Lab 6: Regression with Dimension Reduction Methods PCR and PLSR

Donggyun Kim
27008257

3/16/2018

Principal Components Regression (PCR)

```
library(ISLR)
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```
pcr_fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE, validation = "none")

# remove observations that have NAs in Salary
Hitters <- na.omit(Hitters)


X <- model.matrix(Salary ~ ., data = Hitters)

X <- X[, -1]
X <- scale(X)

y <- Hitters$Salary

V <- svd(X)$v

Z <- X %*% V

Z[1:3, ]
```

```
##                        [,1]       [,2]       [,3]       [,4]      [,5]
## -Alan Ashby   -0.009630358 -1.8669625 -1.2627377 -0.9337009 -1.107524
## -Alvin Davis   0.410650757  2.4247988  0.9074630 -0.2637096 -1.229687
## -Andre Dawson  3.460224766 -0.8243753 -0.5544124 -1.6136499  0.855856
##                     [,6]        [,7]        [,8]       [,9]       [,10]
## -Alan Ashby   -1.209666  0.06502176 -0.09806158 -0.2519428 -0.63588901
## -Alvin Davis  -1.823141 -0.35920809 -1.19710045 -0.3711125  0.01518571
## -Andre Dawson  1.026755  0.99748363  0.84264033  0.1970036  0.76902393
##                     [,11]       [,12]       [,13]       [,14]       [,15]
## -Alan Ashby   0.45846720 -0.61468829  0.47717836  0.38766763 -0.06347540
## -Alvin Davis  0.17590012 -0.08037596 -0.40271546 -0.01934883 -0.06502464
## -Andre Dawson 0.05553923 -0.10473855 -0.02156399 -0.37548104 -0.23384538
##                       [,16]       [,17]       [,18]        [,19]
## -Alan Ashby   -0.14451398 -0.08006351 -0.03806717 -0.019213448
## -Alvin Davis  -0.16075777 -0.04425909 -0.01480600 -0.003388059
## -Andre Dawson -0.01150363  0.21786749 -0.03775314 -0.066374190
```

```
pcr_fit$scores[1:3, ]
```

```
##                     Comp 1     Comp 2     Comp 3     Comp 4    Comp 5
## -Alan Ashby   -0.009630358 -1.8669625 -1.2627377 -0.9337009 -1.107524
## -Alvin Davis   0.410650757  2.4247988  0.9074630 -0.2637096 -1.229687
## -Andre Dawson  3.460224766 -0.8243753 -0.5544124 -1.6136499  0.855856
##                  Comp 6      Comp 7      Comp 8     Comp 9     Comp 10
## -Alan Ashby   -1.209666  0.06502176 -0.09806158 -0.2519428 -0.63588901
## -Alvin Davis  -1.823141 -0.35920809 -1.19710045 -0.3711125  0.01518571
## -Andre Dawson  1.026755  0.99748363  0.84264033  0.1970036  0.76902393
##                  Comp 11     Comp 12     Comp 13     Comp 14     Comp 15
## -Alan Ashby   0.45846720 -0.61468829  0.47717836  0.38766763 -0.06347540
## -Alvin Davis  0.17590012 -0.08037596 -0.40271546 -0.01934883 -0.06502464
## -Andre Dawson 0.05553923 -0.10473855 -0.02156399 -0.37548104 -0.23384538
##                   Comp 16     Comp 17     Comp 18      Comp 19
## -Alan Ashby   -0.14451398 -0.08006351 -0.03806717 -0.019213448
## -Alvin Davis  -0.16075777 -0.04425909 -0.01480600 -0.003388059
## -Andre Dawson -0.01150363  0.21786749 -0.03775314 -0.066374190
```

```
z1 <- Z[, 1]
b1 <- as.numeric(solve(crossprod(z1, z1)) %*% crossprod(z1, y))

yhat1 <- b1 * z1 + mean(y)

head(yhat1)
```

```
##       -Alan Ashby      -Alvin Davis     -Andre Dawson -Andres Galarraga
##          534.8996          579.6895          904.6869          263.8013
##  -Alfredo Griffin        -Al Newman
##          645.2411          112.5090
```

```
head(pcr_fit$fitted.values[, , 1])
```

```
##       -Alan Ashby      -Alvin Davis     -Andre Dawson -Andres Galarraga
##          534.8996          579.6895          904.6869          263.8013
##  -Alfredo Griffin        -Al Newman
##          645.2411          112.5090
```

```
