Package ‘KOBT’

February 20, 2020

Type Package
Title Knockoff Boosted Tree
Version 0.1.0
Description A novel strategy for conducting variable selection without prior model topology knowledge using the knockoff method (Barber and Candes (2015) <doi:10.1214/15-AOS1337>) with extreme boosted tree models (Chen and Guestrin (2016) <doi:10.1145/2939672.2939785>). This method is inspired by the original knockoff method, where the differences between original and knockoff variables are used for variable selection with false discovery rate control. In addition to the original knockoff generating methods, two new sampling methods are available to be implemented, namely the sparse covariance and principal component knockoff methods. As results, the indices of selected variables are returned.

Depends R (>= 3.4.0)
Imports glmnet (>= 2.0-18), knockoff, spcov, xgboost, Rdpack (>= 0.11-0), stats, MASS

R topics documented:

create.pc.knockoff ........................................... 2
generate.knockoff ........................................... 2
importance.score ............................................ 3
kobt.select ................................................... 4
reduce.dim .................................................... 5
create.pc.knockoff

Create PC Knockoffs

Description
Create non-parametric knockoffs based on principal component regression and residuals permutation.

Usage
create.pc.knockoff(X, pc.num)

Arguments
X
An input original design matrix.
pc.num
The number of principal components to be used for generating knockoff matrices.

Value
A principal component knockoff matrix.

Examples
set.seed(10)
X <- matrix(rnorm(100), nrow = 10)
tmp <- create.pc.knockoff(X = X, pc.num = 5)

generate.knockoff

Generate Knockoff Matrix

Description
Generate different types of knockoff matrices given an original one.

Usage
generate.knockoff(X, type, num, num.comp = 10)

Arguments
X
An input original design matrix.
type
The knockoff type to be generated. There are three choices available: (1) "shrink" for the shrink Gaussian knockoff; (2) "sparse" for the sparse Gaussian knockoff; and (3) "pc" for the principal component knockoff.
num
The number of knockoff matrices to be created.
num.comp
The number of principal components to be used for generating knockoff matrices, the default is 10.
importance.score

**Value**

A list of created knockoff matrices.

**References**


**Examples**

```r
set.seed(10)
X <- matrix(rnorm(100), nrow = 10)
Z <- generate.knockoff(X = X, type = "shrink", num = 5)

importance.score(fit, Y, X)
```

<table>
<thead>
<tr>
<th>Importance Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Generate SHAP (SHapley Additive exPlanations) and Saabas scores.</td>
</tr>
</tbody>
</table>

**Usage**

```r
importance.score(fit, Y, X)
```

**Arguments**

- **fit**: A fitted object of class `xgb.Booster`.
- **Y**: A vector of responses.
- **X**: An input design matrix.

**Value**

A list of (1) shap, a vector of Hapley Additive exPlanations for each feature; (2) saabas, a vector of an individualized heuristic feature attribution method, which can be considered as an approximation for shap.

**References**

Examples

```r
set.seed(10)
X <- matrix(rnorm(100), nrow = 10)
Y <- matrix(rnorm(10), nrow = 10)
dtrain <- xgboost::xgb.DMatrix(X, label = Y)
fit.model <- xgboost::xgb.train(data = dtrain, nrounds = 5)
tmp <- importance.score(fit = fit.model, Y = Y, X = X)
```

kobt.select  Knockoff Variable Selection

Description

Use knockoff to conduct variable selection with false discovery rate control.

Usage

```r
kobt.select(score, fdr = 0.1, type = "modified")
```

Arguments

- **score**: An n by 2p matrix of test statistics, which includes test statistics from n samples, p variables (first p columns), and p knockoff variables (last p columns).
- **fdr**: The targeted false discovery rate (FDR), the default value is 0.1.
- **type**: A character showing the type of calculated false discovery rate: (1) modified and (2) usual FDR, the default value is modified.

Value

Indices of selected columns/variables in the n by p original design matrix.

References


Examples

```r
set.seed(1010)
n <- 100
p <- 100
signal.num <- 20
W_left <- matrix(rnorm(n = n*signal.num, mean = 1, sd = 1), nrow = n)
W_right <- matrix(rnorm(n = n*(2*p-signal.num), mean = 0, sd = 1), nrow = n)
W <- cbind(W_left, W_right)
selected.index <- kobt.select(score = W)
```
reduce.dim

Reduce Dimensionality

Description

Reduce the dimensionality (i.e., the column number) of a design matrix to a desired level using Lasso.

Usage

reduce.dim(fit, X, bound)

Arguments

- **fit**: The fitted cross validation object generated by glmnet::cv.glmnet.
- **X**: An input design matrix whose column number is the dimensionality to be reduced.
- **bound**: The targeted number of dimensionality after reducing.

Value

A list of (1) index.X, indices of selected columns in the design matrix; (2) sub.X, indices of selected columns in the design matrix.

Examples

```r
set.seed(10)
X <- matrix(rnorm(100), nrow = 10)
Y <- matrix(rnorm(10), nrow = 10)
set.seed(11)
cvob1 <- glmnet::cv.glmnet(X, Y)
tmp <- reduce.dim(fit = cvob1, X = X, bound = 3)
```
Index

create.pc.knockoff, 2
generate.knockoff, 2
importance.score, 3
kobt.select, 4
reduce.dim, 5