연세대학교 통계 데이터 사이언스 학회 ESC 23-2 Final Project

Estimating Number of Factors by Adjusted Eigenvalues Thresholding

1조 : 전인태 이상윤 왕재혁 정석훈 최영준 노희준

1.Abstract

Abstract

Recall

Orthogonal Factor Model with m Common Factors

$$\underset{(p\times 1)}{\mathbf{X}} = \underset{(p\times 1)}{\boldsymbol{\mu}} + \underset{(p\times m)}{\mathbf{L}}\underset{(m\times 1)}{\mathbf{F}} + \underset{(p\times 1)}{\boldsymbol{\varepsilon}}$$

$$\begin{aligned}
\mu_i &= \textit{mean} \text{ of variable } i \\
\varepsilon_i &= i\text{th } \textit{specific factor} \\
F_j &= j\text{th } \textit{common factor} \\
\ell_{ij} &= \textit{loading} \text{ of the } i\text{th variable on the } j\text{th factor}
\end{aligned} \tag{9-4}$$

The unobservable random vectors $\mathbf{F}$ and $\boldsymbol{\varepsilon}$ satisfy the following conditions:

$\mathbf{F}$ and $\boldsymbol{\varepsilon}$ are independent

$E(\mathbf{F}) = \mathbf{0}$, $\text{Cov}(\mathbf{F}) = \mathbf{I}$

$E(\boldsymbol{\varepsilon}) = \mathbf{0}$, $\text{Cov}(\boldsymbol{\varepsilon}) = \boldsymbol{\Psi}$, where $\boldsymbol{\Psi}$ is a diagonal matrix

$$\begin{aligned}
X_1 - \mu_1 &= \ell_{11}F_1 + \ell_{12}F_2 + \cdots + \ell_{1m}F_m + \varepsilon_1 \\
X_2 - \mu_2 &= \ell_{21}F_1 + \ell_{22}F_2 + \cdots + \ell_{2m}F_m + \varepsilon_2 \\
&\vdots \\
X_p - \mu_p &= \ell_{p1}F_1 + \ell_{p2}F_2 + \cdots + \ell_{pm}F_m + \varepsilon_p
\end{aligned}$$

$$\underset{(p\times 1)}{\mathbf{X} - \boldsymbol{\mu}} = \underset{(p\times m)}{\mathbf{L}}\underset{(m\times 1)}{\mathbf{F}} + \underset{(p\times 1)}{\boldsymbol{\varepsilon}}$$

$$E(\mathbf{F}) = \underset{(m\times 1)}{\mathbf{0}}, \qquad \text{Cov}(\mathbf{F}) = E[\mathbf{F}\mathbf{F}'] = \underset{(m\times m)}{\mathbf{I}}$$

$$E(\boldsymbol{\varepsilon}) = \underset{(p\times 1)}{\mathbf{0}}, \qquad \text{Cov}(\boldsymbol{\varepsilon}) = E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'] = \underset{(p\times p)}{\boldsymbol{\Psi}} = \begin{bmatrix} \psi_1 & 0 & \cdots & 0 \\ 0 & \psi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_p \end{bmatrix}$$

Abstract

What it means?

Estimating Number of Factors by Adjusted Eigenvalues Thresholding

⇒ Why we should estimate Number of Factors?

$$
\begin{aligned}
X_1 - \mu_1 &= \ell_{11}F_1 + \ell_{12}F_2 + \cdots + \ell_{1m}F_m + \varepsilon_1 \\
X_2 - \mu_2 &= \ell_{21}F_1 + \ell_{22}F_2 + \cdots + \ell_{2m}F_m + \varepsilon_2 \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
X_p - \mu_p &= \ell_{p1}F_1 + \ell_{p2}F_2 + \cdots + \ell_{pm}F_m + \varepsilon_p
\end{aligned}
$$

$$
\underset{(p\times 1)}{\mathbf{X} - \boldsymbol{\mu}} = \underset{(p\times m)}{\mathbf{L}}\ \underset{(m\times 1)}{\mathbf{F}} + \underset{(p\times 1)}{\boldsymbol{\varepsilon}}
$$

When $m = p$?

Totally useless model.

