Package ‘iWeigReg’

May 20, 2022

Type Package

Title Improved Methods for Causal Inference and Missing Data Problems

Version 1.1

Date 2022-05-19

Author Zhiqiang Tan and Heng Shu

Maintainer Zhiqiang Tan <`ztan@stat.rutgers.edu`>

URL `http://www.stat.rutgers.edu/~ztan`

Description Improved methods based on inverse probability weighting and outcome regression for causal inference and missing data problems.

Depends R (>= 2.9.1), MASS (>= 7.2-1), trust

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-20 13:50:02 UTC

R topics documented:

`iWeigReg-package` *A R package for improved methods for causal inference and missing data problems*

Description

Improved methods based on inverse probability weighting and outcome regression for causal inference and missing data problems.

Details

The R package `iWeigReg` – version 1.0 can be used for two main tasks:

- to estimate the mean of an outcome in the presence of missing data,
- to estimate the average treatment effect in causal inference.

There are 4 functions provided for the first task:

- `mn.lik`: the non-calibrated (or non-doubly robust) likelihood estimator in Tan (2006),
- `mn.clik`: the calibrated (or doubly robust) likelihood estimator in Tan (2010),
- `mn.reg`: the non-calibrated (or non-doubly robust) regression estimator,
- `mn.creg`: the calibrated (or doubly robust) regression estimator in Tan (2006).

In parallel, there are also 4 functions for the second task, `ate.lik`, `ate.clik`, `ate.reg`, and `ate.creg`. Currently, the treatment is assumed to be binary (i.e., untreated or treated). Extensions to multi-valued treatments will be incorporated in later versions.

In general, the function recommended to use is the calibrated (or doubly robust) likelihood estimator, `mn.clik` or `ate.clik`, which is a two-step procedure with the first step corresponding to the non-calibrated (or non-doubly robust) likelihood estimator. The calibrated (or doubly robust) regression estimator, `mn.creg` or `ate.creg`, is a close relative to the calibrated likelihood estimator, but may sometimes yield an estimate lying outside the sample range, for example, outside the unit interval (0,1) for estimating the mean of a binary outcome.

The package also provides two functions, `mn.HT` and `ate.HT`, for the Horvitz-Thompson estimator, i.e., the unaugmented inverse probability weighted estimator. These functions can be used for balance checking.

See the vignette for more details.

ate.clik *Calibrated likelihood estimator for the causal-inference setup*

Description

This function implements the calibrated (or doubly robust) likelihood estimator of the average treatment effect in causal inference in Tan (2010), Biometrika.

Usage

```
ate.clik(y, tr, p, g0,g1, X=NULL, evar=TRUE, inv=”solve”)
```

Arguments

`y` A vector of observed outcomes.

`tr` A vector of treatment indicators (=1 if treated or 0 if untreated).

`p` A vector of known or fitted propensity scores.

`g0` A matrix of calibration variables for treatment 0 (see the details).

`g1` A matrix of calibration variables for treatment 1 (see the details).

`X` The model matrix for the propensity score model, assumed to be logistic (set `X=NULL` if `p` is known or treated to be so).

`evar` Logical; if `FALSE`, no variance estimation.

`inv` Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The two-step procedure in Tan (2010, Section 5.4) is used when dealing with estimated propensity scores. The first step corresponds to the non-calibrated (or non-doubly robust) likelihood estimator implemented in `ate.lik`.

The columns of `g0` (or respectively `g1`) correspond to calibration variables for treatment 0 (or treatment 1), which can be specified to include a constant and the fitted outcome regression function for treatment 0 (or treatment 1). See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted treatment-specific mean should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix `X` does not need to be provided and can be set to `NULL`, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix `X` is provided, then the "score," `(tr-p)X`, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated means for treatments 1 and 0.

`diff` The estimated average treatment effect.

`v` The estimated variances of `mu`, if `evar=TRUE`.

`v.diff` The estimated variance of `diff`, if `evar=TRUE`.

`w` A matrix of two columns, giving calibrated weights for treatments 1 and 0 respectively.

`lam` A matrix of two columns, giving lambda maximizing the log-likelihood for treatments 1 and 0 respectively.

