Package ‘eive’

February 15, 2013

Type Package

Title An algorithm for reducing errors-in-variable bias in simple linear regression

Version 1.0

Date 2012-11-02

Author Mehmet Hakan Satman (PhD), Erkin Diyarbakirlioglu (PhD)

Maintainer Mehmet Hakan Satman <mhsatman@istanbul.edu.tr>

Depends compiler

Description EIVE performs a compact genetic algorithm search to reduce errors-in-variables bias in linear regression.

License GPL

Repository CRAN

Date/Publication 2012-12-07 22:45:17

NeedsCompilation no

R topics documented:

eive-package .......................................................... 2
cga .......................................................................... 2
cga_generate_chromosome ....................................... 3
eive.cga ................................................................. 4
f ........................................................................... 6
generate.eive.data .................................................... 7

Index 8
Description

This package includes functions for compact genetic algorithms and errors-in-variable estimation. The function 'eive' performs a genetic search to reduce the errors-in-variable bias in ordinary least squares estimator.

Details

Package: eive
Type: Package
Version: 1.0
Date: 2012-11-02
License: GPL

Author(s)

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>
Maintainer: Mehmet Hakan Satman <mhsatman@istanbul.edu.tr>

Description

Function performs a compact genetic algorithm search for a given evaluation function.

Usage

cga(chsize, psize, evalFunc)

Arguments

chsize Number of bits
popsie Number of population. By default it is 20
evalFunc Function to minimize
**cga_generate_chromosome**

**Details**

CGA (Compact genetic algorithms) sample chromosomes using this probability vector. A probability vector contains \([P_1,P_2,...,P_N]\) and the function generates and returns a chromosome \([B_1,B_2,...,B_N]\). The probability of \(B_K\) having the value of 1 is \(P_K\). So, it has more chance to have \([1,1,0,0]\) than \([0,0,0,1]\) when the probability vector is \([0.9,0.9,0.9,0.1]\).

**Value**

Returns the best chromosome with size of chsize.

**Author(s)**

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>

---

**Description**

Generates vector of zeros and ones for a given probability vector.

**Usage**

cga_generate_chromosome(prob_vec)

**Arguments**

- **prob_vec**: Vector of probabilities.

**Details**

This function is not directly called by user. CGA (Compact genetic algorithms) sample chromosomes using this probability vector. A probability vector contains \([P_1,P_2,...,P_N]\) and the function generates and returns a chromosome \([B_1,B_2,...,B_N]\). The probability of \(B_K\) having the value of 1 is \(P_K\). So, it has more chance to have \([1,1,0,0]\) than \([0,0,0,1]\) when the probability vector is \([0.9,0.9,0.9,0.1]\).

**Value**

Returns the generated chromosome for a given probability vector. Return type is vector.

**Author(s)**

Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>
Errors-in-variable estimation in linear regression with compact genetic algorithms

Description
Method performs a genetic search to find dummy variables that used in a two stage linear regression to reduce errors-in-variables bias in linear regression.

Usage
eive.cga(dirtyx, otherx = NULL, y, numdummies = 10, popsize = 20)

Arguments
- dirtyx: Vector of values of independent variable measured with error
- otherx: Matrix of other independent variables.
- y: Vector of values of dependent variable
- numdummies: Number of dummy variables used in algorithm. By default, it is 10.
- popsize: Population size parameter used in CGA. By default it is 20.

Details
Algorithm performs a genetic search to separate mismeasured independent variable into clean and error parts.

Value
- ols: lm object calculated using original values
- eive: lm object calculated using the predicted variable by eive
- proxy: lm object of proxy regression obtained by genetic search.

Author(s)
Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>

Examples
# Creating an artificial data
# Loading required package
require("eive")

# Setting random number generator seed to 12345
# so each time the script runs, same numbers will be generated
set.seed(12345)
# Number of observations is set to 30
n<-3/zero.noslash

# Unobserved X values are drawn from a Normal distribution
# with mean 10 and variance 7
clean.x <- rnorm(n, mean=10, sd=sqrt(7))

# Measurement error values are drawn from a Normal distribution
# with mean 0 and variance 3
delta.x <- rnorm(n, mean=0, sd=sqrt(3))

# Error term of regression. Normally distributed with mean 0 and
# variance 5
e <- rnorm(n, mean=0, sd=sqrt(5))

# Generating Y values using the linear model
# In this model, intercept is 20 and slope is 10.
y<- 20 + 10* clean.x + e

# Generating observed X values by adding measurement errors
# to unobserved X
dirty.x <- clean.x + delta.x

# Performs a genetic search to find dummy variables that
# used in two stage least squares.
# Please un-comment the line below
# result <- eive.cga (dirtyx=dirty.x, y=y, numdummies=10)

# Print the result
# Please un-comment the line below
# print(result)

########################################### OUTPUT #############################################
# $ols
# # Call:
# # lm(formula = y ~ dirtyx)
# # Coefficients:
# # (Intercept)   dirtyx
# # 63.59/zero.noslash   5.533
# #
# # $eive
# # Call:
# # lm(formula = y ~ ols.proxy$fitted.values)
# # Coefficients:
# # (Intercept)   ols.proxy$fitted.values
# # 23.863   9.229
# #
# $proxy
#
# Call:
# lm(formula = dirtyx ~ matrix(best, nrow = n))
#
# Coefficients:
#      (Intercept) matrix(best, nrow = n)1 matrix(best, nrow = n)2
# 12.9321 -0.6252 -1.9923
# matrix(best, nrow = n)3 matrix(best, nrow = n)4 matrix(best, nrow = n)5
# 0.7537 -0.7076 -0.5247
# matrix(best, nrow = n)6 matrix(best, nrow = n)7 matrix(best, nrow = n)8
# -0.9196 -2.0802 -0.9246
# matrix(best, nrow = n)9 matrix(best, nrow = n)10
# -0.6164  1.9694
#
# END OF OUTPUT

---

f

Cost function of cga in errors-in-variables estimation (eive).

Description
This function is not directly called by the user.

Usage
f(x)

Arguments

x  
Candidate dummies required by the cost function of CGA.

Details
This function is not directly called by the user.

Value
Returns a cost value for a candidate solution in CGA.

Author(s)
Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>
generate.eive.data

Generates data for errors-in-variables model

Description
This function generates data using a linear regression model with size of n. Then one of the independent variables is contaminated by adding measurement errors. Another independent variable can be included in model.

Usage
generate.eive.data(n, e.sd, delta.sd, seed = 12345, useotherx = FALSE)

Arguments

- **n**: Number of observations.
- **e.sd**: Standard deviation of error term of regression.
- **delta.sd**: Standard deviation of measurement error.
- **seed**: Random number seed. By default, it is 12345.
- **useotherx**: Boolean variable. If it is TRUE, another variable will be created with no errors. By default, it is FALSE.

Value
Returns a matrix of contaminated variable, other variable (if exists) and independent variable in its columns.

Author(s)
Mehmet Hakan Satman <mhsatman@istanbul.edu.tr> Erkin Diyarbakirlioglu <ediyarbakirlioglu@gmail.com>
Index

cga, 2
cga_generate_chromosome, 3

eive (eive-package), 2
eive-package, 2
eive.cga, 4

f, 6

generate.eive.data, 7