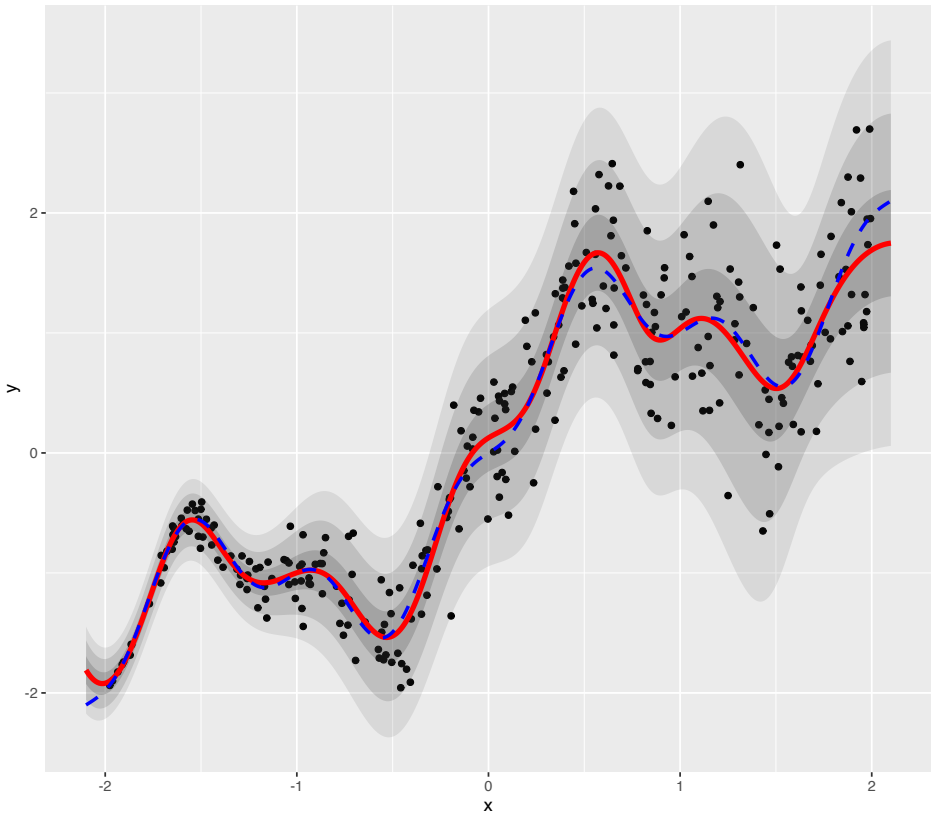
$$\forall \alpha \in (0, 1], \quad P_{\boldsymbol{X},Y}\left(Y \in \mathcal{B}_\alpha(\boldsymbol{X})\right) \geq \alpha \tag{1}$$

- As in the probabilistic case, the calibration of evidential predictions can be checked graphically using a calibration plot (see infra).
- The precision output $h(\boldsymbol{x})$ can be multiplied by a constant $c > 0$ to ensure (1) with predictions as precise as possible.

Example

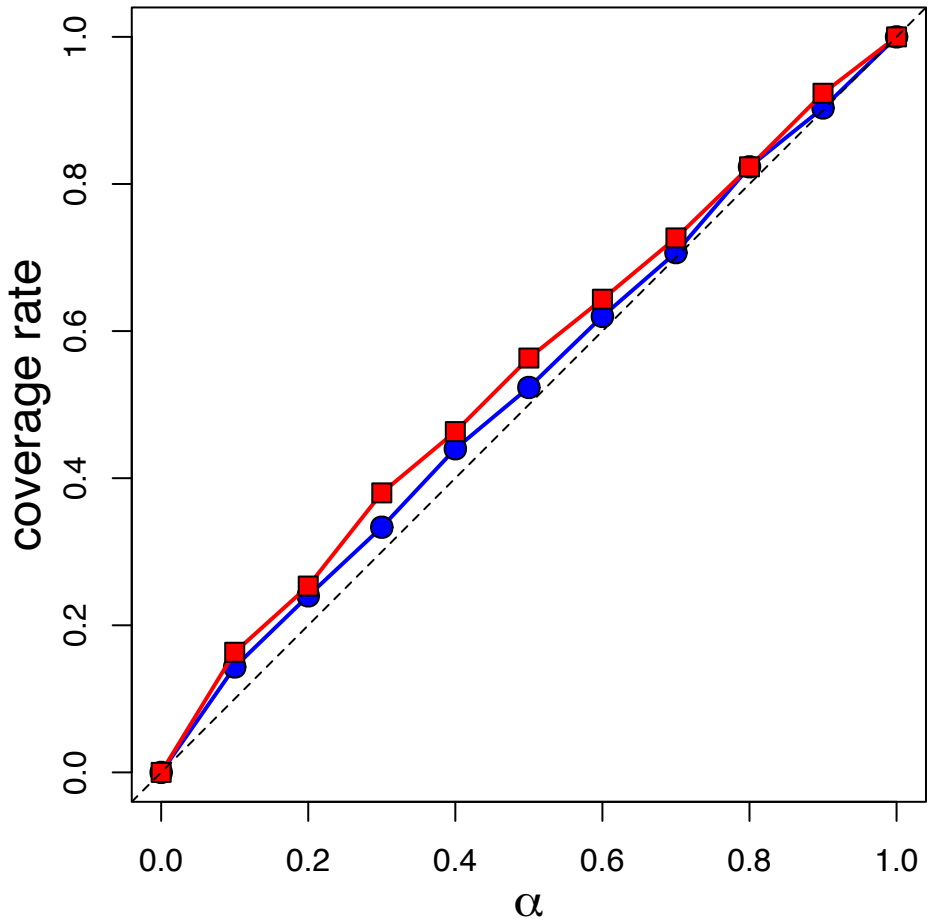
We consider iid data with one-dimensional input $X \sim \text{Unif}(-2, 2)$ and

$$Y = X + (\sin 3X)^3 + \frac{X+2}{4\sqrt{2}} U, \quad U \sim N(0,1)$$



- Learning and validation sets of size $n = 300$.
- Network with $K = 30$ prototypes initialized by the k-means algorithm.
- ξ and ρ determined by minimizing the validation MSE.
- Shown: true regression function (blue), expected values $\mu(x)$ (red) with BPIs at levels 0.5, 0.9 and 0.99

