Statistical Machine Learning GR5241
Spring 2021

Homework 2

Due: Monday, February 22nd by 11:59pm

Homework submission: Please submit your homework as a pdf on Canvas.

Problem 1 (Training Error vs. Test Error, ESL 2.9)

In this problem, we want to use the least squares estimator to illustrate the point that the trainning error is generally an underestimate of the prediction error (or test error).

Consider a linear regression model with p parameters,

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \text{ where } \epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

We fit the model by least squares to a set of trainning data $(x_1, y_1), \ldots, (x_N, y_N)$ drawn independently from a population. Let $\hat{\beta}$ be the least squares estimate obtained from the training data. Suppose we have some test data $(\tilde{x}_1, \tilde{y}_1), \cdots, (\tilde{x}_M, \tilde{y}_M)$ $(N \geq M > p)$ drawn at random from the same population as the training data. If $R_{tr}(\beta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \beta^T x_i)^2$ and $R_{te}(\beta) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}_i - \beta^T \tilde{x}_i)^2$, prove that

$$\mathbb{E}[R_{tr}(\hat{\beta})] \leq \mathbb{E}[R_{te}(\hat{\beta})],$$

where the expectations are over all that is random in each expression.

Problem 2 (kNN-Regression)

Suppose that the true relationship between x and y is given by

$$Y_i = f(X_i) + \epsilon_i, \quad i = 1, 2, \ldots, n,$$

where ϵ_i is white noise (independent) with $\mathsf{E}[\epsilon_i] = 0$ and $\mathsf{Var}\left[\epsilon\right]_i = \sigma^2$. Consider the generic kNN regression model

$$\hat{f}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i, \tag{1}$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points x, in the training sample. Assuming x is **not random**, derive an expression of *prediction error* using squared error loss, i.e., compute and simplify

$$E[(Y - \hat{f}_k(x_0))^2 | X = x_0].$$

Note that x_0 is a single query point (or test point).

Problem 3 (K-Means Clustering Proof)

Consider the traditional k-means clustering algorithm and let $d(x_i, x_j)$ be squared Euclidean distance. Prove the following identity:

$$\frac{1}{2} \sum_{C(i)=k} \sum_{C(j)=\ell} d(x_i, x_j) = |N_\ell| \sum_{C(i)=k} d(x_i, \overline{x}_k),$$

More specifically, prove

$$\frac{1}{|N_k|} \sum_{C(i)=k} \sum_{C(j)=k} \sum_{m=1}^{p} (x_{im} - x_{jm})^2 = 2 \sum_{C(i)=k} \sum_{m=1}^{p} (x_{im} - \bar{x}_{km})^2$$

where

$$\bar{x}_{km} = \frac{1}{|N_k|} \sum_{C(i)=k} x_{im}.$$

Note:

- The syntax $C(i) = k$, or equivalently $\{i : C(i) = k\}$, represents the set of all indices (or observations) having cluster assignment k. The symbol $|N_k|$ represents the number of elements in set $\{i : C(i) = k\}$. For example, suppose our data matrix consists of $n = 10$ observations and each observation (or row) is assigned to $K = 3$ clusters

  $$\begin{aligned} \text{Case:} \quad & \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ \text{Cluster Assignment:} \quad & \{3, 1, 3, 1, 2, 2, 2, 1, 3, 2\} \end{aligned}$$

  Then

  $$\begin{aligned} \{i : C(i) = 1\} &= \{2, 4, 8\} & |N_1| = 3 \\ \{i : C(i) = 2\} &= \{5, 6, 7, 10\} & |N_2| = 4 \\ \{i : C(i) = 3\} &= \{1, 3, 9\} & |N_3| = 3 \end{aligned}$$

Problem 4 (PCA, LDA and Logistic Regression)

The zipcode data are high dimensional, and hence linear discriminant analysis suffers from high variance. Using the training and test data for the 3s, 5s, and 8s, compare the following procedures:

1. LDA on the original 256 dimensional space.
2. LDA on the leading 49 principle components of the features.
3. Multiple linear logistic regression (multinomial regression) using the same filtered data as in the previous question.

Note:

- For all the above exercises, use R or Python functions to perform the PCA, LDA and multinomial regression, i.e., there is no need to manually code these procedures.
- **For all the above exercises**, compare the procedures with respect to training and test misclassification error. You need to report both training and test misclassification error in your submission.
- When evaluating the test error based on the **filtered** trained model, don't forget to first **project** the test features onto space generated by the leading 49 principle components. In `R` use the `predict()` function.
- The data of interest is already split into a training and testing set.

Problem 5 (PCA: Finance)

1. For each of the 30 stocks in the Dow Jones Industrial Average, download the closing prices for every trading day from January 1, 2020 to January 1, 2021. You can use http://finance.yahoo.com to find the data. To download the prices, for example for symbol `AAPL`, we use the R package `quantmod`. The `R` code is as the following:

   ```
   library(quantmod)
   data <- getSymbols("AAPL", auto.assign = F, from ="2020-01-01", to = "2021-01-01")
   ```

   Please find a way to download data for the 30 stocks efficiently.

2. Perform a PCA on the un-scalled closing prices and create the biplot. Do you see any structure in the biplot, perhaps in terms of the types of stocks? How about the screeplot – how many important components seem to be in the data?

3. Repeat part 2 using the scaled variables.

4. Use the closing prices to calculate the return for each stock, and repeat Part 3 on the return data. In looking at the screeplot, what does this tell you about the 30 stocks in the DJIA? If each stock were fluctuating up and down randomly and independent of all the other stocks, what would you expect the screeplot to look like?

Problem 6 (PCA on Digits)

In this problem students will run PCA on the zipcode training data from Problem 4. This problem will help solidify how to interpret PCA in a high dimensional setting.

1. Open and run the file "`Problem_6_Setup.Rmd`" to establish your data matrix X and *unobserved* test cases. Note that the data matrix X has dimension 1753×256 after removing three cases.

2. Run PCA on the data matrix X. Produce a graphic showing the cumulative explained variance as a function of the number of PC's (or features). Identify how many PC's yield 90% explained variance.

3. Display the first 16 principle components as images. Try to combine all 16 images in a single plot.

4. Approximate the test cases `ConstructCase_1`, `ConstructCase_2` and `ConstructCase_3` by projecting these cases into the subspace (eigenspace) generated by $d = 3, 58$ and 256 principal components. Your final result should be represented as 9 images, 3 per test case.