High dimensional PCA - Demo

PCA vs Sparse PCA Examples

STAT 32950-24620

Spring 2025 (wk8)

PCA and sparse PCA Example 1

```
library(elasticnet)
```

Example 1 ($n = p$)

$V_1 \sim N(0, \sigma_1^2)$, $V_2 \sim N(0, \sigma_2^2)$, V_1, V_2 are independent.

$V_3 = c_1 V_1 + c_2 V_2 + \varepsilon_0$.

$X_i = V_1 + \varepsilon_i$, for $i = 1, 2, 3, 4$;

$X_i = V_2 + \varepsilon_i$, for $i = 5, 6, 7, 8$;

$X_i = V_3 + \varepsilon_i$, for $i = 9, 10$,

$\varepsilon_i \sim N(0, 1)$ are independent.

Data: i.i.d. samples from $X = (X_1, \cdots, X_p) \in \mathbb{R}^p$

Example 1 - Data, Dimensions, parameters

Discussions:

- The number of observations
- The dimension = number of variables
- The "true" dimension

Simulation parameters:

$p = 10$

$\sigma_1 = 290$, $\sigma_2 = 300$,

$c_1 = -0.3, c_2 = 0.95$.

$X = (X_1, \cdots, X_p) \in \mathbb{R}^p$

Create data p = 10, n=10

```
# Simulated data p = 10, n=10
set.seed(246)
n=10; p1=4; p2=4; p3=2
V1=rnorm(n,mean=0,sd=(290)); V2=rnorm(n,mean=0,sd=(300))
V3 = -0.3*V1 + 0.95*V2 + rnorm(n)
Xa = matrix(0,n, p1)
for (i in 1:p1){
  Xa[,i] = V1 + rnorm(n) }
Xb = matrix(0,n, p2)
for (i in 1:p2){
  Xb[,i] = V2 + rnorm(n) }
Xc = matrix(0,n, p3)
for (i in 1:p3){
  Xc[,i] = V3 + rnorm(n) }
X = cbind(Xa,Xb,Xc)
```

PCA on raw data

```r
# Variance(X_i)
round(diag(cov(X))/1000)
```

```
##  [1]  60  60  60  60 157 157 158 157 136 136
```

```r
# PCA use original data (p=10, n=10)
summary(princomp(X))
```

```
## Importance of components:
##                            Comp.1   Comp.2    Comp.3    Cc
## Standard deviation     900.6754 464.5826 1.516e+00 1.27(
## Proportion of Variance   0.7898   0.2102 2.239e-06 1.57:
## Cumulative Proportion    0.7898   1.0000 1.000e+00 1.00(
##                             Comp.6    Comp.7    Comp.8
## Standard deviation     5.719e-01 4.605e-01 3.115e-01 1.:
## Proportion of Variance 3.185e-07 2.065e-07 9.445e-08 1.4
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.(
```


PCA on raw data, 2 PC's

The first two principal components:

```r
princomp(X)$loadings[,1:2] # Obtain first 2 PCs by PCA
```

```
##          Comp.1    Comp.2
##  [1,] 0.05138  0.489877
##  [2,] 0.05167  0.489281
##  [3,] 0.05179  0.489435
##  [4,] 0.05094  0.489929
##  [5,] 0.41721  0.004499
##  [6,] 0.41779  0.005370
##  [7,] 0.41848  0.005022
##  [8,] 0.41787  0.005246
##  [9,] 0.38135 -0.143305
## [10,] 0.38163 -0.142863
```

Each is a linear combination of all p variables.

PCA on scaled variance=1 data

```r
# PCA using var=1 data, comparable PC variance
summary(princomp(X,cor=T)) # PCA using scaled data
```

```
## Importance of components:
##                        Comp.1 Comp.2    Comp.3    Comp.4
## Standard deviation     2.4841 1.9569 4.840e-03 3.977e-03
## Proportion of Variance 0.6171 0.3829 2.343e-06 1.582e-06
## Cumulative Proportion  0.6171 1.0000 1.000e+00 1.000e+0(
##                           Comp.7    Comp.8    Comp.9    (
## Standard deviation     1.394e-03 1.098e-03 4.048e-04 1.9
## Proportion of Variance 1.944e-07 1.205e-07 1.639e-08 3.8
## Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00 1.(
```


