Statistical physics

lecture 12

Szymon Stoma

18-01-2010

X

Stochastic simulation:

1. First θ^0 is sampled from the π^0
2. Choose θ^1 is sampled from the $p(\theta^1, .)$
3. Repeat n steps step 2: sample θ^{n+1} from $p(\theta^n, .)$

X

```
markov_path <- function(n,P,pi0)
# n - path length
# P - transition matrix
# pi0 - initial distribution
{
    v = vector( "numeric",n )
    r = length( pi0 )
    v[ 1 ] = sample( 1:r, 1, prob=pi0 )
    for ( i in 2:n )
    {

        v[ i ] = sample( 1:r, 1, prob=P[ v[ i-1 ], ] )
    }
    ts( v )
}
```

X

Fact

Example: Markov chain in contnous space

$$Z_t = \alpha Z_{t-1} + \epsilon_t, \;\; \epsilon_t \sim N(0, \sigma^2)$$

$$p(x, y) = p(y|x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-\alpha x}{\sigma}\right)^2}$$

Lotka-Volterra

$$Y_1 \longrightarrow 2Y_1$$

$$Y_1 + Y_2 \longrightarrow 2Y_2$$

$$Y_2 \longrightarrow \emptyset$$

Lotka-Volterra

\

$$\frac{d[Y_1]}{dt} = k_1[Y_1] - k_2[Y_1][Y_2]$$

$$\frac{d[Y_2]}{dt} = k_2[Y_1][Y_2] - k_3[Y_2]$$

Lotka-Volterra

$$\frac{d}{dt}\left(\begin{bmatrix} [Y_1] \\ [Y_2] \end{bmatrix}\right) = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} k_1[Y_1] \\ k_2[Y_1][Y_2] \\ k_3[Y_2] \end{bmatrix}$$

Lotka-Volterra

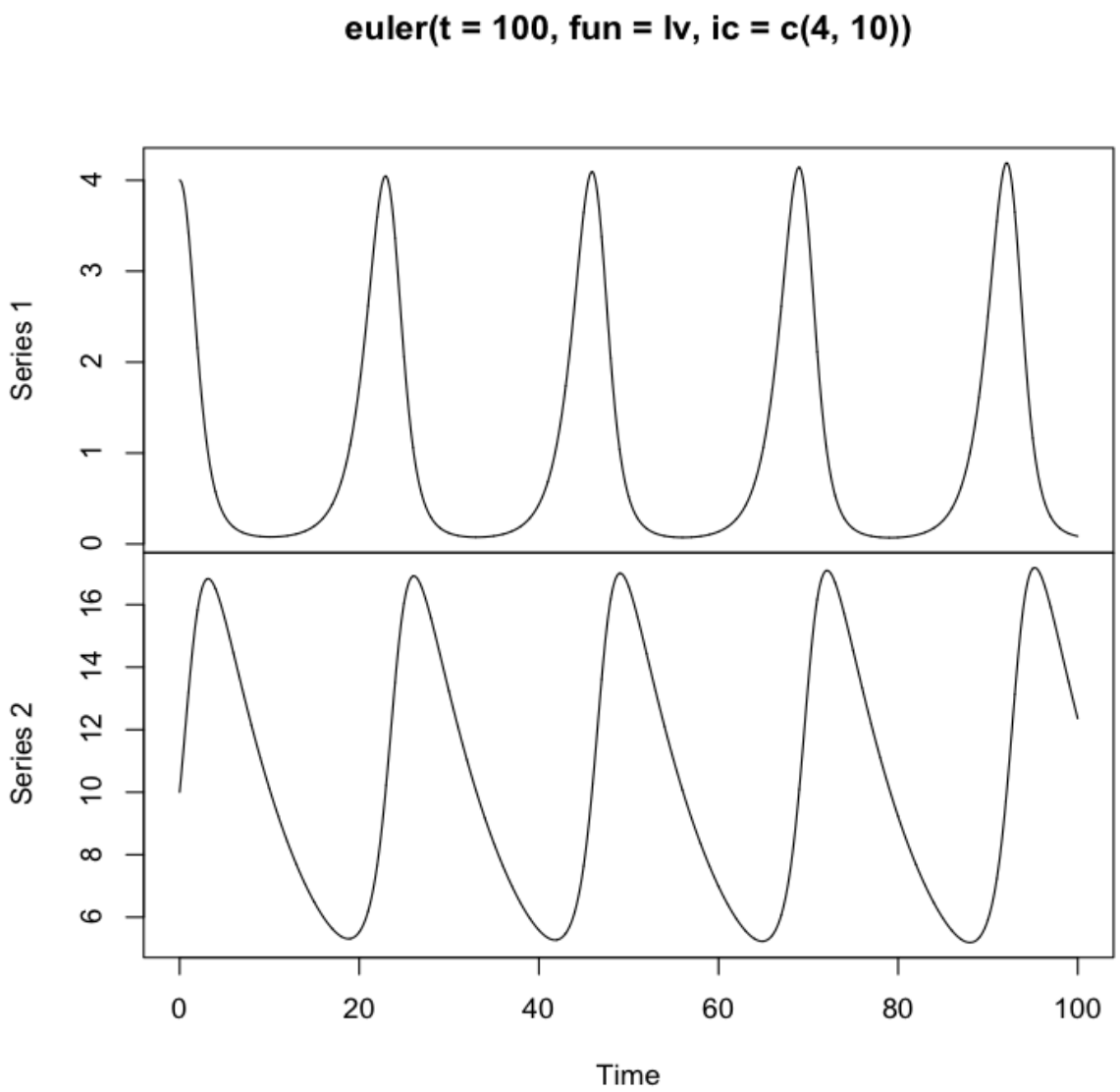
Stable state of Lotka-Voltera:

