Density dependence in structured populations - Tutorial in R

The matrix analyses for the population projection allow great flexibility to simulate different scenarios in populations. It may, for example, include density effects on a stage of the population and the opposite effect in another (positive or negative density dependence) or even include decreased survival and increased fertility in a stage. Here, we will include the density effect on the adult class permanence probability. Let's use an equation in which $ P_{33} $, the permanence in the adult class, is a fraction of the transition probability when $N = 0$. One possibility is the function:

$$ P_{33} = P'_{33} * \frac{h}{h+N_3}$$

What does this function mean? Let's see what happens with the expression $\frac{h}{h+N_3}$:

```
des<-function(n,h){h/(h+n)}
des.mat<-outer(seq(1,10,by=0.1), seq(1,10,by=0.1), des)
    filled.contour(x=seq(1,10,by=0.1),y=seq(1,10,by=0.1),des.mat,  color =
terrain.colors, xlab="N", ylab="h", main="Fraction of the maximum rate")
```

Exercise

- 1. Change the ranges of n and h and see what is the result.
- 2. Can h be negative? What would be the biological interpretation for that?

Real world scenario

We will work now with data[1)] from a population of a threatened species of cactus (*Coryphantha robbinsorum*) that occurs in limestone outcrops in Arizona and neighboring areas in Mexico. Its life stages were defined similarly to the previous example: small juveniles, large juveniles and adults, and only the adult reproduces.

```
#############################
### Cory transition matrix ##
#############################
cory<-matrix(c(0.434, 0.333,0,0,0.61, 0.304,0.56, 0, 0.956), ncol=3, nrow=3)
cory
```

The next function incorporates directly the density dependence in the transition matrix. See how the *cory* population behaves!

```
ddf<-function(n, mat, h, st)
{
    mat[st, st]<-mat[st,st]*(h/(h+n[st]))
        return(mat)
}
cory
# starting population
n0=matrix(c(10,5,2), ncol=1)
## time 1
    n1 =  cory %*% n0
    n1
    n1d= ddf(n0,cory,h=10, st=3)%*%  n0
    n1
    n1d
## time 2
    n2 =  cory %*% n1
    n2
    n2d= ddf(n1,cory,h=10, st=3)%*%  n1
    n2
    n2d
## time 3
    n3 =  cory %*% n2
    n3
    n3d= ddf(n2,cory,h=10, st=3)%*%  n1
    n3
    n3d
```

Let's encapsulate this function in a more general one, that will enable us to project the population for every time step:

```
################################
## density-dependent function ##
################################
proj.dd<-function(n0, matproj,h, st, tmax=10)
{
    res.mat<-matrix(NA,nrow=tmax+1,ncol=length(n0))
        res.mat[1,]<-n0
        for(i in 2:(tmax+1))
        {
            res.mat[i,]=ddf(res.mat[(i-1),], matproj,h, st)  %*%
res.mat[(i-1),]
        }
    return(res.mat)
}
## tmax = 20
res.corydd<-proj.dd(n0,matproj=cory, h=100, st=3, tmax=20)
    res.corydd
    matplot(0:20,res.corydd, type="l")
prop.estdd<-res.corydd/apply(res.corydd,1,sum)
    matplot(0:20,prop.estdd, type="l", lty=2:4, col=2:4)
## tmax = 100
```

```
res.corydd<-proj.dd(n0,matproj=cory, h=100, st=3, tmax=100)
    res.corydd
    matplot(0:100,res.corydd, type="l", ylab="N")
prop.estdd<-res.corydd/apply(res.corydd,1,sum)
    matplot(0:100,prop.estdd, type="l", lty=2:4, col=2:4, ylab="Class
proportion")
```

Exercise

- 1. Increase the projection time to (tmax=100) in the example above and compare with a population that does not present density dependence. Show graphs!
- 2. Change the value of h (both increasing and decreasing its value) and see what is the effect of this change in the population trajectory.
- 3. Does the initial population affect the projection for a long time in the future? Is there any similarity between this model and the model that does not present density dependence?

Eigenvectors and eigenvalues

The eigenvalues/eigenvectors analysis can be done to linearly transform matrices and allow for summaries of multivariate data. We represent them as:

$ Aw = \lambda w $

Here, A is the matrix, w is the eigenvector and λ is the eigenvalue. See this link for a more in-depth explanation.

What happens is that often in Ecology we hear of eigenvalues and eigenvectors. They appears in: multivariate analysis of ordination; stability analysis with one or more species and matrix population projections! For our case of structured population growth, we can find the asymptotic $ \lambda $ simply finding the dominant eigenvalue of the projection matrix. Eigenvalues are represented by $ \lambda $ and correspond to the solution to the equation above. The dominant eigenvalue is the one with the highest absolute value and is often a complex number. In projection matrices, the $ \lambda_{1} $ (i.e., the dominant eigenvalue) is always positive and real. We can use eigenvalue analyses to find the $ \lambda_{1} $ as if by magic. To do operations with complex numbers in R will use the Re function.

