CS 536 : Perceptrons

16:198:536

In the usual way, we need data that we can fit and analyze using perceptrons. Consider generating data points $(\underline{X}, Y)$ in the following way:

- For $i = 1, \ldots, k-1$, let $X_i \sim N(0,1)$ (i.e., each X_i is an i.i.d. standard normal)
- For $i = k$, generate X_k in the following way: let $D \sim \text{Exp}(1)$, and for a parameter $\epsilon > 0$ take

$$X_k = \begin{cases} (\epsilon + D) & \text{with probability } 1/2 \\ -(\epsilon + D) & \text{with probability } 1/2. \end{cases} \tag{1}$$

The effect of this is that while $X_1, \ldots, X_{k-1}$ are i.i.d. standard normals, X_k is distributed randomly with some gap (of size 2ϵ) around $X_k = 0$. We can then classify each point according to the following:

$$Y = \begin{cases} +1 & \text{if } X_k > 0 \\ -1 & \text{if } X_k < 0. \end{cases} \tag{2}$$

We see that the class of each data point is determined entirely by the value of the X_k feature.

1) Show that there is a perceptron that correctly classifies this data. Is this perceptron unique? What is the 'best' perceptron for this data set, theoretically?

2) We want to consider the problem of learning perceptrons from data sets. Generate a set of data of size $m = 100$ with $k = 20$, $\epsilon = 1$.

   – Implement the perceptron learning algorithm. This data is separable, so the algorithm will terminate. How does the output perceptron compare to your theoretical answer in the previous problem?

3) For any given data set, there may be multiple separators with multiple margins - but for our data set, we can effectively control the size of the margin with the parameter ϵ - the bigger this value, the bigger the margin of our separator.

   – For $m = 100$, $k = 20$, generate a data set for a given value of ϵ and run the learning algorithm to completion. Plot, as a function of $\epsilon \in [0, 1]$, the average or typical number of steps the algorithm needs to terminate. Characterize the dependence.

4) One of the nice properties of the perceptron learning algorithm (and perceptrons generally) is that learning the weight vector $\underline{w}$ and bias value b is typically independent of the ambient dimension. To see this, consider the following experiment:

   – Fixing $m = 100, \epsilon = 1$, consider generating a data set on k features and running the learning algorithm on it. Plot, as a function k (for $k = 2, \ldots, 40$), the typical number of steps to learn a perceptron on a data set of this size. How does the number of steps vary with k? Repeat for $m = 1000$.

5) As shown in class, the perceptron learning algorithm always terminates in finite time - if there is a separator. Consider generating non-separable data in the following way: generate each $X_1, \ldots, X_k$ as i.i.d. standard normals $N(0,1)$. Define Y by

$$Y = \begin{cases} +1 & \text{if } \sum_{i=1}^{k} X_i^2 \geq k \\ -1 & \text{else.} \end{cases} \tag{3}$$

For data defined in this way, there is no universally applicable linear separator.

For $k = 2$, $m = 100$, generate a data set that is not linearly separable. (How can you verify this?) Then run the perceptron learning algorithm. What does the progression of weight vectors and bias values look like over time? If there is no separator, this will never terminate - is there any condition or heuristic you could use to determine whether or not to terminate the algorithm and declare no separator found?