Package ‘CausalSelect’

August 16, 2025

Type Package

Title Variable Selection for High-Dimensional Causal Inference

Version 1.0

Date 2025-05-26

Author Rime NAAMAN and Ismaila Balde

Maintainer Ismaila Baldé <ismaila.balde@umoncton.ca>

Description
Variable selection methods for high-dimensional causal inference, including the Outcome-Adaptive Lasso (OAL), Generalized Outcome-
Adaptive Lasso (GOAL), CBS, SIS_OAL, and SIS_GOAL.

License GPL-3

RoxygenNote 7.3.2

Imports MASS,
stats,
Ball,
glmnet

Encoding UTF-8

Contents

CausalSelect-package *CausalSelect: Variable Selection for High-Dimensional Causal Inference*

Description

The CausalSelect package provides variable selection tools tailored for high-dimensional causal inference problems. It implements OAL, GOAL, CBS, SIS_OAL, and SIS_GOAL.

Details

In high-dimensional causal inference settings where the number of covariates p is much larger than the sample size n ($p \gg n$), variable selection is crucial to control for confounding.

This package includes the following methods:

- OAL: The original Outcome-Adaptive Lasso as proposed by Shortreed and Ertefaie (2017). It uses outcome-informed variable selection in the propensity score model to estimate causal effects.
- GOAL: The Generalized Outcome-Adaptive Lasso, proposed by Baldé et al. (2023), improves OAL.
- SIS_GOAL: A screening-based extension of GOAL for ultra-high dimensional settings, using Sure Independence Screening (SIS) to preselect variables before applying GOAL.
- SIS_OAL: A screening-based extension of OAL that improves computational efficiency in high-dimensional scenarios.
- CBS: The Causal Ball Screening method, which combines conditional correlation-based screening, weighted adaptive lasso, and doubly robust estimation for treatment effect estimation.

The SIS-based approaches are especially useful when $p \gg n$, where matrix operations (e.g., Cholesky decomposition) become unstable or computationally prohibitive.

Author(s)

Rime Naaman and Ismaila Balde

References

Shortreed, S. M., & Ertefaie, A. (2017). *Outcome-adaptive Lasso: Variable selection for causal inference*. **Biometrics**, 73(4), 1111–1122.

Baldé, I., Lefebvre, G., & Ernst, O. (2023). *Generalized Outcome-Adaptive Lasso for Causal Inference in High Dimensions*. arXiv preprint arXiv:2303.08882.

See Also

OAL, GOAL, SIS_GOAL, SIS_OAL, CBS

`CBS` *Ultra-high dimensional variable selection for doubly robust causal inference*

Description

Performs variable screening using the Causal Ball Screening (CBS) procedure and estimates the average treatment effect (ATE) using a double robust estimator.

Usage

```
CBS(X, A, Y, d.n = 30, alpha = 0.05)
```

Arguments

`X`	A numeric matrix of covariates (n rows by p columns).
`A`	A binary treatment indicator vector of length n (0 or 1).
`Y`	A numeric outcome vector of length n.
`d.n`	Number of variables to retain after the CBS screening step (default is 30).
`alpha`	Significance level for confidence interval computation (default is 0.05).

Details

This function implements the full CBS method for confounder selection based on the ball covariance metric, followed by a double robust estimation procedure.

Value

A numeric vector containing:

`CBS`	Estimated average treatment effect (ATE).
`mCBS`	A binary vector indicating which variables (among the top `d.n`) were selected in the final model.

Note

Requires the functions `Causal.cor`, `create_weights`, `wAMD_function`, and `DR` to be defined in the working environment. Also depends on the **glmnet** package.