We should find efficient number of factors, for

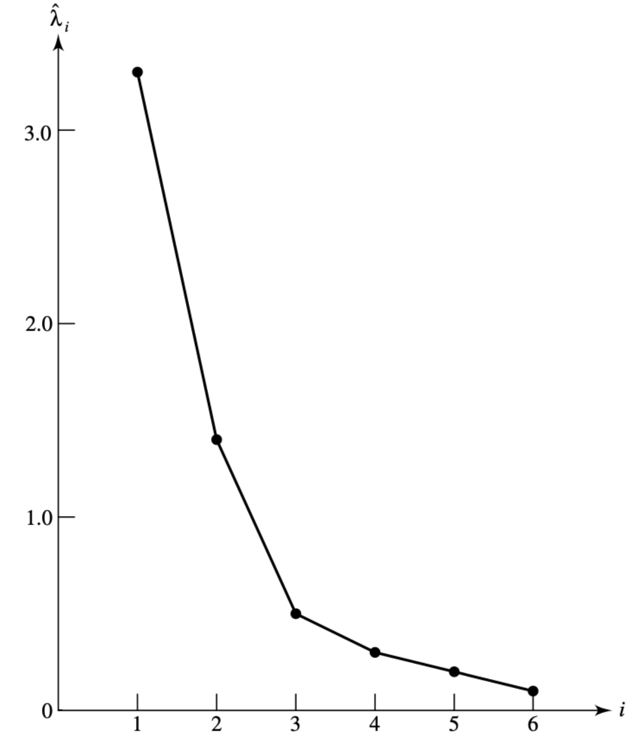
1. Selecting Meaningful Factors
2. Dimension Reduction

Abstract

How to estimate it? - Before

$$\begin{pmatrix}\text{Proportion of total}\\ \text{sample variance}\\ \text{due to } j\text{th factor}\end{pmatrix} = \begin{cases} \dfrac{\hat{\lambda}_j}{s_{11}+s_{22}+\cdots+s_{pp}} & \text{for a factor analysis of } \mathbf{S} \\[2ex] \dfrac{\hat{\lambda}_j}{p} & \text{for a factor analysis of } \mathbf{R} \end{cases}$$

1. # of eigenvalues of Covariance matrix, bigger than certain threshold.
    - # of eigenvalues of Correlation matrix, bigger than 1.
2. Draw an elbow(scree) plot



Abstract

How to estimate it? - Now

Use eigenvalues of Correlation Matrix.

- To solve different magnitude issues.

Then, is that still # of eigenvalues of Correlation matrix, bigger than 1?

$\Rightarrow$ Similar, but there exists a problem.

- Inconsistency in estimating eigenvalues of **High dimensional Population** Correlation matrix.

To solve this, correct the biases in estimating the top eigenvalues,

And take into account of estimation errors in eigenvalue estimation.

We propose

1. Bias Correction of sample eigenvalues
2. adjusted correlation thresholding (ACT)

Abstract

Bias Correction of sample eigenvalues

Let $\hat{\lambda}_j = \lambda_j(\hat{\mathbf{R}})$ and $\lambda_j = \lambda_j(\mathbf{R})$ for $j \in [p]$. For any given j , define

$$
\begin{aligned}
m_{n,j}(z) &= (p-j)^{-1}\Big[\sum_{\ell=j+1}^{p} (\hat{\lambda}_\ell - z)^{-1} + ((3\hat{\lambda}_j + \hat{\lambda}_{j+1})/4 - z)^{-1}\Big], \\
\underline{m}_{n,j}(z) &= -(1-\rho_{j,n-1})z^{-1} + \rho_{j,n-1} m_{n,j}(z),
\end{aligned}
$$

with $\rho_{j,n-1} = (p-j)/(n-1)$. Let the corrected eigenvalue of $\hat{\lambda}_j$ be

$$
\hat{\lambda}_j^C = -\frac{1}{\underline{m}_{n,j}(\hat{\lambda}_j)}, \quad j \in [r_{\max}].
$$

Abstract

Bias Correction of sample eigenvalues
+ adjusted correlation thresholding (ACT)

Summary of Method: We propose

$$\hat{K} = \max\{j : \hat{\lambda}_j^C > 1 + \sqrt{\rho_{n-1}}, j \in [r_{\max}]\},$$

where $\rho_{n-1} = p/(n-1)$. This is a simple and tuning free method.

