Hour 8: Simulation

Although simulation is not directly related to the event history analysis, it is a useful tool to diagnose the model parameters, to test model formulation, and to do post estimation inferences. We can use simulation, and Monte Carlo style studies to investigate the validity of model assumptions, e.g. test duration dependence, or to compare different methods in control the unobserved heterogeneity.

Recently the non-parametric maximum likelihood method becomes popular in estimation of hazard rate models. With simulation and Monte Carlo study we can uncover many properties of NPMLE. Especially since the asymptotic properties of NPMLE are still unknown, one thus can use simulation and bootstrap method to get estimation errors for inference.

Also, the interpretation of estimates from the hazard rate formulation is not intuitive. One needs to construct proper and real world related statistics to show the results from estimation. for example, if we have estimated a hazard rate model for unemployment duration, and wish to express the causal effect of program participation during unemployment spells. We could report that participation to program increase the hazard rate to job by ##%, but for policy maker and public, hazard rate is not easy to communicate. In that case, we can by simulating the unemployment duration with and without estimated program effect, to demonstrate how the spell length can be reduced in average.

Random number drawing in Stata

In Stata (or equivalently in any statistics programs), all the random number drawing is based on the uniform() function. The uniform() draws a random number from (0,1), based on default or user supplied random seed. The drawn random number can then be manipulated to simulate numbers from different statistical distribution, or other tasks, such as bootstrap draw.

Let’s look at some examples:

1. Open Stata in interactive mode
2. **set mem 50m**
3. **set obs 100** /* create a dataset with 100 observations and no variable for the time-being. */
4. **gen x1=uniform()**
   we can do a **summarize** to check the variable x1 now.
5. if we wish to draw a random number from a standard normal distribution, we can do this:
   gen x2=invnormal(uniform())
6. if we wish to draw a random number from a normal distribution with $N(\mu,\sigma^2)$, we can rescale the drawn from invnormal function.

gen x3=1+sigma*invnormal(uniform()) /*mu=1, sigma=0.5, variance is thus 0.25*/
do a **summarize x3, detail**, to see the distribution of x3. you can see that this seems to be a normal distributed variable with mean 1, variance 0.25.

7. you can draw normal density based on the x3, and plot the density against x3.
   gen p3=normalden(x3,1,0.5)
   sort x3 /* for create graph connected, such that x3 are connected in ascending order*/
   graph connected p3 x3

8. draw a number from Gamma distribution.

Distribution of Gamma:

$$f(v)=\begin{cases}\dfrac{\lambda^n v^{n-1}e^{-\lambda v}}{\Gamma(n)}, & v>0\\ 0, & v\le 0\end{cases}, \quad n,\lambda>0$$

mean: $E(v)=\dfrac{n}{\lambda}$

variance $\operatorname{var}(v)=\dfrac{n}{\lambda^2}$

n is the shape parameter, λ is the scale parameter.

For n=1 Gamma distribution reduces to exponential distribution. Therefore choose a standard Gamma with shape, scale

$$f(v)=\begin{cases}\dfrac{v^{n-1}e^{-v}}{\Gamma(n)}, & v>0\\ 0, & v\le 0\end{cases}, \quad n,\lambda=1$$

Suppose we wish to draw a number x from Gamma distribution with $E(x)=1$, from formula above, we get immediately

$$E(x)=\frac{n}{\lambda}=1\Rightarrow\ n=\lambda;\ \ \Rightarrow Var(x)=\frac{1}{\lambda}$$

suppose we set variance of Gamma distribution to 0.6475 (I made this in the similar way as in Zhang(2003)). We then get the value for $n=\lambda=\dfrac{1}{0.6475}\approx 1.544$.

In Stata, we have standard gamma density function gammap(a, b), and Gamma density function gammaden(a,b,g,x) with shape parameter a, scale parameter b, and location parameter g. In fact, gammaden(a,b,g,x)=gammap(a,(x-g)/b).

so x=invgammap(a,p) should be equivalent as x=b*invgammap(a,p)+g. set g=0,

$a = \frac{1}{b} = \frac{1}{\theta}$

θ is the gamma parameter reported by Stata, and all this simply means $E(x) = 1, Var(x) = b = \theta$.

```
gen x4=0.6475*invgammap(1/0.6475, uniform())
sum x4, detail
```

Simulate single risk continuous time duration data

Recall the hazard rate function for transition to *k* is:

(1) $$\theta(t_k) = \tilde{\theta}(t_k) \cdot \psi(x_k) \cdot \upsilon_k = \tilde{\theta}(t_k) \cdot \exp\left(x_k{}'\beta_k + \ln(\upsilon_k)\right) = \tilde{\theta}(t_k) \cdot \exp\left(x_k{}'\beta_k + v_k\right)$$

The survival function is

(2) $$S(x_k, t_k) = \exp(-\psi(x_k)\upsilon_k \int_0^t \tilde{\theta}(s_k) ds_k)$$

To simulate the continuous duration, we must take the inverse of survival function, which can be derived from the solution of (taking $S(x_k, t_k) = 1 - u_k$)

(3) $$\int_0^t \tilde{\theta}(s_k) ds_k = -\frac{\ln(1-u_k)}{\psi(x_k)\upsilon_k}$$

where u_k is uniform distributed random variable.

