PyMC library

open source Probabilistic Programming framework

Sum Up

PyMC (formerly known as PyMC3) is a Python package for Bayesian statistical modeling and probabilistic machine learning which focuses on advanced Markov chain Monte Carlo and variational fitting algorithms.

Coding Bayesian statistical models the statistician way.

Installation / Import

- conda create -c conda-forge -n pymc_env "pymc>=4"
- conda activate pymc_env

>> import pymc as pm

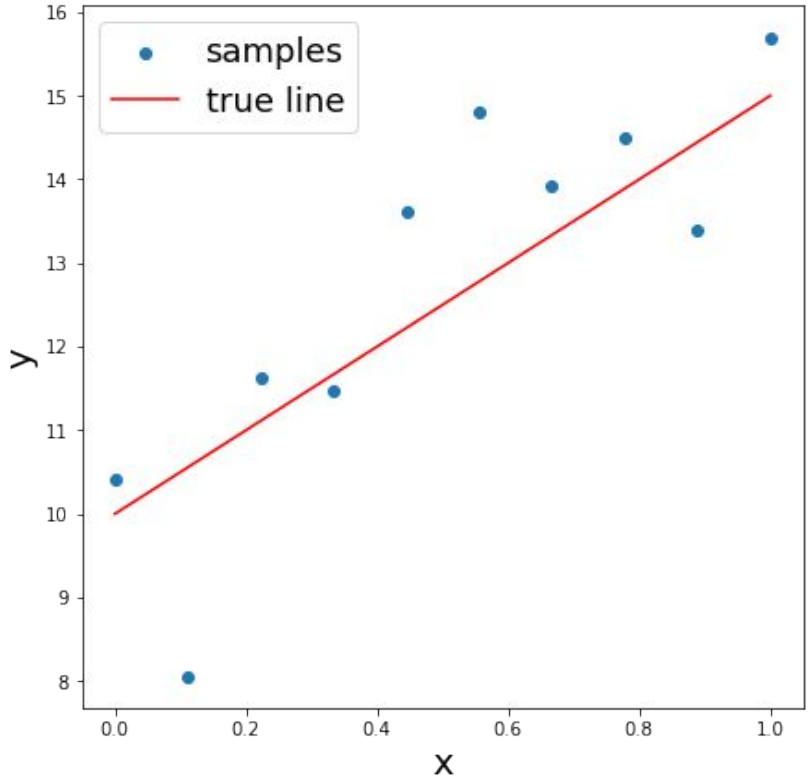
Usual way on a small dataset (10 points)

Case of linear regression:

y_true = mx + b

Fitting an ususal linear model:

y_obs = y_true + N(0,σ)



Sample distribution way

Let’s consider the following independant priors:

- m ~ N(0,20)
- b ~ N(0,20)
- σ ~ Exp(1)

Likelihood : y | m,b,σ ~ N(mx+b,σ)

Posterior : m,b,σ | y ~ **?**

Bayesian theorem gives us:
P(m,b,σ | y) ∝ P(y | m,b,σ) x P(m) x P(b) x P(σ)