$$0 = k_1[Y_1] - k_2[Y_1][Y_2]$$

$$0 = k_2[Y_1][Y_2] - k_3[Y_2]$$

Solutions are: $[Y_1] = 0, [Y_2] = 0$ and $[Y_1] = k3/k_2, [Y_2] = k_1/k3$

Lotka-Volterra



Numerical solving

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

$$\frac{dX}{dt}(t) = lim_{\Delta t \to 0} \frac{X(t+\Delta t) - X(t)}{\Delta t}$$

$$f(X(t)) \approx \frac{X(t+\Delta t) - X(t)}{\Delta t}$$

$$X(t+\Delta t) \approx X(t) + f(X(t))\Delta t$$

R

```
euler <- function(t=50, dt=0.01, fun=f, ic=c(1,1), ...)
{
    p=length(ic)
    n=t/dt
    xmat=matrix(0,ncol=p,nrow=n)
    x=ic
    xmat[1,]=x
    for (i in 2:n)
    {
        x = x +fun(x, ...)*dt
        xmat[i,] = x
    }
    return(ts(xmat, start=0,deltat=dt))
}
```

R

```
lv <- function(x, k1=1,k2=0.1,k3=0.1)
{
c( k1*x[1] - k2*x[1]*x[2],
k2*x[1]*x[2] - k3*x[2] )
}

plot(euler(t=100,fun=lv,ic=c(4,10)))
```

Molecular approach

Consider a single pair of molecules: $x : X, y : Y$ moving randomly in a volume V. It can be shown that the reaction $X + Y \rightarrow ?$ have a constant hazard, assuming e.g. well stired, thermal equilibrium, no exchange with environment. Let c_i describe stochastic rate constant of reaction i.

Molecular approach

Stochastic petrinet: $N = (P, T, Pre, Post, M, R)$.
$R = [h_1(x, c_1), ..., h_m(x, c_m)]$, where m is reaction number $(dim(T))$. The probability that the R_i will occur in the time interval $(t, t + dt)$ is given by the $h_i(x, c_i)dt$. Thus in the abscence of any other reactions the time to such reaction would be $Exp(h_i(x, c_i))$

Example

$$N = (P, T, Pre, Post, M, R),\ P = (Prey, Predator)',$$

$$T = (PreyRepr., Pred2Prey, PredDeath), Pre = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, Post =$$

$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}, R = (c_1 y_1, c_2 y_1 y_2, c_3 y_2)', M = (50, 100)', c =$$

