Package ‘fad’

January 24, 2020

Type Package

Title Factor Analysis for Data

Version 0.2-1

Date 2020-01-16

Maintainer Somak Dutta <somakd@iastate.edu>

Description Compute maximum likelihood estimators of parameters in a Gaussian factor model using the matrix-free methodology described in Dai et al. (2019) <doi:10.1080/10618600.2019.1704296>. In contrast to the factanal() function from 'stats' package, fad() can handle high-dimensional datasets where number of variables exceed the sample size and is also substantially faster than the EM algorithms.

License GPL-3

URL https://github.com/somakd/fad

BugReports https://github.com/somakd/fad/issues

Encoding UTF-8

Depends R (>= 3.0.2), methods, RSpectra (>= 0.16-0)

Imports Matrix (>= 1.1-0), Rcpp (>= 0.11.5)

LinkingTo Rcpp

LazyData true

NeedsCompilation yes

Suggests knitr

VignetteBuilder knitr

RoxygenNote 7.0.2

Author Somak Dutta [aut, cre],
Fan Dai [aut],
Ranjan Maitra [ctb]

Repository CRAN

Date/Publication 2020-01-24 18:20:02 UTC
Factor Analysis for data (high or low dimensional).

Description

Perform fast matrix-free maximum-likelihood factor analysis on a covariance matrix or data matrix, works if number of variables is more than number of observations.

Usage

fad(
  x, factors, data = NULL, covmat = NULL, n.obs = NA, subset, na.action, start = NULL, scores = c("none", "regression", "Bartlett"), rotation = "varimax", control = NULL, lower = 0.005,
  ...
)

Arguments

x A formula or a numeric matrix or an object that can be coerced to a numeric matrix.
factors The number of factors to be fitted.
data An optional data frame (or similar: see model.frame), used only if x is a formula. By default the variables are taken from environment(formula).
covmat A covariance matrix, or a covariance list as returned by cov.wt. Of course, correlation matrices are covariance matrices.
n.obs The number of observations, used if covmat is a covariance matrix.
subset A specification of the cases to be used, if x is used as a matrix or formula.
na.action The na.action to be used if x is used as a formula.
start

NULL or a matrix of starting values, each column giving an initial set of uniquenesses.

scores

Type of scores to produce, if any. The default is none, "regression" gives Thompson's scores, "Bartlett" given Bartlett's weighted least-squares scores. Partial matching allows these names to be abbreviated. Also note that some of the scores-types are not applicable when \( p > n \).

rotation

character. "none" or the name of a function to be used to rotate the factors: it will be called with first argument the loadings matrix, and should return a list with component loadings giving the rotated loadings, or just the rotated loadings. The options included in the package are: varimax, promax, quartimax, equamax.

control

A list of control values:

- **nstart** The number of starting values to be tried if `start = NULL`. Default 1.
- **trace** logical. Output tracing information? Default `FALSE`.
- **opt** A list of control values to be passed to `optim`'s control argument.
- **rotate** a list of additional arguments for the rotation function.

lower

The lower bound for uniquenesses during optimization. Should be > 0. Default 0.005.

... Components of `control` can also be supplied as named arguments to `fad`.

Value

An object of class "fad" with components

- **loadings** A matrix of loadings, one column for each factor. The factors are ordered in decreasing order of sums of squares of loadings, and given the sign that will make the sum of the loadings positive. This is of class "loadings"
- **uniquenesses** The uniquenesses computed.
- **criteria** The results of the optimization: the value of the criterion (a linear function of the negative log-likelihood) and information on the iterations used.
- **factors** The argument `factors`.
- **dof** The number of degrees of freedom of the factor analysis model.
- **method** The method: always "mle".
- **rotmat** The rotation matrix if relevant.
- **scores** If requested, a matrix of scores. `napredict` is applied to handle the treatment of values omitted by the `na.action`.
- **n.obs** The number of observations if available, or `NA`.
- **call** The matched call.
- **na.action** If relevant.
- **loglik, BIC** The maximum log-likelihood and the Bayesian Information Criteria.

See Also

- `factanal`
Examples

```r
set.seed(1234)

## Simulate a 200 x 3 loadings matrix ~i.i.d N(0,1)
L <- matrix(rnorm(200*3),200,3)

## Simulate the uniquenesses i.i.d U(0.2,0.9)
D <- runif(200,0.2,0.9)

## Generate a data matrix of size 50 x 200 with rows
## ~i.i.d. N(0,LL^'+diag(D))
X <- tcrossprod(matrix(rnorm(50*3),50,3),L) + matrix(rnorm(50*200),50,200) %*% diag(sqrt(D))

## Fit a factor model with 3 factors:
fit = fad(X,3)

## Print the loadings:
print(fit$loadings)
```

---

**print.fad**

*Print the Output of Factor Analysis*

### Description

Prints the output of the `fad`.

### Usage

```r
## S3 method for class 'fad'
print(x, digits = 3, ...)
```

### Arguments

- **x**: an object of class `fad`.
- **digits**: number of decimal places to use in printing uniquenesses and loadings.
- **...**: further arguments to print.

### Value

None.
Index

cov.wt, 2
factanal, 3
fad, 2
model.frame, 2
optim, 3
print.fad, 4