Stability and complexity in model ecosystems

Level 2 module in "Modelling course in population and evolutionary biology"

(701-1418-00)

Module author: Sebastian Bonhoeffer

Course director: Sebastian Bonhoeffer
Theoretical Biology
Institute of Integrative Biology
ETH Zürich

1 Introduction

The title of this module is in fact the title of an influential book that Robert M. May[a] wrote in the 1970's[b]. In this book May addressed the relationship between community structure in ecosystems and their properties of stability. At the time considerable evidence had been accumulated to suggest that in nature structurally complex communities tend to be more stable towards perturbations than simple ones. These observations led to an almost dogmatic view that stability results from the complexity of natural ecosystems. May challenged this view in the 1970's[c] by showing mathematically that in model ecosystems stability decreases with the number of species in a community and can therefore not be regarded as a straightforward consequence of complexity. To derive this result May investigated the stability properties of

[a] Robert M. May (now Lord May of Oxford) has had an outstanding impact on theoretical ecology. He has made seminal contributions to many areas such as chaos in biology (see also the module on the logistic difference equation), theoretical epidemiology (see modules on infectious diseases), and to stability of ecosystems (this module). Being a native of Australia, Robert May was first a theoretical physicist. Becoming the successor to Robert McArthur as a theoretical ecologist at Princeton University, Robert May moved on later to Oxford University. As one of very few scientist Robert May was made a Lord by the British Queen in 2001. Robert May has also received an honorary doctorate from the ETH Zurich.

[b] R.M. May, "Stability and complexity in model ecosystems", 2nd ed, Princeton University Press, 1974

[c] In addition to the book mentioned above, May published an article in Nature in 1972 entitled "Will a large complex system be stable?".

randomly assembled interacting species by means of eigenvalue analysis of random interaction matrices. Specifically he showed that the likelihood that an ecosystem is stable decreased with the fraction of interactions between species that are realized (connectivity) and the strength of these interactions.

May's work opened the door towards a more formal investigation of the relationship between ecosystem structure and stability. However, May's approach also has some limitations. May studied randomly assembled communities, while real communities are likely not random, and neither are species extinctions: clearly, rare species are more likely to become extinct than abundant ones. Until today the relationship between community structure and stability remains one of the major unresolved questions in ecology.

2 Multi-species Lotka-Volterra models

In this module we do not follow May's original approach, but instead will numerically simulate multi-species Lotka-Volterra models to investigate how ecosystem stability is related to size. The Lotka-Volterra model (LVM) for two species[d] is defined by

$$\begin{aligned} dn_1/dt &= r_1[1-(n_1+\gamma_{12}n_2)/K_1]n_1 \qquad (1)\\ dn_2/dt &= r_2[1-(n_2+\gamma_{21}n_1)/K_2]n_2 \qquad (2)\end{aligned}$$

where n_1 and n_2 are the densities of populations 1 and 2. The interpretation of the parameters $r_{1,2}$ and $K_{1,2}$ are equivalent to the logistic differential equation, i.e. $r_{1,2}$ are the maximal growth rates and $K_{1,2}$ are the carrying capacities for the respective species. The interpretation of parameters γ_{12} and γ_{21} becomes clear from inspecting the above equations. They describe the effects of competition between the two populations. If $\gamma_{12} > 1$, this implies that the negative effect of species 2 on species 1 is stronger than the negative effect of species 1 on itself. Hence, in this case for species 1 intraspecific competition is weaker than interspecific competition. Conversely, if $\gamma_{12} < 1$ then intraspecific competition is stronger than interspecific competition for species 1.

The multispecies LVM for N species is given by

$$dn_i/dt = r_i(1-\sum_{j=1}^{N} a_{ij}n_j)n_i \qquad \text{for } i=1,\dots,N \qquad (3)$$

The parameters r_i describe the maximal growth rates; the interactions between species are defined by the parameters a_{ij}, which describe the effect of species j on species i. If $a_{ij} > 0$ then the species j inhibits the growth of species i. The individual carrying capacity of each species is now determined by the coefficient of its own self-limitation (a_{ii}), such that $K_i = 1/a_{ii}$. Furthermore, if $a_{ij} > a_{ii}$ then the effect of species j on i is stronger that the effect of species i on itself. If all $a_{ij} \geq 0$ then the multispecies LVM describes a community with only competitive interactions (i.e. without predator-prey relations between species). Predator-prey relations are

[d] See also the script of S. Bonhoeffer's lecture "Ecology and Evolution II: Populations" on the main course web page.

characterized by interactions where $a_{ij} < 0$ and $a_{ji} > 0$. (Here species i is a predator of species j). Note, however, that for reasons of simplicity we will focus only on competitive interactions. (One of the advanced topics suggested below is to extend the approach to include predator-prey type interactions.)

