Package 'ICTransCFA'

January 4, 2023

Type Package

Title Factor Augmented Transformation Models for Interval-Censored Failure Time Data

Version 1.0.3

Author Hongxi Li, Shuwei Li, Liuquan Sun and Xinyuan Song

Maintainer Hongxi Li <ailihongxi@163.com>

Description This package provides a joint modeling approach for regression analysis of interval-censored failure time data with latent variables. The proposed model comprises a confirmatory factor analysis model to group multiple observed variables into a few latent factors and a class of semiparametric transformation models to examine the effects of the latent factors and other observed covariates on the failure event.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 1.0.9), stats (>= 4.2.2), mvtnorm (>= 1.1-3), fastGHQuad (>= 0.2)

LinkingTo Rcpp, RcppEigen

Depends R (>= 3.5.0)

RoxygenNote 7.2.2

R topics documented:

`ICTransCFA-package` *A statistical package for semiparametric regression analysis of interval-censored failure time data with latent variables*

Description

This package provides a joint modeling approach for regression analysis of interval-censored failure time data with latent variables. The proposed model comprises a confirmatory factor analysis model to group multiple observed variables into a few latent factors and a class of semiparametric transformation models to examine the effects of the latent factors and other observed covariates on the failure event.

Details

Package:	ICTransCFA
Type:	Package
Version:	1.0.3
Date:	2023-01-04

Author(s)

Hongxi Li, Shuwei Li, Liuquan Sun and Xinyuan Song
Maintainers: Hongxi Li <ailihongxi@163.com>

`bootJointTransCFA` *A function used to calculate the standard error of the estimate with the bootstrap method*

Description

A function used to calculate the standard error of the estimate with the bootstrap method. The bootstrap method is time consuming and we recommend using the profile fisher score method if one only needs to estimate the covariance matrix of the estimates of the finite-dimensional parameters. See function `ProfileJointTransCFA()`.

Usage

```
bootJointTransCFA(L, R, CensorIndMat, X, V, LoadingsInitial,
                  r = 0, B = 100, ip = 15, seed = 999)
```

Arguments

L	The left endpoint of the observed interval. Its corresponding entry should be '0' if an observation is left-censored.
R	The right endpoint of the observed interval. Its corresponding entry should be '9999' if an observation is right-censored.
CensorIndMat	The censoring indicator matrix (n times 3). Each component of the first column of the matrix indicates "whether an observation is left-censored (1) or not (0)", each component of the second column of the matrix indicates "whether an observation is interval-censored (1) or not (0)", and each component of the third column of the matrix indicates "whether an observation is right-censored (1) or not (0)".
X	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.
V	The observed covariate matrix (n times p) that are used to form latent risk factors. These covariates should be included in the CFA model.
LoadingsInitial	The initial value of the loading matrix in CFA model, the most important argument in this function. One needs to specify a correct structure to enable the function to accurately identify the complex relationship between the observed variables and their corresponding latent factors. Please see 'examples' in function JointTransCFAreg.
r	A number that is greater than or equal to 0, which is the argument in the transformation function. r = 0 and 1 correspond to the proportional hazards and odds models, respectively. The default value is 0.
B	An integer that is greater than 1, representing the number of resampling in the bootstrap procedure. This parameter will be useful when seeMethod = "bootstap". The default value is 100.
ip	The number of quadrature points used for each latent variable. The default value is 15. Note that the total number of quadrature points is ip^q.
seed	An integer, used to create reproducible results when calculating the standard error of the estimate with the bootstrap method. The default value is 999.

Value

A vector containing the standard error estimates through the bootstrap method.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

JointTransCFAreg, ProfileJointTransCFA

Examples

```
data("data_example")
```

```
L<-data_example[,1]
R<-data_example[,2]

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

# The numbers of the observed variables and the corresponding latent factors
# are 6 and 2, respectively. The LoadingsInitial should be 6*2 matrix.
# The first 4 variables in V correspond to the first latent factor,
# so the first 4 elements in the first column of  LoadingsInitial should be 1, and 0 elsewhere.
# The last 2 variables in V correspond to the second latent factor,
# so the last 2 elements in the second column of  LoadingsInitial should be 1, and 0 elsewhere.

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

LoadingsInitial
#      [,1] [,2]
# [1,]    1    0
# [2,]    1    0
# [3,]    1    0
# [4,]    1    0
# [5,]    0    1
# [6,]    0    1



# fit PH model
fit <- JointTransCFAreg(L, R, CensorIndMat, X, V, LoadingsInitial, seeMethod = "Profile",
                        r = 0, ip = 10)

bootJointTransCFA(L, R, CensorIndMat, X, V, LoadingsInitial, r = 0, B = 10, ip = 10)
```

`data_example` *The simulated dataset.*

Description

A data example (100 observations and 16 variables) can be used in the 'INTransCFA' package.