Author(s)

Rime Naaman

References

Tang, D., Kong, D., Pan, W., Wang, L. (2023) Ultra-high dimensional variable selection for doubly robust causal inference. Biometrics, 79, 903–914. https://doi.org/10.1111/biom.13625

Examples

```
## Generate a multivariate normal X matrix
# set information for simulating coviariates
mean_x = 0
sig_x = 1

# pairwise correlation between covariates
rho = 0

# set number of monte carlo (MC) simulation
S=5

# sample size
n = naug = 300

# total number of predictors
p = 1000

# threshold for screening (taken from Fan and Lv (2008))
#d.n=floor(n/log(n))
#threshold = min(d.n,p)

# set information for data augmentation
#paug=threshold

# note:  pC, pP and pI are number of confounders,
#pure predictors of outcome and pure predictors of exposure, respectively
pC = pP = pI  = 2

# pS number of spurious covariates
pS = p - (pC+pP+pI)

# list of all p variables
var.list = c(paste("Xc",1:pC,sep=""),
             paste("Xp",1:pP,sep=""),
             paste("Xi",1:pI,sep=""),
             paste("Xs",1:pS,sep=""))

# list of threshold variables
#var.list_Ball = c(paste("X",1:threshold,sep=""))

# set strength of relationship between covariates and outcome
beta_v =  c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )

# Set strength of relationship between covariates and treatment
alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
names(beta_v) = names(alpha_v) = var.list

# set true average treatment effect (taken from Shortreed and Ertefaie (2017))
bA = 0


Data=NULL
### simulate data
Sigma_x = matrix(rho*sig_x^2,nrow=length(var.list),ncol=length(var.list))
diag(Sigma_x) = sig_x^2
```

```
Mean_x = rep(mean_x,length(var.list))
Data = as.data.frame(MASS::mvrnorm(n = n,mu=Mean_x,Sigma = Sigma_x,empirical = FALSE))
names(Data) = var.list

X=Data



## Data generation setting
## alpha: Xc's scale is 0.2 0.2 and Xi's scale is 0.3 0.3
## so this refers that there is 2 Xc and Xi
## beta: Xc's scale is 2 2 and Xp's scale is 2 2
## so this refers that there is 2 Xc and Xp
## rest with following setup
Data_fun <- Data_G(X, alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
, beta_v = c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
, bA = 0, sig_x=sig_x, linearY=TRUE,pC=2,pP=2,pI=2)

X=Data_fun$X
A=Data_fun$A
Y=Data_fun$Y
res2=CBS(X, A, Y, d.n = 30, alpha = 0.05)
```

Data_G *Simulate Treatment and Outcome Data from Covariates*

Description

This function generates simulated data for estimating the average treatment effect (ATE).

Usage

```
Data_G(X, alpha_v, beta_v, bA, sig_x, linearY = TRUE, pC, pP, pI)
```

Arguments

X	A matrix of covariates ($n \times p$).
alpha_v	A vector of coefficients for the treatment assignment model.
beta_v	A vector of coefficients for the outcome model.
bA	True treatment effect to be added to the outcome.
linearY	Logical. If TRUE, generates a linear outcome. (Currently always linear).
sig_x	Standard deviation of the covariates in the simulated data.
pC	Number of covariates associated with both treatment and outcome.
pP	Number of covariates associated only with the outcome.
pI	Number of covariates associated only with the treatment.

Value

A list containing:

X	Matrix of covariates
A	Binary treatment vector
Y	Continuous outcome vector affected by covariates and treatment effect.

Author(s)

Rime Naaman and Ismaila Baldé

Examples

```
## Generate a multivariate normal X matrix
# set information for simulating coviariates
mean_x = 0
sig_x = 1

# pairwise correlation between covariates
rho = 0

# set number of monte carlo (MC) simulation
S=5

# sample size
n = naug = 30

# total number of predictors
p = 100

# threshold for screening (taken from Fan and Lv (2008))
#d.n=floor(n/log(n))
#threshold = min(d.n,p)

# set information for data augmentation
#paug=threshold

# note:  pC, pP and pI are number of confounders,
#pure predictors of outcome and pure predictors of exposure, respectively
pC = pP = pI  = 2

# pS number of spurious covariates
pS = p - (pC+pP+pI)

# list of all p variables
var.list = c(paste("Xc",1:pC,sep=""),
             paste("Xp",1:pP,sep=""),
             paste("Xi",1:pI,sep=""),
             paste("Xs",1:pS,sep=""))


# list of threshold variables
#var.list_Ball = c(paste("X",1:threshold,sep=""))

# set strength of relationship between covariates and outcome
beta_v =  c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )

# Set strength of relationship between covariates and treatment
alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
names(beta_v) = names(alpha_v) = var.list

# set true average treatment effect (taken from Shortreed and Ertefaie (2017))
bA = 0
```

```
Data=NULL
### simulate data
Sigma_x = matrix(rho*sig_x^2,nrow=length(var.list),ncol=length(var.list))
diag(Sigma_x) = sig_x^2
Mean_x = rep(mean_x,length(var.list))
Data = as.data.frame(MASS::mvrnorm(n = n,mu=Mean_x,Sigma = Sigma_x,empirical = FALSE))
names(Data) = var.list

X=Data



## Data generation setting
## alpha: Xc's scale is 0.2 0.2 and Xi's scale is 0.3 0.3
## so this refers that there is 2 Xc and Xi
## beta: Xc's scale is 2 2 and Xp's scale is 2 2
## so this refers that there is 2 Xc and Xp
## rest with following setup
Data_fun <- Data_G(X, alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
, beta_v = c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
, bA = 0, sig_x=sig_x, linearY=TRUE,pC=2,pP=2,pI=2)
```

