Package 'boss'

February 19, 2015

Type Package

Title Boosted One-Step Statistics: Fast and accurate approximations for GLM, GEE and Mixed models for use in GWAS

Version 2.1

Date 2012-03-09

Author Arend Voorman, Colleen Sitlani

Maintainer Arend Voorman <voorma@u.washington.edu>

Depends lme4 (>= 1.0), geepack, Matrix, ncdf

Description BOSS uses parameter estimates obtained without genotype to boost standard one-step approximations, and precomputes as much as possible without genotype (using boss.set) to minimize effort required.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-09-21 10:00:36

R topics documented:

`boss.fit` *fit a model with genotype.*

Description

Uses an object from 'boss.set' and calls an appropriate fitter function to add genotype to the model. For GEE, degrees of freedom estimates can be calculated in addition to standard output.

Usage

```
boss.fit(g, init, thresh = 1e-7, robust=TRUE,...)
```

Arguments

`g`	A genotype variable, assumed to be in the same order as the outcome variable used in 'boss.set()'. For the 'swap' method, this can be a matrix.
`init`	An object from 'boss.set()'
`thresh`	p-value threshold beyond which further iteratively re-weighted least squares iterations should be performed. Ignored for OLS and linear GEE, where estimates are fully converged, also ignored for "swap" method.
`robust`	logical. Whether or not robust standard errors should be used. Defaults to TRUE. Ignored for mixed models, where they do not exist, and "swap" method.
`...`	optional arguments to be passed to the fitter functions if further iterations are desired. The argument 'sattdf=TRUE' can be used with GEE models to calculate an approximate degrees of freedom for a t-reference distribution - see examples.

Value

`beta.main`	estimated coefficient for the main effect of genotype
`v.main`	the estimated variance of beta.main
`beta.inter`	when interaction term is supplied, the estimated coefficient for the interaction with genotype
`v.inter`	when interaction term is supplied, the estimated variance of beta.inter
`cov.inter`	when interaction term is supplied, the estimated covariance between beta.inter and beta.main
`df`	when 'sattdf=TRUE', the satterwaite approximation for the residual degrees of freedom

Note

For GEE models, the option sattdf=TRUE, gives an satterwaite approximation for the residual degrees of freedom. For linear models the option 'kurtosis.correction=TRUE' returns an estimated residual degrees of freedom.

Both of these options work only with robust standard errors. For linear GEE, the degrees of freedom is currently only supported for independence working correlations.

Author(s)

Arend Voorman, Colleen Sitlani.

References

Fast Computation for Genome Wide Association Studies using Boosted One-Step Statistics Arend Voorman; Ken Rice; Thomas Lumley Bioinformatics 2012; doi: 10.1093/bioinformatics/bts291

See Also

boss.set glm geese lmer

Examples

```
#generate data:
n <- 500
nsnps <- 1000
snpMat <- replicate(nsnps,rbinom(n,2,.01))
colnames(snpMat) <- 1e6+(1:nsnps)
rownames(snpMat) <- 1:n

#generate 3 observations per person with random intercept and covariates:
id <- as.numeric(gl(n,3))
y <- rnorm(3*n) + rnorm(n,sd=.3)[id]
data <- data.frame(y=y, x1=rnorm(3*n,mean=y), x2 =rnorm(3*n),
site=sample(6,n,replace=TRUE)[id], id=id )

# set up a GEE with independence working correlation structure, using "chol" method:
init1 <- boss.set(y~x1+x2+factor(site), id=data$id, type = "gee",
method = "chol", corstr= "independence", data=data)

# set up a linear mixed model (lmm) with random intercept using "swap" method:
init2 <- boss.set(y~x1+x2+factor(site) + (1|id), type = "lmm", method = "swap", data=data)

#set up an interaction with logistic regression, which must use the "chol" method:
y <- rbinom(n,1,0.5)
data.bin <- data.frame(y=y, x1=rnorm(n,mean=y), x2 =rnorm(n),
e = rnorm(n), site=sample(6,n,replace=TRUE), id=1:n)

init3 <- boss.set(y~x1+x2+factor(site)+e, E.name = "e", id=id,
family=binomial(), type="glm", data = data.bin)

##fit the model (at one locus):
g <- snpMat[,1]

###GEE
#one step approximation, using satterwaite degrees of freedom.
re <- boss.fit(g[id], init1, robust=TRUE, sattdf=TRUE)
re

#p-value with df approximation:
2*pt(abs(re$beta.main/sqrt(re$v.main)),df=re$df, lower=FALSE)
```

```
#p-value with normal approximation:
2*pnorm(abs(re$beta.main/sqrt(re$v.main)), lower=FALSE)


#full iteration
boss.fit(g[id], init1, robust=TRUE, thresh=1)


##LMM
#one step approximation:
boss.fit(g[id], init2)

#full iteration (uses lmer, since "swap" method does not permit further iterations currently)
mod <- lmer(update(init2$formula,.~.+g[id]),data=data)
list(beta.main = getME(mod,"beta")[9], v.main=vcov(mod)[9,9],
     chi2 = getME(mod,"beta")[9]^2/vcov(mod)[9,9])

##Logistic Reg.
boss.fit(g, init3)

#full iteration
boss.fit(g, init3, thresh = 1)

#for swap method, matrix arguments can be used to simultaneously fit many models:
results <- boss.fit(t(snpMat[,id]),init2)
```