Usage

```
data("data_example")
```

Format

A data frame with 100 observations on the following 16 variables.

`L` The left endpoint of the observed interval, L = 0 if the observation is left-censored.

`R` The right endpoint of the observed interval, R = 9999 if the observation is right-censored.

`d1` The left-censored indicator, 1 if the observation is left-censored.

d2 The interval-censored indicator, 1 if the observation is interval-censored.

d3 The right-censored indicator, 1 if the observation is right-censored.

X1 An observed variable X1, which is 'NOT' used to form the latent risk factors.

X2 An observed variable X2, which is 'NOT' used to form the latent risk factors.

X3 An observed variable X3, which is 'NOT' used to form the latent risk factors.

X4 An observed variable X4, which is 'NOT' used to form the latent risk factors.

X5 An observed variable X5, which is 'NOT' used to form the latent risk factors.

V1 An observed variable V1, which corresponds to the first latent risk factors.

V2 An observed variable V2, which corresponds to the first latent risk factors.

V3 An observed variable V3, which corresponds to the first latent risk factors.

V4 An observed variable V4, which corresponds to the first latent risk factors.

V5 An observed variable V5, which corresponds to the second latent risk factors.

V6 An observed variable V6, which corresponds to the second latent risk factors.

References

[1] Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

[2] Alzheimer's disease neuroimaging initiative. https://adni.loni.usc.edu/.

Examples

```
data(data_example)
```

---

dGfun *The derivative function of transformation link function G(x)*

Description

Calculating the derivative of the transformation function G(x).

Usage

```
dGfun(u,r)
```

Arguments

u	A number, vector or matrix at which the derivative of transformation function is evaluated.
r	A number that is greater than or equal to 0, which is the argument in the transformation function. r = 0 and 1 correspond to the proportional hazards and odds models, respectively. The default value is 0.

Value

A number, vector or matrix containing the results of the derivative of transformation function evaluated at the values of u.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

Gfun, Ginv

Examples

```
# dG(x)/dx = 1/(1+r*(x)) (r >= 0)

mat = matrix(1:9, 3, 3)

dGfun(mat, 0)

dGfun(mat, 1)
```

Efun *A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm (Estimation step, r > 0)*

Description

A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm. This function applies to all models except the Cox model (r = 0). Moreover, this function is only used in the parameter estimation procedure and the calculation of the conditional expectations in the profile Fisher score procedure will be completed by other functions. See Efunprofile, EfunCox and EfunCoxprofile.

Usage

```
Efun(XX, WW, gam, X, b, h, Psi, d3, IndL, IndR, V, B, r, pi)
```

Arguments

XX	A matrix X of ip^q by q quadrature points, which is calculated by the myinitquad function.
WW	A vector W of length ip^q containing the associated weights, which is calculated by the myinitquad function.
gam	A vector containing the current estimates of the effects of latent variables in the transformation model.
X	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.
b	A vector containing the current estimates of the effects of observed variables in the transformation model.
h	A vector containing the current estimates of the non-negative jump sizes of the cumulative baseline hazard function in the transformation model.

Psi	A matrix containing the current estimate of the covariance matrix of the random error terms in the CFA model.
d3	A vector containing the right censoring indicators, in which each component takes the value of 1 if the observation is right-censored and 0 otherwise.
IndL	An indicator matrix (K_n times n). For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=L_i).
IndR	A K_n times n indicator matrix. For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=R_i).
V	A matrix (n times p) containing the observed variables used to form the latent risk factors in the CFA model.
B	A matrix, which is the current estimate of the loadings matrix in CFA submodel (CFA submodel).
r	A number greater than or equal to 0, which is the argument in the transformation link function. (We separate Cox model from other models, so r here must be greater than 0).
pi	A number, which is pi.

