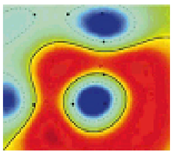
Technische Universität Berlin



Problem Set 3: Kernel Ridge Regression, Cross-Validation

Report Machine Learning Lab Course
Fachgebiet Maschinelles Lernen
Prof. Dr. Klaus-Robert Müller
Fakultät IV Elektrotechnik und Informatik
Technische Universität Berlin

submitted by
Budi Yanto

Instructor: Daniel Bartz
Felix Brockherde

Matrikelnummer: 308819
Email: budiyanto@mailbox.tu-berlin.de

Contents

List of Figures

Chapter 1

Implementation

This chapter explains the implementation of some algorithms that are used in this problem set. It includes: *PCA*, γ*-index* and *LLE*.

1.1 Assignment 1: Cross Validation

In this assignment the function *pca* has to be implemented, which receives a *d x n* matrix *X* and the number of components *m* as parameters, and returns the principal components as well as the projected data points in a *m x n* matrix *Z*. The principle components should be returned as a *d x d* matrix *U* and a *1 x d* vector *D*. The vector *D* contains the principal values, sorted in descending order ($D_1 \geq D_2 ...$), whereas the matrix *U* contains the principal directions, which corresponds to the sorted principle values.

The implemented function was tested on the test data and passed the test. Following steps are performed in the implementation of the function:

1. Substract *X* from its mean.
2. Calculate the covariance matrix from the zero-mean *X*.
3. Calculate the eigenvalues and eigenvectors from the covariance matrix.
4. Sort the eigenvalues and eigenvectors in descending order.
5. Form the feature vectors by taking only the first *m* eigenvectors.
6. Project the zero-mean *X* to the feature vectors.
7. Return the projected data points, principal directions and principal values as *Z*, *U* and *D*.

1.2 Assignment 2: Kernel Ridge Regression

The task in this assignment is to implement the γ-index which can be used to detect outliers in data set. In their paper, **?** formulate the formula to calculate the γ-index for each data point as follows:

$$\gamma(x) = \frac{1}{k} \sum_{j=1}^{k} \|x - z_j(x)\| \tag{1.1}$$

where x is a data point, k is the number of nearest neighbours, and $z_1(x), ..., z_k(x)$ are the k nearest neighbours of x.

The implemented function receives a *d x n* matrix *X* containing the data points and a scalar *k* representing the number of neighbours as parameters. It returns the γ-index for each data point in a *1 x n* vector *y*.

The function was tested on the test data and passed the test. Following steps are performed in the implementation of the function:

1. Implement a helper function *distmat* that calculates the distances from the data points to each other and return the distances as a matrix.

2. Get the distance matrix using the function *distmat* mentioned above.

3. Sort the distance matrix in ascending order.

4. Take only the *k*-nearest data points as neighbours for each data point.

5. Calculate the mean from the distances of the *k*-nearest neighbours and set it as the γ-index.

6. Return the calculated γ-index as a *1 x d* vector *y*.

Chapter 2

Application

In this chapter, we are trying to apply the algorithms that are described and implemented in Chapter **??** to various datasets: *usps*, *banana*, *fishbowl*, *swissroll* and *flatroll*.

2.1 Assignment 3: ROC Curve

This assignment asks us to apply *PCA* to the *usps* data set and visualizing the results. The *usps* data set consists of 2007 images with the dimension of *16 x 16*. The images are hand-written digits of zero to nine, which can be viewed as classes. Firstly, i separate the data set according to each digit into ten classes and then applied *PCA* to each class. The *PCA* was applied to the original data set and noisy data set.

Figure 2.1: Principal components of usps original data set for class 0 - 5

2.2 Assignment 4: Kernel Ridge Regression

In this assignment, the γ-index method is utilized to detect outliers and applied it to the *banana* data set. The positive class of the data set is used as *inliers*, to which the negative class is added as outliers. The γ-index is then used to detect outliers with contamination rates of 1%, 5%, 10% and 25% relative to the positive class. Figure **??** shows the complete original data set, both positive and negative class, whereas Figure **??** shows the contaminated data set.

Figure 2.2: Both positive and negative of banana data set, including the mean of both classes

Figure 2.3: Contaminated banana data set with contamination rate of 1%, 5%, 10% and 25%

There are three methods that should be used to detect the outliers: (a) the γ-index with $k = 3$, (b) the γ-index with $k = 10$ and (c) the distance to the mean for each data point. All of the methods are then applied to the four contamination rates mentioned above. After that, the *AUC* (area under the *ROC*) should be calculated. Figure **??** shows the boxplots that visualize the distribution of the *AUC* values.

Figure 2.4: Boxplots visualizing the distribution of the *AUC* values

The boxplots show that the method using the distance to the mean for each data point performed quite bad, while both of the γ-index methods performed very well, especially for the data set with lower contamination rates. The γ-index with $k = 10$ performed slightly better than the γ-index with $k = 3$.