

Lab 6: Regression with Dimension Reduction

Methods PCR and PLSR

Stat 154, Spring 2018

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 regress `Salary` on the rest of the variables in `Hitters`.

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 ...
```

1) Some Data Preprocessing

Follow the list of steps shown below:

- Remove observations from `Hitters` that have missing values in `Salary`. *Hint:* you may use `na.omit()` for this step.
- 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 predictors. Standardize this matrix; this will be \mathbf{X} .
- The variable `Salary` will be the response \mathbf{y} .

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 (without CV)
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"
```

2.1) Start with PCA

You are going to write R code in order to replicate the results of `pcr()`.

- Use `svd()` to get the Singular Value Decomposition of $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$
- Compute principal components \mathbf{Z} from the standardized model matrix \mathbf{X} and the eigenvectors in \mathbf{V}

$$\mathbf{Z} = \mathbf{X}\mathbf{V}$$

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

2.2) 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:

$$b_1 = (\mathbf{z}_1^\top \mathbf{z}_1)^{-1} \mathbf{z}_1^\top \mathbf{y}$$

- Compute a first vector of fitted 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.

2.3) PC Regression on all PCs

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

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

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

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{Z}(\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{y} \\ \hat{\mathbf{y}} &= \mathbf{Z} \mathbf{b}_{pcr} \end{aligned}$$

- 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.

2.4) 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 last regression that uses all PCs.

Consider the PC regression on the first PC \mathbf{z}_1 . The PCR-coefficient is:

$$b_1 = (\mathbf{z}_1^\top \mathbf{z}_1)^{-1} \mathbf{z}_1^\top \mathbf{y}$$

and the PCR 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 predictor 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:

$$\mathbf{b}_k^* = \mathbf{V}_k \mathbf{D}_k^{-1} \mathbf{U}_k^T \mathbf{y}$$

where the index k indicates matrices associated to the first k components. More specifically, \mathbf{V}_k is a matrix of the first k columns of \mathbf{V} , \mathbf{U}_k is a matrix of the first k columns of \mathbf{U} , and \mathbf{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.

3) Partial Least Squares Regression

Partial Least Squares Regression (PLSR) can be performed with the function `pls()` which is also part of the package "pls". The code below computes PLSR for the regression of `Salary` on the rest of 19 predictors.

```
# partial least squares regression (without CV)
pls_fit <- pls(Salary ~ ., data = Hitters, scale = TRUE, validation = "none")
names(pls_fit)
```

```
## [1] "coefficients"      "scores"            "loadings"
## [4] "loading.weights"   "Yscores"           "Yloadings"
## [7] "projection"        "Xmeans"            "Ymeans"
## [10] "fitted.values"     "residuals"         "Xvar"
## [13] "Xtotvar"           "fit.time"          "na.action"
## [16] "ncomp"             "method"            "scale"
## [19] "call"              "terms"             "model"
```

PLS Regression Algorithm

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 standardized: mean = 0, variance 1.

```
Set  $\mathbf{X}_0 = \mathbf{X}$  and  $\mathbf{y}_0 = \mathbf{y}$ 
for  $h = 1, 2, \dots, r$  do
   $\mathbf{w}_h = \mathbf{X}_{h-1}^\top \mathbf{y}_{h-1}$ 
  normalize weights:  $\|\mathbf{w}_h\| = 1$ 
   $\mathbf{z}_h = \mathbf{X}_{h-1} \mathbf{w}_h / \mathbf{w}_h^\top \mathbf{w}_h$ 
   $\mathbf{p}_h = \mathbf{X}_{h-1}^\top \mathbf{z}_h / \mathbf{z}_h^\top \mathbf{z}_h$ 
   $\mathbf{X}_h = \mathbf{X}_{h-1} - \mathbf{z}_h \mathbf{p}_h^\top$ 
   $b_h = \mathbf{y}_{h-1}^\top \mathbf{z}_h / \mathbf{z}_h^\top \mathbf{z}_h$ 
   $\mathbf{y}_h = \mathbf{y}_{h-1} - b_h \mathbf{z}_h$ 
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. Use the same model matrix of responses \mathbf{X} , and the response vector \mathbf{y} employed in PCR.

3.1) First iteration in PLSR

- Follow the steps of the above algorithm to 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]`,
- Obtain the first fitted $\hat{\mathbf{y}}$, i.e. regressing \mathbf{y} on the first PLS component \mathbf{z}_1 . Add the average of y to fitted values $\hat{\mathbf{y}}$, and compare it with `pls_fit$fitted.values[, ,1]`.

3.2) Implement the PLSR algorithm

Write code to implement the classic PLS algorithm. Create the following objects to store the various results:

- **components**: matrix of PLS components \mathbf{Z}
- **weights**: matrix of weights \mathbf{W}
- **loadings**: matrix of loadings \mathbf{P}
- **coefficients**: matrix of regression coefficients \mathbf{B}
- **fitted**: matrix of fitted values $\hat{\mathbf{Y}}$