

# Package ‘HCR’

October 26, 2018

**Type** Package

**Title** Causal Discovery from Discrete Data using Hidden Compact Representation

**Version** 0.1.1

**Author** Jie Qiao [aut, cre],  
Ruichu Cai [ths, aut],  
Kun Zhang [ths, aut],  
Zhenjie Zhang [ths, aut],  
Zhifeng Hao [ths, aut]

**Maintainer** Jie Qiao <qiaojie.chn@gmail.com>

**Description** This code provides a method to fit the hidden compact representation model as well as to identify the causal direction on discrete data.

We implement an effective solution to recover the above hidden compact representation under the likelihood framework.

Please see the Causal Discovery from Discrete Data using Hidden Compact Representation from NIPS 2018 by Ruichu Cai, Jie Qiao, Kun Zhang, Zhen-

jie Zhang and Zhifeng Hao (2018) <<https://nips.cc/Conferences/2018/Schedule?showEvent=11274>> for a description of some of our methods.

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**Imports** data.table (>= 1.10.4), methods

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-10-26 14:50:32 UTC

## R topics documented:

HCR . . . . .	2
HCR.fast . . . . .	3
simuXY . . . . .	4

HCR

*Hidden Compact Representation Model***Description**

Causal Discovery from Discrete Data using Hidden Compact Representation.

**Usage**

```
HCR(X, Y, score_type = "bic", is_anm = FALSE, is_cyclic = FALSE,
    verbose = FALSE, max_iteration = 1000, ...)
```

**Arguments**

X	The data of cause.
Y	The data of effect.
score_type	You can choose "bic","aic","aicc","log" as the type of score to fit the HCR model. Default: bic
is_anm	If is_anm=TRUE, it will enable a data preprocessing to adjust for the additive noise model.
is_cyclic	If is_anm=TRUE and is_cyclic=TRUE, it will enable a data preprocessing to adjust the cyclic additive noise model.
verbose	Show the score at each iteration.
max_iteration	The maximum iteration.
...	Other arguments passed on to methods. Not currently used.

**Value**

The fitted HCR model and its score.

**Examples**

```
library(data.table)
set.seed(10)
data=simuXY(sample_size=200)
r1<-HCR(data$X,data$Y)
r2<-HCR(data$Y,data$X)
# The canonical hidden representation
unique(r1$data[,c("X","Yp")])
# The recovery of hidden representation
unique(data.frame(data$X,data$Yp))
```

---

`HCR.fast`*The Fast Version for Fitting Hidden Compact Representation Model*

---

### Description

A fast implementation for fitting the HCR model. This implementation caches all intermediate results to speed up the greedy search. The basic idea is that if there are two categories need to be combined, for instance,  $X=1$  and  $X=2$  mapping to the same  $Y'=1$ , then the change of the score only depend on the frequency of the data where  $X=1$  and  $X=2$ . Therefore, after combination, if the increment of the likelihood is greater than the penalty, then we will admit such combination.

### Usage

```
HCR.fast(X, Y, score_type = "bic", ...)
```

### Arguments

<code>X</code>	The data of cause.
<code>Y</code>	The data of effect.
<code>score_type</code>	You can choose "bic","aic","aicc","log" as the type of score to fit the HCR model. Default: bic
<code>...</code>	Other arguments passed on to methods. Not currently used.

### Value

The fitted HCR model and its score.

### Examples

```
library(data.table)
set.seed(1)
data=simuXY(sample_size=2000)
r1=HCR.fast(data$X,data$Y)
r2=HCR.fast(data$Y,data$X)
# The canonical hidden representation
unique(r1$data[,c("X","Yp")])
# The recovery of hidden representation
unique(data.frame(data$X,data$Yp))
```

---

 simuXY

*Simulate the data of hidden compact representation model.*


---

**Description**

Generate the X->Y pair HCR data

**Usage**

```
simuXY(sample_size = 2000, min_nx = 3, max_nx = 15, min_ny = 3,
        max_ny = 15, type = 0, distribution = "multinomial")
```

**Arguments**

sample_size	Sample size
min_nx	The minimum value of  X  (Default: 3)
max_nx	The maximum value of  X  (Default: 15)
min_ny	The minimum value of  Y  (Default: 3)
max_ny	The maximum value of  Y  (Default: 15)
type	type=0: standard version, type=1:  X = Y , type=2:  Y' = Y , type=3:  X = Y' , type=4:  X = Y' = Y  (Default: type=0)
distribution	The distribution of the cause X. The options are "multinomial", "geom", "hyper", "nbinom", "pois". Default: multinomial

**Value**

return the synthetic data

**Examples**

```
df=simuXY(sample_size=100, type=0)
length(unique(df[,1]))
length(unique(df[,2]))
length(unique(df[,3]))

df=simuXY(sample_size=100, type=1)
length(unique(df[,1]))
length(unique(df[,3]))

df=simuXY(sample_size=100, type=2)
length(unique(df[,2]))
length(unique(df[,3]))

df=simuXY(sample_size=100, type=3)
length(unique(df[,1]))
length(unique(df[,2]))
```

```
df=simuXY(sample_size=100,type=4)
length(unique(df[,1]))
length(unique(df[,2]))
length(unique(df[,3]))
```

# Index

HCR, [2](#)  
HCR.fast, [3](#)  
simuXY, [4](#)