Structured models

Ben Bolker

September 24, 2017

Structured models. linear only. (Unrealistic but very useful for understanding short-term growth/growth when rare . . .)

e.g. *Leslie matrix* model: **A**=0 except for first row (fecundities, ≥ 0) and first subdiagonal (survival, $0 < a_{i+1,i} \leq 1$). *Lefkovitch* model: the same except that more column entries can be non-zero (columns add to ≤ 1).

Irreducible non-negative matrix: For every ij there exists k such that $(\mathbf{A}^k)_{ij}$ is not zero. (There are no "dead-end", or absorbing stages. At some point, individuals in any stage will have 'offspring' in any other stage.)

Primitive matrix: There exists k such that for *all* ij, $(\mathbf{A}^k)_{ij}$ is positive. (Equivalently: the greatest divisor of all loop lengths is 1. We cannot have multiple cycles running in parallel.)

Perron-Frobenius theorem: for a non-negative, irreducible, primitive matrix, there is a single positive largest root; its associated eigenvector is the only positive (all-equal-sign) eigenvector. (If the matrix is not primitive then the largest root may be repeated — i.e. out-of-phase cohorts.)

If the max. modulus of the eigenvalues $\max |\lambda|$ (also called *spectral radius*) > 1, then the population grows geometrically in the long-term; the associated eigenvalue is called the *stable age (stage) distribution*. (Transients may be different; e.g. start with no adults.) (Note eigenvectors can be scaled any way you like: $\sum e_i = 1$ is convenient for interpretation, Python scales so that $\sum e_i^2 = 1$.)

Teasel model:

```
import numpy as np
X = np.array(((0,0,0,0,0,322.38),
              (0.966,0,0,0,0,0),
              (0.013,0.010,0.125,0,0,3.448),
              (0.007,0,0.125,0.238,0,30.17),
              (0.008,0,0,0.245,0.167,0.862),
              (0,0,0,0.023,0.75,0)))
start = np.array((1,0,0,0,0,0))
x200 = (np.linalg.matrix_power(X,200)).dot(start)
np.round(x200/np.sum(x200),3)
e1 = np.linalg.eig(X) ## eigenvalues/vectors
np.round(e1[0][0],3)    ## pop. growth
rate
## speed of approach to stable distribution:

##    ratio of first two eigenvalues
## snames <- c("dseed1","dseed2",
##             "smros","medros","lgros","flow")
np.abs(e1[0][1]/e1[0][0])
e1[1][:,0]/np.sum(e1[1][:,0])
```

http://www.stanford.edu/~jhj1/teachingdocs/Jones-Leslie1-050208.pdf