

Comments on Stochastic Simulation
Zoology 535, Spring 2008

Pseudo-random Numbers

Computers generate pseudorandom numbers, which are not truly random. Instead, they are sequences of numbers that appear random to standard statistical tests. Sequences of pseudorandom numbers are always periodic. Ideally, the period is so long that it does not matter for purposes of a particular modeling study.

R uses an algorithm (the "Mersenne Twister") with a cycle length of $2^{19937}-1$, which is more computations than your computer can perform in the expected lifetime of the universe.

Random number generators are initialized with a seed. For a given seed, exactly the same sequence of random numbers will be generated. By default, R takes its seed from the computer clock at the time the random number generator is called. Thus every run of a given program will most likely produce a different answer. Sometimes when you are debugging a program it is useful to generate exactly the same sequence of random numbers each time. To do this in R, use the command "set.seed(seed, kind = NULL)" where "seed" is a positive integer.

It is possible to generate pseudorandom numbers that "look like" particular statistical distributions (Ripley 1987). The distributions used in this course, and the associated R commands for random number generation, are listed below. See the html help for more information about each command.

Distribution	Random number command	Comments
uniform	runif(n, min=a, max=b)	Generate n random numbers for a flat distribution between a minimum value of a and maximum value of b. You must specify n. Default values for a and b are 0 and 1, respectively.
normal	rnorm(n, mean=a, sd=b)	Generate n random numbers for a normal distribution with mean = a and standard deviation = b. You must specify n. Default values for a and b are 0 and 1, respectively.
lognormal	rlnorm(n, meanlog = a, sdlog = b)	Generate n random numbers for a lognormal distribution with mean = a in the log units and standard deviation = b in the log units. You must specify n. Default values for a and b are 0 and 1, respectively.

Some distributions that were not used in this course but are fairly common in ecology can also be simulated with R. For completeness, these are listed here without further explanation. See the html help for more information if you want to use these commands.

Binomial: rbinom(n, size, prob)

Poisson: `rpois(n, lambda)`
 Negative binomial: `rnbinom(n, size, prob, mu)`
 Chi Square: `rchisq(n, df, ncp=0)`
 Exponential: `rexp(n, rate = 1)`
 Gamma: `rgamma(n, shape, rate = 1, scale = 1/rate)`
 Multivariate normal: `mvrnorm(n = 1, mu, Sigma, tol = 1e-6, empirical = FALSE)`
 Student-t: `rt(n, df)`

Process and Observation Error

Earlier handouts discuss models that employ process error and observation error. We assume that nature generates ecosystem dynamics using an unknown model (which may include stochastic terms). Process error results from the scientist using a different model from nature's model, plus errors that result from any stochastic terms in nature's model. Observation error is error that results from noisy measurements of ecosystem dynamics. Using equations, these errors are described as follows.

$$X_R = f(X_P, \theta) + \varepsilon \text{ and } X_{P+R} = \mu + \omega$$

The first equation is the process equation. X_R is the response variables, X_P is the predictor variables, θ is the parameters, and ε is the process errors. Note that the predictor variables could include one or more of the response variables measured at a previous time. The second equation is the observation equation. X_{P+R} represents both the response and predictor variables, μ is the true but unknown values of these variables, and ω is the observation error. Both ε and ω are modeled as random variables (often normally-distributed ones with mean 0 and standard deviations σ_ε and σ_ω respectively). This model can be simulated using the random number generators described above.

