Lecture 1: Syllabus and Overview

Syllabus overview

(First part of lecture is interactive with class, going over course webpage, namely the syllabus except the schedule.)

Philosophy behind course

- The pre-req is math. I will give you the ML. If you do not know how to prove theorems, it will not be easy (use homework 0 to gauge this).
- Utilitarian goal of course: a standard student is comfortable reading learning theory papers and including some theorems in their papers, and an ambitious student is ready to attack learning theory problems.
- **But who cares about theory** (especially when it lags practice).
  - Theory doesn't always lag practice (things look bad right now due to neural nets):
    - Boosting is the result of a theoretical question (by Leslie Valiant).
    - kmeans++ is the default k-means initialization, and started in theory papers.
    - SVMs are from theory.
    - Regularization.
    - Averaging/voting used everywhere with neural networks.
    - Original "neural network" paper is a math paper by a logician and a neuroscientist.
  - Sometimes theory gives algorithms that are then used beyond their guarantees (or guides heuristics).
    - AdaGrad can be derived purely by optimizing a bound; meanwhile, it is successful far beyond its analysis (namely in non-convex settings).
    - Boosted decision trees request near-optimal trees, but instead a heuristic tree solver is used.
    - Sometimes theory is loose, but it still allows some comparisons of methods.
  - *But theory isn't just about algorithms.* Theory gives a framework to reason about algorithms. Thinking about algorithms is fun.
  - Theory and practice share some caveats: experiments can be overly tuned to some data, and theory can look overly good due to a (sometimes sneakily!) specialized setting!
- **Practice cast as theory.** (Lecture will include a drawing! Imagine a function mapping each exact configuration of an algorithm to an error. (Exact configuration means software, hardware, dataset... everything to give reproducibility.) Theory gives estimates of this function in chunks of the space. Meanwhile, experiments give the value of the function at individual points. Due to advances in hardware and also research dissemination, one has easy access to many such points, and it is tempting to extrapolate general behavior...)

  The red dots are for individual runs (experiments converted into theorems); all the hardware being thrown at machine learning means we get a fine granularity estimate, but the gaps are still meaningful... the red bars are lower and upper bounds given by theory; they are loose, but they are not single points.

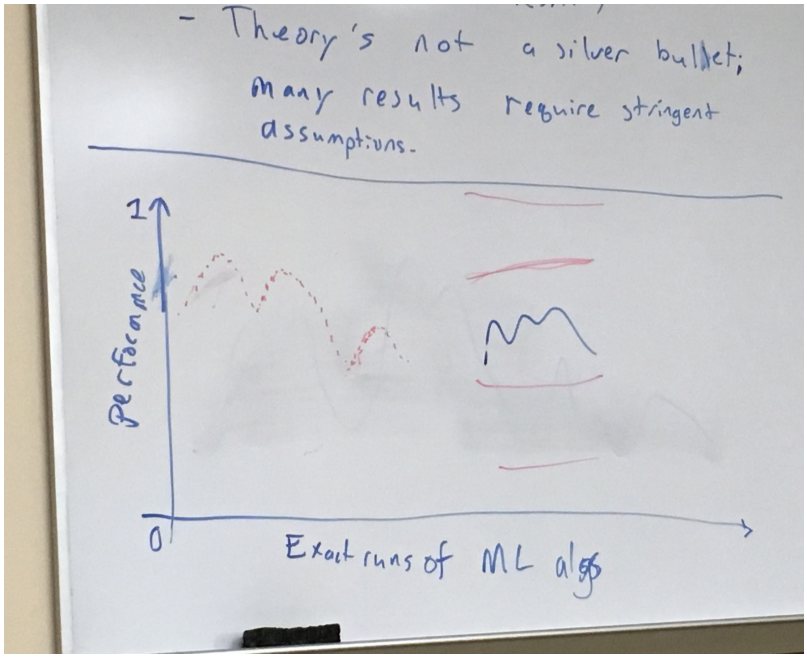

Figure 1: Thanks to Konstantinos for the image!

- **But what should we prove?**
  - There are infinitely many true statements; the collection people care about is down to human taste, culture, coincidence, ...
  - ML has an endless number of open problems! We'll even get some in this lecture.

