Statistical Learning 2024: Lab 1

Random Multivariate Data generation

Nearest Centroid Classifier (NCC) and k-Nearest Neighbours (k-NN)

Goal:

Get familiar with data generation in two dimensions, with basic classifiers such as the Nearest Centroid Classifiers and k-NN

Materials:

Use files contained in the folder Example Basic Classifiers:

- BasicClassifiers.Rmd
- helper_functions.R
- md_helper_functions.R

Please locate all of them in your working directory.

These materials are part of the course "Introduction to Data Science" by Prof. Ian Carmichael, Dept. of Statistics, UC Berkeley, USA.

Procedure and specific tasks

Load BasicClassifiers.Rmd into R-studio.

Task 1

Read the text and execute the chunks till you reach the section:

- **Spherical Gaussian point clouds**

Read the help of function rmvnorm.

Modify the code and generate new data such that:

- Positive class is centred at coordinates (3,2)
- Negative class is centred at coordinates (-3,-2)
- You generate 225 examples per class.
- Standard deviation for each dimension is 1.5.

Task 2

Read the text and execute the chunks till you reach the section:

Skewed Gaussian point clouds

Calculate the correlation coefficient for the covariance matrix. Modify, the covariance matrix such that the correlation coefficient is: i) -0.8, ii) -0.5, iii) 0, ii) +0.5, iii) +0.8. Do so while keeping the standard deviation of each class and dimension equal to 1.5. Generate the data in each case and observe the changes.

Task 3

Read the text and execute the chunks till you reach the section:

Gaussian X

Modify the code such that the positive class has a correlation coefficient of 0.6 and the negative class is -0.6. Observe the changes.

Task 4

Read the text and execute the chunks till you reach the section:

Boston Cream

Modify the code such that the length for the negative class goes from 0.6 to 1.4 and the positive class from 1 to 3. Observe the changes.

Task 5

Read the text and execute the chunks till you reach the section:

Nearest Centroid classifier: Pragmatically

Change the position of the test point to (-0.8, -0.3) and see the new classification.

Are the observed means equal to the true values? Why? In case you have few examples, will this impact the performance of the classifier? What is the reason for this change in performance? Modify the distance, in a way that computes the sum of the absolute differences in both dimensions. Does the classifier still work properly? Look at the decision boundary: is this a linear classifier?

Task 6

Read the text and execute the chunks till you reach the section:

Nearest Centroid classifier: Toy examples

Modify the code to generate new test data from the same distribution as the training data and calculate the error rate accordingly. You will need to investigate the code of the function: *get_nearest_centroid_predictions* to use the function properly.

Task 7

Read the text and execute the chunks till you reach the section:

K-nearest neighbors: Toy examples

Modify the code to experiment with different values of k. Observe the effect on the decision boundaries. For the case of the Gaussian mixture model, optimize manually the value of k that minimizes the error in 200 test points generated with the same distribution.

Upload to Aula the code with the required modifications. Comment on the changes made!