Package 'MBHdesign'

October 30, 2017

Title Spatial Designs for Ecological and Environmental Surveys

Version 1.0.79

Author Scott D. Foster

Description Provides spatially balanced designs from a set of (contiguous) potential sampling locations in a study region. Accommodates , without detrimental effects on spatial balance, sites that the researcher wishes to include in the survey for reasons other than the current randomisation (legacy sites).

Maintainer Scott Foster <scott.foster@csiro.au>

License GPL (>= 2)

Imports mgcv, geometry, randtoolbox, mvtnorm, stats, class

Suggests fields, graphics, knitr, spsurvey

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-30 04:25:20 UTC

R topics documented:

`alterInclProbs` *Alters inclusion probabilities to accommodate legacy sites*

Description

Alters inclusion probabilities to accommodate legacy sites. Inclusion probabilities are deflated around legacy sites, and the resulting set of new sites (and legacy sites) are spatially balanced.

Usage

```
alterInclProbs( legacy.sites, potential.sites=NULL, n=NULL, inclusion.probs=NULL,
                                                    mc.cores=1, sigma=NULL)
```

Arguments

legacy.sites	a matix (MxD) matrix of locations of the M legacy sites as points in D dimensions. For most applications D=2. Each row gives the location (in space) of one of the legacy sites.
potential.sites	a matrix (NxD) of locations of the N potential sampling sites as points in D dimension. These are the locations from which n are taken as the sample. If NULL (default) a maximum of N=10000 samples are placed on a regular grid within a convex hull defined by the legacy locations. This default may or may not make sense for you (but something has to be the default).
n	an integer specificying the number of new sites to sample. One of n and inclusion.probs needs to be specified, but if both are then n is discarded.
inclusion.probs	a vector specifying the inclusion probability for each of the N sampling sites. This is the probability that each site will be included in the final sample. The sum of inclusion.probs must be the number of new sites required. Locations in inclusion.probs must be ordered the same as the potential.sites argument. If NULL (default) equal inclusion probabilities are specified and the number of new sites is taken to be n.
mc.cores	the number of processes to for some of the calculations on (in particular the calculation of distances to legacy sites). See parLapply(qv) in package parallel.
sigma	a parameter defining the spatial influence of the legacy sites. Must be positive. It is sigma in the squared exponential decay function, exp(-distanceFromLegacySite^2 / sigma^2). If NULL (default), then it is chosen so that 95 percent of the legacy sites influence is located within the average patch size around a point. This seems like a useful default. See Foster et al. (in prep) for details.

Details

The inclusion probabilities are adjusted using a variant of the spatially clustered Poisson sampling method in Grafstrom (2012), which is itself a spatially explicit version of Bondesson and Thorburn (2008). The adjustments here are given in Foster et al. (in prep). Basically, the adjustment is similar to that proposed in Section 3.2 of Grafstrom (2012), that is the inclusion probabilities are updated using the squared loss distance metric (as above). However, the weighting function here is given by the distance times the inclusion probability of the new site.

Value

The alterInclProbs function returns a numeric vector containing inclusion probabilities adjusted f

Author(s)

Scott D. Foster

References

Bondesson, L. and Thoburn, D. (2008) A List Sequential Sampling Method Suitable for Real-Time Sampling. Scandinavian Journal of Statistics *35*:466–483.

Foster, S.D., Hosack, G.R., Lawrence, E., Przeslawski, R., Hedge,P., Caley, M.J., Barrett, N.S., Williams, A., Li, J., Ly NA–NA DOI: 10.1111/2041-210X.12782

Grafstrom, A. (2012) Spatially correlated Poisson sampling. Journal of Statistical Plannind and Inference *142*:139–147.

See Also

quasiSamp, modEsti

Examples

```
#big plane today
set.seed(747)
#the number of potential sampling locations
N <- 100^2
#number of samples
n <- 27
#number of legacy sites
nLegacy <- 3
#the grid
X <- as.matrix( expand.grid( 1:sqrt( N), 1:sqrt(N)) / sqrt(N) - 1/(2*sqrt(N)))
#the inclusion probabiltiies with gradient according to non-linear function of X[,1]
p <- 1-exp(-X[,1])
#standardise to get n samples
p <- n * p / sum( p)
#randomly choose legacy sites
legacySites <- sample( 1:N, nLegacy, prob=p)
#alter inclusion probabilities
p2 <- alterInclProbs( legacy.sites=X[legacySites,], potential.sites=X, inclusion.probs=p)
if( requireNamespace( "graphics", quietly = TRUE)) {
  #plotting up old and new inclusion probabilities
  par( mfrow=c(1,2))
  graphics::image( x=unique( X[,1]), y=unique( X[,2]),
    z=matrix( p, nrow=sqrt(nrow(X)), ncol=sqrt(nrow( X))),
    main="Undadjusted Inclusion Probabilities", ylab="y", xlab="x")
  graphics::image( x=unique( X[,1]), y=unique( X[,2]),
    z=matrix( p2, nrow=sqrt(nrow(X)), ncol=sqrt(nrow( X))),
    main="Adjusted Inclusion Probabilities", ylab="y", xlab="x")
  points( X[legacySites,], pch=20, col=1)
}
#tidy
rm( N, n, nLegacy, X, p, legacySites, p2)
```

modEsti *Get a model-based estimate of mean of a sampled area*

Description

For a given survey design in any number of dimensions, calculate the mean prediction (plus SE plus 95% CI) for the area.

