Bias and VC dimension

Machine Learning Theory (MLT)

Edinburgh

Rik Sarkar

Today’s topics

- Uniform convergence and PAC learning for finite $\mathcal{H}$
- No free lunch theorem
- Error and bias-complexity tradeoff
- Infinite $\mathcal{H}$
- VC dimension and fundamental theorem of statistical learning
- Algorithms and loss functions

Uniform convergence

- S is ϵ −representative w.r.t $(\mathcal{Z}, \mathcal{H}, \mathcal{D})$ if:
    - $\forall h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$
- S gives a good estimate of the true loss for each h

- $\mathcal{H}$ has uniform convergence if there is $m_{\mathcal{H}}^{UC}: (0,1)^2 \to \mathbb{N}$
    - Such that a random sample $S \sim \mathcal{D}^m$ of size $m \geq m_{\mathcal{H}}^{UC}(\epsilon, \delta)$
    - Is ϵ −representative with probability at least $1 - \delta$

- That is, a hypothesis class is said to have uniform convergence
    - if for any ϵ, δ:
    - Any random sample S of large enough sample size m (a function of ϵ, δ)
    - Is ϵ – representative with probability of at least $1 - \delta$
        - Will give a good estimate of

- Theorem:
- Every finite $\mathcal{H}$ has uniform convergence
  - i.e. Given a suitably sized random S, $\mathbb{P}[\exists h \in \mathcal{H}: |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon] \leq \delta$

- We proved this using the Chernoff-hoeffding bound:
  - $\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} \theta_i - \mu\right| > \epsilon\right] \leq 2e^{-2m\epsilon^2}$

Representativeness and learnability

- Lemma: If S is $\frac{\epsilon}{2}$-representative then
  - $L_{\mathcal{D}}(h_S) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$

- Proof: [in book]
  - For every $h \in \mathcal{H}$
  - $$\begin{aligned} L_{\mathcal{D}}(h_S) &\leq L_S(h_S) + \frac{\epsilon}{2} \\ &\leq L_S(h) + \frac{\epsilon}{2} \\ &\leq L_{\mathcal{D}}(h) + \frac{\epsilon}{2} + \frac{\epsilon}{2} = L_{\mathcal{D}}(h) + \epsilon \end{aligned}$$

  Can you explain each step?

So, we have

- Any finite $\mathcal{H}$ has uniform convergence [theorem]
- Therefore, any $S \in \mathcal{D}^m$ of suitable size $m \geq m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2}, \delta)$
  - Is $\frac{\epsilon}{2}$-representative for $\mathcal{H}$ with probability at least $1 - \delta$
  - By definition
- Therefore,
  - $L_{\mathcal{D}}(h_S) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$ with probability at least $1 - \delta$ [Lemma]
  - $\mathcal{H}$ is PAC learnable [check yourself!]

- So, we have proved finite classes are all PAC learnable
- How useful is it?

- So, we have proved finite classes are all PAC learnable
- How useful is it?

- $\mathcal{H}$ is described by some parameters
    - E.g. coefficients of polynomials, weights on edges…
- We can always discretize by taking closely spaced discrete points
    - An the log implies that sample size will not be too bad
- In fact in a computer, numbers are always discretized
    - And thus practical hypothesis classes are technically finite
- Though searching the entire class may not be practical…

Do we need to think about hypothesis classes?

- In class, we discussed why we cannot have “all possible” hypotheses as our class.
  - Challenges in computation
  - Challenges in enumerating

- We now see a theorem that says there is no universal PAC learner
- Whatever the algorithm A,
  - There is a learning problem described where it fails
    - To achieve PAC guarantee

No Free Lunch theorem (thm 5.1 in book)

- Given $\mathcal{X}, \mathcal{Y}$
- Assume 0-1 loss. Assume $m < \frac{|\mathcal{X}|}{2}$: less than half of all possible data
- Suppose f is the "perfect" classifier (that is, the "truth")
  - We would like to know f, but we can't
- There exists $\mathcal{D}, f$ such that
  - $L_{\mathcal{D}}(f) = 0$
  - For a random $S \sim \mathcal{D}^m$
  - With probability $\geq \frac{1}{7}$, the true loss $L_{\mathcal{D}}\big(A(S)\big) \geq \frac{1}{8}$

- This violates the guarantees of PAC that should hold for any ϵ, δ
  - Unless we use more than half of all possible data
- We skip the proof

