昆山杜克大学
DUKE KUNSHAN UNIVERSITY

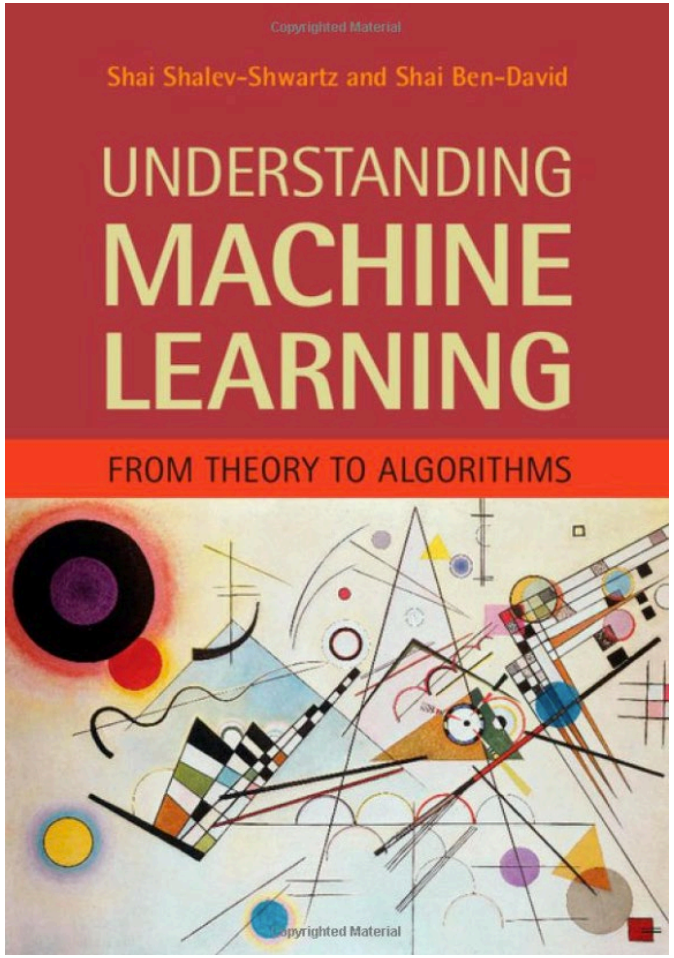
PAC Learning

STATS 303 Statistical Machine Learning

Spring 2022

Lecture 17

textbook for the remaining part of the course



Douban score

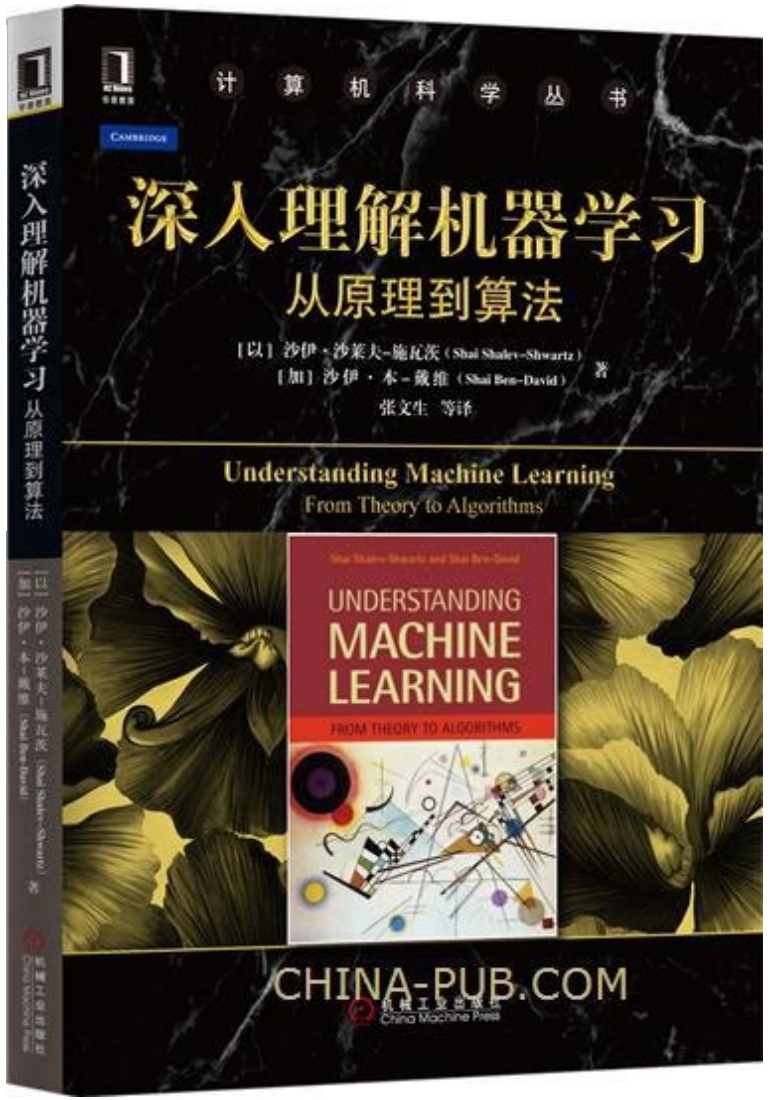
6.0 ★★★☆☆ 27 ratings

5 stars	3.7%
4 stars	40.7%
3 stars	33.3%
2 stars	11.1%
1 star	11.1%

Douban score

8.4 ★★★★☆ 60 ratings

5 stars	61.7%
4 stars	28.3%
3 stars	8.3%
2 stars	1.7%
1 star	0.0%



setting: basics

- Domain set: $\mathcal{X}$ — the set of objects we wish to label
- Label set: $\mathcal{Y}$ — the labels
  - For this moment, we restrict ourselves to binary classification, so $\mathcal{Y} = \{0,1\}$.

- Learner's input: $S = ((x_1, y_1), \cdots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ — the sample of training data

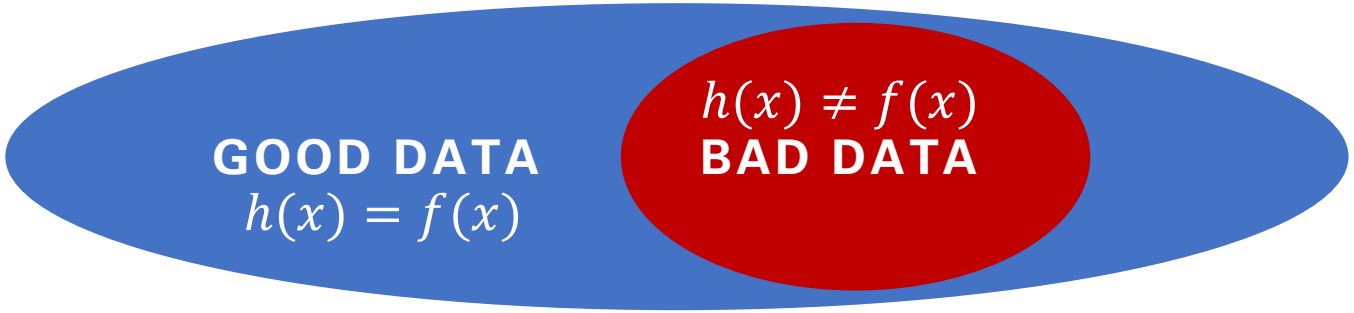
- Learner's output: $h: \mathcal{X} \to \mathcal{Y}$ — a rule for prediction
  - Usually, we require this h is chosen from a hypothesis class $\mathcal{H}$.

setting: error function

- Let f be the correct classifier. Then we want $h \approx f$.
- Given data distribution $\mathcal{D}$, we can define the **error** of h w.r.t. f to be

$$L_{\mathcal{D},f}(h) = P_{x\sim\mathcal{D}}[h(x) \neq f(x)] =: \mathcal{D}(\{x \in \mathcal{X}: h(x) \neq f(x)\})$$

- Remark: the distribution $\mathcal{D}$ is the "true" data distribution, not the distribution of the training data we sample.



setting: training error

- On the other hand, given the training data $S = ((x_1, y_1), \cdots, (x_m, y_m))$, the **training error** (empirical error / empirical risk) of h is defined to be

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

  where $[m] = \{1, \cdots, m\}$.

