MLT Lecture 1 — course and topic overview

Matus Telgarsky

1 Administrivia

- **Topic:** proof-oriented investigation of machine learning.
- **Goal:** everyone taking it can read and produce MLT research.
- **Meta-principles:** no busy-work (project as hard as you make it); mutual respect (you come on time, I end on time).
- **Webpage** `http://mjt.cs.illinois.edu/courses/mlt-f18/` (alternatively google me and follow links): here you can find everything you need (homeworks, grading policy, etc). *[In class I went over the main points on the webpage.]*
- **Other ML Theory courses:** this one differs by 1. broader array of topics (not just statistical learning, online learning, unsupervised learning); 2. discussing representation 3. neural networks (others catching up and surpassing now. . .).
  - **v1 vs v2:** 1. more material on neural nets; 2. more material outside the standard "statistical learning theory" framework.
  - **v2 vs v3:** 1. online learning; 2. k-nn; 3. bandits/RL.
- **Hw0 due on September 3, at the start of class (3:30pm).**
- *[We also did a poll of how lectures should be substituted during my travel. There was a slight preference for those lectures to go on youtube, rather than a day being doubled-up.]*

2 What is ML Theory?

- **What is ML?**

  Machine learning adapts algorithms to data.

  The definition broad because ML is not simply:

  spam filtering; choosing treatments for illnesses; grouping genes by function.

  ML is also (e.g., since neural nets are Turing complete!):

  solving traveling salesman instances (with neural nets. . . I'm serious); playing go, chess, DOTA. . . driving cars.

- TCS is design and analysis of algorithms; *ML Theory is design and analysis of **ML** algorithms.*

  We care about some of the same things you see in pure TCS: time complexity, space complexity,. . .

  We also care about: **sample complexity**, **label complexity**, . . .

  Remark 2.1. Machine learning excels at *noisy data* and *handling the average case*; by contrast, TCS often focuses on the worst case, giving TSP hope here. ◇

Remark 2.2. ML *also* cares about worst case; e.g., "adversarial examples". ◇

- ***Why* ML Theory?**

  Optimistic application-oriented view of MLT:

  > Some ML algs and ideas had their origins in theory, even if they are used beyond the theoretical analysis: sgd, adagrad, boosting, regularization, neural nets...

  Pessimistic application-oriented view of MLT:

  – Most ML algs, however, seem to have no involvement from theory (at any stage in their genesis).

  This seems sad. Wouldn't we like to say "all this hard work gets us better algorithms"? So... why do ML theory?

  – We seek understanding. *[IMO, this is the reason for the increased recent interest in deep learning theory.]*

  – It can give a fresh look that leads to a new idea, even if the new idea isn't fully backed by theory.

3 Abstract formalization of MLT

An abstract definition is as follows.

Nature provides us with a prediction problem mapping elements of X to Y (e.g., marking emails as spam or not; selecting chess moves given a current chess game); nature also gives us a way to obtain data (e.g., obtaining and labeling emails; obtaining human chess games or playing games against ourselves); lastly nature specifies some *coherence* between past and future prediction instances (e.g., the spam may follow a specific distribution, or spammers are lazy and hardly change; we play against humans of a similar level).

We then choose (or are given) a performance criterion (e.g., whether we correctly classified a spam message); a family of predictors/models (e.g., linear predictors or neural networks); an algorithm to fit the predictors to data (e.g., perceptron or sgd).

Remark 3.1. "Coherence" is not a standard term. It is just my way of saying: things don't change too drastically. Soon we'll see examples of prediction problems where, without coherence nature can force us to perform as badly as possible. All learning setups have some notion of "coherence". ◇

4 Concrete ML Theory examples

4.1 Online learning and the perceptron algorithm

Nature provides us the following prediction problem:

For $i \geq 1$:

1. Nature gives us $x_i \in \mathbb{R}^d$.
2. We output $\hat{y}_i \in \{-1, +1\}$.
3. Nature choose $y_i \in \{-1, +1\}$ (seeing our $\hat{y}_i$!).
4. We suffer a loss $\mathbb{1}[\hat{y}_i \neq y_i]$.
   (We may now update our model, using x_i and y_i.)

Remark 4.1. Since nature sees our $\hat{y}_i$, it can always choose $y_i = -\hat{y}_i$, whereby $\mathbb{1}[\hat{y}_i \neq y_i] = 1$ and the cumulative loss is always maximal, regardless of the prediction algorithm. This justifies the need for "coherence". ◇

The *perceptron algorithm* is as follows:

1. Initialize with $w_0 := 0$.
2. Thereafter, recursively set
$$w_i := w_{i-1} + x_i y_i \mathbb{1}\left[\left\langle w_{i-1}, x_i y_i \right\rangle \leq 0\right],$$
meaning we rotate w_i towards $x_i y_i$ when we are mistaken.

Remark 4.2. The update is an sgd step with the ReLU loss
$$w_i = w_{i-1} - \partial_w \curlywedge\left(\left\langle w_{i-1}, x_i y_i \right\rangle\right),$$
where the ReLU is the map $\curlywedge(z) := \max\{0, z\}$. But w_0 is the global optimum; why are we iterating?... ◇

The guarantee we will eventually prove for this problem is as follows.

Theorem 4.3 (Novikoff (1962))**.** *Let $\hat{y}_i, y_i, and \gamma$ be defined as above in the Perceptron algorithm, and suppose there exists a unit vector $\bar{u}$ (meaning $\|\bar{u}\| = 1$) with $\gamma := \inf_i \left\langle \bar{u}, x_i y_i \right\rangle > 0$. Then*
$$\sum_{i \geq 1} \mathbb{1}[y_i \neq \hat{y}_i] \leq \frac{1}{\gamma^2}.$$

Remark 4.4. The assumed vector $\bar{u}$ provides our coherence condition: it provides a constraint on the labels of future examples, given the labels of past examples (namely, there must always be a $\bar{u}$ which separates their union with a margin γ). *[In class, pictures were drawn to explain this assumption and the margin.]* ◇

Remark 4.5. The idea of the algorithm is that each mistake rotates us towards a good vector. Consequently, it is unsurprising that our proof will proceed by bounding
$$\left\| \frac{w_t}{\|w_t\|} - \bar{u} \right\|^2.$$

◇

4.2 Statistical learning theory

As a second key example, consider the setting of *statistical learning theory*: nature has some underlying distribution $\mathcal{P}$, from which it provides us an iid sample $((x_i, y_i))_{i=1}^n$ (our training set), and then uses this same distribution to rate our performance in the future.

[We discussed this only a little bit, and will pick up here next time.]

References

Albert B.J. Novikoff. On convergence proofs on perceptrons. *In Proceedings of the Symposium on the Mathematical Theory of Automata*, 12:615–622, 1962.