hefelys\_cde.R

uqrdissa

Sun Sep 10 20:51:03 2017

#Appendix S1. Annotated R code to implement marked inhomogeneous Poisson point process   
#species distribution model and two-phase nonparametric bootstrap algorithm.  
  
###############################################################################  
# Ensure you are using R Version 3.0.1 or newer  
# Required Packages: install packages 'VGAM', 'mosaic' and 'spatstat'  
###############################################################################  
  
library(VGAM)

## Loading required package: stats4

## Loading required package: splines

library(mosaic)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##   
## Attaching package: 'mosaic'

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following objects are masked from 'package:VGAM':  
##   
## chisq, logit

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median,  
## prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

library(spatstat)

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':  
##   
## collapse

## Loading required package: rpart

##   
## spatstat 1.51-0 (nickname: 'Poetic Licence')   
## For an introduction to spatstat, type 'beginner'

##   
## Note: spatstat version 1.51-0 is out of date by more than 4 months; we recommend upgrading to the latest version.

##   
## Attaching package: 'spatstat'

## The following objects are masked from 'package:mosaic':  
##   
## relrisk, rescale

## The following object is masked from 'package:lattice':  
##   
## panel.histogram

## The following object is masked from 'package:VGAM':  
##   
## dirichlet

###############################################################################  
# Generate single simulated data example   
#  
# Note: the simulated data example is the same as the large sample size used   
# in our simulation study. See manuscript for details.  
###############################################################################  
  
set.seed(1234)  
x=rnorm(10^6,0,1)  
eta.pres=8.5+1\*x  
mat=matrix(exp(eta.pres), nrow=1000, ncol=1000)  
lam=im(mat, xrange=c(0,1), yrange=c(0,1))  
points=rpoispp(lam)  
x.ipp=(log(lam[points])-8.5)/1  
x.int=rnorm(1000,0,1)   
y.IWLR=c(rep(1,length(x.ipp)),rep(0,length(x.int)))  
IPP.data=data.frame(y=y.IWLR,x=c(x.ipp,x.int))  
eta.group=1+0.5\*x.ipp  
y.ZTGLM=rpospois(length(eta.group),exp(eta.group))  
ZTGLM.data=data.frame(group.size=y.ZTGLM,x=x.ipp)  
eta.det=-2+-1\*x.ipp+0.5\*scale(y.ZTGLM)  
detected=rbinom(length(eta.det),1,ilogit(eta.det))  
keep=c(1:round(0.20\*length(x.ipp)))  
Detection.data=data.frame(y=detected[keep],x=x.ipp[keep],group.size=y.ZTGLM[keep])   
IPP.data=IPP.data[c(which(detected==1),which(y.IWLR==0)),]  
ZTGLM.data=ZTGLM.data[which(detected==1),]  
  
###############################################################################  
# Examine the example data   
#  
# 'IPP.data': Column "y" is the response variable used to estimate the inhomogeneous Poisson point process model  
# parameters using infinitely weighted logistic regression (Fithian & Hastie 2013). A value of 1 corresponds to a   
# presence-only location and a value of 0 corresponds to a Monte Carlo integration point. Column "x" is the habitat covariate.  
#  
# 'ZTGLM.data':Column "group.size" is the size of the group at each presence-only location.   
# Column "x" is the habitat covariate.  
#   
# 'Detection.data': Column "y" is the response variable. A value of 1 corresponds to detection and a value of 0 corresponds  
# to non-detection. Column "x" is the detection covariate and "group.size" is the group size of each detection/non-detection.  
###############################################################################  
  
names(IPP.data)

## [1] "y" "x"

names(ZTGLM.data)

## [1] "group.size" "x"

names(Detection.data)

## [1] "y" "x" "group.size"

###############################################################################  
# Estimate the coefficient for the habitat covariate "x" for the inhomogeneous Poisson point process model  
# ignoring non-detection sampling bias using infinitely weighted logistic regression (Fithian & Hastie 2013)  
#  
# Note: Because of non-detection sampling bias the coefficient estimate (0.29911) is not close to the true value of 1.  
###############################################################################  
  
IPP.ignored=glm(y~x,family="binomial",weights=10000^(1-y),data=IPP.data)  
summary(IPP.ignored)

##   
## Call:  
## glm(formula = y ~ x, family = "binomial", data = IPP.data, weights = 10000^(1 -   
## y))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8387 -1.0297 -0.8427 4.3645 4.6849   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.97136 0.04762 -209.381 <2e-16 \*\*\*  
## x 0.39035 0.04429 8.813 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 10864 on 1497 degrees of freedom  
## Residual deviance: 10786 on 1496 degrees of freedom  
## AIC: 10790  
##   
## Number of Fisher Scoring iterations: 6

