Package ‘tsvd’

March 11, 2015

Type Package

Title Thresholding-based SVD for multivariate reduced rank regression

Version 1.1

Date 2015-03-09

Author Xin Ma, Luo Xiao, and Wing H. Wong*

Maintainer Xin Ma <xm24@stanford.edu>

Description This package performs multivariate reduced rank regression with a sparse singular value decomposition on the coefficient matrix (T-SVD). Sparsity is achieved through iterative (hard) thresholding on the left and right singular vectors.

Depends MASS, glmnet

License GPL (>=2)

\textbf{\textit{R} topics documented:}

\begin{verbatim}
    tsvd-package  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
    select_rank  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
    soda  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
    tsvd  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
    tsvd_ini  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
    tsvd_wrapper  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
\end{verbatim}

Index

\begin{verbatim}
    tsvd-package  tsvd
\end{verbatim}

1
Description

Threshold-based SVD for multivariate reduced rank regression. The development of microarray and next-generation sequencing technologies has enabled rapid quantification of various genome-wide features (genomic sequences, gene expressions, noncoding RNA expressions, methylation etc.) in a population of samples. Large consortia have compiled genetic and molecular profiling data in an enormous number of tumors samples. There are some key characteristics in these regulatory relationships. First, multiple pathways are involved and can be viewed as independent programs of regulation. Second, combinatorial nature of the regulation in each program is likely to be sparse. Third, the number of potential predictors and responses often far exceed the sample size. To address these challenges, we develop this computationally efficient multivariate response regression model T-SVD.

Details

This package performs multivariate reduced rank regression with a sparse SVD on the coefficient matrix. The model is

\[ Y = XC + E, \]

where \( C \) is an unknown coefficient matrix. Let \( UDV^T \) be the singular value decomposition (SVD) of \( C \). The method in this package intends to obtain sparse estimates of \( U \) and \( V \).

Author(s)

Xin Ma, Luo Xiao, Wing H. Wong*

References


Usage

```
select_rank(Y, X, constant = 4)
```
soda

Arguments

Y  n by q response matrix
X  n by p design matrix
constant  defaults to 4

Value

the selected rank

Author(s)

Xin Ma <xm24@stanford.edu> Luo Xiao <lxiao@jhsph.edu> Wing H. Wong* <whwong@stanford.edu>

References


Examples

n <- 100
p <- 200
q <- 100
rank <- 3
X <- matrix(rnorm(n*p),n,p)
U <- matrix(rnorm(p*rank),p,rank)
V <- matrix(rnorm(q*rank),q,rank)
D <- diag(rnorm(rank))
C <- U%*%D%*%t(V)
Y <- X%*%C + matrix(rnorm(n*q),n,q)
select_rank(Y,X)

soda

Sparse Orthogonal Decomposition Algorithm

Description

This function is similar to the QR decomposition of a numeric matrix but provides potentially more sparse singular vectors if the input matrix is sparse. If the input has no sparsity, then one gets the same result as the QR decomposition.

Usage

soda(x)
Arguments

x a numeric matrix whose left singular vectors are to be computed

Value

Q left singular vectors with zero rows removed
R R matrix as in the QR decomposition
index indices of the rows of Q
d singular values of x

Author(s)

Xin Ma, Luo Xiao, Wing H. Wong*

References


See Also

qr

Examples

## Example 1: no sparsity in the data
## same as the QR decomposition
x <- matrix(rnorm(200),40,5)
Q <- qr.Q(qr(x))
Q1 <- soda(x)$Q
f <- function(x,y){ min(sum((x-y)^2),sum((x+y)^2))}
diff <- 0
for(r in 1:5) diff <- diff + f(Q[,r],Q1[,r])
print(diff)

## Example 2: sparse data
## different from the QR decomposition
x <- matrix(rnorm(900),30,3)
x[1:10,2] <- 0
x[1:20,3] <- 0
Q <- qr.Q(qr(x))
Q1 <- soda(x)$Q
print(cbind(x,Q,Q1),digits=3)
tsvd

Thresholding-based SVD for multivariate reduced rank regression

Description
This function performs (hard) thresholding-based SVD for multivariate reduced rank regression proposed by Ma, Xiao and Wong (2014).

