Chapter 11
Spatial Dynamics

11.1 Dispersal Kernels

Pathogens move in space because of movement of transmission stages and infected/susceptible hosts. Spatial pattern arises from landscape heterogeneities, dispersal and "reaction-diffusion" dynamics among spatially dispersed susceptible, and infected individuals. The probability distribution that governs dispersal distances is often referred to as the dispersal kernel. A variety of functional forms have been proposed in the ecological and epidemiological literature (e.g., Mollison 1991; Clark 1998; Bjørnstad and Bolker 2000; Smith et al. 2002). From the point of view of basic theory, it is often assumed that dispersal takes an exponential (the probability of dispersing a distance $d \propto \exp(-d/a)$, where a is the range) or Gaussian ($\propto \exp(-(d/a)^2)$) shape. The exponential model arises, for example, if we assume dispersal happens in a constant direction with a constant stopping rate. The Gaussian model arises if the stopping rate is constant but movement direction changes randomly like a Brownian motion. However, other kernels are relevant; Broadbent and Kendall (1953) calculated the movement probabilities of infectious larvae of a gut nematode of sheep, *Trichostrongylus retortaeformis*, that performs a random walk until it encounters a leaf of grass. Assuming the location of the leaves are according to a spatially random point process, they showed that the random walk leads to a dispersal distance distributions that follows a Bessel K_0-function. Ferrari et al. (2006b)

This chapter uses the following R-packages: `ncf` and `animate`.
A conceptual understanding of *spatial spread* is useful prior to this discussion. A 5-min epidemics-MOOC can be seen on YouTube: https://www.youtube.com/watch?v=WPjsAdyD1Gg.

O. N. Bjørnstad, *Epidemics*, Use R!, https://doi.org/10.1007/978-3-319-97487-3_11

used this kernel in a model of pollinator-vectored plant pathogens. Empirical dispersal distribution of free-living organisms typically has an over-representation of rare long-range jumps that are improbable according to these kernels; They are the so-called "fat-tailed" kernels (Clark 1998), which have important consequences for the speed of spatial spread (Kot et al. 1996).

For human infections spatially contiguous, diffusive kernels are often a poor fit to empirical patterns because spread often follows a characteristic "hierarchical" fashion (Grenfell et al. 2001); Infections usually appear in big cities early, thereafter the timing of epidemics on average happens in an order of descending size and increasing isolation. This chapter is focused on inferring the shape of the spread-kernel from spatial patterns over time, and then investigates the dynamical consequences of such spread. We start with considering the simpler diffusive kernels and then consider the more complicated patterns arising from human mobility.

11.2 *Filipendula* Rust

Jeremy Burdon and Lars Ericson surveyed presence/absence of a fungal pathogen on a wild plant, *Filipendula ulmaria*, across islands in a Swedish archipelago (Smith et al. 2003). The `filipendula` data contains observations for 1994 ($y94) and 1995 ($y95), with spatial coordinates $X and $Y. There are additionally a large number of descriptive covariates for each site. Smith et al. (2003) used the data to estimate the most likely dispersal kernel of the rust. The host plant is an herbaceous perennial with pathogen spores overwintering on dead tissue. The infections in 1995 thus arose from the spores produced in 1994.

If spores disperse according to, say, an exponential function with range, a, then the spatial force of infection on any location, i, will be $\propto \sum_j z_j \exp(-d_{ij}/a)$, where z_j is the disease status (0/1) in the previous year and d_{ij} are the distances to other locations. The idea is that in each spring, every local group of hosts will be in the accumulated spore shadow of last year's infected individuals. This leads to a metapopulation "incidence-function" model (Hanski 1994) for the presence/absence of rust among all locations from year to year. Figure 11.1 shows the spatial data.

```r
data(filipendula)
symbols(filipendula$X, filipendula$Y, circles=
   rep(1,162), inches=.1, bg=filipendula$y95+1,
   xlab="X", ylab="Y")
symbols(filipendula$X, filipendula$Y, circles=
   rep(1,162), inches=.05, bg=filipendula$y94+1,
   add=TRUE)
legend("topright", c("infected 94", "infected 95"),
   pch=c(21,21), pt.cex=c(1,2), pt.bg=c(2,2))
```

As for the basic catalytic (Chap. 4) and TSIR (Chap. 7) models, we can use the `glm`-framework to estimate the parameters. Since the response variable is binary,