`norm` A vector of two elements, giving the maximum norm (i.e., L_∞ norm) of the gradient of the log-likelihood at the maximum for treatments 1 and 0 respectively.

`conv` A vector of two elements, giving convergence status from *trust* for treatments 1 and 0 respectively.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, http://www.stat.rutgers.edu/~ztan.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

eta0.glm <- glm(y ~ x, subset=tr==0,
               family=y.fam, control=glm.control(maxit=1000))
eta0.hat <- predict.glm(eta0.glm,
               newdata=data.frame(x=x), type="response")
```

```
#ppi.hat treated as known
out.lik <- ate.clik(y, tr, ppi.hat,
                    g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat))
out.lik$diff
out.lik$v.diff

#ppi.hat treated as estimated (see the details)
out.lik <- ate.clik(y, tr, ppi.hat,
                    g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat), X)
out.lik$diff
out.lik$v.diff
```

ate.creg *Calibrated regression estimator for the causal-inference setup*

Description

This function implements the calibrated (or doubly robust) regression estimator of the average treatment effect in causal inference in Tan (2006), JASA.

Usage

```
ate.creg(y, tr, p, g0,g1, X=NULL, evar=TRUE, inv="solve")
```

Arguments

y	A vector of observed outcomes.
tr	A vector of treatment indicators (=1 if treated or 0 if untreated).
p	A vector of known or fitted propensity scores.
g0	A matrix of calibration variables for treatment 0 (see the details).
g1	A matrix of calibration variables for treatment 1 (see the details).
X	The model matrix for the propensity score model, assumed to be logistic (set X=NULL if p is known or treated to be so).
evar	Logical; if FALSE, no variance estimation.
inv	Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The columns of g0 (or respectively g1) correspond to calibration variables for treatment 0 (or treatment 1), which can be specified to include a constant and the fitted outcome regression function for treatment 0 (or treatment 1). See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted treatment-specific mean should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix `X` does not need to be provided and can be set to `NULL`, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix `X` is provided, then the "score," `(tr-p)X`, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated means for treatments 1 and 0.

`diff` The estimated average treatment effect.

`v` The estimated variances of `mu`, if `evar=TRUE`.

`v.diff` The estimated variance of `diff`, if `evar=TRUE`.

`b` A matrix of two colums, giving the vector of regression coefficients for treatments 1 and 0 respectively.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, http://www.stat.rutgers.edu/~ztan.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

eta0.glm <- glm(y ~ x, subset=tr==0,
                family=y.fam, control=glm.control(maxit=1000))
eta0.hat <- predict.glm(eta0.glm,
                newdata=data.frame(x=x), type="response")

#ppi.hat treated as known
out.reg <- ate.creg(y, tr, ppi.hat,
                      g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat))
out.reg$diff
out.reg$v.diff

#ppi.hat treated as estimated
out.reg <- ate.creg(y, tr, ppi.hat,
                      g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat), X)
out.reg$diff
out.reg$v.diff
```

ate.HT *Horvitz-Thompson estimator for the causal-inference setup*

Description

This function implements the Horvitz-Thompson estimator of the mean outcome of the average treatment effect in causal inference.

Usage

```
ate.HT(y, tr, p, X=NULL, bal=FALSE)
```

Arguments

y	A vector or a matrix of observed outcomes.
tr	A vector of treatment indicators (=1 if treated or 0 if untreated).
p	A vector of known or fitted propensity scores.
X	The model matrix for the propensity score model, assumed to be logistic (set X=NULL if p is known or treated to be so).
bal	Logical; if TRUE, the function is used for checking balance (see the details).

Details

Variance estimation is based on asymptotic expansions, allowing for misspecification of the propensity score model.

For balance checking with bal=TRUE, the input y should correpond to the covariates for which balance is to be checked, and the output mu gives the differences between the Horvitz-Thompson estimates and the overall sample means for these covariates.

Value

`mu` The estimated means for treatments 1 and 0 or, if `bal=TRUE`, their differences from the overall sample means.

`diff` The estimated average treatment effect.

`v` The estimated variances of `mu`.

