

Package ‘CBPS’

August 14, 2014

Version 0.9

Date 2014-08-11

Title R Package for Covariate Balancing Propensity Score

Author

Christian Fong <christianfong@stanford.edu>, Marc Ratkovic <ratkovic@princeton.edu>, Kosuke Imai <kimai@princeton.edu>

Maintainer Christian Fong <christianfong@stanford.edu>

Depends R (>= 2.14), MASS, MatchIt, nnet, numDeriv

Imports

Description CBPS is an R package that implements the covariate balancing propensity score proposed by Imai and Ratkovic (2014; JRSSB). The propensity score is estimated such that it maximizes the resulting covariate balance as well as the prediction of treatment assignment. The method, therefore, avoids an iteration between model fitting and balance checking. The package also implements several extensions of the CBPS beyond the cross-sectional, binary treatment setting. The current version implements the CBPS for longitudinal settings so that it can be used in conjunction with marginal structural models (Imai and Ratkovic, 2013a), treatments with three- and four-valued treatment variables, continuous-valued treatments (Fong, Imai, and Ratkovic), and the situation with multiple distinct binary treatments administered simultaneously. In the future it will be extended to other settings including the generalization of experimental and instrumental variable estimates.

LazyLoad yes

LazyData yes

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-08-14 00:49:56

R topics documented:

balance	2
Blackwell	3
CBMSM	4
CBPS	6
LaLonde	10
plot.CBMSM	11
plot.CBPS	12
summary.CBPS	13
vcov.CBPS	14

Index	16
--------------	-----------

balance	<i>Optimal Covariate Balance</i>
---------	----------------------------------

Description

Returns the mean and standardized mean associated with each treatment group, before and after weighting.

Usage

```
## S3 method for class 'CBPS'
balance(object, ...)
```

Arguments

object	A CBPS (or CBMSM) object.
...	Additional arguments to be passed to balance.

Details

For binary and multi-valued treatments as well as marginal structural models, each of the matrices' rows are the covariates and whose columns are the weighted mean, and standardized mean associated with each treatment group. The standardized mean is the weighted mean divided by the standard deviation of the covariate for the whole population. For continuous treatments, returns the absolute Pearson correlation between the treatment and each covariate.

Value

Returns a list of two matrices, "original" (before weighting) and "balanced" (after weighting).

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai.

Examples

```

## Not run:
###
### Example: Optimal Covariate Balance
###
# Run for 4-treatment case
set.seed(1)
# Generate random X and underlying coefficients for probability.
# Determine probs and treatments.
X<-cbind(rep(1,1000), mvrnorm(1000,c(0,0,0),
  Sigma=matrix(c(5,.5,-.03,.5,1,-.27,-.03,-.27,1),3,3)))
beta<-matrix(rnorm(12),4,3)
baseline.prob<-apply(X%%beta,1,function(x) (1+sum(exp(x)))^-1)
prob<-cbind(baseline.prob, exp(X%%beta[,1])*baseline.prob,
exp(X%%beta[,2])*baseline.prob,
exp(X%%beta[,3])*baseline.prob)
treat.latent<-runif(1000)
treat<-ifelse(treat.latent < prob[,1], 1,
  ifelse(treat.latent < (prob[,1] + prob[,2]), 2,
    ifelse(treat.latent < (prob[,1] + prob[,2] + prob[,3]),
      3, 4)))
X<-X[,-1]
fit4<-CBPS(treat ~ X, ATT = FALSE)
balance(fit4)

## End(Not run)

```

Blackwell

Blackwell Data for Covariate Balancing Propensity Score

Description

This data set gives the outcomes as well as treatment assignments and covariates for the example from Blackwell (2013).

Usage

```
Blackwell
```

Format

A data frame consisting of 13 columns (including treatment assignment, time, and identifier vectors) and 570 observations.

Source

Data from the National Supported Work Study. A benchmark matching dataset. `d.gone.neg` is the treatment. `d.gone.neg.l1`, `d.gone.neg.l2`, and `d.gone.neg.l3` are lagged treatment variables. `camp.length`, `deminc`, `base.poll`, `base.und`, and `office` covariates. `year` and `time` are alternative specifications of the time variable. `demName` is the identifier, and `demprcnt` is the outcome.