Value

A list contains all the calculated conditional expectations, which will be used in the M-step of the EM algorithm.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

Efunprofile, EfunCox, EfunCoxprofile

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

n=length(L)

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

d3 = CensorIndMat[,3]

s  = dim(X)[2]
p  = dim(LoadingsInitial)[1]
q  = dim(LoadingsInitial)[2]
```

```
if(any(L==0)){order.LR = unique(sort(c(L,R)))[-1]}
if(any(L==9999)){order.LR = order.LR[-length(order.LR)]}

Kn       = length(order.LR)

Ind.L    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(L,Kn),n,Kn))*1
Ind.R    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(R,Kn),n,Kn))*1

b.hat    = matrix(rep(0,s))
gam.hat  = rep(0,q)
lamk.hat = rep(1/Kn,Kn)

phi.hat  = diag(rep(1,q))
psi.hat  = diag(rep(1,p))

grid <- myinitquad(Q = 2, Sigma = phi.hat, ip = 20)

Efun(grid$X, grid$W, gam.hat, X, b.hat, lamk.hat, psi.hat, d3,
     Ind.L, Ind.R, V, LoadingsInitial, r = 1, pi)
```

`EfunCox` *A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm (Estimation step, r = 0)*

Description

A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm. This function is used for only the Cox model. Moreover, this function is only used in the parameter estimation procedure and the calculation of the conditional expectations in the profile Fisher score procedure will be completed by other functions. See `Efunprofile`, `Efun` and `EfunCoxprofile`.

Usage

```
EfunCox(XX, WW, gam, X, b, h, Psi, d3, IndL, IndR, V, B, pi)
```

Arguments

`XX`	A matrix X of ip^q by q quadrature points, which is calculated by the `myinitquad` function.
`WW`	A vector W of length ip^q containing the associated weights, which is calculated by the `myinitquad` function.
`gam`	A vector containing the current estimates of the effects of latent variables in the transformation model.
`X`	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.
`b`	A vector containing the current estimates of the effects of observed variables in the transformation model.

h	A vector containing the current estimates of the non-negative jump sizes of the cumulative baseline hazard function in the transformation model.
Psi	A matrix containing the current estimate of the covariance matrix of the random error terms in the CFA model.
d3	A vector containing the right censoring indicators, in which each component takes the value of 1 if the observation is right-censored and 0 otherwise.
IndL	An indicator matrix (K_n times n). For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=L_i).
IndR	A K_n times n indicator matrix. For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=R_i).
V	A matrix (n times p) containing the observed variables used to form the latent risk factors in the CFA model.
B	A matrix, which is the current estimate of the loadings matrix in CFA submodel (CFA submodel).
pi	A number, which is pi.

Value

A list contains all the calculated conditional expectations, which will be used in the M-step of the EM algorithm.

References

Li, H., Li, S., Sun, L., and Song, X. (20233+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

Efunprofile, Efun, EfunCoxprofile

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

n=length(L)

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

d3 = CensorIndMat[,3]

s  = dim(X)[2]
p  = dim(LoadingsInitial)[1]
q  = dim(LoadingsInitial)[2]
```

```
if(any(L==0)){order.LR = unique(sort(c(L,R)))[-1]}
if(any(L==9999)){order.LR = order.LR[-length(order.LR)]}

Kn       = length(order.LR)

Ind.L    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(L,Kn),n,Kn))*1
Ind.R    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(R,Kn),n,Kn))*1

b.hat    = matrix(rep(0,s))
gam.hat  = rep(0,q)
lamk.hat = rep(1/Kn,Kn)

phi.hat  = diag(rep(1,q))
psi.hat  = diag(rep(1,p))

grid <- myinitquad(Q = 2, Sigma = phi.hat, ip = 20)

EfunCox(grid$X, grid$W, gam.hat, X, b.hat, lamk.hat, psi.hat, d3,
        Ind.L, Ind.R, V, LoadingsInitial, pi)
```

`EfunCoxprofile` *A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm (Profile step, r = 0)*

Description

A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm. This function is used for only the Cox model. Moreover, this function is only used in the profile Fisher score procedure and the calculation of the conditional expectations in the parameter estimation procedure will be completed by other functions. See `Efunprofile`, `EfunCox` and `Efun`.

