CSE522, Winter 2011, Learning Theory Lecture 15 - 03/10/2011

Beyond Inductive Learning

Lecturer: Ofer Dekel *Scribe: Thach Nguyen*

Throughout this class, we have been looking at inductive, a.k.a. supervised, learning, where the sample consists of (data, label) pairs and the goal is to predict the labels of future data. In this last class, we look at settings that go beyond this simple model.

1 General Learning Problems

First, we give a general definition of learning problems – that is, a strict generalization of the inductive learning problems that we have grown familiar with.

Definition 1. *A general learning problem consists of a data set Z, a distribution $\mathcal{D}$ on Z, a hypothesis class H and a loss function $\ell : H \times Z \to \mathbb{R}_+$. The goal of a learning algorithm is to output a hypothesis $h \in H$ so as to approximately minimize $\mathbf{E}_{z\sim\mathcal{D}}[\ell(h;z)] = \ell(h;\mathcal{D})$*

Note that if $Z = X \times Y$ where X is a dataset and Y is a label set, then we have a familiar inductive learning problem. To see that this definition is a strict generalization of the definition of inductive learning problems, let us look at some example problems that are not inductive.

Example 1 **k-centroid** Let S be the set of points in $\mathbb{R}^n$ and x be another point in the same space. Define the distance between x and S to be the distance from x to the neareast point in S. We would like to find a set S of k points so as to minimize the expected distance between a point y drawn from a distribution $\mathcal{D}$ on $\mathbb{R}^n$ to S. In this case, we have $Z = \mathbb{R}^n$, $H = \left\{\{c_i\}_{i=1}^k | c_i \in \mathbb{R}^n\right\}$ and $\ell(\{c_i\}_{i=1}^k; x) = \min_i \|x - c_i\|_2$.

Example 2 density estimation In this rather general example, ℓ is the log likelihood of the data.

In the case of inductive learning, we have a nice equivalent between the learnability of the problem and the uniform convergence of the empirical risk minimizer to the risk minimizer. For example, in the case of binary classification, if the problem has finite VC dimension, then

$$sup_{h\in H}|\ell(h;\mathcal{D}) - \ell(h;S)| = \tilde{O}\left(\frac{1}{\sqrt{m}}\right)$$

and the problem is learnable by an empirical risk minimizer. On the other hand, if the problem has infinite VC dimension, then it is not learnable. Hence, one can say that a binary classification problem is learnable if and only if the above condition holds.

Obviously, if the condition holds for a general learning problems then the problem is still learnable (by an empirical risk minimizing algorithm). On the other hand, the reverse direction is no longer true. In the following, we will give an example learning problem that is learnable but doesn't satisfy the uniform convergence condition.

The problem is defined by

- $H = \{h \in \mathbb{R}^\infty : \|h\|_2 \leq 1\}$.
- $Z = \Gamma \times C$ where $\Gamma = \{0,1\}^\infty$ and $C = \{c \in \mathbb{R}^\infty : \|c\|_2 \leq 1\}$
- $\ell(h;(\gamma,c)) = \|\gamma * (h-c)\|_2$ where $*$ is the coordinate-wise product, i.e., $x * y$ is a vector $z \in \mathbb{R}^\infty$ where $z_i = x_i y_i$.
- $\mathcal{D}$ is the distribution where $c = 0$ with probability 1, and $\gamma_i = 1$ with probability $1/2$ and $\gamma_i = 0$ with probability $1/2$.

Consider any set S of m samples from $\mathcal{D}$, with probability 1, there is some coordinate j such that all the sampled γ's have 0 at the jth coordinate, i.e., $\gamma_{1j} = \gamma_{2j} = \cdots = \gamma_{mj} = 0$. Then for $h = e_j$, the vector that has 1 at the jth coordinate and 0 anywhere else, we have $\ell(h; S) = 0$ but $\ell(h; \mathcal{D}) = 1/2$. Thus, the uniform convergence condition doesn't hold.

On the other hand, the problem is learnable. For now, assume that $c = 0$ (since $\|c\|_2$ is bounded by a constant, the general analysis is similar). With this assumption, we have

$$\ell = \sum_{i=1}^{\infty} \gamma_i h_i^2 \leq \sum_{i=1}^{\infty} h_i^2 \leq 1.$$

and $\nabla_h(\ell)$ is bounded by a constant. These two facts imply that the online gradient descent algorithm has a regret of $O(\sqrt{m})$ and that the Azuma's bound holds, hence we can use online to batch conversion to get a good learning algorithm.

