Package 'FinReg'

September 11, 2017

Type Package

Title Regularizing Bayesian Predictive Regressions

Version 0.1.0

Author Guanhao Feng, Nicholas G. Polson

Maintainer Jianeng Xu <jianeng@uchicago.edu>

Depends glmnet, spcov, Matrix

Description This package implements a regularization method for VAR and SUR models and provides functions for regularization path plots.

Encoding UTF-8

License GPL-2

LazyData true

RoxygenNote 6.0.1

R topics documented:

`Breg` *Regularize Bayesian predictive regression*

Description

Given regularization parameters, this function uses an iterative method to estimate the VAR(1) model,

$$X_t = \beta X_{t-1} + \epsilon_t$$

where all eigenvalues of β are smaller than one in modulus. The noise vector is assumed to be multivariate normally distributed with mean 0 and covariance Σ.

or SUR model

$$y_i = X^T \beta_i + \epsilon_i$$

where y_i is $N \times 1$, β_i is $K \times 1$ and ϵ_i is $N \times 1$; X^T is the design matrix. i = 1,2, ..., m.

Usage

```
Breg(x,y=NULL,lambda,gamma,alpha=1,type = "var",tol=10^-7,Time=10,r=10^-14,step.size=10^3)
```

Arguments

x	For `type = "var"`, an $N \times K$ matrix containing observations of the VAR(1) model; For `type = "sur"`, an $N \times K$ design matrix.
y	For `type = "var"`, NULL; For `type = "sur"`, an $N \times M$ response matrix.
lambda	Regularization parameter for β estimation. Must be a nonnegative scalar.
gamma	Regularization parameter for Σ estiamtion. Must be a nonnegative scalar. The penalty is defined as $\|\|P * \Sigma\|\|_1$ where P is a $K \times K$ matrix of all 1 and with 0 on the diagonal to ensure the positive-definiteness.
alpha	Elastic-net mixing parameter as in `glmnet`, with $0 \leq \alpha \leq 1$. The coefficient penalty is defined as $$\lambda * \{(1-\alpha)/2\|\|\beta\|\|_2^2 + \alpha\|\|\beta\|\|_1\}.$$ `alpha=1` is the lasso penalty, and `alpha=0` is the ridge penalty.
type	Model type: `"var"`(default), `"sur"`.
tol	Convergence threshold for the iterative estimation. The convergence criteria is defined as $$\|\|\beta^{(i)} - \beta^{(i-1)}\|\|_m \leq tol$$ $$\|\|\Sigma^{(i)} - \Sigma^{(i-1)}\|\|_m \leq tol$$ where 'm' denotes the maximum modulus of all the elements.
Time	Maximum number of iterations.
r	Condition number threshold when check if correlation matrix is singular.
step.size	Parameter used in spcov.

Value

beta	The estimate of coefficients. For `type = "var"`, an $K \times K$ matrix; For `type = "sur"`, a vector of length K*M.
sigma	The estimate of noise variance matrix. For `type = "var"`, an $K \times K$ matrix. For `type = "sur"`, an $M \times M$ matrix.

References

Guanhao Feng and Nicholas G. Polson (2016), Regularizing Bayesian Predictive Regressions. https://arxiv.org/abs/1606.01701

See Also

glmnet, spcov

Examples

```
## simulating data
beta <- matrix(c(0.9,-0.1,-0.1,0.8),2,2)
x <- t(rep(0,2))
for (i in 1:1000)
  x<-rbind(x,t(beta%*%x[i,])+rnorm(2))
x <- x[-1,]

## estimating the model
Breg(x,lambd=0,gamma=0,alpha=1,tol=10^-3,Time=30)
```

---

coef_Breg *Extract VAR(1) and SUR coefficient and residual covariance from glmnet model*

Description

This function extracts the coefficient matrix β and residual covariance matrix Σ of VAR(1) and SUR model from an glmnet object.