GOAL — *Generalized Outcome-Adaptive Lasso (GOAL) for High-Dimensional Causal Inference*

Description

Estimates the average treatment effect (ATE) in high-dimensional settings using the Generalized Outcome-Adaptive Lasso (GOAL), a variable selection method for causal inference that generalizes the Outcome-Adaptive Lasso (OAL) to overcome its limitations in settings with strong collinearity.

Usage

```
GOAL(X, A, Y)
```

Arguments

X	A numeric matrix or data frame of covariates with n rows (observations) and p columns (variables).
A	A binary treatment vector of length n.
Y	A numeric outcome vector of length n.

Details

The Generalized Outcome-Adaptive Lasso (GOAL), proposed by Baldé et al. (2023), extends the Outcome-Adaptive Lasso (OAL) of Shortreed and Ertefaie (2017) for estimating treatment effects in high-dimensional settings. GOAL improves upon OAL by introducing multiple tuning parameters and an augmentation step to enhance stability and variable selection. It satisfies the oracle property, enabling consistent selection of relevant covariates, and is more robust to multicollinearity. Variable selection is based on minimizing the weighted absolute mean difference (wAMD), and the average treatment effect (ATE) is estimated using inverse probability weighting.

Value

A numeric vector of length $p + 1$, containing:

`GOAL`	Estimated average treatment effect (ATE).
`mGOAL`	A binary vector of length p, indicating selected covariates (1 = selected, 0 = not selected).

Note

This function requires auxiliary functions such as `adaptive.lasso`, `lqa.def`, `expit`, `create_weights`, `wAMD_function`, and `ATE_est` to be defined in the environment.

Author(s)

Rime Naaman

References

Baldé, I., Yang, Y. A., & Lefebvre, G. (2023). Reader reaction to "Outcome-adaptive lasso: Variable selection for causal inference" by Shortreed and Ertefaie (2017). Biometrics, 79(1), 514-520.

Shortreed, S. M., & Ertefaie, A. (2017). Outcome-adaptive lasso: Variable selection for causal inference. *Biometrics*, 73(4), 1111–1122.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301–320.

See Also

`OAL`

Examples

```
## Generate a multivariate normal X matrix
# set information for simulating coviariates
mean_x = 0
sig_x = 1

# pairwise correlation between covariates
rho = 0

# set number of monte carlo (MC) simulation
S=5

# sample size
n = naug = 30

# total number of predictors
p = 10

# note:  pC, pP and pI are number of confounders,
#pure predictors of outcome and pure predictors of exposure, respectively
pC = pP = pI  = 2

# pS number of spurious covariates
pS = p - (pC+pP+pI)
```

```
# list of all p variables
var.list = c(paste("Xc",1:pC,sep=""),
             paste("Xp",1:pP,sep=""),
             paste("Xi",1:pI,sep=""),
             paste("Xs",1:pS,sep=""))

# list of threshold variables
#var.list_Ball = c(paste("X",1:threshold,sep=""))

# set strength of relationship between covariates and outcome
beta_v =  c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )

# Set strength of relationship between covariates and treatment
alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
names(beta_v) = names(alpha_v) = var.list

# set true average treatment effect (taken from Shortreed and Ertefaie (2017))
bA = 0


Data=NULL
### simulate data
Sigma_x = matrix(rho*sig_x^2,nrow=length(var.list),ncol=length(var.list))
diag(Sigma_x) = sig_x^2
Mean_x = rep(mean_x,length(var.list))
Data = as.data.frame(MASS::mvrnorm(n = n,mu=Mean_x,Sigma = Sigma_x,empirical = FALSE))
names(Data) = var.list

X=Data



## Data generation setting
## alpha: Xc's scale is 0.2 0.2 and Xi's scale is 0.3 0.3
## so this refers that there is 2 Xc and Xi
## beta: Xc's scale is 2 2 and Xp's scale is 2 2
## so this refers that there is 2 Xc and Xp
## rest with following setup
Data_fun <- Data_G(X, alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
, beta_v = c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
, bA = 0, sig_x=sig_x, linearY=TRUE,pC=2,pP=2,pI=2)

X=Data_fun$X
A=Data_fun$A
Y=Data_fun$Y
res=GOAL(X,A,Y)
```

