PAC Learning

Machine Learning Theory (MLT)

Edinburgh

Rik Sarkar

Recap: General ML

- **Domain set $\mathcal{X}$.**
- **Label Set $\mathcal{Y}$.** Eg. {0,1} or {-1, +1} red or blue.
- **Training data (sample set): $\boldsymbol{S} = \{(x_1, y_1), \ldots (x_m, y_m)\}$**
- **Model, hypothesis, classifier, predictor $\boldsymbol{h}$**:
  - A function $h: \mathcal{X} \rightarrow \mathcal{Y}$. That is, $h(x)$ returns a predicted label y
- **Hypothesis class $\boldsymbol{\mathcal{H}}$**: The set of functions from which h is chosen
- **Algortihm A:** Chooses hypothesis h based on S
- **Data generating distribution $\mathcal{D}$**
- **Success measure: Loss/error function *L***

Empirical risk minimization

- Empirical risk: Average loss in experiment
- For now, define empirical loss or risk of any hypothesis $h \in \boldsymbol{\mathcal{H}}$ as:

$$L_S(h) \stackrel{\text{def}}{=} \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

- ERM algorithm (A):
  - Find the h with min loss: $\arg\min_{h \in \boldsymbol{\mathcal{H}}} L_S(h)$
  - We can write $h_S = A(S)$ to mean that h was computed by A based on S
  - For a finite $\mathcal{H}$, A can just test all hypothesis and pick the one with the smallest loss.

Overview

- Machine learning has two questions:
  - Sample and prepare data
    - Question: How much data do we need?

  - Apply an algorithm to find a good model in class $\mathcal{H}$
    - Question: What is an algorithm that finds good models for a particular class?
      - What loss function to use
      - What steps the algorithm should take
      - How to modify the algorithm to get desirable properties like privacy, fairness etc
- In the course
  - We will do the data sampling first. (this week and next)
  - Algorithms and their properties in succeeding weeks
  - General approach start with simple cases to build intuition and analysis. Then discuss complex cases

Today’s questions

- How much data do we need for good guarantees?
- What kind of problems are “learnable”?
  - Observe that just because we would like to find a good model does not mean that it is possible!
- Approach: we will start with simple problems and finite hypothesis classes to build intuition and go toward more complex ones
- We will use formal mathematical notations and proofs
  - The ideas are not that hard, but takes getting used to the notations
  - Ask if you have questions
  - This lecture is harder than others. You will need to do some study afterwards!
- It gives us practice at how to think precisely and clearly. This will be useful in later parts of the course
  - You do not need to recreate these proofs in exam. Just make use that you follow the ideas
- Also read from the book

A simple classifier (exercise)

- A supermarket has asked us to build a model to classify ripe papayas
- Green is unripe, yellow is ripe
- A sensor reads the colour
- And returns a value in $[0,1]$
- Assume the supermarket sends us a random sample of labelled readings
- There is a color threshold t^* of ripe papayas but we don't know it.

Sample size problem

- Show that sample size $\mathrm{m} \geq \frac{1}{\epsilon} \ln \frac{2}{\delta}$ suffices to get ϵ, δ accuracy:
  - With probability at least $1 - \delta$
  - At most ϵ fraction of unseen papayas will be misclassified

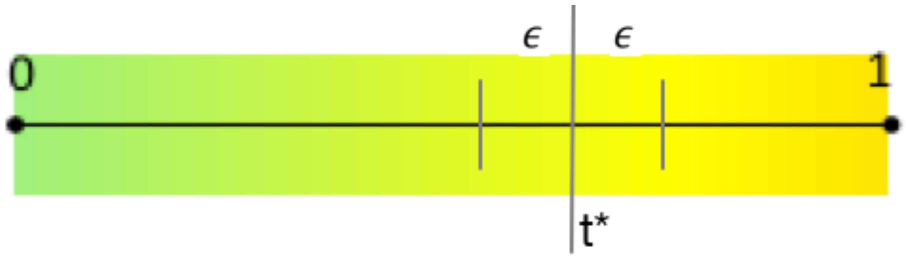
Sample size problem

- Show that sample size $m \geq \frac{1}{\epsilon} \ln \frac{2}{\delta}$ suffices to get ϵ, δ accuracy.
  - With probability at least $1 - \delta$
  - At most ϵ fraction of unseen papayas will be misclassified