Usage
```r
tsvd(Y, X, rank, ini, BICtype = 2, thr = NULL,
       control = list(thr.lower = -1/zero.noslash, thr.upper = -1, thr.by = 0.5))
```

Arguments
- **Y**: n by q response matrix
- **X**: n by p design matrix
- **rank**: rank of the coefficient matrix
- **ini**: initial estimate of the coefficient matrix
- **BICtype**: two BIC types
- **thr**: a pair of tuning parameters or a 2-column matrix of tuning parameters to search over
- **control**: a list of arguments for defining the sequence of tuning parameters; see details

Details
There are two types of BICs that can be used. BICtype 1:
\[
\log(SSE) + \log(q * n)/(q * n)df_f,
\]
where \(df_f\) is the total number of non-zero entries in \(U, V\) and \(D\), subtracted by squared rank. BIC type 2:
\[
\log(SSE) + \log(q * n)/(q * n)df_v + \log(r * n)/(r * n)df_u,
\]
where \(r\) is the rank, \(df_v\) is the number of non-zero entries in \(V\), subtracted by \(r * (r - 1)/2\), and \(df_u\) is the number of non-zero entries in \(U\), subtracted by \(r * (r - 1)/2\). The second BIC type performs better for variable selection on \(U\).

Value
- **est**: estimate of the coefficient matrix in “SparseSVD” structure
- **C**: estimate of the coefficient matrix
- **thr**: selected tuning parameters
- **BICtype**: BICtype used
- **BIC**: value of BIC

Author(s)
Xin Ma, Luo Xiao, Wing H. Wong*
References


Examples

n <- 100
p <- 150
q <- 150

## generate a sparse coefficient matrix
lnorm <- function(a, l=1) (sum(abs(a)^2))^(1/2) # normalizing function
U <- matrix(ncol=3,nrow=p);
V <- matrix(ncol=3,nrow=q);
V[,1] <- c(sample(c(-1,1),5,replace=TRUE),rep(0,5));
V[,2] <- c(rep(0,7),sample(c(-1,1),5,replace=TRUE),rep(0,5));
V[,3] <- c(rep(0,5),V[1:2,1],-V[3:4,1],sample(c(-1,1),2,replace=TRUE),
-V[15:16,2],V[13:14,2],rep(0,9))
V[,1] <- V[,1]/lnorm(V[,1],2);
V[,2] <- V[,2]/lnorm(V[,2],2);
V[,3] <- V[,3]/lnorm(V[,3],2);
U[,1] <- c(sample(c(1,-1),5,replace=TRUE)*runif(5,.7,1),rep(0,5));
U[,2] <- c(rep(0,7),sample(c(-1,1),5,replace=TRUE)*runif(5,.7,1),rep(0,15));
U[,3] <- c(rep(0,5),U[1:2,1],rep(0,5));
U[,1] <- U[,1]/lnorm(U[,1],2);
U[,2] <- U[,2]/lnorm(U[,2],2);
U[,3] <- U[,3]/lnorm(U[,3],2);
D <- diag(c(2,1,5));
C <- U%*%D%*%t(V);

## generate data
X <- matrix(rnorm(n*p),n,p)
E <- matrix(rnorm(n*q),n,q)
sigma2 <- sum(diag(t(t(C)%*%C)))/(n*q) # so that the signal to noise ratio is 1
Y <- X%*%C + sqrt(sigma2)*E

## obtain estimate
ini1 <- tsvd_ini(Y,X,rank=3)
ans1 <- tsvd(Y,X,rank=3,ini=ini1,BICtype=2)
ini2 <- tsvd_ini(Y,X,rank=3,ini=ini1$est,method=2)
ans2 <- tsvd(Y,X,rank=3,ini=ini2,BICtype=2)
print(sum((ans2$C-C)^2))

An initial estimator for multivariate reduced rank regression

Description

This function provides an initial estimate for multivariate reduced rank regression.
Usage

```
htsvdini(Y, X, rank, ini = NULL, method = 1)
```

Arguments

- **Y**
  - n by q response matrix
- **X**
  - n by p design matrix
- **rank**
  - rank of the estimate
- **ini**
  - an initial estimate of the coefficient matrix in “SparseSVD” structure
- **method**
  - method = 1: rank-truncated ridge regression estimate; method = 2: assuming V is known from ini, estimate U by lasso(implemented by glmnet) for which the tuning parameter is selected by BIC.

