Coding and Testing a Markov Chain Monte Carlo Routine in Python*

Erich Wanzek

(Dated: December 9, 2017)

I. INTRODUCTION

I programmed a Markov Chain Monte Carlo routine in Python that is capable of performing MCMC modeling of data that is linear and sine based. The program has a very high acceptance rate, and converges most of the time. It is necessary to toy around with the step sizes of the Markov chain, the initial Markov chain parameters, iterations, and burn in period value, in order to have the posterior distributions converge to reasonable and likely parameter values, especially for more complex models.

II. TESTING MCMC WITH NOISY LINEAR DATA

To the test my MCMC routine, I ran the routine to fit linear function parameters to noisy linear data. In python i generated linear data with slope of 2 and y intercept of 3, i then injected noise into it to produce some synthesize noisy data. The original linear data can be seen in figure 1. The noisy data can be seen in figure 2.

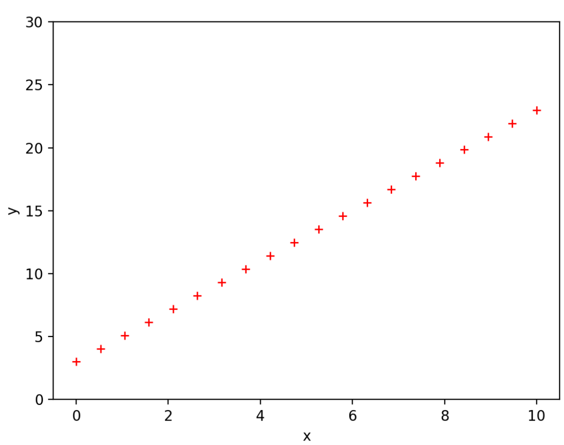

FIG. 1. Linear data

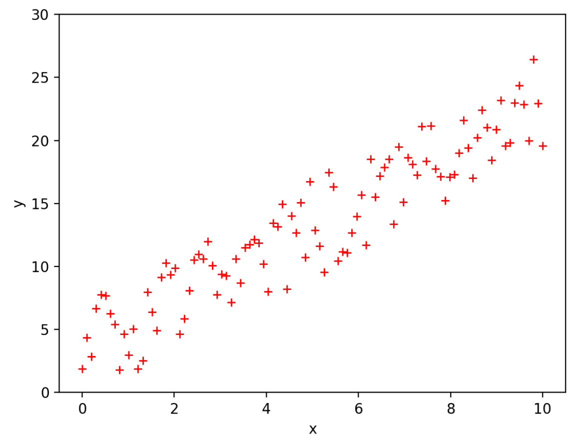

FIG. 2. Noise injected into linear data

I ran my MCMC routine for 100000 iterations. I initiated the Markov chain with initial parameters set bot to zero. I truncated the first 20000 values of the Markov chain to exclude the burn in values. A triangle plot of the posterior distributions an determined values for parameters m and b is shown in figure 3.

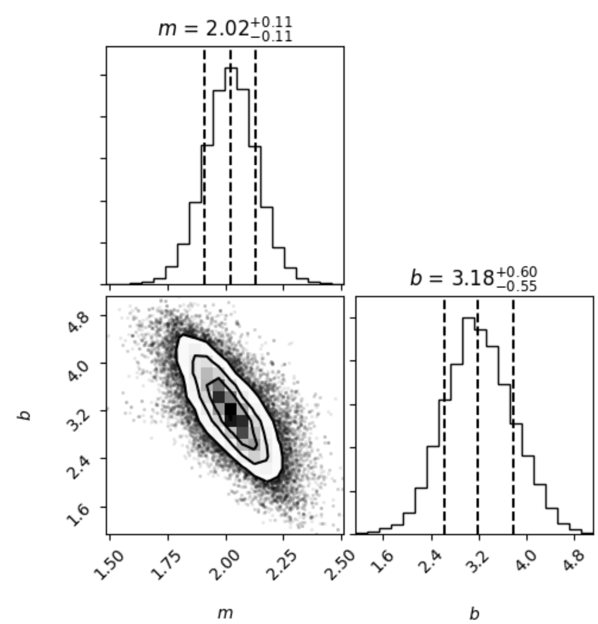

FIG. 3. Triangle plot of posterior distributions

The 2d parameter likelihood density plot shows the pa-

---

* A footnote to the article title

rameters ar covariant because the plot is stretched along one direction and shrunk along another and this signifies there is covariance along the eigenvectors of this plot. By analyzing the sigma values or spread of the posterior distributions, it is realized it corresponds to the level injected noise.