Machine learning theory overview

Machine learning is: using data to tune algorithms.

Example (linear regression). Suppose our goal is to predict $y \in \mathbb{R}$ given $x \in \mathbb{R}^d$.

- **Representation choice:** let's pick $w \in \mathbb{R}^d$ and predict according to $w^\top x$. Next lecture tells us how much this choice costs us.
- How to tune this method from data? The model of **Statistical Learning Theory** is to assume $((x_i, y_i))_{i=1}^n$ are drawn IID from some distribution, and future examples are drawn from the marginal distribution over x.

  We still need some way to prefer one w to others! A common choice is the **least squares error** $(w^\top x - y)^2$. So we can find a $\hat{w}$ on the data:

  $$\hat{w} := \text{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^n (w^\top x_i - y_i)^2,$$

  but the error we really care about is $\mathbb{E}(\hat{w}^\top x - y)^2$. We can split our analysis into three separate questions:

  - **Representation:** what we lost by choosing functional relationship $x \mapsto w^\top x$ (as opposed to any other function).

  - **Optimization:** how well we can solve the argmin.
  - **Generalization:** how well we do on the expected error (which we do not observe directly).

The main part of the schedule is roughly split into these three issues. Note that we've ignored the question "*why use least squares error*"!

- There are many other models!
  - **Online learning.** Suppose we have a sequential process, where in each round we receive an x_t, predict a $\hat{y}_t$, then observe a y_t, adjust our predictor, and repeat.
  - **Active learning.** Suppose we can *choose* the sequence $(x_i)_{i=1}^t$, and are revealed the $(y_i)_{i=1}^t$; sometimes this allows the same speedups that binary search offers over sequential search.
  - **Unsupervised learning.** If y_i is never revealed, the only choice is to look for structure in x_i alone. One idea is to fit two gaussians to $(x_i)_{i=1}^n$ and classify new points according to the gaussian they fall in, but this is making many assumptions! Moreover, the performance metric isn't clear.
  - etc....

Schedule overview

(Schedule then discussed, should make some more sense now.)

A quick proof

Note: a *tremendous* amount of detail was omitted in lecture; this material will only appear on homeworks in a few weeks, once it has been presented properly in the "optimization section". In other words, **this material is optional**!

This proof will continue the above example: the statistical learning model, using linear regression. This means we get an IID sample $((x_i, y_i))_{i=1}^t$, and must predict well on future examples from the same distribution. We'll sketch a derivation that should look natural, and at the end state the theorem it gives.

Regarding the three questions above: "representation" will be discussed in the next class, but we must contend here with optimization and generalization. We will cheat:

- We will use the **stochastic online learning** setting, where we process each of the t examples exactly once in a stream. This will allow us to minimize the error over the distribution directly rather than separately optimizing on a training set, and then using generalization bounds to argue something over the true distribution.

The algorithm will be given momentarily, and makes use of the following objects:

- There is a compact convex constraint set $B \subseteq \mathbb{R}^d$ within which every iterate will lie.
- There is a reference vector $u \in B$ which *need not be optimal*.
- There is a step size η which scales the stochastic gradient iteration. We will solve for η at the end. It's more common to have η decrease over time, but we are making it constant for ease (later lectures will allow it to vary).
- We need to formalize an *objective function*, namely the function we tune/minimize. First, for a fixed example (x, y) and weight vector w,

$$\ell(w, x, y) := \frac{1}{2}(x^\top w - y)^2 \qquad \text{loss},$$

$$\mathcal{R}(w) := \mathbb{E}(\ell(w, X, Y)) = \frac{1}{2}\mathbb{E}(X^\top w - Y)^2 \quad \text{risk}.$$

