

# Package ‘tensorMam’

March 15, 2019

**Type** Package

**Title** tensorMam

**Version** 1.0

**Date** 2018-10-25

**Author** Xu Liu [aut,cre],  
Jian Huang [aut],  
Heng Lian [aut],  
Xiangyong Tan [ctb]

**Maintainer** Xu Liu <liu.xu@sufe.edu.cn>

**Description** A tensor Estimation approach to multivariate additive models. The B-splines are used to approximate unknown function. The number of predictors can be diverged as sample size increases, in which the penalty LASSO, MCP or SCAD can be used.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.11.15), RcppEigen (>= 0.3.2.3.0)

**LinkingTo** Rcpp, RcppEigen

**NeedsCompilation** yes

**Repository** github

**URL** <https://github.com/xliusufe/tensorMam>

**Encoding** UTF-8

## R topics documented:

tensorMam-package . . . . .	2
generateData . . . . .	2
mam . . . . .	3
mam_dr . . . . .	5
mam_sparse . . . . .	6
mam_sparse_dr . . . . .	8
plotfuns . . . . .	10
TransferModalUnfoldings . . . . .	11
<b>Index</b>	<b>12</b>

---

tensorMam-package	<i>A tensor estimation approach to multivariate additive models</i>
-------------------	---

---

### Description

For a high-dimensional multivariate additive model (MAM) using B-splines, with or without aparsity assumptions, treating the coefficients as a third-order tensor and borrowing Tucker decomposition to reduce the number of parameters. The multivariate sparse group lasso (mcp or scad) and the coordinate descent algorithm are used to estimate functions for sparsity situation.

### Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

### Author(s)

Xu Liu

Maintainer: Xu Liu <liu.xu@sufe.edu.cn>

### References

A tensor estimation approach to multivariate additive models.

---

generateData	<i>Generate data from MAM model.</i>
--------------	--------------------------------------

---

### Description

Generate data for a high-dimensional multivariate additive model, with or without aparsity assumptions.

### Usage

```
generateData(n, q, p, s, D2, sigma2=NULL)
```

### Arguments

n	Sample size.
q	The number of responses.
p	The number of covariates.
s	The true covariates associating to response.
D2	The mode of unfolding $D_{(2)}$ .
sigma2	err variance. Default is 0.1.

### Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1, r_2$  and  $r_3$  by BIC or CV.

**Value**

Y	Response, a $n \times q$ -matrix.
X	Design matrix, a $n \times p$ -matrix.
f0	True functions.

**References**

A tensor estimation approach to multivariate additive models.

**See Also**

mam\_sparse

**Examples**

```
# Example 1
D2 <- matrix(runif(30, 0.7, 1), 2, 15)
mydata <- generateData(200, 3, 5, 5, D2)

Y <- mydata$Y
X <- mydata$X

# Example 2
n <- 500
p <- 10
q <- 10
s <- 10
K <- 6
s0 <- s
r10=r20=r30=2
S3 <- matrix(runif(r10*r20*r30,3,7),nrow = r30)
T1 <- matrix(rnorm(s0*r10),nrow = s0)
U1 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(K*r20),nrow = K)
U2 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(q*r30),nrow = q)
U3 <- qr.Q(qr(T1))
D3 <- U3%*%S3%*%t(kronecker(U2,U1))
D2 <- TransferModalUnfoldings(D3,3,2,s0,K,q)
mydata <- generateData(n,q,p,s0,D2)
```

---

mam

---

*Fit MAM without sparsity assumption and with fixed ranks.*


---

**Description**

Fit a low-dimensional multivariate additive model using B-splines, without aparsity assumptions, and given ranks given ranks  $r_1, r_2, r_3$ .