Thus, by (24) and (25), when $s = 1 + \sqrt{\rho}$, we have

$$\lim_{n \to \infty} P(\hat{K}^C(s) = K) = 1.$$

2. Code Implementation

Code Implementation

```r
## for Calculating ACT
under_m <- function(n, j, z, hatEigValues, p){
  rho = (p-j)/(n-1)
  return( -(1-rho)/z + rho*m(n,j,z, hatEigValues, p) )
}
m <- function(n, j, z, hatEigValues, p){
  return( (p - j)^(-1) * ( sum( (hatEigValues[(j+1):p] - z)^(-1) ) + ( (3*hatEigValues[j] + hatEigValues[j+1])/4 - z )^(-1) ) )
}
```

```r
#### Estimate the number of factors
## Method 1: the method of zheng: estFN_by_all[,13]
sampleCovMat = cov(t(X)); hatRR=cov2cor(sampleCovMat) #Sample Correlation Matrix
lambdaHatRR = eigen(hatRR)$values #eigenvalues of Sample Correlation Matrix

lambdaCorrected = c() #Corrected eigenvalues by ACT
for(j in 1:rmax){
  lambdaCorrected[j] = -1/under_m(n, j, lambdaHatRR[j], lambdaHatRR, p)
}

if( all((lambdaCorrected > (1 + sqrt(p/(n-1)))) == F) ){
  estFN_by_ACT[KKK] = 0
  estFN_by_all[KKK,13] = estFN_by_ACT[KKK]
} else{
  estFN_by_ACT[KKK] = tail( which( lambdaCorrected > (1 + sqrt(p/(n-1))) ), n=1 )
  estFN_by_all[KKK,13] = estFN_by_ACT[KKK]
} #estimated Factor Number is max j which lambdaCorrected > (1 + sqrt(p/(n-1))) satisfy.
```

Code Implementation

Simulation Studies ; contained in paper

Case 2: Let $b_{\ell j}$ be iid from $N(0,1)$ and $\nu_1^2, \cdots, \nu_p^2$ be iid from Unif$(0,180)$.

Case 3: Let $b_{\ell j}$ be iid from $N(0,1)$ and $\nu_1^2 = \cdots = \nu_p^2 = 36$. The model is used in Bai and Ng (2002) and Onatski (2010).

Case 4: Let $b_{jj} = 1$, $b_{\ell j}$ be iid from $N(0, 0.04)$ for $j \neq \ell$ and $\nu_1^2, \cdots, \nu_p^2$ be iid from Unif$(0, 5.5)$.

Code Implementation

Case 2: Let $b_{\ell j}$ be iid from $N(0,1)$ and $\nu_1^2, \cdots, \nu_p^2$ be iid from Unif$(0, 180)$.

p		PC_3	IC_3	ON_2	ER	GR	ACT
		Gaussian population					
100	TRUE	0	0	0.1	4.2	4.4	64.3
	OVER	0	0	0	6.6	7.3	0.10
	UNDER	100	100	99.9	89.2	88.3	35.6
	AVE	1.18	1	1.53	2.29	2.37	4.58
300	TRUE	47.0	1.7	31.2	27.0	28.2	98.9
	OVER	0	0	0.1	0.4	0.4	1.1
	UNDER	53.0	98.3	68.7	72.6	71.4	0
	AVE	4.42	2.81	4.17	3.01	3.07	5.01
500	TRUE	0	0	98.8	88.9	89.7	98.9
	OVER	0	0	0	0	0	1.1
	UNDER	100	100	1.2	11.1	10.3	0
	AVE	2.44	1.16	4.99	4.76	4.78	5.01
1000	TRUE	0	0	99.9	99.9	99.9	99.1
	OVER	0	0	0.1	0	0	0.9
	UNDER	100	100	0	0.1	0.1	0
	AVE	1.17	1	5	5	5	5.01

