Lecture 10: Logistic growth models

Fugo Takasu
Dept. Information and Computer Sciences
Nara Women's University
takasu@ics.nara-wu.ac.jp

28 June 2006

1 A deterministic model of logistic growth

In the birth and death process we assumed that the birth and the death rate per unit time were constant (β and δ) both of which were independent of the population size. In most realistic population dynamics, however, they are likely **density-dependent**, e.g., the birth rate decreases as the population size increases due to unfavorable environment caused by overcrowding and the death rate increases as the population size increases because of environmental deterioration or shortage of food resources. This density dependent effect should set an upper limit to the population size. Exponential growth as we modeled in previous lectures cannot last forever in the real world. We now model a deterministic version of such limited population growth with density dependency.

Let N be the population size as density and $birth(N)$ and $death(N)$ be the per-capita birth and death rate per unit time as a function of the population size N, respectively. Then the change of the population size during a time interval Δt is given in general as

$$\Delta N = N(t + \Delta t) - N(t) = birth(N)\Delta t N - death(N)\Delta t N$$

Arranging this equation and letting $\Delta t \to 0$ yields an ordinary differential equation

$$\frac{dN}{dt} = \{birth(N) - death(N)\} N \tag{1}$$

$birth(N) - death(N)$ is the net per-capita increase rate per unit time. If it is given as a linearly decreasing function of N, e.g.,

$$birth(N) - death(N) = r\left(1 - \frac{N}{K}\right)$$

equation (1) is called **logistic growth model** of continuous time. Parameter r is called **intrinsic growth rate** and K as **carrying capacity** ($r, K > 0$). We will later see why these two parameters are so called.

Let us first take a look at general properties of the logistic growth model.

$$\frac{dN}{dt} = r\left(1 - \frac{N}{K}\right)N \tag{2}$$

This ODE is non-linear (the right hand side is quadratic of N) and non-linear differential equations in general are not necessarily analytically tractable. Fortunately we can solve this simple non-linear ODE (See Appendix) but the behavior of the solution can be heuristically explored as follows.

From equation (2) we see that the time derivative of N, dN/dt, is positive when N is close to zero but positive. This means that N is increasing at this position as time advances. This increase continues as long as $N < K$. We thus expect N approaches K upward. On the other hand, when N lies at a position above K, the time derivative is negative and N is decreasing as time passes. We expect N also approaches K downward. This consideration suggests N will converge to K starting from any positive initial position $N(0) > 0$. We now see why K is called the carrying capacity. This is the limit of the environment where the population in focus occurs. Large K implies the environment can support a dense population.

When initial population size is positive and very small, $N \ll 1$, equation (2) can be approximated by ignoring the second order of N as

$$\frac{dN}{dt} = r\left(1 - \frac{N}{K}\right)N = rN - r\frac{N^2}{K} \approx rN$$

This is an exponential growth (approximation valid only $N \ll 1$) and we see why r is called intrinsic growth rate. It is the rate of increase per individual in an ideal situation. Typical dynamics of the logistic growth are shown in Figure 1.

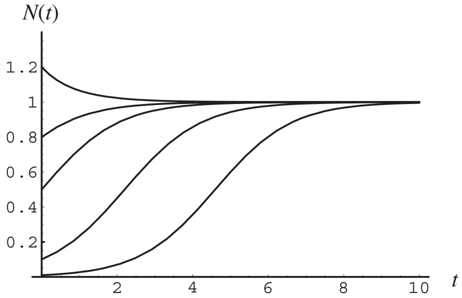


Figure 1: Logistic growth starting from various initial states. $r = K = 1$

What we should note here is that the deterministic logistic growth dynamics is solely determined by the functional form of the subtraction $birth(N) - death(N)$, not the form of individual function $birth(N)$ or $death(N)$. Therefore all the following functional forms produce exactly the same dynamics in the deterministic logistic growth.

Case A: $birth(N) = \beta\left(1 - \frac{N}{K}\right) \quad death(N) = \delta$

Case B: $birth(N) = \beta\left(1 - \frac{N}{2K}\right) \quad death(N) = \delta + \frac{\beta}{2K}N$

Case C: $birth(N) = \beta \quad death(N) = \delta + \frac{\beta}{K} N$

In case A, the birth rate linearly decreases with N, becomes zero at $N = K$ and the death rate is constant. In case B both the birth and death rate depend linearly on N. In case C the birth rate is constant but the death rate linearly increases with N. All these cases can be represented using the following functions

$$birth(N) = b_1 - b_2 N$$
$$death(N) = d_1 + d_2 N$$

where b_1, b_2, d_1, d_2 are positive constant.