Usage

```
coef_Breg(model,lambda,X,Y,type="var")
```

Arguments

model	An glmnet model.
lambda	Regularization parameter used in the estimation of β.
X,Y	Breg input.
type	Model type: "var"(default), "sur".

Value

beta	Coefficient matrix/vector.
recov	Residual covariance matrix.

See Also

coef.glmnet, Breg

cvCom *Compute MSE of predictions in a moving window*

Description

This function is used in `cv.Breg` when `type = "cv"`. Estimation an VAR(1) model in a moving window and make a 1-step ahead prediction. The MSE of prediciton is computed.

Usage

```
cvCom(x,lambda,gamma,alpha,T,width,Time)
```

Arguments

`x`	VAR(1) observations in a selected period.
`width`	Bandwidth of the moving window.
`lambda`	See `Breg`.
`gamma`	See `Breg`.
`alpha`	See `Breg`.
`T`	Number of observations.
`Time`	See `Breg`.

See Also

`Breg`

cv_Breg *Select regularization parameters by cross-validation, AIC and BIC*

Description

This function uses VAR(1) model.

Usage

```
cv_Breg(x,alpha=1,type="cv",width=NULL,lambda,gamma,tol=10^-3,Time=30)
```

Arguments

`x`	See `Breg`.
`type`	`"cv"`(default), `"aic"`,`"bic"`.
`width`	Length of the rolling window in computing 1-step ahead prediction errors when `type = "cv"`.
`lambda`	See `Breg`.
`gamma`	See `Breg`.
`alpha`	See `Breg`.
`tol`	See `Breg`.
`Time`	See `Breg`.

Value

lambda	Selected lambda.
gamma	Selected gamma.
map	MSE/AIC/BIC grid calculated to choose the lambda and gamma from.

See Also

Breg

---

icCom *Compute AIC and BIC of VAR(1) model*

Description

This function calculates the AIC and BIC of estimated VAR(1) model.

Usage

```
icCom(x,type,lambda,gamma,alpha,T,Time)
```

Arguments

x	x matrix as in Breg.
type	"aic", "bic".
lambda	See Breg.
gamma	See Breg.
alpha	See Breg.
T	Number of observations.
Time	See Breg.

See Also

Breg

---

path *Regularizaiton path of Bayesian predictive regression*

Description

This functions computes the Bayesian regularization path for coefficient, prediction, correlation and 1-step ahead impulse, given a sequence of lambda or gamma.

Usage

```
path(x,y=NULL,lambda,gamma,r.type = "var",p.type = "coef",alpha=1,Time=5)
```

Arguments

x	For `r.type = "var"`, an $N \times K$ matrix conataining observations of VAR(1). For `r.type = "sur"` or `r.type = "group"`, an $N \times K$ design matrix.
y	For `r.type = "var"`, NULL; For `r.type = "sur"` or `r.type = "group"`, an $N \times M$ response matrix.
r.type	Model type: `"var"`(default), `"sur"`, `"group"`.
p.type	Path type: `"coef"`(default), `"pred"`, `"corr"`, `"impul"`.
lambda	See `Breg`.
gamma	See `Breg`.
alpha	See `Breg`.
Time	See `Breg`.

Details

Only one of `lambda` and `gamma` can be an increasing sequence and another one is a non-negative scalar (typical 0).

For `p.type = "coef"`, `"pred"`, `lambda` is a sequence.

For `p.type = "corr"`, `"impul"`, `gamma` is a sequence.

Value

path	A vector/matrix/array of regularization result. The first dimension size equals to the length of parameter sequence.

See Also

`Breg`

Examples

```
## simulating data
beta <- matrix(c(0.9,-0.1,-0.1,0.8),2,2)
x <- t(rep(0,2))
for (i in 1:1000)
    x<-rbind(x,t(beta%*%x[i,])+rnorm(2))
x <- x[-1,]

## set the regularization parameter lambda
lambda = seq(0,0.1,by=0.01)

## compute regularization path
path(x=x,gamma=0,lambda=lambda,p.type="pred",r.type="var")
```

Index