$p = 100$

```
[1] 1000
      [,1] [,2]   [,3]  [,4]  [,5]  [,6]
[1,]  0.00    0   0.00  3.40  3.60 63.50
[2,]  0.00    0   0.00  7.90  8.80  0.30
[3,] 100.00  100 100.00 88.70 87.60 36.20
[4,]  1.19    1   1.59  2.31  2.39  4.58
```

$p = 300$

```
[1] 1000
      [,1]  [,2]  [,3]  [,4]  [,5] [,6]
[1,] 46.30  2.00 28.50 23.70 24.50 99.7
[2,]  0.10  0.00  0.00  0.30  0.30  0.3
[3,] 53.60 98.00 71.50 76.00 75.20  0.0
[4,]  4.42  2.82  4.14  2.89  2.94  5.0
```

$p = 500$

```
[1] 200
       [,1]   [,2]  [,3]  [,4]  [,5]  [,6]
[1,]   0.00   0.00 97.00 84.00 85.00 98.50
[2,]   0.00   0.00  1.00  0.00  0.00  1.50
[3,] 100.00 100.00  2.00 16.00 15.00  0.00
[4,]   2.44   1.18  4.99  4.64  4.68  5.02
```

$p = 1000$

```
[1] 100
      [,1] [,2]  [,3] [,4] [,5]  [,6]
[1,]   0.0    0 99.00  100  100 98.00
[2,]   0.0    0  1.00    0    0  2.00
[3,] 100.0  100  0.00    0    0  0.00
[4,]   1.2    1  5.01    5    5  5.02
```

Code Implementation

Case 3: Let $b_{\ell j}$ be iid from $N(0,1)$ and $\nu_1^2 = \cdots = \nu_p^2 = 36$. The model is used in Bai and Ng (2002) and Onatski (2010).

p		PC_3	IC_3	ON_2	ER	GR	ACT
				Gaussian population			
100	TRUE	0	0	0.1	5.5	5.8	0
	OVER	0	0	0	9.6	9.7	0
	UNDER	100	100	99.9	84.9	84.5	100
	AVE	1	1	1.27	2.51	2.54	1.06
300	TRUE	0	0	1.1	4.2	4.6	5.4
	OVER	0	0	0	0.8	0.9	0
	UNDER	100	100	98.9	95	94.5	94.6
	AVE	1	1	2.85	2.1	2.14	2.91
500	TRUE	0	0	32.5	26.0	27.3	71.3
	OVER	0	0	0	0.2	0.2	2.8
	UNDER	100	100	67.5	73.8	72.5	25.9
	AVE	1	1	4.2	2.92	2.97	4.74
1000	TRUE	0	0	99.6	92.3	92.7	96.2
	OVER	0	0	0	0	0	3.8
	UNDER	100	100	0.4	7.7	7.3	0
	AVE	1	1	5	4.81	4.83	5.04

$p = 100$

```
[1] 1000
     [,1] [,2]   [,3]  [,4]  [,5]   [,6]
[1,]    0    0   0.00  4.60  4.80   0.00
[2,]    0    0   0.00  9.90 10.60   0.00
[3,]  100  100 100.00 85.50 84.60 100.00
[4,]    1    1   1.29  2.51  2.58   0.46
```

$p = 300$

```
[1] 1000
     [,1] [,2]  [,3] [,4]  [,5]  [,6]
[1,]    0    0  1.30  5.0  5.20  4.70
[2,]    0    0  0.00  0.9  1.00  0.30
[3,]  100  100 98.70 94.1 93.80 95.00
[4,]    1    1  2.81  2.1  2.14  2.89
```

$p = 500$

```
[1] 200
     [,1] [,2]  [,3]  [,4]  [,5]  [,6]
[1,]    0    0 37.00 31.50 33.00 75.00
[2,]    0    0  0.00  0.00  0.00  1.50
[3,]  100  100 63.00 68.50 67.00 23.50
[4,]    1    1  4.26  3.11  3.16  4.75
```

$p = 1000$

```
[1] 100
     [,1] [,2] [,3]  [,4]  [,5]  [,6]
[1,]    0    0  100 94.00 94.00 97.00
[2,]    0    0    0  0.00  0.00  3.00
[3,]  100  100    0  6.00  6.00  0.00
[4,]    1    1    5  4.84  4.84  5.03
```