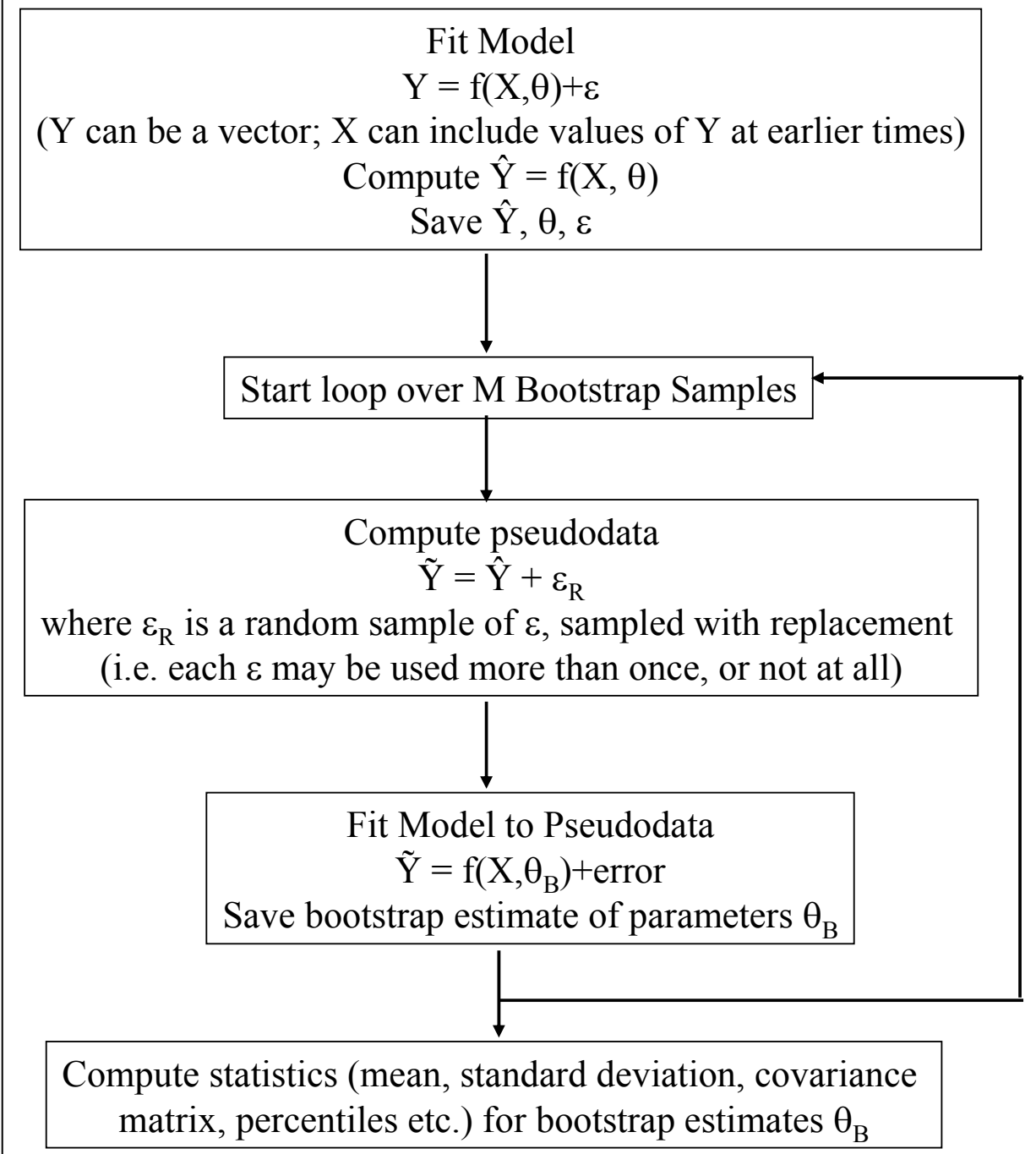
Bootstrapping

Bootstrapping is a method of estimating standard deviations, covariances, and percentiles for parameters of models, by sampling from the data (Efron and Tibshirani 1993). Bootstrapping can also be used to estimate statistics for model predictions. In nonparametric bootstrapping, samples are drawn from the data themselves (see figure). In parametric bootstrapping, a distribution (such as the normal or lognormal distribution) is fitted to the information that will be sampled randomly (the residuals, in the case shown in the figure). Then the random samples are drawn from the fitted distribution.

In R, there is a handy function for bootstrapping. To create a random sample with replacement, called `Xrandomized`, from data vector `X`,

```
nX <- length(X)
Xrandomized <- sample(X, nX, replace=TRUE)
```

Bootstrap Estimates of Model Statistics: Flow Chart



Stochastic Simulation in Continuous Time

It is sometimes useful to consider noise in continuous-time ecological models. In such models, Brownian motion is often used to model the noise. The Wiener process is one way to model Brownian motion (Horsthemke and Lefever 1984). A Wiener random sequence W_t has mean zero. Over a discrete time interval of length h , $(W_{t+h} - W_t)/h$ is a random variable that is normally distributed with mean 0 and variance $1/h$.