PCA on variance-1 data, 2 PC's

```r
# # Obtain the first 2 PCs by PCA using scaled data
princomp(X,cor=T)$loadings[,1:2]
```

```
##        Comp.1   Comp.2
##  [1,] 0.1533  0.47252
##  [2,] 0.1538  0.47226
##  [3,] 0.1539  0.47219
##  [4,] 0.1526  0.47285
##  [5,] 0.3957 -0.09372
##  [6,] 0.3958 -0.09319
##  [7,] 0.3958 -0.09341
##  [8,] 0.3958 -0.09326
##  [9,] 0.3736 -0.19024
## [10,] 0.3737 -0.18990
```

Sparse PCA objectives

In PCA, all variables contribute to the PC's (coefficient $\neq 0$.)

Objective of Sparse PCA

- Obtain sparse PCs with few non-zero coefficients.
- The obtained sparse PCs should be good approximations of the directions of original PCs.
- Sparse PCs should capture a decent amount of data variation.

Sparse PCA steps

To run the sparse PCA algorithm (e.g. `spca` in `R`):

- You decide more or less variables should contribute to the PCs.
- Each sparse PC contains predetermined number of variables allowed to have non-zero coefficients.
- May use data matrix for Sparse PCA (with command `type="predictor"`)
- May use covariance or correlation matrix for Sparse PCA (with command `type="gram"`)

Example 1 sparse PCA (4,4)

```
# 2 PCs by spca, 4 nonzero para(meter) each (Note: var %)
spca(X, K=2, type="predictor",sparse="varnum",para=c(4,4))
```

```
##
## Call:
## spca(x = X, K = 2, para = c(4, 4), type = "predictor", s
##
## 2 sparse PCs
## Pct. of exp. var. : 17.1 10.5
## Num. of non-zero loadings :  4 4
## Sparse loadings
##        PC1   PC2
## [1,] 0.000 0.725
## [2,] 0.000 0.000
## [3,] 0.000 0.000
## [4,] 0.000 0.667
## [5,] 0.000 0.000
## [6,] 0.000 0.000
```


Example 1 sparse PCA (4,4) - Loadings

```
# 2 PCs by spca; check 4 nonzero coeff. each
spca(X, K = 2, type = "predictor", sparse = "varnum",
     para = c(4, 4))$loadings
```

```
##            PC1     PC2
##  [1,]  0.00000  0.7254
##  [2,]  0.00000  0.0000
##  [3,]  0.00000  0.0000
##  [4,]  0.00000  0.6665
##  [5,]  0.00000  0.0000
##  [6,]  0.00000  0.0000
##  [7,] -0.99634  0.0000
##  [8,] -0.08095  0.0000
##  [9,] -0.01254 -0.0142
## [10,] -0.02454 -0.1713
```

Example 1 sparse PCA (4,4,2)

```r
spca(X, K = 3, type = "predictor", sparse = "varnum",
     para = c(4,4,2))$loadings
```

```
##          PC1      PC2      PC3
##  [1,]  0.0000  0.77821  0.00000
##  [2,]  0.0000  0.00000  0.00000
##  [3,]  0.0000  0.08775  0.00000
##  [4,]  0.0000  0.61448  0.00000
##  [5,]  0.0000  0.00000  0.00000
##  [6,]  0.0000  0.00000  0.00000
##  [7,] -0.9709  0.00000  0.00000
##  [8,] -0.1606  0.00000  0.09538
##  [9,] -0.1058  0.00000  0.00000
## [10,] -0.1427 -0.09540 -0.99544
```

Caveat: Reduction of variance explained.

Sparse PCA Example 2

Example 2 ($n < p$)

$V_1 \sim N(0,\sigma_1^2)$, $V_2 \sim N(0,\sigma_2^2)$, V_1, V_2 are independent.
$V_3 = c_1 V_1 + c_2 V_2 + \varepsilon_0$.
$X_i = V_1 + \varepsilon_i$, $i = 1,2,3,4$;
$X_i = V_2 + \varepsilon_i$, $i = 5,6,7,8$;
$X_i = V_3 + \varepsilon_i$, $i = 9,10,\cdots,30$,
$\varepsilon_i \sim N(0,1)$ are independent.
Simulation parameters:
$n = 10, p = 30$.
Keep $\sigma_1 = 290$, $\sigma_2 = 300$, $c_1 = -0.3, c_2 = 0.95$.
$X = (X_1,\cdots,X_p) \in \mathbb{R}^p$

Generate data for Example 2

```r
# Previous example p=10, n=10 extended to p=30, n=10
p4=20
Xd = matrix(0,n, p4)
for (i in 1:p4)
{
  Xd[,i] = V3 + rnorm(n)
}
X2 = cbind(X,Xd)

dim(X)
```

```
## [1] 10 10
```

```r
dim(X2)
```

```
## [1] 10 30
```


If we try PCA routine for $n < p$ ...