Code Implementation

Case 4: Let $b_{jj} = 1$, $b_{\ell j}$ be iid from $N(0, 0.04)$ for $j \neq \ell$ and $\nu_1^2, \cdots, \nu_p^2$ be iid from Unif$(0, 5.5)$.

p		PC_3	IC_3	ON_2	ER	GR	ACT
		Gaussian population					
100	TRUE	0.2	0	0.7	3.9	4.6	98.20
	OVER	0	0	0	1.9	2.4	0.20
	UNDER	99.8	100	99.3	94.2	93	1.60
	AVE	2.4	1	2.85	2.14	2.21	4.99
300	TRUE	99.5	81.7	97.8	81.6	83	99.3
	OVER	0.1	0	0.1	0	0	0.7
	UNDER	0.4	18.3	2.1	18.4	17.0	0
	AVE	5	4.81	4.98	4.55	4.6	5.01
500	TRUE	63.9	18.5	100	99.9	99.9	99.4
	OVER	0	0	0	0	0	0.6
	UNDER	36.1	81.5	0	0.1	0.1	0
	AVE	4.63	3.81	5	5	5	5.01
1000	TRUE	4.9	0.1	99.9	100	100	99.5
	OVER	0	0	0.1	0	0	0.5
	UNDER	95.1	99.9	0.0	0	0	0
	AVE	3.6	2.54	5	5	5	5

$p = 100$

```
[1] 1000
      [,1] [,2]  [,3] [,4]  [,5]  [,6]
[1,]  0.40    0  1.70  4.1  4.70 97.60
[2,]  0.00    0  0.00  1.5  2.00  0.10
[3,] 99.60  100 98.30 94.4 93.30  2.30
[4,]  2.42    1  2.85  2.1  2.17  4.98
```

$p = 300$

```
[1] 1000
     [,1]  [,2]  [,3]  [,4]  [,5] [,6]
[1,] 99.1 85.90 97.80 83.70 85.40 99.8
[2,]  0.2  0.00  0.10  0.00  0.00  0.2
[3,]  0.7 14.10  2.10 16.30 14.60  0.0
[4,]  5.0  4.86  4.98  4.61  4.67  5.0
```

$p = 500$

```
[1] 200
      [,1]  [,2] [,3]  [,4]  [,5] [,6]
[1,] 66.50 17.00  100 99.50 99.50 99.5
[2,]  0.00  0.00    0  0.00  0.00  0.5
[3,] 33.50 83.00    0  0.50  0.50  0.0
[4,]  4.64  3.82    5  4.99  4.99  5.0
```

$p = 1000$

```
[1] 100
      [,1]  [,2]  [,3] [,4] [,5]  [,6]
[1,]  9.00  1.00 99.00  100  100 99.00
[2,]  0.00  0.00  1.00    0    0  1.00
[3,] 91.00 99.00  0.00    0    0  0.00
[4,]  3.67  2.53  5.01    5    5  5.01
```

Code Implementation

```
## Custom Model1  # When loading matrix is diagonal
beta<-diag(1, nrow = p, ncol = r); diagc<-diag(sqrt( runif(p,0,5.5) )
```

$$X_1 - \mu_1 = \ell_{11}F_1 + \ell_{12}F_2 + \cdots + \ell_{1m}F_m + \varepsilon_1$$
$$X_2 - \mu_2 = \ell_{21}F_1 + \ell_{22}F_2 + \cdots + \ell_{2m}F_m + \varepsilon_2$$
$$\vdots \qquad\qquad \vdots$$
$$X_p - \mu_p = \ell_{p1}F_1 + \ell_{p2}F_2 + \cdots + \ell_{pm}F_m + \varepsilon_p$$

$$\underset{(p\times1)}{\mathbf{X} - \boldsymbol{\mu}} = \underset{(p\times m)}{\mathbf{L}}\ \underset{(m\times1)}{\mathbf{F}} + \underset{(p\times1)}{\boldsymbol{\varepsilon}}$$

