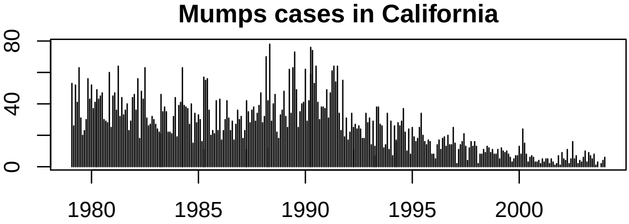
Stochastic Models

John M. Drake & Pejman Rohani

Epidemiological data are noisy

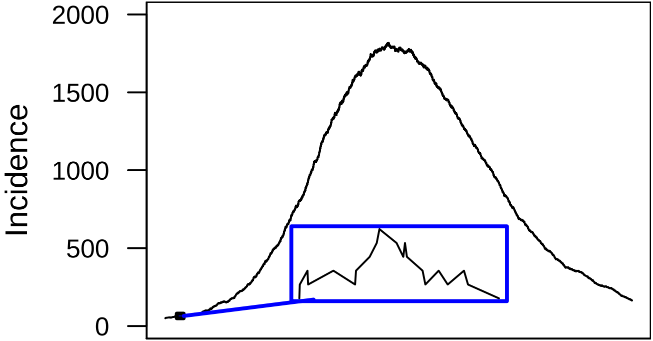
Two types of noise:

- Observation error: the data are probabilistically related to the true state of the system
- Process noise: the system progresses probabilistically
  - Environmental noise: some parameter is a random variable
  - Demographic noise: individual-level chance events



Noise is addressed using stochastic models

The SIR model is a continuum approximation



The *SIR* model (e.g., $dY/dt = \beta XY/N - \gamma Y$) implies that changes in the states X, Y, and Z are continuous. But, in reality individuals are either susceptible, infected, or recovered so that X, Y, and Z are integer-valued and changes in the system state occur as discrete steps. The differential equation is an idealization.

Demographic stochasticity

- What we seek is a stochastic model for which the system of ODEs is an appropriate idealization
- There are an infinite number of such models, but the simplest one is a continuous-time, discrete-space Markov Chain with propensities given by the various terms in the differential equations
- Then the ODEs are a "mean field" theory for the stochastic model (the average of the fluctuations are given by the ODEs)
- This model may also be interpreted as an event-driven model with state transition probabilities

Analytic solution

"Master Equation"

$$dP_k/dt = \sum_l A_{kl} P_l \qquad (1)$$

where A is a matrix of transition propensities

This approach is only tractable for very simple models (e.g. *SI* and *SIS* epidemics)

Simulation approach

Exact simulation is straightforward via Gillespie's Direct method:

- Initialize
- Iteration of a two step process
    1. Determine time of the next event
    2. Determine change of state at the next event time
- Summarize

Step 1: time to next event

Given system state N, let R(N) be the sum of all the propensities for all changes of state and $G_N(s)$ be the probability that no event occurs in subsequent time interval s for system state N.

By the Markov assumption

$$\begin{aligned}
G_N(s+\delta s) &= Pr\left\{\text{no event in}(t, t+s+\delta s)\right\}\\
&= Pr\left\{\text{no event in}(t, t+s)\right\} \times Pr\left\{\text{no event in}(t+s, t+s+\delta s)\right\}\\
&= G_N(s) \times \left\{1 - R(N) \times \delta s\right\}
\end{aligned}$$

Step 1: time to next event

After rearranging

$$\frac{G_N(s+\delta s) - G_N(s)}{\delta s} = -R(N) \times G_N(s)$$

Letting $\delta s \to 0$

$$\frac{dG_N}{ds} = -R(N) \times G_N(s)$$

With solution

$$G_N(s) = e^{-R(N)s}$$

Thus, the probability the next event occurs in $(t, t+s)$ is

$$F_N(s) = 1 - e^{-R(N)s}$$

Step 1: time to next event

Given event time distribution F_N, an exponentially distributed random event time S can be obtained from a uniform random random variate U_1 by setting

$$U_1 = F_N(s) = 1 - e^{-R(N)S}$$

and solving to obtain

$$S = -\log(U_1)/R(N)$$

Step 2: change of state

Let the propensities of event types $E_1, E_2, E_3, ...$ be denoted $R_1, R_2, R_3, ...$ with total rate $R_{sum} = R(N) = \sum_i R_i$. In the long run, events of each type should occur with relative frequency $R_i/R(N)$. We can randomly draw event classes with these frequencies by simulating a second uniform random variate U_2 and assigning event class E_i if

$$R_{sum}^{-1} \sum_{i=1}^{p-1} R_i < U_2 \leq R_{sum}^{-1} \sum_{i=1}^{p} R_i.$$

