THEORETICAL MACHINE LEARNING COS 511

# LECTURE #1

FEBRUARY 3, 2015

LECTURER: ELAD HAZAN SCRIBE: HOLDEN LEE

## 1 Administrative information

All administrative information including grading, final project, background needed, office hours, TA, and scribing notes appear on the course website.

## 2 Machine Learning: Scope and Goals

**Motivation.** By our nature as intelligent beings, we all have intuitive understanding of what it means to "learn". The basic mechanism that allows us to improve our actions based on prior experience. Children learn (and very quickly at that) complex tasks such as walking, crawling, speaking, reading etc. Adults learn to drive cars. Some learn mathematics and theoretical machine learning in universities.

Some features of the learning problems above is already "innate", i.e. much of the knowledge of walking seems to be "hard-wired" into our brains. Another example is of birds - migrating south in winter and north in summer. But that too, can be viewed as another biological mechanism for learning that is part of our genetic material. Biology has "learned" a certain structure for wings of birds that is well-suited for flight, an allocation of chlorophyll in plants to make use of the sun's energy, and a certain physical configuration for our bodies that is optimal for certain behaviors. The theory of evolution can, in this sense, also be seen as a theory of learning.

Learning, both biological evolution as well as learning by intelligent beings, is a fundamental process in nature and a most useful one at that. It is only natural that we try to understand this mechanism, how it works, how to reproduce it and exploit it.

**Machine learning:** refers to the automation of the process of learning from experience.

**Why study machine learning?** Two good reasons are:

1. Learning is a fundamental trait of intelligence, and a key to understanding ourselves and our world.
2. It is hard not to realize that ML is changing the world, very rapidly, and the rate of change is accelerating. Most of the major industries use learning techniques for their products, i.e. speech recognition/video classification and editing / text and automated semantic analysis / language translation / self-driving-cars / etc.

   The reason is that to solve real problems - classical algorithmic theory doesn't cut it. The world is too noisy, inaccurate and changing to model it using classical algorithmic theory. You must have a learning component in your system for it to work. And today learning is advanced and generic enough to allow this.

**What will we study in this course:** The course will start with the general theories of statistical and computational learning. We will proceed with describing various specific problems. These will motivate other learning models, which we will define and analyze rigorously. Through the course we will study algorithms

that form some of the most widely used systems today, analyze their properties and running time. Some of the questions we will try to formulate and answer:

1. what are the intrinsic properties of learning problems that make them learnable / unlearnable, or in the first case, hard/easy to solve?
2. how many examples are needed to be able to learn a particular concept?
3. why are simpler hypothesis better than complex ones?
4. how to define a reasonable and attainable goal in online learning?

### 2.1 examples of learning problems:

1. Learning to play the game of chess.
2. Recommending media to consumers based upon their previous choices.
3. Natural language processing.
   - Given a set of emails, classify them into spam/not spam.
   - Categorize web pages into classes (topics), for example, sports, news...
4. Object recognition. For example, does a picture contain a cat?
5. Understand biological data (genome sequences). Try to infer genetic traits like diseases.
6. Control a system. For example, drive a car.
7. Optimal portfolio selection.
8. Weather forecasting.

These examples show that learning problems come in many shapes and flavors. Distinguishing features include:

- The output of a problem can be binary, multi-class (one of several labels), or a real number. For example, for the problem of driving a car, the output might be a picture of where the car is allowed to move.
- The input can be natural (coming with a distribution), or adversarial, for example, spam emails that try to fool the filter.
- The problem is inherently online, or examples can be studied offline. For example, driving and optimal portfolio selection are online problems.

## 3 The statistical learning model

Consider a very simplistic problem, the apple juice factory problem. We have to classify apples into sweet or sour (we only want to use sweet apples). Suppose we can observe

- weight (kg)
- diameter (cm)

We can make a plot of diameter vs. weight, and use $+$ and $-$ to represent sweet and sour apples, respectively. We want to build a machine so that given a new apple, it will be able to distinguish whether it is sweet or sour. This toy problem motivates the formal model called the statistical learning model.

**Definition 3.1.** The elements of the **statistical learning model** are as follows.

1. Input:
   - Domain $X$; every point of $X$ has features that we have observe.
   - Label set $Y$.
   - Data, a set of labeled examples $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq X \times Y$.
2. Output: A prediction rule/hypothesis $h : X \to Y$.
3. Data generation: There exists a (unknown) distribution over $X$ and the label is given by some mapping $f : X \to Y$ (many times called a "concept"). $S$ is generated by iid samples from $D$, and they are labeled by $f$.
4. Performance metric: The risk/generalization error. For example, we could set

$$\text{err}(h) = \mathbb{P}_{x\sim D}[h(x) \neq f(x)] = \mathbb{E}_{x\sim D}[|h(x) - f(x)|],$$

   the second inequality being true when $Y = \{0, 1\}$.

