COS324: Introduction to Machine Learning

Lecture 4: PAC Learning - Part I

Prof. Elad Hazan & Prof. Yoram Singer

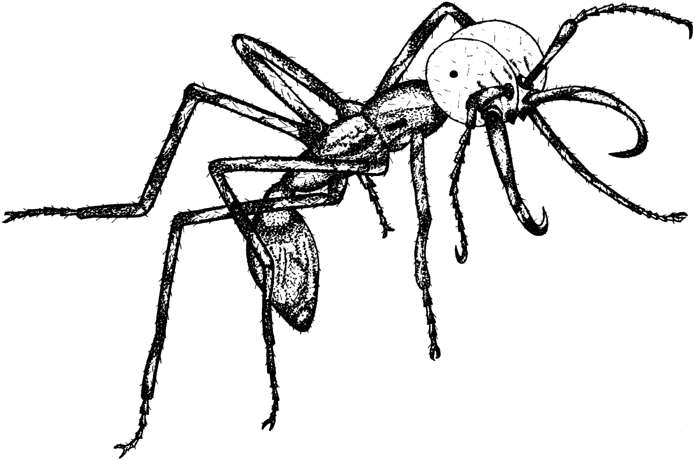
Recap & Today

- So far:
  1. Online decision making and online learning
  2. (Randomized) Weighted Majority for tracking the best expert(s)
  3. Decision making: Kelly criterion

- Going forward:
  1. Batch (offline) learning: setting and defintions
  2. Probably Approximately Correct (PAC) model
  3. PAC learning in noiseless and noisy settting
  4. Efficient PAC learning

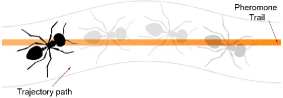
- After PAC – back to learning algorithms:
  1. Convex analysis
  2. Efficient learning algorithms, this time for real

Let's talk about ants

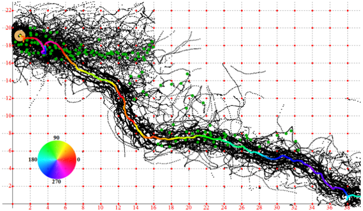


Predicting Ant's Next Move

- Predict next move direction of an ant $\rightarrow \leftarrow \uparrow \downarrow$
- Build elaborate model of sense antenas and olfactory system



- $\Rightarrow$ Potentially good understanding albeit poor predictability
- Record many traces & features (dampness, season, ...)



- $\Rightarrow$ Potentially good predictability albeit poor understanding

Chomsky vs. Jelinek

- Noam Chomsky: modern linguist, *universal grammar* theory, *generative grammar* theory, the Chomsky hierarchy
- At Brains, Minds, and Machines symposium Chomsky derided machine learning research for using statistical learning methods to produce behavior that mimics something in the world, but who don't try to understand the meaning of that behavior.
- Fred Jelinek: EE training from MIT, head of (famous) IBM team for research in speech recognition and machine translation
- "Every time I fire a linguist, the performance of the speech recognizer goes up."

High-Low game

- Goal: predict whether NFL player had > 2 concussions based on his level of Chronic Traumatic Encephalopathy (CTE)
- Input: pairs of (CTE-level,#concussions> 2)

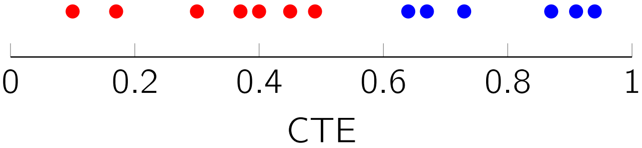
$$(0.07, -)(0.9, +)(0.4, +)(0.73, +)(0.2, -)$$

- Output: prediction rule $\mathrm{sign}[\mathrm{cte} - \theta]$
- "Learning algorithm":
  - Set
  $$\hat{\theta} = \min\{x^i : y^i = +1\}$$
  - Set
  $$\hat{\theta} = \max\{x^i : y^i = -1\}$$
  - Set
  $$\hat{\theta} = \frac{1}{2}\left(\min\{x^i : y^i = +1\} + \max\{x^i : y^i = -1\}\right)$$