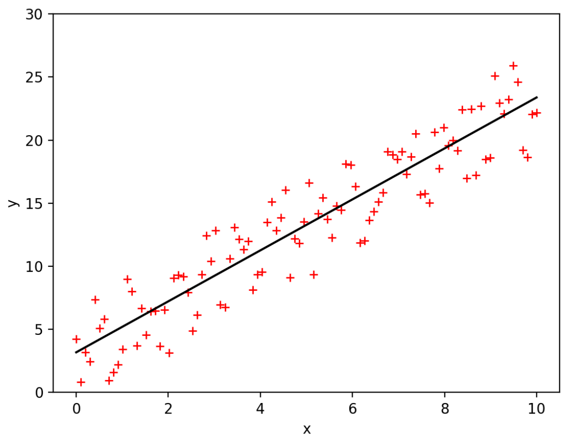

FIG. 4. Fitted model with the determined values to the noisy data.

III. USING MCMC TO FIT DATA

Next, I used my MCMC routine to fit a model of the form:

$$Y = A\sin(BX) + CX \tag{1}$$

to data provided. The original data can be seen in figure 5.

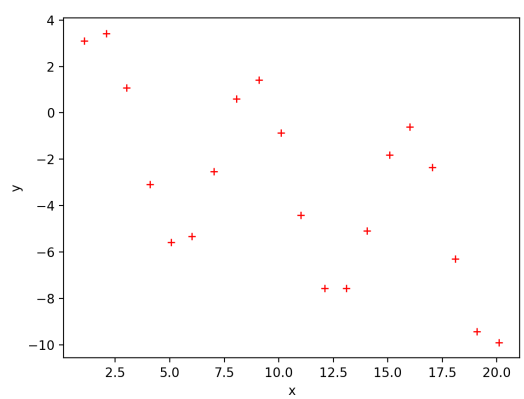

FIG. 5. Original data of the form of a sine function plus a linear function

I initiated the Markov chain at parameters of A=4, B=1, and C=-0.5 by performing a initial eyeball analysis of the data. I ran the MCMC routine for one million iterations. The posterior distributions and determined parameters can be seen in the corner plot of figure 6.

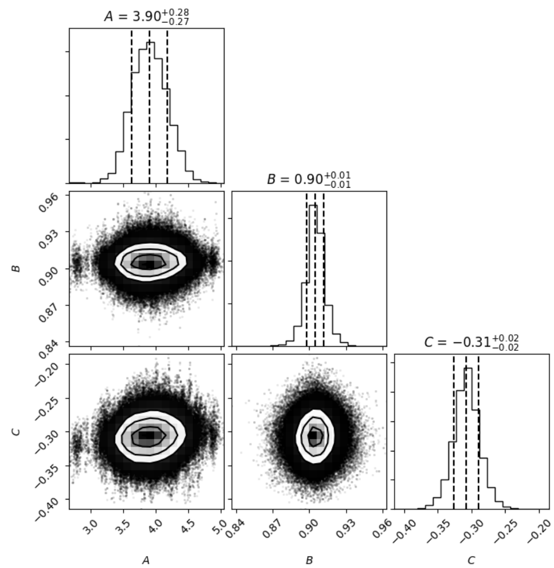

FIG. 6. Triangle plot of posterior distributions

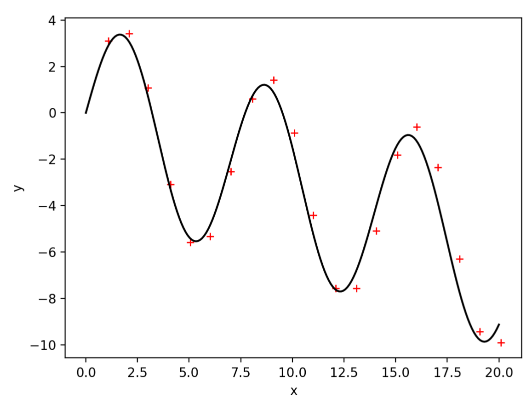

FIG. 7. Fitted model with the determined values to the data