scfm

Ian Eddy

December 18, 2018

Overview

Fire is modelled over a grid of raster cells (pixels) and is treated as a three stage stochastic process of ignition, escape from the cell of origin, and subsequent spread. The original fire model is described by Cumming et al. (1998), and more accessibly in Armstrong and Cumming (2003), and was recently implemented as a collection of SpaDES modules by Cumming, McIntire, and Eddy (in prep). Each fire module corresponds to one of the processes being modelled). The Ignition stage determines if a fire actually starts in a particular grid cell in a given model time step. The Escape stage models the effect of fire suppression, or other ecological or sampling effects that alter the lower end of the fire size distribution. This is simulated in the model by determining a probability that a fire stays within the cell of origin, and accounts for lakes and other non-flammable geographic features. The Spread stage simulates fire growth, using a spread parameter that determines the probability of a fire spreading across a cell boundary, from a burning cell into an unburnt neighbour.

In addition to these core fire modules, the model also makes use of several support modules involved in parameter estimation and model calibration. Fire regime parameters are estimated from historic fire data using the Canadian National Fire Database (Canadian Forest Service, n.d.) in order to reproduce realistic fire patterns. Parameterization and calibration of the model can be done separately for each geographic region used in the model (e.g., by ecoregion), allowing finer scale parameterization across large geographic areas. Individual spread parameters are determined in each geographic region following a four-part process. First, the region is buffered by a set distance and a flammability map is generated for the buffered region using the 2010 Land Cover Map of Canada (Latifovic et al., 2017). Next, fires are repeatedly simulated on the landscape across a range of spread probabilities. These fires cannot start in the buffered area, but may spread to and from the buffer. This reduces the influence of edge effects in determining mean fire size. Then the spread probabilities and resulting fire sizes are fit with a shape-constrained additive model (SCAM) (Pya and Wood, 2015). The SCAM is monotonic to ensure fire size increases for any incremental increase in spread probability. Lastly, an optimization algorithm is used to predict the spread probability that will generate the estimated mean fire size for the region.

This version of the model considers lightning-caused fires only. Fires are started by applying a random number sequence to each cell independently. Fires are extinguished when no further spread events (jumps) occur. After a fire, forested cells are reset to age 1. Thus, fires are presumed to be stand initiating. These processes result in irregular patches of variously sized burns. The locations of fires are not constant across simulations because of stochastic patterns of fire ignition and growth. The model tracks the sizes and compositions of each burn. Fire sizes follow a probability distribution largely determined by the spread parameter(s). The current implementation does not account for differences in vegetation type with respect to fire spread (e.g., flammability), nor is it climate sensitive. The more experimental `fireSense` model (Marchal, Cumming, and McIntire 2017), recently implemented in SpaDES, could be used in future updates to explicitly account for climate and vegetation impacts on fires (in prep).

Technical overview

The calibration modules of scfm are flexible and capable of accommodating various levels of a priori user knowledge regarding fire parameters. They can also account for calibration areas that are distinct from

simulation areas (provided the simulation area is a wholly contained by the calibration area).The minimum needed objects are a study area polygon ('studyArea') and a template raster that will determine the spatial resolution at which fire is simulated ('rasterToMatch'). However, if the simulation borders are determined by some factor other than fire ecology (e.g. administrative boundaries), each fire regime area may not possess sufficient historical fire data to produce accurate parameterization*. Two alternative approaches are outlined below. Users are recommended to identify the scenario that best fits their use case, and supply the corresponding objects, rather than supplying default objects arbitrarily.

Scenario 1: no prior knowledge of fire regimes In this scenario, some or all of the fire regime polygons ('fireRegimePolys') may be too geographically limited to include historical fire data sufficient for calibration. The suggested workflow is to buffer the study area by an appropriate amount to create a larger area for parameterization ('studyAreaLarge'). In this case, scfmLandcoverInit will calculate fire regime attributes for the larger area* ('landscapeAttrLarge'), scfmRegime will estimate fire regime parameters using the larger region, but scfmDriver will calibrate spread probability for the smaller region only. This ensures the simulated fires achieve the correct mean fire size. Additional objects needed: studyAreaLarge, rasterToMatchLarge. If flammableMap or fireRegimePolys is supplied, the large counterpart should also be supplied ('flammableMapLarge', 'fireRegimePolysLarge', respectively).