OAL *Outcome-Adaptive Lasso (OAL) for High-Dimensional Causal Inference*

Description

Estimates the average treatment effect (ATE) using the Outcome-Adaptive Lasso (OAL)method, which performs variable selection by weighting the penalty based on the importance of covariates in the outcome model.

Usage

```
OAL(X,A,Y)
```

Arguments

X	A numeric matrix or data frame of covariates with n rows (observations) and p columns (variables).
A	A binary treatment vector of length n.
Y	A numeric outcome vector of length n.

Details

This function :

- fits a linear model for the outcome to compute adaptive weights
- applies an adaptive lasso penalty to estimate the propensity score model
- computes inverse probability of treatment weights (IPTW) and ATE estimates across a grid of penalty values (λ)
- selects the optimal λ by minimizing the weighted absolute mean difference (wAMD) across covariates

Value

A numeric vector containing:

OAL	Estimated average treatment effect (ATE) for the optimal lambda.
mOAL	A binary vector indicating selected variables (1 if selected, 0 otherwise).

Author(s)

Rime Naaman

References

Shortreed, S. M., & Ertefaie, A. (2017). Outcome-adaptive lasso: Variable selection for causal inference. *Biometrics*, 73(4), 1111–1122.

See Also

GOAL

Examples

```
## Generate a multivariate normal X matrix
# set information for simulating coviariates
mean_x = 0
sig_x = 1

# pairwise correlation between covariates
rho = 0
```

```
# set number of monte carlo (MC) simulation
S=5

# sample size
n = naug = 30

# total number of predictors
p = 10

# note:  pC, pP and pI are number of confounders,
#pure predictors of outcome and pure predictors of exposure, respectively
pC = pP = pI  = 2

# pS number of spurious covariates
pS = p - (pC+pP+pI)

# list of all p variables
var.list = c(paste(”Xc”,1:pC,sep=””),
             paste(”Xp”,1:pP,sep=””),
             paste(”Xi”,1:pI,sep=””),
             paste(”Xs”,1:pS,sep=””))

# list of threshold variables
#var.list_Ball = c(paste(”X”,1:threshold,sep=””))

# set strength of relationship between covariates and outcome
beta_v =  c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )

# Set strength of relationship between covariates and treatment
alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
names(beta_v) = names(alpha_v) = var.list

# set true average treatment effect (taken from Shortreed and Ertefaie (2017))
bA = 0


Data=NULL
### simulate data
Sigma_x = matrix(rho*sig_x^2,nrow=length(var.list),ncol=length(var.list))
diag(Sigma_x) = sig_x^2
Mean_x = rep(mean_x,length(var.list))
Data = as.data.frame(MASS::mvrnorm(n = n,mu=Mean_x,Sigma = Sigma_x,empirical = FALSE))
names(Data) = var.list
X=Data



## Data generation setting
## alpha: Xc's scale is 0.2 0.2 and Xi's scale is 0.3 0.3
## so this refers that there is 2 Xc and Xi
## beta: Xc's scale is 2 2 and Xp's scale is 2 2
## so this refers that there is 2 Xc and Xp
## rest with following setup
Data_fun <- Data_G(X, alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
, beta_v = c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
, bA = 0, sig_x=sig_x, linearY=TRUE,pC=2,pP=2,pI=2)
```

```
X=Data_fun$X
A=Data_fun$A
Y=Data_fun$Y
res=OAL(X,A,Y)
```

SIS_GOAL *SIS_GOAL: ATE Estimation with SIS and Penalized Logistic Regression*

Description

This function implements the SIS + GOAL procedure, combining Sure Independence Screening (SIS) using causal ball correlation and outcome-adaptive lasso (GOAL) with data augmentation for estimating the Average Treatment Effect (ATE) in high-dimensional settings.