**Usage**

```
mam(y, x, K = 7, r1 = NULL, r2 = NULL, r3 = NULL, SABC = NULL, degr = 3,
     eps = 1e-4, max_step = 20)
```

## Arguments

<code>y</code>	A $n \times q$ numeric matrix of responses.
<code>x</code>	A $n \times p$ numeric design matrix for the model.
<code>K</code>	The number of B-spline base function, that is the plus of both degrees of base function and the number of knots. Default is 6.
<code>degr</code>	the number of knots of B-spline base function. Default is 3.
<code>r1</code>	The first dimension of single value matrix of the tensor. Default is 2.
<code>r2</code>	The second dimension of single value matrix of the tensor. Default is 2.
<code>r3</code>	The third dimension of single value matrix of the tensor. Default is 2.
<code>SABC</code>	A user-specified list of initial coefficient matrix of $S, A, B, C$ . By default, initial matrices are provided by random.
<code>eps</code>	Convergence threshold. The algorithm iterates until the relative change in any coefficient is less than <code>eps</code> . Default is $1e-4$ .
<code>max_step</code>	Maximum number of iterations. Default is 20.

## Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1, r_2$  and  $r_3$  by BIC or CV.

## Value

<code>Dnew</code>	Estimator of $D_{(3)}$ .
<code>rss</code>	Residual sum of squares (RSS).
<code>Y</code>	Response $Y$ .
<code>X</code>	Design matrix $X$ .
<code>Z</code>	Design matrix of Bspline approximation.

## References

A tensor estimation approach to multivariate additive models.

## See Also

`mam_sparse`

## Examples

```
D2 <- matrix(runif(50, 0.7, 1), 2, 25)
mydata <- generateData(200, 5, 5, 5, D2)

fit <- mam(mydata$Y, mydata$X)
Coeff <- fit$Dnew
```

---

mam_dr	<i>Fit MAM without sparsity assumption, and with ranks selected by BIC or CV.</i>
--------	---

---

### Description

Fit a low-dimensional multivariate additive model using B-splines, without sparsity assumptions, and with ranks  $r_1, r_2, r_3$  selected by BIC or CV.

### Usage

```
mam_dr(y, x, method = "BIC", ncv = 10, K_index = NULL, r1_index = NULL,
        r2_index = NULL, r3_index = NULL, SABC = NULL, degr = 3, eps = 1e-4,
        max_step = 20)
```

### Arguments

y	A $n \times q$ numeric matrix of responses.
x	A $n \times p$ numeric design matrix for the model.
method	The method to be applied to select parameters. Either "BIC" (the default), or "CV".
ncv	The number of cross-validation folds. Default is 10. If method is "BIC", ncv is useless.
K_index	A user-specified sequence of K values, where K is the number of B-spline base function. Default is k_index=6.
r1_index	A user-specified sequence of $r_1$ values, where $r_1$ is the first dimension of single value matrix of the tensor. Default is r1_index= $1, \dots, \min(\log(n), p)$ .
r2_index	A user-specified sequence of $r_2$ values, where $r_2$ is the second dimension of single value matrix of the tensor. Default is r2_index= $1, \dots, \max K_i index$ .
r3_index	A user-specified sequence of $r_3$ values, where $r_3$ is the third dimension of single value matrix of the tensor. Default is r3_index= $1, \dots, \min(\log(n), q)$ .
SABC	A user-specified list of initial coefficient matrix of $S, A, B, C$ , which is a list with values $S, A, B, C$ . By default, initial matrices are provided by random.
degr	the number of knots of B-spline base function. Default is 3.
eps	Convergence threshold. The algorithm iterates until the relative change in any coefficient is less than eps. Default is 1e-4.
max_step	Maximum number of iterations. Default is 20.

### Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1, r_2$  and  $r_3$  by BIC or CV.

**Value**

Dnew	Estimator of $D_{(3)}$ .
rss	Residual sum of squares (RSS).
rk_opt	The optimal parametres that slected by BIC or CV.
selected	Which $\lambda$ is selection.
Y	Response $Y$ .
X	Design matrix $X$ .
Z	Design matrix of Bspline approximation.