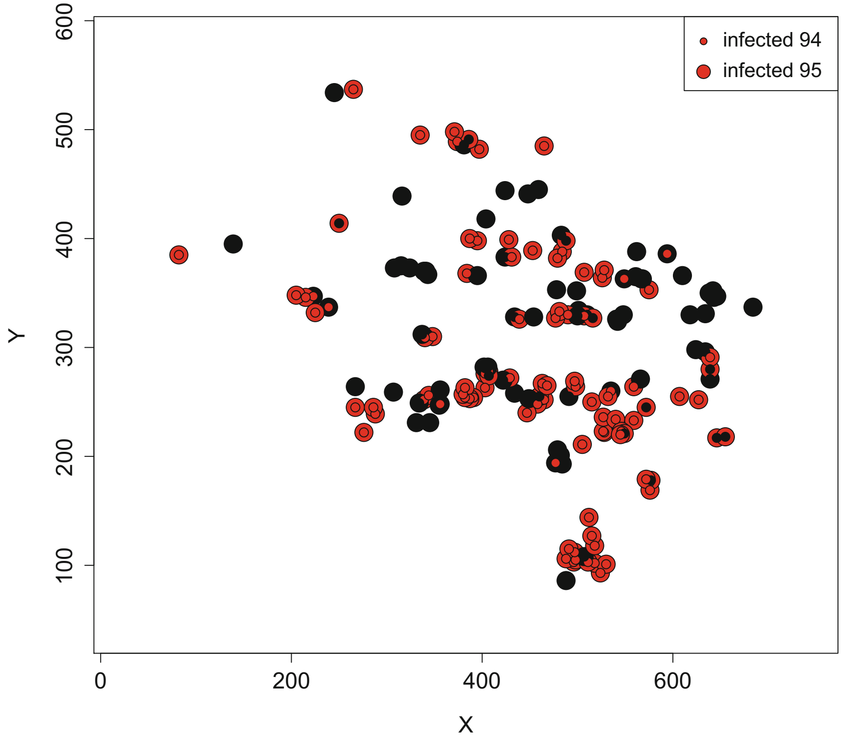

Fig. 11.1 Presence/absence of the rust on its *Filipendula ulmaria* host plant in 1994 and 1995. Red is infected. Black is uninfected

we use logistic regression to calculate a profile likelihood for a. We first calculate the distance matrix among the 162 locations:

```
dst = as.matrix(dist(filipendula[, c("X", "Y")]))
```

Arbitrarily assuming a value of a of 10 m, the 1995 FoI on each location will be proportional to:

```
a = 10
foi = apply(exp(-dst/a) * filipendula$y94, 2, sum)
```

We use glm to evaluate the likelihood. The `deviance` of the `glm` object is 2 times the negative log-likelihood.

```
lfit=glm(y95~foi, family=binomial(), data=filipendula)
lfit$deviance/2

  ## [1] 69.8527
```

Figure 11.2 shows the likelihood profile across candidate values for *a*.

```
a=seq(1,20, length=1001)
llik=rep(NA, length(a))
for(i in 1:length(a)){
   foi=apply(exp(-dst/a[i])*filipendula$y94,2,sum)
   lfit=glm(y95~foi, family=binomial(),
       data=filipendula)
   llik[i]=lfit$deviance/2
}
plot(a, llik, type="l", ylab="Neg. log-like")
abline(h=min(llik)+qchisq(0.95,1)/2)
```

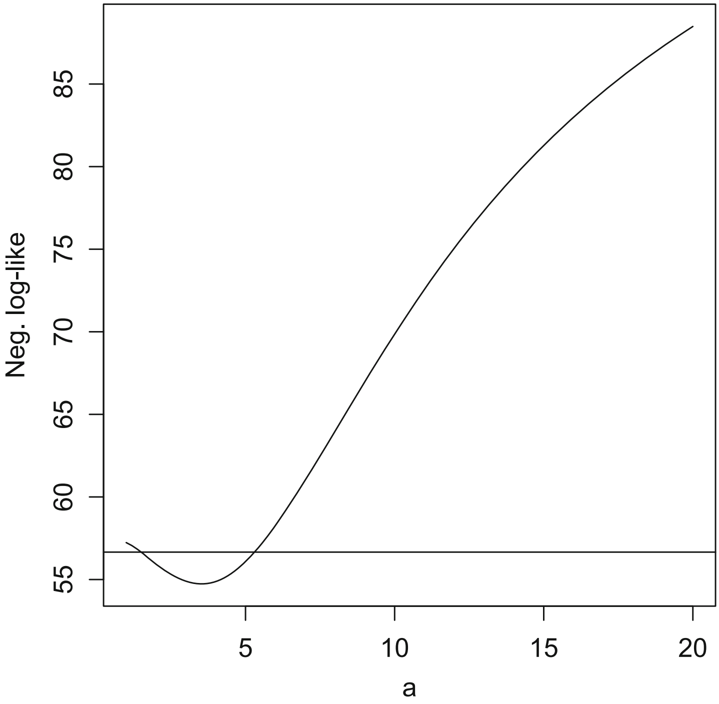

Fig. 11.2 Likelihood profile for *a* the parameter in the exponential dispersal kernel. The horizontal line represents the 95% cut off for the $\chi^2(1)/2$ deviation from the minimum