In deterministic world these birth and death rates give ODE

$$\begin{aligned}\frac{dN}{dt} &= \{b_1 - d_1 - (b_2 + d_2)N\}\, N \\ &= (b_1 - d_1)\left(1 - \frac{N}{\frac{b_1 - d_1}{b_2 + d_2}}\right) N\end{aligned}$$

This is a logistic growth with intrinsic growth rate $r = b_1 - d_1$ and carrying capacity $K = (b_1 - d_1)/(b_2 + d_2)$. In this lecture we assume $b_1 > d_1$ to guarantee the intrinsic growth rate be positive. Otherwise the population always goes extinct and we are not willing to explore such a trivial case.

2 A stochastic model

As in previous lectures we focus on a stochastic version of the logistic growth described above. With analogies from the immigration-emigration model and the birth-death model, we here assume that $birth(N)$ is the probability that an individual gives birth to a new offspring per unit time and that $death(N)$ is the probability that this individual dies. That is, for all individuals, 1) reproduction of a new individual occurs with probability $birth(N)\Delta t$, or 2) death occurs with probability $death(N)\Delta t$, or 3) nothing happens with probability $1 - birth(N)\Delta t - death(N)\Delta t$ where Δt is a time interval. These three events are assumed to occur mutually exclusively. Then the algorithm would be

```
For all individuals repeat
1) Give birth to a new individual with probability $birth(N)\Delta t$.
2) Remove this individual with probability $death(N)\Delta t$.
```

An outline of C program would be like this.

```
#define b1 0.2
#define b2 0.002
#define d1 0.1
#define d2 0.0
#define DT 0.1
#define INTV 10

main()
{
  int pop_size, new_indiv, dead_indiv, i, step;
  double prob_birth, prob_death, ran;

  pop_size = 10;   /* initial population size */

  for(step=0; step<500*INTV; step++){    /* advance the time by DT */

     new_indiv = 0;
     dead_indiv = 0;
     prob_birth = birth_rate(pop_size)* DT;
     prob_death = death_rate(pop_size)* DT;

     for(i=1; i<=pop_size; i++){  /* for all individuals */

       ran = ran2(seed);
       if( ran < prob_birth )
         new_indiv++;
       else if( prob_birth < ran && ran < prob_birth + prob_death )
         dead_indiv++;

     } /* end of for i */

     pop_size += (new_indiv - dead_indiv) ;

  } /* end of for step */
}

double birth_rate(int n)
{
  double tmp;
  tmp = b1 - b2*(double)n;
  return tmp;
}
double death_rate(int n)
{
  double tmp;
  tmp = d1 + d2*(double)n;
  return tmp;
}
```

3 Waiting time

As in the previous stochastic models, it would be better to introduce waiting time to run simulation. Either a birth or a death takes place with the rate $(b_1 - b_2 N)N + (d_1 + d_2 N)N$, and waiting time to the next event (either birth or death) w is given as an exponential distributed random variable. A birth occurs with the conditional probability $(b_1 - b_2 N)N/((b_1 - b_2 N)N + (d_1 + d_2 N)N)$ and a death likewise. In this case we have to be cautious that the birth rate be always positive. For the sake of calculating average population size later, it is convenient to output the population size N with an equal time interval $\Delta T = 1$.

4 Problem

1. Write a C program to carry out the simulation of the stochastic logistic growth process. In the simulation we start with an initial population size, say, $n(0) = 10$ and repeat the dynamics with the same initial condition for several times. The dynamics of the population size should be written into a file. The data should be separated by a white space and write them in one line in the following format.