b_pcr <- solve(crossprod(Z, Z)) %*% crossprod(Z, y)
yhat <- Z %*% b_pcr + mean(y)

head(yhat)
```

```
##                        [,1]
## -Alan Ashby        362.1361
## -Alvin Davis       712.6952
## -Andre Dawson     1171.3111
## -Andres Galarraga  556.7875
## -Alfredo Griffin   493.2515
## -Al Newman         247.3852
```

```
head(pcr_fit$fitted.values[, , 19])
```

```
##       -Alan Ashby      -Alvin Davis     -Andre Dawson -Andres Galarraga
##          362.1361          712.6952         1171.3111          556.7875
##  -Alfredo Griffin        -Al Newman
##          493.2515          247.3852
```

```r
V %*% b_pcr
```

```
##             [,1]
##  [1,] -291.64955
##  [2,]  338.47458
##  [3,]   37.92601
##  [4,]  -60.68796
##  [5,]  -27.04645
##  [6,]  135.33143
##  [7,]  -16.72519
##  [8,] -391.78420
##  [9,]   86.85289
## [10,]  -14.20876
## [11,]  481.66372
## [12,]  261.18691
## [13,] -214.30006
## [14,]   31.30834
## [15,]  -58.52543
## [16,]   78.91146
## [17,]   53.83493
## [18,]  -22.20311
## [19,]  -12.37236
```

```r
pcr_fit$coefficients[, , 19]
```

```
##       AtBat        Hits       HmRun        Runs         RBI       Walks
## -291.64955  338.47458   37.92601  -60.68796  -27.04645  135.33143
##       Years      CAtBat       CHits      CHmRun       CRuns        CRBI
##  -16.72519 -391.78420   86.85289  -14.20876  481.66372  261.18691
##      CWalks     LeagueN   DivisionW     PutOuts     Assists      Errors
## -214.30006   31.30834  -58.52543   78.91146   53.83493  -22.20311
## NewLeagueN
##  -12.37236
```

Partial Least Squares Regression

```r
pls_fit <- plsr(Salary ~ ., data = Hitters, scale = TRUE, validation = "none")
y <- Hitters$Salary
ybar <- mean(y)


w1 <- crossprod(X, scale(y))

w1 <- w1 / as.numeric(sqrt(crossprod(w1, w1)))
z1 <- X %*% w1

p1 <- crossprod(X, z1) / as.numeric(crossprod(z1, z1))


```

```
head(pls_fit$loading.weights[, 1])
```

```
##     AtBat      Hits     HmRun      Runs       RBI     Walks
## 0.2256137 0.2507049 0.1960424 0.2399514 0.2568671 0.2536725
```

```
head(w1)
```

```
##            [,1]
## AtBat 0.2256137
## Hits  0.2507049
## HmRun 0.1960424
## Runs  0.2399514
## RBI   0.2568671
## Walks 0.2536725
```

```
head(pls_fit$scores[, 1])
```

```
##       -Alan Ashby      -Alvin Davis     -Andre Dawson -Andres Galarraga
##        -0.1090169         0.6670947         3.4717021        -2.1298594
##  -Alfredo Griffin        -Al Newman
##         0.9770842        -4.0036686
```

```
head(z1)
```

```
##                         [,1]
## -Alan Ashby       -0.1090169
## -Alvin Davis       0.6670947
## -Andre Dawson      3.4717021
## -Andres Galarraga -2.1298594
## -Alfredo Griffin   0.9770842
## -Al Newman        -4.0036686
```

```
head(pls_fit$loadings[, 1])
```

```
##     AtBat      Hits     HmRun      Runs       RBI     Walks
## 0.2256185 0.2231972 0.2179161 0.2249696 0.2566359 0.2292001
```

```
head(p1)
```

```
##            [,1]
## AtBat 0.2256185
## Hits  0.2231972
## HmRun 0.2179161
## Runs  0.2249696
## RBI   0.2566359
## Walks 0.2292001
```

```
b1 <- crossprod(z1, y) / as.numeric(crossprod(z1, z1))
yhat1 <- as.numeric(b1) * z1

head(pls_fit$fitted.values[, , 1])
```

```
##       -Alan Ashby      -Alvin Davis     -Andre Dawson -Andres Galarraga
##         523.82552         609.97025         921.26845         299.52153
##  -Alfredo Griffin        -Al Newman
##         644.37762          91.53754
```

```
head(yhat1 + mean(y))
```

```
