CS269: Machine Learning Theory
Lecture 7: Perceptron Algorithm
October 18, 2010

Lecturer: Jennifer Wortman Vaughan
Scribe: Shankar Garikapati and Akshay Wadia

In this lecture, we consider the problem of learning the class of *linear separators* in the online learning framework. Recall from the previous lecture that an n-dimensional linear separator through the origin can be represented by an n-dimensional vector $\mathbf{u}$. For any vector $\mathbf{x}$, the label of $\mathbf{x}$ is $+1$ if $\mathbf{u} \cdot \mathbf{x} \geq 0$, and -1 otherwise. In this lecture, we will study the Perceptron algorithm and analyze its mistake bound.

NOTATION. Vectors are represented by lower case bold letters like $\mathbf{u}, \mathbf{w}, \mathbf{x}$, etc., while scalars are normal lower case letters. The inner product between two vectors $\mathbf{u}$ and $\mathbf{x}$ is denoted by $\mathbf{u} \cdot \mathbf{x}$. $\|\mathbf{u}\|$ represents the length of the vector $\mathbf{u}$.

1 Perceptron Algorithm

Before describing the Perceptron algorithm, we review the notion of *margin*. We have the following from the previous lecture.

Definition 1. *Given a linear separator* $\mathbf{u}$ *the* margin γ_t *of* $\mathbf{x_t}$ *with label* $y_t \in \{+1, -1\}$ *is the distance of* $\mathbf{x_t}$ *from the separator. That is,*

$$\gamma_t \quad = \quad y_t(\mathbf{u} \cdot \mathbf{x_t}).$$

Now we present the Perceptron algorithm.

PERCEPTRON ALGORITHM

1. Initialize $t := 1$ and $\mathbf{w_1}$ to be all 0 weight vector.
2. For each $\mathbf{x_t}$ predict $+1$ if $\mathbf{w_t} \cdot \mathbf{x_t} \geq 0$, else -1
3. If there is a mistake (i.e., if $y_t(\mathbf{w_t} \cdot \mathbf{x_t}) < 0$), set $\mathbf{w}_{t+1} \leftarrow \mathbf{w_t} + y_t \cdot \mathbf{x_t}$. Else, set $\mathbf{w}_{t+1} \leftarrow \mathbf{w_t}$.
4. $t \leftarrow t + 1$

Now we prove the mistake bound for the above algorithm.

Theorem 1. *Suppose there exists a* $\mathbf{u}$ *such that* $\|\mathbf{u}\| = 1$*, and* $\gamma > 0$*, such that* $\forall t \;\; y_t(\mathbf{x}_t \cdot \mathbf{u}) \geq \gamma$*, and there exists a real number* D *such that* $\forall t \;\; \|\mathbf{x}_t\| \leq D$*. Then, the number of mistakes made by the (unnormalized) Perceptron is* $\leq \left(\frac{D}{\gamma}\right)^2$.

Proof: Let $m(i)$ be the round in which the i^{th} mistake is made. Define $m(0) = 0$. Let k be the total number of mistakes made in the run of Perceptron algorithm. Finally, let $\mathbf{u}$ be the target separator that we are trying to learn. We prove the theorem through the following two lemmas.

Lemma 1. $\mathbf{w}_{m(k)+1} \cdot \mathbf{u} \geq k\gamma$.

Proof: We prove by induction on the number of mistakes that $\forall i, \mathbf{w}_{m(i)+1} \cdot \mathbf{u} \geq i\gamma$.

For the base case, note that as the initial weight vector $\mathbf{w_1}$ is all 0s, we have $\mathbf{w}_1 \cdot \mathbf{u} = 0$.

For the induction hypothesis, assume that the above statement holds true for all values less than i.

For the induction step, consider $\mathbf{w}_{m(i)+1}$. We have,

$$\begin{aligned} \mathbf{w}_{m(i)+1} \cdot \mathbf{u} &= (\mathbf{w}_{m(i)} + y_{m(i)}\mathbf{x}_{m(i)}) \cdot \mathbf{u} \\ &= \mathbf{w}_{m(i)} \cdot \mathbf{u} + y_{m(i)}(\mathbf{x}_{m(i)} \cdot \mathbf{u}). \end{aligned}$$