References

Blackwell, Matthew. (2013). A framework for dynamic causal inference in political science. *American Journal of Political Science* 57, 2, 504-619.

CBMSM	<i>Covariate Balancing Propensity Score (CBPS) for Marginal Structural Models</i>
-------	---

Description

CBMSM estimates propensity scores such that both covariate balance and prediction of treatment assignment are maximized. With longitudinal data, the method returns marginal structural model weights that can be entered directly into a linear model. The method also handles multiple binary treatments administered concurrently.

Usage

```
CBMSM(formula, id, time, data, type="MSM", twostep = TRUE,
msm.variance = "approx", time.vary = FALSE, ...)
CBMSM.fit(treat, X, id, time, MultiBin.fit, twostep,
msm.variance, time.vary, ...)
```

Arguments

<code>formula</code>	A list of formulas of the form <code>treat ~ X</code> . The function assumes that there is one formula for each time, and they are ordered from the first time to the last time.
<code>id</code>	A vector which identifies the unit associated with each row of <code>treat</code> and <code>X</code> .
<code>time</code>	A vector which identifies the time period associated with each row of <code>treat</code> and <code>X</code> .
<code>data</code>	An optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which CBMSM is called.
<code>twostep</code>	Set to <code>TRUE</code> to use a two-step estimator, which will run substantially faster than continuous-updating. Default is <code>FALSE</code> , which uses the continuous-updating estimator described by Imai and Ratkovic (2014).
<code>msm.variance</code>	Default is <code>FALSE</code> , which uses the low-rank approximation of the variance described in Imai and Ratkovic (2014). Set to <code>TRUE</code> to use the full variance matrix.

<code>time.vary</code>	Default is FALSE, which uses the same coefficients across time period. Set to TRUE to fit one set per time period.
<code>treat</code>	A vector of treatment assignments. For N observations over T time periods, the length of treat should be N*T.
<code>X</code>	A covariate matrix. For N observations over T time periods, X should have N*T rows.
<code>type</code>	"MSM" for a marginal structural model, with multiple time periods or "MultiBin" for multiple binary treatments at the same time period.
<code>MultiBin.fit</code>	A parameter for whether the multiple binary treatments occur concurrently (FALSE) or over consecutive time periods (TRUE) as in a marginal structural model. Setting <code>type = "MultiBin"</code> when calling <code>CBMSM</code> will set <code>MultiBin.fit</code> to TRUE when <code>CBMSM.fit</code> is called.
<code>...</code>	Other parameters to be passed through to <code>optim()</code>

Details

Fits covariate balancing propensity scores for marginal structural models.

Value

<code>weights</code>	The optimal weights.
<code>fitted.values</code>	The fitted propensity score for each observation.
<code>y</code>	The treatment vector used.
<code>x</code>	The covariate matrix.
<code>id</code>	The vector id used in <code>CBMSM.fit</code> .
<code>time</code>	The vector time used in <code>CBMSM.fit</code> .
<code>model</code>	The model frame.
<code>call</code>	The matched call.
<code>formula</code>	The formula supplied.
<code>data</code>	The data argument.

Author(s)

Marc Ratkovic, Christian Fong, and Kosuke Imai; The `CBMSM` function is based on the code for version 2.15.0 of the `glm` function implemented in the `stats` package, originally written by Simon Davies. This documentation is likewise modeled on the documentation for `glm` and borrows its language where the arguments and values are the same.