Calibration curves



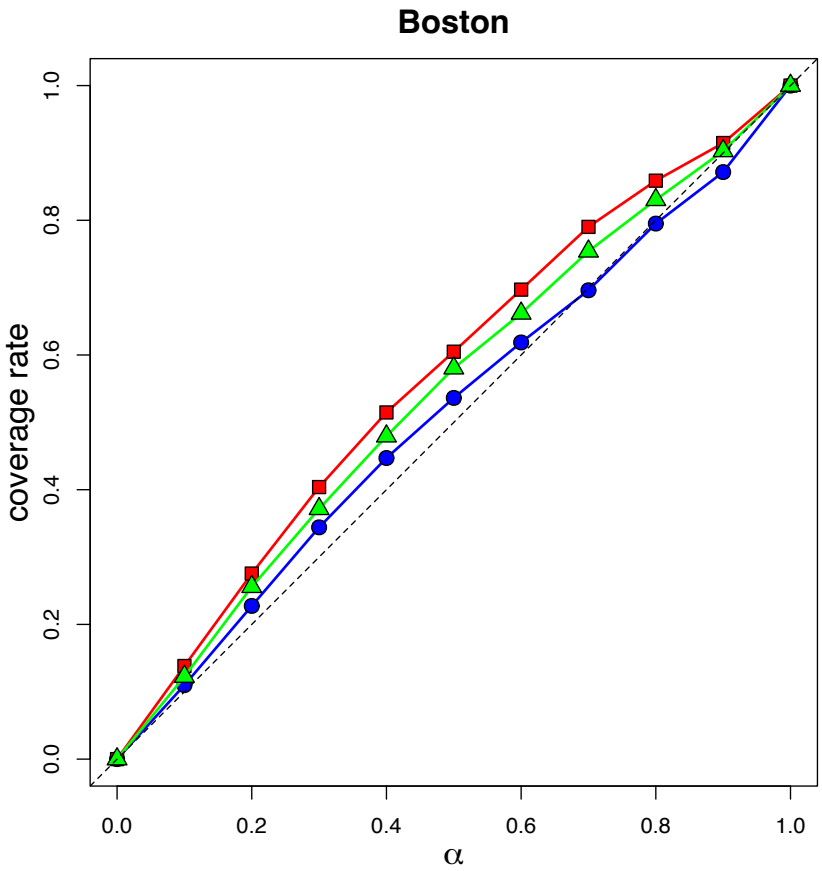
Calibration curves for the probabilistic PIs $\mu(x) \pm u_{(1+\alpha)/2}\sigma(x)$ (in blue) and the BPIs (in red)

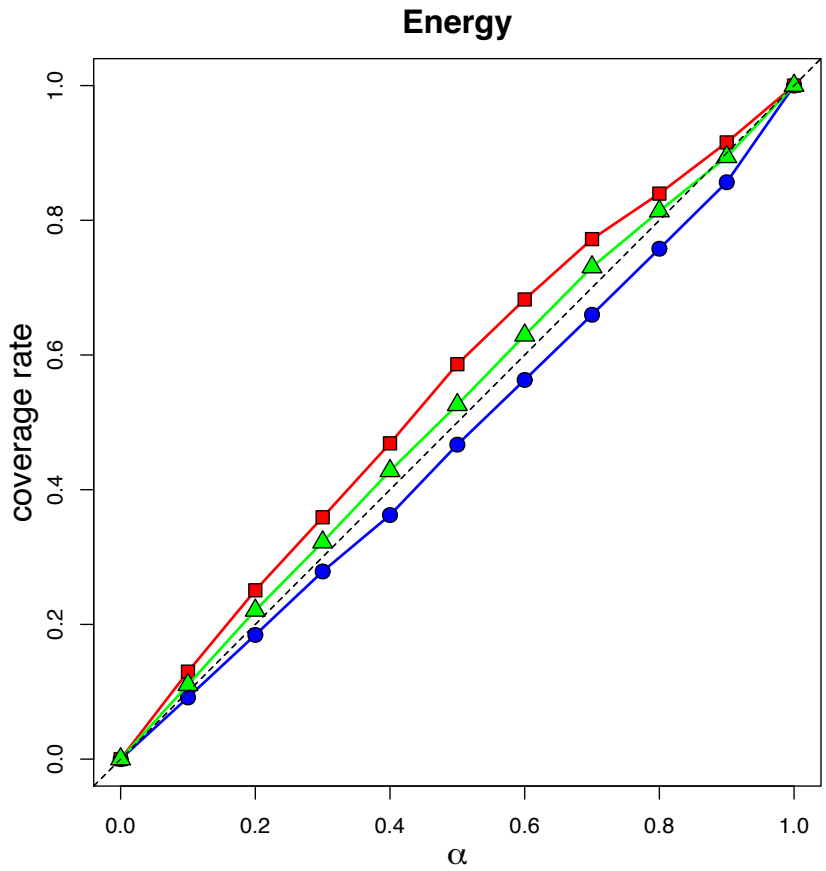
Comparison with SOTA methods (RMS & NLL)

	RMS				
	ENNreg	PBP	MC-dropout	Deep ens.	Deep ev. reg.
Boston	**2.87 ± 0.14**	**3.01 ± 0.18**	**2.97 ± 0.19**	**3.28 ± 1.00**	**3.06 ± 0.16**
Energy	**1.06 ± 0.05**	1.80 ± 0.05	1.66 ± 0.04	2.09 ± 0.29	2.06 ± 0.10
Concr.	**5.10 ± 0.12**	5.67 ± 0.09	**5.23 ± 0.12**	6.03 ± 0.58	5.85 ± 0.15
Yacht	**0.44 ± 0.04**	1.02 ± 0.05	1.11 ± 0.09	1.58 ± 0.48	1.57 ± 0.56
Wine	**0.63 ± 0.01**	**0.64 ± 0.01**	**0.62 ± 0.01**	**0.64 ± 0.04**	**0.61 ± 0.02**
kin8nm	**0.08 ± 0.00**	0.10 ± 0.00	0.10 ± 0.00	0.09 ± 0.00	0.09 ± 0.00

	NLL				
	ENNreg	PBP	MC-dropout	Deep ens.	Deep ev. reg.
Boston	2.53 ± 0.07	2.57 ± 0.09	**2.46 ± 0.06**	**2.41 ± 0.25**	**2.35 ± 0.06**
Energy	**1.14 ± 0.07**	2.04 ± 0.02	1.99 ± 0.02	**1.38 ± 0.22**	1.39 ± 0.06
Concr.	3.38 ± 0.13	3.16 ± 0.02	**3.04 ± 0.02**	**3.06 ± 0.18**	**3.01 ± 0.02**
Yacht	**0.13 ± 0.12**	1.63 ± 0.02	1.55 ± 0.03	1.18 ± 0.21	1.03 ±0.19
Wine	**0.94 ± 0.01**	0.97 ± 0.01	**0.93 ± 0.01**	**0.94 ± 0.12**	**0.89 ± 0.05**
kin8nm	**-1.19 ± 0.00**	-0.90 ± 0.01	-0.95 ± 0.01	**-1.20 ± 0.02**	-1.24 ± 0.01

Calibration plots





Probabilistic predictions (blue), raw evidential predictions (red) and adjusted evidential predictions (green).