Usage

```
SIS_GOAL(X, A, Y,d.n=30)
```

Arguments

`X`	covariate matrix n x p.
`A`	Binary treatment vector (1 for treated and 0 otherwise.
`Y`	Outcome vector
`d.n`	An optional integer specifying the screening threshold (default is 30

Details

`SIS_GOAL` first applies causal ball correlation screening to select the top `d.n` covariates most associated with the outcome and treatment. Then, for a grid of regularization parameters, it estimates the propensity score using outcome-adaptive lasso with data augmentation (GOAL). The ATE is estimated using inverse probability of treatment weighting (IPTW). The final model is selected based on the configuration that minimizes the weighted absolute mean difference (wAMD).

Value

A vector containing:

`GOAL`	Estimated ATE from the selected model.
`mGOAL`	Binary vector indicating selected variables (1 = selected).

Note

The main difference between `SIS_GOAL` and `GOAL` is that `SIS_GOAL` first performs a variable screening step to select a subset of important variables before applying the GOAL method.

This screening helps handle situations where the number of variables `p` is much larger than the sample size `n` by reducing dimensionality before penalized estimation.

In contrast, `GOAL` applies the penalized estimation directly without preliminary screening.

Author(s)

Rime Naaman

References

Baldé, I., Shortreed, S. M., and Ertefaie, A. (2023). *Outcome-adaptive penalized estimation for causal inference in high dimensions*. JASA.

See Also

This function internally uses `ATE_est`, `Causal.cor`, `create_weights`, `wAMD_function`, `lqa.upd`, `lqa.def`, `expit`

Examples

```
## Generate a multivariate normal X matrix
# set information for simulating coviariates
mean_x = 0
sig_x = 1

# pairwise correlation between covariates
rho = 0

# set number of monte carlo (MC) simulation
S=5

# sample size
n = naug = 300

# total number of predictors
p = 1000
# note:  pC, pP and pI are number of confounders,
#pure predictors of outcome and pure predictors of exposure, respectively
pC = pP = pI  = 2

# pS number of spurious covariates
pS = p - (pC+pP+pI)

# list of all p variables
var.list = c(paste("Xc",1:pC,sep=""),
             paste("Xp",1:pP,sep=""),
             paste("Xi",1:pI,sep=""),
             paste("Xs",1:pS,sep=""))

# list of threshold variables
#var.list_Ball = c(paste("X",1:threshold,sep=""))

# set strength of relationship between covariates and outcome
beta_v =  c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )

# Set strength of relationship between covariates and treatment
alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
names(beta_v) = names(alpha_v) = var.list

# set true average treatment effect (taken from Shortreed and Ertefaie (2017))
bA = 0


Data=NULL
### simulate data
```

```
Sigma_x = matrix(rho*sig_x^2,nrow=length(var.list),ncol=length(var.list))
diag(Sigma_x) = sig_x^2
Mean_x = rep(mean_x,length(var.list))
Data = as.data.frame(MASS::mvrnorm(n = n,mu=Mean_x,Sigma = Sigma_x,empirical = FALSE))
names(Data) = var.list

X=Data



## Data generation setting
## alpha: Xc's scale is 0.2 0.2 and Xi's scale is 0.3 0.3
## so this refers that there is 2 Xc and Xi
## beta: Xc's scale is 2 2 and Xp's scale is 2 2
## so this refers that there is 2 Xc and Xp
## rest with following setup
Data_fun <- Data_G(X, alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
, beta_v = c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
, bA = 0, sig_x=sig_x, linearY=TRUE,pC=2,pP=2,pI=2)

X=Data_fun$X
A=Data_fun$A
Y=Data_fun$Y
res=SIS_GOAL(X,A,Y,d.n=30)
```

SIS_OAL *Sure Independence Screening with Outcome-Adaptive Lasso (SIS-OAL)*

Description

Performs covariate selection and estimation of the Average Treatment Effect (ATE) in high-dimensional data using a two-step procedure: Sure Independence Screening (SIS) based on conditional correlation and Outcome-Adaptive Lasso (OAL) for propensity score estimation. This method selects variables that are predictive of the outcome and adjusts for confounding when estimating treatment effects.

