Package ‘cap’

October 1, 2018

Type Package

Title Covariate Assisted Principal (CAP) Regression for Covariance Matrix Outcomes

Version 1.0

Date 2018-09-07

Author Yi Zhao <zhaoyi1026@gmail.com>,
Bingkai Wang <bwang51@jhmi.edu>,
Stewart Mostofsky <mostofsky@kennedykrieger.org>,
Brian Caffo <bcaffo@gmail.com>,
Xi Luo <xi.rossi.luo@gmail.com>

Maintainer Yi Zhao <zhaoyi1026@gmail.com>

Description Performs Covariate Assisted Principal (CAP) Regression for covariance matrix outcomes. The method identifies the optimal projection direction which maximizes the log-likelihood function of the log-linear heteroscedastic regression model in the projection space. See Zhao et al. (2018), Covariate Assisted Principal Regression for Covariance Matrix Outcomes, <doi:10.1101/425033> for details.

Depends MASS, multigroup

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-09-30 23:00:03 UTC

R topics documented:

cap-package 2
capReg 2
cap_beta 5
env.example 7

Index **9**

`cap-package` *Covariate Assisted Principal (CAP) Regression for Covariance Matrix Outcomes*

Description

cap package performs Covariate Assisted Principal (CAP) Regression for covariance matrix outcomes. The method identifies the optimal projection direction which maximizes the log-likelihood function of the log-linear heteroscedastic regression model in the projection space.

Author(s)

Yi Zhao, Johns Hopkins University, <zhaoyi1026@gmail.com>

Bingkai Wang, Johns Hopkins University, <bwang51@jhmi.edu>

Stewart Mostofsky, Johns Hopkins University, <mostofsky@kennedykrieger.org>

Brian Caffo, Johns Hopkins University, <bcaffo@gmail.com>

Xi Luo, Brown University, <xi.rossi.luo@gmail.com>

Maintainer: Yi Zhao <zhaoyi1026@gmail.com>

References

Zhao et al. (2018) *Covariate Assisted Principal Regression for Covariance Matrix Outcomes* <doi:10.1101/425033>

`capReg` *Covariate Assisted Principal Regression for Covariance Matrix Outcomes*

Description

This function identifies the first k projection directions that satisfies the log-linear model assumption.

Usage

```
capReg(Y, X, nD = 1, method = c("CAP", "CAP-C"), CAP.OC = FALSE,
  max.itr = 1000, tol = 1e-04, trace = FALSE, score.return = TRUE,
  gamma0.mat = NULL, ninitial = NULL)
```

Arguments

`Y`	a data list of length n. Each list element is a $T \times p$ matrix, the data matrix of T observations from p features.
`X`	a $n \times q$ data matrix, the covariate matrix of n subjects with $q-1$ predictors. The first column is all ones.
`nD`	an integer, the number of directions to be identified. Default is 1.
`method`	a character of optimization method. `method = "CAP"` considers a weighted L2-norm on the γ vector and solve for the optimizer by block coordinated descent; `method = "CAP-C"` assumes the complete common principal component assumption which identifies the common principal component first and then searches for the optimal PC.
`CAP.OC`	a logic variable. Whether the orthogonal constraint is imposed when identifying higher-order PCs. When `method = "CAP-C"`, this is ignored. Default is `FALSE`.
`max.itr`	an integer, the maximum number of iterations.
`tol`	a numeric value of convergence tolerance.
`trace`	a logic variable. Whether the solution path is reported. Default is `FALSE`.
`score.return`	a logic variable. Whether the log-variance in the transformed space is reported. Default is `TRUE`.
`gamma0.mat`	a data matrix, the initial value of γ. Default is `NULL`, and initial value is randomly chosen.
`ninitial`	an integer, the number of different initial value is tested. When it is greater than 1, multiple initial values will be tested, and the one yields the minimum objective function will be reported. Default is `NULL`.

Details

Considering y_{it} are p-dimensional independent and identically distributed random samples from a multivariate normal distribution with mean zero and covariance matrix Σ_i. We assume there exits a p-dimensional vector γ such that $z_{it} := \gamma' y_{it}$ satisfies the multiplicative heteroscedasticity:

$$\log(\mathrm{Var}(z_{it})) = \log(\gamma' \Sigma_i \gamma) = \beta_0 + x_i' \beta_1$$

, where x_i contains explanatory variables of subject i, and β_0 and β_1 are model coefficients.