$$(1, 0.005, 0.6)'$$

Molecular approach

$$R_i, c_i : \emptyset \longrightarrow X$$

$$h_i(x, c_i) = c_i$$

Molecular approach

$$R_i, c_i : X_j \rightarrow ?$$

$$h_i(x, c_i) = c_i x_j$$

Molecular approach

$$R_i, c_i : X_j + X_k \rightarrow ?$$

$$h_i(x, c_i) = c_i x_j x_k$$

Molecular approach

$$R_i, c_i : 2X_j \rightarrow ?$$

$$h_i(x, c_i) = c_i x_j (x_j - 1)/2$$

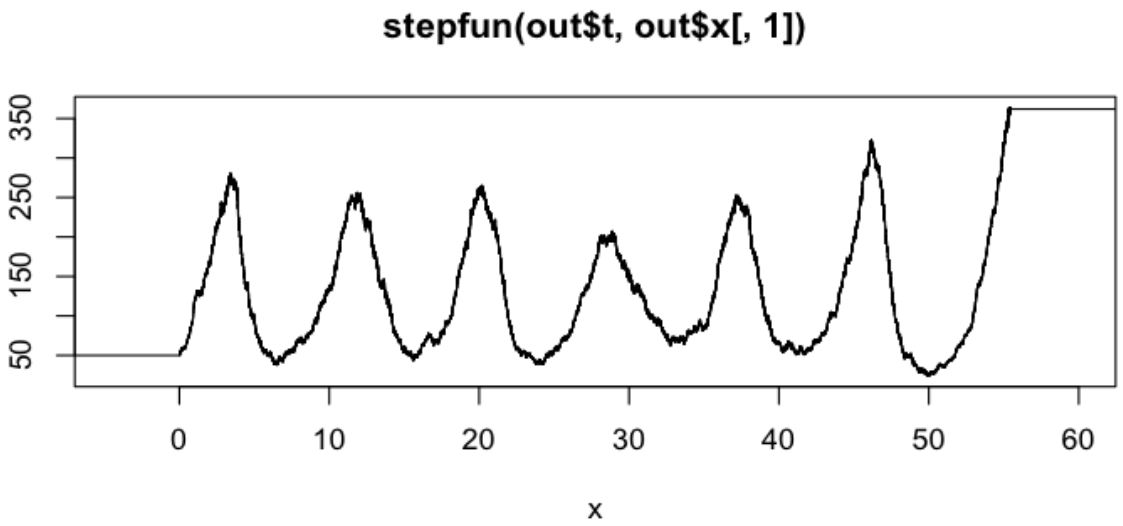
Gillespie algorithm

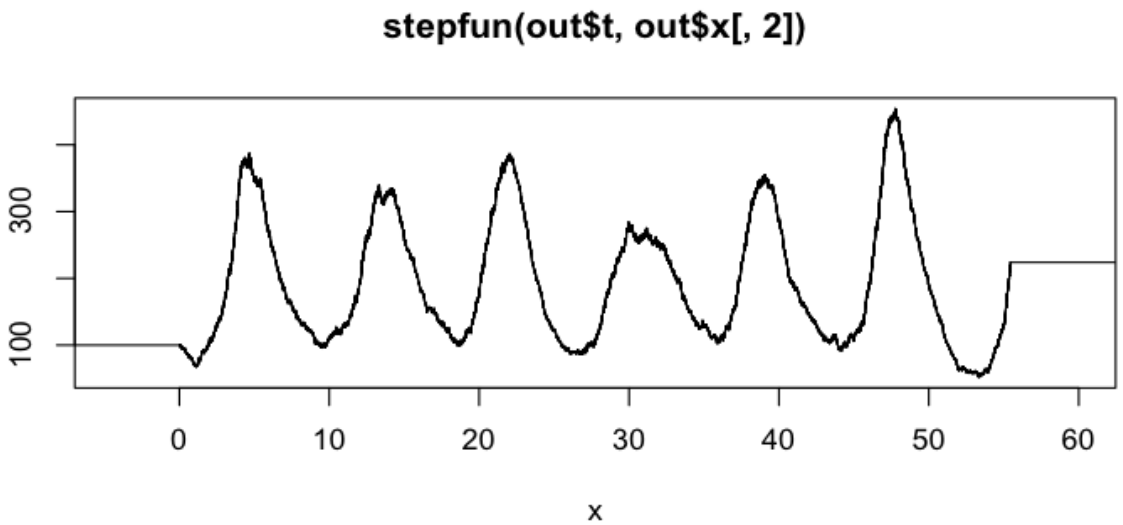
1. Initialize the system at $t = 0$ with rates $c_1, ..., c_m$ and initial numbers of molecules $x_1, ..., x_n$
2. For each $i = 1, ..., m$ calculate $h_i(x, c_i)$
3. Calculate $h_0(x, c) = \sum_{i=1}^{m} h_i(x, c_i)$, a combined reaction hazard
4. Sample a time for the next event dt from $Exp(h_0(x, c))$
5. $t = t + dt$
6. Sample the reaction index i from the discrete random variable with prob. $h_i(x, c_i)/h_0(x, c), i = 1, ..., m$
7. Update x according to the reaction R_i
8. If $t < t_{max}$ goto 2

R

```
gillespie <- function(N,n, ...) {
    tt = 0
    x=N$M
    S=t(N$Post-N$Pre)
    u=nrow(S)
    v=ncol(S)
    tvec=vector("numeric",n)
    xmat=matrix(0,ncol=u,nrow=n+1)
    xmat[1,]=x
    for (i in 1:n) {
        h=N$h(x, ... )
        tt=tt+rexp(1, sum(h) )
        j = sample(v,1, prob=h)
        x=x+S[,j]
        tvec[i]=tt
        xmat[i+1,]=x
    }
    return(list(t=tvec,x=xmat)) }
```

X


stepfun(outt, outx[, 1])
350
250
150
50
0
10
20
30
40
50
60
x


stepfun(outt, outx[, 2])
300
100
0
10
20
30
40
50
60
x