ISYE 6740 – OAN Due: November 11, 2019

Homework 5

Instructor: Yao Xie

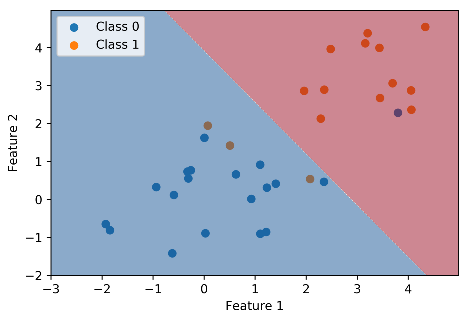
1 SVM and Neural Networks

Divorce classification/prediction

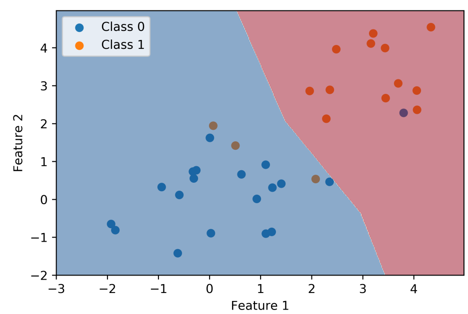
(a) We use linear kernel to train SVM. We set $C = 0.01$ and the initial step size for neural network is 0.1. The training algorithm for neural network is stochastic gradient descent (solver = 'sgd').

The testing accuracy for SVM and simple neural network are 0.9411 and 0.9412, respectively. They achieve a very close performance due to the data set is relatively easy. In general, we expect neural networks will be more powerful in modeling and prediction.

(b) The testing accuracy for SVM and simple neural network are 0.8823 and 0.8824, respectively. The decision boundaries are shown in the following figures. We note that the margin yielded



SVM decision boundary



Neural network decision boundary

by both the SVM and neural network is very close. Therefore, it is reasonable to see a similar performance of neural network and SVM.

Handwritten digits classification

The initial step size for neural network training is now 0.001. The resulting testing accuracy for SVM and simple neural network are 0.9874 and 9899, respectively.

2 AdaBoost

1. We denote y_i as the label for each X_i and $y_1 = 1$ for label $+$ and $y_i = -1$ for label $-$.

   Step 1. We set $D_1(i) = 1/8$ and pick the first decision stump as

$$h_1(X) = \begin{cases} 1, & \text{if the } x\text{-axis of } X \text{ is smaller than } -0.25 \\ -1, & \text{otherwise} \end{cases}.$$

   The misclassified points are 5 and 6. Then we find $\epsilon_1 = \sum_{i=1}^{8} D_1(i)\mathbb{1}\{y_i \neq h_1(X_i)\} = \frac{1}{4}$. Accordingly, we compute $\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\epsilon_1}{\epsilon_1}\right) \approx 0.5493$. The final ingredient is the normalization constant Z_1. We have

$$Z_1 = \sum_{i=1}^{8} D_1(i)e^{-\alpha_1 y_i h_1(X_i)} = \frac{6}{8}\exp(-\alpha_1) + \frac{2}{8}\exp(\alpha_1) = \frac{\sqrt{3}}{2}.$$

Step 2. We set

$$D_2(i) = \begin{cases} \frac{1}{12} & \text{if } i = 1,2,3,4,7,8 \\ \frac{1}{4} & \text{if } i = 5,6 \end{cases}.$$

We pick the second decision stump as

$$h_2(X) = \begin{cases} 1, & \text{if the } x\text{-axis of } X \text{ is greater than } 0.75 \\ -1, & \text{otherwise} \end{cases}.$$

The misclassified points are 1 and 2. Then we find $\epsilon_2 = D_2(1) + D_2(2) = \frac{1}{6}$. Accordingly, we compute $\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\epsilon_2}{\epsilon_2}\right) \approx 0.8047$. The normalization constant Z_2 can be found as

$$Z_2 = \sum_{i=1}^{8} D_2(i) e^{-\alpha_2 y_i h_2(X_i)} = \frac{4}{12}\exp(-\alpha_2) + \frac{2}{4}\exp(-\alpha_2) + \frac{2}{12}\exp(\alpha_2) = \frac{\sqrt{5}}{3}.$$

Step 3. We set

$$D_3(i) = \begin{cases} \frac{1}{4} & \text{if } i = 1,2 \\ \frac{1}{20} & \text{if } i = 3,4,7,8 \\ \frac{3}{20} & \text{if } i = 5,6 \end{cases}.$$

We pick the third decision stump as

$$h_3(X) = \begin{cases} 1, & \text{if the } y\text{-axis of } X \text{ is smaller than } 0.75 \\ -1, & \text{otherwise} \end{cases}.$$

The misclassified points are 7 and 8. Then we find $\epsilon_3 = D_3(7) + D_3(8) = \frac{1}{10}$. Accordingly, we compute $\alpha_3 = \frac{1}{2}\ln\left(\frac{1-\epsilon_3}{\epsilon_3}\right) \approx 1.1513$. The normalization constant Z_3 can be found as

$$Z_3 = \sum_{i=1}^{8} D_3(i) e^{-\alpha_3 y_i h_3(X_i)} = \frac{2}{4}\exp(-\alpha_3) + \frac{2}{20}\exp(-\alpha_3) + \frac{2}{20}\exp(\alpha_3) + \frac{6}{20}\exp(-\alpha_3) = 0.6.$$

The final classifier is

$$H_{\text{final}}(X) = \text{sign}\left(0.5493\text{sign}(-X_x - 0.25) + 0.8047\text{sign}(X_x - 0.75) + 1.1513\text{sign}(-X_y + 0.75)\right),$$

where X_x and X_y denotes the x-coordinate and y-coordinate of X, respectively, and $\text{sign}(\cdot)$ is defined as

$$\text{sign}(a) = \begin{cases} 1 & a > 0 \\ 0 & a = 0 \\ -1 & a < 0 \end{cases}.$$

2. Using $H_{\text{final}}(X)$ to classify $X_1, \ldots, X_8$, we find that the classifier making no error on points 1 to 8. Thus, the training error is 0. AdaBoost can be viewed as an ensemble of single decision stumps. Therefore, we expect AdaBoost to outperform single decision stumps.

Grading Rubric

Please loosely grade these two questions due to their open ended nature. We will give credits majorly depending on the efforts the student has shown (we are not grading based on whether they provide the correct number). The following items list possible deductions for each problem.
Note: The following list is far from comprehensive. In particular, for Problem 1, the learning rate can be different and the training and testing error will vary. For Problem 2, there are different ways of choosing weak learners (decision stumps), and the corresponding classifier and training error can also vary.

Problem 1

Missing a consistent and coherent written report (5 points); Unexpected training and testing errors — should be over 90% since the task is pretty easy (5 points).

Problem 2

Incorrect implementation of the Adaboost algorithm (5 points).