Usage

```
modEsti( y, locations, includeLegacyLocation=TRUE, legacyIDs=NULL, predPts=NULL,
                        family=gaussian(), offset=rep(0,length(y)), control=list())
```

Arguments

y	a numeric vector of all observations (at new sites and legacy sites) collected in the survey
locations	a matrix (or something that can be coerced) containing the set of locations where observations were collected. Note that nrow(locations) == length(y).
includeLegacyLocation	a boolean indicating whether an extra term should be included into the model that corresponds to distance from legacy sites. See Foster et al (in review) for details. If TRUE (default) the extra term is included. If FALSE (so that model is purely spatial), then the extra term is discarded.
legacyIDs	the indexes, for the rows of y and locations, that correspond to legacy sites. For example, if the first, third and sixth rows were legacy sites in y and locations, then legacyIDs would be c(1,3,6).
predPts	A data.frame (or something that can be coerced to a data.frame) containing set of points to do the predictions at. Typically predPts is a dense grid (or similar) of points over the spatial area of interest. Note that the number of columns defines the number of dimensions. Do not include surplus variables in extra columns. If NULL (default), then a dense grid within the convex hull bounding "locations" will be used.
family	A family object giving the distribution of the data. Default is `gaussian()` but other sensible choices in ecology include `nb(link="log")` and `binomial()`. For more details see `?family.mgcv` from the mgcv package.
offset	A numeric vector of length equal to "length(y)". This gives any offset to the linear predictor of the model.
control	A list of control parameters (see details)

Details

This function works by fitting a generalised additive model (see `gam()`), predicting at the points predPts, and then averaging. Well, that is the general idea. The actual implementation uses a Monte Carlo routine to account for parameter uncertainty. This is done mirroring the helpfile of

predict.gam. Basically, lots of sets of parameters are drawn from the parameters (asymptotic) distribution and then predictions are made for each draw. The overall estimate is then the mean (over parameter draws) of the mean (over prediction locations) of the prediction. Standard errors and confidence intervals are likewise calculated.

The control list contains elements with names:

k the number of knot points used in each dimension of locations

N the number of prediction points (in each dimension) for the grid, argument not used if "predPts!=NULL"

B the number of bootstrap samples to take of the parameter estimates

mc.cores the number of computer cores to spread the calculation of distances over (only used if includeLegacyLocation==TRUE)

Value

A list of three elements: 1) a point prediction of mean, 2) standard error of mean (obtained by parametric bootstrap), and 3) 95% confidence interval of the mean.

Author(s)

Scott D. Foster

References

Foster, S.D., Hosack, G.R., Lawrence, E., Przeslawski, R., Hedge,P., Caley, M.J., Barrett, N.S., Williams, A., Li, J., Ly NA–NA DOI: 10.1111/2041-210X.12782

See Also

quasiSamp, alterInclProbs, and `total.est` from the **spsurvey** package. The total.est function provides design-based estimation of both mean and variance.

Examples

```
#set up design parameters
#taken from the example in alterInclProbs()
#big plane today
set.seed(747)
#the number of potential sampling locations
N <- 100^2
#number of samples
n <- 27
#number of legacy sites
nLegacy <- 3
#the grid
X <- as.matrix( expand.grid( 1:sqrt( N), 1:sqrt(N)) / sqrt(N) - 1/(2*sqrt(N)))
#the inclusion probabiltiies with gradient according to non-linear function of X[,1]
p <- 1-exp(-X[,1])
#standardise to get n samples
p <- n * p / sum( p)
#randomly choose legacy sites
```

```
legacySites <- sample( 1:N, nLegacy, prob=p)
#alter inclusion probabilities for legacy sites
p2 <- alterInclProbs( legacy.sites=X[legacySites,], potential.sites=X, inclusion.probs=p)
#get the sample
samp <- quasiSamp( n=n, dimension=2, potential.sites=X, inclusion.probs=p2)
samp <- rbind( cbind( X[legacySites,], inclusion.probabilities=NA, ID=NA), samp)
#generate some fake data
samp$outcomes <- rnorm( nrow( samp))
#get the estimate
esti <- modEsti( y=samp$outcomes, locations=samp[,1:2], includeLegacyLocation=TRUE,
  legacyIDs=1:3, predPts=NULL, family=gaussian(), control=list(mc.cores=2, B=100))
#in real applications the number of bootstrap samples (B) should be larger
print( esti)
#tidy
rm( esti, legacySites, n, N, nLegacy, p, p2, samp, X)
```