##                        [,1]
## -Alan Ashby       523.82552
## -Alvin Davis      609.97025
## -Andre Dawson     921.26845
## -Andres Galarraga 299.52153
## -Alfredo Griffin  644.37762
## -Al Newman         91.53754
```

```
Z <- matrix(0, nrow = nrow(X), ncol = 19)
W <- matrix(0, nrow = 19, ncol = 19)
P <- matrix(0, nrow = 19, ncol = 19)
b <- numeric(19)
Y_fitted <- matrix(0, nrow = nrow(X), ncol = 19)

for (i in 1:19) {
  w <- crossprod(X, y)
  w <- w / as.numeric(sqrt(crossprod(w, w)))
  W[, i] <- w

  z <- X %*% w
  Z[, i] <- z

  p <- crossprod(X, z) / as.numeric(crossprod(z, z))
  P[, i] <- p

  X <- X - tcrossprod(Z[, i], P[, i])

  d <- crossprod(z, y) / as.numeric(crossprod(z, z))
  b[i] <- as.numeric(d)

  Y_fitted[, i] <- as.numeric(d) * z

  y <- y - Y_fitted[, i]
}

Z[1:3, ]
```

```
##            [,1]        [,2]      [,3]       [,4]        [,5]       [,6]
## [1,] -0.1090169 -0.08794742  1.114665 -1.4059430 -0.61582496 -1.2285982
## [2,]  0.6670947  0.87856868 -1.020564  0.9638650  0.03072175  0.1496738
## [3,]  3.4717021  0.52704957  1.297566 -0.3869003  0.62786341  2.0306809
##            [,7]       [,8]       [,9]      [,10]      [,11]      [,12]
## [1,] -0.9846195  0.6415656 -0.2219565  0.3834706  0.2874275 0.18005133
## [2,] -0.5112383  1.1576725 -0.9826639 -0.3058391  0.3091935 0.08420352
## [3,]  0.7066416 -0.5968688  0.2445129  0.3800293 -0.3438737 0.23893800
##            [,13]       [,14]       [,15]       [,16]       [,17]
## [1,] -0.13397459 -0.23746116 -0.10427027  0.15340983 -0.45492154
## [2,]  0.07231222 -0.02991202 -0.38816681  0.03761008  0.02304197
## [3,] -0.02598121  0.19570921  0.07268355 -0.20918617 -0.31372447
##           [,18]       [,19]
## [1,] 0.21553840 0.019482401
## [2,] 0.09910835 0.003300504
## [3,] 0.15648633 0.067814672
```

```
pls_fit$scores[1:3, ]
```

```
##                   Comp 1      Comp 2    Comp 3     Comp 4      Comp 5
## -Alan Ashby   -0.1090169 -0.08794742  1.114665 -1.4059430 -0.61582496
## -Alvin Davis   0.6670947  0.87856868 -1.020564  0.9638650  0.03072175
## -Andre Dawson  3.4717021  0.52704957  1.297566 -0.3869003  0.62786341
##                   Comp 6     Comp 7     Comp 8     Comp 9    Comp 10
## -Alan Ashby   -1.2285982 -0.9846195  0.6415656 -0.2219565  0.3834706
## -Alvin Davis   0.1496738 -0.5112383  1.1576725 -0.9826639 -0.3058391
## -Andre Dawson  2.0306809  0.7066416 -0.5968688  0.2445129  0.3800293
##                  Comp 11    Comp 12     Comp 13     Comp 14     Comp 15
## -Alan Ashby    0.2874275 0.18005133 -0.13397459 -0.23746116 -0.10427027
## -Alvin Davis   0.3091935 0.08420352  0.07231222 -0.02991202 -0.38816681
## -Andre Dawson -0.3438737 0.23893800 -0.02598121  0.19570921  0.07268355
##                   Comp 16     Comp 17    Comp 18     Comp 19
## -Alan Ashby    0.15340983 -0.45492154 0.21553840 0.019482401
## -Alvin Davis   0.03761008  0.02304197 0.09910835 0.003300504
## -Andre Dawson -0.20918617 -0.31372447 0.15648633 0.067814672
```

```
W[1:3, ]
```

```
##           [,1]          [,2]       [,3]        [,4]        [,5]
## [1,] 0.2256137 -3.672182e-05 -0.4662631 -0.13465184 -0.25449655
## [2,] 0.2507049  2.088446e-01 -0.1975273  0.32998021  0.20009164
## [3,] 0.1960424 -1.660701e-01 -0.3460830 -0.07308201 -0.06499369
##             [,6]        [,7]        [,8]       [,9]       [,10]
## [1,] -0.30790899 -0.40858271 -0.22032972  0.1565591 -0.06007446
## [2,]  0.01903897  0.06626299  0.18260953  0.5336905  0.19950085
## [3,]  0.49714619  0.20531354  0.03713366 -0.1432544  0.34482004
##             [,11]      [,12]       [,13]      [,14]       [,15]
## [1,] -0.002828448  0.2395064  0.15459643  0.1714136 -0.06059125
## [2,]  0.046688250  0.2167412 -0.32158889 -0.3782950 -0.11773563
## [3,] -0.186704978 -0.1627907  0.06728287 -0.1183331  0.09961507
##            [,16]       [,17]      [,18]        [,19]
## [1,] -0.02625963  0.06619698 0.42614875 -0.096877565
## [2,]  0.12843150 -0.06699386 0.05012442  0.099119332
## [3,] -0.07922400  0.49737588 0.17614819 -0.006333045
```