**References**

"A tensor estimation approach to multivariate additive models".

**See Also**

mam and mam\_sparse\_dr

**Examples**

```
D2 <- matrix(runif(50, 0.7, 1), 2, 25)
mydata <- generateData(200, 5, 5, 5, D2)

fit <- mam_dr(mydata$Y, mydata$X)
opt <- fit$rk_opt
```

---

mam\_sparse

---

*Fit MAM with sparsity assumption and fixed ranks.*


---

**Description**

Fit a high-dimensional multivariate additive model using B-splines, with or without aparsity assumptions, and given ranks given ranks  $r_1, r_2, r_3$ . The multivariate sparse group lasso (mcp or scad) and the coordinate descent algorithm are used to estimate functions for sparsity situation.

**Usage**

```
mam_sparse(y, x, K = 6, r1 = NULL, r2 = NULL, r3 = NULL, penalty = "LASSO",
           lambda = NULL, SABC = NULL, degr = 3, nlam = 20, lam_min = 1e-3,
           eps1 = 1e-4, maxstep1 = 20, eps2 = 1e-4, maxstep2 = 20, gamma = 2,
           dfmax = NULL, alpha = 1)
```

**Arguments**

y	A $n \times q$ numeric matrix of responses.
x	A $n \times q$ numeric design matrix for the model.
K	The number of B-spline base function, that is the plus of both degrees of base functioin and the number of knots. Default is 7.
degr	The number of knots of B-spline base function. Default is 3.

r1	The first dimension of single value matrix of the tensor. Default is 2.
r2	The second dimension of single value matrix of the tensor. Default is 2.
r3	The third dimension of single value matrix of the tensor. Default is 2.
penalty	The penalty to be applied to the model. Either "LASSO" (the default), "SCAD", or "MCP".
lambda	A user-specified sequence of lambda values. By default, a sequence of values of length nlam is computed, equally spaced on the log scale.
SABC	A user-specified list of initial coefficient matrix of $S$ , $A$ , $B$ , $C$ . By default, initial matrices are provided by random.
nlam	The number of lambda values. Default is 20.
lam_min	The smallest value for lambda, as a fraction of lambda.max. Default is 1e-3.
eps1	Convergence threshold. The algorithm iterates until the relative change in any coefficient is less than eps1. Default is 1e-4.
maxstep1	Maximum number of iterations. Default is 20.
eps2	Convergence threshold. The Coordinate descent method algorithm iterates until the relative change in any coefficient is less than eps2. Default is 1e-4.
maxstep2	The maximum iterates number of coordinate descent method. Default is 20.
gamma	The tuning parameter of the MCP/SCAD penalty (see details).
dfmax	Upper bound for the number of nonzero coefficients. Default is no upper bound. However, for large data sets, computational burden may be heavy for models with a large number of nonzero coefficients.
alpha	Tuning parameter for the Mnet estimator which controls the relative contributions from the LASSO, MCP/SCAD penalty and the ridge, or L2 penalty. alpha=1 is equivalent to LASSO, MCP/SCAD penalty, while alpha=0 would be equivalent to ridge regression. However, alpha=0 is not supported; alpha may be arbitrarily small, but not exactly 0.

### Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1$ ,  $r_2$  and  $r_3$  by BIC or CV.

### Value

betapath	Solution path of $\beta$ .
rss	Residual sum of squares (RSS).
df	Degrees of freedom.
lambda	The sequence of regularization parameter values in the path.
lambda_opt	The value of lambda with the minimum BIC value.
selectedID	The index of lambda corresponding to lambda_opt.
activeA	The active set.
Dnew	Estimator of $D_{(3)}$ .
Y	Response $Y$ .
X	Design matrix $X$ .
Z	Design matrix of Bspline approximation $\lambda$ .

## References

A tensor estimation approach to multivariate additive models.