ℓ and $\mathcal{R}$ are convex in w; there will be an entire lecture (or two) devoted to convexity, but for now this means two things for us: for any vectors w and w', and any set of vectors $(v_i)_{i=1}^m$ and weights $(\alpha_i)_{i=1}^m$ with $\sum_i \alpha_i = 1$ and $\alpha_i \geq 0$,

$$\begin{aligned} \mathcal{R}(w') &\geq \mathcal{R}(w) + \left\langle \nabla \mathcal{R}(w), w' - w \right\rangle && \text{first-order convexity,} \\ \sum_i \alpha_i \mathcal{R}(v_i) &\geq \mathcal{R}\left(\sum_i \alpha_i v_i \right) && \text{Jensen's inequality.} \end{aligned}$$

The stochastic gradient iteration has $w_0 \in B$ arbitrary, and thereafter

$$\begin{aligned} w_{i+1} &:= \Pi_B(w_i - \eta \hat{g}_{i+1}) \\ &= \Pi_B(w_i - \eta(\nabla \mathcal{R}(w_i) + \varepsilon_{i+1})), \end{aligned}$$

where

- Π_B denotes orthogonal projection onto B (we'll analyze this in a homework);
- $\hat{g}_{i+1}$ is a *stochastic gradient* at time $i+1$, meaning it must satisfy $\mathbb{E}(\hat{g}_{i+1}) = \nabla \mathcal{R}(w_i)$, which is easy to accomplish: we can define $\hat{g}_{i+1} = \ell'(w_i, x_{i+1}, y_{i+1}) = x_{i+1}(x_{i+1}^\top w_i - y_{i+1})$;
- we'll assume there exists $R < \infty$ with $\|v\| \leq R$ for $v \in B$;
- we'll assume there exists $G < \infty$ with $\|\hat{g}_{i+1}\| \leq G$ and $\|\nabla \mathcal{R}(w_i)\| \leq G$; this is fine when a bound exists on X and Y, since (by Cauchy-Schwartz and triangle inequality)

  $$\|\hat{g}_t\| = \|x_t(x_t^\top w_t - y_t)\| \leq \|x_t(x_t^\top w_t)\| + \|x_t y_t\| \leq |x_t^\top w_t| \|x_t\| + |y_t| \|x_t\| \leq \|x_t\|^2 \|w_t\| + |y_t| \|x_t\|.$$

  Controlling $\nabla \mathcal{R}$ is similar;
- $\varepsilon_{i+1} := \hat{g}_{i+1} - \mathcal{R}(w_i)$; by definition, $\mathbb{E}(\varepsilon_{i+1}) = 0$, and this rewriting of things as "essential object + noise" is very useful in many problems.

Remark. Many of these assumptions are fine when B is compact and X, Y also lie in compact sets (almost surely). But those compactness assumptions themselves are often false in practice.

To analyze this, imagine that u is very good and unique; we'd like $\|w_i - u\|$ to become small. Indeed (without any assumptions on u except $u \in B$):

$$\begin{aligned} \|w_{i+1} - u\|^2 &= \|\Pi_B(w_i - \eta \hat{g}_{i+1}) - u\|^2 \\ &\leq \|w_i - \eta \hat{g}_{i+1} - u\|^2 && \text{(see below)} \\ &= \|w_i - u\|^2 + 2\eta \left\langle \hat{g}_{i+1}, u - w_i \right\rangle + \eta^2 \|\hat{g}_{i+1}\|^2. \\ &= \|w_i - u\|^2 + 2\eta \left\langle (\nabla_w \mathcal{R})(w_i), u - w_i \right\rangle + 2\eta \left\langle \varepsilon_{i+1}, u - w_i \right\rangle + \eta^2 G^2. \\ &\leq \|w_i - u\|^2 + 2\eta(\mathcal{R}(u) - \mathcal{R}(w_i)) + 2\eta \left\langle \varepsilon_{i+1}, u - w_i \right\rangle + \eta^2 G^2. \end{aligned}$$

where the orthogonal projection was removed essentially due to pythagorean theorem (we'll do this in detail in a homework), and the last step used the first-order convexity inequality. Rearranging and summing over $i \in \{0, \ldots, t-1\}$, and then applying Jensen's inequality,

