MLT Lecture 2 — representation overview

Matus Telgarsky

1 Administrivia

- Please register for Piazza.
- Homework 0 due Monday at 3:30pm on gradescope!

2 Concrete machine learning theory examples

Recall that last lecture we gave the following abstract definition of an ML (theory) problem.

Nature gives us a prediction problem from X to Y; a way to obtain data; a notion of coherence between past and future data.

We choose a performance criterion; a family of predictors/models; an algorithm to fit our predictor to data.

Last lecture, we gave the example of online learning and the perceptron algorithm. Here is another.

2.1 Statistical learning theory

Now consider a second example: *statistical learning theory*.

> **Nature** has a distribution $\mathcal{P}$ over (X, Y). It uses this distribution to generate an iid sample $((x_i, y_i))_{i=1}^n$ (the *training set*, which it hands to us), and also to measure our performance in the future. This distribution provides coherence between past and future examples.

Given a predictor $\hat{f} : X \to Y$, our performance criterion is the *(population) risk*

$$\mathcal{R}(\hat{f}) := \mathbb{E}_{\mathcal{P}} \ell(\hat{f}(X), Y),$$

where $\ell : Y \times Y \to \mathbb{R}$ is a *loss function*, for instance

$$\ell_{\mathrm{zo}}(\hat{y}, y) := \mathbb{1}[\hat{y} \neq y] \qquad \text{zero-one/classification loss,}$$
$$\ell_{\log}(\hat{y}, y) := \ln(1 + \exp(-\hat{y} y)) \qquad \text{logistic loss.}$$

To describe these briefly:

- The logistic loss $\ell_{\log}$ (and its multivariate generalization *cross-entropy loss*) is popular with neural networks.
- The classification loss ℓ_{zo} makes more sense if we consider the full risk:

$$\mathcal{R}_{\mathrm{zo}}(\hat{f}) = \mathbb{E}\ell_{\mathrm{zo}}(\hat{f}(X), Y) = \mathbb{E}\mathbb{1}[\hat{f}(X) \neq Y] = \Pr[\hat{f}(X) \neq Y].$$

Our goal is to choose $\hat{f}$ so that $\mathcal{R}(\hat{f})$ is small. However, we do not have access to $\mathcal{R}$!

Instead, we will use the training set $((x_i, y_i))_{i=1}^n$ to construct an *empirical* counterpart to $\mathcal{R}$: namely, the *empirical risk* $\widehat{\mathcal{R}}$, defined as

$$\widehat{\mathcal{R}}(f) := \widehat{\mathbb{E}}\ell(f(X), Y) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i).$$

We can now complete the story of the abstract framework as above. We choose:

1. Our **performance critrion** is $\mathcal{R}$, but we will not (and can not) optimize this directly.
2. Our **family of predictors/models** is some set $\mathcal{F}$; the choice of $\mathcal{F}$ has many consequences on the problem, which we will explain momentarily.
3. Our **optimization/fitting** algorithm will be a meta-algorithm for now, namely the **empirical risk minimization (ERM)** principle: find a $\hat{f} \in \mathcal{F}$ that approximately minimizes $\widehat{\mathcal{R}}$, meaning

$$\widehat{\mathcal{R}}(\hat{f}) \approx \inf_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f).$$

In order to mathematically analyze how well we can do, the standard approach is to decompose $\mathcal{R}$ into a few different terms. For this purpose, let us keep track of *three* different functions:

$\hat{f} \in \mathcal{F}$	function chosen by our algorithm,
$\bar{f} \in \mathcal{F}$	element of $\mathcal{F}$ we compare against,
$\bar{g}$	arbitrary function we compare against.

(It is natural to choose $\bar{f}$ and $\bar{g}$ so that $\mathcal{R}(\bar{f}) \approx \inf_{f\in\mathcal{F}} \mathcal{R}(f)$ and $\mathcal{R}(\bar{g}) \approx \inf_f \mathcal{R}(f)$, where the last infimum is over all (measurable) functions.)