###############################################################################  
# Estimate the coefficient for the habitat covariate "x" for the zero truncated Poisson generalized linear model   
# ignoring non-detection sampling bias.  
#  
# Note: Because of non-detection sampling bias the estimated intercept (1.15756) is biased. The true value is 1.  
# Non-detection sampling bias does not affect the estimated covariate "x" (0.51988) by a large amount. The true value is 0.5.  
###############################################################################  
  
ZTGLM.ignored=vglm(group.size~x,family="pospoisson",data=ZTGLM.data)  
summary(ZTGLM.ignored)

##   
## Call:  
## vglm(formula = group.size ~ x, family = "pospoisson", data = ZTGLM.data)  
##   
##   
## Pearson residuals:  
## Min 1Q Median 3Q Max  
## loge(lambda) -2.164 -0.794 -0.07959 0.6427 2.845  
##   
## Coefficients:   
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.10486 0.03131 35.28 <2e-16 \*\*\*  
## x 0.53582 0.02033 26.35 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Number of linear predictors: 1   
##   
## Name of linear predictor: loge(lambda)   
##   
## Log-likelihood: -960.3629 on 496 degrees of freedom  
##   
## Number of iterations: 5

################################################################################  
# Correcting for non-detection sampling bias.   
#  
# Note: Steps numbered 1-7 are in reference to the two-phase nonparametric bootstrap algorithm in the manuscript.  
# Bootstrap sampling (i.e., sampling with replacement) and steps 1, 3, and 7 are not needed to get  
# corrected coefficient estimates and are only needed to calculate standard errors and confidence intervals.   
# We will skip bootstrap sampling and steps 1, 3 and 7 until later on.  
#  
# Note: True values of the intercept, "x" and "group.size" used to generate detection data are -2, -1 and 0.5 respectively.   
#   
# Note: Because we corrected for non-detection sampling bias the estimate of the   
# coefficient "x" (0.94315) in step 5 is very close to the true value of 1.   
# The standard error (0.01185), however, should not be trusted as it is estimated without   
# taking into account the variablity in the estimated probability of detection("p.det").  
# Similarly the estimated intercept (0.98370) and coefficient "x" (0.53195) in step 6 is   
# close to the true value of 1 and 0.5 respectively. The standard errors for both estimates should  
# not be trusted.  
###############################################################################  
  
##Step 2 - Fit an appropriate model to the detection data set  
Detection.model=glm(y~x+scale(group.size),family="binomial",data=Detection.data)  
summary(Detection.model)

##   
## Call:  
## glm(formula = y ~ x + scale(group.size), family = "binomial",   
## data = Detection.data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.1366 -0.3757 -0.2758 -0.2114 2.8435   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.9447 0.1369 -14.202 < 2e-16 \*\*\*  
## x -1.1118 0.1489 -7.466 8.24e-14 \*\*\*  
## scale(group.size) 0.5604 0.1680 3.336 0.000851 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 727.16 on 1606 degrees of freedom  
## Residual deviance: 659.92 on 1604 degrees of freedom  
## AIC: 665.92  
##   
## Number of Fisher Scoring iterations: 6

##Step 4 - Estimate the probability of detection for each presence-only location.  
p.det=ilogit(predict(Detection.model,new=ZTGLM.data))  
IPP.data$p.det=c(p.det,rep(1,length(x.int)))  
ZTGLM.data$p.det=p.det  
  
  
##Step 5 - Fit an inhomogeneous Poisson point process that weights the log-likelihood by 1/p.det  
IPP.corrected=glm(y~x,family="binomial",weights=(1/p.det)\*10000^(1-y),data=IPP.data)

## Warning in eval(family$initialize): non-integer #successes in a binomial  
## glm!

summary(IPP.corrected)

##   
## Call:  
## glm(formula = y ~ x, family = "binomial", data = IPP.data, weights = (1/p.det) \*   
## 10000^(1 - y))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -19.907 -3.739 -2.094 10.830 34.092   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.63116 0.01584 -481.7 <2e-16 \*\*\*  
## x 1.12902 0.01060 106.5 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 143313 on 1497 degrees of freedom  
## Residual deviance: 131769 on 1496 degrees of freedom  
## AIC: 131864  
##   
## Number of Fisher Scoring iterations: 7

##Step 6 - Fit an zero-truncated Poisson generalized linear model that weights the log-likelihood by 1/p.det  
ZTGLM.corrected=vglm(group.size~x,weights=1/p.det,family="pospoisson",data=ZTGLM.data)  
summary(ZTGLM.corrected)

