Clase 6 - Módulo 2: Introducción a la analítica

Mauricio Alejandro Mazo Lopera

Universidad Nacional de Colombia
Facultad de Ciencias
Escuela de Estadística
Medellín

Métodos de reducción de dimensionalidad

Este tipo de métodos busca transformar los predictores $X_1, X_2, \ldots, X_p$ y luego ajustar mínimos cuadrados con las variables transformadas.

Este tipo de métodos busca transformar los predictores $X_1, X_2, \ldots, X_p$ y luego ajustar mínimos cuadrados con las variables transformadas.

Definamos a $Z_1, Z_2, \ldots, Z_M$, con $M < p$, como combinaciones lineales de los p predictores originales, es decir,

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

para algunas constantes $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$, donde $m = 1, 2, \ldots, M$.

Luego, ajustamos un modelo de regresión lineal dado por

$$Y_i = \theta_0 + \theta_1 Z_{i1} + \theta_2 Z_{i2} + \cdots + \theta_M Z_{iM} + \epsilon_i$$

para $i = 1, \ldots, n$, utilizando mínimos cuadrados.

Una obervación importante es que:

$$\sum_{m=1}^{M} \theta_m Z_{iM}$$

Una obervación importante es que:

$$\sum_{m=1}^{M} \theta_m Z_{iM} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} X_{ij}$$

Una obervación importante es que:

$$\sum_{m=1}^{M} \theta_m Z_{iM} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} X_{ij} \quad = \quad \sum_{m=1}^{M} \sum_{j=1}^{p} \theta_m \phi_{jm} X_{ij}$$

Una obervación importante es que:

$$\sum_{m=1}^{M} \theta_m Z_{iM} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} X_{ij} = \sum_{m=1}^{M} \sum_{j=1}^{p} \theta_m \phi_{jm} X_{ij}$$

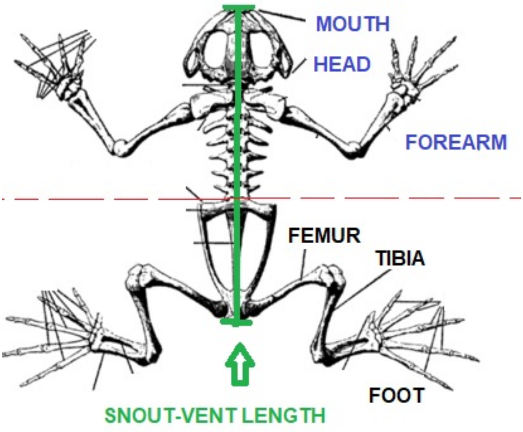
$$= \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} X_{ij}$$

Una obervación importante es que:

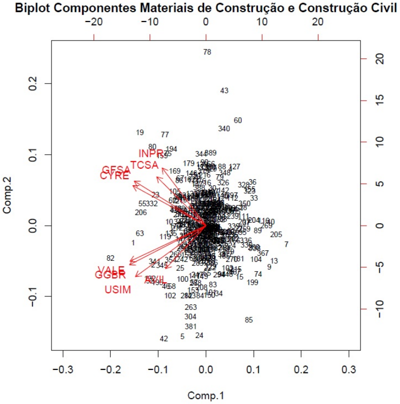
$$
\begin{aligned}
\sum_{m=1}^{M}\theta_m Z_{iM} = \sum_{m=1}^{M}\theta_m \sum_{j=1}^{p}\phi_{jm}X_{ij} &= \sum_{m=1}^{M}\sum_{j=1}^{p}\theta_m\phi_{jm}X_{ij} \\
&= \sum_{j=1}^{p}\sum_{m=1}^{M}\theta_m\phi_{jm}X_{ij} \\
&= \sum_{j=1}^{p}\beta_j X_{ij}
\end{aligned}
$$

Es decir, una vez se obtenga la reducción de dimensionalidad, es posible volver a las variables originales.

Ver **CLASE_6_ARCHIVO_R.R**



Ejemplo introductorio 2: Componentes principales



Setor	Ação	Código
Materiais de Construção	Vale	VALE
	Gerdau	GGBR
	Usiminas	USIM
	Aços Vill	AVIL
Construção Civil	Cyrela Realt	CYRE
	Inpar S/A	INPR
	Tecnisa	TCSA
	Gafisa	GFSA

Las componentes principales consideran modelar la mayor parte de la variabilidad dentro de la matriz de covariables $[\boldsymbol{X}_1 \ \boldsymbol{X}_2 \ \cdots \ \boldsymbol{X}_p]_{n\times p}$.

Las componentes principales consideran modelar la mayor parte de la variabilidad dentro de la matriz de covariables $[\boldsymbol{X}_1 \ \boldsymbol{X}_2 \ \cdots \ \boldsymbol{X}_p]_{n\times p}$.