We can compare our best kernel model with a nonspatial model assuming a homogenous risk among hosts using likelihood-ratio tests (Sect. 8.4). Recall that for nested glm's (i.e., where the simpler model is nested within the more complicated model), the difference in deviances (= 2xlog-likelihood) is $\chi^2(df = \Delta p)$-distributed, where Δp is the number of extra parameters in the complex model. The anova-function provides this calculation in R. Since we first profiled on a, and then use the value $\hat{a}$ that minimizes the negative log-likelihood, we have to correct the residual degrees of freedom of the spatial model to get the correct likelihood-ratio test.

```
ahat=a[which.min(llik)]
foi=apply(exp(-dst/ahat)*filipendula$y94,2,sum)
spmod=glm(y95~foi, family=binomial(), data=filipendula)
nullmod=glm(y95~1, family=binomial(), data=filipendula)
#correct the df of the spmod
spmod$df.residual=spmod$df.residual-1
anova(nullmod, spmod, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y95 ~ 1
## Model 2: y95 ~ foi
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       161     222.10
## 2       159     109.48  2   112.63 < 2.2e-16
##
## 1
## 2 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The spatial model gives a highly significantly better fit than the null model.

The Gaussian dispersal kernel takes the form $\propto \exp(-(d_{ij}/a)^2)$. We can estimate the parameters assuming this alternative kernel:

```
a2=seq(1,20, length=1001)
llik2=rep(NA, length(a2))
for(i in 1:length(a2)){
   foi2=apply(exp(-(dst/a2[i])^2)*filipendula$y94,2,sum)
   lfit2=glm(y95~foi2, family=binomial(),
      data=filipendula)
   llik2[i]=lfit2$deviance/2
}
ahat2=a2[which.min(llik2)]
foi2=apply(exp(-(dst/ahat2)^2)*filipendula$y94,2,sum)
spmod2=glm(y95~foi2, family=binomial(),
   data=filipendula)
spmod2$df.residual=spmod2$df.residual-1
```

Finally, we can visualize the shape of the competing probability kernels (using appropriate scaling for power exponential functions) (Fig. 11.3):

```
curve((2/(ahat2 * gamma(1/2))) * exp(-((x/ahat2)^2)),
    0, 10, col = 2, lty = 2, ylab = "Probability",
    xlab = "Meters")
curve((1/(ahat) * gamma(1)) * exp(-x/ahat), 0, 10,
    add = TRUE)
legend("topright", c("Exponential", "Gaussian"),
    lty = c(1, 2), col = c(1, 2))
```

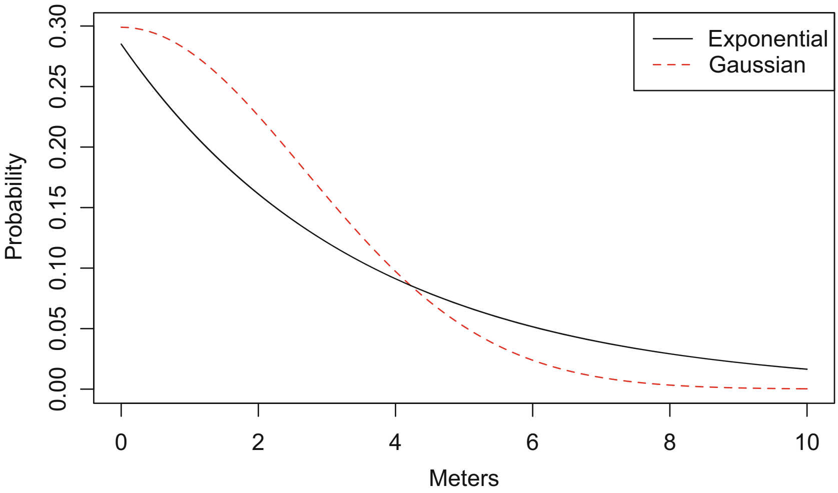

Fig. 11.3 The estimated exponential and Gaussian dispersal distance distributions for the *Filipendula* rust data

The two spatial models are not nested, but we can get model rankings using their AICs:

```
spmod$aic

  ## [1] 113.4775

spmod2$aic

  ## [1] 116.6538
```

The exponential model is favored over the Gaussian.

11.3 Simulation

In addition to being a statistical method, our binomial spatial model also represents a fully specified metapopulation model for presence/absence of the rust.[1] Since we used logistic regression (the default for the binomial-family), our regression provides estimates for $\text{logit}(p) = \beta_0 + \beta_1 * \text{foi}$. The inverse-link is $p = \exp(\beta_0 + \beta_1 * \text{foi})/(1 + \exp(\beta_0 + \beta_1 * \text{foi}))$.

We can write a simulator that stochastically projects the epidemic metapopulation forwards in time (assuming a fixed host plant distribution). We will initiate the simulation with the state of the system in 1995.