Trial 1:	$n(0)$	$n(1)$	$n(2)$	$\cdots$	$n(200)$
Trial 2:	$n(0)$	$n(1)$	$n(2)$	$\cdots$	$n(200)$
Trial 3:	$n(0)$	$n(1)$	$n(2)$	$\cdots$	$n(200)$
$\vdots$					

2. Using *Mathematica*, draw the simulated dynamics to see how the population size changes.

5 Appendix

Logistic growth model of equation(2) can be analytically solved as follows. First separate the variables N and t as

$$\frac{dN}{(1 - N/K)N} = rdt$$

Decompose the left hand side into partial fractions.

$$\frac{dN}{K - N} + \frac{dN}{N} = rdt$$

Integrate both the sides

$$\int \frac{dN}{K - N} + \int \frac{dN}{N} = r \int dt$$

to yield

$$-\log|K - N| + \log|N| = rt + C$$

With some re-arrangement

$$\log\left|\frac{N}{K - N}\right| = rt + C$$

we have

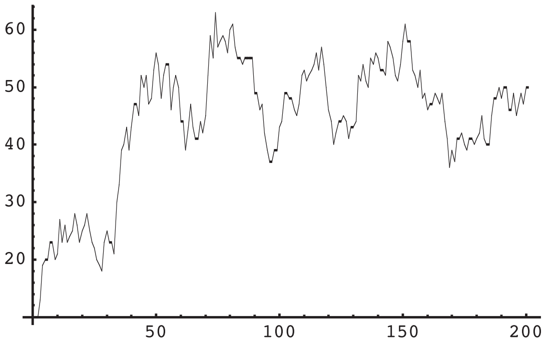
$$\left|\frac{N}{K-N}\right| = C'e^{rt}$$

Re-arranging this again using initial condition $N(0)$ completes the solution.

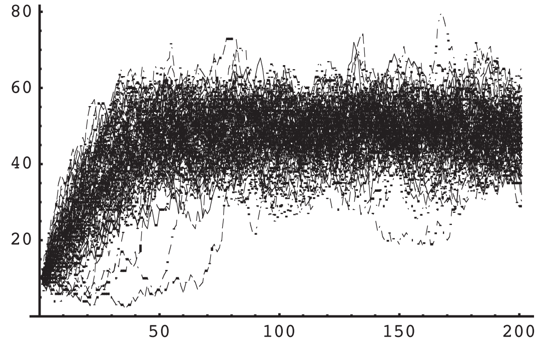
$$N(t) = \frac{N(0)Ke^{rt}}{K + N(0)(e^{rt}-1)}$$

Stochastic logistic process

```
In[20]:=  << Graphics`MultipleListPlot`

In[21]:=  << Statistics`DescriptiveStatistics`

In[22]:=  SetDirectory["/Users/takasu/home/情報科学科の仕事/講義/平成１８年度/H18
             大学院講義/Logistic growth model/logistic_models/build/Development/"]

Out[22]=  /Users/takasu/home/情報科学科の仕事/講義/平成１８年度/H18
            大学院講義/Logistic growth model/logistic_models/build/Development

In[189]:=  data = ReadList["data-eq_intv", Real, RecordLists ? True];
           Length[data]

Out[190]=  100

In[191]:=  ListPlot[data[[1]], PlotJoined ? True, PlotRange ? All]
```

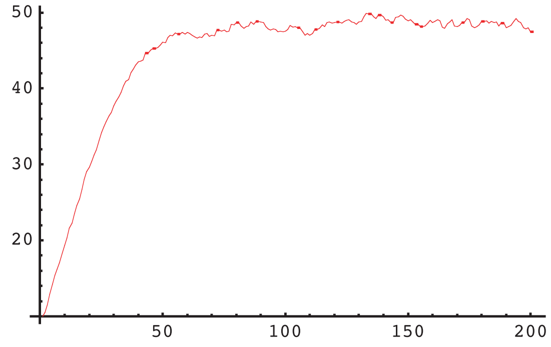


```
Out[191]=  ? Graphics ?

In[192]:=  gSimulation =
            MultipleListPlot[data, PlotJoined ? True, SymbolShape ? None, PlotRange ? All]
```