## See Also

mam, mam\_sparse\_dr

## Examples

```
D2 <- matrix(runif(50, 0.7, 1), 2, 25)
mydata <- generateData(200, 5, 10, 5, D2)

fit <- mam_sparse(mydata$Y, mydata$X)
Coeff <- fit$Dnew
```

---

mam\_sparse\_dr

*Fit MAM with sparsity assumption and ranks selected by BIC or CV.*

---

## Description

Fit a high-dimensional multivariate additive model using B-splines, with or with aparsity assumptions and ranks selected by BIC or CV. The multivariate sparse group lasso (mcp or scad) and the coordinate descent algorithm are used to estimate functions for sparsity situation. The tuning parameter is selected by BIC or CV, which matches the method of rank selection.

## Usage

```
mam_sparse_dr(y, x, method = "BIC", ncv = 10, penalty = "LASSO", K_index = NULL,
  r1_index = NULL, r2_index = NULL, r3_index = NULL, lambda = NULL,
  SABC = NULL, nlam = 50, degr = 3, lam_min = 0.01,
  eps1 = 1e-4, maxstep1 = 20, eps2 = 1e-4, maxstep2 = 20, gamma = 2,
  dfmax = NULL, alpha = 1)
```

## Arguments

y	A $n \times q$ numeric matrix of responses.
x	A $n \times q$ numeric design matrix for the model.
method	The method to be applied to select parameters. Either "BIC" (the default), or "CV".
ncv	The number of cross-validation folds. Default is 10. If method is "BIC", ncv is useless.
penalty	The penalty to be applied to the model. Either "LASSO" (the default), "SCAD", or "MCP".
K_index	A user-specified sequence of K values, where K is the number of B-spline base function. Default is k_index=6.
r1_index	A user-specified sequence of $r_1$ values, where $r_1$ is the first dimension of single value matrix of the tensor. Default is r1_index= $1, \dots, \min(\lceil \log(n) \rceil, p)$ .
r2_index	A user-specified sequence of $r_2$ values, where $r_2$ is the second dimension of single value matrix of the tensor. Default is r2_index= $1, \dots, \max K_i \text{ index}$ .



r3_index	A user-specified sequence of $r_3$ values, where $r_3$ is the third dimension of single value matrix of the tensor. Default is $r3\_index = 1, \dots, \min(\lceil \log(n) \rceil, q)$ .
lambda	A user-specified sequence of lambda values. By default, a sequence of values of length nlam is computed, equally spaced on the log scale.
SABC	A user-specified list of initial coefficient matrix of $S, A, B, C$ . By default, initial matrices are provided by random.
nlam	The number of lambda values. Default is 50.
degr	The number of knots of B-spline base function. Default is $degr = 3$ .
lam_min	The smallest value for lambda, as a fraction of lambda.max. Default is 1e-2.
eps1	Convergence threshold. The algorithm iterates until the relative change in any coefficient is less than eps1. Default is 1e-4.
maxstep1	Maximum number of iterations. Default is 20.
eps2	Convergence threshold. The Coordinate descent method algorithm iterates until the relative change in any coefficient is less than eps2. Default is 1e-4.
maxstep2	The maximum iterates number of coordinate descent method. Default is 20.
gamma	The tuning parameter of the MCP/SCAD penalty (see details).
dfmax	Upper bound for the number of nonzero coefficients. Default is no upper bound. However, for large data sets, computational burden may be heavy for models with a large number of nonzero coefficients.
alpha	Tuning parameter for the Mnet estimator which controls the relative contributions from the LASSO, MCP/SCAD penalty and the ridge, or L2 penalty. $\alpha=1$ is equivalent to LASSO, MCP/SCAD penalty, while $\alpha=0$ would be equivalent to ridge regression. However, $\alpha=0$ is not supported; alpha may be arbitrarily small, but not exactly 0.

## Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1, r_2$  and  $r_3$  by BIC or CV.