Assume weibull baseline hazard

(5) $\tilde{\theta}(t_k) = \lambda^{\alpha} \alpha \tau^{\alpha - 1}$, where α is the shape parameter, and λ is the scale parameter.

we get

$$\int_0^t \theta(s_k)ds_k = (\lambda t)^\alpha = -\frac{\ln(1-u_k)}{\psi(x_k)\upsilon_k}$$

(6) $$\Rightarrow t^\alpha = -\frac{\ln(1-u_k)}{\lambda^\alpha\psi(x_k)\upsilon_k} \Rightarrow \alpha\ln t = \ln\left(-\frac{\ln(1-u_k)}{\lambda^\alpha\psi(x_k)\upsilon_k}\right)$$

$$t = \exp\left(\frac{1}{\alpha}\ln\left(-\frac{\ln(1-u_k)}{\lambda^\alpha\psi(x_k)\upsilon_k}\right)\right) = \left(-\frac{\ln(1-u_k)}{\lambda^\alpha\psi(x_k)\upsilon_k}\right)^{1/\alpha} = \frac{\left(-\ln(1-u_k)\right)^{1/\alpha}}{\lambda\left(\exp(x_k{}'\beta_k + v_k)\right)^{1/\alpha}}$$

a possible choice for shape and scale parameters can be $\lambda = 0.10, \alpha = 0.90$.

The continous_weibull.dta is simulated with scale parameter set to 1.

Let’s have a look of the simulate_weibull.do file to see how to implement these formulas in Stata, both with duration generation and Gamma distributed unobserved heterogeneity simulation.

simulate_weibull_do

run
stset t, failure(d) id(id)
streg x y, d(w) frailty(gamma)
to check the recovery of DGP parameters

Simulate single risk discrete time duration data

If we are willing to assume that the discrete time duration data arises from a underlying continuous time process, and by interval censoring to form a grouped hazard framework, we can easily simulate the discrete time duration data which can be used to estimate by cloglog.

We need to set a piecewise constant baseline from the continuous time duration baseline chosen. IN this example, a continuous time Weibull distribution baseline is chosen.

Recall that the weibull hazard rate for a given interval [d-1,d] is simply the definite integral of continuous Weibull baseline

$$b(t_d) = \ln\left(\int_{d-1}^{d}\alpha t^{\alpha-1}dt\right) = \ln\left(t^\alpha\Big|_{a-1}^{d}\right),\ where\ \alpha = 0.9,\ \lambda{=}1,\ d \in [1,12]$$

with this we can easily calculate (in excel for example) the piecewise constant duration baseline:

Period	Baseline b(t)
1	0
2	-0.14379
3	-0.19625
4	-0.23026
5	-0.25554
6	-0.27568
7	-0.29243
8	-0.30677
9	-0.3193
10	-0.33044
11	-0.34046
12	-0.34956

With this we can based on the complementary loglog function:

$$p(t-1<T<t \mid T>t-1) = 1-\exp\left(-\exp(x_t'\beta + b(t) + \ln(\lambda^{\alpha}))\right)$$

where the scale parameter is:

$$\ln(\lambda^{\alpha}) = \text{-2.072326584 when } \alpha = 0.9,\ \lambda = 0.1$$

simulate the discrete time duration data with underlying Weibull baseline. We can also simulate with the Gamma distributed unobserved heterogeneity.

simulate_discrete_weibull_with_uobs.do

Simulate independent continuous time competing risks duration data

To simulate a dataset with independent continuous time competing risks duration data is a simple matter of extension of single risk data simulation. The only difference is that one needs to simulate different underlying continuous time durations with the inverse function to survival function technique. Then the shortest underlying duration with transition should be the observed duration time and observed transition.

To put in compact form:

1. simulate $t_1 = \dfrac{\left(-\ln(1-u_1)\right)^{1/\alpha_1}}{\lambda_1\left(\exp(x_1'\beta_1+v_1)\right)^{1/\alpha_1}},\ t_2 = \dfrac{\left(-\ln(1-u_2)\right)^{1/\alpha_2}}{\lambda_2\left(\exp(x_2'\beta_2+v_2)\right)^{1/\alpha_2}}$
2. $t = \min(t_1, t_2),\ d = \begin{cases} 1,\ if\ \ t_1 \le t_2 \\ 2,\ if\ \ t_1 > t_2 \end{cases}$

simulate_competing_risk_weibull.do

Simulate independent discrete time competing risks duration data

The simulation with independent discrete time competing risks duration data is different than to simulate continuous time data. Recall that the discrete time grouped hazard model for competing risks is given by:

$$p(t-1<T<t\,|\,T>t-1,K=k)$$
$$=\underbrace{\left(1-\exp\left(-\sum_k h_{kt}\right)\right)}_{\text{Probability that one event occurs}}\times\underbrace{\frac{h_{kt}}{\sum_k h_{kt}}}_{\substack{\text{Conditional probability that}\\\text{the event is of type k}}}$$

this is the event probability for give time interval $[t\text{ -}1,t]$. So the simulation is actually done in two steps:

Step 1: simulate the transition probability that one event occurs.
Step 2: given that one event occurs, simulate the probability that the event is k.

simulate_competing_risk_discrete_weibull.do

run exercise6-2.do file again to check with your simulated data.

Above are examples showing basic ways to simulate some familiar duration data. Many advanced type data, with e.g. bivariate distributed unobserved heterogeneities in competing risks models, with timing-to-event type endogenous transitions can be simulated based on these examples, with much creative improvements of course.

Simulation techniques can also be used after estimation of duration models. One can simulate durations based on estimated coefficients for the hazard rate models, to do model calibrations. Or one can do bootstrap style analysis for the interesting model parameters, especially for the NPMLE estimators, when the asymptotic distributional properties are unclear or unknown.