Summarizing...

The eigenvector of a matrix is a scalar value [2)] that is equivalent to the matrix when multiplying one specific vector. The dominant eigenvalue of our projection matrix is our asymptotic growth rate: λ.

To understand all that, we will redo the graph of the asymptotic population grow!

```
par(mfrow=c(2,2))
res.cory<-proj.mat(n0,matproj=cory, tmax=100)
res.cory
matplot(0:10,res.cory[1:11,], type="l")
```

```
prop.est<-res.cory/apply(res.cory,1,sum)
matplot(0:10,prop.est[1:11,], type="l", lty=2:4, col=2:4)
legend("topleft", lty=2:4, col=2:4, legend=c("small", "large","adults"))
matplot(0:100,prop.est, type="l", lty=2:4, col=2:4)
legend("topleft", lty=2:4, col=2:4, legend=c("small", "large","adults"))
lamb.seq=res.cory[2:101]/res.cory[1:100]
plot(1:100, lamb.seq,type="l", lty=2)
par(mfrow=c(1,1))
```

Now we will figure out the eigenvalues of the matrix, and compare the dominant one with the asymptotic growth rate.

```
eigen.cory=eigen(cory)
lamb=max(Re(eigen.cory$values))
lamb
abline(h=lamb,col="red")
```

In other words, the intuitive way to find the λ is by iterating the population growth for a long time, as with increasing t the growth rate $R(t)$ will approach λ_1. Another, much more direct way, is by calculating the dominant eigenvalue of the transition matrix.

Stable stage distribution

The relative abundance of the different life stages is called stage distribution. For a structured population in which demographic rates are constant, the stage structure will reach a stable stage distribution. The population can grow (as expected by the exponential growth), but the relative abundance at each stage remains constant. And we have seen this before …

So, back to the eigenvalues and eigenvectors, we can use them to find the stable stage distribution. Now what will give us the relative abundances is the dominant eigenvector, w. The dominant eigenvector is in the same position of the dominant eigenvalue. Then we can extract the w, maintaining its real part divided by its sum.

Returning then to our matrix A with hypothetical data from a population of palm trees:

```
aval.A <- eigen(A)
w <- Re(aval.A$vectors[,1]) # the position "1" is corresponding to the
dominant eigenvalue
w
round(w/sum(w), 3)
```

Exercise

Get back to our projection N(t) (based on matrix A) and see after which year the proportion of individuals per life stage fits the distribution found above. Remember that the values are stored in the "N.projected" object.

```
# going back...
matplot(0:years, log(t(N.projected)), type = "l", lty = 1:3, ylab = "n(t)",
```

```
xlab = "Time (t)", xlim=)
legend("topleft", legend = c("Seed", "Juvenile", "Adult"),lty = 1:3, col =
1:3, bty = "n")

# and remember:
colnames(N.projected) <- paste("N", 0:10, sep="")
rownames(N.projected) <- c("seed","juvenile", "adult")

N.projected
```

Reproductive value

The structure of stable stage distribution gives us a measure of the importance of the stage (in terms of relative abundance) in the structured population growth. In the case of reproductive value we have a measure of the importance of the individual in each stage, i.e., the reproductive value is the expected contribution of each individual in present and future reproduction. We consider all individuals in a stage as having the same reproductive value. Using linear algebra we find the reproductive value by solving:

$ vA = \lambda v $

$ VR = \frac{\nu_1}{\sum^{S}_{i=1}{\nu_1}} $

We can get the left eigenvalue performing an eigenvalue analysis in the transposed projection matrix. The positions of the dominant left and right eigenvalues are the same. We extract only the left eigenvector and normalize it so that the reproductive value of the first stage is 1.

Again with our “A” matrix:

```
M <- eigen(t(A))
M
v <- Re(M$vectors[, which.max(Re(M$values))])
VR <- v/v[1]
VR
```

Exercise

How can you interpret the reproductive values found?

To learn more

Gotelli, N. J. 2007. Ecologia. Cap.3- Crescimento Populacional Estruturado. Pp. 49-82. Ed. Planta.

Gurevitch, J, Scheiner, S.M, Fox, G.A. 2009. Ecologia Vegetal. Cap. 5 - Ed. Artmed, São Paulo.

Freckleton, R.P., Silva Matos, D.M., Bovi, M.L.A & Watkinson, A.R. 2003. Predicting the impacts of harvesting using structured population models: the importance of density-dependence and timing of harvest for a tropical palm tree. Journal of Applied Ecology, 40: 846-858.

Silva Matos, D.M., Freckleton, R.P. & Watkinson, A.R. 1999. The role of density dependence in the population dynamics of a tropical palm. Ecology, 80: 2635-2650.

R, uma população, população estruturada
[1)]

Cochran & Ellner, 1992
[2)]

see the definition in https://en.wikipedia.org/wiki/Scalar_%28mathematics%29