Outline

Basic ideas

- We now assume the existence of a parametric statistical model.
- The likelihood function defines a possibility distribution in the parameter space, after observing the data.
- Combination of this possibility distribution with random errors yields a random fuzzy set, which defines a predictive belief function.
- This approach will be described in general, and then applied to regression multi-layer neural networks.

Outline

Relative likelihood

- We consider an observed random vector $\boldsymbol{Y}$ with probability density function (pdf) $f_{\boldsymbol{Y}|\boldsymbol{\theta}}$, where $\boldsymbol{\theta} \in \Theta$ is the unknown parameter.
- The likelihood function after observing $\boldsymbol{Y} = \boldsymbol{y}$ is the map

$$\begin{array}{rccc} L(\cdot;\boldsymbol{y}): & \Theta & \to & \mathbb{R}_+ \\ & \theta & \mapsto & \alpha\, f_{\boldsymbol{Y}|\theta}(\boldsymbol{y}) \end{array}$$

  where α is an arbitrary positive constant.
- Assuming that $\sup_\theta L(\theta;\boldsymbol{y}) < +\infty$, we can rescale the likelihood function in $[0,1]$ and define the relative likelihood function as

$$\pi_{\boldsymbol{\theta}|\boldsymbol{y}} : \theta \mapsto \frac{L(\theta;\boldsymbol{y})}{\sup_{\theta' \in \Theta} L(\theta';\boldsymbol{y})} = \frac{L(\theta;\boldsymbol{y})}{L(\widehat{\theta};\boldsymbol{y})}$$

  where $\widehat{\theta}$ is a maximum likelihood estimate (MLE) of θ.

Possibilistic interpretation

- As remarked by several authors, we can interpret mapping $\pi_{\theta|\boldsymbol{y}} : \Theta \to [0,1]$ as a possibility distribution over Θ or, equivalently, as the fuzzy set of likely values of $\boldsymbol{\theta}$ after observing $\boldsymbol{Y} = \boldsymbol{y}$.
- Why does it make sense?
  - Marginalisation: if $\boldsymbol{\theta} = (\boldsymbol{\psi}, \boldsymbol{\eta})$, where $\boldsymbol{\eta}$ is a nuisance parameter, the profile relative likelihood is defined as a function of $\boldsymbol{\psi}$ as

$$\pi_{\boldsymbol{\psi}|\boldsymbol{y}}(\psi) = \sup_{\eta} \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\psi, \eta)$$

    which is identical to marginalisation of a possibility distribution.
  - Combination: given two independent observations $\boldsymbol{y}$ and $\boldsymbol{y}'$,

$$\pi_{\boldsymbol{\theta}|(\boldsymbol{y},\boldsymbol{y}')} = \frac{\pi_{\boldsymbol{\theta}|\boldsymbol{y}} \cdot \pi_{\boldsymbol{\theta}|\boldsymbol{y}'}}{\sup_\theta \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta) \cdot \pi_{\boldsymbol{\theta}|\boldsymbol{y}'}(\theta)} = \pi_{\boldsymbol{\theta}|\boldsymbol{y}} \cap_p^* \pi_{\boldsymbol{\theta}|\boldsymbol{y}'}$$

    where $\cap_p^*$ is the normalised product intersection operation of possibility theory.

Possibilistic interpretation (continued)

- If we accept the possibilistic interpretation of the relative likelihood, the degrees of possibility and necessity of any hypothesis $H \subseteq \Theta$, after observing $\boldsymbol{y}$ can be computed, respectively, as

$$\Pi_{\boldsymbol{\theta}|\boldsymbol{y}}(H) = \sup_{\theta \in H} \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta)$$

and

$$N_{\boldsymbol{\theta}|\boldsymbol{y}}(H) = 1 - \Pi_{\pi_{\boldsymbol{\theta}|\boldsymbol{y}}}(H^c)$$

- Under regularity conditions, the large sample distribution of $-2 \ln \Pi_{\boldsymbol{\theta}|\boldsymbol{y}}(H)$, when H holds, is chi-squared, with df equal to the number r of restrictions imposed by H. Consequently, if H is true,

$$P\left(\Pi_{\boldsymbol{\theta}|\boldsymbol{y}}(H) \leq \exp(-\chi^2_{r;1-\alpha}/2)\right) \approx \alpha$$

i.e., a true hypothesis rarely has a small plausibility (for a large enough sample).

Normal approximation

- Assuming $\ln \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta)$ to be twice differentiable, we can approximate it by its 2nd-order Taylor expansion about $\widehat{\theta}$ as

$$\ln \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta) = \ln \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\widehat{\theta}) + (\theta - \widehat{\theta})^T \left.\frac{\partial \ln \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta)}{\partial \theta}\right|_{\theta=\widehat{\theta}} + \frac{1}{2}(\theta - \widehat{\theta})^T \underbrace{\left.\frac{\partial^2 \ln \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta)}{\partial \theta \partial \theta^T}\right|_{\theta=\widehat{\theta}}}_{\text{observed information } \mathcal{I}(\widehat{\theta})} (\theta - \widehat{\theta}) + \cdots$$

  which gives us the following normal approximation:

$$\pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta) \approx \exp\left[-\frac{1}{2}(\theta - \widehat{\theta})^T \mathcal{I}(\widehat{\theta})(\theta - \widehat{\theta})\right]$$

- Thus, $\pi_{\boldsymbol{\theta}|\boldsymbol{y}} \approx \text{GFV}\left(\widehat{\theta}, \mathcal{I}(\widehat{\theta})\right)$.

Prior information

- Assume that we have a Bayesian prior distribution P_0 on Θ.
- Combining P_0 with the $\pi_{\boldsymbol{\theta}|\boldsymbol{y}}$ using the product-intersection rule of GET yields the posterior probability distribution:

$$P_0 \oplus \pi_{\boldsymbol{\theta}|\boldsymbol{y}} = P_{\boldsymbol{\theta}|\boldsymbol{y}}$$

- Evidential likelihood-based inference is, thus, generalised Bayesian inference: it does not require a probabilistic prior, but yields the same result at Bayesian inference is such a prior is provided.
- Prior knowledge may often be more conveniently expressed as a possibility distribution π_0. In this case, $\pi_0 \oplus \pi_{\boldsymbol{\theta}|\boldsymbol{y}}$ is still a possibility distribution.