The first equality comes from the Perceptron update rule. We did make a mistake on round $m(i)$, so the weights at round $m(i) + 1$ can be computed by applying the update rule to the weights at round $m(i)$. Now, we know that we did not make a mistake between round $m(i - 1) + 1$ and round $m(i)$. Since the Perceptron only updates weights when there is a mistake, we have $\mathbf{w}_{m(i)} \cdot \mathbf{u} = \mathbf{w}_{m(i-1)+1} \cdot \mathbf{u}$. We also have $y_{m(i)}(\mathbf{x}_{m(i)} \cdot \mathbf{u}) \geq \gamma$, by the margin requirement in the statement of the theorem. Thus, we have,

$$\begin{aligned} \mathbf{w}_{m(i)+1} \cdot \mathbf{u} &\geq \mathbf{w}_{m(i-1)+1} \cdot \mathbf{u} + \gamma \\ &\geq i\gamma. \end{aligned}$$

The last inequality follows from the induction hypothesis. □

Lemma 2. $\|\mathbf{w}_{m(k)+1}\|^2 \leq kD^2$.

Proof: We prove by induction on the number of mistakes that $\forall i. \|\mathbf{w}_{m(i)+1}\|^2 \leq iD^2$.

For the base case, we have, $\|\mathbf{w}_{m(0)+1}\|^2 = \|\mathbf{w}_1\|^2 = 0$.

Let us assume that the statement is true for all values less than some i.

For the induction step, note that,

$$\begin{aligned} \|\mathbf{w}_{m(i)+1}\|^2 &= \|\mathbf{w}_{m(i)} + y_{m(i)}\mathbf{x}_{m(i)}\|^2 \\ &= \|\mathbf{w}_{m(i)}\|^2 + \|\mathbf{x}_{m(i)}\|^2 + 2y_{m(i)}(\mathbf{x}_{m(i)} \cdot \mathbf{w}_{m(i)}), \end{aligned}$$

where the first equality holds for the same reason as in Lemma 1 above. Now, as above, we have $\|\mathbf{w}_{m(i)}\|^2 = \|\mathbf{w}_{m(i-1)+1}\|^2$. Further, by the bound on the lengths of vectors in the theorem statement, we have $\|\mathbf{x}_{m(i)}\|^2 \leq D^2$. For the third term in the expression above, note that as there was a mistake in round $m(i)$, our prediction of the label did not match with the correct label. Thus, $y_{m(i)}(\mathbf{x}_{m(i)} \cdot \mathbf{w}_{m(i)}) < 0$. Therefore, we have,

$$\begin{aligned} \|\mathbf{w}_{m(i)+1}\|^2 &\leq \|\mathbf{w}_{m(i-1)+1}\|^2 + D^2 \\ &\leq iD^2. \end{aligned}$$

Here, the last inequality follows from induction. This proves the lemma. □

To prove Theorem 1, we recall the following simple fact from linear algebra: if $\mathbf{z}$ and $\mathbf{u}$ are two vectors such that $\|\mathbf{u}\| = 1$ and θ is the angle between $\mathbf{z}$ and $\mathbf{u}$, then we have,

$$\begin{aligned} \mathbf{z} \cdot \mathbf{u} &= \|\mathbf{z}\|\|\mathbf{u}\|\cos(\theta) \\ &\leq \|\mathbf{z}\|\|\mathbf{u}\| = \|z\| \end{aligned}$$

Putting all the above together, we have,

$$\begin{aligned} D\sqrt{k} &\geq \|\mathbf{w}_{m(k)+1}\| \\ &\geq \mathbf{w}_{m(k)+1} \cdot \mathbf{u} \\ &\geq k\gamma. \end{aligned}$$

Thus, $k \leq (D/\gamma)^2$. In the above, the first inequality is from Lemma 2, the second from the fact above, and the third from Lemma 1. □

2 The "Normalized" Perceptron Algorithm

In this section, we consider a variation of the Perceptron algorithm. This version, called the *normalized* version, differs from the previous version only in its update rule for the weight vector $\mathbf{w}_t$ when there is a mistake. As all other steps are the same, we simply state the new update rule:

NORMALIZED UPDATE RULE.
If there is a mistake, that is, if $y_t(\mathbf{w_t} \cdot \mathbf{x}_t) \leq 0$, then set

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|}.$$

Else, set $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$.

For the normalized Perceptron, we have the following mistake bound.

