Probabilistic Programming in Python

Salman Khan

Motivation

Quantifying Uncertainty – Current frameworks are too rigid and don’t allow for uncertainty in model parameters.

Data Hungry Model – Machine learning requires a lot of data and often requires correctly labelled data

Prior/Expert Knowledge - Need for a framework to incorporate of prior knowledge or beliefs into the analysis/models.

Transparency - Need for transparency and accountability from decisions derived from AI.

Bayesian Vs. Frequentist Statistics

Frequentist

- Parameters are all fixed and unknown constants
- The frequentist approach deals with long-run probabilities i.e. how probable is this data set given the null hypothesis,
- Frequentist approach is purely data driven
- Uncertainty is in the data-generation process – p-values

Bayesian

- Parameters are random variables
- Bayesian approach deals with the probability of a hypothesis given a particular data set.
- Bayesian allows to incorporate prior information.
- Uncertainty in the parameter – modelled via its probability distribution

Source: (Fornacon-Wood 2022)

Bayes Theorem

Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

$$Posterior = \frac{Likelihood * Prior}{Marginal\ Likelihood}$$

Bayesian vs Non-Bayesian Inference

Bayesian Inference

Model Posterior Distribution

$$p(\theta \,|y) = \frac{p(y\,|\theta)\,p(\theta)}{p(y)}$$

where, $p(y) = \int p(y,\theta)P(\theta)d\theta$

Prediction

$$p(\tilde{y}|y) = \int p(\tilde{y}|\theta)p(\theta|y)d\theta$$

Non-Bayesian Inference

Model Likelihood: $p(y|\theta)$

Determine point estimate of unknown parameter

$$\theta^* = \ arg\max_{\theta} p(y|\theta)$$

Predication:

$$p(\tilde{y}|y) = p(\tilde{y}|\theta^*)$$

Bayesian Inference

How do we model posterior?

Can we sample from this posterior distribution?

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{p(y)}$$

If we can sample from posterior i.e. $\{\theta_i\}_i^N$ drawn from $p(\theta \mid y)$

Then Predictions is simply:

$$p(\tilde{y}|y) = \frac{1}{N}\sum_{i=1}^{N} P(\tilde{y}|\theta_i)$$

Markov Chain Monte Carlo (MCMC)

MCMC methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain.

- Metropolis-Hastings
- Gibbs Sampler
- Hamiltonian Monte Carlo (HMC)
- No-U-Turn Sampler (NUTS)
- Sequential Monte Carlo (SMC)

Source: Wikipedia

Probabilistic Modelling

Probabilistic Programming is just using a programming framework for Bayesian statistics.

- Thrive in a world of uncertainty
- Potential to work well with little data
- Ability to incorporate expert knowledge and prior beliefs
- Can be updated as new information is presented
- Offers transparency and explainable models

Workflow of Probabilistic Programming

1. Identify all the unknown parameters and define their prior distribution.
2. Specify Likelihood – probability distribution of observed data which is function of the unknown parameters.
3. Run Sampler for the posterior distribution (Inference Algorithm)

The sample build chains that approximate the posterior probability

Demo

References

Fornacon-Wood, I., Mistry, H., Johnson-Hart, C., Faivre-Finn, C., O'Connor, J.P. and Price, G.J., 2022. Understanding the differences between Bayesian and frequentist statistics. International journal of radiation oncology, biology, physics, 112(5), pp.1076-1082.