Gillespie's direct method

1. Label all possible events $E_1, E_2, E_3, \ldots$
2. Initialize $t = 0$ and state N
3. Update step
    1. Calculate propensities $R_1, R_2, R_3, \ldots$
    2. Calculate $R_{sum} = R(N) = \sum_i R_i$
    3. Generate U_1 and transform to obtain S
    4. Generate U_2 and determine event type E_i
    5. Update state based on E_i
    6. Update time $t = t + S$
4. Go to step (3)

Example with *SIR* model

- Events:
  - E_1: Birth of susceptible individual $(X \rightarrow X + 1)$
  - E_2: Infection $(X \rightarrow X - 1, Y \rightarrow Y + 1)$
  - E_3: Recovery $(Y \rightarrow Y - 1, Z \rightarrow Z + 1)$
  - E_4: Death of susceptible individual $(X \rightarrow X - 1)$
  - E_5: Death of infected individual $(Y \rightarrow Y - 1)$
  - E_6: Death of recovered individual $(Z \rightarrow Z - 1)$
- Propensities
  - R_1: $\mu(X + Y + Z)$
  - R_2: $\beta XY/N$
  - R_3: γY
  - R_4: μX
  - R_5: μy
  - R_6: μZ

R code for example

We create a function `SIR.onestep` to perform calculations of each update step

```
> SIR.onestep <- function (x, params) {  #function to calculate one step of stochastic SIR
+   X <- x[2]                            #local variable for susceptibles
+   Y <- x[3]                            #local variable for infecteds
+   Z <- x[4]                            #local variable for recovereds
+   N <- X+Y+Z                           #total population size (subject to demographic change)
+   with(                                #use with as in deterministic model to simplify code
+        as.list(params),
+        {
+          rates <- c(mu*N, beta*X*Y/N, mu*X, mu*Y, gamma*Y, mu*Z)
+          changes <- matrix(c( 1, 0, 0,
+                              -1, 1, 0,
+                              -1, 0, 0,
+                               0,-1, 0,
+                               0,-1, 1,
+                               0, 0,-1),
+                            ncol=3, byrow=TRUE)
+          U1 <- runif(1)
+          tau <- -log(U1)/sum(rates)   # exponential waiting time
+          U2 <- runif(1)       #uniform random deviate
+          m <- min(which(cumsum(rates)>=U2*sum(rates)))
+          x <- x[2:4] + changes[m,]
+          return(out <- c(tau, x))
+        }
+        )
+ }
```

R code for example

Now we write a function SIR.model that iteratively calls SIR.onestep to simulate an epidemic

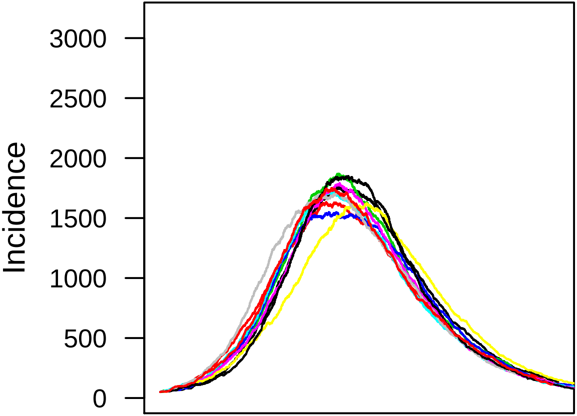
```
> SIR.model <- function (x, params, nstep) {  #function to simulate stochastic SIR
+   output <- array(dim=c(nstep+1,4))         #set up array to store results
+   colnames(output) <- c("time","X","Y","Z") #name variables
+   output[1,] <- x                           #first record of output is initial condition
+   for (k in 1:nstep) {                      #iterate for nstep steps
+     output[k+1,] <- x <- SIR.onestep(x,params)
+   }
+   output                                    #return output
+ }
```

