# Monte Carlo parameter estimation in the Lorenz 63 system

Noah D. Brenowitz

May 12, 2015

# Outline

# Lorenz63 system

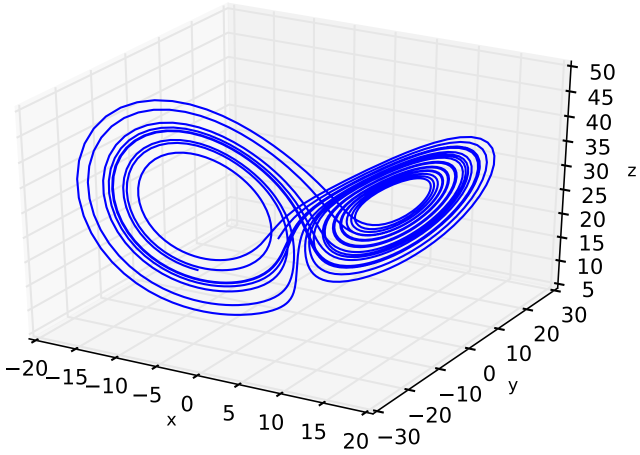
This system is one of the earliest examples of what would later become known as *chaotic* dynamical systems. It is obtained by projection of the Rayleigh-Benard equations for convection of a thin layer onto three orthogonal modes. The system is given by

$$\dot{X} = \sigma(Y - X) \tag{1}$$

$$\dot{Y} = -XZ + rX - Y \tag{2}$$

$$\dot{Z} = XY - bZ. \tag{3}$$

The parameters used by Lorenz are $\sigma = 10$, $b = 8/3$ and $r = 28$.



−20
−15
−10
−5
0
5
10
15
20
x
−30
−20
−10
0
10
20
30
y
5
10
15
20
25
30
35
40
45
50
z

# Notation

- Let $U$ be given by the dynamics

$$\dot{U} = F(U;\theta) + \epsilon \dot{W}$$

  where $\theta$ is some set of parameters.
- Let $\Psi_t^\theta$ be the *propagator*, defined by

$$\Psi_t^\theta U(s) = U(t+s).$$

- *Temporal grid*: Let $T > 0$, $N$, and define $\tau = T/(N+1)$. Then, a temporal grid $t_j = j\tau$ can be defined for $j = 0...N$.

# Dynamic Residuals

Given a set of point estimates $U_j$ and parameters, define the dynamic residual for a time $t_j$ to be

$$e_j(\theta) := e(U_{j-1}, U_j, \theta) := \Psi_\tau^\theta U_{j-1} - U_j.$$

Then, a sensible way to choose the parameters is so that they minimize

$$\sum_{j=1}^{N} |e_j|^2.$$

Because $\theta$ is a potentially high dimensional object, Monte Carlo techniques can be used to perform this optimization.

# Equilibrium distribution

- Vector of parameters $\theta = [\sigma, r, b]$
- Generate samples $\theta_k$ from

$$f(\theta) \propto \exp\left(-\frac{\beta}{N}\sum_{j=1}^{N}|e_j(\theta)|^2\right).$$

- The mode of $f(\theta)$ is the value that minimizes the dynamic residuals

# Proposal Distribution

- Independent gaussian proposal function:

$$P(\theta_0, \theta_1) \propto \exp\left(-\frac{1}{2}(\theta_1 - \theta_0)^T C^{-1}(\theta_1 - \theta_0)\right)$$

- Covariance matrix: $C_{11} = .5^2$, $C_{22} = .1^2$, $C_{33} = 1.0^2$, $C_{ij} = 0$ if $i \neq j$.

# Acceptance Probability

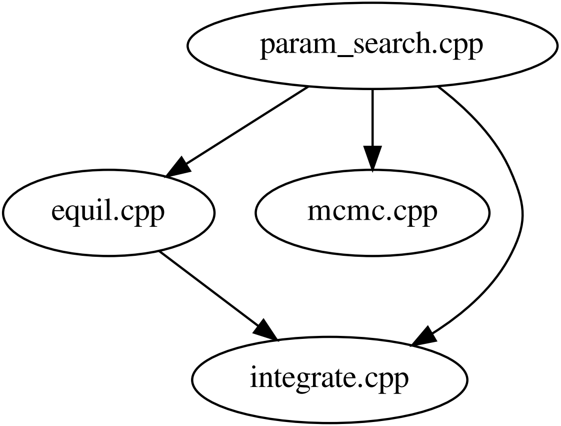
Metropolis-Hastings acceptance probability:

$$A(\theta_0, \theta_1) = \min\left\{1, \frac{f(\theta_1)P(\theta_1, \theta_0)}{f(\theta_0)P(\theta_0, \theta_1)}\right\}$$

