Package 'MetaGIM'

December 7, 2022

Type Package

Title Integrative Analysis of Individual-Level Data and High-Dimensional Summary Statistics

Version 1.0.0

Date 2022-12-07

Author Han Zhang, Kai Yu, Sheng Fu

Maintainer Bill Wheeler <wheelerb@imsweb.com>

Depends R (>= 3.5.0)

Imports numDeriv

Description A general divide-and-conquer integration procedure, to combine high-dimensional summary data with individual-level for more efficient statistical inference.

License MIT + file LICENSE

LazyData true

NeedsCompilation no

R topics documented:

MetaGIM-package 1
dat 2
data 3
gim 3
gim_add 7
metagim 8
metagim_rho 8

Index **10**

`MetaGIM-package` *Integrative Analysis of Individual-Level Data and High-Dimensional Summary Statistics*

Description

A general divide-and-conquer integration procedure, to combine high-dimensional summary data with individual-level for more efficient statistical inference.

Details

Researchers usually conduct statistical analyses based on models built on raw data collected from individual participants (individual-level data). There is a growing interest in enhancing inference efficiency by incorporating aggregated summary information from other sources, such as summary statistics on genetic markers' marginal associations with a given trait generated from genome-wide association studies. However, combining high-dimensional summary data with individual-level data using existing integrative procedures can be challenging. This package overcomes this obstacle using a divide-and-conquer strategy by breaking the task into easier parallel jobs, with each integrating a small proportion of summary data.

Author(s)

Sheng Fu, Kai Yu

References

Fu, S., Deng, L., Zhang, H., Wheeler, W., Qin, J., Yu, K. (2022) Integrative Analysis of Individual-Level Data and High-Dimensional Summary Statistics. Submitted

Zhang, H., Deng, L., Schiffman, M., Qin, J., Yu, K. (2020) Generalized integration model for improved statistical inference by leveraging external summary data.
Biometrika. asaa014, https://doi.org/10.1093/biomet/asaa014

`dat` *Data for example in* `gim`

Description

`dat` is a data frame used in the example of `gim`.

Usage

```
data("dat")
```

Format

A data frame with 4000 observations on the following 6 variables.

`y` a continuous outcome

`d` a binary outcome

`x1` a numeric variable

`x2` a numeric variable

`x3` a numeric variable

`x4` a character variable

Details

This is a dataset from which internal and external data are extracted for the example.

data *Data for examples*

Description

Data for examples

Details

The objects consist of `data`, `score.add`, `models`, and `nsample`.

gim *Fitting Generalized Integration Models*

Description

`gim` is used to fit generalized integration models, which assume linear or logistic regression model on an (internal) individual-level data, while integrating auxiliary or summary information of relevant variables that are estimated from external data, on which different working models could be assumed. `gim` can work even if partial information from working models are available. Compared to conventional regression model, e.g., `glm`, that is based on internal data, the estimate of `gim` method gains additional power by making maximum use of all kinds of available data.

