Math 547: Mathematical Foundations of Statistical Learning Theory

Lectures — Week 1

Instructor: Stas Minsker *Scribe: Stas Minsker*

These notes present a condensed summary of the first week of lectures.

1.1 Informal introduction: Machine Learning (ML) vs Mathematical Statistics (ST)

- **What is Machine Learning (ML)?**
  - ML is the study of algorithms that can "learn from data," gradually improving their accuracy in specific tasks and recognizing patterns without being given explicit instructions.
- **How is ML different from Mathematical Statistics (ST)?**
  - **Statistical Inference**: Statistics is the science of basing *inferences* on observed data and making decisions in the face of *uncertainty*.
  - Quantifying uncertainty is mainly a statistical task, and it often requires a model specification.
  - **Modeling**: In ML, one often deals with data too complex to be approximated by a "model" (e.g., think of handwritten digits). ST often deals with relatively small datasets, while ML commonly handles thousands or even millions of samples.
- **Traditional Categories in ML**:
  - **Unsupervised Learning**:
    - Find structure/patterns in the data.
    - Inputs are *unlabeled*.
  - **Supervised Learning**:
    - Inputs are in the form of **(observation, response)**.
    - Goal: learn to predict the response variable (e.g., a binary label) based on the observation/instance.
  - **Reinforcement Learning**:
    - Learn to maximize some "reward" based on feedback from the environment.
    - Example: Multi-armed bandit problem, algorithms that play games such as Go, Chess, Poker.

1.2 Binary classification framework

- Example of a problem: given an image, decide whether it is an image of a "cat" (+1) or a "dog" (-1).
- **Mathematical Model**:
  - Data represented as a couple $(X, Y) \in (S, \{\pm 1\})$ where S is a measurable space equipped with a σ-algebra $\mathbb{S}$.
    - X: represents "features" (e.g., pixels of a grayscale image).
    - Y: a binary label such as "+1" for dog, "-1" for cat.
- **Statistical Learning**: assume that (X, Y) is randomly chosen from according to some distribution P (e.g., sampled from some database/population), allowing us to use the machinery of probability theory.
- **Prediction rule or a binary classifier**: a measurable function $g : S \to \{-1, +1\}$.
- **Base class:** a collection G of classifiers (can be finite or infinite).
- **Training Data**: a collection $(X_1, Y_1), \ldots, (X_n, Y_n)$ of independent copies of (X, Y) (e.g., images chosen at random from a large collection).
- **Algorithm**: a measurable map that takes training data as input and outputs a prediction rule g in G.
- **Generalization error** or **Classification error** of a classifier g:
  - $L(g) = \Pr(Y \neq g(X))$.
- **Two Scenarios** that the classification problems are commonly divided into:
  - **(A) Realizable learning**: there exists $g_* \in G$ such that $Y = g(X)$ with probability 1.
  - **(B) Agnostic learning**: there is no g_* in G such that $Y = g(X)$ with probability 1.

  Oftentimes, we will just talk about "learning," without making distinction between realizable and agnostic cases.

1.3 Realizable learning with finite base classes

Here, we consider the simplest learning problem. Suppose that G is finite, in other words, the cardinality of G, denoted $|G|$, satisfies $|G| < \infty$. Furthermore, assume the framework of realizable learning. Given the training data, how can one find a binary classifier g such that $L(g)$ is small? Note that we only have access to the training data, and not the distribution P (equivalently, the whole "population"), hence we cannot expect to find g_* in general. Instead, we aim to find some data-dependent $\hat{g}_n$ that has small generalization error with high probability over the choice of training data.

- **Key Questions**:
  - How to find $\hat{g}_n$?
  - How does performance of $\hat{g}_n$ depend on the amount of data?
- **Empirical Risk Minimization (ERM) principle**:
  - Choose $\hat{g}_n$ such that $\hat{g}_n(X_j) = Y_j$ for all $j \leq n$ (why does it always exist?)
  - Alternatively,
$$\hat{g}_n \in \operatorname*{argmin}_{g \in G} \frac{1}{n} \sum_{j=1}^{n} I\{Y_j \neq g(X_j)\}$$
where $I\{\cdot\}$ is the indicator function, i.e.
$$I\{y \neq g(x)\} = \begin{cases} 1, & y \neq g(x), \\ 0, & y = g(x) \end{cases}.$$

Let $\varepsilon \in (0,1)$, and define $G_\varepsilon := \{g \in G : \ L(g) > \varepsilon\}$. Note that

$$\Pr(L(\hat{g}_n > \varepsilon) = \Pr(\hat{g}_n \in G_\epsilon) \leq \sum_{g \in G_\epsilon} \Pr(\hat{g}_n = g).$$

At the same time, $\Pr(\hat{g}_n = g) = \prod_{j=1}^{n} \Pr(g(X_j) = Y_j)$, hence

$$\Pr(L(\hat{g}_n) \geq \epsilon) \leq |G|(1-\epsilon)^n \leq |G| e^{-n\epsilon}.$$

In other words, to guarantee that $\Pr(L(\hat{g}_n > \varepsilon) \leq \delta$, it suffices that the size of the training data satisfies

$$n \geq \frac{\log(\frac{|G|}{\delta})}{\epsilon}.$$

The smallest n that guarantees the inequality $\Pr(L(\hat{g}_n > \varepsilon) \leq \delta$ is called the *sample complexity*. Note that in our derivations, we did not make *any* assumptions regarding the nature of the distribution of X (such as multivariate normality).