- By training we can achieve the empirical risk minimizer (ERM)

$$h_S \in \operatorname{argmin}_{h \in \mathcal{H}} L_S(h)$$

- Want: (generalization) h_S is correct for examples we don't see in S.

hope: perfect h

- Assume:
  1. Each x_i is sampled i.i.d. according to $\mathcal{D}$
  2. There exists $h^* \in \mathcal{H}$ s.t. $L_{\mathcal{D},f}(h^*) = 0$ (realizability assumption).

- Remark: the realizability assumption automatically implies that $L_S(h^*) = 0$. Moreover, since $h_S \in \operatorname{argmin}_{h \in \mathcal{H}} L_S(h)$, we also have $L_S(h_S) = 0$.

- Hope: find a perfect algorithm for finding h so that $L_{\mathcal{D},f}(h) = 0$.

no way

- Claim: There does not exist an (algorithm for finding) h for which $L_{\mathcal{D},f}(h) = 0$.

no way

- Claim: There does not exist an (algorithm for finding) h for which $L_{\mathcal{D},f}(h) = 0$.
- Why?
  - For every $\epsilon \in (0,1)$, consider $\mathcal{X} = \{x_1, x_2\}$ where $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$.
  - The training set contains m i.i.d. examples.
  - It is possible that we don't see x_2 at all in the training set. Then any algorithm would not be able to determine the value at x_2.

approximately correct?

- We only hope for $L_{(\mathcal{D},f)}(h) \leq \epsilon$ for a given ϵ (specified by the user). This ϵ is called the accuracy parameter.

approximately correct?

- We only hope for $L_{(\mathcal{D},f)}(h) \leq \epsilon$ for a given ϵ (specified by the user). This ϵ is called the accuracy parameter.

- But even this is **not possible** if we keep sampling a non-representative data point.
- For instance, consider $\mathcal{X} = \{x_1, x_2\}$ where $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$. There is a probability of ϵ^m that we keep sampling x_2. In this case, the error could be very large, since x_1 is the representative data point.

approximately correct?

- We only hope for $L_{(\mathcal{D},f)}(h) \leq \epsilon$ for a given ϵ (specified by the user). This ϵ is called the accuracy parameter.

- But even this is **not possible** if we keep sampling a non-representative data point.
- For instance, consider $\mathcal{X} = \{x_1, x_2\}$ where $\mathcal{D}(\{x_1\}) = 1 - \epsilon$, $\mathcal{D}(\{x_2\}) = \epsilon$. There is a probability of ϵ^m that we keep sampling x_2. In this case, the error could be very large, since x_1 is the representative data point.

This is not very probable.

probably approximately correct

- We allow our algorithm to "fail" (produce an error larger than ϵ) with probability δ.
- That is, we are $(1 - \delta)$ "confident" that our algorithm will "succeed" (produce an error less than or equal to ϵ).
- This δ is called a confidence parameter.

Can we be probably, approximately correct?

Can we be probably, approximately correct?

- Let $S|_x = (x_1, \cdots, x_m)$ be the instances of the training set.
- We would like to upper bound the probability that h_S has large error (which means our training examples are "unlucky" choices). That is, we'd like to find an upper bound of

$$\mathcal{D}^m\left(\left\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\right\}\right)$$

- Let $\mathcal{H}_B$ be the set of "bad" hypotheses

$$\mathcal{H}_B = \{h \in \mathcal{H} : L_{(\mathcal{D},f)}(h) > \epsilon\}$$

- Let M be the set of misleading examples

$$M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

Can we be probably, approximately correct?

