Computational Learning Theory
Michaelmas Term 2024

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. Problems marked with an asterisk are optional. 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. You are *not permitted* to search for solutions online.

1 Learning Rectangles using Statistical Queries

We will consider an extension of the statistical query model, where in addition to making queries of the form (χ, τ) to the oracle $\mathsf{STAT}(c, D)$, the learning algorithm is allowed access to *unlabelled* examples from D, *i.e.,* it may get points $\mathbf{x} \in X$ drawn according to D, but not the labels $c(\mathbf{x})$.

1. Briefly argue why any concept that is (efficiently) learnable with access to $\mathsf{STAT}(c, D)$ and unlabelled examples, is also (efficiently) learnable with access to the noisy example oracle, $\mathsf{EX}^{\eta}(c, D)$.
2. Give an efficient algorithm for learning axis-aligned rectangles in the plane using $\mathsf{STAT}(c, D)$ and unlabelled examples.

Hints:

1. We can obviously get unlabelled examples sampled according to D if we have access to $\mathsf{EX}^{\eta}(c, D)$.
2. Using the unlabelled examples, figure out how to make queries to implement *binary search* to find the boundaries in either direction. Think about why you need unlabelled examples.

2 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 $\mathrm{err}(h) \leq \epsilon$, in addition to the usual PAC-guarantees. Suppose that this same class C is efficiently PAC-learnable, but not necessarily efficiently *proper* PAC-learnable, in the presence of random classification noise. Show that, in fact, C is also efficiently *proper* PAC-learnable in the presence of random classification noise.

Hint: Think how you would simulate $\mathsf{EX}(c, D)$ given access to $\mathsf{EX}^{\eta}(c, D)$ and the *improper* efficient learning algorithm using $\mathsf{EX}^{\eta}(c, D)$.

3 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.

3.1 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 $\mathbf{x} \in \{0,1\}^n$ 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:

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

The random choice is independent for each $\mathbf{x} \in \{0,1\}^n$. The algorithm can query a point $\mathbf{x} \in \{0,1\}^n$, and it receives $c'(\mathbf{x})$. 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). For sufficiently large n, give an *efficient* (possibly randomised) algorithm that learns the class $\mathsf{PARITIES}$ 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.)

Extra: Explain why your solution doesn't work for small n.

3.2 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 $\mathbb{P}_{\mathbf{x}\sim\mathcal{U},c'}[c(\mathbf{x}) \neq c'(\mathbf{x})] \leq \eta$. We will now allow c' to be chosen by an adversary, with the only constraint being that $\mathbb{P}_{\mathbf{x}\sim\mathcal{U}}[c(\mathbf{x}) \neq c'(\mathbf{x})] \leq \eta$, *i.e.,* it must be the case that $c(\mathbf{x}) = c'(\mathbf{x})$ 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. The only constraints on the adversary are that the total error cannot be more then η and it has to introduce the noise before the learning algorithm asks queries.

1. When $\eta = \frac{1}{5}$, show that the class $\mathsf{PARITIES}$ can be learnt with membership queries under adversarial noise for sufficiently large n. 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 $\mathsf{PARITIES}$ is not PAC-learnable with membership queries under adversarial label noise, when $\eta \geq \frac{1}{4}$.

Hints:

Persistent Random Classification Noise: Use the fact that the algorithm can make correlated queries, i.e. it can query $c'(\mathbf{x})$ and $c'(\mathbf{x}')$ where $\mathbf{x}$ and $\mathbf{x}'$ are not (necessarily) stochastically independent. Can you use this to find $c(\mathbf{x})$ for some $\mathbf{x}$ of your choosing (whp)?

Adversarial Noise

1. Show that your argument from the previous part can still be adapted.
2. Why does the same argument now fail? Show that the adversary can construct a corrupted function that is equally close to three distinct parity functions, and that the learning algorithm has no means to distiguish between them.