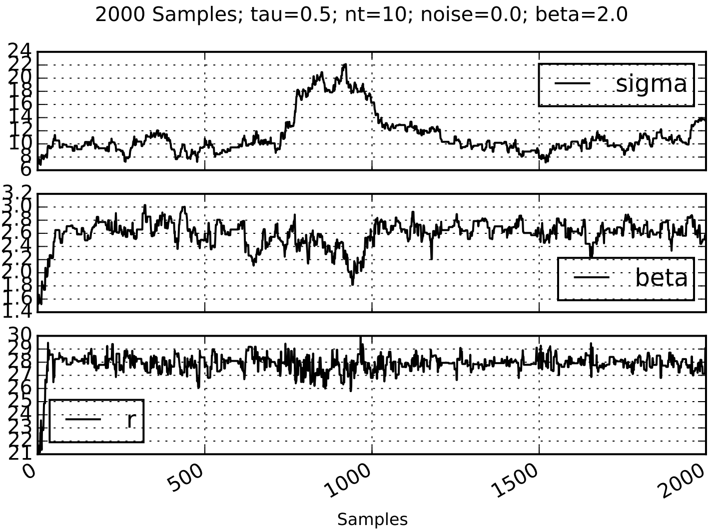
# Languages and Libraries

- Language: C++ for sampler, Python for plotting/analysis
- C++ Libraries:
  - Armadillo++ for convenient vector arithmetic
  - GNU Scientific Library for random number generation

