Bayesian Inference and Explainable AI

Quantifying the uncertainty of the explanations

Vasilis Gkolemis[1]

[1]ATHENA Research and Innovation Center

November 2021

Program

Traditional Machine Learning

- Parametric model $f_{\theta} : \boldsymbol{x} \rightarrow y$
- Define a distance function $d(\cdot, \cdot)$ and measure the distance (loss) from observed data

$$L(\boldsymbol{\theta}) = \sum_{i}^{N} d(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y^i) \tag{1}$$

- Search for the parameter set $\hat{\boldsymbol{\theta}}$ that reproduces the observed data best

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{2}$$

We search for a single configuration (point-estimate) $\hat{\boldsymbol{\theta}}$

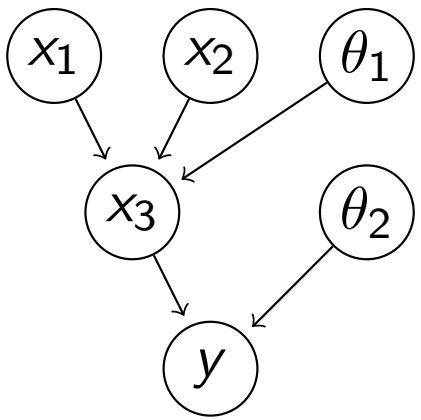
Bayesian Formulation

- On the modelling part:
  - we need the joint distribution $p(\boldsymbol{x}, y, \boldsymbol{\theta})$
  - to replace the parametric model $f_{\boldsymbol{\theta}} : \boldsymbol{x} \rightarrow y$
- Training part:
  - infer the posterior distribution $p(\boldsymbol{\theta}|D)$
  - to replace the optimal point estimate $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$
- Prediction part:
  - infer the predictive distribution $p(y|\boldsymbol{x}, D)$
  - to replace the point-estimate prediction $y = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})$

We replace point estimates with distributions (uncertainty quantification)

Modelling part

- joint distribution $p(\boldsymbol{x}, y, \boldsymbol{\theta}) = p(y|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x})p(\boldsymbol{\theta})$
- $p(\boldsymbol{\theta})$, our prior belief about the parameters of the model
- $p(y|\boldsymbol{x}, \boldsymbol{\theta})$, the likelihood of the model
- joint distribution can be defined as a DAG



- We need to model $p(\boldsymbol{\theta})$ and $p(y|\boldsymbol{x}, \boldsymbol{\theta})$

Training part

- We use Bayes law to infer the posterior distribution

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} \propto \prod_{i}^{N} p(y^i|\boldsymbol{x}^i, \boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{3}$$

  where $D = \{\boldsymbol{x}^i, y^i\}_{i=\{1,\dots,N\}}$, the observed data (training-set)
- In the extreme case where $p(\boldsymbol{\theta}|D) = \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$, we get a point-estimate is in traditional ML

The 'training process' leads to many possible models, each one with different probability (uncertainty about the model)

Inference part

- We need to solve/approximate the predictive distribution $p(y|\boldsymbol{x}, D) = \int_{\boldsymbol{\theta}} p(y|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|D) \partial \boldsymbol{\theta}$
- We consider the posterior $p(\boldsymbol{\theta}|D)$ as known (computed exactly or approximated)
- In the extreme case where $p(\boldsymbol{\theta}|D) = \delta(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$, we get all the mass of the prediction $p(y|\boldsymbol{x}, D) = p(y|\boldsymbol{x}, \hat{\boldsymbol{\theta}})$ from a single model

---

The 'prediction process' gets one prediction per each plausible model (uncertainty about the model leads to uncertainty about the prediction)

Bayesian Formulation - Disadvantages

What we lose

- On the modelling part
  - Time to think how the input features x_i relate to each other i.e. building the DAG
- On the training-prediction (inference) part
  - Expressions difficult to approximate
  - $p(\boldsymbol{\theta}|D)$ - how to compute the posterior distribution?
  - $p(y|\boldsymbol{x}, D) = \int_{\boldsymbol{\theta}} p(y|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|D) \partial \boldsymbol{\theta}$ - how to compute the predictive distribution?