Conclusion: Prior knowledge is necessary

- $\mathcal{H}$ represents what we know or can guess about the problem
- A more restricted (smaller) class is suitable when we have a pretty good idea, or prior knowledge
- A more complex (larger) class is suitable when we have less knowledge about the problem
- Taking the set of all possible hypotheses or functions will imply no knowledge of domain
- Corollary (5.2 in book): If $\mathcal{X}$ is infinite, and $\mathcal{H}$ is set of all possible functions from $\mathcal{X}$ to $\{0,1\}$, then $\mathcal{H}$ is not PAC learnable

Impact of Prior knowledge: Bias – complexity tradeoff

- So, we have to choose a fixed $\mathcal{H}$
  - We make some assumptions about the type of solution that can work
  - This introduces a *Bias* we are looking for certain types of solutions instead of all possible solutions

- Less complex $\mathcal{H}$ means more restrictive assumptions
  - And greater bias

- More complex $\mathcal{H}$ means less restrictive assumptions
  - And less bias

The various losses and errors

- Suppose an algorithm A computes model h_S on training data S
- Training Loss: $L_S(h_S)$
- Generalisation Loss or true loss: $L_{\mathcal{D}}(h_S)$
- Generalisation Gap: $L_{\mathcal{D}}(h_S) - L_S(h_S)$

Decomposition of true loss

- True loss: $L_{\mathcal{D}}(h_S) = \epsilon_{app} + \epsilon_{est}$

- Approximation error $\epsilon_{app} = L_{\mathcal{D}}(h^*)$
  - Min true error in the hypothesis class
  - Limitation of the choice of hypothesis class
- Estimation error $\epsilon_{est} = L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h^*)$:
  - Difference between approximation error and true error
  - Error due to sampling and overfitting choosing suboptimal h_S

Bias complexity tradeoff

- Small, restricted, less complex $\mathcal{H}$
  - Small estimation error
    - Needs less data
  - Large approximation error

- Large, rich, complex $\mathcal{H}$
  - Large estimation error (due to overfitting)
    - Needs more data
  - Small approximation error

- A lot of ML is about designing good $\mathcal{H}$, balancing errors
  - Often by making use of our knowledge of the domain

Infinite hypothesis classes

- We have proved that finite $\mathcal{H}$ are PAC learnable
- What about infinite $\mathcal{H}$?

- We already proved PAC guarantee for an infinite $\mathcal{H}$
  - The threshold classifier for ripe papayas: real numbers [t, 1]
- Why were we able to prove the guarantees?

Observation

- Sample complexity $m \geq \frac{\log\left(\frac{2|\mathcal{H}|}{\delta}\right)}{2\epsilon^2}$ can be written as
  - $m \geq \frac{\log(|\mathcal{H}|)+\log\left(\frac{2}{\delta}\right)}{2\epsilon^2}$

- It has two components
  - Complexity (size) of hypothesis class
  - Confidence probability

- For finite classes: $|\mathcal{H}|$ is a measure of complexity
- For infinite classes, we can use dimensionality
  - As a measure of complexity of $\mathcal{H}$
  - Allows us to get efficient results in the 1-D case

- VC-dimension: A complexity measure for infinite classes
  - Ability of $\mathcal{H}$ to split different arrangements of points into different subsets

Shattering

- Take a point set $C \subset \mathcal{X}$
- C is shattered by $\mathcal{H}$ if
- Any classification of points in C can be achieved by $\mathcal{H}$
- That is, for each possible 0-1 labelling of points in C
  - There is an $h \in \mathcal{H}$ that selects all of the ones and none of the zeros

VC dimension

- VC dimension of $\mathcal{H}$ is
- The size of the largest set $C \subset \mathcal{X}$ that can be shattered by $\mathcal{H}$

- For VC dim to be d, we have to show:
  - There is one set of size d that is shattered by $\mathcal{H}$
  - No set of size $d + 1$ is shattered by $\mathcal{H}$

VC dim examples

- Threshold functions: Dim 1
- Intervals : Dim 2
- Axis aligned rectangles: Dim 4