# Code Architecture

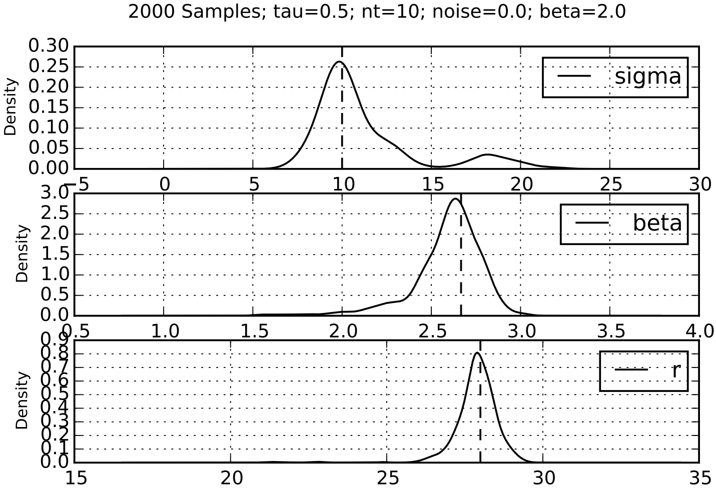


# $\tau = .5$ Traces

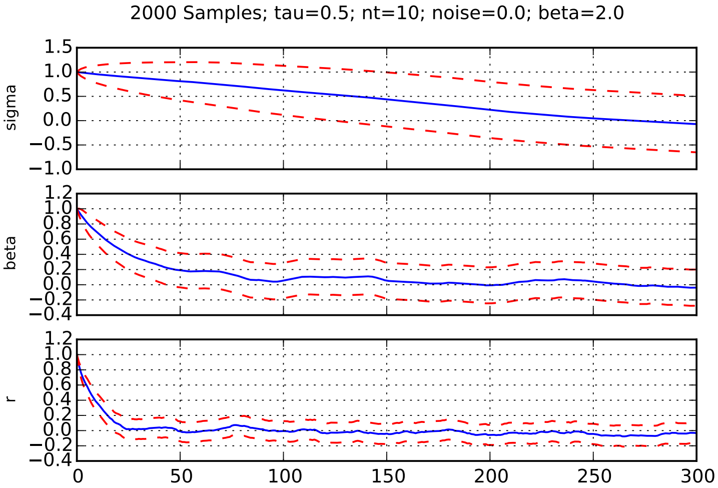


2000 Samples; tau=0.5; nt=10; noise=0.0; beta=2.0
sigma
beta
r
Samples
0
500
1000
1500
2000

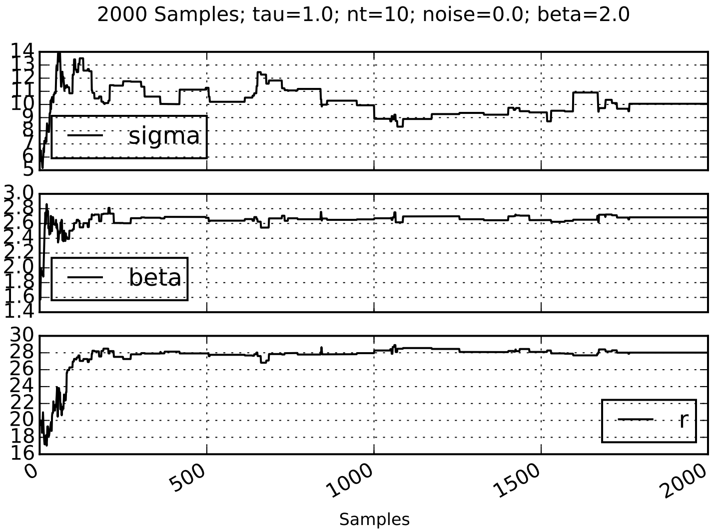
# $\tau = .5$ Distributions



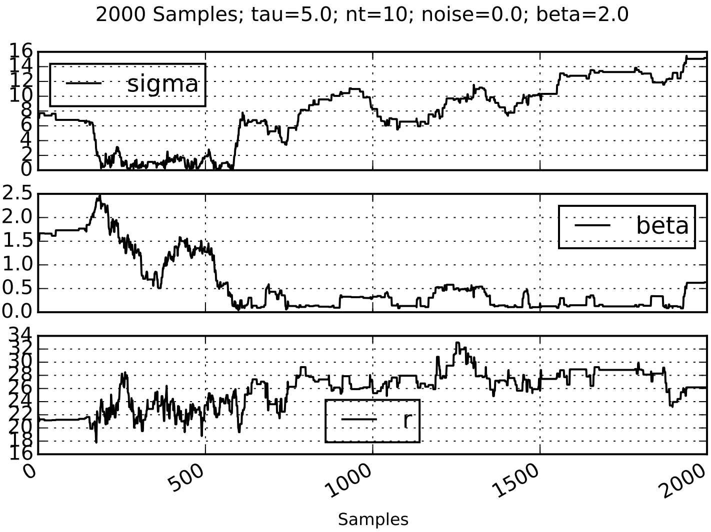
# $\tau = .5$ Autocorrelation functions



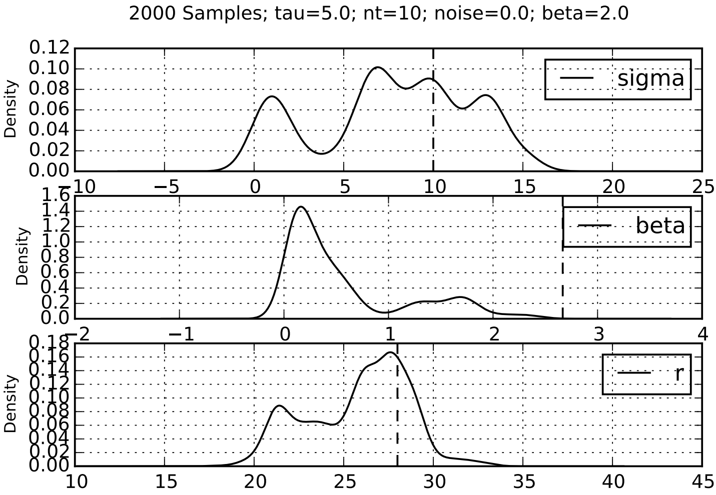
# $\tau = 1.0$ Traces



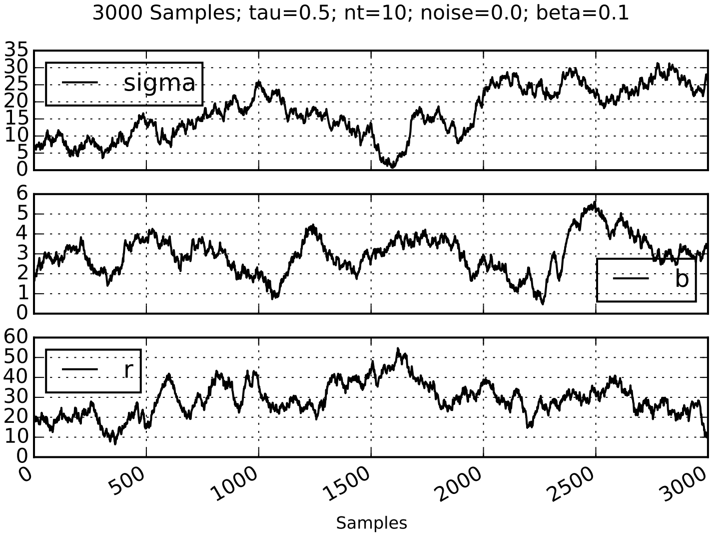
# $\tau = 5.0$ Traces



# $\tau = 5.0$ Distributions

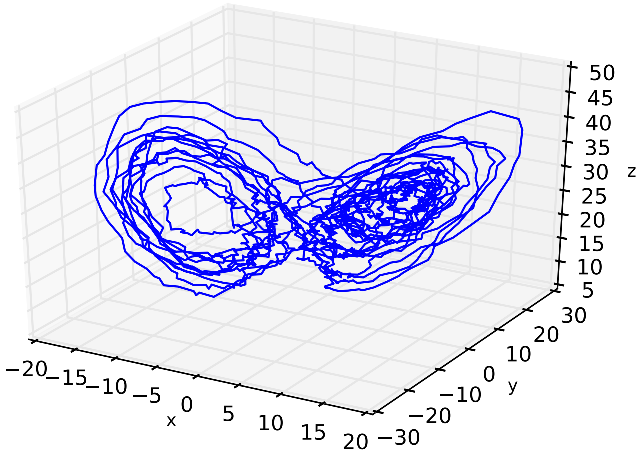


# Reducing $\beta$ gives large drift and autocorrelations

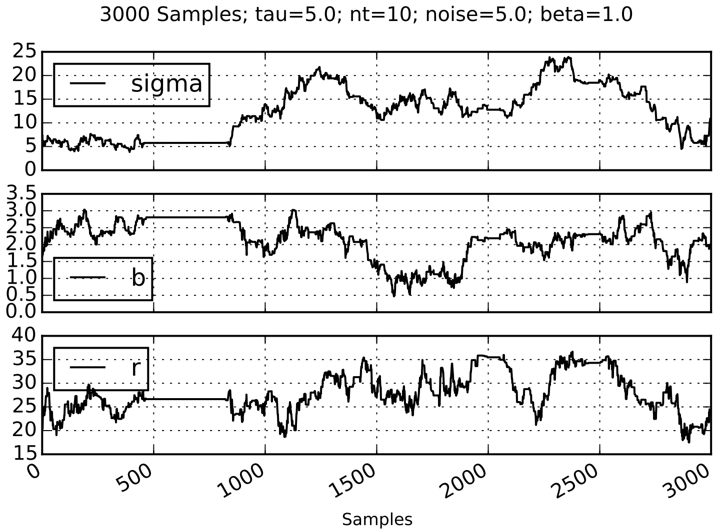