Parameters γ and $\beta = (\beta_0, \beta_1')'$ are study of interest, and we propose to estimate them by maximizing the likelihood function,

$$\ell(\beta, \gamma) = -\frac{1}{2} \sum_{i=1}^{n} T_i (x_i' \beta) - \frac{1}{2} \sum_{i=1}^{n} \exp(-x_i' \beta) \gamma' S_i \gamma,$$

where $S_i = \sum_{t=1}^{T_i} y_{it} y_{it}'$. To estimate γ, we impose the following constraint

$$\gamma' H \gamma = 1,$$

where H is a positive definite matrix. In this study, we consider the choice that

$$H = \bar{\Sigma}, \quad \bar{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{T_i} S_i.$$

For higher order projecting directions, an orthogonal constraint is imposed as well.

Value

When `method = "CAP"`,

- `gamma` — the estimate of γ vectors, which is a $p \times nD$ matrix.
- `beta` — the estimate of β for each projecting direction, which is a $q \times nD$ matrix, where $q-1$ is the number of explanatory variables.
- `orthogonality` — an ad hoc checking of the orthogonality between γ vectors.
- `DfD` — output of both average (geometric mean) and individual level of "deviation from diagonality".
- `score` — an output when `score.return = TRUE`. A $n \times nD$ matrix of $\log(\hat{\gamma}' S_i \hat{\gamma})$ value.

When `method = "CAP-C"`,

- `gamma` — the estimate of γ vectors, which is a $p \times nD$ matrix.
- `beta` — the estimate of β for each projecting direction, which is a $q \times nD$ matrix, where $q-1$ is the number of explanatory variables.
- `orthogonality` — an ad hoc checking of the orthogonality between γ vectors.
- `PC.idx` — a vector of length `nD`, the order index of identified γ vectors among all the common principal components.
- `aPC.idx` — the order index of all the principal components that satisfy the log-linear model and the eigenvalue condition.
- `minmax` — a logic output, whether the identified γ vectors are estimated from the minmax approach. If `FALSE`, indicating the eigenvalue condition is not satisfied for any principal component.
- `score` — an output when `score.return = TRUE`. A $n \times nD$ matrix of $\log(\hat{\gamma}' S_i \hat{\gamma})$ value.

Author(s)

Yi Zhao, Johns Hopkins University, <zhaoyi1026@gmail.com>

Bingkai Wang, Johns Hopkins University, <bwang51@jhmi.edu>

Stewart Mostofsky, Johns Hopkins University, <mostofsky@kennedykrieger.org>

Brian Caffo, Johns Hopkins University, <bcaffo@gmail.com>

Xi Luo, Brown University, <xi.rossi.luo@gmail.com>

References

Zhao et al. (2018) *Covariate Assisted Principal Regression for Covariance Matrix Outcomes* <doi:10.1101/425033>

Examples

```
#############################################
data(env.example)
X<-get("X",env.example)
Y<-get("Y",env.example)
```

```
# method = "CAP"
# without orthogonal constraint
re1<-capReg(Y,X,nD=2,method=c("CAP"),CAP.OC=FALSE)
# with orthogonal constraint
re2<-capReg(Y,X,nD=2,method=c("CAP"),CAP.OC=TRUE)

# method = "CAP-C"
re3<-capReg(Y,X,nD=2,method=c("CAP-C"))
#############################################
```

`cap_beta` *Inference of model coefficients*

Description

This function performs inference on the model coefficient β.

Usage

```
cap_beta(Y, X, gamma = NULL, beta = NULL, method = c("asmp", "LLR"),
    boot = FALSE, sims = 1000, boot.ci.type = c("bca", "perc"),
    conf.level = 0.95, verbose = TRUE)
```

Arguments

`Y`	a data list of length n. Each list element is a $T \times p$ matrix, the data matrix of T observations from p features.
`X`	a $n \times q$ data matrix, the covariate matrix of n subjects with $q-1$ predictors. The first column is all ones.
`gamma`	a p-dimensional vector, the projecting direction γ. Default is `NULL`. If `gamma = NULL`, an error warning will be returned.
`beta`	a q-dimensional vector, the model coefficient β. Default is `NULL`. If `beta = NULL`, when `boot = FALSE`, β will be estimated using the provided γ.
`method`	a character of inference method. If `method = "asmp"`, the inference is made based on the asymptotic variance; if `method = "LLR"`, the likelihood ratio test is conducted. When `boot = TRUE`, this argument is ignored.
`boot`	a logic variable, whether bootstrap inference is performed.
`sims`	a numeric value, the number of bootstrap iterations will be performed.
`boot.ci.type`	a character of the way of calculating bootstrap confidence interval. If `boot.ci.type = "bca"`, the bias corrected confidence interval is returned; if `boot.ci.type = "perc"`, the percentile confidence interval is returned.
`conf.level`	a numeric value, the designated significance level. Default is 0.95.
`verbose`	a logic variable, whether the bootstrap procedure is printed. Default is `TRUE`.

