Machine Learning 2024 spring

HW2: Classification

Deadline: 2024.4.24

1 Part I.

In HW2_training.csv, there is a group of professional baseball players from four different teams. Now we analyze the Performance-Rating of "offensive" and the "defensive" for every player, as shown in Figure 1, where the training data with different colors represent the players from Team 0, Team 1, Team 2, and Team 3, respectively. Noted that there are 300, 250, 350, and 400 players from Team 0, Team 1, Team 2, and Team 3, respectively. In HW2_testing,csv, the testing data are from Team 0, Team 1, Team 2, and Team 3, respectively. Note that there are 200, 300, 150, and 100 players, respectively.

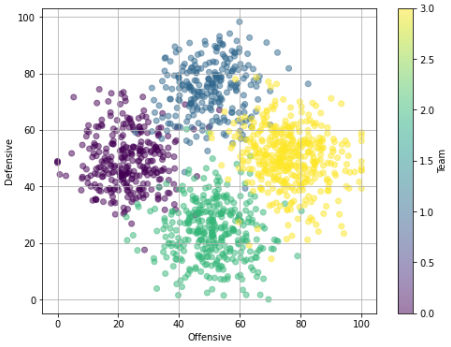

Figure 1:

1. Please implement the algorithms of the generative model and the discriminative model to classify the data and plot the corresponding decision boundaries by training data, like Figure 2. You can use any toolbox or function to plot the decision boundaries for your prediction results. (**Please note that Figure 2 is for illustrative purposes only and may not represent the correct answer.**)

2. For the generative model and the discriminative model, compute the confusion matrix and the prediction accuracy of training data and testing data as follows:

$$\text{Confusion Matrix} = \begin{pmatrix} T_0 & F_{01} & F_{02} & F_{03} \\ F_{10} & T_1 & F_{12} & F_{13} \\ F_{20} & F_{21} & T_2 & F_{23} \\ F_{30} & F_{31} & F_{32} & T_3 \end{pmatrix} \tag{1}$$

   - T_i: The number of samples in class i that are actually predicted as i by the model.
   - F_{ij}: The number of samples in class i that are actually predicted as j by the model.

$$\text{Accuracy} = \frac{\sum_{i=0}^{3} T_i}{\sum_{i=0}^{3}(T_i + \sum_{j=0}^{3} F_{ij})} \times 100\% \tag{2}$$

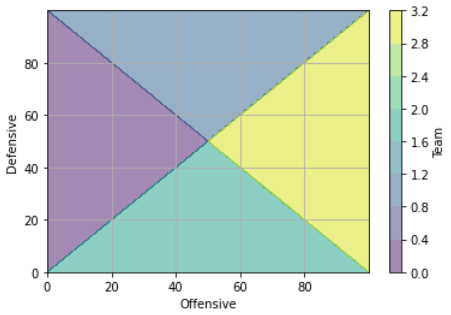

Figure 2:

Key: For the generative model, you can assume the data comes from four Gaussian distribution with the same covariance matrix but different mean vectors, and then try to find the parameters of these Gaussian distribution according to the given data. On the contrary, for the discriminative model, you do not have any assumption of the distribution for the given data. Instead, you define lots of basis functions and just try to learn the corresponding weights for every basis function according to the given data.

Basis Function:
Please use the following set of basis function $\boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \phi_2(\mathbf{x})]^T$

$$\phi_0(\mathbf{x}) = 1,\ \phi_1(\mathbf{x}) = x_1,\ \phi_2(\mathbf{x}) = x_2$$

- x_1: the offensive value of one data point
- x_2: the defensive value of one data point

Since the closed-form solution of the optimal weights does not exist, you need to sequentially adapt the weights based on the gradient descent method or Newton-Raphson method.

2 Part II.

According to the overall Performance-Rating of "offensive" and the "defensive" for each team in Figure 1, we classify Team 0 and Team 3 to be class A, Team 1 to be class B, Team 2 to be class C as illustrated in Figure 3. (Please note that Figure 3 is for illustrative purposes.)

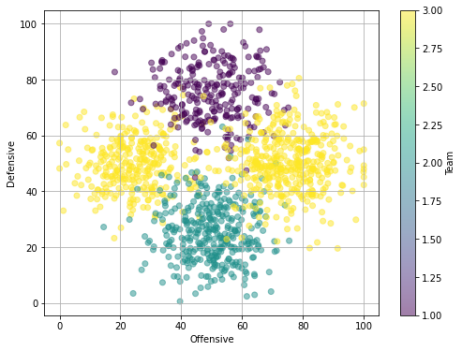

Figure 3:

Please classify the data in Figure 3 for the same requirement in **Part I** (You need to revise the data in HW2_training.csv and HW2_testing.csv: change the team label of the Team 0 from 0 to 3).

Homework Rules and Grading Policy

Homework will be graded by:

1. Report 50%
   - The correctness of your classification.
   - Your discussion of what you observe in the classification problems. For example,
     - What is the difference between the generative model and the discriminative model?
     - How do you implement the code of the generative model and the discriminative model?
     - And any other topics you would like to discuss.
2. Demo 50%
   - Writing a program using C++ can earn you 10 points more in the demo segment compared to using Python. In other words, if you write the program with correctness in C++, you can get 100 in the demo. However, if you write the program with correctness in Python, you can get only 90 in the demo. If the program is incorrect in either language, you will receive a score of 0.
3. The final score is determined by an equal weight of 50% for both the demo and report. In order to achieve an average score of around B+, the report will be graded around 60.

Upload:

- [Web] E3
- [File Name] hw2_StudentID.zip (ex: hw2_1234567.zip)
  The file should include your code and report.

Remind:

1. Your report in the format of .pdf.
2. Deadline:
   If you have a late submission by 1 to 7 days, you will only get 70% of the score. We DO NOT accept any late submissions after 7 days after the deadline.
3. We encourage open discussion to ensure program correctness, but plagiarism is strictly prohibited. Violators will receive a score of 0.

Demonstration:

1. In the demo part, you need to train your program based on the training data of HW2_demo_training.csv and test your program with the additional test data prepared by TAs, HW2_demo_testing.csv. You will be required to evaluate the accuracy and confusion of this dataset.

2. **You need to demonstrate the training accuracy and the confusion matrix of training data HW2_demo_training.csv, the testing accuracy and the confusion matrix of additional testing data HW2_demo_testing.csv.**

3. Your submitted program should be executable directly (with the dataset and program placed in the same folder).