COS-511: Learning Theory **Spring 2017**

Lecture 8: Perceptron Algorithm & The Hardness of Learning

Lecturer: Roi Livni

8.1 Learning Half Spaces cont.

In the last lecture we've seen an efficient algorithm for learning half spaces in the realizable setting. We next present a much more practical algorithm, yet with dependence on what is known as *the margin* of the problem.

8.1.1 The Perceptron

We next show an algorithm that learns halfspaces efficiently **In the realizable setting**, but with dependence on a *margin*.

```
Input: A sample $S = \{(x_i, y_i)\}_{i=1}^m$.
Intialize: $\mathbf{w}_0 = 0$
For i=1 to m
Set $\hat{y}_i = \text{sgn}(\mathbf{w} \cdot \mathbf{x}_i)$
Update $\mathbf{w}_i \to \mathbf{w}_{i-1} + \frac{1}{2}(y_i - \hat{y}_i)\mathbf{x}_i$.
End For
```

Theorem 8.1. *Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ s.t. $\|\mathbf{x}_i\| = R$. Assume S can be separated by a margin γ: Namely, there exists $\mathbf{w}^* \in \mathbb{R}^d$ such that $\|\mathbf{w}\| = 1$ and $y \cdot \mathbf{w} \cdot \mathbf{x} > \gamma$ for all $(\mathbf{x}, y) \sim D$. Then the Perceptron algorithm, will converge after $O(\frac{R^2}{\gamma^2})$ steps.*

Proof. Note that whenever the algorithm labels a point $\mathbf{x}_i$ correctly, there is no update. Now assume the k'th error is made on example t, set $\mathbf{w}^{(k)} = \mathbf{w}_t$, then we have

$$\mathbf{w}_{t+1} \cdot \mathbf{w}^* = (\mathbf{w}_t + y_{t+1}\mathbf{x}_{t+1})\mathbf{w}^* =$$
$$(\mathbf{w}^{(k-1)} + y_{t+1}\mathbf{x}_{t+1})\mathbf{w}^* \geq \mathbf{w}_t \cdot \mathbf{w}^* + \gamma$$

Where the first equality is from the update rule, the second equality is because $\mathbf{w}_t$ is not update when there are no mistake. The last inequality follows from the assumption $y\mathbf{w}^*\mathbf{x} \geq \gamma$. It follows by induction on k that $\mathbf{w}^{(k)} \cdot \mathbf{w}^* \geq k\gamma$. By Cauchy Schwartz, this means that $\|\mathbf{w}^{(k)}\| \geq k\gamma$. On the other hand, we have:

$$\|\mathbf{w}^{(k)}\|^2 = \|\mathbf{w}^{(k-1)} + y_t\mathbf{x}_t\|^2 = \|\mathbf{w}^{(k-1)}\|^2 + \|\mathbf{x}_t\|^2 + 2y_t\mathbf{x}_t\mathbf{w}^{(k)} \leq$$
$$\|\mathbf{w}^{(k-1)}\|^2 + \|\mathbf{x}_t\|^2 \leq \|\mathbf{w}^{(k-1)}\|^2 + R^2$$

Hence, by induction we obtain that $\|\mathbf{w}^{(k)}\|^2 \leq kR^2$. Taken together we obtain that

$$k\gamma \leq \sqrt{k}R.$$

Alternatively

$$k \leq \frac{R^2}{\gamma^2}.$$

□

Given efficient algorithms for learning half spaces (in the realizable setting), we might be encouraged to believe that the computational complexity of learning is always so well behaved. We end with a pessimistic note of a class that is learnable in the statistical model but not learnable in the computational model of learning, even in the realizable model. The result is due to Klivans and Sherstov and a proof can be found in their paper [1]. A similar result under different complexity assumption (and different assumption on the growth of k) has been proved in [2].

