Evidential Turing Processes

International Conference on Learning Representations 2022

Authors:

- Melih Kandemir, *University of Southern Denmark*
- Abdullah Akgül, *Istanbul Technical University*
- Manuel Haußmann, *Aalto University*
- Gozde Unal, *Istanbul Technical University*

Paper URL:

https://openreview.net/forum?id=84NMXTHYe-

Code:

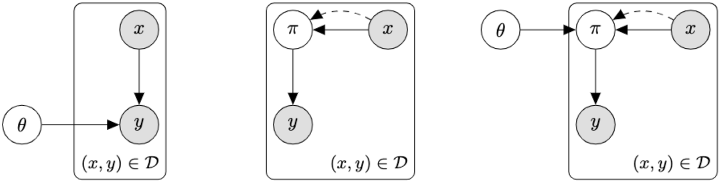
https:
//github.com/ituvisionlab/EvidentialTuringProcess

The Total Calibration Problem

We aim to develop a *standalone* predictor that can minimize the risk of

- **Model misfit** $\Rightarrow$ Reducible model uncertainty
  Measured directly by Negative Log-Likelihood (NLL)
- **Class overlap** $\Rightarrow$ Data Uncertainty
  Measured indirectly by Expected Calibration Error (ECE)
- **Domain mismatch** $\Rightarrow$ Irreducible model uncertainty
  Measured indirectly by error of in-domain/out-of-domain classification with predictive uncertainty

Three approaches to Bayesian modeling



(a) Parametric Bayesian Model (b) Evidential Bayesian Model (c) Complete Bayesian Model

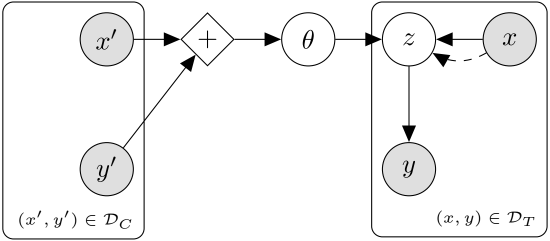
Variances of the posterior predictives decompose as below:

- Parametric = Reducible model unc + Data unc
- Evidential = Irreducible model unc + Data unc
- Complete = Reducible model unc + Irreducible model unc + Data unc

The recipe for total calibration

1. Use the Complete Bayesian Model framework
2. Learn to calibrate at training time
3. Model the relationship of global and local latent variables as a *Neural Process*
4. Use a random portion of each minibatch as context
5. Accumulate uncertainty information into an *external memory*
6. Use the memory content to determine the hyperparameters of the global latent variable

Neural Processes (Step 3)



The Neural Process is a conditional variational auto-encoder with three additional elements

- A global variable θ
- A context set $\mathcal{D}_c$
- A permutation-invariant aggregator function $+$

External Memory: Neural Turing Machines (Step 5)

A differentiable external memory M, where a value x is queried by a key function $k_\phi(h_\psi(x), h_\psi(x'))$ after being embedded into a latent space via h_ψ. It returns

$$v = \sum_{x' \in M} \mathrm{softmax}(k_\phi(h_\psi(x), h_\psi(x')))x'$$

The memory is updated by an application-specific rule

$$M' \leftarrow f(M, x).$$

Turing Processes: Neural Turing Machine $+$ Neural Process

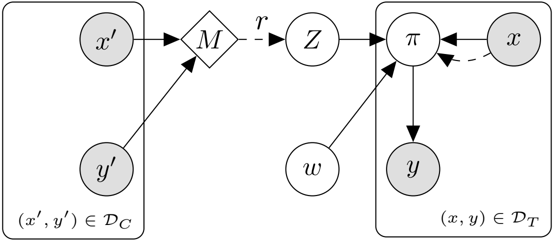
A Turing Process is a collection of random variables $Y = \{y_i | i \in \mathcal{I}\}$ for an index set $\mathcal{I} = \{1, \ldots, K\}$ with arbitrary $K \in \mathbb{N}^+$ that satisfy the two properties

i) $p_M(Y) = \int \prod_{y_i \in Y} p_M(y_i|\theta) p_M(\theta) d\theta$,

ii) $p_{M'}(Y|Y') = \int \prod_{y_i \in Y} p(y_i|\theta) p_{M'}(\theta|Y') d\theta$.

for another random variable collection Y' of arbitrary size that lives in the same probability space as Y, a probability measure $p_M(\theta)$ with free parameters M and some function $M' = r(M, Y')$.