Prediction

- We now address the prediction problem: how to predict/forecast the value of a r.v. Y_0 with sample space $\mathcal{Y}$, whose distribution depends on $\boldsymbol{\theta}$?
- Basic approach:
  1. Express Y_0 as a function φ of $\boldsymbol{\theta}$ and a pivotal r.v. U with known probability distribution P_U.
  2. Combine $\pi_{\boldsymbol{\theta}|\boldsymbol{y}}$, φ and P_U and marginalise on Y_0. The result is a random fuzzy set.

φ-equation

- We can always write Y_0 as

$$Y_0 = \varphi(\boldsymbol{\theta}, U)$$

where

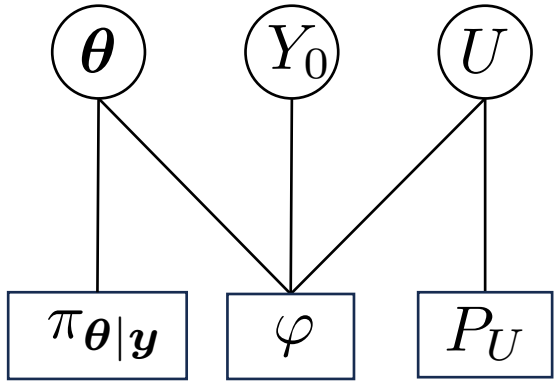
  - U is a pivotal r.v. with known distribution and sample space $\mathcal{U}$
  - φ is a mapping from $\Theta \times \mathcal{U}$ to $\mathcal{Y}$.

- Example: if Y_0 is a real r.v. with c.d.f. $F(y; \boldsymbol{\theta})$, we can write

$$Y_0 = F^{-1}(U; \boldsymbol{\theta}) \quad \text{with} \quad U \sim \text{Unif}([0, 1])$$

where F^{-1} is the (generalised) inverse of F.

Predictive random fuzzy set



- Combining $\pi_{\boldsymbol{\theta}|\boldsymbol{y}}$, φ and P_U, and marginalising on Y_0, we obtain a random fuzzy set

$$\begin{array}{rccc} \widetilde{Y}: & \mathcal{U} & \to & \mathcal{F}(\mathcal{Y}) \\ & U & \mapsto & \pi_{Y_0|\boldsymbol{y},U} \end{array}$$

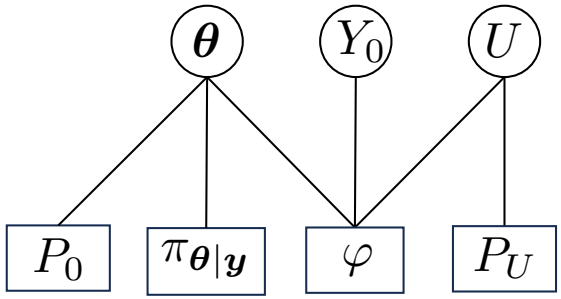
with

$$\pi_{Y_0|\boldsymbol{y},U}(y) = \sup_{\{\theta\in\Theta:\varphi(\theta,U)=y\}} \pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\theta)$$

- $\widetilde{Y}$ induces a predictive belief function $Bel_{\widetilde{Y}}$.

Special cases

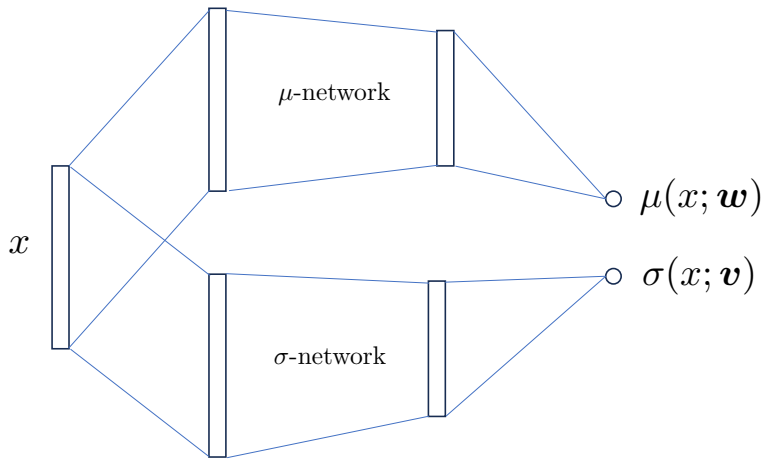
1. Perfect knowledge of $\boldsymbol{\theta}$: if $\pi_{\boldsymbol{\theta}|\boldsymbol{y}} = \mathbb{1}_{\{\theta_0\}}$, then $Bel_{\widetilde{Y}} = P_{Y_0}$.
2. Bayesian prior:



If $\pi_{\boldsymbol{\theta}|\boldsymbol{y}}$ is combined with a Bayesian prior P_0, then $Bel_{\widetilde{Y}} = P_{Y_0|\boldsymbol{y}}$, the Bayesian predictive probability measure.

Outline

Model



We consider the following model:

$$Y = \mu(x; \boldsymbol{w}) + \sigma(x; \boldsymbol{v})U$$

where

- $U \sim N(0, 1)$ is a random noise with standard normal distribution
- $\mu(x; \boldsymbol{w})$, $\sigma(x; \boldsymbol{v})$ are the outputs of two neural networks with distinct weight vectors $\boldsymbol{w}$ and $\boldsymbol{v}$, referred to, respectively, as the μ and σ-networks

Log-likelihood

- Let $\boldsymbol{\theta} = (\boldsymbol{w}^T, \boldsymbol{v})^T$ denote the vector of all parameters in the model. Given i.i.d. data $\{(x_i, y_i)\}_{i=1}^n$, the log-likelihood of $\boldsymbol{\theta}$ is

$$\ell(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \ln \sigma(x_i, \boldsymbol{v}) - \frac{1}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{n}\left(\frac{y_i - \mu(x_i; \boldsymbol{w})}{\sigma(x_i, \boldsymbol{v})}\right)^2$$

- We fit the model by maximising the regularised log-likelihood

$$\ell^{(R)}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}) - \sum_{j=1}^{N_w} \lambda_j w_j^2 - \sum_{j=1}^{N_v} \xi_j v_j^2$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{N_w})$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{N_v})$ are vectors of regularisation coefficients.