Usage

```
EfunCoxprofile(XX, WW, gam, X, b, h, Psi, d3, IndL, IndR, V, B, pi)
```

Arguments

`XX`	A matrix X of ip^q by q quadrature points, which is calculated by the `myinitquad` function.
`WW`	A vector W of length ip^q containing the associated weights, which is calculated by the `myinitquad` function.
`gam`	A vector containing the current estimates of the effects of latent variables in the transformation model.
`X`	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.
`b`	A vector containing the current estimates of the effects of observed variables in the transformation model.

h	A vector containing the current estimates of the non-negative jump sizes of the cumulative baseline hazard function in the transformation model.
Psi	A matrix containing the current estimate of the covariance matrix of the random error terms in the CFA model.
d3	A vector containing the right censoring indicators, in which each component takes the value of 1 if the observation is right-censored and 0 otherwise.
IndL	An indicator matrix (K_n times n). For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=L_i).
IndR	A K_n times n indicator matrix. For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=R_i).
V	A matrix (n times p) containing the observed variables used to form the latent risk factors in the CFA model.
B	A matrix, which is the current estimate of the loadings matrix in CFA submodel (CFA submodel).
pi	A number, which is pi.

Value

A list contains all the calculated conditional expectations, which will be used in the M-step of the EM algorithm.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

Efunprofile, EfunCox, Efun

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

n=length(L)

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

d3 = CensorIndMat[,3]

s  = dim(X)[2]
p  = dim(LoadingsInitial)[1]
q  = dim(LoadingsInitial)[2]
```

```
if(any(L==0)){order.LR = unique(sort(c(L,R)))[-1]}
if(any(L==9999)){order.LR = order.LR[-length(order.LR)]}

Kn       = length(order.LR)

Ind.L    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(L,Kn),n,Kn))*1
Ind.R    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(R,Kn),n,Kn))*1

b.hat    = matrix(rep(0,s))
gam.hat  = rep(0,q)
lamk.hat = rep(1/Kn,Kn)

phi.hat  = diag(rep(1,q))
psi.hat  = diag(rep(1,p))

grid <- myinitquad(Q = 2, Sigma = phi.hat, ip = 20)

EfunCoxprofile(grid$X, grid$W, gam.hat, X, b.hat, lamk.hat, psi.hat, d3,
               Ind.L, Ind.R, V, LoadingsInitial, pi)
```

`Efunprofile` *A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm (Profile step, r > 0)*

Description

A function used to calculate the conditional expectations involving multivariate normal random variable in E-step of the EM algorithm. This function applies to all models except the Cox model (r=0). Moreover, this function is only used in the profile Fisher score procedure and the calculation of the conditional expectations in the parameter estimation procedure will be completed by other functions. See `Efun`, `EfunCox` and `EfunCoxprofile`.

Usage

```
Efunprofile(XX, WW, gam, X, b, h, Psi, d3, IndL, IndR, V, B, r, pi)
```

Arguments

`XX`	A matrix X of ip^q by q quadrature points, which is calculated by the `myinitquad` function.
`WW`	A vector W of length ip^q containing the associated weights, which is calculated by the `myinitquad` function.
`gam`	A vector containing the current estimates of the effects of latent variables in the transformation model.
`X`	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.
`b`	A vector containing the current estimates of the effects of observed variables in the transformation model.

h	A vector containing the current estimates of the non-negative jump sizes of the cumulative baseline hazard function in the transformation model.
Psi	A matrix containing the current estimate of the covariance matrix of the random error terms in the CFA model.
d3	A vector containing the right censoring indicators, in which each component takes the value of 1 if the observation is right-censored and 0 otherwise.
IndL	An indicator matrix (K_n times n). For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=L_i).
IndR	A K_n times n indicator matrix. For k = 1, ..., K_n and i = 1, ..., n, the (k,i)th entry of this matrix is I(t_k<=R_i).
V	A matrix (n times p) containing the observed variables used to form the latent risk factors in the CFA model.
B	A matrix, which is the current estimate of the loadings matrix in CFA submodel (CFA submodel).
r	A number greater than or equal to 0, which is the argument in the transformation link function. (We separate Cox model from other models, so r here must be greater than 0).
pi	A number, which is pi.

Value

A list contains all the calculated conditional expectations, which will be used in the M-step of the EM algorithm.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

Efun, EfunCox, EfunCoxprofile

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

n=length(L)

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

d3 = CensorIndMat[,3]
```

```
s  = dim(X)[2]
p  = dim(LoadingsInitial)[1]
q  = dim(LoadingsInitial)[2]


if(any(L==0)){order.LR = unique(sort(c(L,R)))[-1]}
if(any(L==9999)){order.LR = order.LR[-length(order.LR)]}

Kn       = length(order.LR)

Ind.L    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(L,Kn),n,Kn))*1
Ind.R    = (matrix(rep(order.LR,each = n),n,Kn) <= matrix(rep(R,Kn),n,Kn))*1

b.hat    = matrix(rep(0,s))
gam.hat  = rep(0,q)
lamk.hat = rep(1/Kn,Kn)

phi.hat  = diag(rep(1,q))
psi.hat  = diag(rep(1,p))

grid <- myinitquad(Q = 2, Sigma = phi.hat, ip = 20)

Efunprofile(grid$X, grid$W, gam.hat, X, b.hat, lamk.hat, psi.hat, d3,
            Ind.L, Ind.R, V, LoadingsInitial, r = 1, pi)
```

