Distribution-Free Data Uncertainty for Neural Network Regression

Domokos M. Kelen[1], Adám Jung[1], Péter Kersch[2], András A. Benczúr[1 3]

[1]HUN-REN SZTAKI, [2]Ericsson Hungary, [3]Széchényi University, Győr, Hungary

SZTAKI

MOTIVATION & SUMMARY

- **Goal**: predict the **aleatoric (data) uncertainty** distribution of the target variable
- **Problem**: Most uncertainty methods assume **parametric distributions**
- **Solution**: We propose a method that capable of learning and representing **arbitrary distributions**

TYPES OF UNCERTAINTY

Epistemic Uncertainty:

"We lack information to know the right model"

- Not enough data
- Methods: *Bayesian Neural Nets, Deep Ensembles, Monte Carlo Dropout, etc.*

Aleatoric Uncertainty:

"We lack information to know the exact value of Y"

- Not enough features
- Methods: *Parametric, Mixture Density Networks, **Ours***

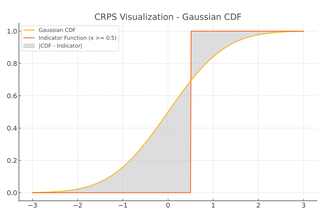
CRPS LOSS

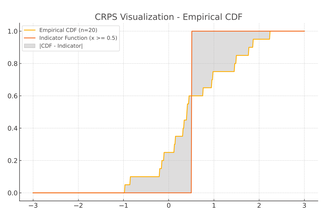
- We use the **Contonuous Ranked Probability Score:**

$$\mathrm{CRPS}(F, y) = \int_{\mathbb{R}} (F(z) - \mathbb{1}\{y \le z\})^2 \, \mathrm{d}z$$

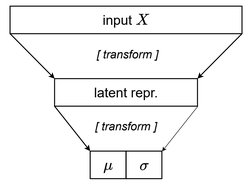
- Is a **strictly proper scoring rule**
- Can be approximated using **sampling**
- We derive an **unbiased** $\mathcal{O}(n \log n)$ formula

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} (\tilde{y}_{(i)} - y) \left((n-1)\mathbb{1}\{y < \tilde{y}_{(i)}\} - i + 1 \right)$$

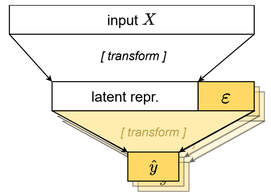




SAMPLING ARCHITECTURE



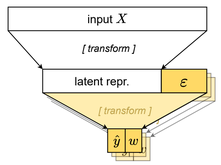
Traditional



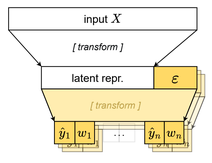
Ours

- Traditional: outputs are **parameters** of a parametric distribution (e.g., Gaussian)
- Ours: nondeterministic **sampling** architecture

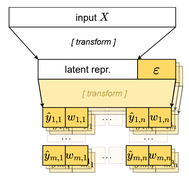
WEIGHTED SAMPLING ARCHITECTURE



Weighted



Multiple sampling heads



Multiple sampling layers

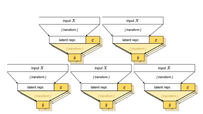
- Weighted loss formula and architectures enable much more expressive output distributions

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} w_{(i)} (\hat{y}_{(i)} - y) \left((n-1)\, \mathbb{1}\{y < \hat{y}_{(i)}\} - s_i + \frac{W - n + w_{(i)} + 1}{2} \right)$$

EXTENSIONS

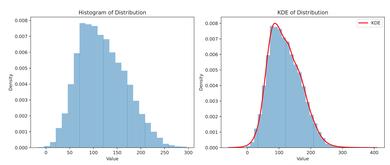
$$\mathrm{ES}(F, y) = \frac{1}{n} \sum_i \|y - \tilde{y}_i\|_2 - \frac{1}{2n^2} \sum_i \sum_j \|\tilde{y}_i - \tilde{y}_j\|_2$$

MULTIVARIATE: ENERGY SCORE



Deep Ensembles

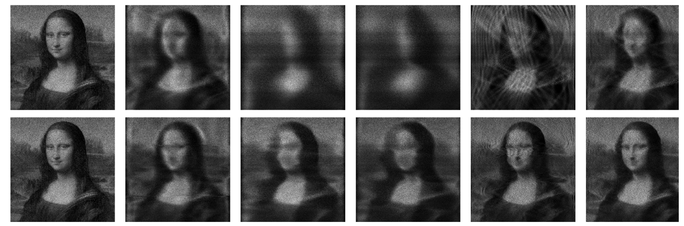
EPISTEMIC UNCERTAINTY



KERNEL DENSITY ESTIMATION

EXPERIMENTS

SYNTHETIC: MONA LISA



(a) Original (b) MDN (c) Simple (d) Weighted (e) MH (f) LMH

UCI REGRESSION UNCERTAINTY BENCHMARK

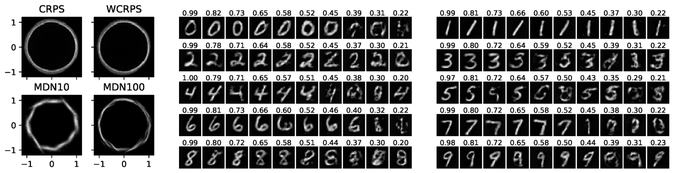
Table 1: NLL results on the UCI Datasets benchmark (lower is better)