In the case of $n < p$, classical PCA does not apply.

```r
#princomp(X2)
```

```
R output


Error in princomp ......

......

'princomp' can only be used with more units than variables

......
```

Example 2 - Sparse PCA (30,30)

```
outall=spca(X2, K=2, type="predictor", sparse="varnum",
            para = c(30,30))
names(outall)
```

```
## [1] "call"     "type"     "K"        "loadings" "pev"
## [8] "para"     "lambda"
```

```
#outall$var.all  # total variance of the predictors
#outall$lambda  # quadratic penalty par default 1e-6
outall$pev  # % of explained variance
```

```
## [1] 0.93297 0.06703
```

Sparse PCA works in high dimensional case $n < p$.

Example 2 - Sparse PCA (10,10)

```
out=spca(X2, K=2, type="predictor", sparse="varnum",
         para = c(10,10))
out$loadings[c(1:12),]
```

```
##           PC1      PC2
##   [1,] 0.00000  0.24197
##   [2,] 0.00000  0.72856
##   [3,] 0.00000  0.51187
##   [4,] 0.00000  0.31775
##   [5,] 0.00000  0.01885
##   [6,] 0.00000  0.05075
##   [7,] 0.46042  0.00000
##   [8,] 0.00000  0.11047
##   [9,] 0.00000 -0.02732
##  [10,] 0.09481  0.00000
##  [11,] 0.00000  0.00000
##  [12,] 0.18810  0.00000
```


SPCA(10,4) Example 2: Check variation explained

```
round(t(out$loadings[13:21,1]),2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    0    0    0    0    0    0 0.05  0.4 0.27
```

```
round(t(out$loadings[22:30,1]),2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    0 0.14    0    0 0.07 0.62 0.33    0    0
```

```
out$pev
```

```
## [1] 0.24584 0.05348
```

Sparsity is obtained.
However variation explained can be reduced substantially.

Alternative setting: Sparse PCA L1 penalty

You may specify sparse="penalty" instead of sparse="varnum".

```
outL1=spca(X2, K=2, type="predictor", sparse="penalty",
           para = c(.5,.5))
#round(t(outL1$loadings[,1]),3)
colSums(outL1$loadings != 0)
```

```
## PC1 PC2
##   9   8
```

```
outL1=spca(X2, K=3, type="predictor", sparse="penalty",
           para = c(5,7,9))
#round(t(outL1$loadings[,1]),3)
colSums(outL1$loadings != 0)
```

```
## PC1 PC2 PC3
##   9   4   3
```

Sparse PCA data formats

PCA and Sparse PCA procedure can start with any of

data matrix
covariance matrix
correlation matrix

```
spca(x,K,para,type=c("predictor","Gram"),
sparse=c("penalty","varnum"),
use.corr=FALSE,lambda=1e-6,max.iter=200,
trace=FALSE,eps.conv=1e-3)
```

When using covariance or correlation matrix instead of data matrix,

need to specify `type = "Gram"`

Example 3 - Sparse PCA with correlation matrix

Example 3

Data of pip prop measurements, already in correlation matrix form.

```
# Example of sPCA for correlation matrix
data(pitprops)  # 13 input vars corr matrix from 180 obs
round(pitprops,1)[1:10,1:6]
```

```
##         topdiam length moist testsg ovensg ringtop
## topdiam     1.0    1.0   0.4    0.3   -0.1     0.3
## length      1.0    1.0   0.3    0.3   -0.1     0.3
## moist       0.4    0.3   1.0    0.9   -0.1     0.2
## testsg      0.3    0.3   0.9    1.0    0.2     0.4
## ovensg     -0.1   -0.1  -0.1    0.2    1.0     0.4
## ringtop     0.3    0.3   0.2    0.4    0.4     1.0
## ringbut     0.5    0.5   0.0    0.2    0.3     0.8
## bowmax      0.4    0.4  -0.1   -0.1    0.0     0.1
## bowdist     0.6    0.6   0.1    0.1    0.0     0.2
## whorls      0.5    0.6  -0.1    0.0    0.0     0.3
```