`boss.ncdf` *Wrapper function for anlayzing data in net CDF files*

Description

Takes an object from boss.set() and a net CDF file, reads the data into memory in chunks and performs the analysis

Usage

```
boss.ncdf(nc, init, id.labels = NULL, g.labels = NULL, subset = NULL, gdim = 1,
chunk = 1000, verbose = TRUE, outfile = NULL, ...)
```

Arguments

`nc`	a net CDF file
`init`	An object from boss.set()
`id.labels`	Name in the net CDF file where sample id's are given, to be matched with the id's used in boss.set().
`g.labels`	Name in the net CDF file where genotype labels can be found. Used to label output
`subset`	optional, labels of a subset of markers on which to perform the analysis

gdim	Dimension in the net CDF file where genotype can be found, either 1 (rows) or 2 (columns)
chunk	Size of chunks to read from the net CDF file. For 'swap' method, a few thousand at a time is the fastest. For the 'chol' method, 100 or so gives good efficiency.
verbose	logical. Whether or not print progress indicators. a printed '!' indicates a failed regression, typically due to a constant genotype.
outfile	optional location of where to write the output, as a csv file
...	aditional arguments to be given to boss.fit()

Value

Returns matrix of results giving the MAF of genotype, genotype-related coefficients and variances, and Chi Squared statistics. If 'outfile' is specified, it prints the results to 'outfile.csv', and no object is returned.

Author(s)

Arend Voorman

References

Fast Computation for Genome Wide Association Studies using Boosted One-Step Statistics Arend Voorman; Ken Rice; Thomas Lumley Bioinformatics 2012; doi: 10.1093/bioinformatics/bts291

See Also

boss.fit boss.set ncdf

Examples

```
#generate fake data:
n <- 500
nsnps <- 1000
snpMat <- replicate(nsnps,rbinom(n,2,.2))
colnames(snpMat) <- 1e6+(1:nsnps)
rownames(snpMat) <- 1:n

snpdim <- dim.def.ncdf("position","bases", as.numeric(colnames(snpMat) ))
sampledim <- dim.def.ncdf("sample", "count", as.numeric(rownames(snpMat) ))

varsnp <- var.def.ncdf("snp", "rs", dim = snpdim, missval = -1, prec = "integer")
vargeno <- var.def.ncdf("genotype","base", dim = list(snpdim, sampledim),
missval = -1, prec = "double")
genofile <- create.ncdf("Example.nc",list(varsnp,vargeno))

for(i in 1:n){
put.var.ncdf(genofile,"genotype", snpMat[i,], start=c(1,i), count = c(-1,1))
}
```

```
nc <- open.ncdf("Example.nc")

#generate 3 observations per person, with random intercept:
id <- as.numeric(gl(3*n,3))
y <- rnorm(3*n) + rnorm(n,sd=.3)[id]
data <- data.frame(y=y, x1=rnorm(3*n,mean=y), x2 =rnorm(3*n),
site=sample(6,n,replace=TRUE)[id], id=id )

init <- boss.set(y~x1+x2+factor(site), id=data$id, data=data,
type = "gee", method = "chol", corstr= "ar1")

g.labels <- "position"
id.labels <- "sample"

results <- boss.ncdf(nc, init, id.labels, g.labels, gdim = 1,chunk=100, robust = TRUE, thresh=0)

close.ncdf(nc)
```

`boss.set` *Create an object to be passed to boss.fit*

Description

Takes as arguments everything in a GWAS model that does not explicitely involve genotype and pre-computes as much as possible

Usage

```
boss.set(formula, E.name = NULL, family = gaussian(), id = NULL,
corstr = "independence", type = "glm", method = "chol", data, ...)
```

