Lab 6: Regression with Dimension Reduction Methods PCR and PLSR

Prof. Gaston Sanchez

Stat 154, Fall 2017

Introduction

In this lab, you are going to write R code to implement Principal Component Regression (PCR), as well as Partial Least Squares Regression (PLSR). You will also be using the data `Hitters` from the package `"ISLR"`. More specifically, you will sue `Salary` as the response variable, and the rest of the variables in `Hitters` as the predictors.

Data `Hitters`

The data set `Hitters` is part of the R package `"ISLR"`.

```
str(Hitters, vec.len = 1)
```

```
## 'data.frame':    322 obs. of  20 variables:
##  $ AtBat    : int  293 315 ...
##  $ Hits     : int  66 81 ...
##  $ HmRun    : int  1 7 ...
##  $ Runs     : int  30 24 ...
##  $ RBI      : int  29 38 ...
##  $ Walks    : int  14 39 ...
##  $ Years    : int  1 14 ...
##  $ CAtBat   : int  293 3449 ...
##  $ CHits    : int  66 835 ...
##  $ CHmRun   : int  1 69 ...
##  $ CRuns    : int  30 321 ...
##  $ CRBI     : int  29 414 ...
##  $ CWalks   : int  14 375 ...
##  $ League   : Factor w/ 2 levels "A","N": 1 2 ...
##  $ Division : Factor w/ 2 levels "E","W": 1 2 ...
##  $ PutOuts  : int  446 632 ...
##  $ Assists  : int  33 43 ...
##  $ Errors   : int  20 10 ...
##  $ Salary   : num  NA 475 ...
##  $ NewLeague: Factor w/ 2 levels "A","N": 1 2 ...
```

Principal Components Regression (PCR)

Principal Components Regression can be performed with the function `pcr()` which is part of the package `"pls"`. The code below computes PCR for the regression of `Salary` on the rest of 19 predictors.

```
# principal component regression
pcr_fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE, validation = "none")
names(pcr_fit)
```

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

1) Start with PCA

You are going write R code in order to replicate the results of `pcr()`. Follow the list of steps shown below:

- Remove observations from `Hitters` that have missing values in `Salary`
- Use `model.matrix()` to create a design matrix based on the formula `"Salary ~ ."`
- Note that the generated model matrix includes a constant column for the intercept term. Do not use this column.
- The model matrix (without constant column) will be the matrix of responses. Standardize the model matrix of responses; this will be $\mathbf{X}$
- The variable `Salary` will be the response $\mathbf{y}$
- Use `svd()` to get the Singular Value Decomposition of $\mathbf{X} = \mathbf{UDV}^\mathsf{T}$
- Compute principal components $\mathbf{Z}$ from the standardized model matrix $\mathbf{X}$ and the eogenvectors in $\mathbf{V}$

$$\mathbf{Z} = \mathbf{XV}$$

- Confirm that your principal components match those of `pcr_fit$scores`

2) PC Regression on the first component

- Use the first PC$\mathbf{z_1}$ to compute the regression of $\mathbf{y}$ on $\mathbf{z_1}$. That is, obtain the first PCR coefficient b_1 given by:

$$b_1 = (\mathbf{z_1}^\mathsf{T}\mathbf{z_1})^{-1}\mathbf{z_1}^\mathsf{T}\mathbf{y}$$

- Compute the vector of predicted values $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = b_1\mathbf{z_1}$$

- Compare your computed $\hat{\mathbf{y}}$ against `pcr_fit$fitted.values[ , ,1]`, which is the fitted response using PC1 provided by `pcr()`. Add the average of y to your predicted value before comparison.

3) PC Regression on all PCs

- Compute the vector of PCR-coefficients $\mathbf{b}_{pcr}$ by regressing $\mathbf{y}$ on all principal components $\mathbf{Z}$:

$$\mathbf{b}_{pcr} = (\mathbf{Z}^\mathsf{T}\mathbf{Z})^{-1}\mathbf{Z}^\mathsf{T}\mathbf{y}$$

- Compute the vector of predicted values $\hat{\mathbf{y}}$ using all PCs:

$$\hat{\mathbf{y}} = \mathbf{Z}(\mathbf{Z}^\mathsf{T}\mathbf{Z})^{-1}\mathbf{Z}^\mathsf{T}\mathbf{y}$$
$$\hat{\mathbf{y}} = \mathbf{Z}\mathbf{b}_{pcr}$$

- Compare your computed $\hat{\mathbf{y}}$ against `pcr_fit$fitted.values[ , ,19]` and confirm that you have the same results as `pcr()`. Add the average of y to your predicted value before comparison.