Example 3 corr matrix (cont.)

```
round(pitprops,1)[1:13,8:13]
```

```
##         bowmax bowdist whorls clear knots diaknot
## topdiam    0.4     0.6    0.5   0.1   0.0     0.1
## length     0.4     0.6    0.6   0.1   0.0     0.1
## moist     -0.1     0.1   -0.1   0.2   0.2     0.1
## testsg    -0.1     0.1    0.0   0.1   0.2     0.0
## ovensg     0.0     0.0    0.0  -0.1  -0.1    -0.2
## ringtop    0.1     0.2    0.3   0.0   0.0    -0.3
## ringbut    0.4     0.5    0.7  -0.1  -0.2    -0.4
## bowmax     1.0     0.5    0.6   0.1  -0.4    -0.2
## bowdist    0.5     1.0    0.5   0.1  -0.1    -0.1
## whorls     0.6     0.5    1.0  -0.3  -0.4    -0.3
## clear      0.1     0.1   -0.3   1.0   0.0     0.0
## knots     -0.4    -0.1   -0.4   0.0   1.0     0.2
## diaknot   -0.2    -0.1   -0.3   0.0   0.2     1.0
```


Example 3 data info

$p = 13$ explanatory var's, $n = 180$ obs on pit prop (mining lumber)
The response variables are maximum compressive strength.

```
TOPDIAM: the top diameter of the prop in inches;
LENGTH: the length of the prop in inches;
MOIST: the moisture content of the prop, expressed as a pe
TESTSG: the specific gravity of the timber at the time of t
OVENSG: the oven-dry specific gravity of the timber;
RINOTOP: the number of annual rings at the top of the prop
RINGBUT: the number of annual rings at the base of the prop
BowMAx: the maximum bow in inches;
BOWDIST: the distance of the point of maximum bow from the
WHORLS: the number of knot whorls;
CLEAR: the length of clear prop from the top of the prop in
KNOTS: the average number of knots per whorl;
DIAKNOT: the average diameter of the knots in inches.
```

Example 3 - PCA PCs

```r
round(princomp(covmat=pitprops)$loadings[,1:6],1)
```

```
##         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## topdiam    0.4    0.2    0.2    0.1    0.1    0.1
## length     0.4    0.2    0.2    0.1    0.1    0.2
## moist      0.1    0.5   -0.1   -0.1   -0.3   -0.3
## testsg     0.2    0.5   -0.4   -0.1   -0.4   -0.1
## ovensg     0.1   -0.2   -0.5    0.0   -0.2    0.6
## ringtop    0.3    0.0   -0.5    0.1    0.3    0.1
## ringbut    0.4   -0.2   -0.3    0.1    0.2    0.0
## bowmax     0.3   -0.2    0.2   -0.3   -0.2   -0.1
## bowdist    0.4    0.0    0.2   -0.1    0.1    0.0
## whorls     0.4   -0.2    0.1    0.2   -0.2   -0.2
## clear      0.0    0.2    0.1   -0.8    0.3    0.2
## knots     -0.1    0.3   -0.1    0.3    0.6   -0.2
## diaknot   -0.1    0.3    0.3    0.3   -0.1    0.6
```


Exampe 3 PCA PCs - hard to interpret

A classical example showing the difficulty of interpreting the PC.

```r
princomp(covmat=pitprops)$sdev^2/13  # % of PC variance
```

```
##   Comp.1   Comp.2   Comp.3   Comp.4   Comp.5   Comp.6
## 0.324510 0.182931 0.144479 0.085338 0.070004 0.062724 0.
##   Comp.9  Comp.10  Comp.11  Comp.12  Comp.13
## 0.027129 0.014680 0.003890 0.003190 0.002979
```

```r
sum((princomp(covmat=pitprops)$sdev)^2)
```

```
## [1] 13
```