- Assume that papayas are uniformly distributed in [0,1] (the result works without this, but we are doing the easier version in class)

Algorithm

- Draw enough samples
  - So that there are samples in ϵ intervals to the left and right of t*



- Take the highest “unripe” label and lowest “ripe” label.
- Select any point between these two

Sketch of proof

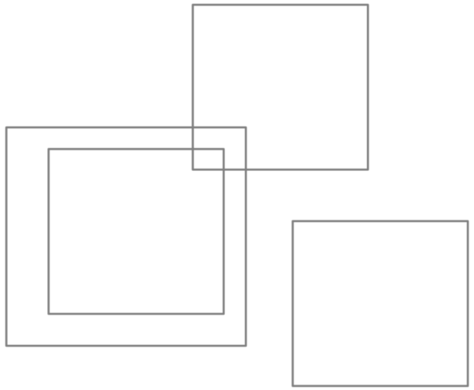
- Of sample size
- Consider only one interval r of size ϵ
- And a sample size of $\mathrm{m} \geq \frac{1}{\epsilon} \ln \frac{2}{\delta}$
- Show that there is a sample in r, with probability at least $1 - \frac{\delta}{2}$
- Hints:
  - Use the probability that none of the m samples are in r
  - Use the inequality that $(1-p)^{\frac{1}{p}} \leq \frac{1}{e}$

Finite hypothesis classes

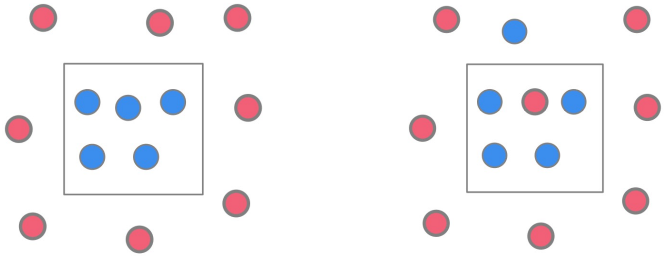
- To start with, we assume the number of possible hypotheses is finite.
- Suppose the sensor values are in range [0,100] and we can choose thresholds at only integer positions. What is $|\mathcal{H}|$?
- Suppose sensor values are in range [0, 1] and we are choosing from pre-fixed thresholds at intervals of ϵ. How many thresholds are there?

Simplifying assumptions for basic analysis

- Assumption 1: Finite $\boldsymbol{\mathcal{H}}$
  - Limit the hypothesis class to have a finite number of hypotheses
  - What



- Assumption 2: Realizability:
  - There is $h^* \in \boldsymbol{\mathcal{H}}$ that achieves perfect separation between classes
  - i.e. zero loss: $L_{(\boldsymbol{\mathcal{D}}, f)}(h^*) = 0$
  - It implies that the in-sample loss $L_S(h^*) = 0$



Sampling assumption (i.i.d)

- Assumption:
  - Examples in training set are independent and identically distributed according to $\boldsymbol{\mathcal{D}}$
  - Written as $S \sim \boldsymbol{\mathcal{D}}^m$

- Algorithm A:
  - Check all $h \in \mathcal{H}$
  - Pick $h_S = \underset{h \in \mathcal{H}}{\operatorname{argmin}} L_S(h)$
- Note that h_S is best (zero loss) in training data, but may not be good in true loss on $\mathcal{D}$

Sampling bound

- With these assumptions, we can show that

$$m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$$

- Samples suffice for ϵ, δ guarantee: $\mathbb{P}[L_{\mathcal{D},f}(h_S) \leq \epsilon]$
  - The best hypothesis on training data has small true loss
  - With probability $1 - \delta$,

Proof

- The algorithm expects and finds 0 empirical loss in the training set
  - Outputs an h with 0 empirical loss (there can be many of these)
  - These “Look good” in data
- A “really good” hypothesis also has 0 true loss in $\mathcal{D}$ (realizability)
- Certain hypothesis are “bad”: have a true loss $L_{\mathcal{D},f}(h) > \epsilon$