`Gfun` *The transformation link function G(x)*

Description

The derivative function of transformation link function.

Usage

```
Gfun(u,r)
```

Arguments

`u`	A number/vector/matrix that need to be transformed.
`r`	A number greater than or equal to 0, which is the argument in the transformation link function. r = 0 or 1 corresponds to Proportion hazards and odds model, respectively. The default value is 0.

Value

A number/vector/matrix, result after transformation.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

dGfun, Ginv

Examples

```
# G(x) = log(1+r*x)/r (r >= 0)

mat = matrix(1:9, 3, 3)

Gfun(mat, 0)

Gfun(mat, 1)
```

Ginv *Inverse function of the transformation function G(x)*

Description

The inverse function of transformation link function G(x).

Usage

```
Ginv(u, r)
```

Arguments

u	A number/vector/matrix that need to be transformed.
r	A number greater than or equal to 0, which is the argument in the transformation link function. r = 0 or 1 corresponds to Proportion hazards and odds model, respectively. The default value is 0.

Value

The result of calculation with inverse of transformation function G(x).

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

Gfun, dGfun

Examples

```
# G.inverse(x) = (exp(u*r)-1)/r (r >= 0)

mat = matrix(1:9, 3, 3)

dGfun(mat, 0)

dGfun(mat, 1)
```

`JointTransCFAreg` *The main function of this package, which can be used to estimate the joint model involving two submodels: the transformation and CFA models.*

Description

Fitting the joint model to interval-censored data via the EM algorithm proposed by Li et al. (2023+). The first part of the joint model is the CFA model that characterizes the relationship between the observed and latent variables, and the second part of the joint model is a class of transformation models that describes the effects of latent and other observed variables on the failure time of interest.

Usage

```
JointTransCFAreg(L, R, CensorIndMat, X, V, LoadingsInitial, seeMethod = "Profile",
                 r = 0, B = 100, ip = 15, seed = 999)
```

Arguments

`L`	The left endpoint of the observed interval. Its corresponding entry should be '0' if an observation is left-censored.
`R`	The right endpoint of the observed interval. Its corresponding entry should be '9999' if an observation is right-censored.
`CensorIndMat`	The censoring indicator matrix (n times 3). Each component of the first column of the matrix indicates "whether an observation is left-censored (1) or not (0)", each component of the second column of the matrix indicates "whether an observation is interval-censored (1) or not (0)", and each component of the third column of the matrix indicates "whether an observation is right-censored (1) or not (0)".
`X`	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.
`V`	The observed covariate matrix (n times p) that are used to form latent risk factors. These covariates should be included in the CFA model.
`LoadingsInitial`	The initial value of the loading matrix in CFA model, the most important argument in this function. One needs to specify a correct structure to enable the function to accurately identify the complex relationship between the observed variables and their corresponding latent factors. Please see 'examples'.
`seeMethod`	Methods to be used for the standard error estimates of the parameter. Set `seeMethod = "Profile"` will get SEE based on the profile Fisher score approach, and `seeMethod = "bootstap"` corresponds to bootstap estimates. The bootstap approach will consume a lot of time and we recommend using the profile fisher score method.
`r`	A number that is greater than or equal to 0, which is the argument in the transformation function. `r = 0` and 1 correspond to the proportional hazards and odds models, respectively. The default value is 0.
`B`	An integer that is greater than 1, representing the number of resampling in the bootstrap procedure. This parameter will be useful when `seeMethod = "bootstap"`. The default value is 100.

ip	The number of quadrature points used for each latent variable. The default value is 15. Note that the total number of quadrature points is ip^q.
seed	An integer, used to create reproducible results when calculating the standard error of the estimate with the bootstrap method. This parameter will be useful when seeMethod = "bootstap". The default value is 999.

