Package ‘TensorClustering’

June 26, 2021

Type Package
Title Model-Based Tensor Clustering
Version 1.0.2
License MIT + file LICENSE
Encoding UTF-8
LazyData false
Imports MASS, tensr, rTensor, TRES, abind, combinat, pracma, stats, mvtnorm, Rcpp
RoxygenNote 7.1.1
LinkingTo Rcpp
NeedsCompilation yes
Author Kai Deng [aut, cre], Yuqing Pan [aut], Xin Zhang [aut], Qing Mai [aut]
Maintainer Kai Deng <kd18h@stat.fsu.edu>
Repository CRAN
Date/Publication 2021-06-26 05:40:02 UTC

R topics documented:

DEEM .......................... 2
TEMM ................................ 4
TGMM .............................. 6
tune_K ............................ 8
tune_lamb ........................ 9
tune_u_joint .................... 11
tune_u_sep ...................... 12
DEEM Index

Doubly-enhanced EM algorithm

Description

Doubly-enhanced EM algorithm for tensor clustering

Usage

DEEM(X, nclass, niter = 100, lambda = NULL, dfmax = n, pmax = nvars, pf = rep(1, nvars),
eps = 1e-04, maxit = 1e+05, sml = 1e-06, verbose = FALSE, ceps = 0.1,
initial = TRUE, vec_x = NULL)

Arguments

X Input tensor (or matrix) list of length n, where n is the number of observations. Each element of the list is a tensor or matrix. The order of tensor can be any positive integer not less than 2.
nclass Number of clusters.
niter Maximum iteration times for EM algorithm. Default value is 100.
lambda A user-specified lambda value. lambda is the weight of L1 penalty and a smaller lambda allows more variables to be nonzero
dfmax The maximum number of selected variables in the model. Default is the number of observations n.
pmax The maximum number of potential selected variables during iteration. In middle step, the algorithm can select at most pmax variables and then shrink part of them such that the number of final selected variables is less than dfmax.
pf Weight of lasso penalty. Default is a vector of value 1 and length p, representing L1 penalty of length p. Can be modified to use adaptive lasso penalty.
eps Convergence threshold for coordinate descent difference between iterations. Default value is 1e-04.
maxit Maximum iteration times for coordinate descent for all lambda. Default value is 1e+05.
sml Threshold for ratio of loss function change after each iteration to old loss function value. Default value is 1e-06.
verbose Indicates whether print out lambda during iteration or not. Default value is FALSE.
ceps Convergence threshold for cluster mean difference between iterations. Default value is 1.
initial Whether to initialize algorithm with K-means clustering. Default value is TRUE.
vec_x Vectorized tensor data. Default value is NULL.
DEEM

Details

The DEEM function implements the Doubly-Enhanced EM algorithm (DEEM) for tensor clustering. The observations $X_i$ are assumed to be following the tensor normal mixture model (TNMM) with common covariances across different clusters:

$$X_i \sim \sum_{k=1}^{K} \pi_k \text{TN} (\mu_k; \Sigma_1, \ldots, \Sigma_M), \quad i = 1, \ldots, n,$$

where $0 < \pi_k < 1$ is the prior probability for $X$ to be in the $k$-th cluster such that $\sum_{k=1}^{K} \pi_k = 1$, $\mu_k$ is the cluster mean of the $k$-th cluster and $\Sigma_1, \ldots, \Sigma_M$ are the common covariances across different clusters. Under the TNMM framework, the optimal clustering rule can be showed as

$$\hat{Y}_{opt} = \arg \max_k \{ \log \pi_k + \langle X - (\mu_1 + \mu_k)/2, B_k \rangle \},$$

where $B_k = [\mu_k - \mu_1; \Sigma_1^{-1}, \ldots, \Sigma_M^{-1}]$. In the enhanced E-step, DEEM imposes sparsity directly on the optimal clustering rule as a flexible alternative to popular low-rank assumptions on tensor coefficients $B_k$ as