Bayesian Formulation is difficult from both the mathematical and the computational point-of-view

Bayesian Formulation - Advantages

What we get:

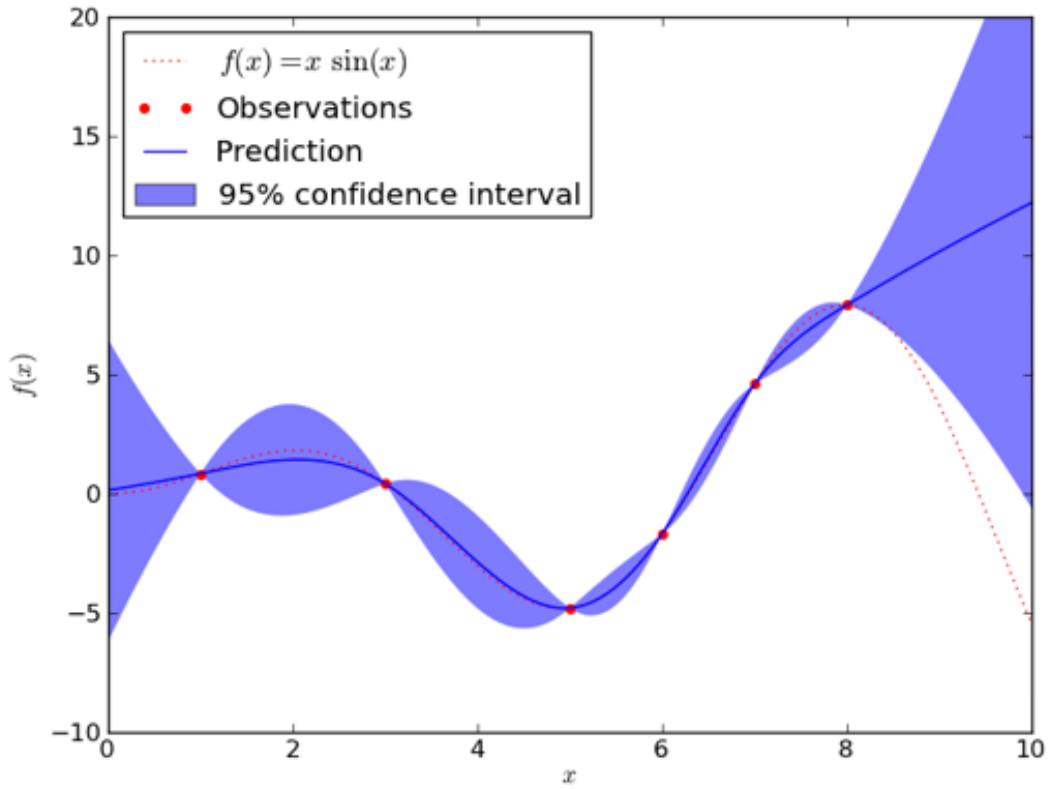
- On the modelling part
  - Specify who the features relate to each other
  - check Model-based Machine Learning - a new approach for ML model building
- On the inference part
  - Uncertainty estimation! Why we need it?
  - Most times the available data is not enough to reveal a single instance $\hat{\boldsymbol{\theta}}$
  - Sometimes we want to predict on a new $\boldsymbol{x}$ that is very different from the training set

---

Let's be wise enough and be uncertain about our predictions.

