Package ‘tsvd’

March 17, 2015

Type Package

Title Thresholding-based SVD for multivariate reduced rank regression

Version 1.2

Date 2015-03-18

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)

R topics documented:

<table>
<thead>
<tr>
<th>tsvd-package</th>
<th>tsvd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Index

1
select_rank

Description

Thresholding-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 consortiums 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


select_rank

Rank selection for multivariate reduced rank regression

Description

This function selects the rank of the coefficient matrix for multivariate response regression using
the approach proposed by Bunea, She and Wegkamp (2011). Specifically, let \( P = X(X^TX)^{-}X^T \),
where the generalized inverse is the Moore-Penrose pseudoinverse, and \( \lambda_k(Y^TPY) \) be the \( k \)-th
eigenvalue of \( Y^TPY \). Then the selected rank is
\[ K = \max\{k : \lambda_k(Y^TPY) \geq c \max(q,n)\sigma^2\}, \]
where \( c \) is the constant and \( \sigma \) is the noise level of the data, estimated by \( 1.4826\, \text{MAD}(\text{abs}(Y)) \).

Usage

select_rank(Y, X, sigma2 = NULL, constant = 4, abs.tol = 10^(-7), rel.tol = 1e-04)
Arguments

Y       n by q response matrix
X       n by p design matrix
sigma2  random error variance in Y
constant defaults to 4
abs.tol absolute tolerance for numerical error, defaults to 1e-07
rel.tol relative tolerance for numerical error, defaults to 1e-04

Value

the selected rank

Author(s)

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

References


Examples

```r
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 = -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
- **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:

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

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

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

where $r$ is the rank, $df_v$ is the number of non-zero entries in $V$, subtracted by $r \times (r - 1)/2$, and $df_u$ is the number of non-zero entries in $U$, subtracted by $r \times (r - 1)/2$. 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
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(0,20));
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]/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(5,0.7,1),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]/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(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=ans1$est,method=2)
ans2 <- tsvd(Y,X,rank=3,ini=ini2,BICtype=2)
print(sum((ans2$C-C)^2))
```

An initial estimator for multivariate reduced rank regression

### Description

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

tsvd_ini(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

Ma, X., Xiao, L. and Wong, W.H. (2014), Learning Regulatory Programs by Threshold SVD
Regression. PNAS, 2014 111(44) 15675-15680.

Friedman, J., Hastie, T. and Tibshirani, R. (2010), Regularization paths for generalized linear mod-

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(0,20));
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]/lnorm(V[1]);
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

```r
 tsvd_wrapper(Y, X, rank=NULL, 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; 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`.

---

**tsvd_wrapper**

```r
t[1,1] <- 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]/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(C)%*%C))/(n*q) ## so that the signal to ##noise ratio is 1
Y<-X%*%C+ sqrt(sigma2)*E
## obtain initial estimate
ini <- tsvd_ini(Y,X,rank=3)
```
The wrapper has four steps: (1) obtain an initial estimate from rank-truncated ridge regression; (2) run \textit{tsvd}; (3) with \(V\) from step (2), obtain another estimate of \(U\) and \(D\) from lasso; (4) run \textit{tsvd}.

There are two types of BICs that can be used. BIC type 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 \(U\).

\begin{itemize}
  \item \textbf{Value}
    \begin{itemize}
      \item \texttt{est}\hspace{1cm} estimate of the coefficient matrix in “SparseSVD” structure
      \item \texttt{C}\hspace{1cm} estimate of the coefficient matrix
      \item \texttt{thr}\hspace{1cm} selected tuning parameters
      \item \texttt{BICtype}\hspace{1cm} BIC type used
      \item \texttt{BIC}\hspace{1cm} value of BIC
    \end{itemize}
  \item \textbf{Author(s)}
    \begin{itemize}
      \item Xin Ma, Luo Xiao, Wing H. Wong*
    \end{itemize}
  \item \textbf{References}
    \begin{itemize}
    \end{itemize}
  \item \textbf{Examples}
    \begin{itemize}
      \item \texttt{n} <- 100
      \item \texttt{p} <- 150
      \item \texttt{q} <- 150
      \item \# generate a sparse coefficient matrix
        \item \texttt{lnorm} <- function(a,l=1) (sum(abs(a)^2))^(1/2) # normalizing function
        \item \texttt{U} <- matrix(ncol= 3,nrow= p);
        \item \texttt{V} <- matrix(ncol= 3,nrow= q);
        \item \texttt{V[,1]} <- c(sample(c(-1,1),5,replace= TRUE),rep(0,20));
        \item \texttt{V[,2]} <- c(rep(0,12),sample(c(-1,1),8,replace= TRUE),rep(0,5));
        \item \texttt{V[,3]} <- c(rep(0,6),V[,2],1,1,-V[,3:4,1],sample(c(-1,1),2,replace= TRUE),
          -V[15:16,2],V[13:14,2],rep(0,9))
        \item \texttt{U[,1]} <- c(sample(c(-1,1),5,replace= TRUE)*runif(5,0.7,1),rep(0,20));
        \item \texttt{U[,2]} <- c(rep(0,5),sample(c(-1,1),5,replace= TRUE)*runif(5,0.7,1),rep(0,15));
        \item \texttt{U[,3]} <- c(rep(0,10),sample(c(-1,1),5,replace= TRUE)*runif(5,0.7,1),rep(0,10));
        \item \texttt{U[,1]} <- U[,1]/lnorm(U[,1],2);
        \item \texttt{U[,2]} <- U[,2]/lnorm(U[,2],2);
        \item \texttt{U[,3]} <- U[,3]/lnorm(U[,3],2);
    \end{itemize}
\end{itemize}
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