- We have defined

$$M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

- Also, by realizability, $L_S(h_S) = 0$.

- Therefore,

unlucky choice of examples

$$\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subset M$$

because if $S|_x \in$ LHS, then $h_S \in \mathcal{H}_B$.

Can we be probably, approximately correct?

- Since $\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subset M$, we have

$$\mathcal{D}^m\left(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}\right)$$

$$\leq \mathcal{D}^m(M) = \mathcal{D}^m\{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

$$= \mathcal{D}^m\left(\bigcup_{h\in\mathcal{H}_B} \{S|_x : L_S(h) = 0\}\right)$$

$$\leq \sum_{h\in\mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \qquad \text{(union bound)}$$

Can we be probably, approximately correct?

- Since $\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subset M$, we have

$$\mathcal{D}^m\left(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}\right)$$
$$\leq \mathcal{D}^m(M) = \mathcal{D}^m\{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

$$= \mathcal{D}^m\left(\bigcup_{h\in\mathcal{H}_B} \{S|_x : L_S(h) = 0\}\right)$$

$$\leq \sum_{h\in\mathcal{H}_B} \underbrace{\mathcal{D}^m(\{S|_x : L_S(h) = 0\})}$$

$$\prod_{i=1}^{m} \mathcal{D}(\{x_i : h(x_i) = f(x_i)) = \left(1 - L_{(\mathcal{D},f)}(h)\right)^m \leq (1-\epsilon)^m \leq e^{-\epsilon m}$$

$\mathcal{D}(\{x : h(x) \neq f(x)\})$

Note that
$1 - \varepsilon \leq e^{-\varepsilon}$
since by Taylar's
Theorem,
$e^{-\varepsilon} = 1 - \varepsilon + \frac{\eta^2}{2}$

Can we be probably, approximately correct?

- Since $\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subset M$, we have

$$\mathcal{D}^m\left(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}\right)$$
$$\leq \mathcal{D}^m(M) = \mathcal{D}^m\{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

$$= \mathcal{D}^m\left(\bigcup_{h\in\mathcal{H}_B}\{S|_x : L_S(h) = 0\}\right)$$

$$\leq \sum_{h\in\mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\})$$

$$\leq |\mathcal{H}_B| e^{-\epsilon m} \leq |\mathcal{H}| e^{-\epsilon m}$$

→ If we want "not approximately correct" has a probability $\leq \delta$. then we set $|\mathcal{H}| e^{-\epsilon m} \leq \delta$. That is

$$-\epsilon m \leq \log(\delta/|\mathcal{H}|).$$

That is $m \geq \frac{1}{\epsilon}\log(|\mathcal{H}|/\delta)$.

Can we be probably, approximately correct?

- Since $\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subset M$, we have

$$\mathcal{D}^m\left(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}\right)$$
$$\leq \mathcal{D}^m(M) = \mathcal{D}^m\{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

$$= \mathcal{D}^m\left(\bigcup_{h\in\mathcal{H}_B}\{S|_x : L_S(h) = 0\}\right)$$

$$\leq \sum_{h\in\mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\})$$
$$\leq |\mathcal{H}_B|e^{-\epsilon m} \leq |\mathcal{H}|e^{-\epsilon m}$$

As long as $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$,we will have

$$\mathcal{D}^m\left(\{S|_x : L_{(\mathcal{D},f)}(h_S) \geq \epsilon\}\right) \leq \delta$$

$$\Updownarrow$$

$$\mathcal{D}^m\left(\{S|_x : L_{(\mathcal{D},f)}(h_S) \leq \epsilon\}\right) \geq 1-\delta$$

When the hypothesis class is finite, as long as we take a large number of training data, our model will probably be approximately correct!

