Introduction to Computational Learning Theory **Spring 2021**

Lecture 6: Hardness of Learning

Lecturer: Roi Livni *Scribe:*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.* We next review certain hardness results: here we will show that in principle it may be hard to learn certain instances which means we cannot hope to derive a tractable algorithm without certain assumptions.

6.1 The class of 3-DNFs

Consider the hypothesis class of 3-DNF. The class of 3-DNF is a class over the instance space $\mathcal{X}_n = \{0,1\}^n$ where each hypothesis is represented by a Boolean formula

$$h(x) = A_1(\mathbf{x}) \vee A_2(\mathbf{x}) \vee A_3(x),$$

where each $A_i(\mathbf{x})$ is a conjunction formula (i.e. $A_i(\mathbf{x}) = \mathbf{x}_{i_1} \wedge \mathbf{x}_{i_2} \wedge \ldots \mathbf{x}_{i_t}$ for some set of literals $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_t}$.)

Note that there are at most $O(2^{3n}$ 3-DNF formulas, hence the class is learnable with sample complexity $O(n)$.

It turns out that *proper* learning of 3-DNFs is computationally hard. and we cannot efficiently implement an ERM procedure, [**?**]

To show we can learn improperly 3-DNFs, let us consider the class $\mathcal{C}_n$ of conjunctions over $\mathcal{X}_n = \{0,1\}^n$. In particular, each hypothesis in $\mathcal{C}_n$ is described by a formula

$$h(\mathbf{x}) = \bigwedge_k \mathbf{x}_{i_k}.$$

for some set of indicies $i_1, \ldots, i_k \leq n$. Note that in the realizable model, we can learn $\mathcal{C}_n$ in time $O(n)$.

Conjunctions are efficiently learnable : Given a sample set $\{\mathbf{x}^{(i)}, y_i\}$ we can implement an ERM rule as follows: We start with the conjunction $h_0(\mathbf{x}) = \mathbf{x}_1 \wedge \mathbf{x}_2 \ldots, \wedge \mathbf{x}_n$. Then if it contradicts some 1-labeled examples, we remove all the literal that it doesn't satisfy (e.g. if $\mathbf{x}_1^{(1)} = 0$ then we remove the literal $\mathbf{x}_1$ from h_0) to obtain h_1, and so on, until we obtain h_m that is consistent with all the positively labeled examples. Since it is the most restrictive hypothesis that is consistent, then in the realizable case, we obtain an hypothesis that is consistent with the sample and we implemented an ERM rule.

Also note, that the class of conjunctions contains order of $O(2^n)$ hypotheses, hence has sample complexity $O(n)$, and is thus overall efficiently learnable.

6.1.1 But we can learn 3-DNFs improperly

We next describe an *improper* learning algorithm for 3-DNFs. We first note that because $\wedge$ is distributive with respect to $\vee$, each 3-DNF formula can be written as

$$A_1(\mathbf{x}) \vee A_2(\mathbf{x}) \vee A_3(\mathbf{x}) = \bigwedge_{i_1 \in A_1, i_2 \in A_2, i_3 \in A_3} \mathbf{x}_{i_1} \vee \mathbf{x}_{i_2} \vee \mathbf{x}_{i_3}.$$

So for each 3-DNF formula, there exists a conjunction over triplets of literals that has the same truth value. Thus, by embedding our data from $\mathcal{X}_n = \{0,1\}^n$ into a boolean space $\mathcal{X}_{n^3} = \{0,1\}^{n^3}$ (by considering all triplets of literals): We can learn a conjunction function in the embedded space.

The embedding takes $O(n^3)$ time, and since we are learning conjunctions the sample complexity will again be $O(n^3/\epsilon \log 1/\delta)$.

6.2 Hardness of (improper) Binary Learning

We've shown that learning half-spaces in the realizable case can be done efficiently. Unfortunately, the case for agnostic learning is less positive. [1] Showed that agnostic learning is hard in the proper case.

Later, [2] showed that under certain cryptographic assumptions, learning a class of $\Omega(n^\epsilon)$-*intersections of halfspaces* is also hard even in the improper model, for any $\epsilon > 0$. From this result one can derive hardness results for Agnostic learning of halfspaces.

The class of k intersection of half-sapces - INTERSECT$_k$ is parameterized by matrices $W \in \mathbb{R}^{k \times d}$ and every function in the class if of the form

$$f_W(\mathbf{x}) = \min_{i \leq k} \operatorname{sign}(\mathbf{w}_i \cdot \mathbf{x}).$$

Application to the study of Neural Networks The fact that the above results are given in the *improper* framework has an important consequence: Any hypothesis class that can express INTERSECT$_k$ (i.e. INTERSECT$_k \subseteq \mathcal{H}$) is hard to learn, that includes almost all existing neural network architecture.

The last result follows directly when we consider neural networks with threshold activation function (as we did in the last few classes, and we showed that we can implement conjunctions). In reality, one usually uses sigmoidal activation functions or some other surrogate for the sign function.

In more detail, when we take a smooth activation function, the network cannot implement "hard conjunctions" but only smoothed conjunctions. However, when one consider the hardness result presented in [2] to its fullest extent, one can also apply these results to sigmoidal-like neural networks (see for example [3] for further reading). Intuitively this is true because the hard instance in [2] does not rely on points that reside too close to the decision boundary, hence a smoothed relaxation of the decision boundary also leads to correct classification.

References

[1] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[2] Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

[3] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in neural information processing systems*, pages 855–863, 2014.