Details

Considering y_{it} are p-dimensional independent and identically distributed random samples from a multivariate normal distribution with mean zero and covariance matrix Σ_i. We assume there exits a p-dimensional vector γ such that $z_{it} := \gamma' y_{it}$ satisfies the multiplicative heteroscedasticity:

$$\log(\mathrm{Var}(z_{it})) = \log(\gamma' \Sigma_i \gamma) = \beta_0 + x_i' \beta_1,$$

where x_i contains explanatory variables of subject i, and β_0 and β_1 are model coefficients.

The β coefficient is estimated by maximizing the likelihood function. The asymptotic variance is obtained based on maximum likelihood estimator theory.

Value

When `method = "asmp"`, the output is a $q \times 6$ data frame containing the estimate of β coefficient, the asymptotic standard error, the test statistic, the p-value, and the lower and upper bound of the confidence interval.

When `method = "LLR"`, the output is a $q \times 3$ data frame containing the estimate of β coefficient, the test statistic, and the p-value.

When `boot = TRUE`,

- `Inference` — point estimate of the β coefficient, as well as the corresponding standard error, test statistic, p-value, and the lower and upper bound of the confidence interval.
- `beta.boot` — the estimate of the β coefficient in each iteration.

Author(s)

Yi Zhao, Johns Hopkins University, <zhaoyi1026@gmail.com>

Bingkai Wang, Johns Hopkins University, <bwang51@jhmi.edu>

Stewart Mostofsky, Johns Hopkins University, <mostofsky@kennedykrieger.org>

Brian Caffo, Johns Hopkins University, <bcaffo@gmail.com>

Xi Luo, Brown University, <xi.rossi.luo@gmail.com>

References

Zhao et al. (2018) *Covariate Assisted Principal Regression for Covariance Matrix Outcomes* <doi:10.1101/425033>

Examples

```
#############################################
data(env.example)
X<-get("X",env.example)
Y<-get("Y",env.example)
Phi<-get("Phi",env.example)

# asymptotic variance
re1<-cap_beta(Y,X,gamma=Phi[,2],method=c("asmp"),boot=FALSE)
```

```
# likelihood ratio test
re2<-cap_beta(Y,X,gamma=Phi[,2],method=c("LLR"),boot=FALSE)

# bootstrap confidence interval

re3<-cap_beta(Y,X,gamma=Phi[,2],boot=TRUE,sims=500,verbose=FALSE)

#############################################
```

---

env.example *Simulated data*

Description

"env.example" is an R environment containing the data generated from the proposed model with $p = 2$.

Usage

```
data("env.example")
```

Format

An R environment

X a $n \times q$ data matrix, the covariate matrix of n subjects with $q - 1$ predictors. The first column is all ones.

Y a list of length n. Each list element is a $T \times p$ matrix, the data matrix of T observations from p features.

Phi a $p \times p$ matrix, the true projection matrix used to generate the data.

beta a $q \times p$ matrix, the true coefficient matrix used to generate the data.

Sigma a $p \times p \times n$ array, the covariance matrix of the n subjects.

Details

For subject i observation t ($i = 1, \ldots, n$, $t = 1, \ldots, T$), $y_{it} = (y_{it1}, \ldots, y_{itp})$ was generated from a p-dimensional normal distribution with mean zero and covariance Σ, where

$$\Sigma = \Phi\Lambda\Phi,$$

Φ is an orthonormal matrix and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ is a diagonal matrix. The eigenvalues λ_{ij} ($j = 1, \ldots, p$) satisfies the following log-linear model

$$log(\lambda_{ij}) = x_i^\top \beta_j,$$

where β_j is the jth column of beta.

Examples

```
data(env.example)
X<-get("X",env.example)
Y<-get("Y",env.example)
```

Index

∗Topic **datasets**
    env.example, 7
∗Topic **models**
    cap_beta, 5
    capReg, 2
∗Topic **package**
    cap-package, 2

cap *(*cap-package*)*, 2
cap-package, 2
cap_beta, 5
capReg, 2

env.example, 7