Theorem 2. *Suppose there exists a* $\mathbf{u}, \|\mathbf{u}\| = 1$*, and* $\gamma_n > 0$ *such that* $\forall t, y_t(\frac{\mathbf{x}_t}{\|\mathbf{x}\|} \cdot \mathbf{u}) \geq \gamma_n$*. Then the number of mistakes made by the normalized Perceptron is* $\leq \left(\frac{1}{\gamma_n}\right)^2$.

Proof Sketch: A run of the normalized Perceptron can be thought of as a run of the (unnormalized) Perceptron algorithm with a preprocessing step: we can imagine that before the Perceptron algorithm is run, for all rounds t, the vector $\mathbf{x}_t$ is normalized to $\frac{\mathbf{x}_t}{\|\mathbf{x}_t\|}$ (the margin bound is suitably modified). Now, the behavior of the two algorithms in terms of classification of points is identical: all points are labeled consistently by the two algorithms, and the mistakes also occur at the same points. Thus, we can use Theorem 1 to derive a mistake bound for the normalized Perceptron: in the pre-processing version, as all vectors $\mathbf{x}_t$ are normalized, $D = 1$. This gives the bound $k \leq \frac{1}{\gamma_n^2}$. □

Advantage of Normalized Perceptron. In this sub-section we would like to show that in some situations the normalized Perceptron might perform better (in terms of mistakes made) than the Perceptron algorithm. Note that for the Perceptron algorithm, we don't actually need to know the margin bound γ to run the algorithm. Consider a particular run of the algorithm. After the run define,

$$\gamma \quad := \quad \min_t y_t(\mathbf{x_t} \cdot \mathbf{u}).$$

This margin bound will be consistent with the run of the algorithm. Similarly, we can retrospectively define the normalized margin bound for the normalized Perceptron. We have,

$$\begin{aligned}
\gamma_n &:= \min_t y_t\left(\frac{\mathbf{x}_t}{\|x_t\|} \cdot \mathbf{u}\right) \\
&= \min_t y_t(\mathbf{x_t} \cdot \mathbf{u}) \frac{1}{\|\mathbf{x}_t\|} \\
&\geq \left(\min_t y_t(\mathbf{x}_t, \mathbf{u})\right)\left(\min_t \frac{1}{\|\mathbf{x}_t\|}\right) \\
&\geq \frac{\gamma}{D}.
\end{aligned}$$

Thus, $\frac{1}{\gamma_n^2} \leq \left(\frac{D}{\gamma}\right)^2$.

The above statement in a way shows that although in some cases the normalized margin assumption may seem less natural than the original margin assumption, the mistake bound from the normalized Perceptron is *always* at least as good as the mistake bound for the regular Perceptron, and sometimes better. In particular, it is better in cases in which there are points with small $\|x_t\|$ and small margin, making γ small, and other different points with big $\|x_t\|$ and big margin, making D big. Also note that even though the bound is always better, the actual empirical performance of the normalized Perceptron isn't necessarily better in all cases.

3 Learning Majority Functions

From mistake bounds derived in the previous sections, it seemed that the number of mistakes made by the algorithm does not depend upon the dimension n. This seems counter-intuitive, as one would expect the mistake bound to grow with n. In this section, we justify this intuition by showing an explicit relation between mistake bound and the dimension (using the Perceptron algorithm) to learn a special class of linear separators called *majority functions*.

For this section, our data points will be n-dimensional vectors in the space $\{-1, +1\}^n$. Let r be a non-negative odd integer of value at most n. Then, a majority function is represented by a vector $\mathbf{u}$ of the form $< 0, 0, 1/\sqrt{r}, 0, 1/\sqrt{r}, ... >$ with 0's in $n - r$ positions and $1/\sqrt{r}$ in r positions (r is the number of *relevant features* of $\mathbf{u}$, which we don't need to know in advance). Clearly, $\|\mathbf{u}\| = 1$. The labeling rule is: label $\mathbf{x}_t$ as $+1$ if $\mathbf{x}_t \cdot \mathbf{u} \geq 0$, and as -1 otherwise.

We want to learn this class using the Perceptron algorithm. Clearly, $D = \sqrt{n}$. Let $\gamma = \min_t y_t(\mathbf{x}_t \cdot \mathbf{u})$. Note that as each component of $\mathbf{x}_t$ is either $+1$ or -1, and r is odd, the dot product is a multiple of $1/\sqrt{r}$. Thus, $\gamma \geq 1/\sqrt{r}$. Using Theorem 1, we get the mistake bound as $(D/\gamma)^2 = nr$. It's easy to verify that we get the same mistake bound even if the points are first normalized.