Value

- **rank**
  - rank
- **u**
  - left singular vectors with zero rows removed
- **v**
  - right singular vectors with zero rows removed
- **d**
  - singular values
- **u.index**
  - indices of rows of u
- **v.index**
  - indices of rows of v

Author(s)

Xin Ma, Luo Xiao, Wing H. Wong*

References


Examples

```
n <- 100
p <- 150
q <- 150

# generate a sparse coefficient matrix
lnorm <- function(a, l=1) (sum(abs(a)^2))^(1/2) #normaling function
U <- matrix(ncol= 3,nrow= p);
V <- matrix(ncol= 3,nrow= q);
V[,1] <- c(sample(c(-1,1),5,replace= TRUE),rep(0,20));
V[,2] <- c(rep(0,12),sample(c(-1,1),8,replace= TRUE),rep(0 ,5));
V[,3] <- c(rep(0,6),V[,1],-V[,2],sample(c(-1,1),2,replace= TRUE),
           -V[,15:16,2],V[13:14,2],rep(0,9))
V[,1] <- V[,1]/lnorm(V[,1],2);
V[,2] <- V[,2]/lnorm(V[,2],2);
V[,3] <- V[,3]/lnorm(V[,3],2);
```
\[ U[,1] \leftarrow \text{c(sample(c(1,-1),5,replace= TRUE)*runif(5,0.7,1),rep(0,20))}; \]
\[ U[,2] \leftarrow \text{c(rep(0,5),sample(c(1,-1),5,replace= TRUE)*runif(5,0.7,1),rep(0,15))}; \]
\[ U[,3] \leftarrow \text{c(rep(0,10),sample(c(1,-1),5,replace= TRUE)*runif(5,0.7,1),rep(0,10))}; \]
\[ U[,1] \leftarrow U[,1]/\text{lnorm(U[,1],2)}; \]
\[ U[,2] \leftarrow U[,2]/\text{lnorm(U[,2],2)}; \]
\[ U[,3] \leftarrow U[,3]/\text{lnorm(U[,3],2)}; \]
\[ D \leftarrow \text{diag(c(2,1,5))}; \]
\[ C \leftarrow U\%*\%X\%*\%t(Y); \]

## generate data
\[ X \leftarrow \text{matrix(rnorm(n*p),n,p)} \]
\[ E \leftarrow \text{matrix(rnorm(n*q),n,q)} \]
\[ \text{sigma2} \leftarrow \text{sum(diag(t(C))}\%*\%C)/n\%*\%q \]  

## so that the signal to noise ratio is 1
\[ Y\leftarrow X\%*\%C+ \text{sqrt(\text{sigma2})}\%*\%E \]

## obtain initial estimate
\[ \text{ini} \leftarrow \text{tsvd_ini(Y,X,rank=3)} \]

---

### tsvd_wrapper

**Thresholding-based SVD for multivariate reduced rank regression**

**Description**

This function is a wrapper for performing (hard) thresholding-based SVD for multivariate reduced rank regression proposed by Ma, Xiao and Wong (2014).

**Usage**

\[
\text{tsvd_wrapper}(Y, X, \text{rank=NULL, BICtype = 2, thr = NULL, control = list(thr.lower = -10, thr.upper = -1, thr.by = 0.5))}
\]

**Arguments**

- **Y**: n by q response matrix
- **X**: n by p design matrix
- **rank**: rank of the coefficient matrix; if **NULL**, estimated by `select_rank`
- **BICtype**: two BIC types
- **thr**: a pair of tuning parameters or a 2-column matrix of tuning parameters to search over
- **control**: a list of arguments for defining the sequence of tuning parameters; see details

**Details**

The model is
\[ Y = XC + E, \]

where \( C \) is an unknown coefficient matrix. Let \( UDV^T \) be the singular value decomposition (SVD) of \( C \). The method in this package intends to obtain sparse estimates of \( U \) and \( V \).

