Advanced Machine Learning
Hilary Term 2017
Week 7

Problem Sheet 5

Instructions: The problem sheets are designed to increase your understanding of the material taught in the lectures, as well as to prepare you for the final exam. You should attempt to solve the problems on your own after reading the lecture notes and other posted material, where applicable. Once you have given sufficient thought to a problem, if you are stuck, you are encouraged to discuss with others in the course and with the lecturer during office hours. Please avoid posting on Piazza until after the submission deadline. You are *not permitted* to search for solutions online. Questions marked with an asterisk (*) are optional.

1 Proper Learning with Classification Noise

Let C be a concept class that is efficiently *proper* PAC-learnable, *i.e.,* there exists a learning algorithm that outputs $h \in C$, such that $\text{err}(h) \leq \epsilon$, in addition to the usual PAC-guarantees. Suppose that this same class C is PAC-learnable, but not necessarily *proper* PAC-learnable, in the presence of random classification noise. Show that, in fact, C is also *proper* PAC-learnable in the presence of random classification noise.

2 Learning Parities in the Presence of Noise

For this problem the distribution is fixed to be the uniform distribution, $\mathcal{U}$, over $\{0,1\}^n$. Thus, any learning algorithm that you design only has to succeed assuming that the data is generated from the uniform distribution; the error will also be measured with respect to the uniform distribution.

2.1 Persistent Random Classification Noise

We will allow the algorithm (membership) query access to the target function. However, the answers received by the algorithm may be noisy. Furthermore, if the learning algorithm queries the same point $x \in \{0,1\}^n$ several times, it receives the same answer each time. (Otherwise, it could simply query each point several times and use the majority label as the noise-free label.) This model of noise is called the persistent random classification noise model.

Formally, let $c \in C$ be the target concept. For noise rate η, define a (randomly chosen) function $c' : \{0,1\}^n \to \{0,1\}$ as follows:

$$c'(x) = \begin{cases} c(x) & \text{with probabilty } 1-\eta \\ 1-c(x) & \text{with probabilty } \eta \end{cases}$$

The random choice is independent for each $x \in \{0,1\}^n$. The algorithm can query a point $x \in \{0,1\}^n$, and it receives $c'(x)$. Since the algorithm knows that the distribution is *uniform*

over $\{0,1\}^n$, it does not require random (noisy) labelled examples from the distribution; it can generate uniform random points in $\{0,1\}^n$ itself and query their labels.

For simplicity, we will only require that the learning algorithm succeed with probability $\frac{1}{2}$, instead of the usual $1-\delta$. The probability is over the random choice of c' as well as any internal randomisation used by the algorithm (if required). Give a (possibly randomised) algorithm that learns the class PARITIES using membership queries in the presence of persistent random classification noise for any $\eta < \frac{1}{2}$. (You may assume that the algorithm knows η.)

2.2 Adversarial Noise

We will now consider a *stronger* form of noise. An adversary may corrupt the data that the algorithm receives; however the adversary is somewhat constrained. Recall that when c' was chosen randomly, it is the case that $\mathbb{P}_{x\sim\mathcal{U},c'}[c(x) \neq c'(x)] \leq \eta$. We will now allow c' to be chosen by an adversary, with the only constraint being that $\mathbb{P}_{x\sim\mathcal{U}}[c(x) \neq c'(x)] \leq \eta$, *i.e.,* it must be the case that $c(x) = c'(x)$ on all but at most η fraction of $\{0,1\}^n$. However, the points where the label is corrupted may be chosen by the adversary to inflict maximum damage on any learning algorithm.

1. When $\eta = \frac{1}{5}$, show that the class PARITIES can be learnt with membership queries under adversarial noise. Recall that your goal is still to output some h, such that $\mathrm{err}(h; c, \mathcal{U}) \leq \epsilon$, *i.e.,* your error has to be low with respect to the true target c, not the corrupted c'.

2. Show that PARITIES is not PAC-learnable with membership queries under adversarial label noise, when $\eta \geq \frac{1}{4}$.

3 Agnostic Learning MONOTONE-CONJUNCTIONS

In this problem, we will consider the *agnostic* setting. In the agnostic setting, we make no assumptions whatsoever about how the data is labelled. In particular, we let D be an arbitrary distribution over $X \times \{0,1\}$, where X is the instance space. (For example, it is not ruled out that you sometimes observe $x \in X$ with label 1, and at other times with label 0.) For agnostic learning, the example oracle $\mathsf{EX}(c, D)$ in the case of PAC-learning is replaced by $\mathsf{EX}(D)$, where D is now a joint distribution over instances and labels.

As there is no promise of data being labelled according to any concept in a class, we will relax the requirement that the learning algorithm output a highly accurate hypothesis. We will say that a concept class C is *agnostically* learnable if the output hypothesis of the learning algorithm, h, satisfies with probability at least $1-\delta$,

$$\mathrm{err}(h; D) \leq \min_{c \in C} \mathrm{err}(c; D) + \epsilon,$$

where for a distribution D over $X \times \{0,1\}$ and a boolean function $f : X \to \{0,1\}$, $\mathrm{err}(f; D)$ is defined to be $\Pr_{(x,y)\sim D}[f(x) \neq y]$. In other words, we are saying that C is agnostically learnable if it is possible to predict as well as the best concept from C, not matter what the observed data distribution.

1. Show that the class MONOTONE-CONJUNCTIONS is not efficiently *proper agnostically* learnable, *i.e.,* if the output hypothesis h is required to be in MONOTONE-CONJUNCTIONS, unless RP = NP.

   Hint: Consider reducing from Vertex-Cover. *Create exactly the same labelled sample as in the proof to show proper learning* 3-TERM-DNF *is hard unless* RP = NP.

*2. Show that this remains the case even when the algorithm is allowed to output a hypothesis that is a conjunction (not necessarily monotone).

   Hint: Consider adding the positive example, $((1, 1, 1, \ldots, 1), 1)$ *to your previous sample. You may need to spread the probability mass around in a non-uniform manner.*