$$\min_{B_2, \ldots, B_K} \left[ \sum_{k=2}^{K} (\langle B_k, [B_k, \tilde{\Sigma}_1^{(t)}, \ldots, \tilde{\Sigma}_M^{(t)}] \rangle - 2 \langle B_k, \tilde{\mu}_k^{(t)} - \tilde{\mu}_1^{(t)} \rangle) + \lambda^{(t+1)} \sum_{J} \left[ \sum_{k=2}^{K} b_{k,J}^2 \right] \right],$$

where $\lambda^{(t+1)}$ is a tuning parameter. In the enhanced M-step, DEEM employs a new estimator for the tensor correlation structure, which facilitates both the computation and the theoretical studies.

Value

- pi: A vector of estimated prior probabilities for clusters.
- mu: A list of estimated cluster means.
- sigma: A list of estimated covariance matrices.
- gamma: A n by nclass matrix of estimated membership weights.
- y: A vector of estimated labels.
- iter: Number of iterations until convergence.
- df: Average zero elements in beta over iterations.
- beta: A matrix of vectorized $B_{\cdot,k}$.

Author(s)

Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai

References


See Also

tune_lamb, tune_K
Examples

dimen = c(5,5,5)
nvars = prod(dimen)
K = 2
n = 100
sigma = array(list(),3)
sigma[[1]] = sigma[[2]] = sigma[[3]] = diag(5)

B2=array(0,dim=dimen)
B2[1:3,1,1]=2

y = c(rep(1,50),rep(2,50))
M = array(list(),K)
M[[1]] = array(0,dim=dimen)
M[[2]] = B2

vec_x=matrix(rnorm(n*prod(dimen)),ncol=n)
X=array(list(),n)
for (i in 1:n){
  X[[i]] = array(vec_x[,i],dim=dimen)
  X[[i]] = M[[y[i]]] + X[[i]]
}

myfit = DEEM(X, nclass=2, lambda=0.05)

---

**TEMM**

*Fit the Tensor Envelope Mixture Model (TEMM)*

**Description**

Fit the Tensor Envelope Mixture Model (TEMM)

**Usage**

`TEMM(Xn, u, K, initial = "kmeans", iter.max = 500, stop = 1e-3, trueY = NULL, print = FALSE)`

**Arguments**

- **Xn** The tensor for clustering, should be array type, the last dimension is the sample size n.
- **u** A vector of envelope dimension
- **K** Number of clusters, greater than or equal to 2.
- **initial** Initialization method for the regularized EM algorithm. Default value is "kmeans".
- **iter.max** Maximum number of iterations. Default value is 500.
stop Convergence threshold of relative change in cluster means. Default value is 1e-3.
trueY A vector of true cluster labels of each observation. Default value is NULL.
print Whether to print information including current iteration number, relative change in cluster means and clustering error (%) in each iteration.

Details

The TEMM function fits the Tensor Envelope Mixture Model (TEMM) through a subspace-regularized EM algorithm. For mode $m$, let $(\Gamma_m, \Gamma_0m) \in \mathbb{R}^{p_m \times p_m}$ be an orthogonal matrix where $\Gamma_m \in \mathbb{R}^{p_m \times u_m}$, $u_m \leq p_m$, represents the material part. Specifically, the material part $X_{*,m} = X \times_m \Gamma_m^T$ follows a tensor normal mixture distribution, while the immaterial part $X_{d,m} = X \times_m \Gamma_0m^T$ is unimodal, independent of the material part and hence can be eliminated without loss of clustering information. Dimension reduction is achieved by focusing on the material part $X_{*,m} = X \times_m \Gamma_m^T$.

Collectively, the joint reduction from each mode is

$$X_* = [X; \Gamma_1^T, \ldots, \Gamma_M^T] \sim \sum_{k=1}^{K} \pi_k \mathcal{TN}(\alpha_k; \Omega_1, \ldots, \Omega_M), \quad X_* \perp \perp X_{d,m},$$

where $\alpha_k \in \mathbb{R}^{u_1 \times \cdots \times u_M}$ and $\Omega_m \in \mathbb{R}^{u_m \times u_m}$ are the dimension-reduced clustering parameters and $X_{d,m}$ does not vary with cluster index $Y$.