8.2 The hardness of learning

Theorem 8.2 (Intersection of Half–space is hard)**.** *Let $\chi = \{\pm 1\}^n$, let*

$$H^a = \{\mathbf{x} \to \sigma_{sgn}(\mathbf{w}\cdot\mathbf{x} - b - 1/2) : b \in \mathbb{N},\ \mathbf{w} \in \mathbb{N}^n, |b| + \|\mathbf{w}\|_1 \leq poly(n)\}$$

and let $H^a_k = \{x \to h_1(\mathbf{x}) \wedge h_2(\mathbf{x}) \wedge \ldots \wedge h_k(\mathbf{x}) : \forall i, h_i \in H^a\}$ where $k = d^\rho$ for some constant ρ. Then under certain cryptographic assumption, H^a_k is not efficiently learnable.

Remark 1. *The result is true in the* realizable *sense, i.e. in the hard instance, the true labeling stems from the class. Therefore this result demonstrate hardness both in the agnostic case and in the realizable case.*

Remark 2. *There has been various hardness results in the literature, showing that certain classes are not efficiently learnable. We note that what distinguish this result from others, is that it is true in the* improper *model – i.e. we do not assume that the learner returns a hypothesis from the class and yet the problem is hard.*

Because the result is true in the improper model, it is a powerfull tool to prove hardness of various classes, for example we can use this result to prove hardness of learning neural networks:

Corollary 8.3. *Every class of neural network that is fully connected (i.e. every neuron in layer t is connected to neuron at layer $t+1$, $\mathcal{N}_{(V,E),\sigma_{\text{sgn}}}$ with at least one hidden layer, with $V^{(0)} = n$ and $V^{(1)} = k$ where $k = d^\rho$, is not efficiently learnable.*

Proof. To prove the result, it is enough to show that $H^a_k \subseteq \mathcal{N}_{(V,E),\sigma}$. Once we show that, if we could learn $\mathcal{N}_{(V,E),\sigma}$ (improperly) then we would obtain an efficient algorithm for learning H^a_k. To see that above relation, we need to show that we can implement every $f \in H^a_k$ using a neural network. Let $f \in H^a_k$ such that:

$$f = h_1(\mathbf{x}) \wedge, \ldots, \wedge h_k(\mathbf{x})$$

and assume $h_i(\mathbf{x}) = \text{sgn}(\mathbf{w}^{(i)} \cdot \mathbf{x} - b_i - 1/2)$. Then we define the weight vector for the wedge between neuron $\mathbf{v}^{(1)}_j \in V^{(1)}$ and input node $\mathbf{x}_m$ to be $\mathbf{w}^{(j)}_m$ and we define the bias term to be $b(\mathbf{v}^{(1)}_j) = -b_i - 1/2$. The output of neuron $\mathbf{v}_j$ is then

$$\mathbf{v}_j = h_j(\mathbf{x}).$$

If the network has one hidden layer we set as the output node $\mathbf{v}^{(2)}(\mathbf{x}) = \sigma_{\sigma_{\text{sgn}}}(\sum \mathbf{v}^{(1)}_j(\mathbf{x}) - k + 1/2)$. One can verify that $f(\mathbf{x}) = \mathbf{v}^{(x)}(\mathbf{x})$ (This is similar to our argument a node can implement conjunctions).

If the network has more then one hidden layer, we let some neuron in the second layer implement f, then we simply propagate this neuron to the final layer by implementing and identity over the neuron (i.e. we connect it to some neuron in the next layer with weight 1, and so on up to the final layer and we disregrad all other neurons). $\square$

References

[1] Klivans, Adam R., and Alexander A. Sherstov. *Cryptographic hardness for learning intersections of halfspaces.* Journal of Computer and System Sciences 75.1 (2009): 2-12.

[2] Daniely, Amit, and Shai Shalev-Shwartz. *Complexity theoretic limitations on learning dnf's.* CoRR, abs/1404.3378 1.2.1 (2014): 2-1.