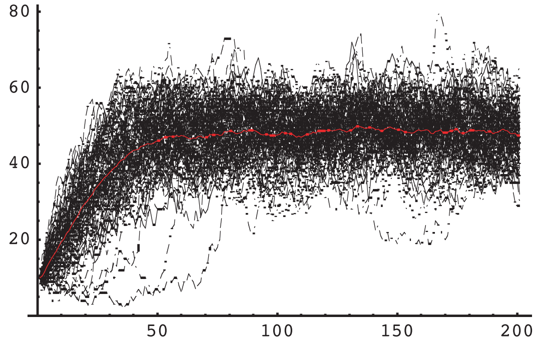


```
Out[192]=  ? Graphics ?

In[193]:=  dataT = Transpose[data];

In[194]:=  averageDyna = Map[ Mean, dataT];
```

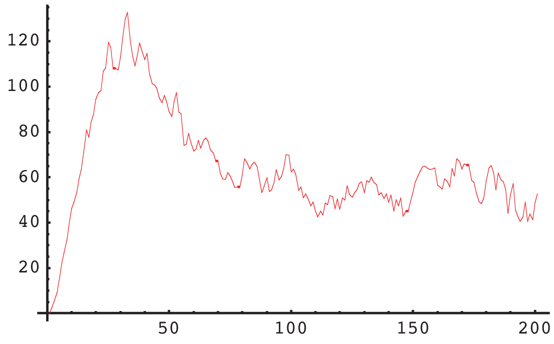
```
In[195]:= gAverage = ListPlot[ averageDyna,
            PlotJoined ? True, PlotStyle ? RGBColor[1, 0, 0], PlotRange ? All]
```



```
Out[195]=  ? Graphics ?

In[196]:= Show[gSimulation, gAverage]
```

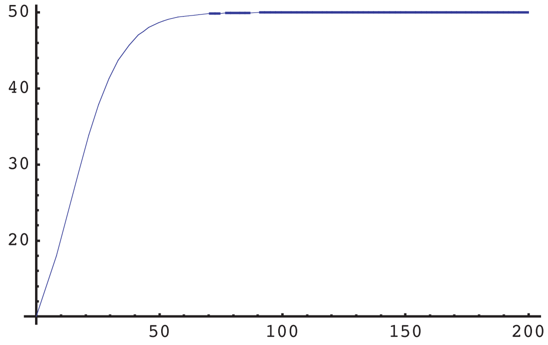


```
Out[196]=  ? Graphics ?

In[197]:= varianceDyna = Map[ VarianceMLE, dataT];

In[198]:= gVariance = ListPlot[varianceDyna,
            PlotJoined ? True, PlotStyle ? RGBColor[1, 0, 0], PlotRange ? All]
```

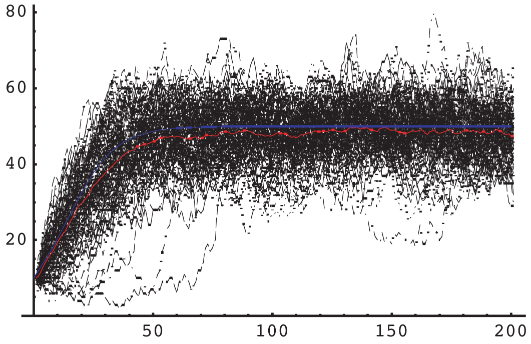


```
Out[198]=  ? Graphics ?

In[199]:= var = (b1 d2 + b2 d1) / (b2 + d2)^2 /. {b1 ? 0.2, b2 ? 0.002, d1 ? 0.1, d2 ? 0}

Out[199]=  50.
```

```
In[200]:=  deq = {n'[t] ?  r (1 - n[t] / k) n[t], n[0] ? 10} /. {r ? 0.1, k ? 50}

                                n[t]
Out[200]=  {n?[t] == 0.1 (1 - ??????) ? n?[t], n[0] == 10}
                                 50

In[201]:=  sol = NDSolve[deq, n[t], {t, 0, 200}]

Out[201]=  {{n[t] ? InterpolatingFunction[{{0., 200.}}, <>][t]}}

In[202]:=  gAna = Plot[Evaluate[n[t] /. sol],
             {t, 0, 200}, PlotRange ? All, PlotStyle ? RGBColor[0, 0, 1]]
```



```
Out[202]=  ? Graphics ?

In[203]:=  Show[gSimulation, gAverage, gAna]
```



```
Out[203]=  ? Graphics ?

In[204]:=  data = ReadList["data-wt_intv", Real, RecordLists ? True];
           Length[data]
           data2 = Map[Partition[#, 2] &, data];

Out[205]=  100
```