

# Package ‘GFM’

October 12, 2022

**Type** Package

**Title** Generalized Factor Model

**Version** 1.1.0

**Date** 2021-12-24

**License** GPL-3

**Author** Wei Liu [aut, cre],  
Huazhen Lin [aut],  
Shurong Zheng [aut],  
Jin Liu [aut]

**Maintainer** Wei Liu <weiliu@smail.swufe.edu.cn>

**Description** Generalized factor model for ultra-high dimensional variables with mixed types.

We develop a two-step iterative procedure so that each update can be carried out in parallel across all variables and samples. The fast computation version is provided for ultra-high dimensional data, see examples for more details. More details can be referred to Wei Liu, Huazhen Lin, Shurong Zheng and Jin Liu. (2021) <[doi:10.1080/01621459.2021.1999818](https://doi.org/10.1080/01621459.2021.1999818)>.

**URL** <https://github.com/feiyong/GFM>

**BugReports** <https://github.com/feiyong/GFM/issues>

**Depends** doSNOW, parallel, R (>= 3.5.0)

**Imports** MASS, stats

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-01-05 09:10:02 UTC

**R topics documented:**

Factorm	2
gendata	3
gfm	4
measurefun	6
singleIC	7

<b>Index</b>	<b>9</b>
--------------	----------

---

Factorm	<i>Factor Analysis Model</i>
---------	------------------------------

---

**Description**

Factor analysis to extract latent linear factor and estimate loadings.

**Usage**

```
Factorm(X, q=NULL)
```

**Arguments**

X	a n-by-p matrix, the observed data
q	an integer between 1 and p or NULL, default as NULL and automatically choose q by the eigenvalue ratio method.

**Value**

return a list with class named fac, including following components:

hH	a n-by-q matrix, the extracted latent factor matrix.
hB	a p-by-q matrix, the estimated loading matrix.
q	an integer between 1 and p, the number of factor extracted.
sigma2vec	a p-dimensional vector, the estimated variance for each error term in model.
propvar	a positive number between 0 and 1, the explained propotion of cummulative variance by the q factors.
egvalues	a n-dimensional( $n \leq p$ ) or p-dimensional( $p < n$ ) vector, the eigenvalues of sample covariance matrix.

**Note**

nothing

**Author(s)**

Liu Wei

**References**

Fan, J., Xue, L., and Yao, J. (2017). Sufficient forecasting using factor models. *Journal of Econometrics*.

**See Also**

[gfm](#).

**Examples**

```
dat <- gendata(n = 300, p = 500)
res <- Factorm(dat$X)
measurefun(res$hH, dat$h0) # the smallest canonical correlation
```

---

gendata	<i>Generate simulated data</i>
---------	--------------------------------

---

**Description**

Generate simulated data from high dimensional generalized nonlinear factor model.

**Usage**

```
gendata(seed=1, n=300, p=50, type='homonorm', q=6, rho=1)
```

**Arguments**

seed	a nonnegative integer, the random seed, default as 1.
n	a positive integer, the sample size.
p	an positive integer, the variable dimension.
type	a character, specify the variables type, including type = c('homonorm', 'heternorm', 'pois', 'norm_pois', 'pois_bino').
q	a positive integer, the number of factors.
rho	a positive number, controlling the magnitude of loading matrix.

**Value**

return a list including two components:

X	a n-by-p matrix, the observed data matrix.
H0	a n-by-q matrix, the true latent factor matrix.
B0	a p-by-q matrix, the true loading matrix, the last pzero rows are vectors of zeros.
ind_nz	a integer vector, the index vector for which rows of B0 not zeros.

**Note**

nothing

**Author(s)**

Wei Liu

**References**

Wei Liu, Huazhen Lin, Shurong Zheng & Jin Liu (2019) . Generalized factor model for ultra-high dimensional mixed data. Submitted.

**See Also**

[Factorm](#); [gfm](#).

**Examples**

```
dat <- gendata(n=300, p = 500)
str(dat)
```

---

gfm

*Generalized Factor Model*

---

**Description**

This function is used to conduct the Generalized Factor Model.