**The danger of overfitting.** Suppose each apple is sampled uniformly at random from a rectangle up to 200g and 10cm, there is a small rectangle $R$ inside this rectangle, and

$$f(x) = \begin{cases} +, & \text{if } x \in R \\ -, & \text{otherwise.} \end{cases}$$

Suppose $\frac{\text{Area}(R)}{\text{Area(big rectangle)}} = \frac{1}{2}$.

A natural approach is to output a hypothesis that does well on the sampled training data. Let $h_{\text{ERM}} : X \to Y$ (where ERM stands for **empirical risk minimization**) be a function that minimizes the error for the sample $S$. In any statistical learning problem, the following is a valid choice for $h_{\text{ERM}}$:

$$h_{ERM}(x) = \begin{cases} y_i, & \text{if } x = x_i \text{ for some } i \\ +, & \text{otherwise} \end{cases}$$

This hypothesis is optimal w.r.t. the data - it has zero error on the training set.

However, this hypothesis does miserable outside of the sample: $\text{err}_D(h_{ERM}) = \frac{1}{2}$. Why does it fail so miserably?

The reason is that we have not restricted the hypothesis class in which we searched for an explanation to the data: it was too large and we have **overfitted** the data. We need a restrict the hypothesis to those that "make sense".

The way to remedy for overfitting is by using prior knowledge. We make the following modification. We must have $h \in \mathcal{H}$ where $\mathcal{H}$ is a predetermined **hypothesis class**; it is a set of functions $h : X \to Y$. A simple restriction is that $\mathcal{H}$ be finite.

For example, for the apple factory problem we let $\mathcal{H}$ be (the indicator function for) all rectangles which are axis aligned with a precision of 1g and 1cm. Then $|\mathcal{H}| \leq 200^2 \cdot 100^2 \cdot 2$. (There is some overcounting here. The factor of 2 is because the apples can be sweet inside the specified rectangle, or outside.) Given this restriction, can we learn the hypothesis class?

Note there is a potential computational complexity issue; we don't want to go through the functions in $\mathcal{H}$ one by one and pick out the best hypothesis. This issue we ignore for now.

By restricting the hypothesis class we might have restricted ourselves already too much, to hypothesis that do not generalize well. For now, we remedy this by making an additional assumption, called the **realizability assumption**: $f \in \mathcal{H}$ (in other words, there exists $h \in \mathcal{H}$ such that $\text{err}_D(h) = 0$). This assumption essentially means that our prior knowledge was justified: we restricted our search to a reasonable hypothesis space.

# 4 Learning in the statistical model: the ERM algorithm

We return to the natural strategy from before: producing a hypothesis that minimizes the empirical error on the training set, which we called the ERM hypothesis. However, this time, we restrict ourselves to hypothesis from the class $\mathcal{H}$. We will show that with the additional restriction of $\mathcal{H}$ and the realizability assumption, the statistical learning model is learnable by the ERM algorithm. We will now see the first of a series of results establishing this.

**Theorem 4.1.** *In the statistical learning model, assume that the function $f : X \to Y$ is realizable, $\mathcal{H}$ is finite, and let $h_{ERM}$ be defined on $S$ as before. Then for $|S| \geq \frac{1}{\varepsilon} \lg \frac{|\mathcal{H}|}{\delta}$, for all $\varepsilon, \delta > 0$, we have with probability $1 - \delta$ that*

$$err(h_{ERM}) \leq \varepsilon.$$

**discussion:** The theorem is very general; it applies to any learning problem in the statistical model. The sample size of $S$ needed depends logarithmically on the size of the hypothesis set, which is very nice. For example, in the apple classification problem,

$$\lg |\mathcal{H}| \approx 4 \lg(40000) \leq 50, \quad \delta = .01, \quad \varepsilon = .01.$$

Then by Theorem 4.1, we can take the sample size to be

$$|S| = 100 \ln(100^2 \cdot 200^2 \cdot 2) \leq 10000.$$

Here are questions that the theorem doesn't address (more on this later):

1. What if $f \notin \mathcal{H}$? This is called **agnostic learning**. We can't say that $\text{err}(h_{\text{ERM}}) \leq \varepsilon$ but we can find $h$ that best approximates $f$ in the hypothesis class. (See next week.)
2. How to choose $\mathcal{H}$?
3. Can we relax the iid assumption?