2.1 Numerical integration of the multi-species LVM in R

The accompanying starting R script provides an implementation for the numerical solution of the multi-species LVM. Please consult this script in parallel to reading this section. The R script is commented extensively. The numerical simulation of the multi-species LVM can be programmed in a few lines in R (if you know how!). Essentially what you need is a function that defines the derivative of the LVM (i.e. the right hand side of equation 3). This function looks like this:

```
###
# lvm(t,x,parms)
# Use: Function to calculate derivative of multispecies Lotka-Volterra equations
# Input:
#  t: time (not used here, because there is no explicit time dependence)
#  x: vector containing current abundance of all species
#  parms:  dummy variable, which is not used here (normally used to pass on
# parameter values, but not needed here because a and r are defined globally)
# Output:
# dx: derivative of Lotka-Volterra equations at point x
lvm <- function(t,x,parms){
        dx <- r*(1 - a %*% x) * x
        list(dx)
}
```

Here, dx, r and x are vectors (whose length corresponds to the number of species) and a is an $N \times N$ matrix (where N is the number of species). The symbol $\% * \%$ denotes matrix multiplication in R. (Note that x and dx correspond to n and dn/dt in eq. 3.) The command `list` at the end of the function just returns the values of dx.

The numerical integration of the multi-species LVM can then be done using the function *lsoda* from the R library *deSolve*. A function to integrate a model such as the lvm looks for example like this:

```
###
# n.integrate(time,init.x,model)
# Use: Numerical integration of model
# Input:
#  t:  list with elements time$start, time$end, and time$steps, giving start and
# endpoint of integration and the number of time points in between
#  init.x: vector containing initial values (at time = time$start) of all species
#  model:  name of the function to integrate (here lvm)
# Output:
# data frame with n+1 columns. The first column contains the time points at which
# x is evaluated. The next n columns are the values of the n species at these
# time points
# Description:
# Generates a vector of time points for the integration and uses function lsoda
# (from library deSolve) to integrate the model
n.integrate <- function(time=time,init.x= init.x,model=model){
        t.out <- seq(time$start,time$end,length=time$steps)
        as.data.frame(lsoda(init.x,t.out,model,parms=parms))
}
```

In the function we first generate a vector of times `t.out` between `time$start` and `time$end` of length `time$steps`. Then we call the function `lsoda`, which requires initial values for the species vector x. These are specified by the vector `init.x`. The variable `model` stands for the model to integrate, which is here the function `lvm`. The command `as.data.frame` returns the output of lsoda as an R dataframe.

If we have specified all the relevant parameters and initial values then the LVM can be integrated by executing the following steps in R. First load the library *deSolve* by executing `library(deSolve)` on the R command line. This only needs to be done once at the beginning of an R session and it requires that the library *deSolve* is installed on your computer. Next you call `n.integrate(time=time,init.x=init.x,model=lvm)`, where `time` is for example defined as `time<- list(start=0,end=30,steps=100)`. Note, however, that you can only run the model provided you have specified all relevant parameters and initial values, such as the interaction matrix `a`, the growth rates `r`, the initial values for all species `init.x`, etc.

2.2 Generate and initialize parameter values

The initialization of all parameters is in principle straightforward. For example you could execute the following commands in R:

```
# Number of Species
n<-10
# Generate n uniformly distributed random values (between 0 and 1) for the
# growth rate vector
r <- runif(n)
# Generate n uniformly distributed random values for initial values of species
init.x <- runif(n)
# Generate n x n uniformly distributed random values for interaction matrix
a <- matrix(runif(n*n),nrow=n)
```

```
# Integration window
time<- list(start=0,end=30,steps=100)
# dummy variable for lvm() function defined above
parms <- c(0) ### dummy variable (can have any numerical value)
```

Note that the dummy variable `parms` is required for the integration routine, but is of no further importance.

Once you have specified these initial values you can numerically integrate the LVM by calling `n.integrate(time=time,init.x=init.x,model=lvm)`. The output is a dataframe with the columns "time", for the time points, and "1", "2", ... for the abundances of the species at these time points. The start and end time and the time intervals in between are specified by the parameter vector `time`.

2.3 Notes on generating the interaction matrix

There are several points that one needs to consider when generating the interaction matrix a_{ij}:

- The values a_{ii} have to be greater than zero, because otherwise a species might grow to infinity. Therefore, when generating the values for the interaction matrix one has to take care that the diagonal elements of the matrix (i.e. the a_{ii}) are greater than zero.
- When the interaction matrix a_{ij} is generated by drawing $N \times N$ uniformly distributed random numbers then essentially all species interact with all others (since the probability of generating a value of 0 vanishes). However, in natural ecosystems there will be many species that do not interact and therefore should have interaction parameters with value 0.

2.4 Tools to plot the output of the simulation

The R script offers several functions to plot the generated output.

- `plot.lvm.time`: This function uses the output of the function `n.integrate` to plot the time course of a simulation of the LVM.
- `plot.matrix`: This function plots the interaction matrix.
- `plot.frequency`: This function uses the output of the function `n.integrate` to plot the frequencies of all species at the end of the simulation.

Examples for plots generated by these functions are given in figure 1. To understand how these functions work in detail, please refer to the R script file.

