

Contents

Introduction	1
------------------------	---

title: “Demographic models for Life Histories”
author: Neil McRoberts, Robin Choudhury
output: html_document

Introduction

In the lecture on May’s approach to the analysis of the community matrix, we saw that a matrix that contains coefficients for interactions or transitions can be used in a simple dynamic model to update a vector containing values representing the current state of the system captured in the matrix. In the current case, instead of thinking about the system being different possible species in an ecological community, we are going to think of the system being the life history of an organism. Specifically, we are going to consider the host plant life history and we will use a particular kind of matrix to model the impact of disease on the survival and reproduction of the plant. We’re going to do this at a superficial level, and as with most of the topics in the class, the aim here is to introduce the concepts and methods to give you a starting point if you want to take things further. Hal Caswell’s textbook¹ is the best overall source to consult if you want to read beyond the lecture notes and you’ll find a couple of papers under the topic heading on the Canvas site, along with the usual pdf, Markdown, and .R files.

Demographic matrices for populations

The community matrix model looks at the propagation of a perturbation in population sizes in effect by equating individuals of one species as fractions of individuals from other species. What sort of options are available if we are more interested in the effects of disease on the dynamics of a single species? For example, in an ecological setting we might be interested in the impact of seed-borne pathogens on seedling mortality and recruitment to the plant community, or the impacts of a range of pathogens on the probability that a plant will be able to complete its life-cycle. In an ecology or conservation setting this might have consequences for what kind of intervention we would design (or help us decide if intervention is feasible). In an agricultural setting such analyses could be used as part of the design for disease management plans, helping identify key

¹We have a couple of copies in QBE research group, so if you want to look at a copy, speak to me about getting access. It’s expensive enough that you’d want to take a look at it before deciding to buy it. I am not sure if it’s available online through the UCD library.

points in a growing season or the lifetime of an orchard when intervention will be most helpful. In all these cases, what we are concerned with is the impact of disease on the so-called *life history parameters* of a population that define its *demographic* properties. In thinking about these sorts of ideas we are also setting the stage for later in the course when we'll take a look at the use of state models for thinking about policy analysis and intervention.

Leslie and Lefkovitch Matrices

The methodology for building demographic models in matrix format was first developed by Patrick Holt Leslie who focused specifically on age classes; the method was later extended by Leonard Lefkovitch so that models could be built for more general classes based on, for example, size, or developmental stage. In a typical Leslie matrix model the time step length is chosen so that transitions always occur and there is no probability that individuals stay in the same stage for more than one time step. In a Lefkovitch model it is possible for individuals to stay in the same stage for more than one time step.

Irrespective of whether we're making a Leslie or a Lefkovitch model, the projection matrix has the same basic structure. We assume that the population can be divided into stages or classes that represent increasing age, development level or size. Births usually occur into the first class/stage only. Because the matrix represents a series of stages in an aging/development process, there is a one-way flow through the matrix (and the corresponding life history graph).

The main diagonal (top left to bottom right) of a projection matrix holds the probabilities that individuals remain in each state during the time step of the projection. As already noted, in a typical Leslie matrix application, the time step length is chosen so that these values are zero and all individuals age by one life stage (or leave the population for some reason). The transitions between successive life stages are probability values which are placed on the sub-diagonal of the matrix that runs immediately below the main diagonal.² In a Lefkovitch model, or in some Leslie models, the time step does not correspond to a span of time that guarantees individuals will definitely move between stages. In such cases, some of the values on the main diagonal will be >0 .

The first row of the matrix in these models represents the offspring life stage (e.g. in a plant life cycle model this could represent seeds, or new plants generated on runners or stolons). Any reproductive life stages that produce offspring will have *fecundity* numbers in the first row of the matrix. These fecundities are the average number of offspring produced by an individual in each stage in a time step. Note that the presence of fecundity values in the first row means that a Leslie/Lefkovitch matrix contains two completely different types of number:

²I think of the diagonal as a set of stairs going down from left to right. If individuals are marbles being born on the top left step, they progress through life by rolling down one step at a time. Each step down in the matrix represents passing from one node to the next (left to right) in the state diagram.

- State transition probabilities that capture the development (or aging process)
- Fecundity values, which are numbers of individuals.

In many life cycles only the last life stage or oldest individuals would be considered to reproduce, so the matrix would have a single value in the upper right corner of the matrix. In other life histories (most tree species, for example) multiple age classes or development stages might be capable of reproduction and so the first row of the matrix will contain several fecundity values.