Relative log-likelihood

- Let $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{w}}^T, \widehat{\boldsymbol{v}})^T$ be a global maximiser of $\ell^{(R)}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\boldsymbol{\theta})$. We define the possibility distribution of $\boldsymbol{\theta}$ as

$$\pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\boldsymbol{\theta}) = \exp\left[\ell^{(R)}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\boldsymbol{\theta}) - \ell^{(R)}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\widehat{\boldsymbol{\theta}})\right]$$

- We can write

$$\pi_{\boldsymbol{\theta}|\boldsymbol{y}}(\boldsymbol{\theta}) \propto \underbrace{\exp(\ell(\boldsymbol{\theta}) - \ell(\widehat{\boldsymbol{\theta}}))}_{\text{relative likelihood}} \times \underbrace{\exp\left(-\boldsymbol{\theta}^T \text{diag}(\boldsymbol{\lambda},\boldsymbol{\xi})\boldsymbol{\theta}\right)}_{\text{prior } \pi_0(\boldsymbol{\theta})}$$

where $\pi_0 \sim \text{GFV}\left(\mathbf{0}, 2\,\text{diag}(\boldsymbol{\lambda},\boldsymbol{\xi})\right)$.

Normal approximation

- Using the normal approximation of the relative likelihood, we can write

$$\pi_{\boldsymbol{\theta}|\boldsymbol{y}} \sim \text{GFV}\left(\widehat{\boldsymbol{\theta}}, \mathcal{I}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\widehat{\boldsymbol{\theta}})\right)$$

  with precision matrix

$$\mathcal{I}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\widehat{\boldsymbol{\theta}}) = -\frac{\partial^2 \ell^{(R)}_{\boldsymbol{\lambda},\boldsymbol{\xi}}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}(\widehat{\boldsymbol{\theta}})$$

- This matrix can be computed by backpropagation or by numerical differentiation (slower, but this calculation needs to be performed only once).

φ-equation and linearisation

- We now wish to predict the response Y_0 for a new input vector $x = x_0$; it can be written as

$$Y_0 = \mu(x_0, \boldsymbol{w}) + \sigma(x_0, \boldsymbol{v})U = \varphi(x_0, \boldsymbol{\theta}, U)$$

with $U \sim N(0, 1)$.

- To approximate the condition possibility distribution $\pi_{Y_0|\boldsymbol{y},u}$ of Y_0 given $U = u$, we need to linearise both $\mu(x_0; \boldsymbol{w})$ and $\sigma(x_0, \boldsymbol{v})$ around, respectively, $\widehat{\boldsymbol{w}}$ and $\widehat{\boldsymbol{v}}$. We write

$$\mu(x_0; \boldsymbol{w}) \approx \mu(x_0; \widehat{\boldsymbol{w}}) + \nabla\mu_0^T(\boldsymbol{w} - \widehat{\boldsymbol{w}})$$

$$\sigma(x_0; \boldsymbol{v}) \approx \sigma(x_0; \widehat{\boldsymbol{v}}) + \nabla\sigma_0^T(\boldsymbol{v} - \widehat{\boldsymbol{v}})$$

with

$$\nabla\mu_0 = \left.\frac{\partial\mu(x_0; \boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\widehat{\boldsymbol{w}}} \quad \text{and} \quad \nabla\sigma_0 = \left.\frac{\partial\sigma(x_0; \boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}=\widehat{\boldsymbol{v}}}$$

Predictive RFS

- We thus have

$$Y_0 \approx \begin{pmatrix} \nabla\mu_0 \\ U\nabla\sigma_0 \end{pmatrix}^T \boldsymbol{\theta} + \mu(x_0, \widehat{\boldsymbol{w}}) - \nabla\mu_0^T \widehat{\boldsymbol{w}} + (\sigma(x_0, \widehat{\boldsymbol{v}}) - \nabla\sigma_0^T \widehat{\boldsymbol{v}})U$$

- Assuming matrix $\mathcal{I}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\widehat{\boldsymbol{\theta}})$ to be positive definite, and using a result about affine transformations of GRVs, it can be shown that, approximately,

$$\pi_{Y_0|\boldsymbol{y},u} \sim \text{GFN}\left(\mu(x_0, \widehat{\boldsymbol{w}}) + \sigma(x_0, \widehat{\boldsymbol{v}})u, h(x_0, \widehat{\boldsymbol{\theta}}, u)\right)$$

with

$$h(x_0, \widehat{\boldsymbol{\theta}}, u) = \left[\begin{pmatrix} \nabla\mu_0^T & u\nabla\sigma_0 \end{pmatrix} \mathcal{I}_{\boldsymbol{\lambda},\boldsymbol{\xi}}(\widehat{\boldsymbol{\theta}})^{-1} \begin{pmatrix} \nabla\mu_0 \\ u\nabla\sigma_0 \end{pmatrix}\right]^{-1} = \frac{1}{\alpha + \gamma u + \delta u^2}$$

- Uncertainty about Y_0 can, thus, be described by the predictive RFS

$$\widetilde{Y}(x_0) : U \mapsto \pi_{Y_0|\boldsymbol{y},U} \sim \text{GFN}\left(\mu(x_0; \widehat{\boldsymbol{w}}) + U\sigma(x_0; \widehat{\boldsymbol{v}}), h(x_0, \widehat{\boldsymbol{\theta}}, U)\right)$$

GRFN approximation

- From Jensen's inequality,

$$\mathbb{E}\left[\frac{1}{\alpha+\gamma U+\delta U^2}\right] \geq \frac{1}{\alpha+\gamma\mathbb{E}[U]+\delta\mathbb{E}[U^2]} = \frac{1}{\alpha+\delta}$$

- Replacing $h(x_0, \widehat{\boldsymbol{\theta}}, U)$ by the lower bound of its expectation

$$\widetilde{h}(x_0, \widehat{\boldsymbol{\theta}}) = \frac{1}{\alpha+\delta}$$

we obtain the conservative approximation:

$$\boxed{\widetilde{Y}(x_0) \sim \widetilde{N}\left(\mu(x_0; \widehat{\boldsymbol{w}}), \sigma^2(x_0; \widehat{\boldsymbol{v}}), \widetilde{h}(x_0, \widehat{\boldsymbol{\theta}})\right)}$$