$$\begin{aligned} \|w_0 - u\|^2 + 2\eta \sum_{i=0}^{t-1} \left\langle \varepsilon_{i+1}, u - w_i \right\rangle + t\eta^2 G^2 &\geq 2\eta \sum_{i=0}^{t-1} \big(\mathcal{R}(w_i) - \mathcal{R}(u) \big) \\ &= 2\eta t \sum_{i=0}^{t-1} \frac{1}{t} \big(\mathcal{R}(w_i) - \mathcal{R}(u) \big) \\ &\geq 2\eta t \big(\mathcal{R}(\bar{w}_{t-1}) - \mathcal{R}(u) \big) \end{aligned}$$

where $\bar{w}_{t-1} = \sum_{i=0}^{t-1} w_i/t$ is the *averaged iterate*, which sometimes appears in practice (though not commonly). Noting $w_0 \in B$ implies $\|u - w_0\| \leq 2R$ and dividing by $2\eta t$ gives

$$\mathcal{R}(\bar{w}_{t-1}) - \mathcal{R}(u) \leq \frac{4R^2}{\eta t} + \frac{1}{t}\sum_{i=0}^{t-1} \langle \varepsilon_{i+1}, u - w_i \rangle + \eta G^2.$$

To simplify further, consider the middle term of the right hand side. Since $\mathbb{E}(\varepsilon_{i+1}) = 0$, it seems there should be a way to make the entire expression small. But the two sides of the inner product are *not* independent, we can't just wrap everything in an expectation. The trick is to note that the left hand side is independent of the right when *given* the right, and that this conditional expectation of ε_{i+1} is also zero:

$$\begin{aligned}\mathbb{E}(\langle \varepsilon_{i+1}, u - w_i \rangle) &= \mathbb{E}(\mathbb{E}(\langle \varepsilon_{i+1}, u - w_i \rangle \mid ((x_1, y_1), \ldots, (x_i, y_i)))) \\ &= \mathbb{E}(\big\langle \mathbb{E}(\varepsilon_{i+1} \mid ((x_1, y_1), \ldots, (x_i, y_i)), u - w_i \big\rangle)) = 0.\end{aligned}$$

This gives

$$\mathbb{E}(\mathcal{R}(\bar{w}_{t-1}) - \mathcal{R}(u)) \leq \frac{4R^2}{\eta t} + \eta G^2.$$

The last step is to optimize for η; $1/\sqrt{t}$ is fine, but the best choice (minimizing the bound) is $2R/(G\sqrt{t})$, which gives the following.

Theorem. Suppose $w_0 = 0$, and otherwise w_i is produced by the sgd iteration with $\eta = 2R/(G\sqrt{t})$. Then

$$\mathbb{E}(\mathcal{R}(\bar{w}_{t-1}) - \mathcal{R}(u)) \leq \frac{4RG}{\sqrt{t}},$$

where the expectation is over the training set $((x_i, y_i))_{i=1}^t$.

Remarks.

- What we really want is a bound that holds with high probability, not in expectation. For this, we can use Azuma's inequality on the term with ε_{i+1}, which makes it $\mathcal{O}(RG\sqrt{t})$. We'll discuss this more.
- The optimal choice of η requires problem-dependant constants; this hints why in practice there is still a big mess when tuning η.

Open problems. Here are a few suggested by the lecture.

- **This problem recommended to me by Alekh Agarwal of MSR NYC.** The stochastic gradient iteration processes t IID examples. What if one more iteration is executed, but on some example that was already seen? The error analysis breaks. This is a big gap between theory in practice: in practice, people do multiple sgd passes, which breaks the theory. This is very sensitive, some algorithms really do break, others don't..
- Why squared loss? The best paper I know here is due to Zhang (2004). This is an *amazing* paper. There is also a very nice follow-up by Bartlett, Jordan, and McAuliffe (2006), indeed many cite only the second paper and aren't aware of the first, but while the second paper is more general, it loses the detailed loss-specific analysis!

References

Bartlett, Peter L., Michael I. Jordan, and Jon D. McAuliffe. 2006. "Convexity, Classification, and Risk Bounds." *Journal of the American Statistical Association* 101 (473): 138–56.

Zhang, Tong. 2004. "Statistical Behavior and Consistency of Classification Methods Based on Convex Risk Minimization." *The Annals of Statistics* 32: 56–85.