Bayesian Formulation - Example

- In areas without training points our uncertainty is bigger



Program

Overview of my MSc Dissertation

- In BI, estimating the posterior is the major difficult task
- Remember, $\textit{posterior} \propto \textit{likelihood} * \textit{prior}$
- When the likelihood is intractable $\rightarrow$ Likelihood-free inference (LFI) methods
- That was my MSc about:
  - Extending ROMC, a likelihood-free inference method
  - Initial paper by Ikonomov and Gutmann 2020
  - Implementing ROMC ELFI, a python package for LFI
  - https://elfi.readthedocs.io/en/latest/

LFI methods approximate the posterior when the likelihood is intractable - very difficult problem

Cases of intractable likelihood

- What causes an intractable likelihood?
  - Many reasons (e.g. intractable partition functions in unnormalized statistical models)
  - Most common reason $\rightarrow$ unobserved/latent variables
  - Latent variables are really important in many modelling cases (example in next slide)
- Simulator-based models (those with intractable likelihood) are widely-used in natural sciences i.e. biology, epidemiology, neuroscience
- An overview of the field in Cranmer, Brehmer, and Louppe 2020

Simulator-based models provide valuable modelling freedom

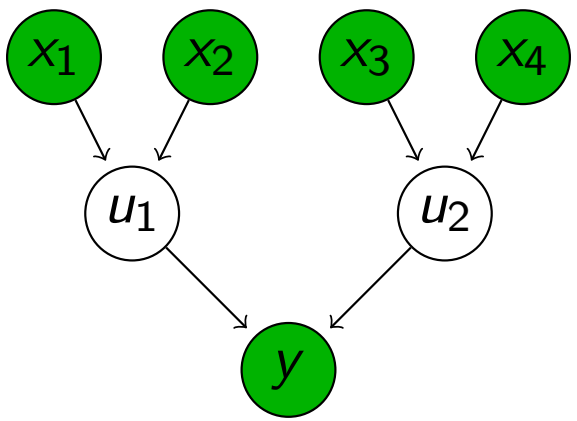
Example of intractable likelihood (1)

- Predict the grade y of a candidate at an important test
- Grade is a direct consequence of two things
  - $u_1 \in [0, 10]$ the mental readiness of the candidate
  - $u_2 \in [0, 10]$ the knowledge of the topic
- the mental readiness is direct consequence of
  - x_1 how many hours he has slept the previous night
  - x_2 how many times he has been in stressful exams in the past
- knowledge of the topic
  - x_3, years of experience in software engineering
  - x_4, number of application he has developed

Think more about it; it is sensible to model the latent variables!

Example of intractable likelihood (2)

- $L(\boldsymbol{\theta}) = p(D|\boldsymbol{\theta}) \propto \prod_i^N p(y^i|x^i, \boldsymbol{\theta}) = \int_{u_1,u_2} p(y^i, u^i|x^i, \boldsymbol{\theta})$
- The likelihood is defined over an integral - intractable in the general case
- But sampling is feasible i.e. draw $y \sim p(y|x, \boldsymbol{\theta})$



If we want latent variables, we have to use likelihood-free methods.

Optimization Monte Carlo (OMC)

- Core idea: Convert random sampling to a deterministic process Meeds and Welling 2015

$$x \sim p(x|\boldsymbol{\theta}) \Rightarrow f(\boldsymbol{\theta}, v) \rightarrow x \tag{4}$$

- v is the nuisance variable that absorbs all the randomness
- $\arg\max_{\boldsymbol{\theta}} p(x = x_0|\boldsymbol{\theta}) \rightarrow \arg\min_{\boldsymbol{\theta}} |f(\boldsymbol{\theta}, v = v_0) - x_0|$
- In a computer program, the value that governs all randomness is the random seed

---

Maximizing the probability of generating some data can be converted to a deterministic optimization process

Robust Optimization Monte Carlo (ROMC), step (1)

- What is the objective? A way to approximate the intractable $L(\boldsymbol{\theta}) = p(D|\boldsymbol{\theta})$
- What I have? Only a way to simulate points from $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ (random simulator)
- Draw random seeds s and generate deterministic simulators $f_i(\boldsymbol{\theta})$
- For every f_i, search for the $\boldsymbol{\theta}_i^* = argmin_{\boldsymbol{\theta}} |f_i(\boldsymbol{\theta}) - D|$
- For every f_i, define an area $\mathcal{S}_i$ around $\boldsymbol{\theta}_i^*$ such that $|f_i(\boldsymbol{\theta}) - D| < \epsilon$
- $\mathcal{S}_i$ is the acceptance region of the $i - th$ simulator

---

Convert random sampling to an optimization process and find the parameters $\boldsymbol{\theta}_i$ that generate data close to the observations.