`v.diff` The estimated variance of `diff`.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#ppi.hat treated as known
out.HT <- ate.HT(y, tr, ppi.hat)
out.HT$diff
out.HT$v.diff

#ppi.hat treated as estimated
out.HT <- ate.HT(y, tr, ppi.hat, X)
out.HT$diff
out.HT$v.diff

#balance checking
out.HT <- ate.HT(x, tr, ppi.hat, X, bal=TRUE)
out.HT$mu
out.HT$v

out.HT$mu/ sqrt(out.HT$v)   #t-statistic
```

ate.lik — *Non-calibrated likelihood estimator for the causal-inference setup*

Description

This function implements the non-calibrated (or non-doubly robust) likelihood estimator of the average treatment effect in causal inference in Tan (2006), JASA.

Usage

```
ate.lik(y, tr, p, g0,g1, X=NULL, evar=TRUE, inv=”solve”)
```

Arguments

`y` A vector of observed outcomes.

`tr` A vector of treatment indicators (=1 if treated or 0 if untreated).

`p` A vector of known or fitted propensity scores.

`g0` A matrix of calibration variables for treatment 0 (see the details).

`g1` A matrix of calibration variables for treatment 1 (see the details).

`X` The model matrix for the propensity score model, assumed to be logistic (set `X=NULL` if `p` is known or treated to be so).

`evar` Logical; if `FALSE`, no variance estimation.

`inv` Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The columns of `g0` (or respectively `g1`) correspond to calibration variables for treatment 0 (or treatment 1), which can be specified to include a constant and the fitted outcome regression function for treatment 0 (or treatment 1). See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted treatment-specific mean should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix `X` does not need to be provided and can be set to `NULL`, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix `X` is provided, then the "score," `(tr-p)X`, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated means for treatments 1 and 0.

`diff` The estimated average treatment effect.

`v` The estimated variances of `mu`, if `evar=TRUE`.

`v.diff` The estimated variance of `diff`, if `evar=TRUE`.

`w` The vector of calibrated weights.

`lam` The vector of lambda maximizing the log-likelihood.

`norm` The maximum norm (i.e., L_∞ norm) of the gradient of the log-likelihood at `lam`.

`conv` Convergence status from *trust*.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, http://www.stat.rutgers.edu/~ztan.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

eta0.glm <- glm(y ~ x, subset=tr==0,
               family=y.fam, control=glm.control(maxit=1000))
eta0.hat <- predict.glm(eta0.glm,
               newdata=data.frame(x=x), type="response")

#ppi.hat treated as known
out.lik <- ate.lik(y, tr, ppi.hat,
                     g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat))
```

```
out.lik$diff
out.lik$v.diff

#ppi.hat treated as estimated
out.lik <- ate.lik(y, tr, ppi.hat,
                   g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat), X)
out.lik$diff
out.lik$v.diff
```

ate.reg *Non-calibrated regression estimator for the causal-inference setup*

Description

This function implements the non-calibrated (or non-doubly robust) regression estimator of the average treatment effect in causal inference.

Usage

```
ate.reg(y, tr, p, g0,g1, X=NULL, evar=TRUE, inv=”solve”)
```

Arguments

`y` A vector of observed outcomes.

`tr` A vector of treatment indicators (=1 if treated or 0 if untreated).

`p` A vector of known or fitted propensity scores.

`g0` A matrix of calibration variables for treatment 0 (see the details).

`g1` A matrix of calibration variables for treatment 1 (see the details).

`X` The model matrix for the propensity score model, assumed to be logistic (set `X=NULL` if `p` is known or treated to be so).

`evar` Logical; if `FALSE`, no variance estimation.

`inv` Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The columns of `g0` (or respectively `g1`) correspond to calibration variables for treatment 0 (or treatment 1), which can be specified to include a constant and the fitted outcome regression function for treatment 0 (or treatment 1). See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted treatment-specific mean should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix `X` does not need to be provided and can be set to `NULL`, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix `X` is provided, then the "score," `(tr-p)X`, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions similar to those for `ate.creg` in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated means for treatments 1 and 0.

`diff` The estimated average treatment effect.

`v` The estimated variances of `mu`, if `evar=TRUE`.

`v.diff` The estimated variance of `diff`, if `evar=TRUE`.