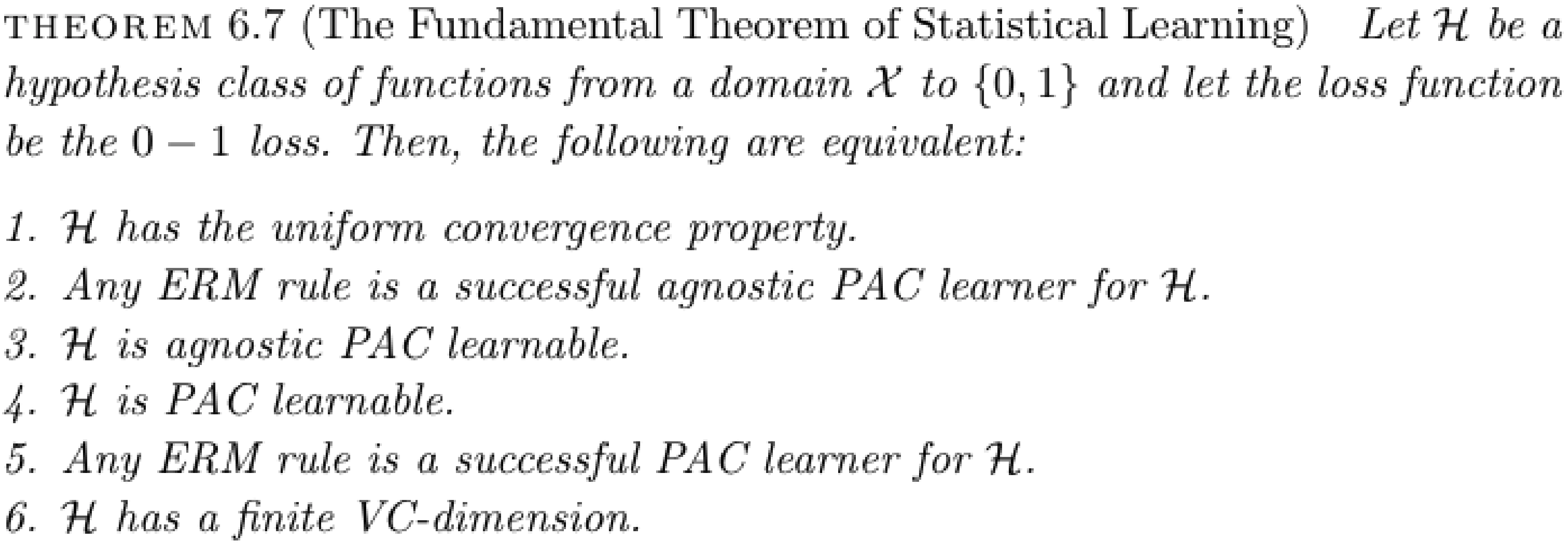
Finite classes – relation to VC dimension

- On C there are $2^{|C|}$ possibly binary classifications
- Thus, C cannot be shattered if $|\mathcal{H}| < 2^{|C|}$
- Therefore: $VCdim(\mathcal{H}) \leq \log_2|\mathcal{H}|$

Number of parametres

- Number of parameters of $\mathcal{H}$ is a good measure of complexity
- Often equals VCdim
  - But not always