## Value

Dnew	Estimator of $D_{(3)}$ .
rss	Residual sum of squares (RSS).
df	Degrees of freedom.
activeA	The active set.
lambda	The sequence of regularization parameter values in the path.
selectedID	The index of lambda corresponding to lambda_opt.
lambda_opt	The value of lambda with the minimum BIC or CV value.
RSS	The values of BIC or CV, which is a vector.
rk_opt	The optimal parameters that selected by BIC or CV.
Y	Response $Y$ .
X	Design matrix $X$ .
Z	Design matrix of Bspline approximation.

## References

A tensor estimation approach to multivariate additive models.

## See Also

mam\_dr, mam\_sparse

## Examples

```
D2 <- matrix(runif(50, 0.7, 1), 2, 25)
mydata <- generateData(200, 5, 10, 5, D2)

fit <- mam_sparse_dr(mydata$Y, mydata$X)
opt <- fit$rk_opt
```

---

plotfuns

*Plot the estimated curves from \codetensorMam.*

---

## Description

Plot the curves fit by mam, mam\_dr, mam\_sparse, and mam\_sparse\_dr

## Usage

```
plotfuns(fit, funTrueID)
```

## Arguments

fit                    ??????mam or mam\_sparse object.  
funTrueID            Which function be plotted.

## Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1, r_2$  and  $r_3$  by BIC or CV.

## References

A tensor estimation approach to multivariate additive models.

## See Also

mam

## Examples

```
## Not run:
D2 <- matrix(runif(50, 0.7, 1), 2, 25)
mydata <- generateData(200, 5, 5, 5, D2)
fit <- mam(mydata$Y, mydata$X)
plotfuns(fit, c(1,1))

## End(Not run)
```

---

TransferModalUnfoldings

*Transfer a tensor's modal unfoldings to another.*


---

### Description

Transfer a tensor's modal unfoldings to another.

### Usage

```
tensorTransModeUnfold(T, d1, d2 , r1, r2, r3)
```

### Arguments

T	A mode-d1-unfolding of a tensor with size $r_1 \times r_2 \times r_3$ , input unfolding
d1	An integer, the mode of unfolding $T_{(d_1)}$
d2	An integer, the mode of output unfolding $T_{(d_2)}$
r1	The first dimension of tensor
r2	The second dimension of tensor
r3	The third dimension of tensor

### Details

This function transfers an input mode-d1-unfolding  $T_{(d_1)}$  to mode-d2-unfolding  $T_{(d_2)}$

### Value

D the output mode-d2-unfolding,  $T_{(d_2)}$

### References

A tensor estimation approach to multivariate additive models.

### Examples

```
T <- matrix(1:24,nrow = 4)
TransferModalUnfoldings(T,1,2,4,3,2)
```

# Index

\*Topic **High-dimensional, Sparse models; Tensor estimation; Tucker decomposition.**

tensorMam-package, [2](#)

TransferModalUnfoldings, [11](#)

\*Topic **High-dimensional; Sparse models; Tensor estimation; Tucker decomposition.**

generateData, [2](#)

mam, [3](#)

mam\_dr, [5](#)

mam\_sparse, [6](#)

mam\_sparse\_dr, [8](#)

plotfuns, [10](#)

generateData, [2](#)

mam, [3](#)

mam-function (mam), [3](#)

mam\_dr, [5](#)

mam\_dr-function (mam\_dr), [5](#)

mam\_sparse, [6](#)

mam\_sparse-function (mam\_sparse), [6](#)

mam\_sparse\_dr, [8](#)

mam\_sparse\_dr-function (mam\_sparse\_dr),  
[8](#)

plotfuns, [10](#)

tensorMam (tensorMam-package), [2](#)

tensorMam-package, [2](#)

TransferModalUnfoldings, [11](#)

TransferModalUnfoldings-function  
(TransferModalUnfoldings), [11](#)