Details

The above function fits the joint model to interval-censored data via the EM algorithm proposed by Li et al. (2023+). Details about the choices of the initial values can be found in Li et al. (2023+). The EM algorithm is declared to achieve convergence when the maximum of the absolute differences of the estimates at two successive iterations is less than 0.001. This function depends on an important R package fastGHQuad, which is used to calculate the conditional expectations with the multivariate Gaussian-Hermite quadrature.

Value

A list containing the estimate of regression coefficients and baseline hazard function in survival submodel, the estimates of loading matrix, variance matrix of the error term and factor correlation matrix in CFA model, and the estimated variance of the above parameters. The list also contains BIC, censoring rate and some other results of the fitted model. See summary() function.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

bootJointTransCFA,ProfileJointTransCFA

Examples

```
data("data_example")
L<-data_example[,1]
R<-data_example[,2]

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

# The numbers of the observed variables and the corresponding latent factors
# are 6 and 2, respectively. The LoadingsInitial should be 6*2 matrix.
# The first 4 variables in V correspond to the first latent factor,
# so the first 4 elements in the first column of  LoadingsInitial should be 1, and 0 elsewhere.
# The last 2 variables in V correspond to the second latent factor,
# so the last 2 elements in the second column of  LoadingsInitial should be 1, and 0 elsewhere.

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

LoadingsInitial
#      [,1] [,2]
# [1,]    1    0
```

```
# [2,]    1    0
# [3,]    1    0
# [4,]    1    0
# [5,]    0    1
# [6,]    0    1

# If one want run this function on other dataset, it should be generated under the same rules.
# For exmple, if we have
LoadingsExample = matrix(c(rep(1, 3), rep(0, 8), rep(1, 2),
                           rep(0, 8), 1,rep(0,8),rep(1,2)), ncol = 4)
LoadingsExample
# LoadingsExample
#      [,1] [,2] [,3] [,4]
# [1,]    1    0    0    0
# [2,]    1    0    0    0
# [3,]    1    0    0    0
# [4,]    0    1    0    0
# [5,]    0    1    0    0
# [6,]    0    0    1    0
# [7,]    0    0    0    1
# [8,]    0    0    0    1
# That means that the dimension of V is 8, and relate to 4 latent factors.
# Specifically, the first 3 columns in V correspond to the first latent factor;
# the 4 and 5th columns corresonding to the second latent factor;
# the 6th columns in V corresponds to the third latent factor;
# and the last 2 columns in V correspond to the fourth latent factor.

# fit PH model
fit <- JointTransCFAreg(L, R, CensorIndMat, X, V, LoadingsInitial, seeMethod = "Profile",
                        r = 0, ip = 10)

fit$beta.hat
fit$gamma.hat
fit$Loadings.hat
```

`myinitquad` *Quadrature points*

Description

Creates a flattened, rotated grid that incorporates correlation through an eigenvalue decomposition of the covariance matrix.

Usage

```
myinitquad(Q, Sigma, ip)
```

Arguments

`Q`	Number of dimensions.
`Sigma`	prior covariance matrix.
`ip`	Number of quadrature points per dimension.

Value

A list with a matrix X of ip^Q by Q quadrature points and a vector W of length ip^Q associated weights.

References

Jaeckel, P. (2005). A note on multivariate Gauss-Hermite quadrature. London: ABN-Amro. Retrieved from http://www.pjaeckel.webspace.virginmedia.com/ANoteOnMultivariateGaussHermiteQuadrature.pdf

Examples

```
sigma <- matrix(c(1,0.2,0.2,1),2,2)
grid <- myinitquad(Q = 2, Sigma = sigma, ip = 10)
```

---

plot.JointTransCFA *Plot the curve of the baseline hazard function*

Description

Plot the curve of the baseline hazard function estimated in joint analysis. `x` must be the `JointTransCFA` object, typically result of function `JointTransCFAreg`.

Usage

```
## S3 method for class JointTransCFA
plot(x, type = "hazard", ...)
```

Arguments

`x`	An object of class `JointTransCFA`, returned by the `JointTransCFAreg` function.
`type`	Character string specifying the type of plot. `Type = "hazard"` means that we plot the baseline cumulative hazard function. `Type = "survival"` means that we plot the baseline survival function. The default value is "hazard".
`...`	Other arguments.