Experimental results

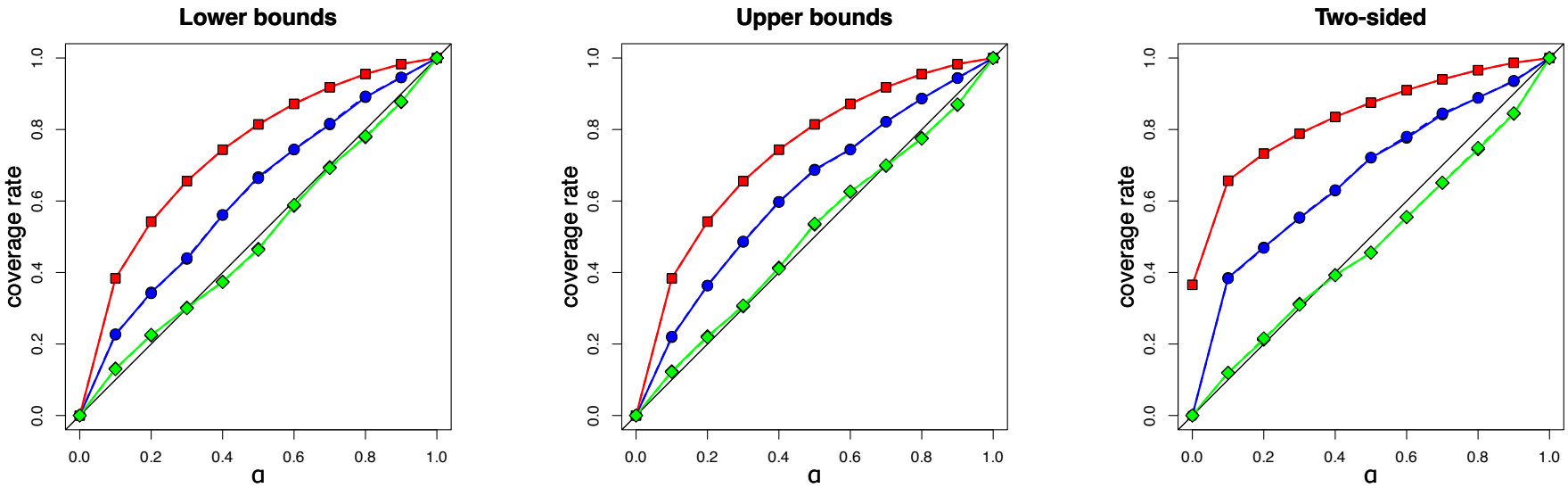
- Real datasets:

	n	p	response
Boston	506	13	`medv`
Energy	768	8	`Y2`
Concrete	1030	8	`strength`
Yacht	308	5	`Y`
Airfoil	1503	5	`Y`

- Neural networks with ELU activation functions, one layer of 5 units for the σ-network, one layer of 30 or 50 units for the μ-network.

Calibration curves

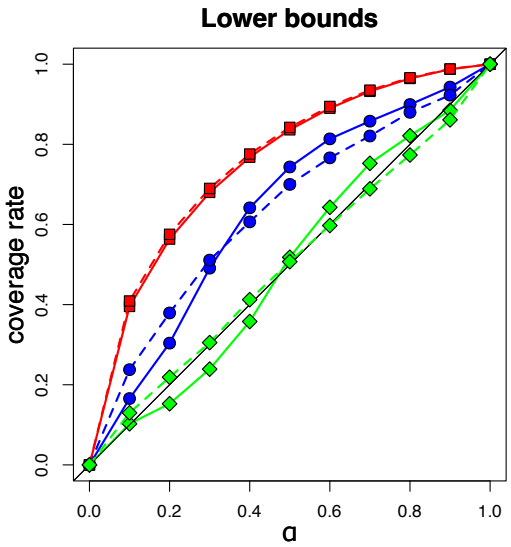
Boston dataset

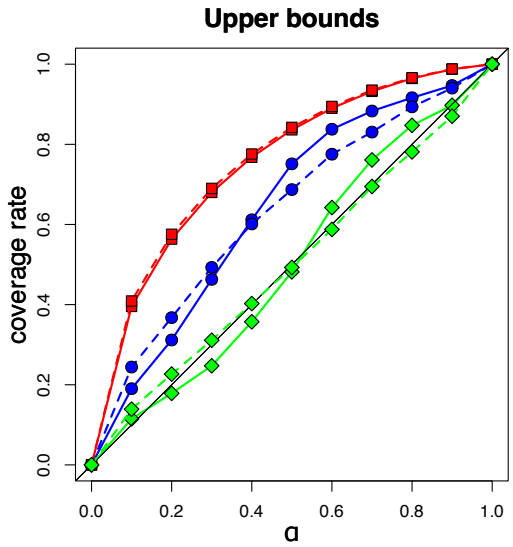


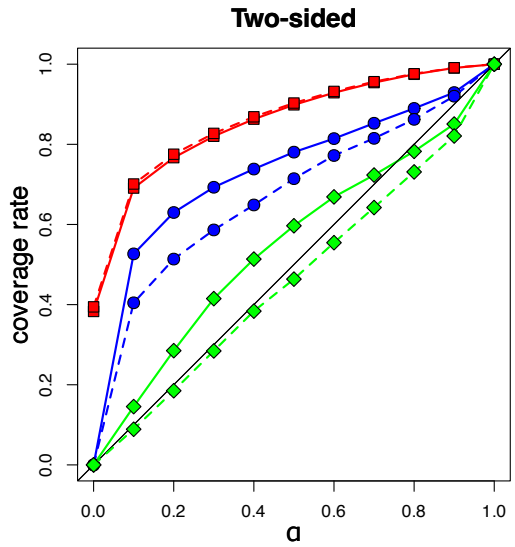
Calibration curves for belief lower bounds (left), belief upper bounds (middle) and two-sided belief intervals (right) computed using the validation data. The lower green curves, middle blue curve and upper red curves correspond, respectively, to the coverage rates of probabilistic predictive intervals, coverage rates of belief intervals, and average plausibilities of belief intervals

