HW5: PCR, PLSR, and Model Validation

Stat 154, Fall 2019

Problem 1) Properties of PLS Regression

Recall that in Partial Least Squares Regression (PLSR), we obtain uncorrelated componentes $\mathbf{z_1}, \mathbf{z_2}, \dots, \mathbf{z_h}$ that summarize variability in predictors $\mathbf{X}$ as well as capture variation in the response $\mathbf{y}$.

Here's the "classic" algorithm for PLS Regression (assume standardized data)

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

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

1.a) Orthogonal PLS compoenents

One of the properties of the original PLS regression algorithm is that it produces orthogonal components. Show that any two components $\mathbf{z_h}$ and $\mathbf{z_l}$ $(h \neq l)$ are indeed orthogonal, that is:

$$\mathbf{z_h^\top z_l} = 0 \quad \text{for} \quad h \neq l$$

Hint: The demonstration is done by recursivity.

1.b) Weights and loadings are collinear

Prove that the inner product of weights and loadings is: $\mathbf{w}_h^\top \mathbf{p}_h = 1$

1.c) Weights and predictor-residuals are orthogonal

Another property involves $\mathbf{w_h^\top X_l^\top} = 0, l \geq h$. Prove this property for the case $l = h$.

Problem 2) Bias of Regression Coefficients in PCR

Assume that there is a linear relationship between the response and the predictors: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $\mathbb{E}(\boldsymbol{\varepsilon}) = \mathbf{0}$. PCR can be used to reduce the dimensionality of the regression by dropping those dimensions that contribute to the collinearity problem. The estimated regression coefficients $\mathbf{b}_Z^{(k)}$ for the k principal components ($\mathbf{Z}_k$) are given by:

$$\mathbf{b}_Z^{(k)} = (\mathbf{Z}_k^\mathsf{T}\mathbf{Z}_k)^{-1}\mathbf{Z}_k^\mathsf{T}\mathbf{y}$$

It is usual to transform the PCR coefficients $\mathbf{b}_Z^{(k)}$ into coefficients $\hat{\boldsymbol{\beta}}_Z^{(k)}$ of the original input variables:

$$\hat{\boldsymbol{\beta}}_Z^{(k)} = \mathbf{V}_k\mathbf{b}_Z^{(k)} \quad \Longrightarrow \quad \hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_Z^{(k)}$$

where $\mathbf{V}_k$ is the matrix of k loadings.

2.a) Bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$

Show that the theoretical bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$ is:

$$E\left[\hat{\boldsymbol{\beta}}_Z^{(k)} - \boldsymbol{\beta}\right] = (\mathbf{V}_k\boldsymbol{\Lambda}_k^{-1}\mathbf{V}_k^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X} - \mathbf{I})\boldsymbol{\beta}$$

- $\mathbf{V}_k$ is the matrix of k loadings associated to the k PCs (i.e. eigenvectors).
- $\boldsymbol{\Lambda}_k$ is the diagonal matrix of k PC variances (i.e. eigenvalues)

2.b) Bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$ when $k = p$

Suppose that $\mathbf{X}$ is of full column-rank p. What is the bias of $\hat{\boldsymbol{\beta}}_Z^{(k)}$ if all PCs are used in PCR, that is, if $k = p$?

Problem 3) Bike Sharing: Fitting Models

In this problem you will work with the *Bike Sharing Data Set* (courtesy of Hadi Fanaee) that contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information. Visit the following website for more information:

https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

All the data files are in a zip file. This file contains a `Readme.txt` file, and two CSV data files: `day.csv` and `hour.csv`. One option to download the zip file, and then extract its contents in R, is with the functions `download.file()` and `unzip()`—of course, you can use other approaches to get the data.

```r
# =======================================================
# do NOT include the code in this chunk in your Rmd file
# =======================================================
# assembling url
uci <- 'https://archive.ics.uci.edu/ml/machine-learning-databases/'
zip <- '00275/Bike-Sharing-Dataset.zip'
url <- paste0(uci, zip)

# download zip file, and unzip its contents in your working directory
download.file(url, 'Bike-Sharing-Dataset.zip')
unzip('Bike-Sharing-Dataset.zip')
```

Motivation

OLS Regression analysis can be used to answer the following question:

> *What is the predicted number of riders for a Sunday that is expected to be sunny and cool (around 75 F degrees)?*

We may try to predict ridership, given the weather conditions, day of the week, time of the year and so on. This type of prediction could actually be useful for bike-sharing services that may need to prepare for days with high demand.

Using the daily data set (i.e. file `day.csv`), the idea is to fit various models of different complexity and examine some of their outputs to assess their predictive performance.

