

# Package ‘nmcs’

April 10, 2021

**Title** Nested Model Confidence Set and LogP measure

**Version** 0.1.1

**Author** Yuanyuan Li, Jiming Jiang

**Maintainer** Yuanyuan Li<yynli9696@gmail.com>

**Description** In high-dimensional data analysis, different variables, or models, are usually chosen when applying different model selection procedures; it is not clear which model is the model suitable when selection results are different via different procedures. Alternatively, instead of focusing on a single model, one may suggest a few models as possibilities. Such a group of models naturally form a model set, which may be associated with a model confidence set (MCS). Another type of measure has to do with measuring the error in model selection, such as the LogP measure, which is an estimated logarithm of the probability of selecting a non-optimal model. Through this package, we provide functions to compute these two state-of-the-art measures of uncertainty for shrinkage model selection methods. For more details, see Li, Y., Jiang, J.(2021).

**License** GPL ( $i=3$ )

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1.9001

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

**Imports** ncvreg,  
gamsel,  
glmnet,

## R topics documented:

Generate_Y . . . . .	2
NMCS . . . . .	2
output_NMCS . . . . .	4

---

Generate_Y	<i>Simulate response variable from a glm family</i>
------------	---

---

### Description

This is a generic function that can simulate new observations from a distribution in exponential family.

### Usage

```
Generate_Y(predy, sigmasq = 1, n, family = "gaussian")
```

### Arguments

predy	Values of a linear predictor that can be written as $X\beta$ .
sigmasq	Variance of errors for a Gaussian linear model. Default value is 1.
n	sample size
family	response type. Either a character string representing one of the families: "gaussian", "binomial", "gam", or else a <code>glm()</code> family object. Default is "gaussian".

### Value

Returns a vector of length `n` with elements drawn from a specified family

### Examples

```
set.seed(0)
n=50
p=10
X = matrix(rnorm(n*p),nrow=n,ncol=p)
true_b = c(1:3, rep(0,p-3))
predy = X %*% true_b
#Simulate obs from Gaussian linear model
Generate_Y(predy, n)
```

---

NMCS	<i>Nested Model Confidence Set and LogP measure</i>
------	---

---

### Description

This function allows you to obtain a nested model confidence set and the LogP uncertainty measure for a given shrinkage model selection method.

**Usage**

```

NMCS(
  Y,
  X,
  family = "gaussian",
  B = 200,
  alpha = 0.05,
  delta = 1e-04,
  penalty = "adlasso",
  tune = "bic"
)

```

**Arguments**

Y	response variable.
X	covariates matrix, of dimension $nobs \times nvars$ ; each row is an observation vector.
family	response type. Either a character string representing one of the families: "gaussian", "binomial", "gam", or else a <code>glm()</code> family object. Default is "gaussian".
B	number of bootstrap replicates to perform; Default value is 200.
alpha	Significance level(s). The confidence level of NMCS set is 1-alpha. Default value is 0.05.
delta	A small positive number added inside of LogP when the bootstrap probability of selected model is 1.
penalty	Default value is "adlasso"; user can choose from "adlasso", "lasso", "scad".
tune	method of tuning parameter $\lambda$ . Default method is "bic"; user can choose from "bic", "aic", "cv"(stands for "cross validation").

**Value**

The NMCS method returns an object of class "NMCS". An object of class "NMCS" is a list containing at least the following components:

mcs	a list containing alpha level, and the Bootstrap coverage probability, width, lower bound model, upper bound model of corresponding $(1-\alpha)\%$ confidence set.
hat_prob	the Bootstrap probability for single selected model.
hat_logp	the LogP measure.
hat_M	a list containing all the information about the selected model based on original data.
len	Size of the selected model, which equals the number of non-zero coefficients.
var.order	Entering order of all predictors.
beta	the estimated coefficients of the selected model.
predy	the fitted values by a linear predictor $\eta = X\beta$ .
sigmasq	Mean sum of residual squares.

## Examples

```
n=100
p=10
B=200
X = matrix(rnorm(n*p),nrow=n,ncol=p)
true_b = c(1:3, rep(0,p-3))
predy = X %*% true_b
#Gaussian
Y=Generate_Y(predy,sigmasq = 1,n=n)
alpha=c(0.05,0.1,0.3)
result=NMCS(Y, X, alpha=alpha,B=B)
output_NMCS(result,alpha=alpha)#NMCS result
result$hat_logP#LogP measure
#Binomial
Y=Generate_Y(predy, n=n, family = "binomial")
result=NMCS(Y, X, family="binomial",alpha=alpha, B=B)
output_NMCS(result,alpha=alpha)#NMCS result
result$hat_logP#LogP measure
#GAM
Xn=X
Xn[,2]=-1/3*X[,1]^3+rnorm(n)
predy_n = Xn %*% true_b
Yn=Generate_Y(predy_n, n=n, family = "gam")
result=NMCS(Yn, Xn, family="gam",alpha=alpha, B=B)
output_NMCS(result,alpha=alpha)#NMCS result
result$hat_logP#LogP measure
```

---

output\_NMCS

*Summary of nested model confidence sets*

---

## Description

This is a generic function used to produce result summaries of nested model confidence sets.

## Usage

```
output_NMCS(nmcs.r, alpha, predictors = NULL)
```

## Arguments

nmcs.r	a result of NMCS.
alpha	A sequence of significance levels. Default value is 0.05.
predictors	The indexes of all predictors. Default value is 1:p.

## Value

Returns a list including the predictors indexes of selected model, and a dataframe including all model confidence sets for user-given **alpha** levels.

hat_M	indexes of predictors in the selected model.
MCS.frame	a dataframe containing the information about model confidence sets. <b>CL</b> confidence levels.

**bcp** the bootstrap coverage probabilities.

**width** width of NMCS, which equals the size difference between lbm and ubm.

**lbm** lower bound models.

**ubm** upper bound models.

### Examples

```
set.seed(0)
n=50
p=10
X = matrix(rnorm(n*p),nrow=n,ncol=p)
true_b = c(1:3, rep(0,p-3))
Y = X %*% true_b + rnorm(n)
alpha=c(0.05,0.1)
result=NMCS(Y, X, alpha=alpha)
output_NMCS(result,alpha=alpha)
```

# Index

\*Topic **Confidence**  
NMCS, [2](#)

\*Topic **NMCS**  
output\_NMCS, [4](#)

\*Topic **Nested**  
NMCS, [2](#)

\*Topic **Set**  
NMCS, [2](#)

\*Topic **glm,**  
Generate\_Y, [2](#)

\*Topic **output,**  
output\_NMCS, [4](#)

\*Topic **simulation**  
Generate\_Y, [2](#)

Generate\_Y, [2](#)

NMCS, [2](#)

output\_NMCS, [4](#)