`b` A matrix of two colums, giving the vector of regression coefficients for treatments 1 and 0 respectively.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, http://www.stat.rutgers.edu/~ztan.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

eta0.glm <- glm(y ~ x, subset=tr==0,
               family=y.fam, control=glm.control(maxit=1000))
eta0.hat <- predict.glm(eta0.glm,
               newdata=data.frame(x=x), type="response")
```

```
#ppi.hat treated as known
out.reg <- ate.reg(y, tr, ppi.hat,
                     g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat))
out.reg$diff
out.reg$v.diff

#ppi.hat treated as estimated
out.reg <- ate.reg(y, tr, ppi.hat,
                     g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat), X)
out.reg$diff
out.reg$v.diff
```

`histw` *Weighted histogram*

Description

This function plots a weighted histogram.

Usage

```
histw(x, w, xaxis, xmin, xmax, ymax,
           bar=TRUE, add=FALSE, col="black", dens=TRUE)
```

Arguments

`x`	A data vector.
`w`	A weight vector, which will be rescaled to sum up to one.
`xaxis`	A vector of cut points.
`xmin`	The minimum of `x` coordinate.
`xmax`	The maximum of `x` coordinate.
`ymax`	The maximum of `y` coordinate.
`bar`	bar plot (if `TRUE`) or line plot.
`add`	if `TRUE`, the plot is added to an existing plot.
`col`	color of lines.
`dens`	if `TRUE`, the histogram has a total area of one.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, misspecified
ppi.glm <- glm(tr~x, family=binomial(link=logit))

ppi.hat <- ppi.glm$fitted

#outcome regression model, correct
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ z, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

eta0.glm <- glm(y ~ z, subset=tr==0,
               family=y.fam, control=glm.control(maxit=1000))
eta0.hat <- predict.glm(eta0.glm,
               newdata=data.frame(x=x), type="response")

#causal inference
out.clik <- ate.clik(y, tr, ppi.hat,
               g0=cbind(1,eta0.hat),g1=cbind(1,eta1.hat))

#balance checking
gp1 <- tr==1
gp0 <- tr==0

par(mfrow=c(2,3))
look <- z1

histw(look[gp1], rep(1,sum(gp1)), xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8)
histw(look[gp0], rep(1,sum(gp0)), xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8, bar=0, add=TRUE, col="red")

histw(look[gp1], 1/ppi.hat[gp1], xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8)
histw(look[gp0], 1/(1-ppi.hat[gp0]), xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8, bar=0, add=TRUE, col="red")

histw(look[gp1], 1/out.clik$w[gp1,1], xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8)
histw(look[gp0], 1/out.clik$w[gp0,2], xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8, bar=0, add=TRUE, col="red")

look <- z2
```

```
histw(look[gp1], rep(1,sum(gp1)), xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8)
histw(look[gp0], rep(1,sum(gp0)), xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8, bar=0, add=TRUE, col=”red”)

histw(look[gp1], 1/ppi.hat[gp1], xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8)
histw(look[gp0], 1/(1-ppi.hat[gp0]), xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8, bar=0, add=TRUE, col=”red”)

histw(look[gp1], 1/out.clik$w[gp1,1], xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8)
histw(look[gp0], 1/out.clik$w[gp0,2], xaxis=seq(-3.5,3.5,.25),
    xmin=-3.5, xmax=3.5, ymax=.8, bar=0, add=TRUE, col=”red”)
```

`KS.data` *A simulated dataset*

Description

A dataset simulated as in Kang and Schafer (2007).

Usage

```
data(KS.data)
```

Format

A data frame containing 1000 rows and 10 columns.

Details

The dataset is generated as follows.

```
set.seed(0)

n <- 1000

z <- matrix(rnorm(4*n, 0, 1), nrow=n)

ppi.tr <- as.vector( 1/(1+exp(-z%*%c(-1,.5,-.25,-.1))) )
tr <- rbinom(n, 1, ppi.tr)

y.mean <- as.vector( 210+z
y <- y.mean+rnorm(n, 0, 1)

x <- cbind(exp(z[,1]/2), z[,2]/(1+exp(z[,1]))+10,
          (z[,1]*z[,3]/25+.6)^3, (z[,2]+z[,4]+20)^2)
x <- t(t(x)/c(1,1,1,400)-c(0,10,0,0))
```

```
KS.data <- data.frame(y,tr,z,x)
colnames(KS.data) <-
   c("y", "tr", "z1", "z2", "z3", "z4", "x1", "x2", "x3", "x4")

save(KS.data, file="KS.data.rda")
```

References

Kang, J.D.Y. and Schafer, J.L. (2007) "Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data," *Statistical Science*, 22, 523-539.

