CS 4540/CS 8803: Algorithmic Theory of Intelligence Fall 2023

Lecture 1: The Perceptron Algorithm

Instructor: Santosh Vempala *Lecture date: 8/21,8/23*

1 The Perceptron Algorithm

Binary classification is to classify data into two categories. Given data with labels, one expects to learn the classifier. Inspired by the neurons from brain, the Perceptron algorithm was introduced to learn the binary classifier from the labeled data.

1.1 Setting

Consider labeled data $(x, l(x))$, where $x \in X \subset \mathbb{R}^n$, $l(x) \in \{1, -1\}$. Each data point is labeled as

$$l(x) = \text{sign}(\langle w^*, x\rangle) = \text{sign}(\sum_{i=1}^{n} w_i^* x_i).$$

Here $w^* \in \mathbb{R}^n$ is the unknown normal vector to a separating hyperplane. We assume that $\|w^*\| = 1, \|x\| \leq 1$. Moreover, we denote that the margin for w^* as $\gamma := \min_{x\in D} |\langle w^*, x\rangle|$. This implies that for any x with label $l(x) = 1$, $\langle w^*, x\rangle \geq \gamma$; for any x with label $l(x) = -1$, $\langle w^*, x\rangle \leq -\gamma$.

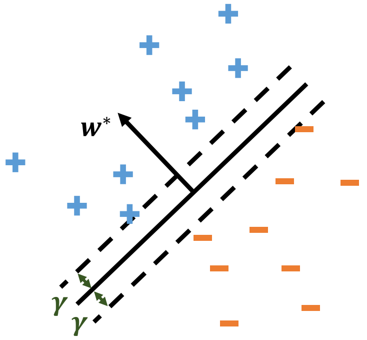

Figure 1: A sketch for the binary classification with normal vector w^* and margin γ.

1.2 Algorithm and Analysis

Algorithm 1 Perceptron Algorithm

```
Start with $w = 0$.
for each input $x$ do
    Predict sign($\langle w, x\rangle$)
    On a mistake, set $w \leftarrow w + l(x)x$
end for
```

Theorem 1.1. *The number of mistakes made by the Perceptron Algorithm on* ***any*** *input sequence is at most* $1/\gamma^2$.

Proof. Consider the potential function $\langle w, w^* \rangle / \|w\|$, starting at 0. Whenever the algorithm makes a mistake, we update as $\tilde{w} := w + l(x)x$. For the numerator, we have

$$\langle \tilde{w}, w^* \rangle = \langle w + l(x)x, w^* \rangle = \langle w, w^* \rangle + l(x)\langle x, w^* \rangle \geq \langle w, w^* \rangle + \gamma$$

The last inequality comes from the definition of margin γ. For the denominator of the potential function, we have

$$\|\tilde{w}\|^2 = \langle \tilde{w}, \tilde{w} \rangle = \langle w + l(x)x, w + l(x)x \rangle = \langle w, w \rangle + \langle x, x \rangle + 2l(x)\langle w, x \rangle$$

Since x is incorrectly predicted by w, we know $l(x) \neq \text{sign}(\langle w, x \rangle)$. This implies that $l(x)\langle w, x \rangle \leq 0$. Also we assume that $\|x\|$ is bounded by 1. So we have

$$\|\tilde{w}\|^2 \leq \langle w, w \rangle + \langle x, x \rangle \leq \|w\|^2 + 1$$

So after the algorithm makes T mistakes, the numerator is $\langle \tilde{w}, w^* \rangle \geq \gamma T$, while the denominator is $\|\tilde{w}\| \leq \sqrt{T}$. By the Cauchy-Schwarz inequality, $|\langle \tilde{w}, w^* \rangle| \leq \|\tilde{w}\| \|w^*\| = \|\tilde{w}\|$, which implies that $\langle \tilde{w}, w^* \rangle / \|\tilde{w}\| \leq 1$ at all time steps. Therefore,

$$1 \geq \frac{\langle \tilde{w}, w^* \rangle}{\|\tilde{w}\|} \geq \frac{\gamma T}{\sqrt{T}} = \gamma\sqrt{T}$$

It follows that $T \leq 1/\gamma^2$.

