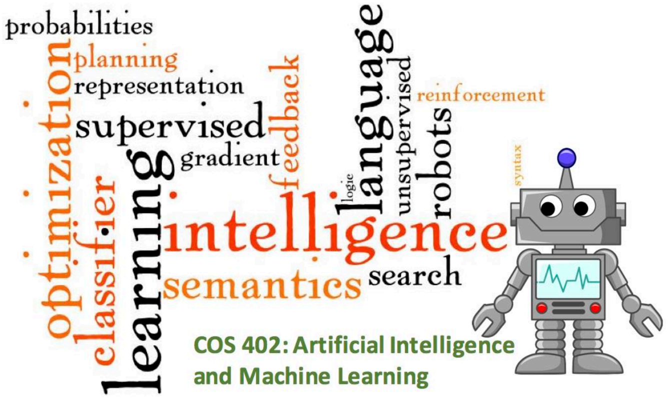


COS 402 – Machine Learning and Artificial Intelligence
Fall 2016

Lecture 3: Learning Theory

Sanjeev Arora   Elad Hazan

PRINCETON UNIVERSITY

Admin

- Exercise 1 – due next Tue, in class
- Enrolment…

Recap

We have seen:

- AI by introspection (Naïve methods)
- Classification by decision trees
- Informally: overfitting, generalization

Agenda

- Statistical & computational learning theories for learning from examples
  - Setting
  - Generalization
  - The ERM algorithm
  - Sample complexity
- Fundamental theorem of statistical learning for finite hypothesis classes
- The role of optimization in learning from examples

Wanted: theory for learning from examples

- Captures: spam detection, chair classification,…
- Generic (doesn’t commit to a particular method)
- Answers our questions from last lecture
  - Overfitting
  - Generalization
  - Sample complexity

Setting

Input:

- X = Domain of examples (emails, pictures, documents, …)

Output:

- Y = label space (for this talk, binary Y={0,1})

Data access model:

- Data access model: learner can obtain i.i.d samples from D = distribution over (X,Y) (the world)

Goal:

- produce hypothesis h: $X \mapsto Y$ with low *generalization error*

Learning from examples

A learning problem: $L = (X, Y, c, \ell)$

- X = Domain of examples (emails, pictures, documents, …)
- Y = label space (for this talk, binary Y={0,1})
- D = distribution over (X,Y) (the world)
- Data access model: learner can obtain i.i.d samples from D
- Concept = mapping $c: X \mapsto Y$
- Loss function $\ell: (Y, Y) \mapsto R$, such as $\ell(y_1, y_2) = 1_{y_1 \neq y_2}$
- Goal: produce hypothesis h: $X \mapsto Y$ with low *generalization error*

$err(h) = E_{(x,y)\sim D}\,[\ell(h(x), c(x))]$ *, 4 today:* $err(h) = Pr_{(x,y)\sim D}\,[h(x) \neq c(x)]$

No free lunch

Learning algorithm:

1. observe m samples $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\} \sim D$ from the world distribution.
2. Produce hypothesis $h: X \mapsto Y = \{0,1\}$

Extreme overfitting:

$$h(x) = \begin{cases} y_i & if \quad x = x_i \\ 0 & otherwise \end{cases}$$

No-free lunch theorem: Consider any domain X of size |X | = 2m, and any algorithm A which outputs a hypothesis h ∈ H given m sample S. Then there exists a concept c : X → {0, 1} and a distribution D such that:

- err(c) = 0
- With probability at least $\frac{1}{10}$, err(A(S)) ≥ $\frac{1}{10}$.

Definition: learning from examples w.r.t. hypothesis class

A learning problem: $L = (X, Y, c, \ell, H)$

- X = Domain of examples (emails, pictures, documents, …)
- Y = label space (for this talk, binary Y={0,1})
- D = distribution over (X,Y) (the world)
- Data access model: learner can obtain i.i.d samples from D
- Concept = mapping $c: X \mapsto Y$
- Loss function $\ell: (Y, Y) \mapsto R$, such as $\ell(y_1, y_2) = 1_{y_1 \neq y_2}$
- H = class of hypothesis: $H \subseteq \{X \mapsto Y\}$
- Goal: produce hypothesis h∈ H with low *generalization error*

$$err(h) = E_{(x,y)\sim D}\left[\ell(h(x), c(x))\right]$$

Realizability

A learning problem: $L = (X, Y, c, \ell, H)$ is realizable if there exists a hypothesis that has zero generalization error in H.

