socialecological: Simulations of livestock resilience model with caspr

Florian D. Schneider

2016-07-14

Contents

The livestock resilience model

The model was designed as a simplification of our previously developed grazing model. Besides reflecting the mechanisms of positive local feedbacks, local facilitation and associational resistance, it also implements negative local feedbacks, such as competition and attractant-decoy effects.

It is a cellular automata model with cells that can take two possible cell states: occupied by vegetation (state: `1`) and unoccupied bare soil (state: `0`). Transition of cell states are defined as probabilities which are functions of the global cover ρ_1 (i.e. fraction of occupied cells) and the local cover $q_{1|1}$ (i.e. the

1

proportion of occupied cells in the direct 4-cell neighborhood of the focal plant, i.e. an occupied cell). At each time step, these probabilities are evaluated against random numbers (i.e. the model is a probabilistic cellular automata) to see if a transition happens.

model definition

For vegetation mortality, M, we assume intrinsic plant mortality, i.e. the inverse of average lifespan, and add the mortality due to grazing risk as a classic type II functional response.

$$M(\rho_1) = m\rho + \frac{a\rho_1 L}{1 + ah\rho_1} \,,$$

where m denotes the intrinsic mortality rate, a and h are the search efficiency and handling time of the nonlinear functional response (saturating at $1/h$) which is multiplied with the livestock density, L. Note that livestock density is defined in the unit number of grazers per area $[\frac{ind.}{ha}]$. Search efficiency a is given in area browsed in per time and grazer individual $[\frac{ha}{y \times ind.}]$, and handling time h is given in time per area consumed $[\frac{y}{ha}]$, which alltogether defines feeding rate as an area consumed per year by a single individual $[\frac{ha}{y \times ind.}]$. The overall change of cover, $M(\rho)$, is thus independent from the size and unit of the observed area $[\frac{1}{y}]$.

The vegetation growth, G, is defined as a logistic growth function.

$$G(\rho_1) = r\rho_1(1 - \frac{\rho_1}{K}) \,,$$

where the intrinsic growth rate, r, is neutralized when cover approaches the global carrying capacity, K. This reflects the global competition of plants fo resources of homoegeneous distribution, such as space: when all habitable space is occupied, plant growth falls to zero.

Global feedback mechanisms such as water runoff (negative effect on growth at low cover) or plant refuges against grazing for instance due to fencing (type III funuctional response) are added to the model using the following substitutions.

water runoff ...

Fencing will be leading to a sigmoidal functional response ...

local feedbacks

The basic model assumes mortality and growth to depend only on the global vegetation cover, and neglects their dependence on spatially-explicit effects. Here, we add plant-plant interactions at the local scale by taking the local cover, q, in the direct vicinity of plants into account. By adding such interactions at the local level into the functions of growth and mortality, patterns emerge that act as positive or negative feedbacks on vegetation cover.

For vegetation growth we assume that the local environmental suitability is enhanced by local facilitation (Milchunas et al, Kéfi et al 2008). The reduction in growth rate, r, by aridity, i.e. the inverse of environmental quality b, is compensated by a function of $q_{0|1}$, the local density of cells in state 1 (i.e. vegetation) given that the focal cell is in state 0 (i.e. empty), and maximizes to one if the cell has four neighbors.

$$b = b^* + (1 - b^*) f q_{0|1} \,,$$

with the effect of aridity, a_0 in absence of local vegetation being increased to the value of f if additive facilitation due to the presence of neighboring plants occurs. This term gradually determines the enhancement of the growth with an increasing local vegetation cover (maximizing at $q_{0|1} = 1$ if the cell has 4 neighbors).

As an opposing effect, we assume that local competition of plants inhibits colonisation locally by depleting nutrients or light. As cover increases and the interspace areas are closing in, the space available for rejuvenation approaches zero. This is assuming that growth is diminished locally by competition, c. This affects carrying capacity rather than growth rate itself.

$$K = K^*(1 - c q_{0|1})$$

That is, if a cell has a fully vegetated local neighborhood, it's carrying capacity will be reduced by value c ($c \geq K$).

Regarding plant mortality, we implement two interactions into the model. In sparse environments, high local cover will be attracting grazers and increase mortality of plants locally. We assume this attractivity enhances search efficiency, a, locally by the value v

$$a = a^* + v q_{1|1}$$

Thus, at low local cover feeding will be high on plants with neighbors, whereas at high cover the term has no effect since handling time is limiting consumption. An enhanced search efficiency will raise the critical thresholds for a collapse.