Arguments

`formula`	A formula, of the kind used by the 'type' specified, that does not include genotype.
`E.name`	For interaction studies, the name (as a character) of the variable with which to perform the interaction
`family`	A family object, only gaussian() and binomial() supported
`id`	A vector of unique id's for subjects in the sample, to be matched with id's associated with genotype. For GEE models, also used to generate clusters.
`corstr`	For GEE models only, one of "independence", "ar1" and "exchangeable", specifying the working correlation structure
`type`	One of "glm", "gee" or "lmm"
`method`	One of "chol" or "swap", default is "chol". "swap" cannot be used with robust standard errors or interaction studies, but is substantially faster.
`data`	An optional data frame in which the variables in 'formula' are located.
`...`	Arguments to be passed to the 'glm()', 'lmer()' or 'geese()' used internally in boss.set. The argument 'sattdf=TRUE' can be used for

Value

An object to be used in 'boss.set()'

Author(s)

Arend Voorman, Colleen Sitlani

References

Voorman, A.; Rice, K. and Lumley, T. *Fast Computation for Genome Wide Association Studies using Boosted One-Step Statistics*. Bioinformatics 2012; doi: 10.1093/bioinformatics/bts291

Satterwaite F. *An Approximate Distribution of Estimates of Variance Components*. Biometrics Bulletin 1946; 2(6) 110-114.

Pan W, Wall M. *Small-sample adjustments in using the sandwich variance estimator in generalized estimating equations*. Statist Med 2002; 21:1429-1441.

See Also

boss.fit boss.set glm geese lmer

Examples

```
#generate data:
n <- 500
nsnps <- 1000
snpMat <- replicate(nsnps,rbinom(n,2,.01))
colnames(snpMat) <- 1e6+(1:nsnps)
rownames(snpMat) <- 1:n

#generate 3 observations per person with random intercept and covariates:
id <- as.numeric(gl(n,3))
y <- rnorm(3*n) + rnorm(n,sd=.3)[id]
data <- data.frame(y=y, x1=rnorm(3*n,mean=y), x2 =rnorm(3*n),
site=sample(6,n,replace=TRUE)[id], id=id )

# set up a GEE with independence working correlation structure, using "chol" method:
init1 <- boss.set(y~x1+x2+factor(site), id=data$id, type = "gee",
method = "chol", corstr= "independence", data=data)

# set up a linear mixed model (lmm) with random intercept using "swap" method:
init2 <- boss.set(y~x1+x2+factor(site) + (1|id), type = "lmm", method = "swap", data=data)

#set up an interaction with logistic regression, which must use the "chol" method:
y <- rbinom(n,1,0.5)
data.bin <- data.frame(y=y, x1=rnorm(n,mean=y), x2 =rnorm(n),
e = rnorm(n), site=sample(6,n,replace=TRUE), id=1:n)

init3 <- boss.set(y~x1+x2+factor(site)+e, E.name = "e", id=id,
family=binomial(), type="glm", data = data.bin)

##fit the model (at one locus):
```

```
g <- snpMat[,1]

###GEE
#one step approximation, using satterwaite degrees of freedom.
re <- boss.fit(g[id], init1, robust=TRUE, sattdf=TRUE)
re

#p-value with df approximation:
2*pt(abs(re$beta.main/sqrt(re$v.main)),df=re$df, lower=FALSE)
#p-value with normal approximation:
2*pnorm(abs(re$beta.main/sqrt(re$v.main)), lower=FALSE)


#full iteration
boss.fit(g[id], init1, robust=TRUE, thresh=1)


##LMM
#one step approximation:
boss.fit(g[id], init2)

#full iteration (uses lmer, since "swap" method does not permit further iterations currently)
mod <- lmer(update(init2$formula,.~.+g[id]),data=data)
list(beta.main = getME(mod,"beta")[9], v.main=vcov(mod)[9,9],
     chi2 = getME(mod,"beta")[9]^2/vcov(mod)[9,9])

##Logistic Reg.
boss.fit(g, init3)

#full iteration
boss.fit(g, init3, thresh = 1)

#for swap method, matrix arguments can be used to simultaneously fit many models:
results <- boss.fit(t(snpMat[,id]),init2)
```

---

updateR2 *Internal function to update choleski decomposition.*

Description

This is a modified version of the function 'updateR()' function from the 'lars' package. This version can handle rank >= 1 updates at a time.

Usage

```
updateR2(xnew, R = NULL, xold, eps = .Machine$double.eps)
```

Arguments

xnew	variable(s) to be added to choleski decomposition
R	choleski decomposition of crossprod(xold)
xold	old design matrix
eps	limit of what to consider a zero

Author(s)

Arend Voorman

References

Trevor Hastie and Brad Efron (2012). lars: Least Angle Regression, Lasso and Forward Stagewise. R package version 1.1. http://CRAN.R-project.org/package=lars

See Also

boss.fit

Index