`loglik` *The non-calibrated objective function ("log-likelihood")*

Description

This function computes the objective function, its gradient and its Hessian matrix for the non-calibrated likelihood estimator in Tan (2006), JASA.

Usage

```
loglik(lam, tr, h)
```

Arguments

`lam`	A vector of parameters ("lambda").
`tr`	A vector of non-missing or treatment indicators.
`h`	A constraint matrix.

Value

`value`	The value of the objective function.
`gradient`	The gradient of the objective function.
`hessian`	The Hessian matrix of objective function.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))
p <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

#
g1 <- cbind(1,eta1.hat)
h <- cbind(p, (1-p)*g1)

loglik(lam=rep(0,dim(h)[2]-1), tr=tr, h=h)
```

`loglik.g` *The calibrated objective function ("log-likelihood")*

Description

This function computes the objective function, its gradient and its Hessian matrix for the calibrated likelihood estimator in Tan (2010), Biometrika.

Arguments

`lam`	A vector of parameters ("lambda").
`tr`	A vector of non-missing or treatment indicators.
`h`	A constraint matrix.
`pr`	A vector of fitted propensity scores.
`g`	A matrix of calibration variables.

Value

`value`	The value of the objective function.
`gradient`	The gradient of the objective function.
`hessian`	The Hessian matrix of the objective function.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))
p <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

#
g1 <- cbind(1,eta1.hat)
h <- cbind(p, (1-p)*g1)

loglik.g(lam=rep(0,dim(g1)[2]), tr=tr, h=h, pr=p, g=g1)
```

mn.clik *Calibrated likelihood estimator for the missing-data setup*

Description

This function implements the calibrated (or doubly robust) likelihood estimator of the mean outcome in the presence of missing data in Tan (2010), Biometrika.

Usage

```
mn.clik(y, tr, p, g, X=NULL, evar=TRUE, inv="solve")
```

Arguments

`y` A vector of outcomes with missing data.

`tr` A vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).

`p` A vector of known or fitted propensity scores.

`g` A matrix of calibration variables (see the details).

`X` The model matrix for the propensity score model, assumed to be logistic (set `X=NULL` if `p` is known or treated to be so).

`evar` Logical; if `FALSE`, no variance estimation.

`inv` Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The two-step procedure in Tan (2010, Section 3.3) is used when dealing with estimated propensity scores. The first step corresponds to the non-calibrated (or non-doubly robust) likelihood estimator implemented in `mn.lik`.

The columns of `g` correspond to calibration variables, which can be specified to include a constant and the fitted outcome regression function. See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted mean among "responders" should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix `X` does not need to be provided and can be set to `NULL`, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix `X` is provided, then the "score," `(tr-p)X`, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated mean.

`v` The estimated variance of `mu`, if `evar=TRUE`.

`w` The vector of calibrated weights.

`lam` The vector of lambda maximizing the log-likelihood.

`norm` The maximum norm (i.e., L_∞ norm) of the gradient of the log-likelihood at `lam`.

`conv` Convergence status from *trust*.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, `http://www.stat.rutgers.edu/~ztan`.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#missing data
y[tr==0] <- 0

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

#ppi.hat treated as known
out.lik <- mn.clik(y, tr, ppi.hat, g=cbind(1,eta1.hat))
out.lik$mu
out.lik$v

#ppi.hat treated as estimated
out.lik <- mn.clik(y, tr, ppi.hat, g=cbind(1,eta1.hat), X)
out.lik$mu
out.lik$v
```

`mn.creg` *Calibrated regression estimator for the missing-data setup*

Description

This function implements the calibrated (or doubly robust) likelihood estimator of the mean outcome in the presence of missing data in Tan (2006), JASA.

Usage

