Package ‘fanc’

November 15, 2022

Type Package
Title Penalized Likelihood Factor Analysis via Nonconvex Penalty
Version 2.3.9
Date 2022-10-20
Depends Matrix, ellipse, tcltk
Description Computes the penalized maximum likelihood estimates of factor loadings and unique variances for various tuning parameters. The pathwise coordinate descent along with EM algorithm is used. This package also includes a new graphical tool which outputs path diagram, goodness-of-fit indices and model selection criteria for each regularization parameter. The user can change the regularization parameter by manipulating scrollbars, which is helpful to find a suitable value of regularization parameter.
License GPL (>= 2)
URL https://doi.org/10.1007/s11222-014-9458-0,
https://doi.org/10.1016/j.csda.2014.05.011,
https://doi.org/10.1007/s41237-016-0007-3,
https://keihirose.com
Repository CRAN
Date/Publication 2022-11-15 00:20:02 UTC
NeedsCompilation yes
Author Kei Hirose [aut, cre] (<https://orcid.org/0000-0001-9827-0356>),
Michio Yamamoto [aut],
Haruhisa Nagata [aut]
Maintainer Kei Hirose <mail@keihirose.com>

R topics documented:

fanc ......................................................... 2
out ......................................................... 5
plot.fanc ................................................... 6
select ....................................................... 7

Index .......................... 9
fanc (penalized maximum likelihood factor analysis via nonconvex penalties)

Description

This package computes the solution path of penalized maximum likelihood estimates via MC+ penalties.

Usage

fanc(x, factors, n.obs, rho, gamma, cor.factor=FALSE, normalize=TRUE, 
normalize.penalty=FALSE, covmat, type="MC", model="FA", control=list())

Arguments

x A data matrix.
factors The number of factors.
cor.factor An indicator of the factor correlation. If "TRUE", the factor correlation is considered. Default is "FALSE".
normalize If "TRUE", each variable is normalized, otherwise it is left alone.
normalize.penalty If "TRUE", the penalty term for each variable has a weight so that the loading matrix is normalized.
rho The values of rho. It can be a scalar or a matrix.
gamma The values of gamma. It must be a vector.
covmat A covariance matrix, which is needed if the data matrix "x" is not available.
n.obs The number of observations, which is needed to calculate the model selection criteria and goodness-of-fit indices when the data matrix "x" is not available.
type Type of penalty. If "MC", the MC penalty is used. If "prenet", the prenet penalty is used. If "enet", the elastic penalty is used. Default is "MC".
model Type of model. "FA", the factor analysis model is used. If "PPCA", the probabilistic principal component analysis is conducted. In the PPCA, the unique variances have the same value. Default is "FA".
control A list of control parameters. See ‘Details’.

Details

The control argument is a list that can supply any of the following components:

length.rho Candidates of tuning parameters which is used for grid search of reparametrization of MC+.
length.gamma A length of tuning parameter which controls sparsenesses. For each rho, gamma=Inf yields soft threshold operator (i.e., lasso penalty) and gamma=+1 produces hard threshold operator.
**max.rho**  Maximum value of rho.
**max.gamma**  A maximum value of gamma (excludes Inf.).
**min.gamma**  A minimum value of gamma.
**eta**  A tuning parameter used for preventing the occurrence of improper solutions. eta must be non-negative.
**ncand.initial**  The number of candidates of initial values of factor loadings.
**ncand.initial.prenet**  The number of candidates of initial values for prenet penalty. Because the prenet penalty is unstable when rho is large, ncand.initial.prenet must be large. Default is 1000.
**maxit.em**  A maximum number of iterations for EM algorithm.
**maxit.cd**  A maximum number of iterations for coordinate descent algorithm.
**maxit.bfgs**  A maximum number of iterations for BFGS algorithm used in the update of factor correlation.
**maxit.initial**  A maximum number of iterations for choosing the initial values.
**start**  Type of start. If "cold", the initial value of factor loadings is randomly chosen for each tuning parameter, which can be slow.
**Delta**  A proportion of maximum value of rho to minimum value of rho, i.e., \( \text{rho.min} = \text{Delta} \times \text{rho.max} \).
**min.uniquevar**  A minimum value of unique variances.
**tol.em**  A positive scalar giving the tolerance at which the parameter in EM is considered close enough to zero to terminate the algorithm.
**tol.cd**  A positive scalar giving the tolerance at which the factor loadings in coordinate descent is considered close enough to zero to terminate the algorithm.
**tol.bfgs**  A positive scalar giving the tolerance at which the factor correlation in BFGS algorithm is considered close enough to zero to terminate the algorithm.
**min.rhozero**  If "TRUE", the minimum value of "rho" is zero.
**zita**  A value of hyper-parameter of factor correlation.
**progress**  If "TRUE", the progress for each tuning parameter is displayed.
**openmp**  If "TRUE", the parallel computation via OpenMP is executed.
**num.threads**  The number of threads of the openmp. Only used when openmp is "TRUE".
**gamma.ebic**  The number of threads of the openmp. Only used when openmp is "TRUE".