**Usage**

```
gfm(X, group, type, q = NULL, parallel = TRUE, para.type =
    "doSNOW", ncores = 10, dropout = 0, dc_eps = 1e-04,
    maxIter = 50, q_set = 1:10, output = TRUE,
    fast_version = FALSE)
```

**Arguments**

X	a matrix with dimension of $n \times p$ ( $p = (p_1 + p_2 + \dots + p_d)$ ), observational mixed data matrix, d is the types of variables, $p_j$ is the dimension of j-th type variable.
group	a vector with length equal to p, specify each column of X belonging to which group.
type	a d-dimensional character vector, specify the type of variables in each group. For example, <code>type=c('poisson', 'binomial')</code> , and it is referred to the help file of <a href="#">glm.fit</a> function for more details.
q	a positive integer or empty, specify the number of factors. If q is NULL, then IC criteria is used to determined $q$ automatically.

parallel	a logical value with TRUE or FALSE, indicates wheter to use prallel computat- ing. Optional parameter with default as TRUE.
para.type	a character specifying the type of parallel including 'doSNOW' and 'parallel'.
ncores	a positive integer, specify the number of cores used for parallel computing.
dropout	a proper subset of $\{1, 2, \dots, d\}$ , specify which group to be dropped in obtaining the initial estimate of factor matrix $H$ , and the aim is to ensure the convergence of algorithm leaded by weak signal variable types. Optional parameter with default as 0, no group dropping.
dc_eps	positive real number, specify the tolerance of varing quantity of objective func- tion in the algorithm. Optional parameter with default as $1e-4$ .
maxIter	a positive integer, specify the times of iteration. Optional parameter with default as 50.
q_set	a positive integer vector, specify the candidates of factor number $q$ , (optional) default as $c(1:10)$ according to Bai,2013.
output	a logical value with TRUE or FALSE, specify whether ouput the mediate infor- mation in iteration process, (optional) default as FALSE.
fast_version	logical value with TRUE or FALSE, $fast\_version = TRUE$ : use the fast algo- rithm which omit the one-step updating, but it cannot ensure the estimation effi- cieny; $fast\_version = FALSE$ : use the original algorithm; (optional) default as FALSE;

### Details

This function also has the MATLAB version at <https://github.com/feiyong/MGFM/blob/master/gfm.m>, which runs faster in MATLAB environment.

### Value

return a list with class name 'gfm' and including following components,

hH	a $n \times q$ matrix, the estimated factor matrix.
hB	a $p \times q$ matrix, the estimated loading matrix.
hmu	a $p$ -dimensional vector, the estimated intercept terms.
obj	a real number, the value of objective function when the convergence achieves.
q	an integer, the used or estimated factor number.
history	a list including the following 7 components: (1)dB: the varied quantity of $B$ in each iteration; (2)dH: the varied quantity of $H$ in each iteration; (3)dc: the varied quantity of the objective function in each iteration; (4)c: the objective value in each iteration; (5)realIter: the real iterations to converge; (6)maxIter: the tolerance of maximum iterations; (7)elapsedTime: the elapsed time.

### Note

nothing

**Author(s)**

Liu Wei

**References**

Liu, W., Lin, H., Zheng, S., & Liu, J. (2021). Generalized factor model for ultra-high dimensional correlated variables with mixed types. *Journal of the American Statistical Association*, (just-accepted), 1-42.

Bai, J. and Liao, Y. (2013). Statistical inferences using large estimated covariances for panel data and factor models.

**See Also**

nothing

**Examples**

```
## mix of normal and Poisson

dat <- gendata(seed=1, n=60, p=60, type='norm_pois', q=2, rho=2)
group <- c(rep(1,ncol(dat$X)/2), rep(2,ncol(dat$X)/2))
type <- c('gaussian','poisson')
## we set maxIter=2 for example.
gfm2 <- gfm(dat$X, group, type, dropout = 2, q=2, output = FALSE, maxIter=2, parallel =FALSE)
measurefun(gfm2$hH, dat$h0, type='ccor')
measurefun(gfm2$hB, dat$b0, type='ccor')
```

---

 measurefun

---

*Assess the performance of an estimator on a matrix*


---

**Description**

Evaluate the smallest cononical correlation (ccor) coefficients or F-norm (fnorm) between two matrices, where a larger ccor is better; a smaller fnorm is better.

