Package ‘BNN’

February 2, 2018

Type Package
Title Bayesian Neural Network for High-Dimensional Nonlinear Variable Selection
Version 1.0.2
Date 2018-02-02
Depends R (>= 3.0.2)
Imports mvtnorm
Description Perform Bayesian variable selection for high-dimensional nonlinear systems and also can be used to test nonlinearity for a general regression problem. The computation can be accelerated using multiple CPUs. You can refer to <doi:10.1080/01621459.2017.1409122> for more detail.
License GPL-2
LazyLoad yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2018-02-02 21:51:53 UTC
RoxygenNote 6.0.1

Author Bochao Jia [ctb, aut, cre, cph],
Faming Liang [ctb],
Robert Gentleman [cph],
Ross Ihaka [cph],
The R Core Team [cph]

Maintainer Bochao Jia <jbcTPY@uflNd>.

R topics documented:

  BNN-package .......................................................... 2
  BNNprior ............................................................... 3
  BNNsel ................................................................. 4
  Topotecan ............................................................. 5

Index 6
Description

Perform Bayesian variable selection for high-dimensional nonlinear systems. The computation can be accelerated using multiple CPUs. The package can also be used to test nonlinearity for a general regression problem.

Details

The Bayesian neural network used in the package is a one-hidden layer feedforward neural network with shortcut connections. The first module of the package is to calculate the prior probabilities assigned to the class of linear models (i.e., those networks with only shortcut connections) and the class of nonlinear models; and the second module is to perform nonlinear variable selection and calculate the posterior probabilities of the classes of linear and nonlinear models.

Author(s)

Bochao Jia, Faming Liang Maintainer: Bochao Jia<jbc409@ufl.edu>

References


Examples

library(BNN)
BNNprior(50, 1, hid_num = 3, lambda=0.025, total_iteration = 10000, popN = 10)
**BNNprior**

**Prior Probability of Bayesian Neural Networks**

### Description
Calculating the prior probability of linear and nonlinear classes of BNN models.

### Usage
```
BNNprior(dimX, dimY, hid_num = 3, lambda = 0.025, total_iteration = 1000000, popN = 20)
```

### Arguments
- `dimX`: Dimension of the input data.
- `dimY`: The dimension of response data. It is restricted to 1 in the current version of the package.
- `hid_num`: Number of hidden units. The default setting is 3.
- `lambda`: The prior probability for each connection of the neural network being selected for the final model. The default setting is 0.025.
- `total_iteration`: Number of total iterations, default of 1000000.
- `popN`: Number of Markov Chains, default of 20.

### Value
- `prob`: Prior probability assigned to the class of linear models.

### Author(s)
Bochao Jia and Faming Liang

### References

### Examples
```
library(BNN)
BNNprior(50, 1, hid_num = 3, lambda = 0.025, total_iteration = 10000, popN = 5)
```
**Description**

Perform variable selection and calculate posterior probabilities for the classes of linear and nonlinear models.

**Usage**

```r
BNNsel(X, Y, train_num, hid_num=3, lambda=0.025, total_iteration=1000000, popN=20, nCPUs=20)
```

**Arguments**

- `X`: a nxp input data matrix.
- `Y`: response vector.
- `train_num`: Number of training samples, should be smaller or equal to n. The default setting is the first 80% rows of the input samples.
- `hid_num`: Number of hidden units. The default setting is 3.
- `lambda`: The prior probability for each connection of the neural network being selected for the final model. The default setting is 0.025.
- `total_iteration`: Number of iterations, recommend to be larger than or equal to 10,000. The default setting is 1000000.
- `popN`: Number of Markov Chains in a parallel run, should be larger than or equal to 3. The default setting is 20.
- `nCPUs`: Number of CPUs to be used in the simulation. The default setting is 20.

**Value**

A list of five elements:

- `net`: Marginal inclusion probability of each connection of the neural network.
- `prob`: Posterior probability of the class of linear models.
- `mar`: Marginal inclusion probability of each input variable, which can be used for variable selection based on a multiple-hypothesis test or the median probability model criterion.
- `fit`: Fitted value for the response vector of training data.
- `pred`: Predicted value for the response vector of testing data.

**Author(s)**

Bochao Jia and Faming Liang
References


Examples

```r
## simulate data ##
library(BNN)
library(mvtnorm)
n <- 200
p <- 50
X <- rmvnorm(n, mean = rep(1,p), sigma = diag(rep(1,p)))
eps <- rnorm(n,0,0.5)
y <- X[,1]*X[,2]+3*sin(X[,3])-2*cos(X[,4])+X[,5]^2+eps

## fit BNN model ##
BNNsel(X,y,hid_num = 3,lamba=0.025,total_iteration = 1000000, popn = 10,nCPUs = 10)
```

Topotecan  Example dataset

Description

A subset drug response data extracted from the cancer cell line encyclopedia (CCLE) database for the drug topotecan.

Usage

data(Topotecan)

Format

- **X** a \( n \times p \) data matrix; \( n=491, p=89 \)
- **Y** response vector.

References


Index

*Topic BNNprior
  BNNprior, 3
*Topic BNNsel
  BNNsel, 4
*Topic datasets
  Topotecan, 5
*Topic package
  BNN-package, 2

BNN-package, 2
BNNprior, 3
BNNsel, 4
Topotecan, 5