```
zprev = filipendula$y95
x = filipendula$X
y = filipendula$Y
beta0 = spmod$coef[1]
beta1 = spmod$coef[2]
```

Infection probabilities for next year are:

```
foi = apply(exp(-dst/ahat) * zprev, 2, sum)
logitp = beta0 + beta1 * foi
p = exp(logitp)/(1 + exp(logitp))
```

A stochastic realization is:

```
znew = rbinom(162, 1, p)
symbols(x, y, circles = rep(1, 162), bg = znew +
    1, inches = 0.1, xlab = "X", ylab = "Y")
```

We can animate the next 100 years (if uncommented, the `Sys.sleep` argument makes the computer go to sleep for 0.1 s to help visualization):

```
simdat=matrix(NA, ncol=100, nrow=162)
for(i in 1:100){
     zprev=znew
     foi=apply(exp(-dst/ahat)*zprev,2,sum)
     logitp=beta0+beta1*foi
     p=exp(logitp)/(1+exp(logitp))
     znew=rbinom(162, 1, p)
     simdat[,i]=znew
     #symbols(x, y, circles=rep(1,162), bg=znew+1,
```

[1] Just like the chain-binomial model in Sects. 3.4 and 3.5 the spatial logistic model can be used both as a statistical method and a stochastic simulator.

```
    #  inches=.1, xlab="X", ylab="Y")
    #Sys.sleep(0.1)
}
```

Figure 11.4 shows the predicted relative spatial risk from the stochastic simulation. The `spatial.plot`-function in the `ncf`-library is a wrapper for `symbols` that plots values larger (smaller) than the mean as red circles (black squares). In this case we see that spatial configuration alone can result in heterogenous infection risk across the metapopulation. A corollary of this is that specialist plant pathogens may regulate the spatial distribution of host plant recruitment through locally density-dependent mortality and thus promote species diversity according to the Janzen-Connell hypothesis (e.g., Clark and Clark 1984; Petermann et al. 2008).

```
require(ncf)
mprev = apply(simdat, 1, mean)
spatial.plot(x, y, mprev, ctr = TRUE)
```

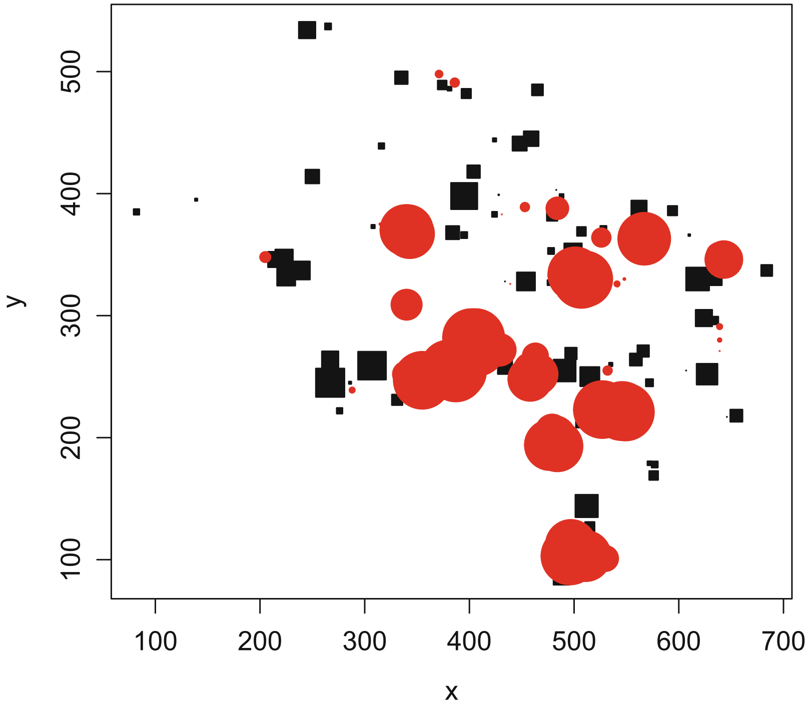

Fig. 11.4 Plot of predicted relative risk of rust infection from the metapopulation model. Risks larger (smaller) than the mean are shown as red circles (black squares). The size of the symbols reflects the deviation from the mean

11.4 Gypsy Moth

Various viruses and parasitoids of insects cause population instabilities and cycles in their hosts. The 5–10-year cycles in the gypsy moth (*Lymantria dispar*) are caused by the ldNPV-virus. Larvae get infected when ingesting viral occlusion bodies. The virus subsequently kills the larvae to release more of these infectious particles. USDA forest service conducts surveys each year of defoliation by the gypsy moth across the Northeastern USA to reveal complex spatiotemporal patterns. A web-optimized animated gif of the annual defoliation across the Northeastern USA between 1975 and 2002 can be viewed from https://github.com/objornstad/epimdr/blob/master/mov/gm.gif.

Spatiotemporal models can help to better understand such dynamics. There are specialized models for both the local and spatiotemporal dynamics of the gypsy moth (Dwyer et al. 2004; Abbott and Dwyer 2008; Bjørnstad et al. 2010). Here we will consider a simpler spatially extended SIR model.

11.5 Coupled Map Lattice SI Models

Coupled map lattice models[2] are constructed by assuming that spatiotemporal dynamics happens in two steps (Kaneko 1993; Bascompte and Solé 1995). First, local growth according to some model, for example, the seasonally forced (discrete time) SI model. Followed, second, by spatial redistribution of a fraction, `m`, of all individuals to other neighboring patches.

Because R is a vectorized language we can simulate CMLs using very compact code. We first write the function for the local SI dynamics according to the expectation from the chain-binomial formulation (Sect. 3.4). We assume a birth/death rate of μ and sinusoidal forcing on the transmission rate according to $\beta_0 + \beta_1 \cos(2 * \pi * t/26)$ (so there are 26 time-steps in a year). We assume infected individual stays infected and infectious for one time step.