Error Analysis - I

- Range of CTE level is $[0, 1]$
- Distribution of CTE levels $\mathcal{D}$ is uniform over $[0, 1]$
- $\exists \theta^\star$ s.t. $y^i = \text{sign}(x^i - \theta^\star)$
- Use the first rule $\hat{\theta} = \min\{x^i \,:\, y^i = +1\}$
- Denote $\tilde{\theta} = \max\{x^i \,:\, y^i = -1\}$
- "Noman land" $(\tilde{\theta}, \hat{\theta})$
- Samples are identically distributed and independent

Example



Error Analysis - II

- How likely a prediction using $\hat{\theta}$ would wrong w.p. $> \varepsilon$ on *unseen* samples?
- A single sample falls outside no man's land w.p.

$$1 - (\hat{\theta} - \tilde{\theta}) \leq 1 - \varepsilon$$

- Set of m i.i.d samples all fall outside no man's land w.p.

$$(1 - \varepsilon)^m \leq e^{-\varepsilon m}$$

- Suppose we want that $\varepsilon \leq 0.001$ and be at least 99% sure about that we found an accurate threshold
- Then, $e^{-\varepsilon m} \leq \delta$ where $\varepsilon = 0.001$ and $\delta = 0.01$
- Number of samples suffices to be

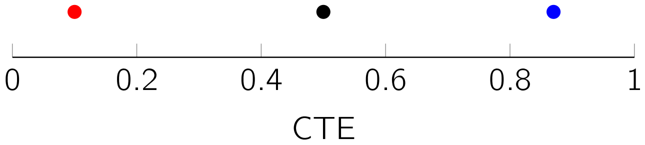
$$m \geq \frac{1}{\varepsilon} \log\left(\frac{1}{\delta}\right) = 1000 \times \log(100) \approx 4605$$

Adversial High-Low Game

- Online setting $x^t \rightsquigarrow \hat{y}^t = \mathrm{sign}(x^t - \theta^t) \rightsquigarrow y^t \rightsquigarrow \theta^{t+1}$
- Instead of i.i.d. samples, assume $\exists$adversy:
  - Remembers past examples

$$a = \max_{y^i=-1} x^i \quad ; \quad b = \min_{y^i=+1} x^i$$

  - Picks next $x^t = \frac{1}{2}(a+b)$
  - Gets $\hat{y}^t$ and sets $y^t = -\hat{y}^t$



Batch (Offline) Learning

- Instance domain: $\mathcal{X}$ (e.g. $\mathcal{X} = \mathbb{R}^d$)
- Target (label) domain: $\mathcal{Y}$ (e.g. $\mathcal{Y} = \{-1, +1\}$)
- Learner's input – training data:

$$S = \left\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^m, y^m)\right\} \in (\mathcal{X} \times \mathcal{Y})^m$$

- Learner's output – predictor / classifier:

$$h : \mathcal{X} \to \mathcal{Y} \quad [\hat{y} = h(\mathbf{x})]$$

Batch (Offline) Learning

- Instance domain: $\mathcal{X}$ (e.g. $\mathcal{X} = \mathbb{R}^d$)
- Target (label) domain: $\mathcal{Y}$ (e.g. $\mathcal{Y} = \{-1, +1\}$)
- Learner's input – training data:

$$S = \{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^m, y^m)\} \in (\mathcal{X} \times \mathcal{Y})^m$$

- Learner's output – predictor / classifier:

$$h : \mathcal{X} \to \mathcal{Y} \quad [\hat{y} = h(\mathbf{x})]$$

- What should be the learning goal?
- Find h with (mostly) correct predictions on unseen examples