# Model Error can help

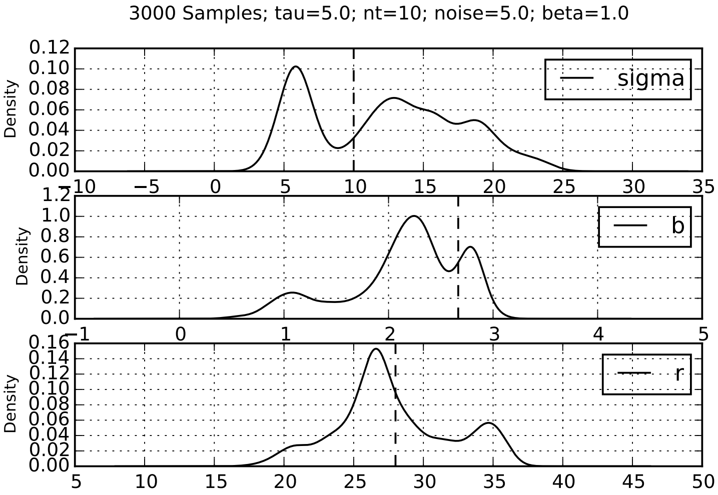
- Let $\epsilon = 5.0$



# Model error can help a little



# Model error can help a little



# Conclusions

- Using the dynamic residuals method to find parameters works well for short observations times $\tau \leq .5$.
- MCMC performance is very poor for larger $\tau$.
- Making a model error can help regularize the fit
- $\sigma$ is the most difficult parameter to estimate

# Future Directions

- Use dynamic residuals might work better for the data assimilation problem
- Parameter fitting might be better done by matching equilbrium *statistics* of the attractor rather than exact paths.