If **rank** is **NULL**, an estimate is obtained from **select_rank**.
The wrapper has four steps: (1) obtain an initial estimate from rank-truncated ridge regression; (2) run tsvd; (3) with V from step (2), obtain another estimate of U and D from lasso; (4) run tsvd.

There are two types of BICs that can be used. BICtype 1:

$$\log(SSE) + \log(q \ast n)/(q \ast n)df,$$

where $df$ is the total number of non-zero entries in $U,V$ and $D$, subtracted by squared rank. BIC type 2:

$$\log(SSE) + \log(q \ast n)/(q \ast n)df_v + \log(r \ast n)/(r \ast n)df_u,$$

where $r$ is the rank, $df_v$ is the number of non-zero entries in $V$, subtracted by $r \ast (r-1)/2$, and $df_u$ is the number of non-zero entries in $U$, subtracted by $r \ast (r-1)/2$. The second BIC type performs better for variable selection on $U$.

**Value**

- est estimate of the coefficient matrix in “SparseSVD” structure
- C estimate of the coefficient matrix
- thr selected tuning parameters
- BICtype BICtype used
- BIC value of BIC

**Author(s)**

Xin Ma, Luo Xiao, Wing H. Wong*

**References**


**Examples**

```r
n <- 100
p <- 150
q <- 150

## generate a sparse coefficient matrix

inorm <- function(a, l=1) (sum(abs(a)^2))^(1/2) #normaling function

U <- matrix(ncol= 3,nrow= p);
V <- matrix(ncol= 3,nrow= q);
V[,1] <- c(sample(c(-1,1),5,replace= TRUE),rep(0,20));
V[,2] <- c(rep(0,12),sample(c(-1,1),8,replace= TRUE),rep(0 ,5));
V[,3] <- c(rep(0,6),V[1:2,1],-V[3:4,1],sample(c(-1,1),2,replace= TRUE),
           -V[15:16,2],V[13:14,2],rep(0,9))
V[,1] <- V[,1]/inorm(V[,1],2);
V[,2] <- V[,2]/inorm(V[,2],2);
V[,3] <- V[,3]/inorm(V[,3],2);

U[,1] <- c(sample(c(1,-1),5,replace= TRUE)*runif(5,0.7,1),rep(0,20));
U[,2] <- c(rep(0,5),sample(c(1,-1),5,replace= TRUE)*runif(5,0.7,1),rep(0,15));
U[,3] <- c(rep(0,10),sample(c(1,-1),5,replace= TRUE)*runif(5,0.7,1),rep(0,10));
U[,1] <- U[,1]/inorm(U[,1],2);
U[,2] <- U[,2]/inorm(U[,2],2);
```
U[,3] <- U[,3]/lnorm(U[,3],2);
D <- diag(c(20,10,5));

C <- U%*%D%*%t(V);
## generate data
X <- matrix(rnorm(n*p),n,p)
E <- matrix(rnorm(n*q),n,q)
sigma2 <- sum(diag(t(C)%*%C))/(n*q) ## so that the signal to noise ratio is 1
Y <- X%*%C + sqrt(sigma2)*E

## obtain estimate
ans <- tsvd_wrapper(Y,X,BICtype=2) ##rank is estimated
print(sum((ans$C-C)^2))
## Index

*Topic **array**
  soda, 3

*Topic **multivariate**
  select_rank, 2
tsvd, 5
tsvd_ini, 6
tsvd_wrapper, 8

*Topic **package**
  tsvd-package, 1

*Topic **regression**
  select_rank, 2
tsvd, 5
tsvd_ini, 6
tsvd_wrapper, 8

qr, 4

select_rank, 2
soda, 3
tsvd, 5
tsvd-package, 1
tsvd_ini, 6
tsvd_wrapper, 8