Usage

```
gim(formula, family, data, model, nsample = NULL,
    ncase = NULL, nctrl = NULL, ref = NULL, ...)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted on the given dataset. More details of model specification are illustrated in 'Details' and 'Examples'.
`family`	a character. `"gaussian"` for linear regression. For binary outcome fitted by logistic regression, use `"binomial"` for random sample, or `"case-control"` for case-control data. `gim` employs different methods to make inference on random sample and case-control data. If your data are collected in case-control studies, do not use `"binomial"`, otherwise inference may be problematic.
`data`	a data frame containing all variables that are specified in `formula` and `model`. Incomplete lines will be discarded.
`model`	a list describing auxiliary information and working models that are used to generate such information. See 'Details' and 'Examples' for more details.
`nsample`	a matrix specifying the number of samples shared in datasets that are used to fit the working models given in `model`. Specify this argument when `family` is `"gaussian"` or `"binomial"`, otherwise `NULL`. See 'Details' and 'Examples' for more details.

ncase — a matrix specifying the number of cases shared in datasets that are used to fit the working models given in model. Specify this argument when family is "case-control", otherwise NULL. ncase and nctrl should be specified simultaneously. See 'Details' and 'Examples' for more details.

nctrl — a matrix specifying the number of controls shared in datasets that are used to fit the working models given in model. Specify this argument when family is case-control, otherwise NULL. See 'Details' and 'Examples' for more details.

ref — a data frame containing the covariates specified in formula and model. It is the reference sample for modeling summary statistics in model. This assumes that ref is sampled from the external population. By default it is NULL which means that the internal and external populations are the same, therefore gim will use data as the reference. Outcome could be absent or missing in ref because gim will anyway ignore it. See 'Details' for more details.

... — for test purpose, use its default value.

Details

formula formula is the model to be used to fit a conventional regression model if no additional information is available. It could be very general as long as it is acceptable to the glm or lm functions. It can eliminate the intercept, y ~ .-1, or involve arithmetic expressions, e.g., log(x), or other operators like * for interactions as.factor(x1)*I(x2 > 0).

model Summary information are calculated on data of external studies, but we do not have access to their raw data. Instead, estimates from working model fitted on external data are given (e.g., reported in literature). The argument model is a list, each component contains information of a working model. Specifically, a component is also a list of two entries form and info, where form is a formula representing the fitted working model, and info is a data frame with two columns var and bet, the names of variables and their estimates from the working model, respectively. Usually the estimate of intercept of a working model is unavailable as people fit but do not reporte it. If user is able to provide such an estimate, the name in column var must be "(Intercept)". See below for an example.

Note that multiple working models could be fitted on the same external data, in that case, the summary information of each working model should be given in model separately. For example, on an external dataset, if two models y ~ x1 and y ~ x2 are fitted, then the estimates of x1 and x2 should be given as two components in model. This happens as many research groups can study the same datasets from different angles.

data gim requires an internal dataset data in which individual-level samples are available. Statistically, this data is critical to provide information of correlation between covariates. This data is also known as the reference data in the literatures. Since general formula is supported in gim, it is important to provide variables in data so that R can find columns of all variables parsed from formulas in formula and model. Read vignettes (upcoming) for more examples about how to create a proper data for gim. We will also release a function to help users with this. gim will discard incomplete lines in data.

nsample Some of summary information can be calculated from datasets that share samples. Ignoring this will lead to underestimated standard error. For example, if a dataset is studied by two different models, the estimates from these two models are not independent but highly correlated. Therefore, this correlation must be properly handled when calculating the standard error of gim estimate, from which a hypothesis testing is conducted. nsample is a squared matrix of dimension p, which is equal to the length of model. Thus, the (i,i) entry in nsample is the number of samples used in fitting the working model specified in model[[i]]$formula, while the (i,k) entry is the number of samples that are involved in fitting working models model[[i]]$formula and model[[k]]$formula. For example, if two working models, e.g., y ~ x1 and y ~ x2 are fitted on

the same dataset of 100 samples, then nsample is a matrix of all entries being 100. Read example below and vignettes (upcoming) for more examples.

ncase and nctrl Specify these two arguments when data are sampled from case-control studies. Refer to nsample for their formats.

ref By default, ref is NULL if it is not specified explicitly. This assumes that the internal and external populations are the same, and gim will assign data to ref implicitly. If this assumption holds, and you have additional covariates data (no outcome), e.g. add.ref, that also comes from the internal population, you can specified ref as rbind(data, add.ref) where the column of missing outcome in add.ref is set as NA. You can also rbind data and add.ref, with outcome in data being deleted. If the external population is different from the internal population, you have to assign add.ref to ref as reference.

Value

gim returns an object of class "gim". The function summary can be used to print a summary of the results. We will support the use of anova in later versions.

The generic accessor functions coef, confint, and vcov can be used to extract coefficients, confidence intervals, and variance-covariance of estimates from the object returned by gim.

An object of class "gim" is a list containing the following components:

coefficients	a named vector of coefficients
vcov	the variance-covariance matrix of estimates, including the intercept
sigma2	estimated variance of error term in a linear model. Only available for the gaussian family
call	the matched call
V.bet	the variance-covariance matrix of external estimate bet in model

Author(s)

Han Zhang

References

Zhang, H., Deng, L., Schiffman, M., Qin, J., Yu, K. (2020) Generalized integration model for improved statistical inference by leveraging external summary data. Biometrika. asaa014, https://doi.org/10.1093/biomet/as

Examples

```
## An artificial dataset is lazyloaded to illustrate the concept of GIM method
## It contains:
## A continuous outcome y.
## Four covariates x1, x2, x3, x4 (character).
## A binary outcome d

head(dat)

## internal data of 500 samples
dat0 <- dat[1:500, ]

## three external datasets.
## dat2 and dat3 share some samples
dat1 <- dat[501:1500, c('y', 'x1', 'x2')]
```

```
dat2 <- dat[1501:2500, c('y', 'x1', 'x3', 'x4')]
dat3 <- dat[2001:3000, c('y', 'x3', 'x4')]

## four working models are fitted
form1 <- 'y ~ I(x1 < 0) + I(x2 > 0)'
form2 <- 'y ~ x3 + x4'
form3 <- 'y ~ I(x4 == "a")'
form4 <- 'y ~ sqrt(x3)'

## two working models are fitted on dat3
## thus nsample is a 4x4 matrix
nsample <- matrix(c(1000, 0, 0, 0,
                    0, 1000, 500, 500,
                    0, 500, 1000, 1000,
                    0, 500, 1000, 1000),
                  4, 4)

fit1 <- summary(lm(form1, dat1))$coef
fit2 <- summary(lm(form2, dat2))$coef
fit3 <- summary(lm(form3, dat3))$coef ## <-- dat3 is used twice
fit4 <- summary(lm(form4, dat3))$coef ## <-- dat3 is used twice

options(stringsAsFactors = FALSE)
model <- list()
## partial information is available
model[[1]] <- list(form = form1,
                   info = data.frame(var = rownames(fit1)[2],
                                     bet = fit1[2, 1]))