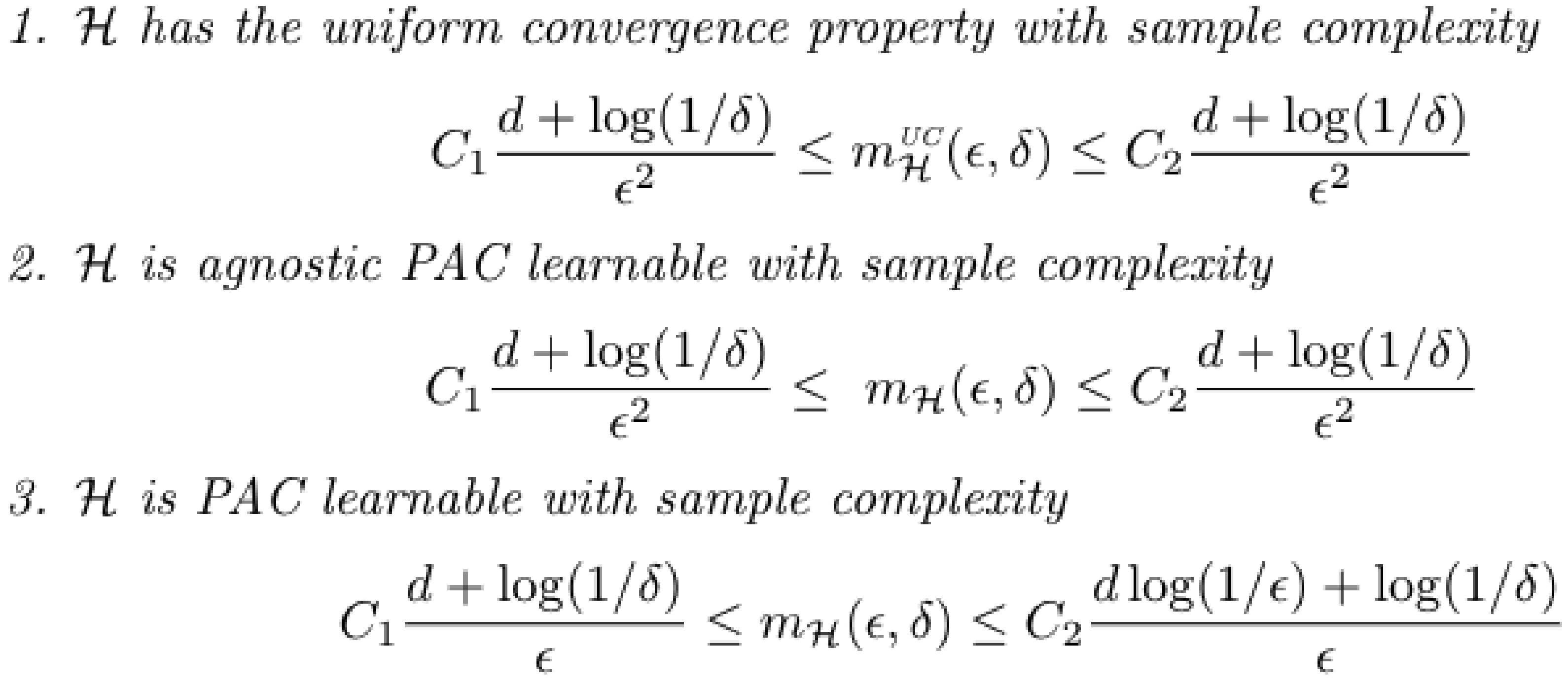
Fundamental theorem of statistical learning



THEOREM 6.7 (The Fundamental Theorem of Statistical Learning) *Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ and let the loss function be the $0 - 1$ loss. Then, the following are equivalent:*

1. *$\mathcal{H}$ has the uniform convergence property.*
2. *Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$.*
3. *$\mathcal{H}$ is agnostic PAC learnable.*
4. *$\mathcal{H}$ is PAC learnable.*
5. *Any ERM rule is a successful PAC learner for $\mathcal{H}$.*
6. *$\mathcal{H}$ has a finite VC-dimension.*

In more detail



1. *$\mathcal{H}$ has the uniform convergence property with sample complexity*

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m^{UC}_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

2. *$\mathcal{H}$ is agnostic PAC learnable with sample complexity*

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq \ m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

3. *$\mathcal{H}$ is PAC learnable with sample complexity*

$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(1/\epsilon) + \log(1/\delta)}{\epsilon}$$

- Proof: Omitted

- Other types of loss functions:
    - Similar properties hold. But do search for exact results before use.

Structural Risk Minimization

- We have till now treated $\mathcal{H}$ as binary choice: hypothesis in the class are all equally valid, while outside the class are disallowed

- What if all $h \in \mathcal{H}$ are not equally desirable?

- Structural Risk Minimization
  - Assign different preferences to different hypothesis
- Examples:
  - Assign a weight $w(\mathrm{h})$ to each hypothesis. Higher weight reflects higher preference
  - Divide hypothesis class into subclasses, assign a weight to each class

Some example weighting functions

- Polynomial degree
  - We would usually prefer lower degree polynomials
- Polynomial coefficients
  - We prefer smaller coefficients as that describes gentler/smoother functions
- Minimum description length
  - How many bits or characters does it take to represent the class?
  - Shorter description length (simpler class) is more desirable

Regularization

- A common way to achieve simpler models
- Include a “penalty” function to the loss
- E.g. sum of coefficients, or description length, sum of weights
- Minimizing the loss now includes minimizing the actual loss and minimizing the penalty
- More on regularization later

Discussion: PAC, VCDim and ML in practice

- The issue: PAC and VC analysis does not work too well on Deep learning
- VC dim of neural networks are hard to compute. A simple bound is $\mathrm{VCDim} = O(|E|)$
  - Not so useful
  - Varies somewhat by activation function etc.

Recap till now

- Hypothesis classes
- Empirical and true loss. Empiricial loss minimization
- Sample complexity
- PAC learnability (realizable, finite)
- Agnostic PAC learnability (finite)
- Bias- complexity tradeoff
- VC dim
- PAC learning infinite classes: Fundamental theorem of statistical learning