---

quasiSamp *Generates a spatial design using Quasi-random numbers*

Description

Generates a spatially balanced design for given inclusion probabilities over a grid of potential sampling locations

Usage

```
quasiSamp( n, dimension=2, study.area=NULL, potential.sites=NULL, inclusion.probs=NULL,
                                        randStartType=2, nSampsToConsider=5000)
```

Arguments

n	the number of samples to take
dimension	the number of dimensions that the samples are located in. Equal to 2 for areal sampling. Care should be taken with large dimensions as:1) the number of potential sampling sites needed for effective coverage starts to explode (curse of dimensionality); and 2) the well-spaced behaviour of the Halton sequence starts to deteriorate (but this requires very very many dimensions to be problematic – included as a warning here for largely academic reasons).
study.area	a numeric matrix with dimension columns. This defines the sampling area from where the sites are selected (each row defines a vertex of the sampling area). If NULL (default), the study.area is defined to be the smallest (hyper-)rectangle that bounds the potential.sites. If potential.sites is also NULL (default), then the study area is taken to be the unit (hyper-)square. This argument is closely related to potential.sites.
potential.sites	a matrix (of size Nxdimension) of the spatial coordinates of the N sampling locations, of which n«N are taken as the sample. If NULL (default) N=10000

samples are placed on a regular grid. If study.area is defined, then this grid is over the smallest bounding (hyper-)rectangle for the study.area. If study.area is NULL, the grid is over the unit (hyper-)square.

`inclusion.probs`
: a vector specifying the inclusion probability for each of the N potential sampling sites. This is the probability that each site will be included in the final sample. Locations are ordered the same as the potential.sites argument. If NULL (default) equal inclusion probabilities are specified.

`randStartType`
: the type of random start Halton sequence to use. The choices are 2 (default) as described in Robertson et al (2013), and 1 which is a mis-interpretation of that method (constrained so that the size of the skip in each dimension is equal). Note that randStartType=1 is used in Foster et al (2017).

`nSampsToConsider`
: the total number of samples to consider in the BAS step (rejection sampling). The default is 5000, which means that 5000 halton numbers are drawn and then thinned according to the inclusion probabilities. You may want to increase this number if your inclusion probabilities are extremely unbalanced or if the number of samples required is close to 5000. Reduce if you want the code to run quicker and are confident that a sample will be found using less.

Details

This function is an implementation of the balanced adaptive sampling (BAS) designs presented in Robertson et al. (2013), which forms the basis for the methods in Foster et al (in review). The BAS approach uses Halton sequences of quasi-random numbers, which are evenly spread over space, as the basis for generating spatially balanced designs. In this implementation, we requrie that the inclusion probabilities be given as points in space and the BAS design is the set of these points that lie closest to a continuous-space Halton sequence. Computational speed has been rudimentily optimised, but (of course) it could be done better – like coding outside of R, for example.

Value

`The quasiSamp function returns a matrix of (dimension+2) columns. The first columns (of number dim`

Author(s)

Scott D. Foster

References

Robertson, B. L., Brown, J. A., McDonald, T. and Jaksons, P. (2013) BAS Balanced Acceptance Sampling of Natural Resources. Biometrics *69*: 776–784 DOI: 10.1111/biom.12059

Foster, S.D., Hosack, G.R., Lawrence, E., Przeslawski, R., Hedge,P., Caley, M.J., Barrett, N.S., Williams, A., Li, J., Ly NA–NA DOI: 10.1111/2041-210X.12782

See Also

`alterInclProbs`, `modEsti`

Examples

```
#generate samples on a 100 x 100 grid
#jet plane
set.seed(707)
#the number of potential sampling locations
N <- 100^2
#number of samples
n <- 10
#the grid on unit square
X <- as.matrix( expand.grid( 1:sqrt( N), 1:sqrt(N)) / sqrt(N) - 1/(2*sqrt(N)))
#the inclusion probabiltiies with gradient according to non-linear function of X[,1]
p <- 1-exp(-X[,1])
#standardise to get n samples
p <- n * p / sum( p)
#get the sample
samp <- quasiSamp( n=n, dimension=2, potential.sites=X, inclusion.probs=p)
par( mfrow=c(1,3))
plot( samp[,1:2], main="n=10")
#now let's get sillier
n <- 250
#get the sample
samp <- quasiSamp( n=n, dimension=2, potential.sites=X, inclusion.probs=p)
plot( samp[,1:2], main="n=250")
#silly or sublime?
n <- 1000
#get the sample
samp <- quasiSamp( n=n, dimension=2, potential.sites=X, inclusion.probs=p, nSampsToConsider=5000)
plot( samp[,1:2], main="n=1000")
#I'm sure that you get the idea now.
#tidy
rm( N, n, X, p, samp)
```

Index