Package 'KoulMde'

May 16, 2016

Title Koul's Minimum Distance Estimation in Linear Regression and Autoregression Model

Version 2.0.1

Description Consider linear regression model and autoregressive model of order q where errors in the linear regression model and innovations in the autoregression model are independent and symmetrically distributed. Hira L. Koul proposed a nonparametric minimum distance estimation method by minimizing L2-type distance between certain weighted residual empirical distribution functions. He also proposed a simpler version of the loss function by using symmetry of the integrating measure in the distance. This package contains three functions: KoulLrMde(), KoulArMde(), and Koul2StageMde(). KoulLrMde() and KoulArMde() provide minimum distance estimators for linear regression model and autoregression model, respectively, where both are based on Koul's method. These two functions take much less time for the computation than those based on parametric minimum distance estimation methods. Koul2StageMde() provides estimators for regression and autoregressive coefficients of linear regression model with autoregressive errors through minimum distant method of two stages.

Depends R (>= 3.2.2)

License GPL-2

LazyData true

NeedsCompilation no

Author Jiwoong Kim [aut, cre]

Maintainer Jiwoong Kim <kimjiwo2@stt.msu.edu>

Repository CRAN

Date/Publication 2016-05-16 15:41:51

R topics documented:

Koul2StageMde *Two-stage minimum distance estimation in linear regression model with autoregressive error.*

Description

Estimates both regression and autoregressive coefficients in the model $Y = X\beta + \epsilon$ where ϵ is autoregressive process of known order q

Usage

```
Koul2StageMde(Y, X, D, RegIntMeasure, AR_Order, ArIntMeasure)
```

Arguments

Y	- Vector of response variables in linear regression model.
X	- Design matrix of explanatory variables in linear regression model.
D	- Weight matrix. Dimension of D should match that of X. "default" uses XA where A=(X'X)^(-1/2).
RegIntMeasure	- Symmetric and σ-finite measure used for estimating β.
AR_Order	- Order of the autoregressive error.
ArIntMeasure	- Symmetric and σ-finite measure used for estimating autoregressive coefficients of the error.

Value

MDE1stage - The list of the first stage minimum distance estimation result. It contains betahat1stage, residual1stage, and rhohat1stage.

- betahat1stage - The first stage minimum distance estimators of regression coefficients.
- residual1stage - Residuals after the first stage minimum distance estimation.
- rhohat1stage - The first stage minimum distance estimators of autoregressive coefficients of the error.

MDE2stage - The list of the second stage minimum distance estimation result. It contains betahat2stage, residual2stage, and rhohat2stage.

- betahat2stage - The second stage minimum distance estimators of regression coefficients.
- residual2stage - Residuals after the second stage minimum distance estimation.
- rhohat2stage - The second stage minimum distance estimators of autoregressive coefficients of the error.

References

[1] Koul, H. L (1985). Minimum distance estimation in linear regression with unknown error distributions. Statist. Probab. Lett., 3 1-8.

[2] Koul, H. L (1986). Minimum distance estimation and goodness-of-fit tests in first-order autoregression. Ann. Statist., 14 1194-1213.

[3] Koul, H. L (2002). Weighted empirical process in nonlinear dynamic models. Springer, Berlin, Vol. 166

See Also

KoulArMde() and KoulLrMde()

Examples

```
####################
n <- 10
p <- 2
X <- matrix(runif(n*p, 0,20), nrow=n, ncol=p)   #### Generate n-by-p design matrix X
beta <- c(-2, 1.5)                          #### Generate true beta = (-2, 1.5)'

q <- 1
rho <- 0.8                               ##### Generate true parameters rho = 0.8
eps <- vector(length=n)
xi <- rnorm(n, 0,1)                             #### Generate innovation from N(0,1)
                                      #### Generate autoregressive process of order q=1
for (i in 1:n){
 tempCol <- rep(0, times=q)
 for (j in 1:q){
   if(i-j<=0){
     tempCol[j] <- 0
   }else{
     tempCol[j] <- eps[i-j]
   }
 }
 eps[i] <- t(tempCol)%*% rho + xi[i]
}

Y <- X%*%beta + eps
####################
D <- "default"                                   #### Use the default weight matrix

Lx <- function(x){return(x)}                     ##### Define Lebesgue measure
MDEResult <- Koul2StageMde(Y,X, "default", Lx, q, Lx)
MDE1stageResult <- MDEResult$MDE1stage
MDE2stageResult <- MDEResult$MDE2stage

beta1 <- MDE1stageResult$betahat1stage
residual1 <- MDE1stageResult$residual1stage
rho1 <- MDE1stageResult$rhohat1stage

beta2 <- MDE2stageResult$betahat2stage
```

```
residual2 <- MDE2stageResult$residual2stage
rho2 <- MDE2stageResult$rhohat2stage
```

`KoulArMde` *Minimum distance estimation in the autoregression model of the known order.*

Description

Estimates the autoressive coefficients in the $X_t = \rho' Z_t + \xi_t$ where Z_t is the vector of q observations at times $t-1, ..., t-q$.