Geométricamente, las componentes principales $Z_1, Z_2, \ldots, Z_M$ forman un hiperplano de dimensión $M < p$, de tal manera que la mayor parte de la variabilidad de los datos se refleje Z_1, la segunda variabilidad más alta en Z_2, la siguiente en Z_3 y así sucesivamente.

Las componentes principales consideran modelar la mayor parte de la variabilidad dentro de la matriz de covariables $[\boldsymbol{X}_1 \ \boldsymbol{X}_2 \ \cdots \ \boldsymbol{X}_p]_{n\times p}$.

Geométricamente, las componentes principales $Z_1, Z_2, \ldots, Z_M$ forman un hiperplano de dimensión $M < p$, de tal manera que la mayor parte de la variabilidad de los datos se refleje Z_1, la segunda variabilidad más alta en Z_2, la siguiente en Z_3 y así sucesivamente.

Además, $Z_1, Z_2, \ldots, Z_m$ se contruyen de tal manera que sean ortogonales y, por tanto, no correlacionadas.

El método de regresión con componentes principales (PCR, por sus siglas en inglés) consiste en:

1. Estandarizar los predictores $X_1, X_2, \ldots, X_p$
2. Obtener las M componentes principales $Z_1, Z_2, \ldots, Z_M$.
3. Usando mínimos cuadrados, ajustar un modelo de regresión lineal que tenga a Y como variable respuesta y a $Z_1, Z_2, \ldots, Z_M$ como variables explicativas.

NOTA: Generalmente M se obtiene con validación cruzada.

Mínimos cuadrados parciales (PLS por sus siglas en inglés) es un método de reducción de dimensionalidad que tiene en cuenta no solo los predictores $X_1, X_2, \ldots, X_p$ sino también la variable respuesta Y, para generar las nuevas variables $Z_1, Z_2, \ldots, Z_M$.

Mínimos cuadrados parciales (PLS por sus siglas en inglés) es un método de reducción de dimensionalidad que tiene en cuenta no solo los predictores $X_1, X_2, \ldots, X_p$ sino también la variable respuesta Y, para generar las nuevas variables $Z_1, Z_2, \ldots, Z_M$.

El método se describe paso a paso como:

1. Estandarizar los predictores $X_1, X_2, \ldots, X_p$.
2. Ajuste un modelo de regresión lineal simple entre Y y X_j, para cada $j = 1, \ldots, p$, y en cada caso obtenga ϕ_{j1} como el coeficiente de la regresión simple.

3. La primera combinación lineal sería:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p$$

4. Ajuste un modelo de regresión lineal simple entre X_j y Z_1, para cada $j = 1, \ldots, p$, y en cada caso obtenga los residuales de dicha regresión, r_{j1} hasta generar la matriz $[\boldsymbol{r}_{11} \ \boldsymbol{r}_{21} \ \cdots \ \boldsymbol{r}_{p1}]_{n \times p}$. Esta matriz se puede ver como la información restante que no fue capturada por Z_1.

5. Ajuste un modelo de regresión lineal simple entre Y y r_{j1}, para cada $j = 1, \ldots, p$, y en cada caso obtenga ϕ_{j2} como el coeficiente de la regresión simple.
6. La segunda combinación lineal sería:

$$Z_2 = \phi_{12} r_{11} + \phi_{22} r_{21} + \cdots + \phi_{p2} r_{p1}$$

7. Repita los pasos 4 (reemplazando los predictores X's por los residuos r's), 5 (generando unos nuevos residuos r's) y 6 hasta obtener las componentes PLS $Z_1, Z_2, \dots, Z_M$.
8. Usando mínimos cuadrados, ajustar un modelo de regresión lineal que tenga a Y como variable respuesta y a $Z_1, Z_2, \dots, Z_M$ como variables explicativas.

NOTA: Generalmente M se obtiene con validación cruzada.

Consideraciones en grandes dimensiones

- Decimos que estamos en una situación con grandes dimensiones si $p > n$ o también $p \approx n$.

Consideraciones en grandes dimensiones

- Decimos que estamos en una situación con grandes dimensiones si $p > n$ o también $p \approx n$.
- El primer problema que surge en situaciones con grandes dimensiones es el **sobreajuste**, con casos de residuos iguales a cero.

- Decimos que estamos en una situación con grandes dimensiones si $p > n$ o también $p \approx n$.

- El primer problema que surge en situaciones con grandes dimensiones es el **sobreajuste**, con casos de residuos iguales a cero.

- El segundo problema es que el R^2 (además del $R^2_{Ajustado}$) se eleva al punto de ser casi igual a 1 y el MSE (con el conjunto de entrenamiento) se hace casi igual a cero. Esto puede llevar a la conclución engañosa de que el modelo funciona muy bien.