```
local.dyn = function(t, S, I, b0, b1, mu, N) {
    beta = b0 * (1 + b1 * cos(2 * pi * t/26))
    I = S * (1 - exp(-beta * I))
    S = (1 - mu) * S + mu * N - I
    list(S = S, I = I)
}
```

Next we construct the redistribution matrix among the `nx`$-by-$`ny` locations (we consider a 30×30 lattice). Nearest-neighbors will be <1.5 spatial units apart (to

[2] The name refers to how the most stylized of these models assumes a lattice (checker board) of locations at which local numbers change from one generation to the next according to some “mapping”-rule such as the discrete logistic, the Nicholson-Baily model (see Chap. 14) or, in this case, a discrete-time seasonally forced SI model.

be exact $<\sqrt{3}$). Assume that the fraction that disperses to neighboring patches is $m = 0.25$ and that movement is random and independent of disease status.

```
m = 0.25
ny = nx = 30
# generate coordinates
xy = expand.grid(x = 1:nx, y = 1:ny)
# make distance matrix
dst = as.matrix(dist(xy))
# make redistribution matrix with zeros
redist = matrix(0, nrow = ny * nx, ncol = ny * nx)
# populate the matrix so each of the 8 neighbors
# gets their share
redist[dst < 1.5] = m/8
# the remaining fraction stays put
diag(redist) = 1 - m
```

The S and I matrices will hold the results from the simulation. We will run the model for IT=520 iterations (= 20 years). Assume that all patches have S0 = 100 susceptibles and that 1 infected is introduced in location {400, 1}:

```
IT = 520
S = I = matrix(NA, nrow = ny * nx, ncol = IT)
S[, 1] = 100
I[, 1] = 0
I[400, 1] = 1
```

We define the remaining parameters necessary for the local dynamics:

```
b0 = 0.04
b1 = 0.8
mu = 0.02/26
N = 1000
```

We are now ready to simulate the model. The %*%-operator represents matrix-multiplication and the matrix-multiplication of a vector of abundances with the redistribution-matrix moves all individuals appropriately.

```
for (t in 2:IT) {
    # local growth:
    tmp = local.dyn(t, S = S[, t - 1], I = I[, t -
        1], b0 = b0, b1 = b1, mu = mu, N = N)
    # spatial movement
    S[, t] = redist %*% tmp$S
    I[, t] = redist %*% tmp$I
    # progress monitor
    cat(t, " of ", IT, "\r")
}
```

The simulation can be visualized as an inline animation. The predicted incidence from the spatial SI-model varies so widely it is useful to transform incidence (using a fourth-root) so that low values shows up better.

```
x = xy[, 1]
y = xy[, 2]
scIcubed = I^(1/4)/(max(I[, 10:IT]^(1/4)))
```

```
for (k in 1:IT) {
    symbols(x, y, fg = 2, circles = scIcubed[, k],
        inches = FALSE, bg = 2, xlab = "", ylab = "")
    Sys.sleep(0.05)
}
```

Analyses of a variety of host-parasit(oid) CML models (Hassell et al. 1991; Bjørnstad et al. 1999b; Earn et al. 2000a) have revealed a variety of emergent spatiotemporal patterns including complete synchrony, waves, spatial chaos, and frozen patterns. The pattern in any given system depends on the local dynamics and mobility. We will visit on these CML models further in Chap. 14.

11.6 Making Movies

We can make permanent movies by writing the plots to a sequence of jpeg's and then use an open-source utility like ImageMagick to convert the sequence to a movie.[3]

```
for(k in 100:IT){
png(filename=paste("m",1000+k,".jpg", sep=""))
    symbols(x, y, fg=2, circles=scIcubed[,k],
    inches=FALSE, bg=2,xlab="",ylab="")
dev.off()
}
```

[3] The `system()`-function in R passes the `convert` and `rm` calls to the command-line. A web-optimized version of the animated gif can be viewed on https://github.com/objornstad/epimdr/blob/master/mov/simovie.gif.