In the E-step, the membership weights are evaluated as

$$\hat{\eta}(s)_{ik} = \frac{\hat{\pi}_k^{(s)}}{\sum_{k=1}^{K} \hat{\pi}_k^{(s)}} f_k(X_i; \hat{\theta}^{(s)}),$$

where $f_k$ denotes the conditional probability density function of $X_i$ within the $k$-th cluster. In the subspace-regularized M-step, the envelope subspace is iteratively estimated through a Grassmann manifold optimization that minimize the following log-likelihood-based objective function:

$$G_m^{(s)}(\Gamma_m) = \log |\Gamma_m^T M_m^{(s)} \Gamma_m| + \log |\Gamma_m^T (N_m^{(s)})^{-1} \Gamma_m|,$$

where $M_m^{(s)}$ and $N_m^{(s)}$ are given by

$$M_m^{(s)} = \frac{1}{np-m} \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{\eta}_ik^{(s)} (e_{ik}^{(s)})_m (\hat{\Sigma}_m^{(s)-1})_m^{-1} (e_{ik}^{(s)}),$$

$$N_m^{(s)} = \frac{1}{np-m} \sum_{i=1}^{n} (X_i)_m (\hat{\Sigma}_m^{(s)-1})_m^{-1} (X_i)^T.$$

The intermediate estimators $M_m^{(s)}$ can be viewed the mode-$m$ conditional variation estimate of $X \mid Y$ and $N_m^{(s)}$ is the mode-$m$ marginal variation estimate of $X$.

Value

id A vector of estimated labels.
pi A vector of estimated prior probabilities for clusters.
eta A $n$ by $K$ matrix of estimated membership weights.
Mu.est A list of estimated cluster means.
SIG.est A list of estimated covariance matrices.
Mm Estimation of $M_m$ defined in paper.
Nm Estimation of $N_m$ defined in paper.
Gamma.est A list of estimated envelope basis.
PGamma.est A list of envelope projection matrices.

Author(s)
Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai

References

See Also
TGMM, tune_u_sep, tune_u_joint

Examples
A = array(c(rep(1,20),rep(2,20))+rnorm(40),dim=c(2,2,10))
myfit = TEMM(A,u=c(2,2),K=2)

---

**TGMM**
*Fit the Tensor Gaussian Mixture Model (TGMM)*

Description
Fit the Tensor Gaussian Mixture Model (TGMM)

Usage
TGMM(Xn, K, shape = "shared", initial = "kmeans",
iter.max = 500, stop = 1e-3, trueY = NULL, print = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xn</td>
<td>The tensor for clustering, should be array type, the last dimension is the sample size n.</td>
</tr>
<tr>
<td>K</td>
<td>Number of clusters, greater than or equal to 2.</td>
</tr>
<tr>
<td>shape</td>
<td>&quot;shared&quot; if assume common covariance across mixtures, &quot;distinct&quot; if allow different covariance structures. Default value is &quot;shared&quot;.</td>
</tr>
<tr>
<td>initial</td>
<td>Initialization method for the regularized EM algorithm. Default value is &quot;kmeans&quot;.</td>
</tr>
</tbody>
</table>
iter.max  Maximum number of iterations. Default value is 500.

stop Convergence threshold of relative change in cluster means. Default value is 1e-3.

trueY A vector of true cluster labels of each observation. Default value is NULL.

print Whether to print information including current iteration number, relative change in cluster means and clustering error (%) in each iteration.