References

- Imai, Kosuke and Marc Ratkovic. 2014. "Covariate Balancing Propensity Score." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*. <http://imai.princeton.edu/research/CBPS.html>
- Imai, Kosuke and Marc Ratkovic. "Robust Estimation of Inverse Probability Weights for Marginal Structural Models." Unpublished Manuscript, Princeton University. <http://imai.princeton.edu/research/MSM.html>

See Also[plot.CBMSM](#)**Examples**

```
## Not run:

data(Blackwell)
name.cands<-sort(unique(Blackwell$demName))
treat.hist<-matrix(NA,nr=length(unique(name.cands)),nc=length(unique(Blackwell$time)))
for(i in 1:length(unique(name.cands))) for(j in 1:length(unique(Blackwell$time))){
  treat.hist[i,j]<-with(Blackwell, d.gone.neg[demName==name.cands[i] & time==j])}
treat.hist.fac<-apply(treat.hist,1,function(x) paste(x, collapse="+"))
cm.treat<-rowSums(treat.hist)

msm1<-CBMSM(d.gone.neg ~ d.gone.neg.l1 + d.gone.neg.l2 + d.neg.frac.l3 + camp.length +
camp.length + deminc + base.poll + as.factor(year) + base.und + office,
id = as.numeric(as.factor(Blackwell$demName)), time = Blackwell$time,
data = Blackwell, type = "MSM", twostep = TRUE, msm.variance = "approx",
time.vary = FALSE)
summary(lm(Blackwell$demprcnt[Blackwell$time==1] ~ treat.hist.fac,
  weights=msm1$w[Blackwell$time==1]))
summary(lm(Blackwell$demprcnt[Blackwell$time==1] ~ cm.treat,
  weights=msm1$w[Blackwell$time==1]))

### Example: Multiple Binary Treatments Administered at the Same Time
n<-200
k<-4
set.seed(1040)
X1<-cbind(1,matrix(rnorm(n*k),ncol=k))

betas.1<-betas.2<-betas.3<-c(2,4,4,-4,3)/5
probs.1<-probs.2<-probs.3<-(1+exp(-X1 %*% betas.1))^-1

treat.1<-rbinom(n=length(probs.1),size=1,probs.1)
treat.2<-rbinom(n=length(probs.2),size=1,probs.2)
treat.3<-rbinom(n=length(probs.3),size=1,probs.3)
treat<-c(treat.1,treat.2,treat.3)
X<-rbind(X1,X1,X1)
time<-c(rep(1,nrow(X1)),rep(2,nrow(X1)),rep(3,nrow(X1)))
id<-c(rep(1:nrow(X1),3))
y<-cbind(treat.1,treat.2,treat.3) %*% c(2,2,2)+X1 %*% c(-2,8,7,6,2) + rnorm(n,sd=5)

multibin1<-CBMSM(treat~X,id=id,time=time,type="MultiBin",twostep=TRUE)
summary(lm(y~-1+treat.1+treat.2+treat.3+X1, weights=multibin1$w))

## End(Not run)
```

Description

CBPS estimates propensity scores such that both covariate balance and prediction of treatment assignment are maximized. The method, therefore, avoids an iterative process between model fitting and balance checking and implements both simultaneously. For cross-sectional data, the method can take continuous treatments and treatments with a control (baseline) condition and either 1, 2, or 3 distinct treatment conditions.

Usage

```
CBPS(formula, data, na.action, ATT = NULL, method = "over",
      iterations = NULL, standardize = TRUE, twostep = TRUE, ...)
CBPS.fit(treat, X, ATT, X.bal = X, method, iterations,
         standardize, twostep, ...)
```

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.
data	An optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which CBPS is called.
na.action	A function which indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of options, and is <code>na.fail</code> if that is unset.
ATT	Set to TRUE to find the average treatment effect on the treated, and FALSE to find the average treatment effect. Default is TRUE. For 3- and 4-valued treatments as well as continuous treatments, only the ATE is available.
treat	A vector of treatment assignments.
X	A covariate matrix.
X.bal	A matrix to be balanced.
method	Choose "over" to fit an over-identified model that combines the propensity score and covariate balancing conditions; choose "exact" to fit a model that only contains the covariate balancing conditions.
iterations	An optional parameter for the maximum number of iterations for the optimization. Default is 1000.
standardize	Default is TRUE, which gets inverse propensity score weights (IPW) as described in Hirano, Imbens, and Ridder (2003). Set to FALSE to return Horvitz-Thompson weights. Does not affect weights for continuous treatments.
twostep	Default is TRUE for a two-step estimator, which will run substantially faster than continuous-updating. Set to FALSE to use the continuous-updating estimator described by Imai and Ratkovic (2014).
...	Other parameters to be passed through to <code>optim()</code>

Details

Fits covariate balancing propensity scores.