Pitch canker example

To make all of this more concrete we will work with an example taken from the literature. Reynolds *et al.* (2019) looked at the effect of Pitch Canker, caused by *Fusarium circinatum* on the life history of the Monterey Pine, *Pinus radiata*, in its native range on the central coast of California. The original paper is provided on Canvas but is Open Access and can be accessed via the link above on the journal website. *F.circinatum* has different impacts on tree health depending on which life stage is infected. Seedlings and small juvenile trees can be killed outright because the pathogen can completely girdle the main stem stopping vascular movement in both directions. In older trees pitch cankers on individual branches may cause dead tips or branch die-back. These infections rarely kill a tree on their own, but might accelerate death when trees are also stressed by drought or other diseases. They do, however, reduce the production of seed cones, more or less in direct proportion to the incidence of branch “strikes” by the disease.

In its native range in coastal California and northern Mexico, *P.radiata* can be a relatively long-lived tree, surviving for up to 200 years in some cases. Older trees are typically erect and bear little-to-no foliage close to the ground, particularly when they occur in dense stands. They are partly adapted to fire and their cones will typically only release seed after heating, or after being exposed to long periods of weathering. Recruitment to tree stands is rarely limited by the availability of seeds and seedlings have a high mortality rate even in the absence of disease as a result of competition for light and lack of water. In temperate climates *P.radiata* is very fast growing and produces high quality timber that is valued for a wide range of uses. For these reasons, it is one of the most widely grown commercial timber trees in the world. The few remaining native populations are fragmented and face a range of challenges to survival. Conservation plans exist in some areas, including the Monterey peninsula of California.

Reynolds *et al.* modeled the *P.radiata* life history in 7 stages: Cones, seeds, seedlings, small saplings, medium saplings, large saplings, and mature trees. All of the sapling life stages and mature trees have fecundity values, corresponding to the number of cones produced by that type of individual in a decade. Seeds are generated in the projection matrix by a reproduction number which

corresponds to the average number of seeds per cone. In this respect the model of Reynolds *et al.* is neither a true Leslie nor Lefkovitch model. The time step for the model was assumed to be a decade. Considering this duration in relation to the life history of the tree, seedling, and small sapling stages were considered to be completed within the time step, so the transitions for these life stages are on the sub-diagonal only. The larger life stages were considered to be able to last longer than a decade and so have probabilities on the major diagonal.

First we will write the model in graph form and then in matrix form, initially leaving out the presence of the pathogen. The code from our brief look at May's community matrix concept will be a useful starting template.

±-----+

```
grViz("
  digraph pitch_canker {
    graph [rankdir=TD]
    node [shape=circle, style=filled, color=grey]
    node [fillcolor=Sienna]
    Cone
    node [fillcolor=Goldenrod]
    Seed
    node [fillcolor=YellowGreen]
    Seedling
    node [fillcolor=Green]
    SSap; MSap; LSap
    node [fillcolor=ForestGreen]
    Mature

    #edges
    edge [color=grey]
    Cone->Seed [arrowhead=vee] [label=R]
    Seed->Seedling[arrowhead=vee] [label=p(seed-seedling)]
    Seedling->SSap[arrowhead=vee] [label=p(seedling-ssapling)]
    SSap->MSap[arrowhead=vee] [label=p(ssapling-msapling)]
    MSap->LSap[arrowhead=vee] [label=p(msapling-lsapling)]
    LSap->Mature[arrowhead=vee] [label=p(LSap-Mature)]
    SSap->Cone[arrowhead=box] [label=fSsap]
    MSap->Cone[arrowhead=box] [label=fMsap]
    LSap->Cone[arrowhead=box] [label=fLsap]
    Mature->Cone[arrowhead=box] [label=fMature]

  }
")

# Analysis of the dynamics inherent in the matrix.
tmax<- 20      #At one decade per step, this is equivalent to 200 years
```

```

times<- seq(2,tmax,1)      #define a sequence of time steps
stages <-c("Cone","Seed","Seedling","SSap","MSap","LSap","Mature")    # stage labels

# define a matrix to hold the output. Each column of the matrix will be one time step
pine_proj<- matrix(nrow=length(stages), ncol=tmax, dimnames=list(stages,seq(1,tmax,1)))

pine_proj[,]<-0 #put zero in every cell of the matrix

pine_proj[,1]<-c(20,1000,1000,10,2,2,1) #define the initial population

#Make the projection matrix and fill it with interaction coefficients
A<-matrix(nrow=length(stages), ncol=length(stages), dimnames=list(stages,stages),
          byrow=TRUE, data=c(0,0,0,20,25,50,75,
                             40,0,0,0,0,0,0,
                             0,0.5,0,0,0,0,0,
                             0,0,0.01,0,0,0,0,
                             0,0,0,0.1,0.5,0,0,
                             0,0,0,0,0.5,0.6,0,
                             0,0,0,0,0,0.4,0.95))

for (i in times) { #Here we see the use of a for() loop to iterate a calculation
  pine_proj[,i]<-A%*%pine_proj[,i-1] #note that we filled the first column of
                                     #data outside of the loop, and then started
                                     #counting the loop iterations at i=2, so we
                                     #had already established the first i, i-1 pair.
}

#The next block of commands builds a line plot of each row of the projections
#The example illustrates various line options and one way to add a legend

pineplot<-plot(seq(1,20,1), pine_proj[1,], ty="l", lwd=2, col=1,
               ylab="individuals", xlab="decades")
lines(seq(1,20,1), pine_proj[2,], lwd=2, lty=2, col=2)
lines(seq(1,20,1), pine_proj[3,], lwd=2, lty=3, col=3)
lines(seq(1,20,1), pine_proj[4,], lwd=2, lty=4, col=4)
lines(seq(1,20,1), pine_proj[5,], lwd=2, lty=1, col=5)
lines(seq(1,20,1), pine_proj[6,], lwd=2, lty=2, col=6)
lines(seq(1,20,1), pine_proj[7,], lwd=2, lty=3, col=7)

legend("topright", inset=0.015, legend=stages, lty=c(1,2,3,4,1,2,3),
      col=c(1,2,3,4,5,6,7), lwd=2)

#Now we want to look at the eigenvalues of A
eigvals_A<-eigen(A)
eigvals_A