Target Model: The Evidential Turing Process



$$p(y, \mathcal{D}_T, \pi, \theta) = p(w) \underbrace{p_M(Z)}_{\text{External memory}} \prod_{(x,y)\in\mathcal{D}_T} \Big[p(y|\pi) \underbrace{p(\pi|Z, w, x)}_{\text{Input-specific prior}} \Big]$$

Local latent variable π is a function of global Z and input x:

- Model its dependency on x as in evidential learning
- Model the relationship of Z and x via an attention mechanism

Case Study: Classification with ETP

Model:

$$
\begin{aligned}
Z|M &\sim \prod_{r=1}^{R} \mathcal{N}(z_r|m_r, \kappa^2 I), \\
w &\sim \mathcal{N}(w|0, \beta^{-1} I), \\
\pi|w &\sim Dir\big(\pi|\exp(a(v_w(x); Z))\big), \\
y|\pi &\sim Cat(y|\pi).
\end{aligned}
$$

Memory update rule:

$$
m \leftarrow \mathbb{E}_Z\left[\tanh\left(\gamma m + (1-\gamma)\sum_{(x,y)\in\mathcal{D}_C} \phi(v_w(x), z)[y + \sigma(v_w(x))]\right)\right].
$$

Application 1: Accurate classification with reliable uncertainties

Domain Data (Architecture)	IMDB (LSTM)	Fashion (LeNet5)	SVHN (LeNet5)	CIFAR10 (LeNet5)	CIFAR100 (ResNet18)
Prediction accuracy as % test error					
BNN	**16.4** ±**0.6**	**7.9**±**0.1**	7.9±0.1	**15.3**±**0.3**	30.2 ±0.3
EDL	38.3 ±13.3	8.6±0.1	7.3±0.1	18.5±0.2	45.2 ±0.4
ENP	50.0 ±0.0	**7.9**±**0.2**	**6.7**±**0.1**	**14.8**±**0.2**	39.0 ±0.3
ETP (Target)	**15.8** ±**1.3**	**7.9**±**0.2**	**6.9**±**0.1**	**15.3**±**0.2**	**29.2** ±**0.3**
In-domain calibration as % Expected Calibration Error (ECE)					
BNN	14.4 ±0.4	6.7±0.	6.5±0.	5.5±0.3	15.2±0.0
EDL	41.1 ±2.6	3.7±0.2	4.0±0.1	9.0±0.2	**5.3**±**0.4**
ENP	**0.8** ±**1.6**	6.0±0.2	10.7±0.2	7.2±0.3	39.7±0.4
ETP (Target)	**3.1** ±**0.4**	**2.6**±**0.2**	**2.6**±**0.1**	**2.7**±**0.1**	6.6±0.1
Model fit as negative test log-likelihood					
BNN	0.47 ±0.0	0.65±0.0	0.71±0.0	0.50±0.0	1.78±0.0
EDL	0.66 ±0.1	0.37±0.0	0.34±0.0	0.72±0.0	2.24±0.0
ENP	0.69 ±0.0	0.34±0.0	0.33±0.0	0.50±0.0	2.52±0.0
ETP (Target)	**0.37** ±**0.0**	**0.29**±**0.0**	**0.26**±**0.0**	**0.46**±**0.0**	**1.36**±**0.0**
Out-of-domain detection as % Area Under ROC Curve					
OOD Data	Random	MNIST	CIFAR100	SVHN	TinyImageNet
BNN	**60.9** ±**4.2**	75.9±2.3	86.2±0.5	**84.1**±**1.3**	97.2±0.5
EDL	**55.1** ±**5.1**	77.5±2.0	90.9±0.3	79.2±0.7	89.6±0.3
ENP	**53.7** ±**5.7**	**88.9**±**1.0**	**92.4**±**0.4**	**81.4**±**0.8**	**100.0**±**0.1**
ETP (Target)	**59.1** ±**5.1**	**90.0**±**0.9**	90.0±0.4	**82.1**±**0.6**	99.6±0.1

Application 2: Robustness against data corruption