Data Model Assumptions

- Must assume relation between training and test (unseen) data
- Assumptions need to be realistic & enable *efficient* learning
- Statistical assumption, not adversarial, not "time dependent"
- Examples are Independently Identically Distributed (i.i.d.):

$$\mathbf{x}^i \sim \mathcal{D} \;\Rightarrow\; \mathcal{D}(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m) = \prod_{i=1}^{m} \mathcal{D}(\mathbf{x}^i)$$

- For instance:

$$\text{Dice} \;\; \mathcal{D}(\mathbf{x}^1 = i^1, \ldots, \mathbf{x}^m = i^m) = (1/6)^{-m}$$

$$\text{Hypercube} \;\; \mathcal{X} = \{-1, +1\}^n \;\; \mathcal{D}[\mathbf{b}] = 2^{-n}$$

$$\text{Normal} \;\; \mathcal{X} = \mathbb{R}^n \;\; \mathcal{D}(\mathbf{x}) = (2\pi)^{-n/2} \exp(-1/2\|\mathbf{x}\|^2)$$

Realizability

- If $\mathbf{x}$ is R.V. what about y ?
- General case, uncertainty in $Y = y$ even when we observe $X = \mathbf{x}$

$$\mathcal{D}(X, Y)$$

- Realizable case:

$$\mathcal{D}(Y = +1 | X = \mathbf{x}) = 1 \text{ or } \mathcal{D}(Y = -1 | X = \mathbf{x}) = 1$$

- We can define

$$h^\star(\mathbf{x}) = \text{sign}\left(\mathcal{D}(Y = +1 | \mathbf{x}) - \frac{1}{2}\right)$$

Unseen Examples

- Let $h^\star$ be the (unknown) correct classifier
- We should find h s.t. $h \approx h^\star$
- Define error of h w.r.t. $h^\star$ to be

$$\mathcal{L}_{\mathcal{D}}(h) = \mathbb{P}_{x \sim \mathcal{D}}\left[h(x) \neq h^\star(x)\right]$$

  where $\mathcal{D}$ is some (unknown) probability measure over $\mathcal{X}$

- Is it possible to find h s.t. $\mathcal{L}_{\mathcal{D}}(h)$ is arbitrarily small ?

Only Approximately

- Claim: We cannot hope to find h s.t. $\mathcal{L}_{\mathcal{D}}(h) = 0$
- Proof: for every $\varepsilon \in (0, 1)$ set

$$\mathcal{X} = \{0, 1\} \quad \mathcal{D}(0) = 1 - \varepsilon \quad \mathcal{D}(1) = \varepsilon$$

- Probability not to see $\mathbf{x} = 1$ among m i.i.d. samples is

$$(1 - \varepsilon)^m \approx e^{-\varepsilon m}$$

- If $m < 2/\varepsilon$ we won't observe $\mathbf{x}^i = 1$ w.p. of $e^{-2} > 10\%$
- We can only guess the label when $\mathbf{x} = 1$
- Relaxed Goal: find h s.t. $\mathcal{L}_{\mathcal{D}}(h) \leq \varepsilon$ for a pre-specified ε

Only Probably

- Examples are generated according to an i.i.d. process
- We may obtain the same (or very similar) sample over and over
- No algorithm can guarantee $\mathcal{L}_{\mathcal{D}}(h) \leq \varepsilon$ for sure
- Relaxed goal: learning algorithm may fail (completely) w.p. δ
  (Probability is over random choices of examples)

Probably **A**pproximately **C**orrect (PAC)

- Learning algorithm (learner) does not know $\mathcal{D}$ and $h^\star$
- Learner receives accuracy, ε, and confidence, δ, parameters
- Learner obtains training data, S, of $m(\varepsilon, \delta)$ examples
- Number of examples may depend on ε and δ
- Number of examples can *neither* depend on $\mathcal{D}$ nor on $h^\star$
- Learner finds an hypothesis h s.t.

$$\mathcal{L}_{\mathcal{D}}(h) \leq \varepsilon \;\; \text{w.p. } 1 - \delta$$

- Predictor h is:
  - **P**robably (w.p $\geq 1 - \delta$)
  - **A**pproximately (within accuracy of ε)
  - **C**orrect