Value

The plot of the estimated baseline hazard function or baseline survival function.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

`JointTransCFAreg`, `summary.JointTransCFA`

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

# fit PH model
fit <- JointTransCFAreg(L, R, CensorIndMat, X, V, LoadingsInitial, seeMethod = "Profile",
                        r = 0, ip = 10)

plot(fit, type = "hazard")
plot(fit, type = "survival")
```

`ProfileJointTransCFA` *A function used to estimate the standard error of unknown parameters (Profile Fisher score method)*

Description

A function used to estimate the standard errors of unknown parameters in the joint model through profile fisher score approach.

Usage

```
ProfileJointTransCFA(L, R, CensorIndMat, X, V, LoadingsInitial,
                     r = 0, theta, lamk, ip = 15)
```

Arguments

`L`	The left endpoint of the observed interval. Its corresponding entry should be '0' if an observation is left-censored.
`R`	The right endpoint of the observed interval. Its corresponding entry should be '9999' if an observation is right-censored.
`CensorIndMat`	The censoring indicator matrix (n times 3). Each component of the first column of the matrix indicates "whether an observation is left-censored (1) or not (0)", each component of the second column of the matrix indicates "whether an observation is interval-censored (1) or not (0)", and each component of the third column of the matrix indicates "whether an observation is right-censored (1) or not (0)".
`X`	The observed covariate matrix (n times s) after excluding the covaraites used to form the latent risk factors. These covariates may have direct effects on the failure time and should be included directly in the transformation model.

V	The observed covariate matrix (n times p) that are used to form latent risk factors. These covariates should be included in the CFA model.
LoadingsInitial	The initial value of the loading matrix in CFA model, the most important argument in this function. One needs to specify a correct structure to enable the function to accurately identify the complex relationship between the observed variables and their corresponding latent factors. Please see 'examples' in function JointTransCFAreg.
r	A number that is greater than or equal to 0, which is the argument in the transformation function. r = 0 and 1 correspond to the proportional hazards and odds models, respectively. The default value is 0.
theta	The initial values of the unknown finite dimension parameters in the joint model, which can be 'theta.hat' in the fitted result of funciton JointTransCFAreg.
lamk	The initial values of the unknown infinite dimension parameters 'Lambda(t)' in the joint model, which can be 'lamk.hat' in the fitted result of funciton JointTransCFAreg.
ip	The number of quadrature points used for each latent variable. The default value is 15. Note that the total number of quadrature points is ip^q.

Value

A vector contains the standard error estimates through profile fisher score method.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

JointTransCFAreg, bootJointTransCFA

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors

# The numbers of the observed variables and the corresponding latent factors
# are 6 and 2, respectively. The LoadingsInitial should be 6*2 matrix.
# The first 4 variables in V correspond to the first latent factor,
# so the first 4 elements in the first column of  LoadingsInitial should be 1, and 0 elsewhere.
# The last 2 variables in V correspond to the second latent factor,
# so the last 2 elements in the second column of  LoadingsInitial should be 1, and 0 elsewhere.

LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

LoadingsInitial
```

```
#      [,1] [,2]
# [1,]    1    0
# [2,]    1    0
# [3,]    1    0
# [4,]    1    0
# [5,]    0    1
# [6,]    0    1


# fit PH model
fit <- JointTransCFAreg(L, R, CensorIndMat, X, V, LoadingsInitial, seeMethod = "Profile",
                        r = 0, ip = 10)

theta.ini = fit$theta.hat
theta.ini[1] = theta.ini[1] - 0.005
lamk.ini = fit$lamk.hat

ProfileJointTransCFA(L, R, CensorIndMat, X, V, LoadingsInitial, r = 0,
                     theta.ini, lamk.ini, ip = 10)
```

`simulataJointTransCFA` *Generate the simulated data*

Description

Generate the simulated data from the joint model given in Li et al (2022+).

Usage

```
simulataJointTransCFA(
  n = 200,
  r = 0,
  beta = matrix(0.5),
  gam = matrix(c(0.5, -0.5)),
  Loadings = array(c(rep(0.8, 3), rep(0, 6), rep(0.8, 3)), dim = c(6, 2)),
  Psi = diag(rep(0.3, 6)),
  Phi = array(c(1, 0.2, 0.2, 1), dim = c(2, 2)),
  Lam = "linear"
)
```

Arguments

`n`	A number denoting the sample size. The default value is 200.
`r`	A number that is greater than or equal to 0, which is the argument in the transformation function. r = 0 and 1 correspond to proportional hazards and odds models, respectively. The default value is 0.
`beta`	A vector, which is the true values of the coefficients of observed variables in the trnasformation model. The default value is 0.5.
`gam`	A vector, which is the true values of the coefficients of latent variables in the trnasformation model. The default value is (0.5, -0.5).
`Loadings`	A matrix, which is the true value of the loadings matrix in CFA submodel. The default value is (0.8, 0.8, 0.8, 0, 0, 0 \ 0, 0, 0, 0.8, 0.8, 0.8)^T.