As a simple example, suppose we have some ecological model written as a differential equation

$$\frac{dX}{dt} = f(X)$$

where X is the ecological variable and f is a function for the net rate of change (inputs and losses, with appropriate mathematical expressions and parameters). To add noise, write

$$\frac{dX}{dt} = f(X) + \sigma \frac{dW}{dt}$$

where σ is the standard deviation we want for the noise. Multiply both sides by dt to get

$$dX = f(X)dt + \sigma dW$$

This is called a stochastic differential equation (SDE). There are two kinds of calculus for this sort of problem, Ito and Stratonovich (see Chapter 5 of Horsthemke and Lefever 1984 for more explanation). For ecological applications, it usually does not matter which calculus you choose. Ito calculus is slightly easier to use. Using Ito calculus, we can convert the SDE to an equation that can be solved using the computer over a small increment of time h :

$$X_{t+h} = X_t + f(X_t)h + \sigma Z_t \sqrt{h}$$

Here the Z_t are normally-distributed random numbers with mean 0 and variance 1. Notice this looks like the Euler method for solving differential equations, except that the noise increments are multiplied by the square root of h (not by h). This ensures that the variance turns out to be σ^2 as desired.

In the example above, we added process noise to a differential equation model. This is a way to represent high-frequency shocks to an underlying ecological process. Another use of SDEs is to study the effects of high-frequency shocks to particular parameters. To illustrate this point, consider a commonly-used model for biomass of exploited fish stocks, the Pella-Tomlinson model (Hilborn and Walters 1992):

$$\frac{dB}{dt} = rB - \frac{r}{k_o} B^c - qEB$$

In this model, B is the fish biomass, r is the reproduction parameter, k_0 is a parameter related to the biomass of the stock if there was no fishing ("carrying capacity"), c is a shape parameter that allows the biomass dynamics to be asymmetric, q ("catchability") is a parameter for the effectiveness of fishing in capturing fish, and E is total effort by the fishing fleet.

Suppose we believe that the carrying-capacity parameter fluctuates stochastically, due to changes in food supply that we cannot easily monitor. To model these fluctuations, we might break k_0 into deterministic and stochastic components: $1/k_0 = (1/k) + \sigma$. Now we can write the SDE

$$dB = rB - \frac{r}{k} B^c - qEB + \sigma B^c dW$$

which can be solved on the computer over small steps of time h using

$$B_{t+h} = B_t + [rB_t - \frac{r}{k} B_t^c - qEB_t]h + \sigma B_t^c Z_t \sqrt{h}$$

where as above Z_t is a sequence of normally-distributed random numbers with mean 0 and variance 1.

Particle Paths: Individual Based Models

Just as equilibria are used to understand deterministic models, stationary distributions are used to understand stochastic models. Suppose we run a large population of "replicates" of a stochastic model. At each point in time, the result is a probability distribution for the state variables. In the limit as time $\rightarrow \infty$, the distribution may approach a stationary distribution. In a rough sense, the stationary distribution represents a balance between the stabilizing forces in the model (which tend to pull the system toward stable attractors) and the stochastic forces in the model (which tend to spread the solutions away from attractors).

The stationary distribution can be computed using equations described in Chapter 6 of Horsthemke and Lefever (1984).

Alternatively, Monte Carlo methods can be used to study the stationary distribution. Using the computer, a large number of replicate simulations are run, starting from a wide range of initial conditions. If there is a stationary distribution, the distribution of replicate simulations will approach it over time. Often it is possible to get a reasonable picture of the stationary distribution with replicate runs lasting around 3X the residence time of the slowest state variable in the model.

In ecology, the approach of simulating a large number of replicates of a stochastic model is sometimes called "individual-based modeling" or IBM (Grimm 1999). The idea is to build a stochastic model for behavior of an individual organism, or individuals of a few interacting species, or "patches" of a landscape. In addition to these models of individual units, the IBM includes rules for interaction of individuals, such as competition, predation, or spread of

something (disease, fire, propagules etc.) among patches. Advantages of this approach to ecosystem modeling are discussed by DeAngelis and Mooij (2003).

References

DeAngelis, D.L. and W.M. Mooij. 2003. In praise of mechanistically-rich models. Pp. 63-82 in Canham, C.D., J.J. Cole and W.K. Lauenroth (eds), *Models in Ecosystem Science*. Princeton University Press, Princeton, N.J.

Efron, B. and R.J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, NY.

Grimm, V. 1999. Ten years of individual-based modeling in ecology: What have we learned and what could we learn in the future? *Ecological Modelling* 115: 129-148.

Hilborn, R. and C.J. Walters. 1992. *Quantitative Fisheries Stock Assessment*. Chapman and Hall, N.Y.

Horsthemke, W. and R. Lefever. 1984. *Noise-Induced Transitions*. Springer-Verlag, NY.

Ripley, B.D. 1987. *Stochastic Simulation*. Wiley, N.Y.