PyMC will use priors and likelihood to compute a sample of the Posterior

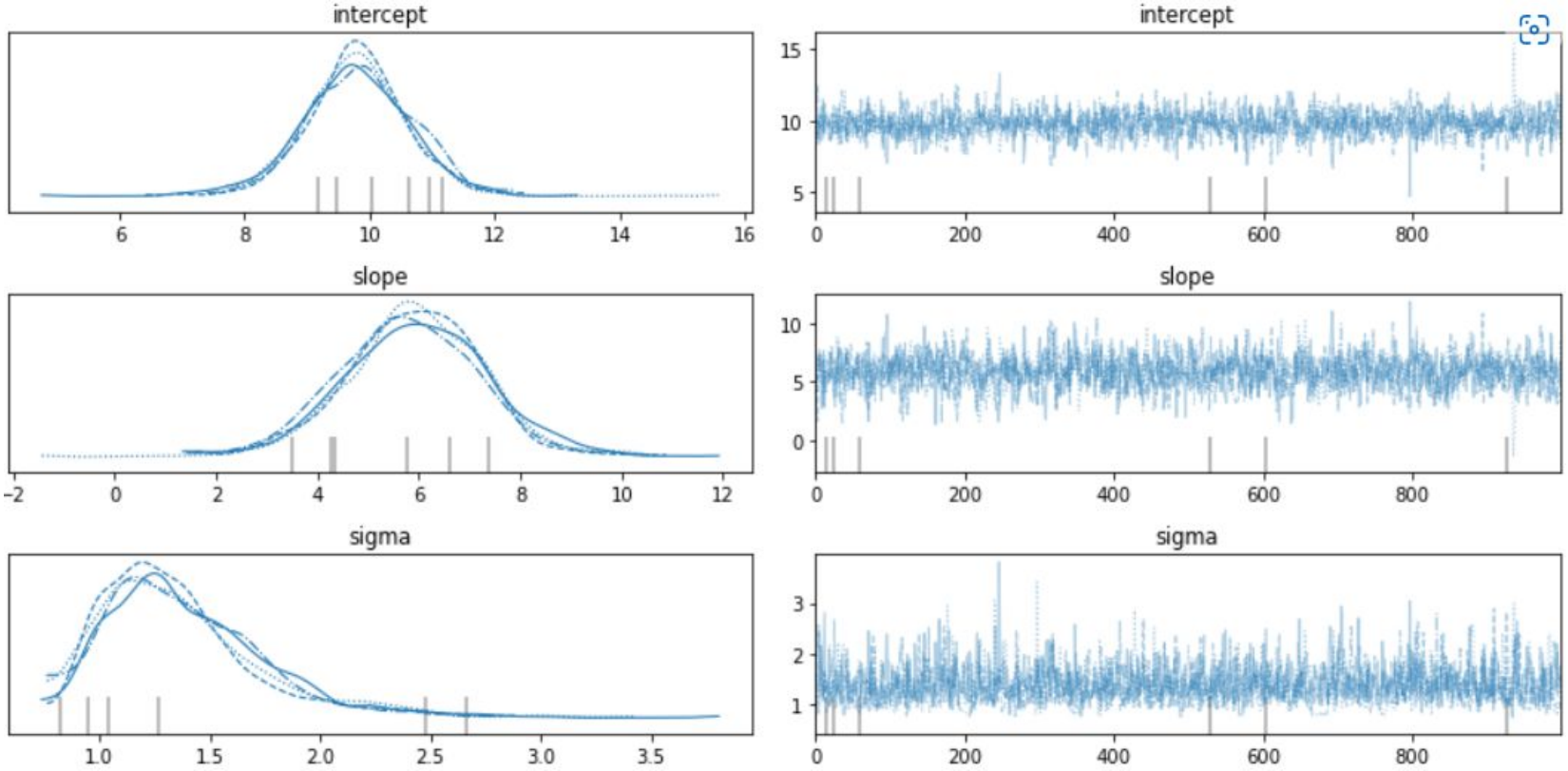
```
with pm.Model() as model:
    #priors
    sigma = pm.Exponential("sigma", lam=1.0)
    intercept = pm.Normal("intercept", mu=0, sigma=20)
    slope = pm.Normal("slope", mu=0, sigma=20)

    #Likelihood
    likelihood = pm.Normal("y", mu=slope*x_vals + intercept, sigma=sigma, observed=y_

    #posterior
    trace = pm.sample(1000, cores=4)
```

credits : https://www.youtube.com/watch?v=SP-sAAYvGT8

Result


intercept
6
8
10
12
14
16
intercept
15
10
5
0
200
400
600
800
slope
−2
0
2
4
6
8
10
12
slope
10
5
0
0
200
400
600
800
sigma
1.0
1.5
2.0
2.5
3.0
3.5
sigma
3
2
1
0
200
400
600
800