Calibration curves

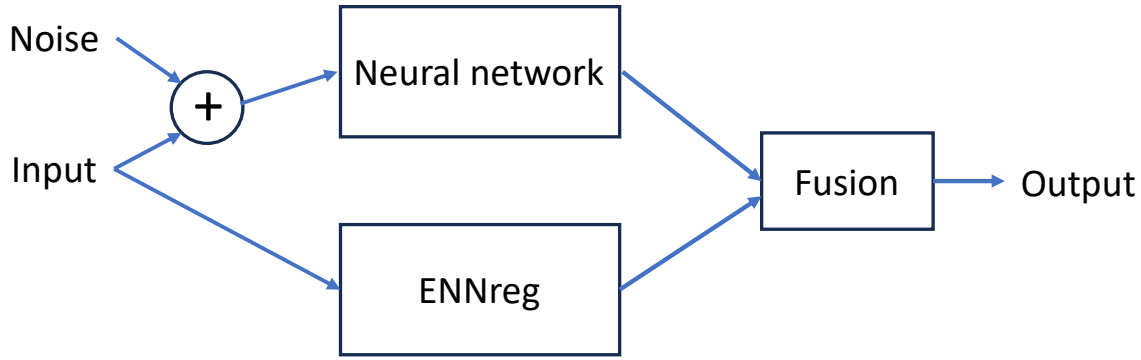
Energy dataset





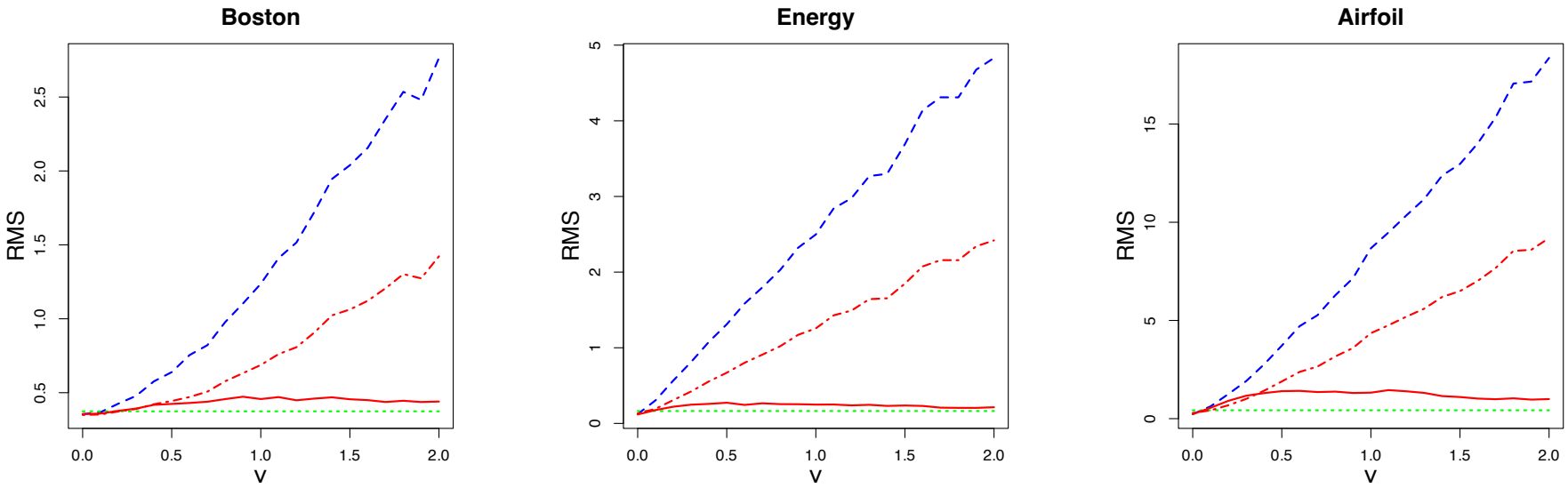


Experiment 1



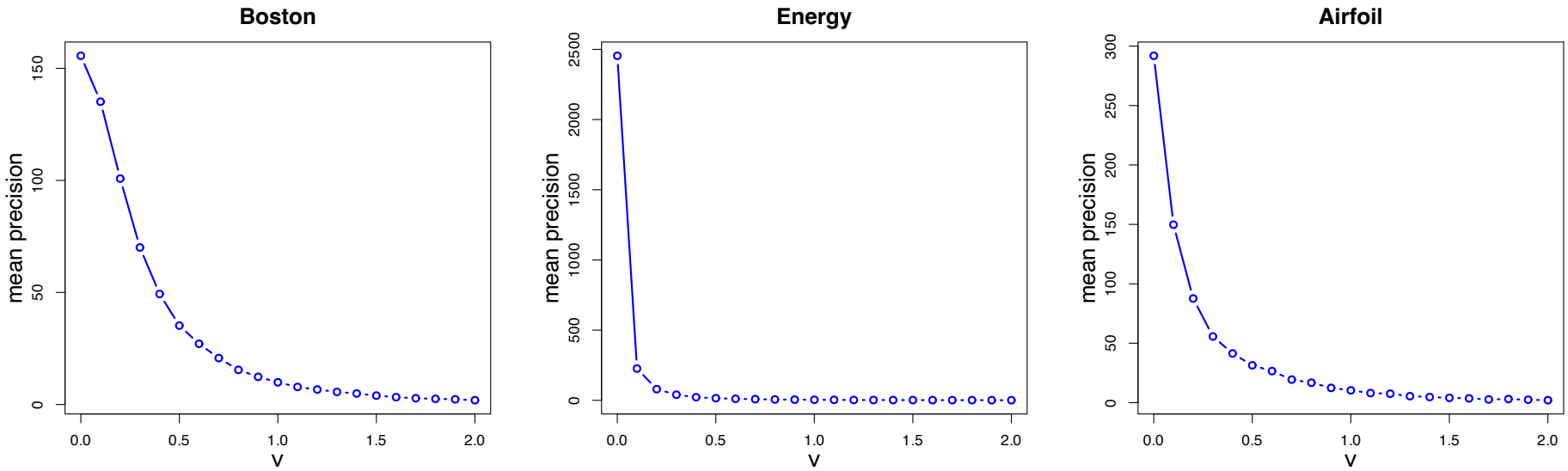
- To each test input of the neural network, we added a Gaussian noise with mean 0 and standard deviation ν.
- The outputs of the two models were combined by
  - averaging (using only the point predictions)
  - the product-intersection rule.
- The process was repeated ten times for each noise level, with ten 80%-20% training/test partitions.

Results



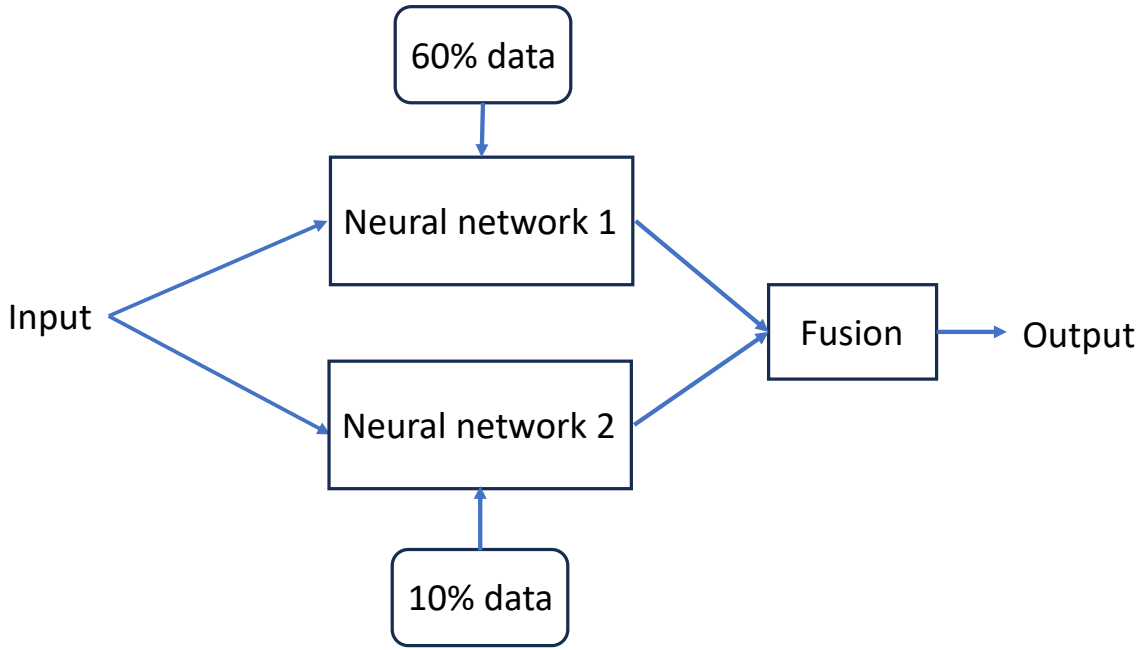
Root-mean-squared (RMS) error as a function of noise standard deviation ν for the following methods: (1) neural network with corrupted inputs (blues broken line), (2) ENNreg with uncorrupted inputs (horizontal green dotted line), average prediction (red dash-dotted line), (3) combination by the product-intersection rule (solid red line).