```
mn.creg(y, tr, p, g, X=NULL, evar=TRUE, inv="solve")
```

Arguments

`y` A vector of outcomes with missing data.

`tr` A vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).

`p` A vector of known or fitted propensity scores.

`g` A matrix of calibration variables (see the details).

`X` The model matrix for the propensity score model, assumed to be logistic (set `X=NULL` if `p` is known or treated to be so).

`evar` Logical; if `FALSE`, no variance estimation.

`inv` Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The columns of `g` correspond to calibration variables, which can be specified to include a constant and the fitted outcome regression function. See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted mean among "responders" should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix `X` does not need to be provided and can be set to `NULL`, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix `X` is provided, then the "score," `(tr-p)X`, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated mean.

`v` The estimated variance of `mu`, if `evar=TRUE`.

`b` The vector of regression coefficients.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, `http://www.stat.rutgers.edu/~ztan`.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#missing data
y[tr==0] <- 0
```

```
#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

#ppi.hat treated as known
out.reg <- mn.creg(y, tr, ppi.hat, g=cbind(1,eta1.hat))
out.reg$mu
out.reg$v

#ppi.hat treated as estimated
out.reg <- mn.creg(y, tr, ppi.hat, g=cbind(1,eta1.hat), X)
out.reg$mu
out.reg$v
```

`mn.HT` *Horvitz-Thompson estimator for the missing-data setup*

Description

This function implements the Horvitz-Thompson estimator of the mean outcome in the presence of missing data.

Usage

```
mn.HT(y, tr, p, X=NULL, bal=FALSE)
```

Arguments

`y`	A vector or a matrix of outcomes with missing data.
`tr`	A vector of non-missing indicators (=1 if `y` is observed or 0 if `y` is missing).
`p`	A vector of known or fitted propensity scores.
`X`	The model matrix for the propensity score model, assumed to be logistic (set `X=NULL` if `p` is known or treated to be so).
`bal`	Logical; if `TRUE`, the function is used for checking balance (see the details).

Details

Variance estimation is based on asymptotic expansions, allowing for misspecification of the propensity score model.

For balance checking with bal=TRUE, the input y should correpond to the covariates for which balance is to be checked, and the output mu gives the differences between the Horvitz-Thompson estimates and the overall sample means for these covariates.

Value

mu	The estimated mean(s) or, if bal=TRUE, their differences from the overall sample means.
v	The estimated variance(s) of mu.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#missing data
y[tr==0] <- 0

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#ppi.hat treated as known
out.HT <- mn.HT(y, tr, ppi.hat)
out.HT$mu
out.HT$v

#ppi.hat treated as estimated
out.HT <- mn.HT(y, tr, ppi.hat, X)
out.HT$mu
out.HT$v

#balance checking
out.HT <- mn.HT(x, tr, ppi.hat, X, bal=TRUE)
out.HT$mu
out.HT$v
out.HT$mu/ sqrt(out.HT$v)   #t-statistic
```

mn.lik *Non-calibrated likelihood estimator for the missing-data setup*

Description

This function implements the non-calibrated (or non-doubly robust) likelihood estimator of the mean outcome in the presence of missing data in Tan (2006), JASA.

Usage

```
mn.lik(y, tr, p, g, X=NULL, evar=TRUE, inv="solve")
```

Arguments

y	A vector of outcomes with missing data.
tr	A vector of non-missing indicators (=1 if y is observed or 0 if y is missing).
p	A vector of known or fitted propensity scores.
g	A matrix of calibration variables (see the details).
X	The model matrix for the propensity score model, assumed to be logistic (set X=NULL if p is known or treated to be so).
evar	Logical; if FALSE, no variance estimation.
inv	Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The columns of g correspond to calibration variables, which can be specified to include a constant and the fitted outcome regression function. See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted mean among "responders" should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix X does not need to be provided and can be set to NULL, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix X is provided, then the "score," (tr-p)X, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

`mu` The estimated mean.

`v` The estimated variance of `mu`, if `evar=TRUE`.

`w` The vector of calibrated weights.

`lam` The vector of lambda maximizing the log-likelihood.

`norm` The maximum norm (i.e., L_∞ norm) of the gradient of the log-likelihood at `lam`.

`conv` Convergence status from *trust*.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, `http://www.stat.rutgers.edu/~ztan`.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#missing data
y[tr==0] <- 0

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
               family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
               newdata=data.frame(x=x), type="response")

#ppi.hat treated as known
out.lik <- mn.lik(y, tr, ppi.hat, g=cbind(1,eta1.hat))
out.lik$mu
out.lik$v

#ppi.hat treated as estimated
out.lik <- mn.lik(y, tr, ppi.hat, g=cbind(1,eta1.hat), X)
out.lik$mu
out.lik$v
```