```

±-----+

If you run the projection with the code supplied you'll see that there are some oscillations in the most of the life stages of the pine until after, approximately, 100 years (decade 10). After that, the proportions of the population in the different development stages stay constant. This distribution of the population in different age groups or development stages is known as the *stable stage distribution*, *SSD*. It is a feature of matrix models of this type that they generate a *SSD*. Sometimes properties of models that are inherent to the model framework itself can be problematic³ but in this case the behavior of the model actually replicates something that is seen frequently in actual biological populations. The reason for this is that the matrix is simply capturing the basic rules about how populations grow and age and providing a convenient calculation device to process those rules; in using a projection matrix we are not really adding a lot of mathematical detail that might not correspond well to biology. The *SSD* can be found without doing the numerical projection for the matrix because it depends on the *eigenvectors*. Another useful feature of matrix demography models is that we can also calculate the *stage-specific reproductive values (SSRV)*, which are the relative contributions each life history makes to the overall reproductive output over a projection interval. Caswell gives a quite accessible account of how to calculate the *SSD* and *SSRV* from matrix *eigenvalues*. It's a useful exercise to write the code needed to do this (it only takes three or four simple lines to do this), but in R there is a package called "popbio" that makes all of these calculations (and many others) much simpler. The code needed to install popbio, run the basic analysis of the population demography features and produce a couple of simple diagnostic barplots is as follows:

±-----+

```
# install.packages("popbio")
library(popbio)
pop_stats_Pradiata<-eigen.analysis(A)
barplot(pop_stats_Pradiata$stable.stage)
barplot(log(pop_stats_Pradiata$repro.value))
```

±-----+

If we run that code for the *P.radiata* model in the absence of disease we find that (assuming the model is a reasonable reflection of reality) 98% of the *P.radiata* individuals present at any time in the *SSD* are either seeds or seedlings. Mature trees and larger saplings, which are the trees valued for their aesthetics on the Monterey peninsula comprise only about 15 in 10,000 of the individuals alive

³One well-known case of this are the Lotka-Volterra equations for predator-prey dynamics. These equations are a pair of linked differential equations (one for the predator and one for the prey) which produce out-of-phase oscillations. Unfortunately, the oscillations in the classical form of the equations are *pathological* - in this context that word means that the oscillations are an inevitable consequence of the mathematical form of the equations and do not reflect any tendency for predator-prey systems to oscillate in real life. Despite this shortcoming the Lotka-Volterra equations have been incredibly useful in ecology to help frame questions about inter-specific interactions.

in any decade. Conversely, cones, seeds and seedlings have very low *SSRV*, with medium and large saplings and mature trees accounting for almost all the reproductive value. This is not surprising since these are the stages which produce the majority of the cones. However, none of the individuals producing cones would be present if they hadn't, at some point in the past, passed through the cone, seed and seedling stages. The very large numbers of these individuals in the population, combined with their lack of reproduction, means that their *SSRV* values are low, but some of the *SSRV* for the larger (older) trees belongs to the earlier life stages. If we look at the numbers carefully we can see that the *SSRV* for mature trees is actually slightly lower than for medium and large saplings. This is a consequence of the fact that (in the model) there is no mortality for the sapling stages but 5% of the mature trees are lost each decade. *SSRV* is essentially an expected value - that is, it is the number (of offspring) that an individual in each stage could produce multiplied by the probability that those offspring *will be* produced. For saplings that don't die (in the model) reproduction is a certainty, but for mature trees that can die of old age, there is a penalty to their *SSRV* since they may die before producing all of their possible offspring. One pertinent question to ask about the influence of pitch canker in the system is whether its presence changes the SSD and *SSRV* and whether any changes have practical consequences for genetic conservation or for preservation of the value of the landscape. In order to look at those questions we have to introduce the effect of disease into the model.