Opposing to that mechanism, plants in grazed habitats develop protective traits, such as thorns or cushion growth. They thereby provide protected habitat to their direct neighborhood, or share the investments in those traits with their neighbors. Overall grazing mortality thus is reduced through an increase in handling time by associational resistance, p, which affects the perceived local density of livestock.

$$a = a^{**}(1 - pq_{1|1})$$

Combining the attractant decoy and the associational resistance effect yields

$$a = (a^* + vq_{1|1})(1 - pq_{1|1})$$

Since $q_{1|1}$ and $q_{1|0}$ are spatially explicit expressions, these equations are referring to the situation at a particular location in space which is currently in one state or the other. The substitution into $M(\rho)$ and $G(\rho)$ yields transition probabilities, i.e. the probability for death and colonization for the given location, based on the global vegetation cover ρ_1 and the local vegetation cover $q_{1|1}$ or $q_{1|0}$ for vegetated or empty locations, respectively:

$$M(\rho_1, q_{1|1}) = w_{1,0} = m + \frac{(a^* + vq_{1|1})(1 - pq_{1|1})L\rho_1}{1 + (a^* + vq_{1|1})(1 - pq_{1|1})h\rho_1}$$

$$G(\rho_1, q_{1|0}) = w_{0,1} = \frac{r\rho_1(b^* + (1 - b^*)fq_{0|1})(1 - \frac{\rho_1}{K^*(1 - cq_{0|1})})}{1 - \rho_1}$$

Parameters

parameter	default value	unit	definition
r	1.0	$[\frac{1}{y}]$	max. reproduction rate of vegetation per year
b	0.5	unitless	environmental quality, a factor that indicates quality of envir
K	0.9	unitless	carrying capacity of the system, a landscape specific value th
sigma	0	unitless	environmental noise, standard deviation of random noise on
alpha	0		water runoff

parameter	default value	unit	definition
f	0	unitless	local facilitation, positive effect of plants on the colonization
c	0	unitless	local competition, negative effect of plants on colonization pr
m	0.05	$[\frac{1}{y}]$	intrinsic mortality of plants (inverse of av. lifespan)
q	0		hill exponent of functional response
v	0	unitless	attractant-decoy effect, negative effect of plants on mortality
p	0	unitless	associational resistance, positive protection effects of plants o
L	2	$[\frac{ind.}{ha}]$	Livestock density
h	10	$[\frac{y}{ha}]$	handling time, time required to consume one hectar of vegeta
a	5	$[\frac{ha}{y \times ind.}]$	search efficiency of livestock,

Model implementation

Two spatially-explicit implementations of this model do exist and are contained in this repository: A model object for the cellular automata framework caspr for R, as well as a convenient dynamic implementation for Netlogo.

The caspr package

Installation

Installation instructions and further information can be found on the package repository website on Github and the package vignette.

In short, you only need a running version of R installed on your computer. All package dependencies will be installed automatically. Launch R and run:

```
install.packages("devtools")
devtools::install_github("fdschneider/caspr")
devtools::install_github("cascade-wp6/socialecological")
```

Then, once installed, you will have to call the library each time you start a new R session

```
library(caspr)
library(socialecological)
```

```
##
## Attaching package: 'socialecological'

## Das folgende Objekt ist maskiert 'package:caspr':
##
##     livestock
```
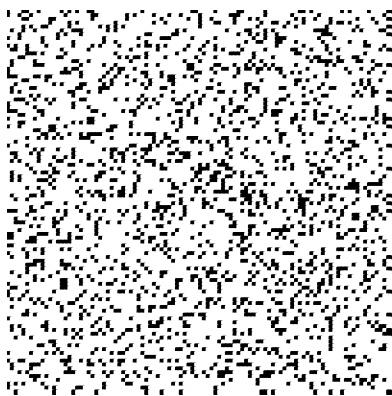
run simulations

The package is capable of running a couple of different models. For our project, I added the livestock resilience model to the package. You can review it's parameters and specifications by typing `?livestock`.

To run a model simulation, you have to provide an initial landscape object, i.e. a grid of cells that matches the specifications of the model.

```
l <- init_landscape(states = c("1","0"), cover = c(0.2,0.8), width = 100)
par(mar = c(0,0,0,0))
plot(l)
```

```
## Warning in plot.landscape(l): No colors provided for landscape plotting:
## choosing ad-hoc grey levels
```