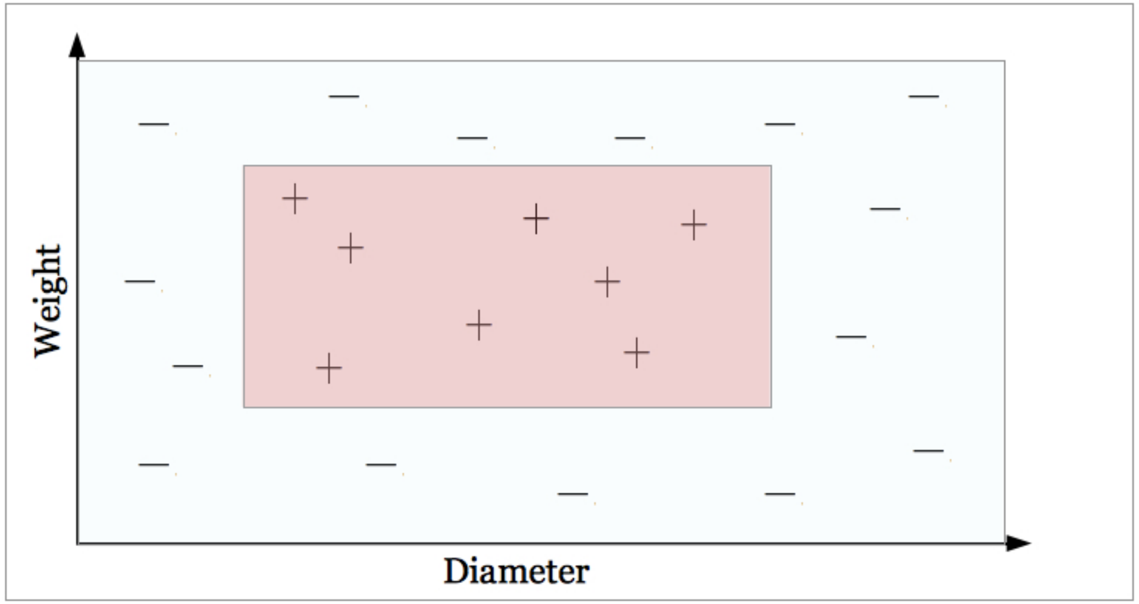
$$\exists h \in H \;\; s.t. \; err(h) = E_{(x,y)\sim D}\left[\ell\big(h(x), c(x)\big)\right] = 0$$

If not satisfied, agnostic (competitive) learning: try to minimize generalization error compared to best-in-class, i.e. minimize

$$err(h) - \min_{h^* \in H} err(h^*)$$

Examples

- Apple factory:
  - Apples are sweet (box) or sour (for export)
  - Features of apples: weight and diameter
  - Weight, diameter are distributed uniformly at random in a certain range
  - The concept:



  - X,Y,c = ?
  - Reasonable loss function?
  - Reasonable hypothesis class?
  - Realizable?

Examples

- MPG example from last lecture
- X,Y,c,H = ?

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

Examples

- Spam detection:
  - X,Y,c = ?
  - Reasonable loss function?
  - Reasonable hypothesis class?
  - Realizable?
- Chair classification

PAC learnability

Learning algorithm:

1. observe m samples $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\} \sim D$ from the world distribution.
2. Produce hypothesis $h \in H$

Learning problem $L = (X, Y, H, c, \ell)$ is PAC-learnable if there exists a learning algorithm s.t. for every $\delta, \epsilon > 0$, there exists $\mathrm{m} = f(\epsilon, \delta, H) < \infty$, s.t. after observing S examples, for $|S| = m$, returns a hypothesis $h \in H$, such that with probability at least

$$1 - \delta$$

it holds that

$$err(h) \leq \epsilon$$

$f(\epsilon, \delta, H)$ is called the **sample complexity**

agnostic PAC learnability

Learning problem $L = (X, Y, H, c, \ell)$ is agnostically PAC-learnable if there exists a learning algorithm s.t. for every $\delta, \epsilon > 0$, there exists $\mathrm{m} = f(\epsilon, \delta, H) < \infty$, s.t. after observing S examples, for $|S| = m$, returns a hypothesis $h \in H$, such that with probability at least

$$1 - \delta$$

it holds that

$$err(h) \leq \min_{h^* \in H} err(h^*) + \epsilon$$

How good is this definition?