`Psi`	A matrix, which is the true value of the covariance matrix of the random error terms in the CFA model. The default value is diag(0.3, 0.3, 0.3, 0.3, 0.3, 0.3).
`Phi`	A matrix, which is the true value of the covariance matrix (correlation matrix) of the latent variables in the CFA model. The default value is (1, 0.2 \ 0.2, 1).
`Lam`	The type of the true cumulative baseline hazard function used. `Lam = "linear"` indicates that Lambda(t)=0.5*t, and `Lam = "logarithm"` indicates that Lambda(t)=log(t+1). The default value is `Lam = "linear"`.

Value

A list containing the simulated data.

References

Li, H., Li, S., Sun, L., and Song, X. (2022+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

JointTransCFAreg

Examples

```
n = 200
r = 0
beta = matrix(0.5)

gam = matrix(c(0.5, -0.5))
gam
#      [,1]
# [1,]  0.5
# [2,] -0.5

Loadings = array(c(rep(0.8, 3), rep(0, 6), rep(0.8, 3)), dim = c(6, 2))
Loadings
#      [,1] [,2]
# [1,]  0.8  0.0
# [2,]  0.8  0.0
# [3,]  0.8  0.0
# [4,]  0.0  0.8
# [5,]  0.0  0.8
# [6,]  0.0  0.8


Psi = diag(rep(0.3, 6))
Psi
#      [,1] [,2] [,3] [,4] [,5] [,6]
# [1,]  0.3  0.0  0.0  0.0  0.0  0.0
# [2,]  0.0  0.3  0.0  0.0  0.0  0.0
# [3,]  0.0  0.0  0.3  0.0  0.0  0.0
# [4,]  0.0  0.0  0.0  0.3  0.0  0.0
# [5,]  0.0  0.0  0.0  0.0  0.3  0.0
# [6,]  0.0  0.0  0.0  0.0  0.0  0.3

Phi = array(c(1, 0.2, 0.2, 1), dim = c(2, 2))
```

```
Phi
#      [,1] [,2]
# [1,]  1.0  0.2
# [2,]  0.2  1.0

Lam = "linear"

simulataJointTransCFA(n, r, beta, gam, Loadings, Psi, Phi, Lam)
```

`summary.JointTransCFA` *Summarize the results obtained in the propsoed joint analysis*

Description

Summary the results estimated in joint analysis. `object` must be the `JointTransCFA` object, typically result of function `JointTransCFAreg`.

Usage

```
## S3 method for class JointTransCFA
summary(object, ...)
```

Arguments

`object`	An object of class `JointTransCFA`, returned by the `JointTransCFAreg` function.
`...`	Other arguments.

Value

A vector contains the standard error estimates through profile fisher score method.

References

Li, H., Li, S., Sun, L., and Song, X. (2023+). Factor Augmented Transformation Models for Interval-Censored Failure Time Data. Submitted.

See Also

`JointTransCFAreg`, `plot.JointTransCFA`

Examples

```
data("data_example")

L<-data_example[,1]
R<-data_example[,2]

CensorIndMat<-as.matrix(data_example[,3:5])

X<-as.matrix(data_example[,6:10])

V<-as.matrix(data_example[,11:16]) # The observed covariate matrix used to form the latent factors
```

```
LoadingsInitial = matrix(c(rep(1, 4), rep(0, 6), rep(1,2)), ncol = 2)

# fit PH model
fit <- JointTransCFAreg(L, R, CensorIndMat, X, V, LoadingsInitial, seeMethod = "Profile",
                        r = 0, ip = 10)

summary(fit)
```

`tr` *calculate the trace of a matrix*

Description

calculate the trace of a matrix

Usage

```
tr(A)
```

Arguments

`A`	a matrix

Value

a number

Examples

```
A = matrix(1:9, 3, 3)

tr(A)
```

Index