Value

<code>coefficients</code>	A named vector of coefficients
<code>sigmasq</code>	The sigma-squared value, for continuous treatments only.
<code>residuals</code>	The working residuals from the final iteration of CBPS
<code>fitted.values</code>	The fitted propensity score.
<code>rank</code>	The numeric rank of the fitted model
<code>family</code>	The family object used.
<code>deviance</code>	Minus twice the log-likelihood of the CBPS fit. Will be lower than the maximum likelihood deviance.
<code>weights</code>	The optimal weights. For the marginal structural model and MultiBins, returns a list in which stabilized ("stabilized") weights, unstabilized ("unstabilized") weights, and unconditional treatment probabilities ("unconditional") are all available. Weights are Horvitz-Thompson if <code>standardize</code> is set to <code>FALSE</code> , and IPW if <code>standardize</code> is set to <code>TRUE</code> .
<code>y</code>	The treatment vector used
<code>x</code>	The covariate matrix
<code>model</code>	The model frame
<code>converged</code>	Convergence value. Returned from the call to <code>optim()</code> .
<code>call</code>	The matched call
<code>formula</code>	The formula supplied.
<code>data</code>	The data argument.
<code>J</code>	The J-statistic at convergence.
<code>mle.J</code>	The J-statistic for the parameters from maximum likelihood estimation.
<code>bal</code>	The balance loss at convergence.
<code>mle.bal</code>	The balance loss for the parameters from maximum likelihood estimation.
<code>df</code>	The degrees of freedom.
<code>var</code>	The covariance matrix, evaluated numerically from <code>optim()</code> .

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai; The CBPS function is based on the code for version 2.15.0 of the `glm` function implemented in the `stats` package, originally written by Simon Davies. This documentation is likewise modeled on the documentation for `glm` and borrows its language where the arguments and values are the same.

References

Imai, Kosuke and Marc Ratkovic. 2014. "Covariate Balancing Propensity Score." *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*. <http://imai.princeton.edu/research/CBPS.html> Fong, Christian and Kosuke Imai. "Covariate Balancing Propensity Scores for General Treatment Regimes." Unpublished Manuscript. <http://imai.princeton.edu/research/files/CBGPS.pdf>

See Also

[summary.CBPS](#)