Expected Effect

Loading Matrix is a diagonal → only one variable is affected by factor
No common factor exists ⇒ Although $r = 5$, expected factor number is 0.

```
[1] 200
     [,1] [,2]   [,3]  [,4]  [,5]  [,6]
[1,]    0    0   0.00  6.50  6.50 0e+00
[2,]    0    0   0.00 12.00 13.00 0e+00
[3,]  100  100 100.00 81.50 80.50 1e+02
[4,]    1    1   0.69  2.91  3.01 7e-02
```

Code Implementation

```
## Custom Model2  # no error term
beta<-matrix(rnorm(p*r,0,1),p,r); diagc<-diag(rep(0, p))
```

$$X_1 - \mu_1 = \ell_{11}F_1 + \ell_{12}F_2 + \cdots + \ell_{1m}F_m + \varepsilon_1$$
$$X_2 - \mu_2 = \ell_{21}F_1 + \ell_{22}F_2 + \cdots + \ell_{2m}F_m + \varepsilon_2$$
$$\vdots \qquad\qquad\qquad \vdots$$
$$X_p - \mu_p = \ell_{p1}F_1 + \ell_{p2}F_2 + \cdots + \ell_{pm}F_m + \varepsilon_p$$

$$\underset{(p\times 1)}{\mathbf{X} - \boldsymbol{\mu}} = \underset{(p\times m)}{\mathbf{L}}\underset{(m\times 1)}{\mathbf{F}} + \underset{(p\times 1)}{\boldsymbol{\varepsilon}}$$

Expected Effect

No error term ⇒ estimation will be great.

```
[1] 200
     [,1]  [,2]  [,3] [,4]  [,5] [,6]
[1,]    0 41.00 89.50  100 13.00  100
[2,]  100 12.50 10.50    0 87.00    0
[3,]    0 46.50  0.00    0  0.00    0
[4,]   10  4.67  5.11    5  6.35    5
```

Code Implementation

```
## Custom Model3.1  # Multicollinearity, one column
A=matrix(NA, nrow = p, ncol = r)
A[,1:(r-1)] = matrix( rnorm(p*(r-1),0,1), nrow = p, ncol = r-1)
A[,r]=runif(r-1) %*% t( A[,1:r-1] )  #make last column be a linear combination of other columns.
beta=A
diagc<-diag(sqrt( runif(p,0,20) ))*3
```

Expected Effect

Last factor's effect will be covered by rest factor's effects. ⇒ estimate 4 common factors.

```
[1] 200
       [,1]   [,2] [,3]   [,4]   [,5]  [,6]
[1,]   0.00   0.00  0.5   0.00   0.00  2.00
[2,]   0.00   0.00  0.0   0.00   0.00  0.00
[3,] 100.00 100.00 99.5 100.00 100.00 98.00
[4,]   2.46   1.41  4.0   1.51   1.52  4.02
```

Code Implementation

```
## Custom Model3.2  # Multicollinearity, two columns
A=matrix(NA, nrow = p, ncol = r)
A[,1:(r-2)] = matrix( rnorm(p*(r-2),0,1), nrow = p, ncol = r-2)
A[,(r-1)]=runif(r-2) %*% t( A[,1:(r-2)] )  #make column be a linear combination of other columns.
A[,r]=runif(r-2) %*% t( A[,1:(r-2)] )  #make column be a linear combination of other columns.
beta=A
diagc<-diag(sqrt( runif(p,0,20) ))*3
```

Expected Effect

Last two factor's effect will be covered by rest factor's effects. ⇒ estimate 3 common factors.

```
[1] 200
       [,1]   [,2] [,3]   [,4]   [,5]  [,6]
[1,]   0.00   0.00    0   0.00   0.00  0.50
[2,]   0.00   0.00    0   0.00   0.00  0.00
[3,] 100.00 100.00  100 100.00 100.00 99.50
[4,]   2.17   1.52    3   1.21   1.22  3.02
```

Code Implementation

```
## Custom Model4.1  # different scale for last column
A=matrix(NA, nrow = p, ncol = r)
A[,1:(r-1)] = matrix( rnorm(p*(r-1),0,1), nrow = p, ncol = r-1)
A[,r] = runif(p, 0, 0.01)  #different scale
beta=A
diagc<-diag(sqrt( runif(p,0,20) ))*3
```