Proof

- The algorithm expects and finds 0 empirical loss in the training set
  - Outputs an h with 0 empirical loss (there can be many of these)
  - These “Look good” in data
- A “really good” hypothesis also has 0 true loss in $\mathcal{D}$ (realizability)
- Certain hypothesis are “bad”: have a true loss $L_{\boldsymbol{\mathcal{D}},f}(h) > \epsilon$
- We get a bad output only if a bad hypothesis has zero empirical loss in the sample. Let’s compute the probability
- For a bad hypothesis h, the probability of getting one training label right is:
  - $1 - L_{\boldsymbol{\mathcal{D}},f}(h) \leq 1 - \epsilon$
- The probability of h getting m labels right is $\leq (1-\epsilon)^m \leq e^{-\epsilon m}$
  - This is the probability that a bad hypothesis h *looks good*

- If H_B is the subset of bad hypotheses
- Then by union bound, probability of some bad hypothesis looking good is
  - $\leq |H_B|e^{-\epsilon m} \leq |\boldsymbol{\mathcal{H}}|e^{-\epsilon m}$
- Substitute m to get probability of a bad h succeeding $\leq \delta$
- The probability of not getting a bad result is $\geq 1 - \delta$

QED

Observe

- The proof says that if h^* is the best hypothesis in a finite $\mathcal{H}$,
  - It is always possible to get as close to h^* in accuracy as we want
  - Just need large enough m

- That is, with some assumptions a good enough h_S can always be "learned" from big enough dataset

PAC Learnability

- We have just seen that every finite class is "PAC learnable"

- If $\mathcal{H}$ is finite and realizable, then there is an algorithm that can
  - get as close to the optimum* model as we want,
  - with as high a probability as we want
  - Provided we give it enough data
    - (and happily, that data is not too much!)

- *optimum model or hypothesis *within* $\mathcal{H}$
  - How good that is in absolute accuracy depends on how good an $\mathcal{H}$ we select

PAC learnability (formal definition)

- A hypothesis class $\boldsymbol{\mathcal{H}}$ is PAC learnable if
  - There exists a function $m_{\boldsymbol{\mathcal{H}}}(0,1)^2 \rightarrow \mathbb{N}$ (means: depending on ϵ, δ, there is a suitable number of samples)
  - And an algorithm that:
    - For every ϵ, δ
    - For $\boldsymbol{\mathcal{D}}$ over $\boldsymbol{\mathcal{X}}$
    - With realizability assumption
    - On $m \geq m_{\boldsymbol{\mathcal{H}}}(\epsilon, \delta)$ i.i.d samples from $\boldsymbol{\mathcal{D}}, \boldsymbol{f}$
    - Finds an h that satisfies
      - $L_{(\boldsymbol{\mathcal{D}}, f)}(h) \leq \epsilon$ (finds a good h)
      - with probability at least $1 - \delta$

More general learning

- In general, realizability is not true
  - There may be no perfect $h = f$
- Called Agnostic PAC learning

- E.g. Our $\mathcal{H}$ consists of squares
  - But the data needs a circle to separate classes

- To extend to more general scenarios, let's change our assumptions

More general model – agnostic learning

- Modified data generating distribution:
  - Define $\mathcal{D}$ to be probability distribution over $\mathcal{X}\times\mathcal{Y}$
  - Consequence: The same $x \in \mathcal{X}$ may have labels 0 or 1 probabilistically
- Redefine true risk:

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \stackrel{\text{def}}{=} \mathcal{D}(\{(x,y) : h(x) \neq y\}).$$

- (homework: compare this with how we defined true risk earlier)

- Question: Where can this happen in a real example?