```
system("convert m*.jpg simovie.gif")
system("rm m*.png")
#For mp4-animation:
#system("convert -delay 5 m*.jpg simovie.mp4")
```

Alternatively we can incorporate the animation directly into a pdf—though for this to work we need to work with LaTeX and use the LaTeX animate-package.

```
require("animation")
oopt = ani.options(interval = 0.02, nmax = 100)
test.function = function (xy, I, nmax) {
    x = xy[,1]
    y = xy[,2]
    scIcubed = I^(1/4)/(max(I[,10:IT]^(1/4)))
    for (i in seq_len(ani.options("nmax"))) {
        dev.hold()
        symbols(x,y,fg=2,circles=I[,i],inches=0.1,bg=2,
           xlab="",ylab="")
        ani.pause()
    }
}

saveLatex({
    test.function(xy=xy, I=I, nmax=50)
    },
    ani.basename = "BM", ani.opts = "controls,loop,
    width=0.8\\textwidth", ani.first =
     par(mar = c(3, 3, 1, 0.5), mgp = c(2, 0.5, 0),
     tcl = -0.3, cex.axis = 0.8, cex.lab = 0.8,
     cex.main = 1), latex.filename = "test.tex",
     pdflatex = "/usr/texbin/pdflatex",
     img.name = "Xplot")
ani.options(oopt)
```

11.7 Nonparametric Covariance Functions for Spatiotemporal Data

Keeling et al. (2002) discuss how we may understand the emergent complicated spatiotemporal dynamics of models of natural enemies in terms of the spatial variance (or associated autocorrelation) and covariance of the interacting species.[4] Bjørnstad and Bascompte (2001) proposed to calculate auto- and cross-correlation functions

[4] Seabloom et al. (2005) provide similar calculations for spatial competition models.

from simulated or real data. We can use the Sncf-function in the ncf-package to calculate the "multivariate" spatial correlation function (Bjørnstad et al. 1999b) among the simulated time series (see Chap. 13 for further details on this and other geostatistical methods). We can further look at the spatial *cross-correlation* function between susceptibles and infected (Fig. 11.5). The background synchrony for both compartments (of around 0.3) is due to the common seasonal forcing. The locally higher autocorrelation at shorter distances is due to emergence of dispersal-induced aggregations of infected individuals. The negative local cross-correlation is due to the local S-I cycles.

```
fitI = Sncf(x = xy[, 1], y = xy[, 2], z = sqrt(I[,
    261:520]), resamp = 500)
fitS = Sncf(x = xy[, 1], y = xy[, 2], z = sqrt(S[,
    261:520]), resamp = 500)
fitSI = Sncf(x = xy[, 1], y = xy[, 2], z = sqrt(S[,
    261:520]), w = sqrt(I[, 261:520]), resamp = 500)
par(mfrow = c(1, 3))
plot(fitI, ylim = c(-0.1, 1))
plot(fitS, ylim = c(-0.1, 1))
plot(fitSI, ylim = c(-0.2, 0.2))
```

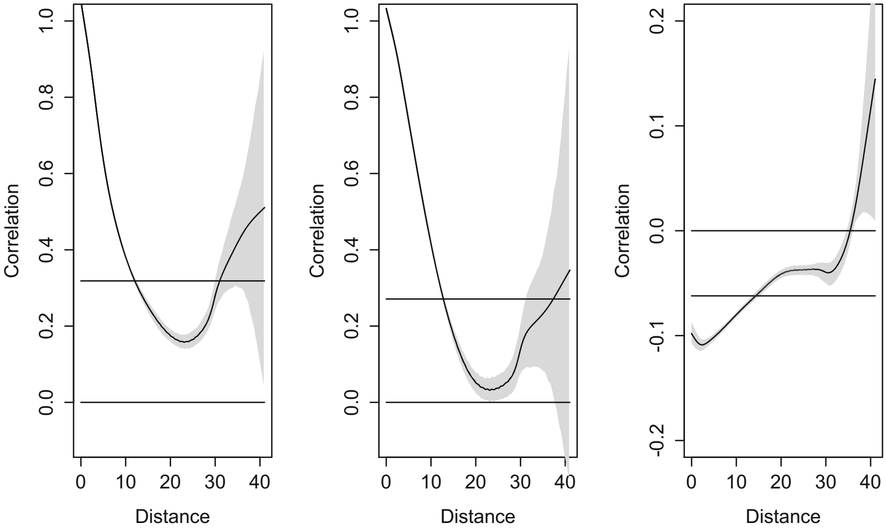

Fig. 11.5 Spatial correlation (1) infecteds, (2) susceptibles, and (3) S-I cross-correlation as a function of distance

One interesting additional application is the so-called time-lagged spatial correlation function (Bjørnstad et al. 2002a). This analysis may help quantify wave-like spread. For example we can look at the spatiotemporal relationship between the infecteds and themselves 5 time-steps later (Fig. 11.6). The peak in correlation is offset from the origin by somewhere between 5 and 10 units. This makes sense, since we assume nearest neighbor dispersal, so the leading edge should move 5 units vertically/horizontally and $5 * \sqrt{2} = 7.1$ units diagonally during 5 time steps.