Including disease in the model

F.circinatum causes damage to Monterey pine tissue when it gains access. In the model there are really only two types of life history event: stage transitions and fecundity values. The transitions implicitly capture survival (dead organisms don't progress through life history stages), so for stages that are small enough to be killed, one way we can include the impact of *F.circinatum* is to reduce the transition probabilities between states. Cankers on limbs and stems of large trees may result in branch tips or side branches dying, which reduces cone production. It is difficult to make accurate field observations of these effects directly, so Reynolds *et al.* made a simplifying assumption about the effect of disease: each life history parameter that is affected by disease results in an effect that is directly proportional to the disease incidence in that state. Incidence could be made specific to each affected parameter, but as a simple example to illustrate how to include the effect we will make the further simplification that disease incidence is the same in every state. The recipe for including this assumption in the model is shown in Table 2 in Reynolds *et al.* For any given disease incidence value, p , the life history parameter is reduced by a factor $(1-p)$ in the presence of disease.

The following snippet of code sets up the disease incidence, adjusts the projection matrix, and repeats the previous analyses.

```
±-----+
```

```

#Make the new projection matrix and and the effect of disease incidence, p.
pine_projD<- matrix(nrow=length(stages), ncol=tmax, dimnames=list(stages,seq(1,tmax,1)))
pine_projD[,]<-0 #put zero in every cell of the matrix
pine_projD[,1]<-c(20,1000,1000,10,2,2,1) #define the initial population
p<-0.05 #assumed disease incidence across the population
AD<-matrix(nrow=length(stages), ncol=length(stages), dimnames=list(stages,stages),
           byrow=TRUE, data=c(0,0,0,20*(1-p),25*(1-p),50*(1-p),75*(1-p),
                               40,0,0,0,0,0,0,
                               0,0.5*(1-p),0,0,0,0,0,
                               0,0,0.001*(1-p),0,0,0,0,
                               0,0,0,0.05*(1-p),0.5*(1-p),0,0,
                               0,0,0,0,0.5*(1-p),0.6,0,
                               0,0,0,0,0,0.4,0.90))

for (i in times) { #Here we see the use of a for() loop to iterate a calculation
  pine_projD[,i]<-AD%*%pine_projD[,i-1] #note that we filled the first column of
  #data outside of the loop, and then started
  #counting the loop iterations at i=2, so we
  #had already established the first i, i-1 pair.
}

#The next block of commands builds a line plot of each row of the projections
#The example illustrates various line options and one way to add a legend

pineplotD<-plot(seq(1,20,1), log(pine_projD[1,]), ty="l", lwd=2, col=1,
                ylab="ln(individuals)", xlab="decades", ylim=c(-2,10))
lines(seq(1,20,1), log(pine_projD[2,]), lwd=2, lty=2, col=2)
lines(seq(1,20,1), log(pine_projD[3,]), lwd=2, lty=3, col=3)
lines(seq(1,20,1), log(pine_projD[4,]), lwd=2, lty=4, col=4)
lines(seq(1,20,1), log(pine_projD[5,]), lwd=2, lty=1, col=5)
lines(seq(1,20,1), log(pine_projD[6,]), lwd=2, lty=2, col=6)
lines(seq(1,20,1), log(pine_projD[7,]), lwd=2, lty=3, col=9)

legend("right", inset=0.015, legend=stages, lty=c(1,2,3,4,1,2,3),
      col=c(1,2,3,4,5,6,9), lwd=2)
pop_stats_PradiataD<-eigen.analysis(AD)
barplot(pop_stats_PradiataD$stable.stage)
barplot(log(pop_stats_PradiataD$repro.value))
±-----+

```

Given the other parameter values in the model, a disease incidence of 10% is high enough to make the dominant eigenvalue for the population dip below 1. This puts the population in a long-term downward trajectory. Qualitatively including disease in this way does not appear to affect the structure of the population much, but the reproductive value of seeds and seedlings is further reduced (because they can be killed outright) while the importance of larger

individuals increases.

Exercises to try

At the moment the model can only include the effect of a single, constant value for disease incidence. One extension to the model would be to make a different disease incidence value for each time and look at the effect of changing disease on the overall outcome. Alternatively, you could expand the model to do what is called a sensitivity analysis by looking at the effect of systematically repeating the analysis for a range of different disease incidence values between 0 and 1. One way to do this is to put the current model inside another *for* loop. This presents some interesting challenges about how to store the results and which things to keep track of as the analysis churns through the combinations.

Written with StackEdit.