Details

The TGMM function fits the Tensor Gaussian Mixture Model (TGMM) through the classical EM algorithm. TGMM assumes the following tensor normal mixture distribution of M-way tensor data \( X \):

\[
X \sim \sum_{k=1}^{K} \pi_k \mathcal{TN}(\mu_k, M_k), \quad i = 1, \ldots, n,
\]

where \( 0 < \pi_k < 1 \) is the prior probability for \( X \) to be in the \( k \)-th cluster such that \( \sum_{k=1}^{K} \pi_k = 1 \), \( \mu_k \) is the mean of the \( k \)-th cluster, \( M_k \equiv \{ \Sigma_{km} \mid m = 1, \ldots, M \} \) is the set of covariances of the \( k \)-th cluster. If \( M_k \)'s are the same for \( k = 1, \ldots, K \), call TGMM with argument shape="shared".

Value

- id A vector of estimated labels.
- pi A vector of estimated prior probabilities for clusters.
- eta A \( n \) by \( K \) matrix of estimated membership weights.
- Mu.est A list of estimated cluster means.
- SIG.est A list of estimated covariance matrices.

Author(s)

Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai

References


See Also

- TEMM

Examples

```r
A = array(c(rep(1,20),rep(2,20))+rnorm(40),dim=c(2,2,10))
myfit = TGMM(A,K=2,shape="shared")
```
tune_K

Select the number of clusters \( K \) in DEEM

Description

Select the number of clusters \( K \) along with tuning parameter \( \lambda \) through BIC in DEEM.

Usage

\[
tune_K(X, \text{seqK}, \text{seqlamb}, \text{initial} = \text{TRUE}, \text{vec_x} = \text{NULL})
\]

Arguments

- **\( X \)**: Input tensor (or matrix) list of length \( n \), where \( n \) is the number of observations. Each element of the list is a tensor or matrix. The order of tensor can be any positive integer not less than 2.
- **\( \text{seqK} \)**: A sequence of user-specified number of clusters.
- **\( \text{seqlamb} \)**: A sequence of user-specified \( \lambda \) values. \( \lambda \) is the weight of L1 penalty and a smaller \( \lambda \) allows more variables to be nonzero.
- **\( \text{initial} \)**: Whether to initialize algorithm with K-means clustering. Default value is \( \text{TRUE} \).
- **\( \text{vec_x} \)**: Vectorized tensor data. Default value is \( \text{NULL} \).

Details

The \( \text{tune}_K \) function runs \( \text{tune}_\lambda \) function \( \text{length}(\text{seqK}) \) times to choose the tuning parameter \( \lambda \) and number of clusters \( K \) simultaneously. Let \( \hat{\theta}^{(\lambda,K)} \) be the output of DEEM with the tuning parameter and number of clusters fixed at \( \lambda \) and \( K \) respectively, \( \text{tune}_K \) looks for the values of \( \lambda \) and \( K \) that minimizes

\[
\text{BIC}(\lambda, K) = -2 \sum_{i=1}^{n} \log\left( \sum_{k=1}^{K} \hat{z}_k^{(\lambda,K)} f_k(X_i; \hat{\theta}_k^{(\lambda,K)}) \right) + \log(n) \cdot |\hat{D}^{(\lambda,K)}|,
\]

where \( \hat{D}^{(\lambda,K)} = \{(k,J) : \hat{b}_{k,J}^{(\lambda,K)} \neq 0\} \) is the set of nonzero elements in \( \hat{B}_2^{(\lambda,K)}, \ldots, \hat{B}_K^{(\lambda,K)} \). The \( \text{tune}_K \) function intrinsically selects the initial point and return the optimal estimated labels.

Value

- **\( \text{opt}_K \)**: Selected number of clusters that leads to optimal BIC.
- **\( \text{opt}_\lambda \)**: Tuned \( \lambda \) value that leads to optimal BIC.
- **\( \text{Krank} \)**: A selection summary.

Author(s)

Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai
References


See Also

DEEM, tune_lamb

Examples

dimen = c(5,5,5)
nvars = prod(dimen)
K = 2
n = 100
sigma = array(list(),3)
sigma[[1]] = sigma[[2]] = sigma[[3]] = diag(5)

B2 = array(0,dim=dimen)
B2[[1:3,1,1]]=2

y = c(rep(1,50),rep(2,50))
M = array(list(),K)
M[[1]] = array(0,dim=dimen)
M[[2]] = B2

vec_x = matrix(rnorm(n*prod(dimen)),ncol=n)
X = array(list(),n)
for (i in 1:n){
    X[[i]] = array(vec_x[,i],dim=dimen)
    X[[i]] = M[y[i]] + X[[i]]
}

mytune = tune_K(X, seqK=2:4, seqlamb=seq(0.01,0.1,by=0.01))

tune_lamb

Parameter tuning in enhanced E-step in DEEM

Description

Perform parameter tuning through BIC in DEEM.