Our goal is for $\mathcal{R}(\hat{f}) - \mathcal{R}(\bar{g})$ to be small. To prove this, we will rewrite this quantity as the following equality (a result of adding 0 three times. . .):

$$\begin{aligned}
\mathcal{R}(\hat{f}) - \mathcal{R}(\bar{g}) \;=\; & \mathcal{R}(\hat{f}) - \widehat{\mathcal{R}}(\hat{f}) && (\square)\\
& + \widehat{\mathcal{R}}(\hat{f}) - \widehat{\mathcal{R}}(\bar{f}) && (\triangle)\\
& + \widehat{\mathcal{R}}(\bar{f}) - \mathcal{R}(\bar{f}) && (\diamond)\\
& + \mathcal{R}(\bar{f}) - \mathcal{R}(\bar{g}). && (\bigstar)
\end{aligned}$$

These quantities are now controlled separately as follows.

Generalization is eq. (□): the error between empirical and (true/population) risk of our algorithm's output $\hat{f}$. Bounding this expression is the topic of part 3 of the course, and is often the sole topic of courses on statistical learning theory. The role of $\mathcal{F}$ here is that as $\mathcal{F}$ increases in complexity (a concept made rigorous in various ways in part 3), this bound increases; on the other hand, holding $\mathcal{F}$ fixed and increasing the sample size n, this quantity goes to 0. (Note, eq. (◇) is a similar, simpler quantity; we'll also control it in part 3.)

Optimization is eq. (△): the gap between the algorithm output $\hat{f}$ and $\bar{f}$ (or any other element of $\mathcal{F}$) as measured by $\widehat{\mathcal{R}}$. This is discussed in part 2 of the course. The penalty of large $\mathcal{F}$ here is *computational*: it takes more work to select a good element of $\mathcal{F}$.

Representation/approximation is eq. (★): the error (in terms of $\mathcal{R}$) incurred simply by choosing $\mathcal{F}$, rather than selecting from all possible functions. This is part 1 of the course, and will be explained momentarily. This quantity shrinks as $\mathcal{F}$ grows.

Remark 2.1. Some of the quantities shrink as $\mathcal{F}$ increases in size, others grow; consequently, it is not simply the case that we can make this quantity large or small in order to decrease $\mathcal{R}(\hat{f}) - \mathcal{R}(\bar{g})$.

The discussion of the size of $\mathcal{F}$ is simplified, and indeed there is a more intricate interplay between the quantities above. We'll discuss this more as the course goes on. ◇

3 Representation/approximation overview

As above, the goal topic of the representation question is to reason about how well our choice of functions/predictors/models $\mathcal{F}$ can compete with some larger set of functions, for instance the set of all continuous functions.

In eq. (★), the quantity of interest was $\mathcal{R}(\bar{f}) - \mathcal{R}(\bar{g})$. Since this depends on the choice of loss function ℓ and the probability distribution generating the data, it is a little complicated. Instead, we will look at two simpler quantities:

$$\sup_{g \text{ cont.}} \inf_{f \in \mathcal{F}} \|f - g\|_1 = \sup_{g \text{ cont.}} \inf_{f \in \mathcal{F}} \int_{[0,1]^d} \left|f(x) - g(x)\right| \mathrm{d}x;$$
$$\sup_{g \text{ cont.}} \inf_{f \in \mathcal{F}} \|f - g\|_{\mathrm{u}} = \sup_{g \text{ cont.}} \inf_{f \in \mathcal{F}} \sup_{x \in [0,1]^d} \left|f(x) - g(x)\right|.$$

As will be fleshed out in the homework, these quantities can be related to $\mathcal{R}(\bar{f}) - \mathcal{R}(\bar{g})$.

Remark 3.1. In class, we discussed flipping the inf and the sup. First, for any sets of functions $\mathcal{F}$ and $\mathcal{G}$, and any pairwise function $H : \mathcal{F} \times \mathcal{G} \to \mathbb{R}$,

$$\begin{aligned} & H(f', g') \le \sup_{g \in \mathcal{G}} H(f', g) && \forall (f', g') \in (\mathcal{F}, \mathcal{G}), \\ \implies \quad & \inf_{f \in \mathcal{F}} H(f, g') \le \inf_{f \in \mathcal{F}} \sup_{g \in \mathcal{G}} H(f, g) && \forall f' \in \mathcal{F} \\ \implies \quad & \sup_{g \in \mathcal{G}} \inf_{f \in \mathcal{F}} H(f, g) \le \inf_{f \in \mathcal{F}} \sup_{g \in \mathcal{G}} H(f, g). \end{aligned}$$

Only under a variety of conditions on $(\mathcal{F}, \mathcal{G}, H)$ is this final inequality an equality (cf. Sion's minimax theorem; note, the full version includes quasiconvex functions, not just convex as in many machine learning texts). ◇

4 Limitations of linear classes

To close this lecture, we'll have our first approximation result, showing that there are situations where linear classifiers are inadequate.