Usage

```
SIS_OAL(X, A, Y, d.n = 30)
```

Arguments

X	A matrix or data frame of covariates (of dimension n x p).
A	A binary treatment indicator vector (0 or 1) of length n.
Y	A continuous or binary outcome variable of length n.
d.n	Number of covariates to retain after the SIS step (default is 30).

Details

This function first ranks covariates based on conditional correlation (using the `Causal.cor` function), then applies the Outcome-Adaptive Lasso to estimate the propensity score using a set of lambda values. The lambda that minimizes the weighted Absolute Mean Difference (wAMD) is selected. The corresponding ATE is returned, along with an indicator vector of selected covariates.

Value

`OAL` Estimated average treatment effect (ATE) corresponding to the lambda value that minimizes the weighted absolute mean difference (wAMD).

`mOAL` A binary vector indicating selected variables in the propensity score model (1 if selected, 0 otherwise).

References

Shortreed, S. M., & Ertefaie, A. (2017). Outcome-adaptive lasso: Variable selection for causal inference. *Biometrics*, 73(4), 1111-1122.

Examples

```
## Generate a multivariate normal X matrix
# set information for simulating coviariates
mean_x = 0
sig_x = 1

# pairwise correlation between covariates
rho = 0

# set number of monte carlo (MC) simulation
S=5

# sample size
n = naug = 300

# total number of predictors
p = 1000

# threshold for screening (taken from Fan and Lv (2008))
#d.n=floor(n/log(n))
#threshold = min(d.n,p)

# set information for data augmentation
#paug=threshold

# note:  pC, pP and pI are number of confounders,
#pure predictors of outcome and pure predictors of exposure, respectively
pC = pP = pI  = 2

# pS number of spurious covariates
pS = p - (pC+pP+pI)

# list of all p variables
var.list = c(paste(”Xc”,1:pC,sep=””),
             paste(”Xp”,1:pP,sep=””),
             paste(”Xi”,1:pI,sep=””),
             paste(”Xs”,1:pS,sep=””))

# list of threshold variables
#var.list_Ball = c(paste(”X”,1:threshold,sep=””))

# set strength of relationship between covariates and outcome
beta_v =  c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
```

```
# Set strength of relationship between covariates and treatment
alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
names(beta_v) = names(alpha_v) = var.list

# set true average treatment effect (taken from Shortreed and Ertefaie (2017))
bA = 0


Data=NULL
### simulate data
Sigma_x = matrix(rho*sig_x^2,nrow=length(var.list),ncol=length(var.list))
diag(Sigma_x) = sig_x^2
Mean_x = rep(mean_x,length(var.list))
Data = as.data.frame(MASS::mvrnorm(n = n,mu=Mean_x,Sigma = Sigma_x,empirical = FALSE))
names(Data) = var.list

X=Data



## Data generation setting
## alpha: Xc's scale is 0.2 0.2 and Xi's scale is 0.3 0.3
## so this refers that there is 2 Xc and Xi
## beta: Xc's scale is 2 2 and Xp's scale is 2 2
## so this refers that there is 2 Xc and Xp
## rest with following setup
Data_fun <- Data_G(X, alpha_v = c( 1, 1, 0, 0, 1, 1,  rep(0,p-6) )
, beta_v = c( 0.6, 0.6, 0.6, 0.6, 0, 0, rep(0,p-6) )
, bA = 0, sig_x=sig_x, linearY=TRUE,pC=2,pP=2,pI=2)

X=Data_fun$X
A=Data_fun$A
Y=Data_fun$Y
res=SIS_OAL(X,A,Y,d.n=30)
```

Index