Usage

tune_lamb(X, K, seqlamb, initial = TRUE, vec_x = NULL)
Arguments

X  Input tensor (or matrix) list of length \( n \), where \( n \) is the number of observations. Each element of the list is a tensor or matrix. The order of tensor can be any positive integer not less than 2.

K  Number of clusters.

seq_lamb  A sequence of user-specified lambda values. lambda is the weight of L1 penalty and a smaller lambda allows more variables to be nonzero.

initial  Whether to initialize algorithm with K-means clustering. Default value is \texttt{TRUE}.

vec_x  Vectorized tensor data. Default value is \texttt{NULL}.

details

The `tune_lamb` function adopts a BIC-type criterion to select the tuning parameter \( \lambda \) in the enhanced E-step. Let \( \widehat{\theta}^{\lambda} \) be the output of \texttt{DEEM} with the tuning parameter fixed at \( \lambda \), `tune_lamb` looks for the value of \( \lambda \) that minimizes

\[
\text{BIC}(\lambda) = -2 \sum_{i=1}^{n} \log \left( \sum_{k=1}^{K} \widehat{\pi}_k^{\lambda} f_k(\mathbf{X}_i; \widehat{\theta}^{\lambda}_k) \right) + \log(n) \cdot |\widehat{D}^{\lambda}|,
\]

where \( \widehat{D}^{\lambda} = \{(k, J) : \hat{b}^{\lambda}_{k,J} \neq 0\} \) is the set of nonzero elements in \( \widehat{B}^{\lambda}_2, \ldots, \widehat{B}^{\lambda}_K \). The `tune_lamb` function intrinsically selects the initial point and return the optimal estimated labels.

Value

\texttt{opt_lamb} Tuned lambda that leads to optimal BIC.

\texttt{opt_bic} BIC value.

\texttt{opt_y} Estimated labels fitted by \texttt{DEEM} with tuned lambda.

Author(s)

Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai

References


See Also

\texttt{DEEM, tune_K}

Examples

dimen = c(5,5,5)
nvars = prod(dimen)
K = 2
n = 100
tune_u_joint

sigma = array(list(), 3)
sigma[[1]] = sigma[[2]] = sigma[[3]] = diag(5)

B2 = array(0, dim=dimen)
B2[1:3, 1, 1] = 2

y = c(rep(1, 50), rep(2, 50))
M = array(list(), K)
M[[1]] = array(0, dim=dimen)
M[[2]] = B2

vec_x = matrix(rnorm(n * prod(dimen)), ncol=n)
X = array(list(), n)
for (i in 1:n) {
  X[[i]] = array(vec_x[, i], dim=dimen)
  X[[i]] = M[y[i]] + X[[i]]
}

mytune = tune_lamb(X, K=2, seqlamb=seq(0.01, 0.1, by=0.01))

---

**tune_u_joint**  
*Tuning envelope dimension jointly by BIC in TEMM.*

**Description**

Tuning envelope dimension jointly by BIC in TEMM.

**Usage**

tune_u_joint(u_candi, K, X, iter.max = 500, stop = 0.001, trueY = NULL)

**Arguments**

- **u_candi**  
  A list of length M containing candidate envelope dimension for each mode.

- **K**  
  Number of clusters, greater than or equal to 2.

- **X**  
  The tensor for clustering, should be array type, the last dimension is the sample size n.

- **iter.max**  
  Maximum number of iterations. Default value is 500.

- **stop**  
  Convergence threshold of relative change in cluster means. Default value is 1e-3.