Remark 4.1. "Linear classifier/function" is often ambiguous in machine learning: sometimes it means classifiers (mapping to one of a discrete set of choices), and sometimes it also means *affine* functions; that is to say, "linear" can be broken in two different ways. ◇

Theorem 4.2 (The XOR problem (Minsky and Papert, 1969, Chapter 0))**.** *Define the following objects.*

- *Linear classifiers over* $\mathbb{R}^2$*:* $\mathcal{L} := \{x \mapsto \operatorname{sgn}(a^\top x - b) : (a, b) \in \mathbb{R}^2 \times \mathbb{R}\}$.
- *A set of four datapoints* $S := \{u, v, -u, -v\}$*, where* $u = (+1, +1)$ *and* $v = (-1, +1)$.
- *Define a target function* $\bar{g}(x) := \mathbb{1}[x_1 x_2 = +1]$*, a polynomial of degree 2.*
- *Define a distribution over whose margin over* X *is uniform on* S*, and* $\Pr[Y = g(X)|X] = 1$.

Then

$$\inf_{f \in \mathcal{L}} \mathbb{E}|f(X) - Y| = \inf_{f \in \mathcal{L}} \mathbb{E}|f(X) - \bar{g}(X)| \ge \frac{1}{4} \qquad \text{and} \qquad \inf_{f \in \mathcal{L}} \sup_{x \in S} |f(X) - \bar{g}(X)| = 2.$$

In other words: there are situations where every linear classifier is bad, but a degree 2 polynomial is perfect.

Proof. We'll give two proofs. The first is based on the original Minsky-Papert proof. (For both proofs, it's useful to draw a picture.)

> **Proof #1.** Fix any element $f \in \mathcal{L}$. The decision boundary of f is a straight line, and $H := f^{-1}(\{+1\})$ is a closed halfspace, and $H^c := f^{-1}(\{-1\})$ is an open halfspace. If $\{u, -u\} \subseteq H$, then their convex hull and in particular 0 lie in H, and $f(0) = +1$. On the other hand, if $\{v, -v\} \subseteq H^c$, then their convex hull and in particular 0 lie in H^c, and $f(0) = -1$. In other words, if f is correct on all of S, then $f(0)$ must take on two values, a contradiction; therefore f makes at least one mistake on S.
>
> (Picture version: draw line segments $[u, -u]$ and $[v, -v]$, which form an 'X' through the origin. . .)

Here is another way of writing down the same proof, without a contradiction.

> **Proof #2.** Pick any $(a, b) \in \mathbb{R}^2 \times \mathbb{R}$, and define $f(x) := \operatorname{sgn}(a^\top x - b) \in \mathcal{L}$. If f is correct on $\{u, -u, v\}$, then
>
> $$a^\top u - b \geq 0, \tag{4.1}$$
> $$a^\top (-u) - b \geq 0, \tag{4.2}$$
> $$a^\top v - b < 0. \tag{4.3}$$
>
> Adding together eqs. (4.1) and (4.2), and the negation of eq. (4.3),
>
> $$\begin{aligned} & a^\top (u - u - v) + (-b - b + b) > 0 \\ \implies \qquad & a^\top (-v) - b > 0; \end{aligned}$$
>
> In other words, if f is correct on $\{u, -u, v\}$, then $f(-v) = +1$, which is incorrect; therefore every element of $\mathcal{L}$ makes at least one mistake on S.

The other parts of the statement follow from every $f \in \mathcal{L}$ being wrong on at least one element of S. □

Remark 4.3. Of course, linear classifiers are used all the time. The standard fix is to instead work with the related class of linear combinations of some basis of functions; not only is this the approach taken by SVMs and boosting, but moreover we'll use this when proving one result about neural network representation!

Note furthermore that engineering features and then using a linear classifier is similar. ◇

Remark 4.4. Theorem 4.2 is credited as triggering the first *AI winter*. If you google around, you might find some weird comments about it. ◇

References

Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.