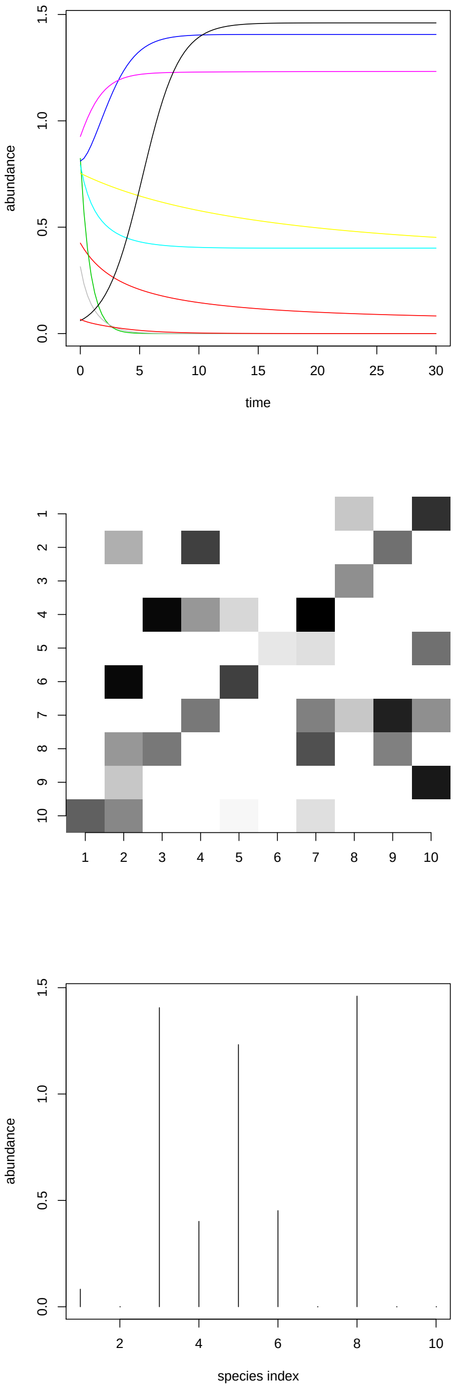

Figure 1: Plots generated by the plotting functions plot.lvm.time (top), plot.matrix (middle) and plot.frequency (bottom).

2.5 Repeated simulations

The R script also offers the possibility to study the invasion of new species. In essence this is done by repeatedly running the LVM model, testing which species have gone extinct (i.e. have fallen in frequency below a pre-defined cutoff level), and replacing these extinct species by new ones that are allowed to invade into the ecosystem (see example loop in the R script). In order to replace the extinct species with new ones one needs to update the interaction matrix, the growth rate vector, and the `init.x` vector, for the initial values of the species at the start of the integration. The function `cutoff` can be used to determine which species went extinct and which survived at the end of a simulation. The function `cutoff` returns a vector with the indices of the extinct species. This vector is used by the function `generate.parameters` to update the growth rate vector and the interaction matrix. The function `generate.parameters` has another input parameter called `sparse`, which determines the fraction of all interactions that are nonzero. This parameter can take any value between 0 and 1. A value of `sparse` of 1 implies full connectivity. The function `generate.parameters` also makes sure that there are no 0 entries in the diagonal of the interaction matrix (i.e. all species inhibit their own growth). For details please refer to the R script.

3 Exercises

3.1 Basic exercises

Eb1. How does ecosystem stability depend on size (i.e. the number of species present)? Hint: Start with a random $N \times N$ matrix, simulate the LVM and count how many species are present at the end of a (sufficiently long) simulation. Plot the number of surviving species against N.

Eb2. How does ecosystem stability depend on connectivity (i.e. the fraction of non-zero entries in the interaction matrix)?

Eb3. How can one define or measure ecosystem stability? Stability with regard to what? Think about measures that would be useful for natural systems. (This exercise is not primarily for simulation, but for discussion in order to think beyond the simulation.)

Eb4. Does the coexistence of a set of species depend on the order in which they are introduced into an ecosystem? Hint: Start with a given interaction matrix, but introduce species in different orders.

3.2 Advanced/additional exercises

Ea1. How does an ecosystem respond to the removal of a species? What is the average effect and what is the range of effects? Does the removal of an abundant species from the ecosystem have a stronger effect than the removal of a rare one? Does the removal of a species with high connectivity have a stronger effect than the removal of one with low connectivity?

Ea2. How does an ecosystem respond to the invasion of a new species? Does invasion of a new species lead to the extinction of another species? How does the effect of invasion depend on the size of the ecosystem? How does it depend on the connectivity of the invading species and the connectivity among the other species?

Ea3. What is more important for the survival of a species: its growth rate, its carrying capacity (self-limitation coefficient), the impact it has on other species, or how strongly itself is affected by the other species?

Ea4. Can "evolved" ecosystems be more complex than random ones? Hint: Start with random interaction matrix and simulate the LVM until no further species goes extinct. Then simulate repeated rounds of invasion of new (randomly parameterized) species, keeping the new steady state (of surviving species) after each round. Can the number of co-existing species increase? (The system might gain, but also lose species in each round).

Ea5. How do the stability properties change if some interactions are predatory? This is computationally an advanced topic since it requires some new programming.