417 hw5

alex.huang

April 2020

1

If we have a initial data set of 80 percent positive and 20 percent negative, we will learn a 0-level decision tree of only predict positive. However, as in the re-weight process we increase the weight of negative points and decrease the weight of positive ones, and keeping the in sample error at 0.5, we will have a data set with half positive predictions and half negative predictions. However, when we are trying to learn a 0-level decision tree on the new data set, we cannot decide what to predict, because we do not have a majority. Suppose this time we randomly choose to predict positive or negative, we are breaking the assumption that weak learners are better than random guessing, and we cannot re-weight next time because no possible $\gamma > 1$ exists. If we do try to randomly choose a decition tree and re-weight, that is meaningless because we are not gaining more insights about the data, just doing random guessing multiple times

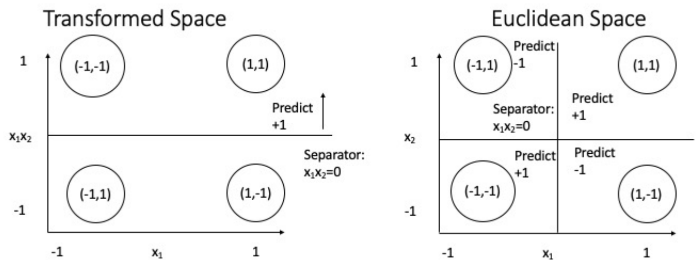
Adaboost does not work with depth 0 decision tree because if given any data set of half positive, half negative, the hypothesis set is too simple to allow us to learn a decision tree that has in sample error < 0.5. That breaks the assumption we have around adaboost. Even if we allow in sample error=0.5, we don't really "learn" anything in this case.

2

If $x = 3.2$, we have the nearest three neighbours as: $(3, 5), (3, 8), (2, 11)$, as we can see the nearest x values to 3.2 we have are $3, 3, 2$. To get our prediction, we simply average them to get $(5 + 8 + 11)/3 = 8$. Our answer is $y = 8$.

3

Please see the diagram below. The margin is 1, because we know all the data points are 1 distance from the separator $x_1x_2 = 0$.



4

In this problem we need to compute the squared Euclidean distance, that is, if $\overrightarrow{a_i} = \{a_{i1}, a_{i2}, \ldots\}$ and $\overrightarrow{a_j} = \{a_{j1}, a_{j2}, \ldots\}$ the squared Euclidean distance between the two points is

$$(a_{i1} - a_{j1})^2 + (a_{i2} - a_{j2})^2 + \ldots = a_{i1}^2 + a_{j1}^2 - 2a_{i1}a_{j1} + a_{i2}^2 + a_{j2}^2 - 2a_{i2}a_{j2} + \ldots$$

$$= (a_{i1}^2 + a_{i2}^2 + \ldots) + (a_{j1}^2 + a_{j2}^2 + \ldots) - 2(a_{i1}a_{j1} + a_{i2}a_{j2} + \ldots) = \overrightarrow{a_i}^T \overrightarrow{a_i} + \overrightarrow{a_j}^T \overrightarrow{a_j} - 2\overrightarrow{a_i}^T \overrightarrow{a_j}$$

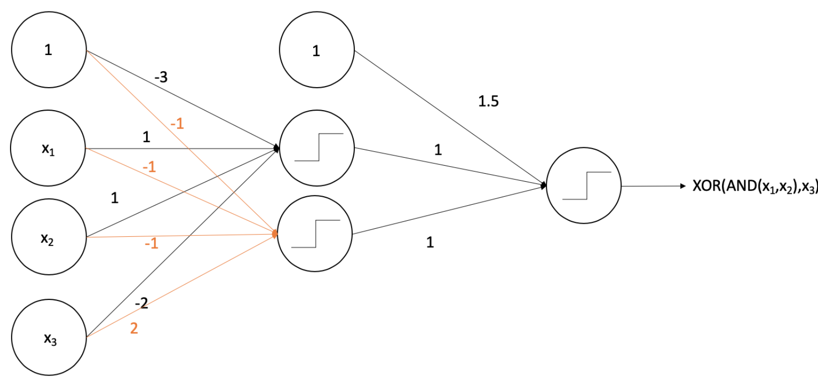
By definition, the Kernel function $K_\Phi(\overrightarrow{x_i}, \overrightarrow{x_j}) = \Phi(\overrightarrow{x_i})^T \Phi(\overrightarrow{x_j})$. Therefore, if we want to calculate the squared Euclidean distance between $\Phi(\overrightarrow{x_i}), \Phi(\overrightarrow{x_j})$ (denoted as $D(\Phi(\overrightarrow{x_i}), \Phi(\overrightarrow{x_j}))$, we can calculate it as:

$$D(\Phi(\overrightarrow{x_i}), \Phi(\overrightarrow{x_j})) = K_\Phi(\Phi(\overrightarrow{x_i}), \Phi(\overrightarrow{x_i})) + K_\Phi(\Phi(\overrightarrow{x_j}), \Phi(\overrightarrow{x_j})) - 2K_\Phi(\Phi(\overrightarrow{x_i}), \Phi(\overrightarrow{x_j}))$$

5

See picture below:

In this part I designed the first layer to combine $(x_1 AND x_2)$ and x_3. The next layer is the same as the right layer in XOR.



6

I have chosen the first article, the one published by MIT news.

6.1

The article analyzed the gender classification error rates for three commercial facial analysis systems. The error rates increase dramatically on darker-skinned/female users. The findings question the validity on how companies train and evaluate neuron networks.

6.2

The major issue raised is that the testing error (which indicates out of sample error) is significantly different among the groups researchers divided according to skin color and gender. The possible problem identified is the sampling bias: the training data is over 77 percent male and over 83 percent white. Therefore, white male has more training data, resulting in a better learned neuron network and better experience vis-a-vis female dark-skinned users.

6.3

There are a number of methods to prevent such bias. The trainer of the neuron network can use the training data set in which female and dark-skinned user data are sufficiently represented to avoid bias. Before selling programs in the market, developers should test the program's error rates across different demographic groups to make sure bias are spotted beforehand. Also, perhaps it is not possible to train one neuron network to recognize people of all races, because they have different facial characteristics. It might be possible to design different neuron networks (or train one network differently) for different ethnicity so each neuron network are trained on more uniform, patternable data to decrease error for each ethnic group.