Examples

```
## Not run:
###
### Example: propensity score matching
###

##Load the LaLonde data
data(LaLonde)
## Estimate CBPS via logistic regression
fit <- CBPS(treat ~ age + educ + re75 + re74 + I(re75==0) + I(re74==0),
data = LaLonde, ATT = TRUE)
summary(fit)
## matching via MatchIt: one to one nearest neighbor with replacement
library(MatchIt)
m.out <- matchit(treat ~ fitted(fit), method = "nearest", data = LaLonde,
replace = TRUE)

### Example: propensity score weighting
###
## Simulation from Kang and Shafer (2007).
set.seed(123456)
n <- 500
X <- mvrnorm(n, mu = rep(0, 4), Sigma = diag(4))
prop <- 1 / (1 + exp(X[,1] - 0.5 * X[,2] + 0.25*X[,3] + 0.1 * X[,4]))
treat <- rbinom(n, 1, prop)
y <- 210 + 27.4*X[,1] + 13.7*X[,2] + 13.7*X[,3] + 13.7*X[,4] + rnorm(n)

##Estimate CBPS with a misspecificied model
X.mis <- cbind(exp(X[,1]/2), X[,2]*(1+exp(X[,1]))^(-1)+10, (X[,1]*X[,3]/25+.6)^3,
(X[,2]+X[,4]+20)^2)
fit1 <- CBPS(treat ~ X.mis, ATT = FALSE)
summary(fit1)

## Horwitz-Thompson estimate
mean(treat*y/fit1$fitted.values)
## Inverse propensity score weighting
sum(treat*y/fit1$fitted.values)/sum(treat/fit1$fitted.values)

rm(list=c("y", "X", "prop", "treat", "n", "X.mis", "fit1"))

### Example: Covariate Balancing Propensity Score - 3 Treatments
##
set.seed(1)
n<-1000
k<-3
X<-mvrnorm(n,rep(0,k), Sigma=matrix(c(5,.5,-.03,.5,1,-.27,-.03,-.27,1),3,3))
beta<-matrix(2*k,k,2)
```

```

prob<-cbind((1+exp(X%%beta[,1])+exp(X%%beta[,2]))^-1,
  exp(X%%beta[,1])*(1+exp(X%%beta[,1])+exp(X%%beta[,2]))^-1,
  exp(X%%beta[,2])*(1+exp(X%%beta[,1])+exp(X%%beta[,2]))^-1)
treat.latent<-runif(n)
treat<-ifelse(treat.latent < prob[,1], 1,
  ifelse(treat.latent < (prob[,1] + prob[,2]), 2, 3))

gamma<-rnorm(ncol(X))
y<-X%%gamma + ifelse(treat == 2, 2, ifelse(treat==3, -1, 0)) + rnorm(1000)

cbpsfit3<-CBPS(treat ~ X)
outfit3<-lm(y ~ as.factor(treat) + X, weights = cbpsfit3$weights)

### Example: Continuous Treatment
set.seed(1)
n<-1000
X<-cbind(mvrnorm(n, mu = rep(0,4), Sigma = diag(4)), rbinom(n,1,runif(1)), rbinom(n,10,0.7))
beta<-rnorm(ncol(X)+1, sd=5)
noise<-rbinom(1,10,0.5)*.50
treat<-cbind(1,X)%%beta+rnorm(n, sd = noise)
prob<-dnorm(treat, mean = cbind(1,X)%%beta, sd = noise, log = TRUE)

treat.effect<-rnorm(1)
effectbeta<-rnorm(ncol(X))
y<-rbinom(n, 1, (1+exp(mean(treat.effect*treat + X%%effectbeta) -
  treat.effect*treat - X%%effectbeta))^-1)

cbpsfitc<-CBPS(treat ~ X)
outfitc<-lm(y ~ treat + X, weights = cbpsfitc$weights)

## End(Not run)

```

LaLonde

LaLonde Data for Covariate Balancing Propensity Score

Description

This data set gives the outcomes as well as treatment assignments and covariates for the econometric evaluation of training programs in LaLonde (1986).

Usage

LaLonde

Format

A data frame consisting of 5 columns (including a treatment assignment vector) and 2787 observations.

Source

Data from the National Supported Work Study. A benchmark matching dataset. Columns consist of an indicator for whether the observed unit was in the experimental subset; an indicator for whether the individual received the treatment; age in years; schooling in years; indicators for black and Hispanic; an indicator for marriage status, one of married; an indicator for no high school degree; and reported earnings in 1974, 1975, and 1978. 1974 and 1975 earnings are pre-treatment. 1978 earnings is taken as the outcome variable.

References

LaLonde, R.J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review* 76, 4, 604-620.

plot.CBMSM	<i>Plotting Covariate Balancing Propensity Score Estimation for Marginal Structural Models</i>
------------	--

Description

Plots the absolute difference in standardized means before and after weighting.

Usage

```
## S3 method for class 'CBMSM'
plot(x, covars = NULL, silent = TRUE, ...)
```

Arguments

x	an object of class “CBMSM”.
covars	Indices of the covariates to be plotted (excluding the intercept). For example, if only the first two covariates from <i>balance</i> are desired, set <i>covars</i> to 1:2. The default is NULL, which plots all covariates.
silent	If set to FALSE, returns the absolute imbalance for each treatment history pair before and after weighting. This helps the user to create his or her own customized plot. Default is TRUE, which returns nothing.
...	Additional arguments to be passed to plot.

Details

Covariate balance is improved if the plot’s points are below the plotted line of $y=x$.

