Bayes Regression using Python

Instructor: Arindam Chaudhuri

Indian Institute of Quantitative Finance

In order to implement Bayesian Regression we are using PyMC3 library. PyMC3 library works on top of Theano framework. So to make it run Theano needs to be installed first

Refer Python notebook attached

This is an example of noisy data around straight line considered

The regression is implemented using polyfit() method with NumPy library

It is to be noted that highest order monomial factor is at index level 0 and intercept is at index level 1

The original parameters are not perfectly recovered due to noise in data

In Bayesian regression is assumed that parameters are distributed in certain way

Consider regression line $\hat{y}(x) = \alpha + \beta \cdot x$

We now assume following priors:

α is normally distributed with mean 0 and standard deviation 20

β is normally distributed with mean 0 and standard deviation 20

For the likelihood, we assume a normal distribution with a mean of $\hat{y}(x)$ and a

In Bayesian regression is assumed that parameters are distributed in certain way

Consider regression line $\hat{y}(x) = \alpha + \beta \cdot x$

We now assume following priors where

(a) α is normally distributed with mean 0 and standard deviation 20

(b) β is normally distributed with mean 0 and standard deviation 20

For likelihood normal distribution is assumed with mean ŷ(x) and uniformly distributed standard deviation in 0 and 10

A major element of Bayesian regression is (Markov Chain) Monte Carlo (MCMC) sampling. In principle this is same as drawing balls multiple times from boxes

For technical sampling there are three different functions:

(a) find_MAP finds starting point for sampling algorithm by deriving local maximum posteriori point

(b) NUTS implements efficient No-U-Turn Sampler with dual averaging (NUTS) for MCMC sampling given assumed priors

(c) sample draws number of samples given starting value from find_MAP and optimal step size from NUTS