Package ‘SOIL’

October 12, 2022

Type Package

Title Sparsity Oriented Importance Learning

Version 1.1

Date 2017-09-20

Author
Chenglong Ye <yexxx323@umn.edu>, Yi Yang <yi.yang6@mcgill.ca>, Yuhong Yang <yyang@stat.umn.edu>

Maintainer Yi Yang <yi.yang6@mcgill.ca>

Imports stats, glmnet, ncvreg, MASS, parallel, brglm2

Description Sparsity Oriented Importance Learning (SOIL) provides a new variable importance measure for high dimensional linear regression and logistic regression from a sparse penalization perspective, by taking into account the variable selection uncertainty via the use of a sensible model weighting. The package is an implementation of Ye, C., Yang, Y., and Yang, Y. (2017+).

License GPL-2

URL https://github.com/emeryyi/SOIL

Date/Publication 2017-09-20 18:24:46 UTC

NeedsCompilation no

Repository CRAN

R topics documented:

`SOIL` *Sparsity Oriented Importance Learning (SOIL)*

Description

Sparsity Oriented Importance Learning (SOIL) provides a new variable importance measure for high dimensional linear regression and logistic regression from a sparse penalization perspective, by taking into account the variable selection uncertainty via the use of a sensible model weighting. The package is an implementation of Ye, C., Yang, Y., and Yang, Y. (2017+) DOI: <doi:10.1080/01621459.2017.1377080>.

Usage

```
SOIL(x, y, n_train = ceiling(n/2), no_rep = 100,
                n_train_bound = n_train - 2, n_bound = n - 2,
                psi = 1, family = c("gaussian",
                "binomial"), method = c("lasso","union", "customize"),
                candidate_models, weight_type = c("BIC", "AIC",
                "ARM"), prior = TRUE, reduce_bias = FALSE)
```

Arguments

`x` Matrix of predictors.

`y` Response variable.

`n_train` Size of training set when the weight function is `ARM` or `ARM` with `prior=TRUE`. The default value is `n_train=ceiling(n/2)`.

`no_rep` Number of replications when the weight function is `ARM` and `ARM` with `prior=TRUE`. The default value is `no_rep=100`.

`n_train_bound` When computing the weights using `ARM`, the candidate models with the size larger than `n_train_bound` will be dropped. The default value is `n_train-2`.

`n_bound` When computing the weights using `AIC` or `BIC`, the candidate models with the size larger than `n_train_bound` will be dropped. The default value is `n-2`.

`psi` A positive number to control the improvement of the prior weight. The default value is 1.

`family` Choose the family for GLM models. So far `gaussian` and `binomial` are implemented. The default is `gaussian`.

`method` Users can choose `lasso`, `union` or `customize`. If `method=="lasso"`, then the program automatically provides the candidate models as a union of solution paths of Lasso, Adaptive Lasso; If `method=="union"`, then the program automatically provides the candidate models as a union of solution paths of Lasso, Adaptive Lasso, SCAD, and MCP; If `method="customize"`, users must provide their own set of candidate models in the input argument `candidate_models` as a matrix, each row of which is a 0/1 index vector representing whether each variable is included/excluded in the model. For details see Example section. The default option is `method=="lasso"`.

`candidate_models`
: Only available when `method="customize"`. It is a matrix of candidate models, each row of which is a 0/1 index vector representing whether each variable is included/excluded in the model. For details see Example section.

`weight_type`
: Options for computing weights for SOIL measure. Users can choose among `ARM`, `AIC` and `BIC`. The default is `BIC`.

`prior`
: Whether to use prior in the weighting function. The default is `TRUE`.

`reduce_bias`
: If the binomial model is used, occasionally the algorithm might has convergence issue when the problem of so-called complete separation or quasi-complete separation happens. Users can set `reduce_bias=TRUE` to solve the issue. The algorithm will use an adjusted-score approach when ftting the binomial model for computing the weights. This method is developed in Firth, D. (1993). Bias reduction of maximum likelihood estimates. Biometrika 80, 27-38.

Details

See the paper provided in Reference section.

Value

A "SOIL" object is retured. The components are:

`importance`
: SOIL importance values for each variable.

`weight`
: The weight for each candidate model.

`candidate_models_cleaned`
: Cleaned candidate models: the duplicated candidate models are cleaned; When computing SOIL weights using AIC and BIC, the models with more than n-2 variables are removed (n is the number of observaitons); When computing SOIL weights using ARM, the models with more than n_train-2 variables are removed (n_train is the number of training observations).

References

Ye, C., Yang, Y., and Yang, Y. (2017+). "Sparsity Oriented Importance Learning for High-dimensional
Linear Regression". *Journal of the American Statistical Association*. (Accepted) DOI: 10.1080/01621459.2017.1377080

BugReport: https://github.com/emeryyi/SOIL

Examples

```
# REGRESSION CASE

# generate simulation data
n <- 50
p <- 8
beta <- c(3,1.5,0,0,2,0,0,0)
b0 <- 1
x <- matrix(rnorm(n*p,0,1),nrow=n,ncol=p)
```

```
e <- rnorm(n)
y <- x %*% beta + b0 + e


# compute SOIL using ARM with prior
v_ARM <- SOIL(x, y, family = ”gaussian”,
weight_type = ”ARM”, prior = TRUE)

# compute SOIL using BIC
v_BIC <- SOIL(x, y, family = ”gaussian”, weight_type = ”BIC”)

# compute SOIL using AIC
v_AIC <- SOIL(x, y, family = ”gaussian”,
weight_type = ”AIC”, prior = TRUE)


# user supplied candidate models
candidate_models = rbind(c(0,0,0,0,0,0,0,1),
c(0,1,0,0,0,0,0,1), c(0,1,1,1,0,0,0,1),
c(0,1,1,0,0,0,0,1), c(1,1,0,1,1,0,0,0),
c(1,1,0,0,1,0,0,0))

v1_BIC <- SOIL(x, y,
psi=1,
family = ”gaussian”,
method = ”customize”,
candidate_models = candidate_models,
weight_type = ”BIC”, prior = TRUE)

# CLASSIFICATION CASE

# generate simulation data
n = 300
p = 8
b <- c(1,1,1,-3*sqrt(2)/2)
x=matrix(rnorm(n*p, mean=0, sd=1), n, p)
feta=x[, 1:4]%*%b
fprob=exp(feta)/(1+exp(feta))
y=rbinom(n, 1, fprob)


# compute SOIL for model_check using BIC with prior
b_BIC <- SOIL(x, y, family = ”binomial”, weight_type = ”BIC”)

candidate_models =
rbind(c(0,0,0,0,0,0,0,1),
c(0,1,0,0,0,0,0,1),
c(1,1,1,1,0,0,0,0),
c(0,1,1,0,0,0,0,1),
c(1,1,0,1,1,0,0,0),
c(1,1,0,0,1,0,0,0),
c(0,0,0,0,0,0,0,0),
c(1,1,1,1,1,0,0,0))
```

```
# compute SOIL for model_check using AIC
# user supplied candidate models
b_AIC <- SOIL(x, y, family = ”binomial”,
method = ”customize”, candidate_models = candidate_models,
weight_type = ”AIC”)
```

Index