##   
## Call:  
## vglm(formula = group.size ~ x, family = "pospoisson", data = ZTGLM.data,   
## weights = 1/p.det)  
##   
##   
## Pearson residuals:  
## Min 1Q Median 3Q Max  
## loge(lambda) -18.38 -2.054 0.582 3.377 10.66  
##   
## Coefficients:   
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.964815 0.009581 100.7 <2e-16 \*\*\*  
## x 0.529984 0.005047 105.0 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Number of linear predictors: 1   
##   
## Name of linear predictor: loge(lambda)   
##   
## Log-likelihood: -18993.02 on 496 degrees of freedom  
##   
## Number of iterations: 5

###############################################################################  
# Implementing the two-phase nonparametric bootstrap algorithm (i.e., steps 1-7)  
#  
# 'set.seed' establishes a random starting seed so simulation results are the same each time the bootstrap code is run.  
# If the set.seed code is ignored, the mean, standard deviation and confidence intervals will vary slightly each time   
# the code is run.  
#  
# The function "tpnbs" is the two-phase nonparametric bootstrap and implements steps 1-7.   
#  
# 'n.bootstraps' is the number of bootstrap samples to be included in the analysis. We used 1,000 in our study.  
#  
# The 'do()' function in package mosaic implements the bootstrap.  
###############################################################################  
set.seed(1234)  
  
tpnbs=function() {  
 bss=resample(1:dim(ZTGLM.data)[1])  
 IPP.data.bss=rbind(IPP.data[bss,1:2],IPP.data[which(IPP.data$y==0),1:2])  
 ZTGLM.data.bss=ZTGLM.data[bss,]  
 Detection.data.bss=resample(Detection.data)  
 Detection.model=glm(y~x+scale(group.size),family="binomial",data=Detection.data.bss)  
 p.det.bss=ilogit(predict(Detection.model,new=ZTGLM.data.bss))  
 IPP.data.bss$p.det=c(p.det.bss,rep(1,length(x.int)))  
 ZTGLM.data$p.det=p.det.bss  
 options(warn=-1)  
 IPP.model=glm(y~x,family="binomial",weights=(1/p.det)\*10000^(1-y),data=IPP.data.bss)  
 ZTGLM.corrected=vglm(group.size~x,weights=1/p.det,family="pospoisson",data=ZTGLM.data.bss)  
 c(coef(IPP.model)[2],coef(ZTGLM.corrected))  
}  
n.bootstraps=1000  
bootstrap.sample = mosaic::do(n.bootstraps) \*tpnbs()  
str(bootstrap.sample)

## Classes 'do.data.frame' and 'data.frame': 1000 obs. of 3 variables:  
## $ x : num 1.25 1.22 1.25 1.13 1.27 ...  
## $ Intercept: num 0.926 0.941 1.015 0.94 0.942 ...  
## $ x.1 : num 0.53 0.554 0.528 0.528 0.552 ...  
## - attr(\*, "lazy")=Class 'formula' language ~tpnbs()  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## - attr(\*, "culler")=function (object, ...)

bootstrap.sample=data.matrix(bootstrap.sample)  
###############################################################################  
# Calculating mean, standard deviation and 95%, equal-tailed confidence intervals  
# from the empirical distribution. See "Introduction to the Bootstrap" (Efron & Tibshirani 1994)   
# for more details.  
#  
# Estimates of the coefficient for the covariate "x" (0.9520575) of the inhomogeneous Poisson point   
# process model and the intercept(0.9832485) and covariate "x" (0.531719) of the zero-truncated   
# Poisson generalized linear model are close to the true values (1, 1, 0.5).   
#  
# Standard errors when detection bias is accounted for(0.1141398; 0.03696048; 0.0224879), and thus 95%   
# confidence intervals, and standard errors (0.01185; 0.0099313; 0.0060487) are larger when the variability   
# in the probability of detection is not accounted for. Note the estimated standard errors(0.01185; 0.0099313; 0.0060487)  
# are obtained from steps 5 & 6 above.   
###############################################################################  
  
colMeans(bootstrap.sample)[1]

## x   
## 1.137915

sd(bootstrap.sample)[1]

## [1] 0.2662821

data(c(.025, .975),bootstrap.sample[,1])  
  
colMeans(bootstrap.sample)[2]

## Intercept   
## 0.9658571

sd(bootstrap.sample)[2]

## [1] NA

data(c(.025, .975),bootstrap.sample[,2])  
  
colMeans(bootstrap.sample)[3]

## x.1   
## 0.5298045

sd(bootstrap.sample)[3]

## [1] NA

data(c(.025, .975),bootstrap.sample[,3])