Monte Carlo Simulation

(Follows FW663's lecture quite closely)

Monte Carlo simulation is useful for understanding the properties of a model, either under the assumptions of the model, or under other assumptions (i.e., under a different model). In addition, such simulation can be useful in the design of studies or experiments. In this vein, it can be used to investigate model-selection and thus, which effects can be detected under different sampling scenarios. For example, one might evaluate how often an effect is detected in simulated data sets as a function of (a) effect size (simulations conducted where the effect is small, medium, and large, e.g., effect of body mass on survival) or (b) sampling effort (*n*) or sampling scheme (# of occasions), etc.

Program MARK has some nice tools that are still being developed. Program RELEASE (see Burnham et al. 1987, Part 9) is very useful in the context of open Capture-recapture models because if offers a range of options that are easy to use. Program CAPTURE has useful tools for evaluating estimator performance competing estimators for data from closed populations.

The idea underlying simulated data is fairly simple. Monte Carlo simulation produces a set random variables based on known values for distributions and parameters in the model. The procedure produces "simulated" data where the generating model and its parameters are known. (SimExmpl.pdf provides a simple example of what might take place in a closed-captures simulation of M(t)).

In generating simulated data, one usually fixes the binomial parameters in the model to a numerical value (e.g., say 1/6 = 0.166666...). Then, one draws values or variates from a uniform distribution from zero to one (i.e., U(0,1)). If the value of this variate is $\leq$0.166666, then a 1 is recorded for the Bernoulli trial represented by this variate; otherwise a 0 is recorded. Thus, you essentially create a coin with a given probability (e.g., 0.166666… or whatever you desire) for obtaining a head on each coin toss and "toss" the coin and record the outcome (where the tosses and outcomes are determined using the uniform random number and the cutoff value). The procedure is carried out *n* times, each time drawing a new variate *y* from a U(0,1) distribution, checking to see if it is $\leq$ the parameter in question (in this case 1/6), and recording a 1 if the condition is satisfied. Such computations can be done by hand if the model is as simple as die throwing.

We will start by working with Closed Capture Simulations and lab 12 will help you explore these in greater detail.

We may also consider simulation in future weeks for CJS models. For these models, the approach used in Program RELEASE and Program MARK is to model the fate of an individual animal. Suppose the animal is released on occasion 1, i.e., it is part of R_1. The first event is whether the animal survives the

first interval, i.e., whether it survives to occasion 2. The probability that the animal dies during interval 1 is $1-\varphi$. The probability that it survives interval 1 is φ. To create data, a random uniform variate is generated. If this value is greater than φ, then the animal died, resulting in the encounter history {1000000}. If the random variate is less than φ, then the animal lived. So now, the animal is available for capture on occasion 2. The probability of capture is p_2. A new : # random variate is generated. If the value is less than p_2., then the animal was captured. The p_2. encounter history would now have the value {11} at the start, indicating this capture. The process is continued until the animal either dies, giving its encounter history, or else it survives through all intervals, also giving its encounter history. Each animal in the study is simulated in the same way, to give the final set of encounter histories.

The approach used in Programs RELEASE and MARK allows for individual heterogeneity, age-specific effects, and thus provides a very general simulation capability.

Using thousands of draws from a uniform random number generator, one can generate a simulated data set where the model and its underlying parameters are known. Such Monte Carlo data sets can then be used for a wide variety of purposes. Typically, 1000 simulated data sets are used for many studies, however, 10,000 is not uncommon and is often required for studies of achieved confidence interval coverage. Each simulated data set is different, just as the outcomes (the y_i) after n throws of a die would be different.

Some reasons for conducting Monte Carlo studies:

<u>Data Have Been Generated under Model A and Studied under the Same Model:</u>

1. Small sample bias in the MLEs of parameters, the $\hat{\theta}$
2. Compare SD of the MLEs from the 1,000 reps vs. the average $s\hat{e}(\hat{\theta})$
3. Confidence interval width and achieved coverage.

<u>Data Have Been Generated under Model A and Studied under a Different Model:</u>

1. Model bias in the MLEs of parameters $\hat{\theta}$ (robustness?)
2. Compare SD of the MLEs from the 1000 reps vs. the average $s\hat{e}(\hat{\theta})$
3. Confidence interval width and achieved coverage.
4. Model selection issues

<u>Design of Studies and Experiments</u>

1. What is the effect of doubling the number released or precision?
2. What is the effect of increasing the duration of the study (k) on precision?
3. What is the effect of increasing p on precision?
4. What are the effects of increasing the level of the treatment?

Using Expected Values as “Data"

A shortcut procedure can often be used to obtain quick and dirty insights into some of the above issues without running 1000 or 10,000 simulations. In this procedure, one computes (only once) the expectations of each encounter history or $m_{i,j}$ cell and rounds to the nearest integer. These values are then used as if they were “data." These values are not random variables.

Running such expectations through an estimation routine (e.g., *MARK*) gives rough ideas about model bias, standard errors, and confidence-interval width. This information is very useful in design of studies and formal experiments. For example, this simple method will give quick insights into the relation between the p_i and precision of some $\hat{\theta}$, effects of increasing study duration or precision, relative allocation of effort, etc.

Once the design has been fine-tuned one may often want to go to some Monte Carlo simulations to obtain confidence interval coverage and comparisons between the standard deviation of the estimates over the reps vs. the average model-based standard error of the estimates.