**Usage**

```
measurefun(hH, H, type='ccor')
```

**Arguments**

hH	a n-by-q matrix, the estimated matrix.
H	a n-by-q matrix, the true matrix.
type	a character taking value within c('ccor', 'fnorm'), default as 'ccor'.

**Value**

return a real number.

**Note**

nothing

**Author(s)**

Liu Wei

**Examples**

```
dat <- gendata(n = 100, p = 200, q=2, rho=3)
res <- Factorm(dat$X)
measurefun(res$hB, dat$B0)
```

---

singleIC

*IC(PC) criteria for selecting number*

---

**Description**

IC(PC) criteria for selecting number of factors in generalized factor models.

**Usage**

```
singleIC(X, group, type, q_set=1:10, dropout=0, dc_eps=1e-4,
         maxIter=10,output=FALSE, fast_version=TRUE)
```

**Arguments**

- |         |  |
|---------|--|
| X       | a matrix with dimension of $n \times p$ ( $p = (p_1 + p_2 + \dots + p_d)$ ), observational mixed data matrix, $d$ is the types of variables, $p_j$ is the dimension of $j$ -th type variable.  |
| group   | a vector with length equal to $p$ , specify each column of $X$ belonging to which group.   |
| type    | a $d$ -dimensional character vector, specify the type of variables in each group. For example, <code>type=c('poisson', 'binomial')</code> , and it is referred to the help file of <a href="#">glm.fit</a> function for more details.  |
| q_set   | a positive integer vector, specify the candidates of factor number $q$ , (optional) default as <code>c(1:10)</code> according to Bai,2013.   |
| dropout | a proper subset of $\{1, 2, \dots, d\}$ , specify which group to be dropped in obtaining the initial estimate of factor matrix $SH$ , and the aim is to ensure the convergence of algorithm leaded by weak signal variable types. Optional parameter with default as 0, no group dropping. |
| dc_eps  | positive real number, specify the tolerance of varing quantity of objective function in the algorithm. Optional parameter with default as $1e-4$ .   |

maxIter	a positive integer, specify the times of iteration. Optional parameter with default as 50.
output	a logical value with TRUE or FALSE, specify whether output the mediate information in iteration process, (optional) default as FALSE.
fast_version	logical value with TRUE or FALSE, fast_version = TRUE: use the fast algorithm which omit the one-step updating, but it cannot ensure the estimation efficiency; fast_version = FALSE: use the original algorithm; (optional) default as FALSE;

### Details

This function also has the MATLAB version at <https://github.com/feiyong/MGFM/blob/master/singleIC.m>, which runs faster in MATLAB environment.

### Value

return an integer, the estimated number of factors.

### Note

nothing

### Author(s)

Liu Wei

### References

Liu, W., Lin, H., Zheng, S., & Liu, J. (2021). Generalized factor model for ultra-high dimensional correlated variables with mixed types. *Journal of the American Statistical Association*, (just-accepted), 1-42.

Bai, J. and Liao, Y. (2013). Statistical inferences using large estimated covariances for panel data and factor models.

### See Also

nothing

### Examples

```
## Homogeneous normal variables
dat <- gendata(q = 2, n=100, p=100, rho=3)
group <- rep(1,ncol(dat$X))
type <- 'gaussian'
# select q automatically
singleIC(dat$X, group, type, q_set = 1:3, output = FALSE)
```



# Index

- \* **Factor**

- Factor, 2

- gendata, 3

- \* **Feature**

- Factor, 2

- gendata, 3

- \* **GFM**

- gfm, 4

- \* **singleIC**

- singleIC, 7

Factor, 2, 4

gendata, 3

gfm, 3, 4, 4

glm.fit, 4, 7

measurefun, 6

singleIC, 7