*Proof of Theorem 4.1.* Define

$$\text{err}_D(h) = \mathbb{P}_{x \sim D}[h(x) \neq f(x)]$$
$$\text{err}_S(h) = \frac{\{(x, y) : h(x_i) \neq y_i\}}{|S|}.$$

We have $\text{err}(h_{\text{ERM}}) = 0$ because by definition, $h_{\text{ERM}} \in \arg\min \{\text{err}_S(h) : h \in \mathcal{H}\}$, $f \in \mathcal{H}$, and $\text{err}_D(f) = \text{err}_S(f) = 0$ by realizability.

We want to show

$$\mathbb{P}[\text{err}_D(h_{\text{ERM}}) > \varepsilon] < \delta.$$

We'll show that if $h$ has $\text{err}_D(h) > \varepsilon$, then there is a low probability that it is correct on all of $S$ and hence chosen for $h_{\text{ERM}}$; a union bound gives that there is low probability that any such $h$ is chosen for $h_{\text{ERM}}$.

Define the following subsets of $\mathcal{H}$:

$$\mathcal{H} \supseteq \mathcal{H}_{\text{bad}} := \{h : \text{err}_D(h) > \varepsilon\}$$
$$\mathcal{H} \supseteq \mathcal{H}_{\text{misleading}} := \{h : \text{err}_S(h) = 0\}.$$

Now if $\text{err}_D(h_{\text{ERM}}) > \varepsilon$, then $\mathcal{H}_{\text{bad}} \cap \mathcal{H}_{\text{misleading}} \neq \phi$. Then

$$\begin{aligned}\mathbb{P}(\mathcal{H}_{\text{bad}} \cap \mathcal{H}_{\text{misleading}} \neq \phi) &= \mathbb{P}\left[\bigcup_{h \in \mathcal{H}_{\text{bad}}} \text{err}_S(h) = 0\right] \\ &\leq \sum_{h \in \mathcal{H}_{\text{bad}}} \mathbb{P}\left[\text{err}_S(h) = 0\right]\end{aligned}$$

by the union bound ($\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$). Let $M = |S|$.

$$\begin{aligned}\sum_{h \in \mathcal{H}_{\text{bad}}} \mathbb{P}[\text{err}_S(h) = 0] &= \sum_{h \in \mathcal{H}_{\text{bad}}} \mathbb{P}\left[(h(x_i) = y_i) \wedge \cdots \wedge (h(x_M) = y_M)\right] \\ &= \sum_{h \in \mathcal{H}_{\text{bad}}} \prod_{i=1}^{M} \mathbb{P}[h(x_i) = y_i] && \text{examples are sampled i.i.d} \\ &\leq \sum_{h \in \mathcal{H}_{\text{bad}}} \prod_{i=1}^{M} (1 - \varepsilon) = |\mathcal{H}_{\text{bad}}|(1 - \varepsilon)^M \\ &\leq |\mathcal{H}| e^{-\varepsilon M} \\ &\leq |\mathcal{H}| e^{-\varepsilon \frac{1}{\varepsilon} \ln \frac{|\mathcal{H}|}{\delta}} && \text{using } |S| = M \geq \frac{1}{\varepsilon} \ln \frac{|\mathcal{H}|}{\delta} \\ &= |\mathcal{H}| e^{-\ln \frac{|\mathcal{H}|}{\delta}} = |\mathcal{H}| \cdot \frac{\delta}{|\mathcal{H}|} = \delta.\end{aligned}$$

□

This theorem is an example of more general learning framework is called called **probably approximately correct** learning. There are 2 imprecisions here:

1. We only succeed with high probability ("probably"), and
2. we cannot hope to get the answer exactly right ("approximately").

The general gist of the theorem is this: we are given data from a labelled domain, sampled i.i.d. If the sample is large enough, then the ERM algorithm will perform well on data you haven't seen yet. The size of the sample required to attain a meaningful guarantee behaves well w.r.t the size of the hypothesis class and other natural parameters.

# 5 Beyond the simple statistical learning model

What are the strengths of the statistical learning model and Theorem 4.1?

1. The setting is general: domain $D$ and hypothesis class $\mathcal{H}$ are unrestricted.
2. The sample complexity - number of samples required for generalization - depends logarithmically on size of the hypothesis class.

The weaknesses are

1. $\mathcal{H}$ needs to be chosen carefully to allow for realizability.
2. It assumes realizability.
3. Noisy data is not handled.
4. $\mathcal{H}$ must be finite.
5. It requires iid samples.
6. It doesn't say anything about computational efficiency.
7. It's in the batch setting (as opposed to online).

Remark: Removing the realizability assumption allows the algorithm to handle noise. If you have a perfect hypothesis and add noise to it, then you have a hypothesis that is close but not perfect.

Next week we will address some of these weaknesses.