The option `width` allows to increase or decrease the landscape size (now $100x100$ cells). Then, you can start a simulation using the default parameters (returning a warning at the end of the simulation) with

```
run <- ca(l, model = livestock, parms = livestock$parms)
plot(run)
```
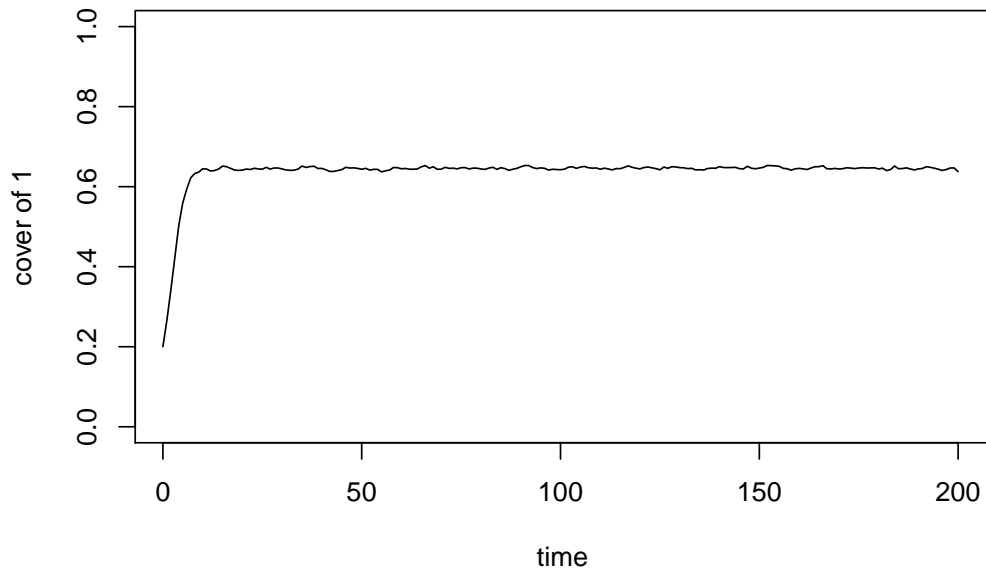


```
summary(run)
```

```
## $name
## [1] "Livestock resilience model"
##
## $time
## [1]   0 200
##
## $mean_cover
##        1        0
## 0.64481 0.35519
##
## $sd_cover
##             1           0
## 0.003720349 0.003720349
```

```
##
## attr(,"class")
## [1] "ca_summary"
```

The function `ca()` runs the simulation. This might take between 10 seconds and a couple of minutes, depending on your computer.

The `plot()` function plots the timeseries of the vegetation cover (cover of cell state "1") and the `summary()` function returns the mean cover of all potential cell states.

Of course, you will want to modify the parameters (remember that `?livestock` shows you the description of all parameters):

```
pp <- livestock$parms
pp$b <- 0.1
pp$p <- 0.2
pp$L <- 12
run <- ca(l, model = livestock, parms = pp, t_max = 200)
plot(run)
```


8
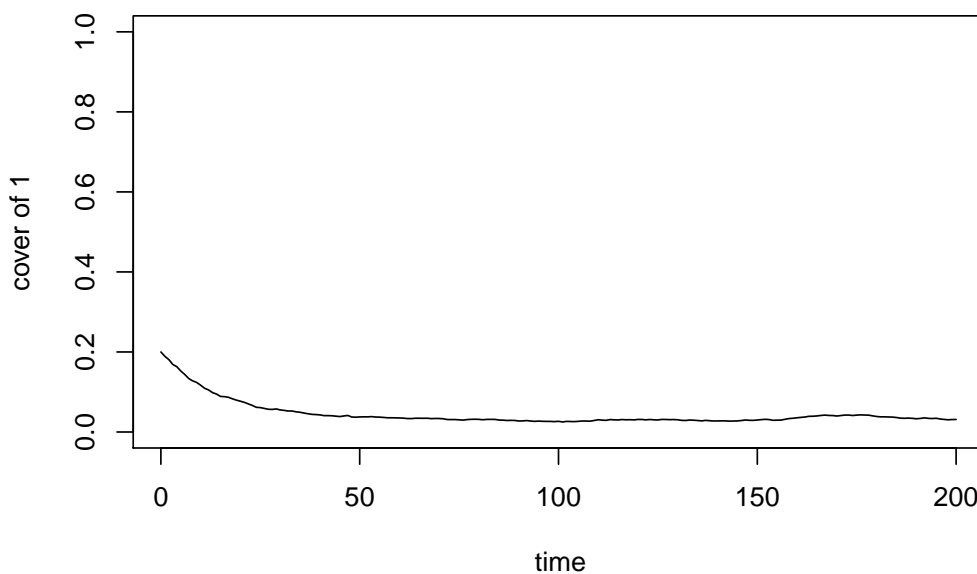
`t_max` sets the maximal number of timesteps. More options for the function `ca()` are available by calling `?ca`.

access the simulation data

If you want to analyse the timeseries of vegetation cover you can extract it from the output object of the simulation, which is now stored in the R variable `run`.

For example you can access the parameters that were used to run the model by calling

```
as.data.frame(run$model$parms)
```

```
##   r b sigma f alpha   K   c    m   v   p L q  h   a
## 1 1 1   0.1 0   0.8   0 0.9 0.2 0.05 0.8 0.2 12 0 50 0.2
```

You also can extract the vegetation cover for each single timestep and save it into an R object. To analyse variation over time, we usually would only look at a period before the end of the simulation.

```
t <- tail(run$time,100)
cover <- tail(run$cover[['1']], 100)

summary(cover)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02510 0.02920 0.03110 0.03271 0.03568 0.04280
```

There are further functions currently under development that allow to run a whole array of parameters. I will keep you updated about this.

contribute

You always can check the current state of the development for the caspr package and report issues or bugs on the GitHub issue tracker.