□

1.3 Lower Bound for the Perceptron Algorithm

Theorem 1.1 shows that the number of mistakes made by the Perceptron Algorithm is bounded by $1/\gamma^2$. However, when the margin is tiny, for instance, $\gamma \sim 1/2^n$, the number of mistakes can be as large as $2^{O(n)}$, which is not polynomial with respect to the input dimension n.

Here we construct a set of data, and show that for any given order, the Perceptron algorithm needs to see at least $2^{O(n)}$ examples before reaching the truth. For each data $x^{(i)} \in \mathbb{R}^n$ with i non-zero entries, we construct as follows.

	x	$l(x)$
$x^{(1)}$	$(1, 0, 0, 0, \cdots, 0)$	1
$x^{(2)}$	$(1, -1, 0, 0, \cdots, 0)$	-1
$x^{(3)}$	$(-1, -1, 1, 0, \cdots, 0)$	1
$x^{(4)}$	$(1, 1, 1, -1, 0, \cdots, 0)$	-1
$\cdots$		
$x^{(i)}$	$((-1)^i, \cdots, (-1)^i, (-1)^{i+1}, 0, \cdots, 0)$	$(-1)^{i+1}$

Lemma 1.2. *For a unit vector $w^* \in \mathbb{R}^n$ that classify the data above correctly, its i-th coordinate satisfies $w_i^* \geq 2^{i-1}$.*

Proof. By rescaling w^*, we can assume wlog that the margin is $\gamma = 1$. Since $l(x^{(1)}) = 1$, we have $\langle w^*, x^{(1)} \rangle \geq 1$. This derives that $w_1^* \geq 1$. Similarly we have $\langle w^*, x^{(2)} \rangle \leq -1$, which derives that $w_1^* - w_2^* \leq -1$, and thus $w_2^* \geq w_1^* + 1$. More generally, for odd i,

$$\langle w^*, x^{(i)} \rangle = -\sum_{j=1}^{i-1} w_j^* + w_i^* \geq 1 \qquad \Rightarrow \qquad w_i^* \geq \sum_{j=1}^{i-1} w_j^* + 1$$

For even i,

$$\langle w^*, x^{(i)} \rangle = \sum_{j=1}^{i-1} w_j^* - w_i^* \leq -1 \qquad \Rightarrow \qquad w_i^* \geq \sum_{j=1}^{i-1} w_j^* + 1$$

So we know $w_1^* \geq 1$ and for each $i \geq 2, w_i^* \geq \sum_{j=1}^{i-1} w_j^* + 1$. By induction, we know for any $i \in [n]$, $w_i^* \geq 2^{i-1}$. □

Corollary 1.3. *To achieve the correct classifier w^*, the Perceptron algorithm needs at least 2^{n-1} examples with mistakes.*

Proof. In each update step of the Perceptron algorithm, we let $w \leftarrow w + l(x)x$. By the property of the data constructed above, the absolute value of the change of w is at most one in each coordinate. By Lemma 1.2, $w_n^* \geq 2^{n-1}$. So we need at least 2^{n-1} update steps. □

2 Modified Perceptron Algorithm

In the last section, we saw that the Perceptron algorithm cannot be polynomial when the margin is tiny. Here we consider the setting where we only care about the data that are away from the margin. Specifically, given a parameter $\sigma > 0$, we would like to find w s.t. $\text{sign}(\langle w, x\rangle)$ is the correct label for all x satisfying

$$|\cos(w, x)| := \frac{|\langle w, x\rangle|}{\|w\|\|x\|} \geq \sigma.$$

We give the algorithm as follows. Instead of updating whenever we make a mistake using current w, here we update when the algorithm makes a mistake on a point far from the threshold (as measured by σ).

Algorithm 2 Modified Perceptron Algorithm

```
Start with $w \leftarrow$ random unit vector.
for each input $x$ do
    if $l(x) \neq \text{sign}(\langle w, x\rangle)$ and $|\cos(w, x)| \geq \sigma$ then
        $w \leftarrow w - \langle w, \bar{x}\rangle \bar{x}$, where $\bar{x} = x/\|x\|$
    end if
end for
```

Since we initialize w as a random vector, we give a high probability guarantee for the algorithm. The claim about the inner product of a random unit vector with a fixed unit vector will be proved in a later class.