The above example shows that even if the problem is learnable, the empirical risk minimizer may not converge to the risk minimizer, hence is a bad indicator of what the true risk minimizer may look like. Surprising, the minimum empirical risk, on the other hand, does converge to the minimum risk, given that the problem is learnable. (Yes, read the two sentences again!)

Let $h^* = \text{argmin}_{h \in H} \ell(h; \mathcal{D})$ and h_{ERM} be the empirical risk minimizer, then the above paragraph says that h_{ERM} may not converge to h^*, but $\ell(h_{ERM}; S)$ always converges to $\ell(h^*; \mathcal{D})$.

This surprising fact stems from a perhaps more intutive fact that if S is a big i.i.d sample from $\mathcal{D}$ then an algorithm seeing a very small fraction of S' of S shouldn't be able to distinguish the two cases: (i) S' was sampled uniformly from S and (ii) S' was sampled from $\mathcal{D}$.

Now consider an algorithm A that learns the problem. Since A cannot distinguish between the uniform distribution over S and $\mathcal{D}$, $\ell(A(S'); \mathcal{D})$ must converge to both $\min_h \ell(h; \mathcal{D}) = \ell(h^*, \mathcal{D})$ and $\min_h \ell(h; S) = \ell(h_{ERM}; S)$. This tells us that $\ell(h^*; \mathcal{D})$ and $\ell(h_{ERM}; S)$ are essentially the same.

2 Non-inductive Learning Problems

Now we give brief introductions to other types of learning problems.

2.1 Transductive learning

Transductive learning problems are almost like inductive learning ones, except that the algorithm knows the test data. Thus, the goal is not to do well with an arbitrary data point from an unknown distribution, but with a known set of data points. Formally, in transductive learning, we have a set X of data, a set Y of labels, a loss function $\ell : H \times (X \times Y) \to \mathbb{R}_+$ and a distribution $\mathcal{D}$ over $X \times Y$. Let $S = \{(x_i, y_i)\}_{i=1}^m$ and $S' = \{(x'_i, y'_i)\}_{i=1}^{m'}$ be sampled from $\mathcal{D}$. A learning algorithm takes as input S and $\{x'_i\}_{i=1}^{m'}$ and outputs a hypothesis h so as to minimize $\ell(h; S')$.

2.2 Semi-supervised learning

The setting of semi-supervised learning is the same as that of transductive learning, i.e, the algorithm also takes as inputs the set S and $S'|_X \{x'_i | (x'_i, y'_i) \in S'\}$. However, its aim is to find a hypothesis h so as to minimize $\ell(h; \mathcal{D})$. Here, $S'|_X$ merely gives the algorithm more information about the underlying distribution $\mathcal{D}$ and plays no role in the test.

2.3 Growing label set

In our discussions of supervised learning, we have always assumed that the label set Y is "static". How about the case where the label set grows with the number of samples? Example includes wikipedia categories and Flickr tags.

2.4 Finite sample space

Another assumption we made in our discussion is that the sampled space is infinite. This allows us to use asymptotic bounds freely. How about situations such as colaborative filtering where the sample space is finite? Take the Netflix test for example. In this test, we are given a matrix representing the scores users gave movies with many entries missing. The task is to learn/predict the missing entries using the present entries.

2.5 Beyond minimizing expected loss

Even in the general learning problems, we set the goal to minimize the expected loss. But other quantities can be optimized as well. For example, we may want to minimize some "local loss" such as, in the case of binary specification with 0/1-loss, $\mathbf{Pr}\left[h(X) \neq Y | X = x\right]$ or $\mathbf{Pr}\left[h(X) \neq Y | \|X - x\| \leq \rho\right]$.

We can also use "funky" loss function, such as the F_1-*score*, which is

$$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive+False positive}} \quad \text{and} \quad \text{Recall} = \frac{\text{True positive}}{\text{True positive+False negative}}$$

Note that this loss is not decomposible into point-wise loss, and these learning problems do not even fit into the general framework.