- **trueY**  
  A vector of true cluster labels of each observation. Default value is NULL.
Details

The `tune_u_joint` function searches over all the combinations of \( u \equiv (u_1, \ldots, u_M) \) in the neighborhood of \( \tilde{u} \), \( \mathcal{N}(\tilde{u}) = \{ u : \max(1, \tilde{u}_m - 2) \leq u_m \leq \min(\tilde{u}_m + 2, p_m), \; m = 1, \ldots, M \} \), that minimizes

\[
BIC(u) = -2 \sum_{i=1}^{n} \log\left( \sum_{k=1}^{K} \hat{\pi}_k^u f_k(X_i; \hat{\theta}_k^u) \right) + \log(n) \cdot K_u.
\]

In the above BIC, \( K_u = (K - 1) \prod_{m=1}^{M} u_m + \sum_{m=1}^{M} p_m (p_m + 1)/2 \) is the total number of parameters in TEMM, \( \hat{\pi}_k^u \) and \( \hat{\theta}_k^u \) are the estimated parameters with envelope dimension fixed at \( u \). The `tune_u_joint` function intrinsically selects the initial point and return the optimal estimated labels.

Value

- opt.u: Optimal envelope dimension selected.
- opt.id: Estimated labels fitted by TEMM with the optimal envelope dimension.
- opt.Mu: Estimated cluster means fitted by TEMM with the optimal envelope dimension.
- bic: BIC value.

Author(s)

Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai

References


See Also

`TEMN, tune_u_sep`

Examples

```r
A = array(c(rep(1,20),rep(2,20))+rnorm(40),dim=c(2,2,10))
mytune = tune_u_joint(u_candi=list(1:2,1:2),K=2,A)
```

tune_u_sep: Tuning envelope dimension separately by BIC in TEMM.

Description

Tuning envelope dimension separately by BIC in TEMM.
tune_u_sep

Usage

tune_u_sep(m, u_candi, K, X, C = 1, oneD = TRUE, iter.max = 500, stop = 0.001, trueY = NULL)

Arguments

m The tensor mode to be tuned, can take value in 1,\ldots,M.

u_candi A vector of candidate envelope dimension.

K Number of clusters, greater than or equal to 2.

X The tensor for clustering, should be array type, the last dimension is the sample size n.

C Constant in separate BIC criterion. Default value is 1.

oneD Whether to apply 1D-BIC tuning. Default value is TRUE.

iter.max Maximum number of iterations. Default value is 500.

stop Convergence threshold of relative change in cluster means. Default value is 1e-3.

trueY A vector of true cluster labels of each observation. Default value is NULL.

Details

For tensor mode \( m = 1,\ldots,M \), the \texttt{tune_u_sep} function selects the envelope dimension \( \bar{u}_m \) by minimizing the following BIC-type criterion over the set \( \{0,1,\ldots,p_m\} \),

\[
\text{BIC}_m(u_m) = \log |\Gamma_m^T \bar{M}_m \Gamma_m| + \log |\Gamma_m^T \bar{N}_m^{-1} \Gamma_m| + C \cdot u_m \log(n)/n.
\]

This separate selection over each mode \( m \) is less sensitive to the complex interrelationships of each mode of the tensor. The default constant \( C \) is set as 1 as suggested by Zhang and Mai (2018).

Value

opt.u Optimal envelope dimension selected.

bic BIC value.

Author(s)

Kai Deng, Yuqing Pan, Xin Zhang and Qing Mai

References


See Also

\texttt{TEMM, tune_u_joint}
Examples

A = array(c(rep(1,20),rep(2,20))+rnorm(40),dim=c(2,2,10))
mytune = tune_u_sep(1,1:2,K=2,A)
## Index

DEEM, 2, 3, 8–10

TEMM, 4, 5, 7, 12, 13
TGMM, 6, 6, 7
tune\_K, 3, 8, 8, 10
tune\_lamb, 3, 8, 9, 9, 10
tune\_u\_joint, 6, 11, 12, 13
tune\_u\_sep, 6, 12, 12, 13