Precisions



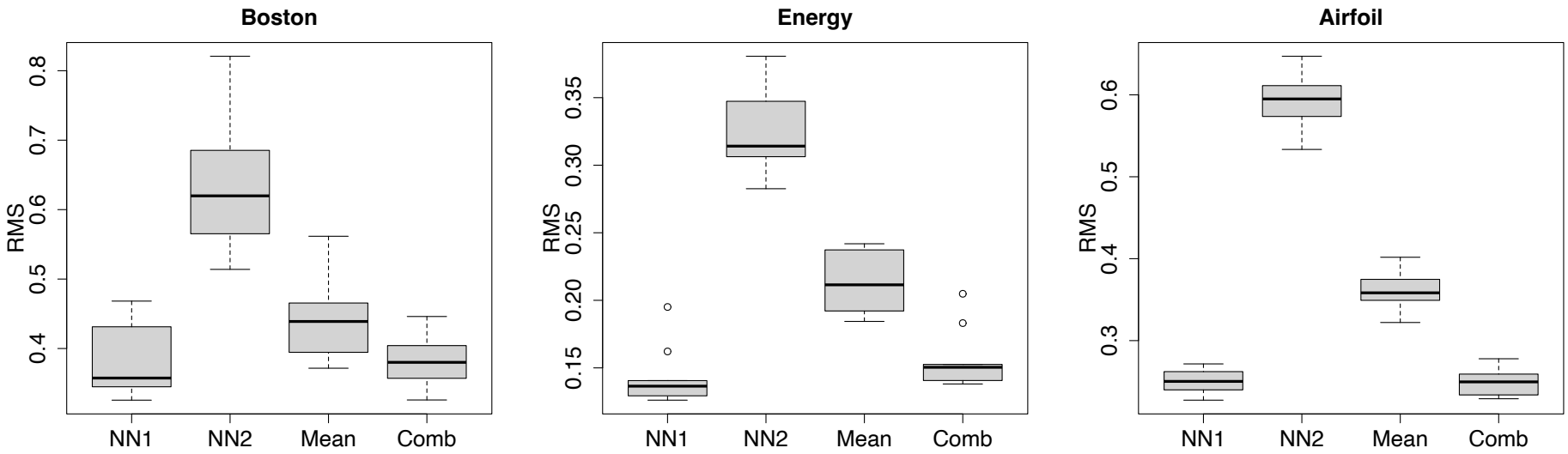
Mean precision as a function of noise standard deviation ν for the five datasets

Experiment 2



- The reliability of predictions also depends on the size of the training set.
- To simulate another fusion scenario involving the combination of predictions with different reliabilities, we combined the outputs of two neural networks, trained with 60% and 10% of the data.
- The process was repeated ten times with different random training/validation/test partitions.

Results



Box plots of root-mean-squared errors for the five datasets and the following methods: neural network trained on 60% of the data (NN1), neural network trained on 10% of the data (NN2), average prediction (Mean) and combination by the product-intersection rule (Comb).

Summary

- Evidential ML = ML with uncertainty quantified by belief functions.
- Two main approaches:
  1. Distance-based (nonparametric)
  2. Likelihood-based (parametric)
- The distance-based approach, initially proposed for classification, can be extended to regression using GRFNs.
- The likelihood-based approach yields predictive RFSs. It can be applied to regression or classification.
- These methods are based on generalised evidence theory, a sound and intuitively appealing theoretical framework.
- They provide cautious predictions, especially useful in situations of high uncertainty:
  - Small datasets
  - Low-quality data
  - Several sources providing partial information

References on ENN

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

T. Denœux.
A neural network classifier based on Dempster-Shafer theory
IEEE Transactions on Systems, Man and Cybernetics A, 30(2):131-150, 2000

Z. Tong, Ph. Xu and T. Denœux.
An evidential classifier based on Dempster-Shafer theory and deep learning
Neurocomputing 450:275–293, 2021

L. Huang, S. Ruan, P. Decazes and T. Denœux
Lymphoma segmentation from 3D PET-CT images using a deep evidential network
International Journal of Approximate Reasoning 149:39-60, 2022

L. Huang, S. Ruan, P. Decazes and T. Denœux
Deep evidential fusion with uncertainty quantification and reliability learning for multimodal medical image segmentation
Information Fusion, 113:102648, 2025

References on ENNreg

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

T. Denœux
Quantifying Prediction Uncertainty in Regression using Random Fuzzy Sets: the ENNreg model
IEEE Transactions on Fuzzy Systems, 31(10):3690–3699, 2023

L. Huang, Y.Xing, S. Mishra, T. Denœux and M. Feng.
Evidential time-to-event prediction with calibrated uncertainty quantification
International Journal of Approximate Reasoning, 181:109403, 2025

References on the likelihood-based approach

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

O. Kanjanatarakul, S. Sriboonchitta and T. Denoeux.
Forecasting using belief functions: an application to marketing econometrics.
International Journal of Approximate Reasoning, 55(5):1113–1128, 2014.

O. Kanjanatarakul, T. Denœux and S. Sriboonchitta.
Prediction of future observations using belief functions: a likelihood-based approach
International Journal of Approximate Reasoning 72:71–94, 2016

T. Denœux
Uncertainty Quantification in Logistic Regression using Random Fuzzy Sets and Belief Functions
International Journal of Approximate Reasoning, Vol. 168, 109159, 2024

T. Denœux
Uncertainty Quantification in Regression Neural Networks using Evidential Likelihood-based Inference
International Journal of Approximate Reasoning, Vol. 182, 109423, 2025