Agnostic PAC learnability

- A hypothesis class $\boldsymbol{\mathcal{H}}$ is Agnostic PAC learnable if
  - There exists a function $m_{\boldsymbol{\mathcal{H}}}(0,1)^2 \to \mathbb{N}$
  - And an algorithm that:
    - For every ϵ, δ
    - For $\boldsymbol{\mathcal{D}}$ over $\mathcal{X} \times \mathcal{Y}$
    - ~~With realizability assumption~~
    - On $m \geq m_{\boldsymbol{\mathcal{H}}}(\epsilon, \delta)$ i.i.d samples from $\boldsymbol{\mathcal{D}}, \boldsymbol{f}$
    - Finds an h that satisfies
      - $\boldsymbol{L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon}$ (gets ϵ close to the best $h' \in \mathcal{H}$)
      - with probability at least $1 - \delta$

Other types of learning problems (defined by suitable loss)

- We have looked at binary classification
- Other possibilities:
- Multi-class classification
  - E.g, Measure loss as the probability of predicting a wrong label
- Regression: labels are real numbers i.e. $\mathcal{Y} = \mathbb{R}$

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} (h(x) - y)^2$$

Generalised loss

- Instead of $\mathcal{X}\times\mathcal{Y},$ we consider a single domain $\mathcal{Z}$ (which may be $\mathcal{X}\times\mathcal{Y}$, or something else)
  - Loss functions are: $\ell: \mathcal{H}\times\mathcal{Z} \rightarrow \mathbb{R}_+$
    - The loss measured for a single element: $\ell(h, z)$
- Generalises to more ML problems e.g. clustering (unsupervised learning)
- True risk function: Expected loss: $L_{\mathcal{D}}(h) = \ \mathbb{E}_{z\in\mathcal{D}}[\ell(h, z)]$
- Empirical risk function: $L_{S(h)} = \frac{1}{m}\sum_{i=1}^{m} \ell(h, z_i)$

- Exercise: Define k-means clustering as a formal ML problem, with hypothesis class, loss function etc.

Agnostic PAC learning with general loss function

- Defined in terms of $\mathcal{Z}$ and general loss functions
- Learning in absence of realizability

Representative data sets

- We use S as a representative of $\mathcal{D}$

- In general, we cannot be sure that
  - we will find an h that does well outside training data,
  - or that for an h, the performance on S matches general performance

- When it does, we say S is a representative sample

Representative sample

- S is ϵ −representative w.r.t $(\mathcal{Z}, \mathcal{H}, \mathcal{D})$ if:
  - $\forall h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$

Representative sample

- 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

- Observe:
  - A sample is representative with respect to $\mathcal{H}, \mathcal{Z}$
  - That is, it is representative with respect to a specifc problem and hypothesis class
- Question: Can there be a notion of represenativeness independent of $\mathcal{H}, \mathcal{Z}$?

Representative sample

- 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

- Lemma:
  - If S is $\frac{\epsilon}{2}$ −representative, and $h_S \in \mathrm{argmin}_{h \in \mathcal{H}} L_S(h)$, then
  - $L_{\mathcal{D}}(h_S) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$

- With representative data, the best empirical (trained) model (h_S) is almost as good as the best model for true data

Uniform convergence

- $\mathcal{H}$ 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 $1 - \delta$

- When $\mathcal{H}$ has uniform convergence, it means we know a large enough m that gives accurate estimates for all h

Corollary

- If $\mathcal{H}$ has uniform convergence with $m_{\mathcal{H}}^{UC}$,
  - Then $\mathcal{H}$ is PAC learnable with $m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2}, \delta)$

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

- And therefore every finite $\mathcal{H}$ is agnostic PAC-learnable

- Proof next week, using Chernoff-hoeffding bound

Chernoff-Hoeffding bound

- Very important result in theoretical CS and ML
- Suppose θ_i are random variables with average $\frac{1}{m}\sum_{i=1}^{m}\theta_i$
- Suppose μ is the expected value of a random θ
- Law of large numbers: with increasing m, $\frac{1}{m}\sum_{i=1}^{m}\theta_i$ approaches μ
  - Ie, $\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right|$ becomes smaller
- But how fast? What m do we need to get ϵ-close to μ?
- Chernoff-Hoeffding bound:
- $\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i - \mu\right| > \epsilon\right] \leq 2e^{-2m\epsilon^2}$

- Proof of uniform convergence for finite $\mathcal{H}$: next week.
- (you can look up in the book!)

- So, we have proved finite classes are all PAC learnable

- Next week, we will cover
  - The proof of uniform convergence
  - No free lunch theorem: There is no universal learner
  - Bias-complexity tradeoff
  - Infinite hypothesis classes and fundamental theorem of statistical learning
  - Starting with ML algorithms