```
fitIlag = Sncf(x = xy[, 1], y = xy[, 2], z = I[,
    261:515], w = I[, 266:520], resamp = 100)
plot(fitIlag, ylim = c(-0.2, 0.2))
```

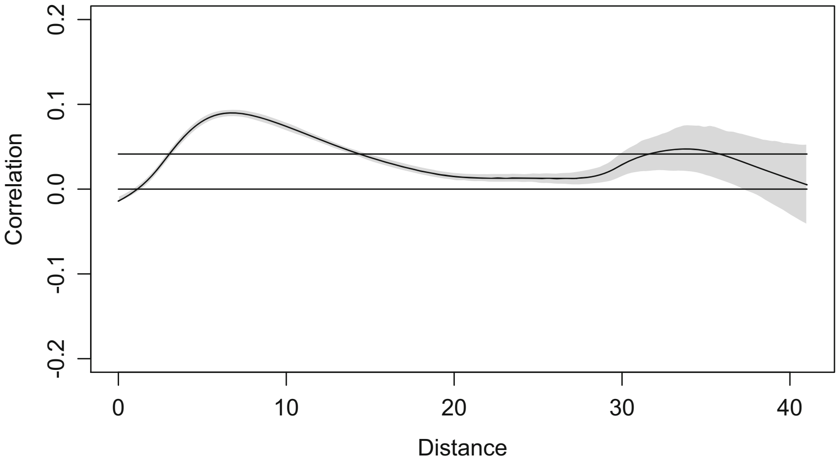

Fig. 11.6 The time-lagged spatial cross-correlation function of predicted prevalence of the SI cml model (with a 5-year time lag)

Bjørnstad et al. (2002b) used time-lagged spatial correlation functions to show that parasitoid-host interactions (see Chap. 14) lead to waves of larch tree defoliation that travels at 210 km per year in a north-easterly direction across the European Alps. Traveling waves have also been documented in the dynamics of dengue (Cummings et al. 2004) and influenza A (Gog et al. 2014).

11.8 Gravity Models

Regional spread of human pathogens rarely forms a simple diffusive pattern because human mobility patterns are more complex—movement may be distant dependent, but overall flow between any two communities also typically depend on the size (and desirability) of both “donor” and “recipient” location (Erlander and Stewart 1990; Fotheringham 1984). Grenfell et al. (2001), for example, showed that the spatiotemporal dynamics of measles across all cities and villages in pre-vaccination England and Wales exhibited “hierarchical waves,” in which the timing of epidemics relative to the big urban conurbations (the donors) depended negatively on distance but positively on the size of the recipient. Viboud et al. (2006) demonstrated similar hierarchical spread of seasonal influenza across the states of continental USA.

Xia et al. (2004) and Viboud et al. (2006) subsequently showed that a metapopulation model where movement among communities followed a “generalized gravity model” approximates the dynamic patterns; The “gravity model” is a model of mobility/transportation from transportation science that posits that transportation volume between two communities depends inversely on distance, d, but bilinearly on the size, N, of the communities (Erlander and Stewart 1990; Fotheringham 1984). Gravity-like models have since been applied to study the spatial dynamics of a variety of human infection settings (e.g., Mari et al. 2012; Truscott and Ferguson 2012; Gog et al. 2014).

The generalized gravity model quantifying the spatial interaction between locations i and j (commonly) take the form $\theta N_i^{\tau_1} N_j^{\tau_2} d_{ij}^{-\rho}$, where θ, τ_1, τ_2, and ρ are nonnegative parameters shaping the topology of the spatial interaction network. The gravity model has at least two important special cases: $\rho = 0, \tau_1 = \tau_2 = 1$ representing a mean field model and $\tau_1 = \tau_2 = 0$ representing simple spatial diffusion.

Viboud et al. (2006) proposed a stochastic multipatch SIR model for the spread of seasonal influenza among the states of the continental USA. We will consider a simpler SIR version of the model (ignoring susceptible recruitment)[5]:

$$\frac{dS_i}{dt} = -(\beta I_i + \sum_{j \neq i} \iota_{j,i} I_j) S_i \tag{11.1}$$

$$\frac{dI_i}{dt} = (\beta I_i + \sum_{j \neq i} \iota_{j,i} I_j) S_i - \gamma I_i \tag{11.2}$$

$$\frac{dR_i}{dt} = \gamma I_i, \tag{11.3}$$

where $\iota_{j,i} I_j$ is the gravity-weighted force of infection exerted by state j on state i. The corresponding R-code is:

```
require(deSolve)
SIR.space = function(t, y, pars) {
    i = c(1:L)
```

[5] Note that we assume that spatial transmission does not dilute local transmission. Keeling and Rohani (2002) provide a discussion of this issue.