3.1) Data Processing (not graded)

- Import the file `day.csv`
- Subset those observations of `yr == 0` (i.e. year 2011).
- Create a new binary variable `clearday` by selecting `weathersit == 1`. In other words, values with `weathersit == 1` should be assigned to one, while the rest should be assigned to zero.
- *Note*: temperature, `temp`, is scaled to have values in a range [0, 1]

$$\frac{\text{Celsius temperature} - \text{minimum}}{\text{maximum} - \text{minimum}}$$

where the min and max were -8 and 39, respectively.

3.2) Fitting models

Use `lm()` or a similar function to fit the following regression models to predict ridership—variable `registered`—on the data of year 2011. Display the fitted models (e.g. estimated coefficients).

Model 1

$$\texttt{registered} = \beta_0 + \beta_1 \texttt{temp} + \varepsilon$$

Model 2

$$\texttt{registered} = \beta_0 + \beta_1 \texttt{temp} + \beta_2 \texttt{temp}^2 + \varepsilon$$

Model 3

$$\texttt{registered} = \beta_0 + \beta_1 \texttt{temp} + \beta_2 \texttt{temp}^2 + \beta_3 \texttt{workingday} + \varepsilon$$

Model 4

$$\texttt{registered} = \beta_0 + \beta_1 \texttt{temp} + \beta_2 \texttt{temp}^2 + \beta_3 \texttt{workingday} + \beta_4 \texttt{clearday} + \varepsilon$$

Model 5

$$\texttt{registered} = \beta_0 + \beta_1 \texttt{temp} + \beta_2 \texttt{temp}^2 + \beta_3 \texttt{workingday} + \beta_4 \texttt{clearday} + \beta_5 (\texttt{temp} \times \texttt{workingday}) + \varepsilon$$

3.3) Basic Models Comparison

In order to assess the performance of each model, we will use the (in-sample) Mean Squared Error (MSE). Recall that the MSE is simply the average of the residuals sum of squares:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

Right now the model performance will be assessed on all the in-sample data (observations in year 2011), which of course will provide a very optimistic measure of error. Later on you will apply validation approaches to better assess the performance of the models.

- Compute the MSE for each fitted model.
- Plot the (in-sample) MSEs against the number of regressor terms in each model. We will treat the number of regressor terms as the "complexity" of the models.
- Which model has the smallest in-sample MSE?
- Do you observe any trend in the graph?

Problem 4) Bike Sharing: Hold-Out method

As we saw in lecture, the most straightforward approach to assess the performance of a model is to split the dataset into two parts: one for training and one for testing. This approach is commonly known as the *holdout method*. A common split is 80-20: use 80% of the data to train the model and 20% to test the model.

- Select 20% of your dataset as holdout. You should use simple random sampling. Before generating the sample, specify a random seed for reproducibility purposes—e.g. using `set.seed()`.
- For each of the regression models in *Problem 3*, train on the remaining 80% of the data, predict the holdout data, and compute the **test MSE**.
- Plot the training and test MSEs, and identify which model gives the lowest holdout test MSE.

Problem 5) Bike Sharing: Cross-validation

Another validation approach is *Cross Validation*, which is an alternative to the holdout method.

5.1) 10-Folds and test MSEs

Create 10 folds to perform cross validation to estimate the prediction error. Specifically,

- For each fold,
  - For each regression model,
    - Train the model based on all observations except the ones in the fold.
    - Predict the observation in the fold.
    - Compute the fold MSE—i.e. MSE_{fold}.
- Compute the cross-validation MSE—MSE_{CV}—for each regression model.

5.2) CV-MSE

Recall that the CV-MSE is defined as:

$$\text{MSE}_{CV} = \frac{1}{\text{number of folds}} \sum_{\text{fold}} \text{MSE}_{\text{fold}}$$

which is simply the average MSE over the folds.

1. Calculate the CV-MSE for each model.
2. Plot the CV-MSEs against the order of the regression models.
3. Which model gives the lowest CV-MSEs? Is it reasonable? Why or why not?

Problem 6) Bike Sharing: Bootstrap

Bootstrap is another popular approach for model assessment. The idea is to iterate the following procedure many times: first, sample with replacement from the data (this serves as the training set); second, train the model on the sampled data; third, test the model on the data that is not in the sample and compute the performance metric, typically the MSE for regression problems. Finally, compute the average MSE over all the iterations, and this is referred as the *bootstrap MSE*.

Do the following tasks with 200 bootstrap samples.

1. Plot the bootstrap MSEs against each regression model.

2. Which model gives the lowest bootstrap MSEs? Is it reasonable? Why or why not?

3. For each model, compute the SD of the 200 MSEs. Plot the SD against the model complexity. What do you notice?

4. For each model, make a histogram of the 200 MSEs. What do you notice?

5. Based on what you saw in 3 and 4, do you think the bootstrap estimate is reliable? Why or why not?