CS229 Lecture notes

Andrew Ng

1 The perceptron and large margin classifiers

In this final set of notes on learning theory, we will introduce a different model of machine learning. Specifically, we have so far been considering **batch learning** settings in which we are first given a training set to learn with, and our hypothesis h is then evaluated on separate test data. In this set of notes, we will consider the **online learning** setting in which the algorithm has to make predictions continuously even while it's learning.

In this setting, the learning algorithm is given a sequence of examples $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(m)}, y^{(m)})$ in order. Specifically, the algorithm first sees $x^{(1)}$ and is asked to predict what it thinks $y^{(1)}$ is. After making its prediction, the true value of $y^{(1)}$ is revealed to the algorithm (and the algorithm may use this information to perform some learning). The algorithm is then shown $x^{(2)}$ and again asked to make a prediction, after which $y^{(2)}$ is revealed, and it may again perform some more learning. This proceeds until we reach $(x^{(m)}, y^{(m)})$. In the online learning setting, we are interested in the total number of errors made by the algorithm during this process. Thus, it models applications in which the algorithm has to make predictions even while it's still learning.

We will give a bound on the online learning error of the perceptron algorithm. To make our subsequent derivations easier, we will use the notational convention of denoting the class labels by $y =\in \{-1, 1\}$.

Recall that the perceptron algorithm has parameters $\theta \in \mathbb{R}^{n+1}$, and makes its predictions according to

$$h_\theta(x) = g(\theta^T x) \tag{1}$$

where

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0. \end{cases}$$

Also, given a training example (x, y), the perceptron learning rule updates the parameters as follows. If $h_\theta(x) = y$, then it makes no change to the parameters. Otherwise, it performs the update[1]

$$\theta := \theta + yx.$$

The following theorem gives a bound on the online learning error of the perceptron algorithm, when it is run as an online algorithm that performs an update each time it gets an example wrong. Note that the bound below on the number of errors does not have an explicit dependence on the number of examples m in the sequence, or on the dimension n of the inputs (!).

Theorem (Block, 1962, and Novikoff, 1962). Let a sequence of examples $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(m)}, y^{(m)})$ be given. Suppose that $||x^{(i)}|| \leq D$ for all i, and further that there exists a unit-length vector u ($||u||_2 = 1$) such that $y^{(i)} \cdot (u^T x^{(i)}) \geq \gamma$ for all examples in the sequence (i.e., $u^T x^{(i)} \geq \gamma$ if $y^{(i)} = 1$, and $u^T x^{(i)} \leq -\gamma$ if $y^{(i)} = -1$, so that u separates the data with a margin of at least γ). Then the total number of mistakes that the perceptron algorithm makes on this sequence is at most $(D/\gamma)^2$.

Proof. The perceptron updates its weights only on those examples on which it makes a mistake. Let $\theta^{(k)}$ be the weights that were being used when it made its k-th mistake. So, $\theta^{(1)} = \vec{0}$ (since the weights are initialized to zero), and if the k-th mistake was on the example $(x^{(i)}, y^{(i)})$, then $g((x^{(i)})^T \theta^{(k)}) \neq y^{(i)}$, which implies that

$$(x^{(i)})^T \theta^{(k)} y^{(i)} \leq 0. \tag{2}$$

Also, from the perceptron learning rule, we would have that $\theta^{(k+1)} = \theta^{(k)} + y^{(i)} x^{(i)}$.

We then have

Okay, this is what it means: θ^{k} is a aritmetic sequence. Let's denote it by a[k].

$$\begin{aligned}(\theta^{(k+1)})^T u &= (\theta^{(k)})^T u + y^{(i)} (x^{(i)})^T u \\ &\geq (\theta^{(k)})^T u + \gamma\end{aligned}$$

A arithmetic sequence says a[k+1] = a[1] + k*d, so a[k+1] > k*d, our d in this case is γ.

By a straightforward inductive argument, implies that

This is where confuses me, what is this straightforward inductive argument? Nvm I know it know.

$$(\theta^{(k+1)})^T u \geq k\gamma. \tag{3}$$

[1] This looks slightly different from the update rule we had written down earlier in the quarter because here we have changed the labels to be $y \in \{-1, 1\}$. Also, the learning rate parameter α was dropped. The only effect of the learning rate is to scale all the parameters θ by some fixed constant, which does not affect the behavior of the perceptron.

Also, we have that

$$\begin{aligned} ||\theta^{(k+1)}||^2 &= ||\theta^{(k)} + y^{(i)}x^{(i)}||^2 \\ &= ||\theta^{(k)}||^2 + ||x^{(i)}||^2 + 2y^{(i)}(x^{(i)})^T\theta^{(i)} \\ &\leq ||\theta^{(k)}||^2 + ||x^{(i)}||^2 \\ &\leq ||\theta^{(k)}||^2 + D^2 \end{aligned} \tag{4}$$

The third step above used Equation (2). Moreover, again by applying a straightfoward inductive argument, we see that (4) implies

$$||\theta^{(k+1)}||^2 \leq kD^2. \tag{5}$$

Putting together (3) and (4) we find that

$$\begin{aligned} \sqrt{k}D &\geq ||\theta^{(k+1)}|| \\ &\geq (\theta^{(k+1)})^T u \\ &\geq k\gamma. \end{aligned}$$

The second inequality above follows from the fact that u is a unit-length vector (and $z^T u = ||z|| \cdot ||u|| \cos\phi \leq ||z|| \cdot ||u||$, where ϕ is the angle between z and u). Our result implies that $k \leq (D/\gamma)^2$. Hence, if the perceptron made a k-th mistake, then $k \leq (D/\gamma)^2$. □