---

mn.reg *Non-calibrated regression estimator for the missing-data setup*

Description

This function implements the non-calibrated (or non-doubly robust) likelihood estimator of the mean outcome in the presence of missing data.

Usage

```
mn.reg(y, tr, p, g, X=NULL, evar=TRUE, inv="solve")
```

Arguments

y	A vector of outcomes with missing data.
tr	A vector of non-missing indicators (=1 if y is observed or 0 if y is missing).
p	A vector of known or fitted propensity scores.
g	A matrix of calibration variables (see the details).
X	The model matrix for the propensity score model, assumed to be logistic (set X=NULL if p is known or treated to be so).
evar	Logical; if FALSE, no variance estimation.
inv	Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Details

The columns of g correspond to calibration variables, which can be specified to include a constant and the fitted outcome regression function. See the examples below. In general, a calibration variable is a function of measured covariates selected to exploit the fact that its weighted mean among "responders" should equal to its unweighted population mean.

To estimate the propensity scores, a logistic regression model is assumed. The model matrix X does not need to be provided and can be set to NULL, in which case the estimated propensity scores are treated as known in the estimation. If the model matrix X is provided, then the "score," (tr-p)X, from the logistic regression is used to generate additional calibration constraints in the estimation. This may sometimes lead to unreliable estimates due to multicollinearity, as discussed in Tan (2006). Therefore, this option should be used with caution.

Variance estimation is based on asymptotic expansions similar to those for mn.creg in Tan (2013). Alternatively, resampling methods (e.g., bootstrap) can be used.

Value

mu	The estimated mean.
v	The estimated variance of mu, if evar=TRUE.
b	The vector of regression coefficients.

References

Tan, Z. (2006) "A distributional approach for causal inference using propensity scores," *Journal of the American Statistical Association*, 101, 1619-1637.

Tan, Z. (2010) "Bounded, efficient and doubly robust estimation with inverse weighting," *Biometrika*, 97, 661-682.

Tan, Z. (2013) "Variance estimation under misspecified models," unpublished manuscript, http://www.stat.rutgers.edu/~ztan.

Examples

```
data(KS.data)
attach(KS.data)
z=cbind(z1,z2,z3,z4)
x=cbind(x1,x2,x3,x4)

#missing data
y[tr==0] <- 0

#logistic propensity score model, correct
ppi.glm <- glm(tr~z, family=binomial(link=logit))

X <- model.matrix(ppi.glm)
ppi.hat <- ppi.glm$fitted

#outcome regression model, misspecified
y.fam <- gaussian(link=identity)

eta1.glm <- glm(y ~ x, subset=tr==1,
                family=y.fam, control=glm.control(maxit=1000))
eta1.hat <- predict.glm(eta1.glm,
                newdata=data.frame(x=x), type="response")

#ppi.hat treated as known
out.reg <- mn.reg(y, tr, ppi.hat, g=cbind(1,eta1.hat))
out.reg$mu
out.reg$v

#ppi.hat treated as estimated
out.reg <- mn.reg(y, tr, ppi.hat, g=cbind(1,eta1.hat), X)
out.reg$mu
out.reg$v
```

myinv *Inverse of a matrix*

Description

This function returns the inverse or generalized inverse of a matrix.

Usage

```
myinv(A, type = "solve")
```

Arguments

`A`	A matrix to be inverted.
`type`	Type of matrix inversion, set to "solve" (default) or "ginv" (which can be used in the case of computational singularity).

Value

The inverse of the given matrix `A`.

Index