dataset	n	Dropout	PBP-MV	Ensembles	MDN	$\mathrm{MDN}_{\mathrm{bnn}}$	WCRPS	WCRPS_{σ}
boston	0.5k	2.40 ±0.04	2.54 ±0.08	2.44 ±0.05	2.52 ±0.04	2.36 ±0.03	2.40 ±0.03	**2.32** ±0.05
concrete	1.0k	**2.94** ±0.02	3.04 ±0.03	2.97 ±0.04	3.13 ±0.04	2.99 ±0.03	3.09 ±0.04	2.95 ±0.05
energy	0.8k	1.21 ±0.01	1.01 ±0.01	0.62 ±0.08	0.77 ±0.04	0.65 ±0.02	0.47 ±0.04	**0.31** ±0.03
kin8nm	8.2k	-1.14 ±0.01	-1.28 ±0.01	-1.34 ±0.00	-1.22 ±0.01	-1.30 ±0.01	-1.32 ±0.01	**-1.38** ±0.01
naval	11.9k	-4.45 ±0.00	-4.85 ±0.06	-6.49 ±0.02	-6.24 ±0.04	**-6.72** ±0.01	-6.65 ±0.03	-6.51 ±0.03
power	9.6k	2.81 ±0.01	2.78 ±0.01	2.69 ±0.01	2.65 ±0.01	2.62 ±0.01	2.66 ±0.01	**2.57** ±0.01
protein	45.7k	2.87 ±0.00	2.77 ±0.01	2.66 ±0.02	2.00 ±0.01	**1.96** ±0.01	2.22 ±0.01	2.05 ±0.01
wine	1.6k	0.93 ±0.01	0.97 ±0.03	0.91 ±0.01	-3.48 ±0.03	**-5.11** ±0.09	0.90 ±0.03	0.80 ±0.03
yacht	0.3k	1.25 ±0.02	1.64 ±0.02	-0.02 ±0.04	0.26 ±0.09	0.63 ±0.04	0.07 ±0.06	**-0.16** ±0.05

Table 2: RMSE results on the UCI Datasets benchmark (lower is better)

dataset	n	Dropout	PBP-MV	Ensembles	MDN	$\mathrm{MDN}_{\mathrm{bnn}}$	WCRPS	WCRPS
boston	0.5k	3.61 ±0.23	3.11 ±0.18	3.37 ±0.19	3.73 ±0.25	2.93 ±0.20	3.07 ±0.23	**2.91** ±0.1
concrete	1.0k	5.45 ±0.19	5.08 ±0.14	5.19 ±0.19	5.99 ±0.12	5.25 ±0.11	5.51 ±0.14	**4.94** ±0.1
energy	0.8k	0.97 ±0.06	0.45 ±0.01	0.86 ±0.12	1.17 ±0.09	0.51 ±0.01	0.45 ±0.02	**0.41** ±0.0
kin8nm	8.2k	0.09 ±0.00	0.07 ±0.00	0.07 ±0.00	0.07 ±0.00	0.07 ±0.00	0.07 ±0.00	**0.06** ±0.0
naval	11.9k	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.0
power	9.6k	4.18 ±0.04	3.91 ±0.04	3.73 ±0.03	3.93 ±0.04	3.86 ±0.04	3.79 ±0.04	**3.62** ±0.0
protein	45.7k	4.39 ±0.02	3.94 ±0.02	4.20 ±0.03	3.96 ±0.02	4.09 ±0.02	3.76 ±0.02	**3.47** ±0.0
wine	1.6k	0.66 ±0.03	0.64 ±0.01	**0.63** ±0.00	0.66 ±0.01	0.66 ±0.01	0.64 ±0.01	**0.63** ±0.0
yacht	0.3k	1.23 ±0.37	0.81 ±0.06	0.92 ±0.08	1.97 ±0.25	**0.77** ±0.07	0.96 ±0.13	0.78 ±0.0

MULTIVARIATE: PARABOLOID, MNIST



(a) Inverting $z^2 = x^2 + y^2$ (b) Modeling the distribution of MNIST pixel data from the label.

SOURCE CODE: PYTHON PACKAGE

TorchNaut

- Implementations of CPRS-based models and MDN
- Support for epistemic methods (BNN, Deep Ensembles)
- GPU-accelerated adaptive-bandwidth KDE
- Multivariate versions
- Easy to use, example notebooks, API documentation

kdomokos@info.ilab.sztaki.hu

https://github.com/proto-n/torch-naut