Robust Optimization Monte Carlo (ROMC), step (2)

- The approximate posterior is the sum of all such regions S_i, scaled by the prior

$$p(\boldsymbol{\theta}|D) \approx p(\boldsymbol{\theta})\frac{1}{N}\sum_i \mathcal{S}_i(\boldsymbol{\theta}) \tag{5}$$

  where $\mathcal{S}_i(\boldsymbol{\theta}) = \frac{1}{V}$ if $|f_i(\boldsymbol{\theta}) - D| < \epsilon$, otherwise 0
- Each point inside the area is equally probable to have generated the data

---

The posterior approximation is the sum of all the regions $\mathcal{S}_i(\boldsymbol{\theta})$ that generate data close to the observations. (as in typical loss minimization)

An example of an acceptance region

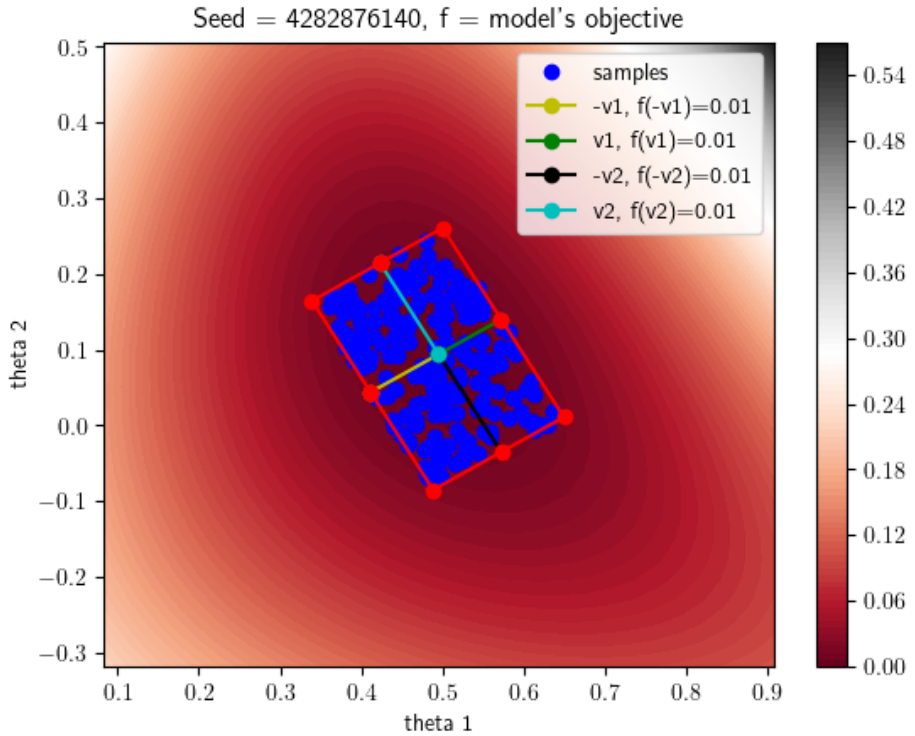


Figure: The acceptance region S_i of a specific optimisation problem, that will contribute to the posterior.