PAC learning

Probably Approximately Correct (PAC) learning

A hypothesis class $\mathcal{H}$ is said to be **PAC learnable** if there exists a function $m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labeling function $f: \mathcal{X} \to \{0,1\}$, if the realizability assumption holds w.r.t. $\mathcal{H}, \mathcal{D}, f$, then
- when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labeled by f, the algorithm returns a hypothesis h such that,
- with probability of at least $1 - \delta$ (over the choice of the examples), $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

sample complexity

- $m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$ is called the sample complexity
- In previous section, we have shown:
  - Every finite hypothesis class is PAC learnable with sample complexity

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil$$

  - We will soon see in this course that “finite”-ness is not essential here.

agnostic PAC learning

- In practice, the realizability assumption is too restrictive. We want to release this in our definition.

- Now consider the data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$: pairs of a data point and a label (instead of over the data point only)

$$L_{\mathcal{D}}(h) = P_{(x,y)\sim\mathcal{D}}[h(x) = y] =: \mathcal{D}(\{(x,y): h(x) \neq y\})$$

(compare with what we had before:

$$L_{\mathcal{D},f}(h) = P_{x\sim\mathcal{D}}[h(x) \neq f(x)] =: \mathcal{D}(\{x \in \mathcal{X}: h(x) \neq f(x)\})$$

)

agnostic PAC learning

A hypothesis class $\mathcal{H}$ is said to be **agnostic PAC learnable** if there exists a function $m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, ~~and for every labeling function, if the realizability assumption holds,~~
- when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ ~~and labeled by the labeling function,~~ the algorithm returns a hypotheses h such that,
- with probability of at least $1 - \delta$ (over the choice of the examples), ~~$L_{(\mathcal{D},f)}(h) \leq \epsilon$~~

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$

agnostic PAC learning

A hypothesis class $\mathcal{H}$ is said to be **agnostic PAC learnable** if there exists a function $m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$,
- when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns a hypotheses h such that,
- with probability of at least $1 - \delta$ (over the choice of the examples),

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$

example: Bayes optimal predictor

- Given any $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, the best label predicting function will be (the naïve Bayes' classifier in Lecture 1)

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & \text{if } \mathbb{P}[y = 1|x] \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

- We can prove (HW exercise) that this classifier is the best possible in the sense that $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$ for any other classifier g. That is,

$$f_{\mathcal{D}} = \underset{h' \in \mathcal{H}}{\operatorname{argmin}} L_{\mathcal{D}}(h')$$

- However, since we don't know $\mathcal{D}$, we don't know $f_{\mathcal{D}}$. Therefore, we can only hope to be "approximately optimal":

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon = L_{\mathcal{D}}(f_{\mathcal{D}}) + \epsilon$$

beyond binary classification

- We can generalize the definition further by considering a general loss function: $\ell: \mathcal{H} \times Z \to \mathbb{R}_+$.
- This Z is a general set:
  - We used to take $Z = \mathcal{X} \times \mathcal{Y}$.
  - For instance, in unsupervised tasks, we could have $Z = \mathcal{X}$.
- Now we need to consider data distributions for this Z. Given this loss function ℓ, the error w.r.t. the data distribution $\mathcal{D}$ will be

$$L_{\mathcal{D}}(h) := \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$$

beyond binary classification

A hypothesis class $\mathcal{H}$ is said to be **agnostic PAC learnable w.r.t.** a set Z and a loss function $\ell: \mathcal{H} \times Z \to \mathbb{R}_+$, if there exists a function $m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over Z, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns a hypotheses h such that,
- with probability of at least $1 - \delta$ (over the choice of the examples),

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$

- where $L_{\mathcal{D}}(h) := \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$.

Questions?

Reference

- *PAC learning :*
  - *[S-S] Ch 2.1-2.3, 3.1*
- *Agnostic PAC learning:*
  - *[S-S] Ch 3.2-3.3, 4.1-4.3*