Ecosystem services

Bruno Melati

2025-08-27

Calculating ecosystem services

Each species i can contribute to the existence of a j ecosystem service. Therefore, we can map the relation species / ecosystem services as follows:

$$E = \mathbf{P}\vec{x}$$

where, $\mathbf{P}$ is a matrix of $k \times s$ dimension, in which k is the ecosystem services vector and s is the species vector and it describes the effects of one individuals species to ecossystem service. For our pourpose, we can deal with the efect of species i on ecosystem services k. The vector $\vec{x}$ is related to the state of species (present, absent, abundance...). Each entry P_{ij} depicts the effect of one individual species j on each ecosystem service i. The P_{ij} value could be the raw value or the relative value of each species i contribution to the ecosystem service. For our purpose,

$$E_{ij} = \begin{cases} P_{ij}, & \text{if } x_i = 1 \\ 0, & \text{if } x_i = 0 \end{cases}$$

Therefore, the effects of species loss could be investigated as $\Delta E = E^{(t_0)} - E^{(t_n)}$.

Running an example

```
set.seed(12)
```

```
#P matrix
#species state


s <- 10 #n species
state = rbinom(10, 0:1, prob = 0.5) #species state (0,1)
k <- 3 #n ecosystem services
P <- matrix(runif(k*s, 0, 1), nrow = s, ncol = k)
P
```

```
##              [,1]       [,2]      [,3]
##  [1,] 0.033895622 0.43933432 0.2679436
##  [2,] 0.178785004 0.45760715 0.5047680
##  [3,] 0.641665366 0.54070755 0.1885869
##  [4,] 0.022877743 0.66567983 0.4394293
```

```
##  [5,] 0.008324827 0.11269894 0.6698193
##  [6,] 0.392697197 0.21836717 0.2408832
##  [7,] 0.813880559 0.78783635 0.8932649
##  [8,] 0.376248455 0.09785304 0.8827564
##  [9,] 0.380812184 0.70983047 0.8140633
## [10,] 0.264918378 0.21782304 0.6332646
```

```
#Ecosystem services status

E <- P*state
E
```

```
##             [,1]      [,2]      [,3]
##  [1,] 0.00000000 0.0000000 0.0000000
##  [2,] 0.00000000 0.0000000 0.0000000
##  [3,] 0.00000000 0.0000000 0.0000000
##  [4,] 0.02287774 0.6656798 0.4394293
##  [5,] 0.00000000 0.0000000 0.0000000
##  [6,] 0.39269720 0.2183672 0.2408832
##  [7,] 0.00000000 0.0000000 0.0000000
##  [8,] 0.00000000 0.0000000 0.0000000
##  [9,] 0.00000000 0.0000000 0.0000000
## [10,] 0.00000000 0.0000000 0.0000000
```

```
#We can see the value of each ecosystem service
status <- colSums(E)
status
```

```
## [1] 0.4155749 0.8840470 0.6803126
```

```
#Delta E

(state_t0 = rbinom(10, 0:1, prob = 0.8)) #species state (0,1)
```

```
##  [1] 0 0 0 1 0 0 0 1 0 1
```

```
(state_tn = rbinom(10, 0:1, prob = 0.3))#species state (0,1)
```

```
##  [1] 0 0 0 0 0 1 0 0 0 1
```

```
(P0 <- P*state_t0)
```

```
##             [,1]       [,2]      [,3]
##  [1,] 0.00000000 0.00000000 0.0000000
##  [2,] 0.00000000 0.00000000 0.0000000
##  [3,] 0.00000000 0.00000000 0.0000000
##  [4,] 0.02287774 0.66567983 0.4394293
##  [5,] 0.00000000 0.00000000 0.0000000
##  [6,] 0.00000000 0.00000000 0.0000000
##  [7,] 0.00000000 0.00000000 0.0000000
##  [8,] 0.37624846 0.09785304 0.8827564
##  [9,] 0.00000000 0.00000000 0.0000000
## [10,] 0.26491838 0.21782304 0.6332646
```

```
(Pn <- P*state_tn)
```

```
##            [,1]      [,2]      [,3]
##  [1,] 0.0000000 0.0000000 0.0000000
##  [2,] 0.0000000 0.0000000 0.0000000
##  [3,] 0.0000000 0.0000000 0.0000000
##  [4,] 0.0000000 0.0000000 0.0000000
##  [5,] 0.0000000 0.0000000 0.0000000
##  [6,] 0.3926972 0.2183672 0.2408832
##  [7,] 0.0000000 0.0000000 0.0000000
##  [8,] 0.0000000 0.0000000 0.0000000
##  [9,] 0.0000000 0.0000000 0.0000000
## [10,] 0.2649184 0.2178230 0.6332646
```

```
colSums(P0)
```

```
## [1] 0.6640446 0.9813559 1.9554503
```

```
colSums(Pn)
```

```
## [1] 0.6576156 0.4361902 0.8741478
```

```
delta <- P0 - Pn
delta
```

```
##              [,1]        [,2]       [,3]
##  [1,]  0.00000000  0.00000000  0.0000000
##  [2,]  0.00000000  0.00000000  0.0000000
##  [3,]  0.00000000  0.00000000  0.0000000
##  [4,]  0.02287774  0.66567983  0.4394293
##  [5,]  0.00000000  0.00000000  0.0000000
##  [6,] -0.39269720 -0.21836717 -0.2408832
##  [7,]  0.00000000  0.00000000  0.0000000
##  [8,]  0.37624846  0.09785304  0.8827564
##  [9,]  0.00000000  0.00000000  0.0000000
## [10,]  0.00000000  0.00000000  0.0000000
```

```
colSums(delta)
```

```
## [1] 0.006429002 0.545165702 1.081302481
```

Beta distribution

To set the costs and benefits parameters we chosed a **beta distribution**. This distribution was chosed for some reasons:

- The values are all positive. As do not exist negative cost or negative benefit, it fit to our data;
- The values are defined on the inverval [0,1]. It could be used as the percentage of costs and benefits;

- As the values are bounded between 0 and 1, we know the maximum value of benefits for one species, that is $B \leq k_i$, where k is the degree of i;
- The shape of the curve assume diffent forms as a function of the paramters. In nature we really do not know the shape of the distribuitions of costs and benefits;
- It have a solution for ou case, which the other option of distribution (lognormal) do not.

The beta distribution have two parameters α abd β and both control the shpae of the distribution. Therefore,

$$X \approx Beta(\alpha, \beta)$$

where, $\alpha > 0$ and $\beta > 0$. Both parameters control de "influence" of of the shape bias. Higher α values, higher the concentration towards the right side of the curve. Higher β values, higher the concentration towards the left side of the curve.

The mean and variance of Beta depends on α and β parameters, as:

$$\bar{x} = \frac{\alpha}{\alpha + \beta}$$

$$\text{Var}(X) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

and

$$E[X] = \frac{\alpha}{\alpha + \beta}$$

```r
# Define the function to integrate
  func_to_integrate <- function(x, alpha, beta) {
    x * dbeta(x, shape1 = alpha, shape2 = beta)
  }

  # Set your alpha and beta values
  alpha_val <- 2
  beta_val <- 3

  # Perform the numerical integration
  integration_result <- integrate(func_to_integrate, lower = 0, upper = 1, alpha = alpha_val, beta = b

  # The expected value is the 'value' element of the result
  expected_value <- integration_result$value
  print(expected_value)
```

```
## [1] 0.4
```