Example 3 - Consider Sparse PCA

- Use correlation matrix directly (type = "Gram")
- Keep track of the progress (trace = TRUE)
- Give the number of principal components desired (K = ...)
- Give the number of non-zero terms in each component (para = c(..., ...))

```r
out0 = spca(pitprops,K=6,type="Gram",sparse="varnum",
    trace=TRUE, para=c(7,4,4,1,1,1))
```

```
## iterations 10
## iterations 20
## iterations 30
## iterations 40
```


Example 3 Sparse PCA (7,4,4,1,1,1)

```r
out0
```

```
##
## Call:
## spca(x = pitprops, K = 6, para = c(7, 4, 4, 1, 1, 1), ty
##     sparse = "varnum", trace = TRUE)
##
## 6 sparse PCs
## Pct. of exp. var. : 28.2 13.9 13.1  7.4  6.8  6.3
## Num. of non-zero loadings :  7 4 4 1 1 1
## Sparse loadings
##            PC1    PC2    PC3 PC4 PC5 PC6
## topdiam -0.477  0.003  0.000   0   0   0
## length  -0.469  0.000  0.000   0   0   0
## moist    0.000  0.785  0.000   0   0   0
## testsg   0.000  0.619  0.000   0   0   0
## ovensg   0.180  0.000 -0.656   0   0   0
## ringtop  0.000  0.000 -0.589   0   0   0
```

Example 3 Sparse PCA (4,4)

```
spca(pitprops,K=2,type="Gram",sparse="varnum",para=c(4,4))
```

```
##
## Call:
## spca(x = pitprops, K = 2, para = c(4, 4), type = "Gram",
##
## 2 sparse PCs
## Pct. of exp. var. : 16.4 15.6
## Num. of non-zero loadings :  4 4
## Sparse loadings
##           PC1   PC2
## topdiam  0.000 0.429
## length   0.000 0.359
## moist    0.000 0.772
## testsg   0.000 0.300
## ovensg   0.000 0.000
## ringtop  0.000 0.000
## ringbut -0.792 0.000
```


Example 3 "Sparse" PCA with all coefficients $\neq 0$

The result is equivalent to PCA

```
spca(pitprops,K=9,type="Gram",sparse="varnum",
     para=rep(13,9))
```

```
##
## Call:
## spca(x = pitprops, K = 9, para = rep(13, 9), type = "Gra
##
## 9 sparse PCs
## Pct. of exp. var. : 32.5 18.3 14.4  8.5  7.0  6.3  4.4
## Num. of non-zero loadings :  13 13 13 13 13 13 13 13 13
## Sparse loadings
##            PC1    PC2    PC3    PC4    PC5    PC6    PC7
## topdiam -0.404  0.218  0.207 -0.091  0.083 -0.120  0.11
## length  -0.406  0.186  0.235 -0.103  0.113 -0.163  0.07
## moist   -0.124  0.541 -0.141  0.078 -0.350  0.276  0.02
## testsg  -0.173  0.456 -0.352  0.055 -0.356  0.054 -0.07
```


Example 3 - Use Sparse PCA L1 penalty

Use correlation matrix, L1 penalty instead of number of element restriction.

```
out1<-spca(pitprops,K=6,type="Gram",sparse="penalty",
           trace=TRUE,para=c(0.06,0.16,0.1,0.5,0.5,0.5))
```

```
## iterations 10
## iterations 20
## iterations 30
## iterations 40
## iterations 50
## iterations 60
```


Example 3 - Comparison of para settings

```
out1  # Same as varnum = c(7,4,4,1,1,1)
```

```
##
## Call:
## spca(x = pitprops, K = 6, para = c(0.06, 0.16, 0.1, 0.5,
##     0.5), type = "Gram", sparse = "penalty", trace = TRU
##
## 6 sparse PCs
## Pct. of exp. var. : 28.0 14.0 13.3  7.4  6.8  6.2
## Num. of non-zero loadings :  7 4 4 1 1 1
## Sparse loadings
##            PC1    PC2    PC3 PC4 PC5 PC6
## topdiam -0.477  0.000  0.000   0   0   0
## length  -0.476  0.000  0.000   0   0   0
## moist    0.000  0.785  0.000   0   0   0
## testsg   0.000  0.619  0.000   0   0   0
## ovensg   0.177  0.000 -0.641   0   0   0
## ringtop  0.000  0.000 -0.589   0   0   0
```