Scenario 2: a priori knowledge of fire regimes In this scenario, a user may have a priori knowledge of the maximum fire size and/or the fire return interval for one or more fire regime areas. These parameters can be passed to override the estimate provided by scfmRegime Additional objects needed: fireRegimePolys Additional params needed: scfmRegime's 'targetBurnRate' (1/Fire Return Interval) for a known fire return interval, and scfmRegime's 'targetMaxFireSize' for a known maximum fire size. Each is a named vector, with names corresponding to the IDs in fireRegimePolys$PolyID.

Technical notes: *the parameter most likely to be underestimated is the maximum fire size (maxFireSize). This is not typically very consequential, as simulated fires are unlikely to reach the maximum size unless it is drastically underestimated. The suggested solution is to pass pre-defined maximums to scfmRegime ('targetMaxFireSize'). The parameterized maximum fire size, in hectares, can be verified in `scfmRegimePars$<PolyID>emfs_ha`. **'Sliver polygons', ie polygons that are smaller than the sliver threshold parameter (default 100 km2), are joined with their nearest non-sliver neighbour. This prevents any areas that have too little fire data from being de-facto nonflammable. If a larger parameterization area is supplied ('fireRegimePolysLarge'), slivers are identified and merged only in the larger area. The simulation fire regimes are derived from these, after the slivers are merged. As a result, supplying fireRegimePolys is unnecessary when fireRegimePolysLarge is also supplied, as the former is overwritten by the latter.

References

- Armstrong, G W, and S G Cumming. 2003. "Estimating the Cost of Land Base Changes Due to Wildfire Using Shadow Prices." Forest Science 49 (5): 719–30. https://doi.org/10.1093/forestscience/49.5.719.
- Cumming, S G, D Demarchi, and C Walters. 1998. "A Grid-Based Spatial Model of Forest Dynamics Applied to the Boreal Mixedwood Region." Working Paper 1998–8. Sustainable Forest Management Network. https://doi.org/10.7939/R35N33.
- Latifovic, R, Pouliot, D, Olthof, I. "Circa 2010 Land Cover of Canada: Local Optimization Methodology and Product Development." Remote Sens. 2017, 9, 1098. https://doi.org/10.3390/rs9111098.
- Marchal, J, S G Cumming, and E J B McIntire. 2017. "Land Cover, More than Monthly Fire Weather, Drives Fire-Size Distribution in Southern Québec Forests: Implications for Fire Risk Management." PLoS ONE 12 (6): 1–17. https://doi.org/10.1371/journal.pone.0179294.
- Pya, N, and S N Wood. 2015. "Shape constrained additive models." Statistics and Computing, Vol. 25, Iss. 3, pp 543-559.