- Generality:
  - Any distribution, domain, label set, loss function
  - Distribution is unknown to the learner
  - No algorithmic component
  - Realizable?
- Uncovered
  - Adversarial / changing environment
  - Algorithms…

What can be PAC learned and how?

Natural classes?

Algorithms for PAC learning?

ERM algorithm: (Empirical Risk Minimization)

- Sample $S = f(\epsilon, \delta, H)$ labelled examples from D
- Find and return the
- hypothesis that minimizes the loss on these examples:

$$h_{ERM} = \arg\min_{h \in H}\{err_S(h)\} \quad \text{where} \quad \mathrm{err}_S = \frac{1}{\mathrm{m}} \sum_{i=1\ to\ m} \ell(h(x_i), y_i)$$

Computational efficiency?

decision trees?

Fundamental theorem for finite H

Theorem:

Every realizable learning problem $L = (X, Y, H, c, \ell)$ for finite H, is PAC-learnable with sample complexity $S = O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right)$ using the ERM algorithm.

- VERY powerful, examples coming up…
- Explicitly algorithmic
- Captures overfitting, generalization...
- Infinite hypothesis classes?
- But first proof…

Fundamental theorem - proof

Proof:

1. Let $S = \{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\} \sim D$ be the sample from the world, size : $m = \frac{\log H + \log\frac{1}{\delta}}{\epsilon}$
2. ERM does NOT learn L only if $err(h_{ERM}) > \epsilon$ with probability larger than δ.
3. We know that $err_S(h_{ERM}) = 0$ (why?)

$$\Pr[err(h_{ERM}) > \epsilon] \quad \leq \quad \Pr[\,\exists h\ s.t.\ err(h) > \epsilon\ \wedge err_S(h) = 0]$$

$$\leq \quad \sum_{h\in H} \Pr[err(h) > \epsilon\ \wedge err_S(h) = 0] \ \leq \sum_{h\in H} \Pr[\forall\, i \in S\ \ h(x_i) = y_i | err(h) > \epsilon]$$

$$= \sum_{h\in H} \prod_{\mathrm{i=1 to\, m}} \quad \Pr[h(x_i) = y_i | err(h) > \epsilon]$$

$$\leq\ \textstyle\sum_{h\in H} \prod_{i=1\ to\ m}(1-\epsilon) = \sum_h (1-\epsilon)^m$$

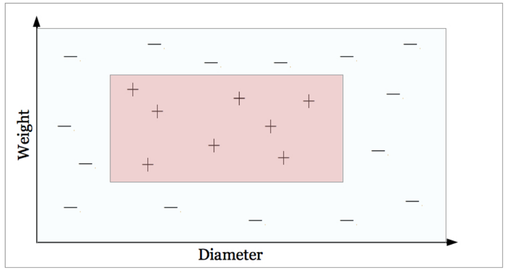
$$= |H|(1-\epsilon)^m \ \leq\ |H|\ e^{-\epsilon m} = |H|\ e^{-\epsilon\frac{\log H + \log\frac{1}{\delta}}{\epsilon}} = \delta$$

Examples – statistical learning theorem

Theorem:

Every realizable learning problem $L = (X, Y, H, c, \ell)$ for finite H, is PAC-learnable with sample complexity $S = O\left(\frac{\log H + \log\frac{1}{\delta}}{\epsilon}\right)$ using the ERM algorithm.

- Apple factory: Wt. is measured in grams, 100-400 scale.
  Diameter: centimeters, 3-20



- Spam classification using decision trees of size 20 nodes
  - 200K words

We stopped here…

- To be continued next class + start optimization!