4) PCR coefficients in terms of the predictor variables

`pcr()` returns regression coefficients—in terms of the predictors—for all possible regressions: with one PC, two PCs, three PCs, and so on, until the regression that uses all 19 PCs.

Consider the PC regression on the first PC $\mathbf{z_1}$. The PCR-coefficient is:

$$b_1 = (\mathbf{z_1}^\mathsf{T}\mathbf{z_1})^{-1}\mathbf{z_1}^\mathsf{T}\mathbf{y}$$

and the fitted $\hat{\mathbf{y}}$ is:

$$\hat{\mathbf{y}} = b_1\mathbf{z_1}$$

You can re-write the regression of PC1 in terms of the response variables as:

$$
\begin{aligned}
\hat{\mathbf{y}} &= b_1 \mathbf{z_1} \\
&= b_1 \mathbf{X} \mathbf{v_1} \\
&= \mathbf{X}(b_1 \mathbf{v_1}) \\
&= \mathbf{X} \mathbf{b_1^*}
\end{aligned}
$$

where:

- $\mathbf{v_1}$ is the loading associated to the first PC, that is, the first column of $\mathbf{V}$
- $\mathbf{b_1^*}$ is a vector of regression coefficients in terms of the predictors

In general, the PC regression coefficients can be expressed in terms of the predictors as:

$$
\mathbf{b_k^*} = \mathbf{V_k} \mathbf{D_k^{-1}} \mathbf{U_k^{\top}} \mathbf{y}
$$

where the index k indicates matrices associated to the first k components. More specifically, V_k is a matrix of the first k columns of V, U_k is a matrix of the first k columns of U, and D_k is a $k \times k$ diagonal matrix.

Your turn:

- Take your previously computed coefficient b_1 and calculate the associated vector of coefficients $\mathbf{b_1^*} = b_1 \mathbf{v_1}$. Confirm that your vector $\mathbf{b_1^*}$ matches that of `pcr_fit$coefficients[ , , 1]`
- Do the same for all possible sets of PCs, and verify your coefficients against the output of `pcr_fit$coefficients`.

The lab continues on the next page.

Partial Least Squares Regression

Below are the steps of the PLSR algorithm (in its "classic" version). Assume that the predictors in $\mathbf{X}$ and the response $\mathbf{y}$ are standardizedL mean = 0, variance 1.

Set $\mathbf{X_0} = \mathbf{X}$ and $\mathbf{y_0} = \mathbf{y}$
for $h = 1, 2, \ldots, r$ **do**
$\mathbf{w_h} = \mathbf{X_{h-1}^\top y_{h-1}}$
normalize weights: $\|\mathbf{w_h}\| = 1$
$\mathbf{z_h} = \mathbf{X_{h-1} w_h / w_h^\top w_h}$
$\mathbf{p_h} = \mathbf{X_{h-1}^\top z_h / z_h^\top z_h}$
$\mathbf{X_h} = \mathbf{X_{h-1}} - \mathbf{z_h p_h^\top}$
$b_h = \mathbf{y_{h-1}^\top z_h / z_h^\top z_h}$
$\mathbf{y_h} = \mathbf{y_{h-1}} - b_h \mathbf{z_h}$
end for

where r is the rank of $\mathbf{X}$

Your mission is to write R code that carries out PLS regression according to the steps shown above. Your code should contain the following objects:

- `components`: matrix of PLS components $\mathbf{Z}$
- `weights`: matrix of weights $\mathbf{W}$
- `loadings`: matrix of loadings $\mathbf{P}$
- `coefficients`: vector of regression coefficients $\mathbf{b}$
- `fitted`: matrix of fitted (predicted) values $\hat{\mathbf{Y}}$

The first steps are the same as with PCR:

- Remove observations from `Hitters` that have missing values in `Salary`
- Use `model.matrix()` to create a design matrix based on the formula `"Salary ~ ."`
- Note that the generated model matrix includes a constant column for the intercept term. Do not use this column.
- The model matrix (without constant column) will be the matrix of responses.
- Standardize the model matrix of responses; this will be $\mathbf{X}$
- The response `Salary` will be $\mathbf{y}$

Check your first PLS component

- Calculate $\mathbf{w_1}, \mathbf{z_1}$, and $\mathbf{p_1}$
- Compare your results with `pls_fit$loading.weights[,1]`, `pls_fit$scores[,1]`, `pls_fit$loadings[,1]`,
- Compare the first fitted $\hat{\mathbf{y}}$, i.e. regressing $\mathbf{y}$ on the first PLS component $\mathbf{z_1}$, and compare it with `pls_fit$fitted.values[,,1]`