- El tercer problema es que se hace muy difícil manejar la multicolinealidad en grandes dimensiones y técnicas como ridge, lasso, además de reducción de dimensionalidad son bastante útiles.

- El tercer problema es que se hace muy difícil manejar la multicolinealidad en grandes dimensiones y técnicas como ridge, lasso, además de reducción de dimensionalidad son bastante útiles.

- La recomendación general, en situaciones con grandes dimensiones, es ajustar un modelo con datos de entrenamiento y evaluar su desempeño con datos de prueba o mediante validación cruzada.

Trabajando con R: PCR

```r
require(ISLR)
require(pls)
Hitters<-na.omit(Hitters)
cedula<-123
set.seed(cedula)
pcr.fit<-pcr(Salary~.,data=Hitters,scale=TRUE,
             validation ="CV")
```

Trabajando con R: PCR

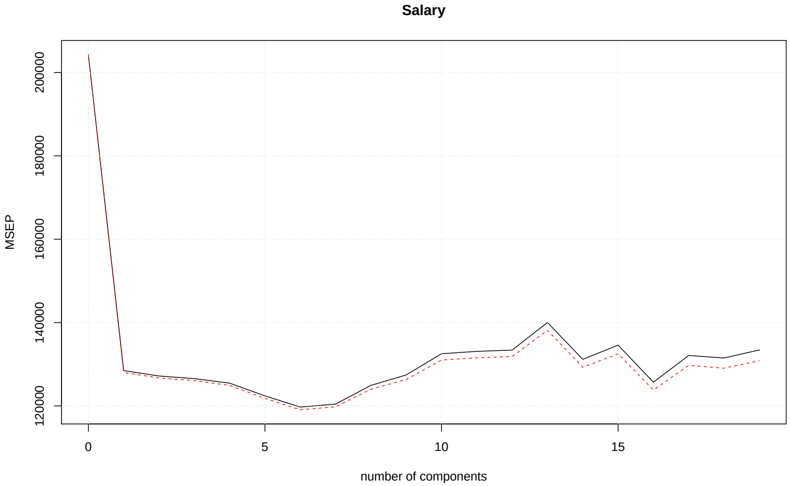
```r
names(pcr.fit)
```

```
##  [1] "coefficients"  "scores"        "loadings"      "Yloadings"
##  [5] "projection"    "Xmeans"        "Ymeans"        "fitted.values"
##  [9] "residuals"     "Xvar"          "Xtotvar"       "fit.time"
## [13] "ncomp"         "method"        "scale"         "validation"
## [17] "call"          "terms"         "model"
```

```r
summary(pcr.fit)
```

```
## Data:    X dimension: 263 19
##  Y dimension: 263 1
## Fit method: svdpc
## Number of components considered: 19
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV             452    358.4    356.6    355.8    354.2    349.9    346.0
## adjCV          452    357.8    356.0    355.1    353.4    349.0    345.1
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       347.1    353.4    357.0     364.0     364.8     365.2     374.2
## adjCV    346.1    352.1    355.5     361.9     362.7     363.1     371.7
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV        362.2     366.9     354.5     363.5     362.6     365.3
## adjCV     359.6     364.0     351.9     360.2     359.2     361.7
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         38.31    60.16    70.84    79.03    84.29    88.63    92.26    94.96
## Salary    40.63    41.58    42.17    43.22    44.90    46.48    46.69    46.75
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X         96.28     97.26     97.98     98.65     99.15     99.47     99.75
## Salary    46.86     47.76     47.82     47.85     48.10     50.40     50.55
##         16 comps  17 comps  18 comps  19 comps
## X          99.89     99.97     99.99    100.00
## Salary     53.01     53.85     54.61     54.61
```

```r
validationplot(pcr.fit ,val.type="MSEP")
grid()
```