Usage

```
KoulArMde(X, AR_Order, IntMeasure)
```

Arguments

`X`	- Vector of `n` observed values.
`AR_Order`	- Order of the autoregression model.
`IntMeasure`	- Symmetric and σ-finite measure.

Value

rhohat - Minimum distance estimator of ρ.

residual - Residuals after minimum distance estimation.

References

[1] Koul, H. L (1985). Minimum distance estimation in linear regression with unknown error distributions. Statist. Probab. Lett., 3 1-8.

[2] Koul, H. L (1986). Minimum distance estimation and goodness-of-fit tests in first-order autoregression. Ann. Statist., 14 1194-1213.

[3] Koul, H. L (2002). Weighted empirical process in nonlinear dynamic models. Springer, Berlin, Vol. 166

See Also

KoulLrMde() and Koul2StageMde()

Examples

```
##### Generate stationary AR(2) process with 10 observations
n <- 10
q <- 2
rho <- c(-0.2, 0.8)    ##### Generate true parameters rho = (-0.2, 0.8)'
eps <- rnorm(n, 0,1)   ##### Generate innovations from N(0,1)
X <- rep(0, times=n)
for (i in 1:n){
 tempCol <- rep(0, times=q)
 for (j in 1:q){
   if(i-j<=0){
     tempCol[j] <- 0
   }else{
     tempCol[j] <- X[i-j]
   }
 }
X[i] <- t(tempCol)%*% rho + eps[i]
}

Lx <- function(x){return(x)}                  ##### Define Lebesgue measure
MDEResult <- KoulArMde(X, q, Lx)            ##### Use Lebesgue measure for the integration
rhohat <- MDEResult$rhohat                    ##### Obtain minimum distance estimator
resid  <- MDEResult$residual                  ##### Obtain residual

Dx <- function(x){                            ##### Define degenerate measure at 0
       if(x==0){
         return(1)
       }else{
         return(0)
       }
     }
MDEResult <- KoulArMde(X, q, Dx)            ##### Use degenerate measure at 0 for the integration
rhohat <- MDEResult$rhohat                    ##### Obtain minimum distance estimator
resid <- MDEResult$residual                   ##### Obtain residual
```

KoulLrMde *Minimum distance estimation in linear regression model.*

Description

Estimates the regression coefficients in the model $Y = X\beta + \epsilon$.

Usage

```
KoulLrMde(Y, X, D, IntMeasure)
```

Arguments

Y	- Vector of response variables in linear regression model.
X	- Design matrix of explanatory variables in linear regression model.
D	- Weight matrix. Dimension of D should match that of X. "default" uses XA where A=(X'X)^(-1/2).
IntMeasure	- Symmetric and σ-finite measure.

Value

betahat - Minimum distance estimator of β.

residual - Residuals after minimum distance estimation.

References

[1] Koul, H. L (1985). Minimum distance estimation in linear regression with unknown error distributions. Statist. Probab. Lett., 3 1-8.

[2] Koul, H. L (1986). Minimum distance estimation and goodness-of-fit tests in first-order autoregression. Ann. Statist., 14 1194-1213.

[3] Koul, H. L (2002). Weighted empirical process in nonlinear dynamic models. Springer, Berlin, Vol. 166

See Also

KoulArMde() and Koul2StageMde()

Examples

```
####################
n <- 10
p <- 3
X <- matrix(runif(n*p, 0,50), nrow=n, ncol=p)   #### Generate n-by-p design matrix X
beta <- c(-2, 0.3, 1.5)                         #### Generate true beta = (-2, 0.3, 1.5)'
eps <- rnorm(n, 0,1)                            #### Generate errors from N(0,1)
Y <- X%*%beta + eps

D <- "default"                                  #### Use the default weight matrix

Lx <- function(x){return(x)}                    ##### Define Lebesgue measure
MDEResult <- KoulLrMde(Y,X,D, Lx)           ##### Use Lebesgue measure for the integration
betahat <- MDEResult$betahat                    ##### Obtain minimum distance estimator
resid <- MDEResult$residual                     ##### Obtain residual

Dx <- function(x){                              ##### Define degenerate measure at 0
       if(x==0){
         return(1)
       }else{
         return(0)
       }
     }
```

```
MDEResult <- KoulLrMde(Y,X,D, Dx)
betahat <- MDEResult$betahat
resid <- MDEResult$residual
```

```
##### Use degenerate measure at 0 for the integration
   ##### Obtain minimum distance estimator
   ##### Obtain residual
```

Index