## intercept is provided, but miss estimate of a covariate
model[[2]] <- list(form = form2,
                   info = data.frame(var = rownames(fit2)[1:2],
                                     bet = fit2[1:2, 1]))

model[[3]] <- list(form = form3,
                   info = data.frame(var = rownames(fit3)[2],
                                     bet = fit3[2, 1]))

model[[4]] <- list(form = form4,
                   info = data.frame(var = rownames(fit4)[2],
                                     bet = fit4[2, 1]))

form <- 'y ~ I(x1 < 0) + I(x1 > 1) + x2 * x4 + log(x3) - 1'
fit <- gim(form, 'gaussian', dat0, model, nsample)

summary(fit)
coef(fit)
confint(fit)

# one can compare the gim estimates with those estimated from internal data
fit0 <- lm(form, dat0)
summary(fit0)

# by default, covariates in dat is used as reference in gim
# which assumes that the external and internal populations are the same
fit1 <- gim(form, 'gaussian', dat0, model, nsample, ref = dat0)
all(coef(fit) == coef(fit1)) # TRUE
```

```
# if additional reference is available,
# and it comes from the internal population from which dat is sampled
# gim can use it
add.ref <- dat[3001:3500, ]
add.ref$y <- NA ## <-- outcome is unavailable in reference
ref <- rbind(dat0, add.ref)
fit2 <- gim(form, 'gaussian', dat0, model, nsample, ref = ref)

# if the external population is different from the internal population
# then reference for summary data specified in model needs to be provided
ext.ref <- dat[3501:4000, ] ## <-- as an example, assume ext.ref is different
                            ##    from dat0
fit3 <- gim(form, 'gaussian', dat0, model, nsample, ref = ext.ref)
```

---

gim_add *gim_add*

Description

Integrating High-Dimensional Summary Statistics

Usage

```
gim_add(models,group,sample.info,form,family,data,scores)
```

Arguments

models	A list of external model and summary statistics.
group	A list of partitioned sample id numbers.
sample.info	A list of two matrices specifying the number of cases/controls shared in datasets that are used to fit the working models given in models.
form	A formula.
family	Model family.
data	A data frame containing all variables that are specified in formula and models.
scores	A matrix of independent scores based on the true batch assignment.

Examples

```
data(data, package="MetaGIM")

# True batch assignment
M      <- ncol(data)-2
bk     <- 10
group2 <- split(1:M, ceiling((1:M)/bk))

gim_add(models, group2, nsample, y~score, "case-control", data, score.add)
```

metagim *MetaGIM*

Description

Integrating High-Dimensional Summary Statistics

Usage

```
metagim(models,group,sample.info,form,family,data)
```

Arguments

models	A list of external model and summary statistics.
group	A list of partitioned sample id numbers.
sample.info	A list of two matrices specifying the number of cases/controls shared in datasets that are used to fit the working models given in models.
form	A formula.
family	Model family.
data	A data frame containing all variables that are specified in formula and models.

Examples

```
data(data, package="MetaGIM")

# Random assignment into 10 batches
set.seed(0)
M      <- ncol(data)-2
bk     <- 10
M0     <- sample(1:M)
group1 <- split(M0, ceiling((1:M)/bk))

metagim(models, group1, nsample, y~score, "case-control", data)
```

metagim_rho *metagim_rho*

Description

Integrating High-Dimensional Summary Statistics

Usage

```
metagim_rho(models,group,sample.info,form,family,data,cut0)
```

Arguments

models	A list of external model and summary statistics.
group	A list of partitioned sample id numbers.
sample.info	A list of two matrices specifying the number of cases/controls shared in datasets that are used to fit the working models given in models.
form	A formula.
family	Model family.
data	A data frame containing all variables that are specified in formula and models.
cut0	A value used for an ill-conditioned variance-covariance matrix of theta

Examples

```
data(data, package="MetaGIM")

# Random assignment into 10 batches
set.seed(0)
M      <- ncol(data)-2
bk     <- 10
M0     <- sample(1:M)
group1 <- split(M0, ceiling((1:M)/bk))

# True batch assignment
group2 <- split(1:M, ceiling((1:M)/bk))

metagim_rho(models, group1, nsample, y~score, "case-control", data, 100)
metagim_rho(models, group2, nsample, y~score, "case-control", data, 100)
```

Index

∗ **Empirical likelihood, Estimating equations, High dimensionality, Summary data**
MetaGIM-package, 1
∗ **Generalized Integration Model, Case-control study, Divide-and-conquer scheme,**
MetaGIM-package, 1
∗ **datasets**
dat, 2
∗ **data**
data, 3

anova, *5*

coef, *5*
confint, *5*

dat, 2
data, 3

formula, *3*

gim, *2*, 3
gim_add, 7
glm, *3*

metagim, 8
MetaGIM-package, 1
metagim_rho, 8
models *(*data*)*, 3

nsample *(*data*)*, 3

score.add *(*data*)*, 3
summary, *5*

vcov, *5*