Usage example

```
library(Require)

Require(c("data.table", "ggplot2",
          "PredictiveEcology/LandR@development",
          "magrittr", "raster", "SpaDES.core", "sf"))
```

```
## Using GITHUB_PAT to access files on GitHub
## Loading required package: data.table
## Loading required package: ggplot2
## Loading required package: LandR
## Loading required package: magrittr
## Loading required package: raster
## Loading required package: sp
## Loading required package: SpaDES.core
## Loading required package: quickPlot
## Loading required package: reproducible
##
## Attaching package: 'reproducible'
## The following object is masked from 'package:Require':
##
##     paddedFloatToChar
##
## Attaching package: 'SpaDES.core'
## The following object is masked from 'package:Require':
##
##     paddedFloatToChar
## The following objects are masked from 'package:stats':
##
##     end, start
## The following object is masked from 'package:utils':
##
##     citation
## Loading required package: sf
## Linking to GEOS 3.9.1, GDAL 3.3.2, PROJ 7.2.1; sf_use_s2() is TRUE
##              data.table (>= 1.11.0)                  ggplot2 (>= 3.4.0)
##                                TRUE                                TRUE
## PredictiveEcology/LandR@development                   magrittr (>= 1.5)
##                                TRUE                                TRUE
##                  raster (>= 3.5-21)                        SpaDES.core
##                                TRUE                                TRUE
##                                  sf
##                                TRUE
```

```r
# Parameters
timeunit <- "year"
times <- list(start = 1, end = 250)
defaultPlotInterval <- 50
defaultInitialSaveTime <- NA #don't be saving nuffink

globCache <- c('.inputObjects')  #don't cache the init event - it is too prone to false positives

parameters <- list(
  scfmLandcoverInit = list(
    useCache = globCache,
    sliverThreshold = 1e8), #polygons <100 km2 are merged with closest non-sliver neighbour
  scfmIgnition = list(
    .useCache = globCache
  ),
  scfmEscape = list(
    .useCache = globCache
  ),
  scfmSpread = list(
    .useCache = globCache,
    .plotInitialTime = times$start,
    .plotInterval = defaultPlotInterval),
  scfmRegime = list(
    .useCache = ".inputObjects",
    fireCause=c("L")
  ),
  scfmDriver = list(
    .useCache = '.inputObjects',
    targetN = 4000 #increase targetN for more robust estimates, longer run-time
  )# default targetN = 4000, for reference.
)


modules <- list(
  "scfmLandcoverInit", "scfmRegime", "scfmDriver",
  "scfmIgnition", "scfmEscape",
  "scfmSpread", "scfmSummary"
)
## NOTE: replace modules with "group_scfm" to include ageModule;
##       ageModule isn't necessary to run and download of `ageMap` takes time

#Paths
paths <- list(
  cachePath = file.path("cache"),
  modulePath = file.path("modules"),
  inputPath = file.path("inputs"),
  outputPath = file.path("outputs")
)
#if you supply studyArea you should supply rtm to make sure the crs are identical.

center <- SpatialPoints(coords = data.frame(x = c(-1209980),
            y = c(7586895)), proj4string = CRS(paste("+proj=lcc +lat_1=49 +lat_2=77 +lat_0=0 +lon_0=-95
            "+datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0")))
```

```
studyArea <- LandR::randomStudyArea(size = 10000 * 100 * 30000, center = center, seed = 1001)

studyAreaLarge <- buffer(studyArea, 50000)

rasterToMatchLarge <- raster(extent(studyAreaLarge), res = c(250, 250))
crs(rasterToMatchLarge) <- crs(studyAreaLarge)
rasterToMatchLarge[] <- 1
rasterToMatchLarge <- mask(rasterToMatchLarge, studyAreaLarge)
rasterToMatch <- postProcess(rasterToMatchLarge, studyArea = studyArea)

studyAreaLarge$name <- "SAL" #make SPDF

studyArea <- st_as_sf(studyArea)
studyAreaLarge <- st_as_sf(studyAreaLarge)
# rasterToMatchLarge <- terra::rast(rasterToMatchLarge)
# rastertoMatch <- terra::rast(rasterToMatch)
#if run with no studyArea, the default is a small area in southwest Alberta with very few fires
objects <- list(
  studyArea = studyArea,
  studyAreaLarge = studyAreaLarge,
  rasterToMatch = rasterToMatch,
  rasterToMatchLarge = rasterToMatchLarge
)

#Run module
#The calibration process may take hours (it's cached)
options("reproducible.cachePath" = paths$cachePath)
options("reproducible.useMemoise" = FALSE)

outSim <- simInitAndSpades(times = times,
                           params = parameters,
                           modules = modules,
                           objects = objects,
                           paths = paths)
```

```
## TODO: add these plots as outputs in new scfmSummary module

dt <- scfmutils::compare Predictions_summaryDT(outSim)

## Some useful plots
gg_mfs <- scfmutils::compare Predictions_meanFireSize(dt)

# removed MAAB as diagnostic plot because it was derived from fire points incorrectly when SAL is suppl
# MAAB can still be calculated manually if a user desires

gg_fri <- scfmutils::compare Predictions_fireReturnInterval(dt)

gg_ign <- scfmutils::compare Predictions_annualIgnitions(dt)

gg_frp <- scfmutils::plot_fireRegimePolys(outSim$fireRegimePolys)

clearPlot()
gridExtra::grid.arrange(fps,  gg_mfs,  gg_fri,  gg_ign, nrow = 2, ncol = 2)
```