Expected Effect

Last factor's effect will be dismissed by different scale with rest factors ⇒ estimate 4 common factors.

```
[1] 200
       [,1]   [,2]  [,3]   [,4]   [,5]  [,6]
[1,]   0.00   0.00  1.00   0.00   0.00  1.50
[2,]   0.00   0.00  0.00   0.00   0.00  0.00
[3,] 100.00 100.00 99.00 100.00 100.00 98.50
[4,]   2.03   1.06  4.01   3.85   3.85  4.02
```

Code Implementation

```
## Custom Model4.2  # different scale for last two columns
A=matrix(NA, nrow = p, ncol = r)
A[,1:(r-2)] = matrix( rnorm(p*(r-2),0,1), nrow = p, ncol = r-2)
A[,(r-1)] = runif(p, 0, 0.01)  #different scale
A[,r] = runif(p, 0, 0.01)  #different scale
beta=A
diagc<-diag(sqrt( runif(p,0,20) ))*3
```

Expected Effect

Last two factor's effect will be dismissed by different scale with rest factors ⇒ estimate 3 common factors.

```
[1] 200
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]   0.00   0.00   0.00   0.00   0.00   0.00
[2,]   0.00   0.00   0.00   0.00   0.00   0.00
[3,] 100.00 100.00 100.00 100.00 100.00 100.00
[4,]   1.67   1.07   3.02   2.88   2.88   3.02
```

3. Conclusion

Conclusion

The main contributions of this paper are as follows:

1. we establish the concise relationship between the eigenvalues of population correlation matrices and the number of common factors.

$$K = \max\{j : \lambda_j(\mathbf{R}) > 1, j \in [p]\}$$

2. we propose a bias corrected estimator $\hat{\lambda}_i^C$ for $\lambda_i(R)$, which in general differs from the i^{th} largest eigenvalue $\hat{\lambda}_i$ of sample correlation matrix and develop a new estimator for the number of common factors as follows:

$$\hat{K}^C = \max\{j : \hat{\lambda}_j^C > s, \quad j \in [r_{\max}]\},$$
$$s = 1 + \sqrt{p/(n-1)}$$

3. we derive the asymptotic properties of the largest K sample eigenvalues of the sample correlation matrix in high dimensional factor models.

In most of our testing cases, our estimation method outperforms the competing ones.
Even in the remaining cases considered in this paper, our estimation method has comparable performance to other competing methods.

Conclusion

When we can use this method?

- Factor Analysis Model
- High dimensional Data
    - finance, economics, neuroscience, genomics . . .

Roles & Responsibilities

Roles & Responsibilities

전인태 : 팀장, 논문 수리적 분석, 수식적 이해를 돕는 자료 제작

왕재혁 : 논문 수리적 분석, 수식적 이해를 돕는 자료 제작

이상윤 : 논문 내용 요약, Assumptions & Conditions 분석과 활용 방안 모색, 중간 발표

최영준 : 논문 내용 요약, Assumptions & Conditions 분석과 활용 방안 모색

노희준 : Code Implementation, Model structure 고안, 최종 발표

정석훈 : Code Implementation, Model structure 고안, 발표 준비

by 인태

Thm. 1. 어떠한 증명 과정을 거쳤더니, $\lambda_i(R)$ 중 k번째로 큰 eigenvalue 까지는 1보다 컸다. ① 또한 $\lambda_i(R)$ for $i \geq k+1$은 1 이하였다. ② 즉,

$$\begin{cases} \lambda_j(R) > 1 & j = 1, \cdots, k, \\ \lambda_j(R) \leq 1 & j = k+1, \cdots, p. \end{cases}$$

<u>Pf (②)</u> If $\lambda_1(Q_2 Q_2^T) = \|Q_2 Q_2^T\| = \|diag(\Sigma)^{-1}\Psi\|^2 \leq 1$ by assumption,

$$\lambda_i(R) \leq \lambda_{k+1}(Q_1 Q_1^T) + \lambda_1(Q_2 Q_2^T) = \lambda_1(Q_2 Q_2^T) \leq 1,$$

implying $\lambda_i(R) \leq 1$ for any $i \geq k+1$.