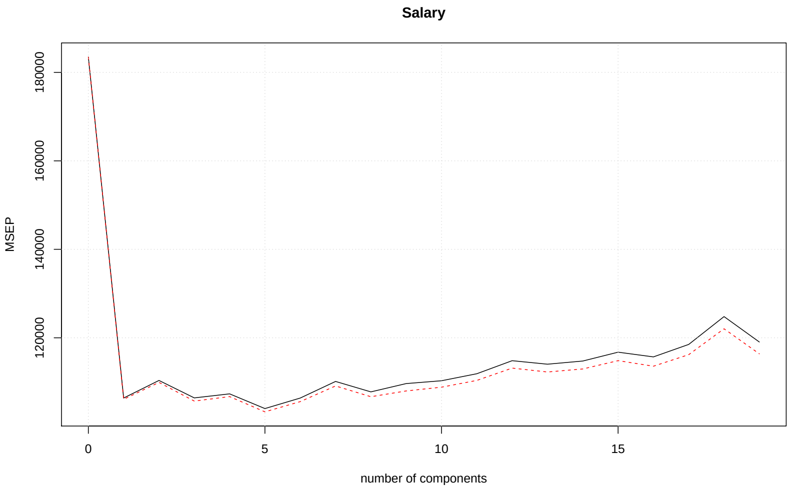


Trabajando con R: PCR

```
cedula<-1
set.seed(cedula)
train<-sample(1: nrow(Hitters), nrow(Hitters)/2)
test<- -train
set.seed(cedula)
pcr.fit.1<-pcr(Salary~.,data=Hitters,subset=train,
               scale =TRUE,validation ="CV")
```

Trabajando con R: PCR

```
validationplot(pcr.fit.1 ,val.type="MSEP")
grid()
```



Trabajando con R: PCR

```r
x<-model.matrix(Salary~.,Hitters)[,-1]
y<-Hitters$Salary
y.test<-y[test]
set.seed(cedula)
pcr.pred.1<-predict(pcr.fit.1 ,x[test,],ncomp =5)
mean((pcr.pred.1-y.test)^2)
```

```
## [1] 142811.8
```

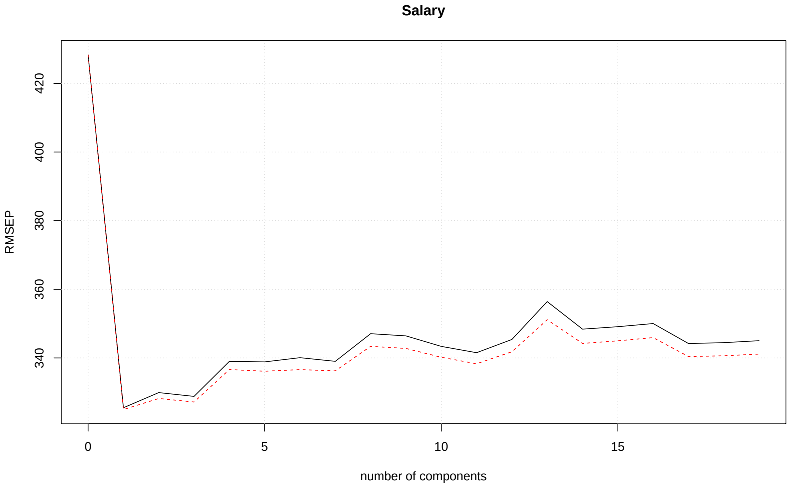
Trabajando con R: PCR

```r
pcr.fit.2<-pcr(y~x,scale =TRUE,ncomp =5)
summary(pcr.fit.2)
```

```
## Data:    X dimension: 263 19
##  Y dimension: 263 1
## Fit method: svdpc
## Number of components considered: 5
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps
## X    38.31    60.16    70.84    79.03    84.29
## y    40.63    41.58    42.17    43.22    44.90
```

```
cedula<-1
set.seed(cedula)
pls.fit<-plsr(Salary~.,data=Hitters,subset=train,
              scale=TRUE,validation ="CV")
```

```r
validationplot(pls.fit ,val.type="RMSEP")
grid()
```



Trabajando con R: PLS

```
pls.pred<-predict(pls.fit,x[test,],ncomp=1)
mean((pls.pred-y.test)^2)
```

```
## [1] 151995.3
```

Trabajando con R: PLS

```
set.seed(cedula)
pls.fit.1<-plsr(Salary~.,data=Hitters,
                scale=TRUE,ncomp=1)
summary(pls.fit.1)
```

```
## Data:    X dimension: 263 19
##  Y dimension: 263 1
## Fit method: kernelpls
## Number of components considered: 1
## TRAINING: % variance explained
##         1 comps
## X         38.08
## Salary    43.05
```

Actividad:

Considere la base de datos **College** de la librería **ISLR**. Se busca predecir el número de solicitudes recibidas usando las demás variables del conjunto de datos. Escriba **?College** para obtener más información.

a. Particione los dados en un conjunto de entrenamiento y otro de validación.
b. Ajuste un modelo PCR considerando los datos de entrenamiento con M (número de componentes principales) seleccionado a través de validación cruzada. ¿Cuál es el error de test obtenido para el M seleccionado?

Actividad:

c. Ajuste un modelo PLS considerando los datos de entrenamiento con M (número de componentes principales) seleccionado a través de validación cruzada. the training ¿Cuál es el error de test obtenido para el M seleccionado?
d. De los dos métodos anteriormente expuestos, ¿cuál muestra mejores resultados?