**Value**

<table>
<thead>
<tr>
<th>loadings</th>
<th>factor loadings</th>
</tr>
</thead>
<tbody>
<tr>
<td>uniquenesses</td>
<td>unique variances</td>
</tr>
<tr>
<td>Phi</td>
<td>factor correlation</td>
</tr>
<tr>
<td>rho</td>
<td>rho</td>
</tr>
<tr>
<td>AIC</td>
<td>AIC</td>
</tr>
<tr>
<td>BIC</td>
<td>BIC</td>
</tr>
<tr>
<td>CAIC</td>
<td>CAIC</td>
</tr>
</tbody>
</table>
df  

degrees of freedom (number of non-zero parameters for the lasso estimation)

criteria  

values of AIC, BIC and CAIC

goodness.of.fit  

values of GFI and AGFI

gamma  

a value of gamma

Npflag  

If the number of observation is larger than the number of variables, 1, otherwise 0.

factors  

the number of factors

cor.factor  

An indicator of the factor correlation

x  

data matrix

convergence  

indicator of convergence of EM algorithm, coordinate descent and BFGS. If all of these variables are 0, the algorithm has been converged

Author(s)

Kei Hirose

<mail@keihirose.com>

References


See Also

out and plot.fanc objects.

Examples

#generate data
set.seed(0)
loadings0 <- matrix(c(rep(0.8,5),rep(0,5),rep(0,5),rep(0.8,5)),10,2)
common.factors0 <- matrix(rnorm(50*2),50,2)
unique.factors0 <- matrix(rnorm(50*10,sd=sqrt(0.36)),50,10)
x <- common.factors0 %*% t(loadings0) + unique.factors0

#fit data
fit <- fanc(x,2)
fit2 <- fanc(x,2,cor.factor=TRUE) #factor correlation is estimated

#output for fixed tuning parameters
out(fit, rho=0.1, gamma=Inf)

#select a model via model selection criterion
select(fit, criterion="BIC", gamma=Inf)
#plot solution path
plot(fit)

**Description**

This function gives us the loadings from a "fanc" object for fixed value of gamma.

**Usage**

```r
out(x, rho, gamma, scores=FALSE, df.method="active")
```

**Arguments**

- `x`: Fitted "fanc" model object.
- `gamma`: The value of gamma.
- `rho`: The value of rho.
- `scores`: Logical flag for outputting the factor scores. Default is FALSE.
- `df.method`: Two types of degrees of freedom are supported. If "reparameterization", the degrees of freedom of the MC+ are reparameterized based on the degrees of freedom of the lasso. If "active", the degrees of freedom of the lasso are the number of nonzero parameters.

**Value**

- `loadings`: factor loadings
- `uniquenesses`: unique variances
- `Phi`: factor correlation
- `scores`: factor scores
- `df`: degrees of freedom (number of non-zero parameters for the lasso estimation)
- `criteria`: values of AIC, BIC and CAIC
- `goodness.of.fit`: values of GFI and AGFI
- `rho`: a value of rho
- `gamma`: a value of gamma

**Author(s)**

Kei Hirose

<mail@keihirose.com>
References


See Also

fanc and plot.fanc objects.

---

plot.fanc  
plot the solution path from a "fanc" object.

Description

This functions plots the solution paths from a "fanc" object for fixed value of gamma.

Usage

```r
## S3 method for class 'fanc'
plot(x, Window.Height=500, type=NULL, df.method="active", ...)
```

Arguments

- `x`  
  Fitted "fanc" model object.
- `Window.Height`  
  A window height. The default is 500.
- `type`  
  Two plot types are supported. If "path", the path diagram is depicted. If "heatmap", the heatmap is depicted.
- `df.method`  
  Two types of degrees of freedom are supported. If "reparametrization", the degrees of freedom of the MC+ are reparametrized based on the degrees of freedom of the lasso. If "active", the degrees of freedom of are the number of nonzero parameters.
- `...`  
  Other graphical parameters to plot

Value

NULL

Author(s)

Kei Hirose  
<mail@keihirose.com>

References

select

See Also
fanc and out objects.

select  select from a "fanc" object for fixed value of gamma.

Description
This functions give us the loadings from a "fanc" object for fixed value of gamma.

Usage

```r
select(x, criterion=c("BIC","AIC","CAIC","EBIC"),
gamma, scores=FALSE, df.method="active")
```

Arguments

- `x`: Fitted "fanc" model object.
- `criterion`: The criterion by which to select the tuning parameter rho. One of "AIC", "BIC", "CAIC", or "EBIC". Default is "BIC".
- `gamma`: The value of gamma.
- `scores`: Logical flag for outputting the factor scores. Default is FALSE.
- `df.method`: Two types of degrees of freedom are supported. If "active", the degrees of freedom of are the number of nonzero parameters. If "reparametrization", the degrees of freedom of the MC+ are reparametrized based on the degrees of freedom of the lasso.

Value

- `loadings`: factor loadings
- `uniquenesses`: unique variances
- `Phi`: factor correlation
- `scores`: factor scores
- `df`: degrees of freedom (number of non-zero parameters for the lasso estimation)
- `criteria`: values of AIC, BIC and CAIC
- `goodness.of.fit`: values of GFI and AGFI
- `rho`: a value of rho
- `gamma`: a value of gamma

Author(s)
Kei Hirose
<mail@keihirose.com>
References


See Also

fanc and plot.fanc objects.
Index

cbExposeCanvas (plot.fanc), 6
cbExposeLabel (plot.fanc), 6
cbExposeLabelGamma (plot.fanc), 6
cbExposeLabelLambda (plot.fanc), 6
cbValueChanged (plot.fanc), 6
cbValueChangedGamma (plot.fanc), 6
cbValueChangedLabel (plot.fanc), 6
cbValueChangedLabelGamma (plot.fanc), 6
cbValueChangedLabelLambda (plot.fanc), 6
cbValueChangedLabelLambda (plot.fanc), 6

fanc, 2

MakeInterface (plot.fanc), 6

out, 5

plot.fanc, 6
print.fanc (fanc), 2

select, 7