1.4 Binary classification without the realizability assumption learning

Now we will consider the more general statistical learning framework and will try to understand what happens when the labels are allowed to be "noisy" (i.e. the distribution P is not concentrated on a diagonal).

Let $(S, \mathcal{S})$ be a measurable space, and let $(X, Y) \in S \times \{\pm 1\}$ be a random couple with distribution P. We will use Π to denote the marginal distribution of X. Recall that X is an instance and Y is a (binary) label. The quality of a classifier is measured in terms of its *generalization error*

$$L(g) := \Pr\left(Y \neq g(X)\right) = \mathbb{E} I\{Y \neq g(X)\}.$$

If we further introduce the *binary loss function* $\ell_{0-1}(y,u) := I\{y \neq u\}$, then we can write

$$L(g) = \mathbb{E}\ell_{0-1}(Y, g(X)).$$

It will be useful to think about generalization error in terms of a loss function since we will consider more general "losses" later in the course.
An important object related to the problem is the so-called *regression function*

$$\eta(x) := \mathbb{E}(Y\,|X = x).$$

Note that $\eta(x) \in [-1, 1]$.

It turns out that one can find the general expression for the optimal classification rule that achieves minimal possible generalization error over all binary classifiers. We will call it the Bayes classifier and denote it via g_*, and its generalization error – the *Bayes risk* L_*. Moreover, it is easy to show that we can take

$$g_*(x) = \text{sign}(\eta(x)),$$

and

$$L_* = \int_S \frac{1 - |\eta(x)|}{2} \Pi(dx).$$

Indeed, since

$$\mathbb{E}(Y\,|X = x) = 1 \cdot \Pr(Y = 1|X = x) + (-1) \cdot \Pr(Y = -1|X = x),$$

and

$$\Pr(Y = 1|X = x) + \Pr(Y = -1|X = x) = 1,$$

we have that

$$\Pr(Y = 1|X = x) = \frac{1 + 1 \cdot \eta(x)}{2},$$
$$\Pr(Y = -1|X = x) = \frac{1 + (-1) \cdot \eta(x)}{2},$$

or $\Pr(Y = t|X = x) = \frac{1+t\cdot\eta(x)}{2}$. Hence, for any binary classifier g

$$\begin{aligned} L(g) &= P(Y \neq g(X)) = \mathbb{E}I\{Y \neq g(X)\} = \mathbb{E}\mathbb{E}\left[I\{Y \neq g(X)\}|X\right] \\ &= \mathbb{E}P(Y \neq g(x)|X = x) = \int_S \frac{1 - g(x)\eta(x)}{2} \Pi(dx) \geq \int_S \frac{1 - |\eta(x)|}{2} \Pi(dx), \end{aligned} \tag{1.1}$$

and the equality is attained precisely for $g = g_*$. Next, define the *excess risk* to be

$$\mathcal{E}(g) := L(g) - L_*.$$

Using (1.1), it can be expressed as (**exercise!**)

$$\mathcal{E}(g) = \int\limits_{\{x:g(x)\neq g_*(x)\}} |\eta(x)|\Pi(dx).$$

Moreover, one can obtain (**exercise!**) the following useful upper bound for the special case when $g(x) = \text{sign}(\hat{\eta}(x))$ for some function $\hat{\eta} : S \mapsto \mathbb{R}$:

$$\mathcal{E}(g) \leq \int_S |\hat{\eta}(x) - \eta(x)| \Pi(dx) \leq \left(\int_S |\hat{\eta}(x) - \eta(x)|^2 \Pi(dx) \right)^{1/2}$$

The main issue is that the true distribution P (and therefore η) describing the relationship between X and Y is usually unknown, so the Bayes classifier can't be used, and that is where statistical learning techniques become useful. We will start with a "naïve" approach: estimate the unknown regression function η via an estimator $\hat{\eta}$ that is a function of the training data, and use $\text{sign}(\hat{\eta})$) as a binary classifier. If the distribution of X is known (e.g., X is multivariate normal with mean a and covariance Σ), then it is enough to estimate the unknown parameters of the the distribution which can be achieved via standard techniques. However, usually nothing can be said about the distribution of X (or it is hard to model using standard parametric families), and in this case naïve approach fails even for low-to-moderate dimensional X due to the phenomenon known as the "curse of dimensionality."