R code for example

Finally, we write a code that calls `SIR.model` to simulate epidemics

```
> set.seed(38499583)                #set seed
> nsims <- 10                       #number of simulations
> pop.size <- 10000                   #total population size
> Y0 <- 50                           #initial number infected
> X0 <- round(0.98*pop.size)        #initial number suscepitlble (~98% of population)
> nstep <- 16000                     #number of events to simulate
> xstart <- c(time=0,X=X0,Y=Y0,Z=pop.size-X0-Y0) #initial conditions
> params <- list(mu=0.00001,beta=60,gamma=365/13) #parameters
> data <- vector(mode='list',length=nsims) #initialize list to store the output
> for (k in 1:nsims) {              #simulate nsims times
+   data[[k]] <- as.data.frame(SIR.model(xstart,params,nstep))
+   data[[k]]$cum.time <- cumsum(data[[k]]$time)
+ }
> max.time<-data[[1]]$cum.time[max(which(data[[1]]$Y>0))] #maximum time in first simulation
> max.y<-1.8*max(data[[1]]$Y)       #find max infected in run 1 and increase by 80% for plot
> plot(Y~cum.time,data=data[[1]],xlab='Time',ylab='Incidence',col=1,xlim=c(0,max.time),ylim=
> box()
> axis(2, cex.axis=0.8, las=2)
> for (k in 1:nsims) {              #add multiple epidemics to plot
+   lines(Y~cum.time,data=data[[k]],col=k,type='l')
+ }
```

R code for example



Some stochastic phenomena

J-U transition in final outbreak size

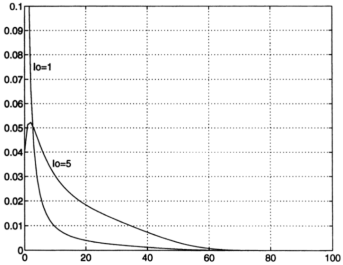


Figure 1. Size distribution of the general epidemic ($N = 100$, $R_0 = 0.9$).

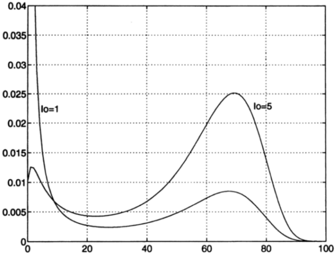
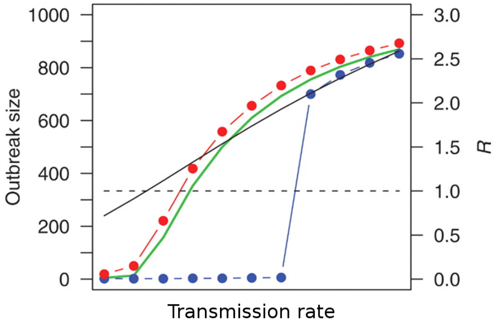


Figure 3. Size distribution of the general epidemic ($N = 100$, $R_0 = 1.5$).

J-U transition illustrated by Nasell (1995) in *Epidemic models: their structure and relation to data*

Some stochastic phenomena

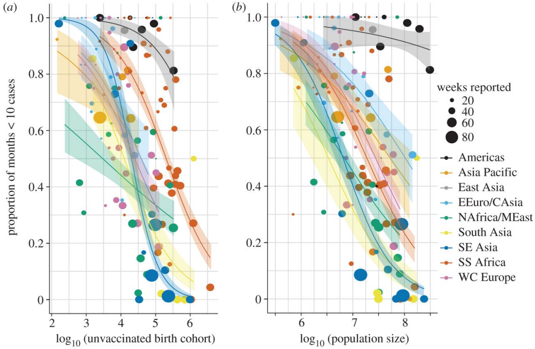
Difference between “likely” outcome (median: blue points) and “worst case scenario” (95^{th} percentile: red points) compared with deterministic approximation (green line) and R_0 (black line)



Park et al. 2009. *Science* 326:726-728

Some stochastic phenomena

Critical community size



Ferrari et al. 2013. *Philosophical Transactions of the Royal Society B* 368:20120141

Summary

- Transmission is obscured by three sources of noise: observation error, environmental variability, and intrinsic demographic noise
- Gillespie's direct method is a straightforward way to study the effects of demographic stochasticity in small populations
- Demographic noise is especially important in systems where $R_0 \approx 1$