<u>Pf (①)</u> Let $v_i^2 := \Psi_{ii}$. Since $tr(R) = p$, we have

$$p = tr(Q_1^T Q_1) + \sum_{j=1}^{p} v_j^2 / \sigma_{jj}.$$

* This is because

- $R = Q_1 Q_1^T + Q_2 Q_2^T$,
- $tr(Q_1 Q_1^T) = tr(Q_1^T Q_1)$, $\cdots$ (a)
- $tr(Q_2 Q_2^T) = tr(diag(\Sigma)^{-1}\Psi)$ $= \sum_{j=1}^{p} v_j^2/\sigma_{jj}$.

(a) $tr(Q_1 Q_1^T) = \sum_{i=1}^{p} (Q_1 Q_1^T)_{ii}$
$= \sum_{i=1}^{p}\sum_{j=1}^{p} (Q_1)_{ij} (Q_1^T)_{ji}$
$= \sum_{j=1}^{p}\sum_{i=1}^{p} (Q_1^T)_{ji} (Q_1)_{ij}$
$= \sum_{j=1}^{p} (Q_1^T Q_1)_{jj} = tr(Q_1^T Q_1)$.

Then $tr(Q_1^T Q_1) = tr(Q_1 Q_1^T) = \sum_{j=1}^{p} \lambda_j(Q_1 Q_1^T) = \sum_{j=1}^{k} \lambda_j(Q_1 Q_1^T)$ $\cdots$ (b)

$= p - \sum_{j=1}^{p} v_j^2/\sigma_{jj} = \|diag(\Sigma)^{-1/2} B\|_F^2$. $\cdots$ (c)

By the assumption, we have

$\lambda_1(Q_1^T Q_1) / \lambda_k(Q_1^T Q_1)$

$= \|B^T[diag(\Sigma)]^{-1}B\| \cdot \|\{B^T[diag(\Sigma)]^{-1}B\}^{-1}\| = O(p^{d_2})$. $\cdots$ (d)

Then for some $C > 0$,

(b) Since $rank(Q_1 Q_1^T) = k < p$, $\{\lambda_j(Q_1 Q_1^T) \mid j = k+1, \cdots, p\} = \{0\}$.

(c) $\|A\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$ (d) $\lambda_k^{-1}(A) = \lambda_1(A^{-1})$

by 재혁

3.2 Bias correction of sample eigenvalues

이 파트의 주요내용은 sample correlation matrix 의 eigenvalue를 사용한 단순한 추정값은 consistent estimator 가 아니므로 bias 를 조정한 consistent estimator에 대해 소개하며, 그 이유에 대해 설명하고 있다.

Notation

$\hat{\lambda}_j$: sample correlation matrix의 j번째 eigenvalue $(\lambda_j(\hat{R}))$ $j \in [p]$

λ_j : population correlation matrix의 j번째 eigenvalue $(\lambda_j(R))$

새롭게 정의할 함수들

$$m_{n,j}(z) = (p-j)^{-1}\left[\sum_{\ell=j+1}^{p} (\hat{\lambda}_\ell - z)^{-1} + \left((3\hat{\lambda}_j + \hat{\lambda}_{j+1})/4 - z\right)^{-1}\right]$$

$$\underline{m}_{n,j}(z) = -(1-\rho_{j,n-1}) z^{-1} + \rho_{j,n-1} m_{n,j}(z) \quad \text{where } \rho_{j,n-1} = \frac{p-j}{n-1}$$

※ Corrected eigenvalue of $\hat{\lambda}_j$ ※

$$\boxed{\hat{\lambda}_j^C = -\frac{1}{\underline{m}_{n,j}(\hat{\lambda}_j)}, \quad j \in [r_{max}]}$$

Corrected eigenvalue 의 consistency를 다음의 theorem 을 통해 증명하였다.

Q&A

References

References

- [1] J. Fan, J. Guo, and S. Zheng, "Estimating number of factors by adjusted eigenvalues thresholding," 2019.
- [2] Johnson and Wichern - Applied Multivariate Statistical Analysis

END