Value

The x-axis gives the imbalance for each covariate-treatment history pair without any weighting, and the y-axis gives the imbalance for each covariate-treatment history pair after CBMSM weighting. Imbalance is measured as the absolute difference in standardized means for the two treatment histories. Means are standardized by the standard deviation of the covariate in the full sample.

Author(s)

Marc Ratkovic and Christian Fong

See Also

[CBMSM](#), [plot](#)

plot.CBPS

Plotting Covariate Balancing Propensity Score Estimation

Description

Plots the absolute difference in standardized means before and after weighting.

Usage

```
## S3 method for class 'CBPS'
plot(x, covars = NULL, silent = TRUE, ...)
```

Arguments

x	an object of class “CBPS”, usually, a result of a call to CBPS.
covars	Indices of the covariates to be plotted (excluding the intercept). For example, if only the first two covariates from balance are desired, set covars to 1:2. The default is NULL, which plots all covariates.
silent	If set to FALSE, returns the absolute imbalance for each contrast pair before and after weighting. This helps the user to create his or her own customized plot. Default is TRUE, which returns nothing.
...	Additional arguments to be passed to plot.

Details

The "Before Weighting" plot gives the balance before weighting, and the "After Weighting" plot gives the balance after weighting.

Value

For binary and multi-valued treatments, plots the absolute difference in standardized means by contrast for all covariates before and after weighting. For continuous treatments, plots the absolute Pearson correlation between the treatment and each covariate.

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai.

See Also

[CBPS](#), [plot](#)

`summary.CBPS`*Summarizing Covariate Balancing Propensity Score Estimation*

Description

Prints a summary of a fitted CBPS object.

Usage

```
## S3 method for class 'CBPS'  
summary(object,...)
```

Arguments

`object` an object of class “CBPS”, usually, a result of a call to `CBPS`.
`...` Additional arguments to be passed to `summary`.

Details

Prints a summary of a CBPS object, in a format similar to `glm`. The variance matrix is calculated from the numerical Hessian at convergence of CBPS.

Value

`call` The matched call.
`deviance.residuals` The five number summary and the mean of the deviance residuals.
`coefficients` A table including the estimate for the each coefficient and the standard error, z-value, and two-sided p-value for these estimates.
`J` Hansen’s J-Statistic for the fitted model.
`Log-Likelihood` The log-likelihood of the fitted model.

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai.

See Also

[CBPS](#), [summary](#)

`vcov.CBPS`*Calculate Variance-Covariance Matrix for a Fitted CBPS Object*

Description

`vcov.CBPS` Returns the variance-covariance matrix of the main parameters of a fitted CBPS object.

Usage

```
## S3 method for class 'CBPS'  
vcov(object, ...)
```

Arguments

<code>object</code>	An object of class <code>formula</code> (or one that can be coerced to that class): a symbolic description of the model to be fitted.
<code>...</code>	Additional arguments to be passed to <code>vcov.CBPS</code>

Details

This is the CBPS implementation of the generic function `vcov()`.

Value

A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.

Author(s)

Christian Fong, Marc Ratkovic, and Kosuke Imai.

References

This documentation is modeled on the documentation of the generic `vcov`.

See Also

[vcov](#)

Examples

```
## Not run:  
###  
### Example: Variance-Covariance Matrix  
###  
  
##Load the LaLonde data  
data(LaLonde)  
## Estimate CBPS via logistic regression
```

```
fit <- CBPS(treat ~ age + educ + re75 + re74 + I(re75==0) + I(re74==0),
  data = LaLonde, ATT = TRUE)
## Get the variance-covariance matrix.
vcov(fit)

## End(Not run)
```

Index

*Topic **datasets**

Blackwell, [3](#)

LaLonde, [10](#)

balance, [2](#)

Blackwell, [3](#)

CBMSM, [4](#), [12](#)

CBPS, [6](#), [12](#), [13](#)

LaLonde, [10](#)

plot, [12](#)

plot.CBMSM, [6](#), [11](#)

plot.CBPS, [12](#)

print.CBPS (CBPS), [6](#)

summary, [13](#)

summary.CBPS, [9](#), [13](#)

vcov, [14](#)

vcov.CBPS, [14](#)