Theorem 2.1. *With probability at least $1/8$, the number of mistakes made by Algorithm 2 is at most $\log n/\sigma^2$.*

Proof. Here we consider the potential function $\cos(w, w^*) = \frac{\langle w, w^*\rangle}{\|w\|\|w^*\|}$. By the random initialization, with probability at least $1/8$, $\langle w, w^*\rangle \geq 1/\sqrt{n}$. Then we consider each update step $w \to \tilde{w}$. For the numerator of the potential function, since $\text{sign}(\langle w, x\rangle) \neq \text{sign}(\langle w^*, x\rangle)$,

$$\langle \tilde{w}, w^*\rangle = \langle w, w^*\rangle - \langle w, \bar{x}\rangle\langle w^*, \bar{x}\rangle \geq \langle w, w^*\rangle$$

Since $|\cos(w, \bar{x})| = \frac{\langle w, \bar{x}\rangle}{\|w\|\|\bar{x}\|} \geq \sigma$, we have $\langle w, \bar{x}\rangle \geq \sigma\|w\|$. Then for the denominator of the potential function,

$$\|\tilde{w}\|^2 = \|w\|^2 + \langle w, \bar{x}\rangle^2\|\bar{x}\|^2 - 2\langle w, \bar{x}\rangle^2 = \|w\|^2 - \langle w, \bar{x}\rangle^2 \leq \|w\|^2(1 - \sigma^2)$$

After T steps,

$$1 \geq \cos(w, w^*) \geq \frac{1}{\sqrt{n}} \frac{1}{(1 - \sigma^2)^{T/2}}$$

So we have

$$(1 - \sigma^2)^{T/2} \geq \frac{1}{\sqrt{n}}$$

And thus

$$(\sigma^2)^T \leq (1+\sigma^2)^T \leq \left(\frac{1}{1-\sigma^2}\right)^T \leq n$$

$$T \leq \frac{\log n}{\sigma^2}$$

□

Algorithm 3 Modified Perceptron Algorithm

```
for round $i = 1, 2, \cdots, \log_{4/3} 1/\delta$ do
    implement Algorithm 2 with output $w$ and the number of mistakes
    if  the number of mistakes made in this around $\geq \ln n/\sigma^2$ then
        continue to the next round
    else
        output $w$
    end if
end for
```

We can extend the algorithm by doing it repeatedly so that the algorithm holds with probability $1-\delta$. See Algorithm 3 for the algorithm and Corollary 2.2 for the guarantee.

Corollary 2.2. *With probability at least $1-\delta$, the number of mistakes of Algorithm 3 is at most $O(\frac{\log n \log(1/\delta)}{\sigma^2})$.*

Proof. By Theorem 2.1, the probability that one round fails is 3/4. So the probability that the Algorithm 3 fails is $(3/4)^{\log_{4/3}(1/\delta)} = \delta$. So the algorithm holds with probability $1-\delta$. So the total number of mistakes are

$$\log_{4/3}(1/\delta) \log n/\sigma^2 = O\left(\frac{\log n \log(1/\delta)}{\sigma^2}\right)$$

□

3 Nonlinear Classifier

We can extend the setting to nonlinear ones. For example, we can label the data outside a ball $B(x_0, r)$ as $+1$ and inside it as -1. That is, $l(x) = \text{sign}(\|x - x_0\|^2 - r^2)$. More generally, we have the form

$$l(x) = \text{sign}(x^\top Ax - b)$$

Take an example of $l(x) = \text{sign}(x_1^2 + x_2^2 - 3x_1x_2)$ for $x \in \mathbb{R}^2$. We can reduce the problem to be the linear one by mapping

$$(x_1, x_2) \to (x_1^2, x_2^2, x_1x_2, x_1, x_2)$$

Then we get a linear classifier in the new coordinates with $w^* = (1, 1, -3, 0, 0)$.