```
    S = y[i]
    I = y[L + i]
    R = y[2 * L + i]
    with(pars, {
        beta = beta[i]
        dS = -(beta * I + m * G %*% I) * S
        dI = (beta * I + m * G %*% I) * S - gamma *
            I
        dR = gamma * I
        list(c(dS, dI, dR))
    })
}
```

`G` is the spatial interaction matrix and `m` is a scaling factor. Combining state-level ILI-data with county-level commuter census data, Viboud et al. (2006) estimated the gravity parameters to be $\tau_1 = 0.3$, $\tau_2 = 0.6$, and $\rho = 3$.[6] The `usflu` data contains coordinates and populations for each of the contiguous lower 48 states plus the District of Columbia. The `gcdist`-function of the `ncf`-package generates spatial distance matrices from latitude/longitude data:

```
require(ncf)
data(usflu)
usdist = gcdist(usflu$Longitude, usflu$Latitude)
```

We define a function to generate the spatial interaction matrix given parameters and distances:

```
gravity = function(tau1, tau2, phi, pop, distance) {
    gravity = outer(pop^tau1, pop^tau2)/distance^phi
    diag(gravity) = 0
    gravity
}
G = gravity(0.3, 0.6, 3, usflu$Pop, usdist)
```

We finally define initial conditions and parameters (scaling β such that R_0 will be the same in all states). Viboud et al. (2006) were interested in exploring spread in a pandemic setting. We therefore assume that everybody is susceptible, with 1 initial index case arriving in New York:

```
gamma = 1/3.5
R0 = 1.8
beta = R0 * gamma/usflu$Pop
```

[6] Viboud et al. (2006) showed that the commuter flows has a heavier tail than predicted by this gravity model which we, for expedience, ignore.

```
m = 1/1000/sum(usflu$Pop)
parms = list(beta = beta, m = m, gamma = gamma, G = G)
L = length(usflu$Pop)

S = usflu$Pop
R = I = rep(0, length(usflu$Pop))
I[31] = 1
inits = c(S = S, I = I, R = R)
```

We are now set to simulate a spatial SIR pandemic across the USA (Fig. 11.7):

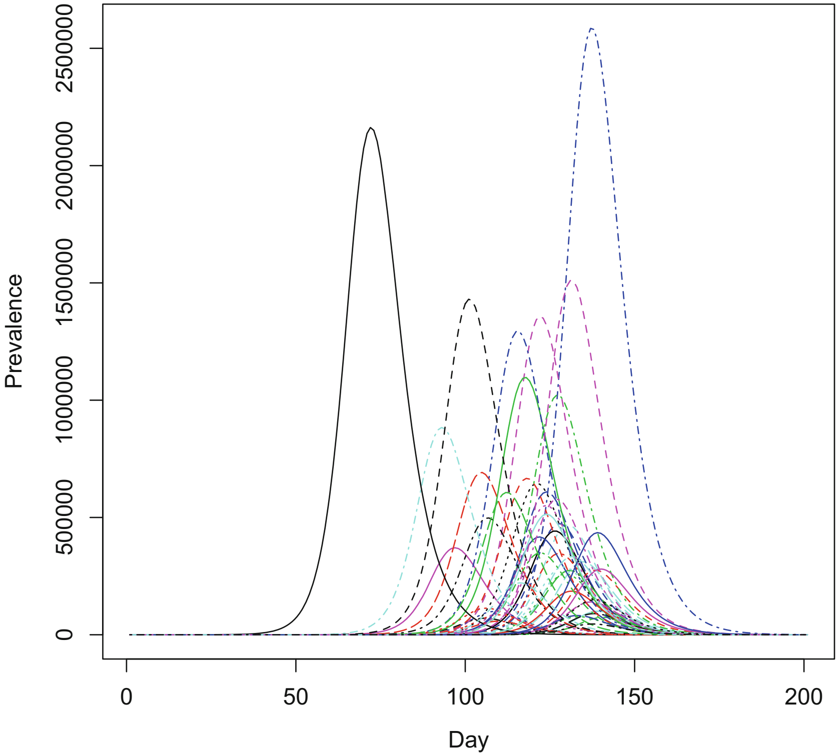

Fig. 11.7 Simulated influenza dynamics across the continental USA using a multipatch SIR model with gravity coupling parameterized according to Viboud et al. (2006)

```
require(deSolve)
times = 0:200
out = ode(inits, times, SIR.space, parms)
matplot(out[,50+(1:L)], type="l", ylab="Prevalence",
     xlab="Day")
```

The outbreak peaks are predicted to be staggered because of the spatial diffusion of the infection across the continent.