```
pls_fit$loading.weights[1:3, ]
```

```
##          Comp 1        Comp 2     Comp 3      Comp 4      Comp 5
## AtBat 0.2256137 -3.672182e-05 -0.4662631 -0.13465184 -0.25449655
## Hits  0.2507049  2.088446e-01 -0.1975273  0.32998021  0.20009164
## HmRun 0.1960424 -1.660701e-01 -0.3460830 -0.07308201 -0.06499369
##            Comp 6      Comp 7      Comp 8     Comp 9     Comp 10
## AtBat -0.30790899 -0.40858271 -0.22032972  0.1565591 -0.06007446
## Hits   0.01903897  0.06626299  0.18260953  0.5336905  0.19950085
## HmRun  0.49714619  0.20531354  0.03713366 -0.1432544  0.34482004
##            Comp 11    Comp 12     Comp 13    Comp 14     Comp 15
## AtBat -0.002828448  0.2395064  0.15459643  0.1714136 -0.06059125
## Hits   0.046688250  0.2167412 -0.32158889 -0.3782950 -0.11773563
## HmRun -0.186704978 -0.1627907  0.06728287 -0.1183331  0.09961507
##           Comp 16     Comp 17    Comp 18      Comp 19
## AtBat -0.02625963  0.06619698 0.42614875 -0.096877565
## Hits   0.12843150 -0.06699386 0.05012442  0.099119332
## HmRun -0.07922400  0.49737588 0.17614819 -0.006333045
```

```
P[1:3, ]
```

```
##           [,1]       [,2]       [,3]        [,4]       [,5]       [,6]
## [1,] 0.2256185 0.34657361 -0.3955083  0.09853735  0.1088757 -0.0924689
## [2,] 0.2231972 0.35568227 -0.3709201  0.14664095  0.1776232 -0.0159006
## [3,] 0.2179161 0.09120089 -0.3076810 -0.01352982 -0.6516902  0.3888872
##            [,7]       [,8]       [,9]       [,10]       [,11]      [,12]
## [1,] -0.1988555 -0.3227867  0.1938575 -0.05658049 -0.15522380  0.1581362
## [2,] -0.1075591 -0.1666536  0.4098268  0.14182703 -0.09122190  0.3860062
## [3,]  0.1699668  0.1308836 -0.3573422  0.57545599 -0.08312296 -0.1982044
##            [,13]      [,14]       [,15]      [,16]       [,17]      [,18]
## [1,] 0.003754557  0.2275980 -0.04922531 -0.1126079 -0.11725032 0.43407764
## [2,] 0.011306019 -0.2691224 -0.17332456  0.2158193 -0.08857127 0.04201206
## [3,] 0.171414512 -0.2107030  0.13390554 -0.7280084  0.42154811 0.17666651
##             [,19]
## [1,] -0.096877565
## [2,]  0.099119332
## [3,] -0.006333045
```

```
pls_fit$loadings[1:3, ]
```

```
##          Comp 1     Comp 2     Comp 3      Comp 4     Comp 5     Comp 6
## AtBat 0.2256185 0.34657361 -0.3955083  0.09853735  0.1088757 -0.0924689
## Hits  0.2231972 0.35568227 -0.3709201  0.14664095  0.1776232 -0.0159006
## HmRun 0.2179161 0.09120089 -0.3076810 -0.01352982 -0.6516902  0.3888872
##           Comp 7     Comp 8     Comp 9     Comp 10     Comp 11    Comp 12
## AtBat -0.1988555 -0.3227867  0.1938575 -0.05658049 -0.15522380  0.1581362
## Hits  -0.1075591 -0.1666536  0.4098268  0.14182703 -0.09122190  0.3860062
## HmRun  0.1699668  0.1308836 -0.3573422  0.57545599 -0.08312296 -0.1982044
##           Comp 13    Comp 14     Comp 15    Comp 16     Comp 17    Comp 18
## AtBat 0.003754557  0.2275980 -0.04922531 -0.1126079 -0.11725032 0.43407764
## Hits  0.011306019 -0.2691224 -0.17332456  0.2158193 -0.08857127 0.04201206
## HmRun 0.171414512 -0.2107030  0.13390554 -0.7280084  0.42154811 0.17666651
##            Comp 19
## AtBat -0.096877565
## Hits   0.099119332
## HmRun -0.006333045
```