ROMC - Training/Fitting part

1: **procedure** ROMC
2: **for** $i \leftarrow 1$ to n_1 **do**
3: $\mathrm{v}_i \sim p(\mathrm{v})$ ▷ Draw nuisance variables
4: $f_i(\boldsymbol{\theta})$ ▷ Define deterministic simulator
5: $\boldsymbol{\theta}_i^* = \text{argmin}_{\boldsymbol{\theta}} |f_i(\boldsymbol{\theta}) - D|$ ▷ Solve optimization problem
6: **if** $|f_i(\boldsymbol{\theta}) - D| > \epsilon$ **then**
7: Go to 2 ▷ Filter solution
8: Define the proposal area $\hat{\mathcal{S}}_i$ ▷ Estimate proposal area

Algorithmic view of the training part of ROMC

ROMC - Training part (Recap)

- Target of Bayesian Inference
  - infer the posterior distribution $p(\boldsymbol{\theta}|D)$
- ROMC proposal

$$p(\boldsymbol{\theta}|D) \approx p(\boldsymbol{\theta}) \frac{1}{N} \sum_i \mathcal{S}_i(\boldsymbol{\theta}) \tag{6}$$

- where
  - $\mathcal{S}_i(\boldsymbol{\theta}) = \frac{1}{V}$ if $|f_i(\boldsymbol{\theta}) - D| < \epsilon$, otherwise 0
  - V is the volume of $\mathcal{S}_i(\boldsymbol{\theta})$

ROMC approximation of the posterior distribution using only the simulator

ROMC - Prediction part

1: ---

2: **procedure** ROMC

3: **for** $i \leftarrow 1$ to n_1 **do**

4: **for** $j \leftarrow 1$ to n_2 **do**

5: $\boldsymbol{\theta}_{ij} \sim q_i$, compute w_{ij} ▷ Sample

6: $E_{p(\boldsymbol{\theta}|D)}[h(\boldsymbol{\theta})]$ ▷ Estimate an expectation

7: $p_{d,\epsilon}(\boldsymbol{\theta})$ ▷ Evaluate the unnormalized posterior

8: ---

Algorithmic view of the prediction part of ROMC; how to sample from the posterior and estimate an expectation.

ROMC - Prediction part

- Target of Bayesian Inference
  - infer the predictive distribution $p(y|x, D)$
- ROMC proposal
  - Sample from each samples from each acceptance region $(w_{ij}, \boldsymbol{\theta}_{ij})$
  - Normalize them to sum to one $w_{ij} = \frac{1}{\sum_j w_{ij}}$
  - $p(y|x, D) \approx \sum_i \sum_j w_{ij} f_i(\boldsymbol{\theta}, x)$

ROMC approximation of the predictive distribution using only the simulator

ROMC - Advantages

- Efficient and Embarrassingly Parallel Framework for likelihood-free inference
- Efficient, because it turns every random-generating process to a deterministic optimization problem $\Rightarrow$ does not spend samples unreasonably
- Embarrassingly Parallel, all optimisation processes can run in parallel $\Rightarrow$ super-fast if many cores are accesible
- Framework, because is a general recipe. The components that will be used depend on the user e.g. gradient-based optimizer or bayesian optimization

---

ROMC is efficient and parallelizable.

Implementation at ELFI

- Fully parallelizable and extendable implementation at ELFI
- https://elfi.readthedocs.io/en/latest/
- Interactive jupyter notebooks with examples (google colab - no need for installing anything)
  - Simple 1D example
  - Simple 2D example
  - Moving Average example
  - Tutorial for extending the ROMC method with a Neural Network
- Paper about to be submited at JSS

Feel-free to experiment with ROMC

- Find ways to make it efficient in high dimensional parametric space (≈ 20 is high for LFI methods)
- Implement ROMC in a package with automatic differentiation, to see how it scales in higer-dimensions
- JAX a good candidate, provides novel way to freeze the seed without global side-effects
- Research for better (accurate and efficient) ways to estimate the regions $\mathcal{S}_i$ in high-dimensions

---

Stay alert, there will be updates!