Trustworthy Machine Learning

Statistical Learning Theory

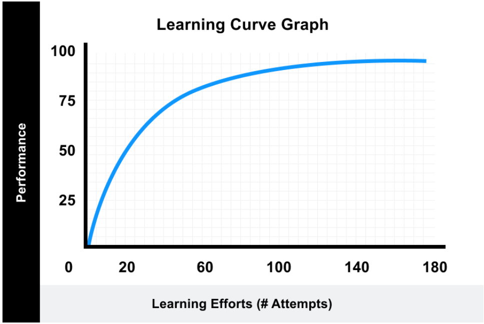
Sangdon Park

POSTECH

February 24, 2025

What is Learning Theory?

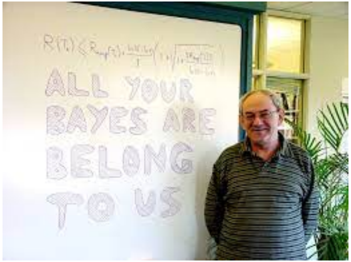
Theory on exploring conditions (or assumptions) when machines can learn from data.



https://www.valamis.com/hub/learning-curve

- Statistical learning theory
- Online learning theory

Historical Figure: Vladimir Vapnik



vapnik

FOLLOW

Professor of Columbia, Fellow of NEC Labs America,
Verified email at nec-labs.com

machine learning   statistics   computer science

TITLE	CITED BY	YEAR
The Nature of Statistical Learning Theory V Vapnik Data mining and knowledge discovery	104201 *	1995
Support-vector networks C Cortes, V Vapnik Machine learning 20, 273-297	62445	1995
A training algorithm for optimal margin classifiers BE Boser, IM Guyon, VN Vapnik Proceedings of the fifth annual workshop on Computational learning theory ...	16380	1992
Backpropagation applied to handwritten zip code recognition Y LeCun, B Boser, JS Denker, D Henderson, RE Howard, W Hubbard, ... Neural computation 1 (4), 541-551	15122	1989
Gene selection for cancer classification using support vector machines I Guyon, J Weston, S Barnhill, V Vapnik Machine learning 46, 389-422	11033	2002
Support vector regression machines H Drucker, CJ Burges, L Kaufman, A Smola, V Vapnik Advances in neural information processing systems 9	6005	1996

- "The Nature of Statistical Learning Theory": summary of his papers up to 1995.
- VC dimension, SVM, ...

Historical Figure: Leslie Valiant

Leslie Valiant

Unknown affiliation
No verified email

FOLLOW

TITLE	CITED BY	YEAR
A theory of the learnable LG Valiant Communications of the ACM 27 (11), 1134-1142	7939	1984
A bridging model for parallel computation LG Valiant Communications of the ACM 33 (8), 103-111	5399	1990
The complexity of computing the permanent LG Valiant Theoretical computer science 8 (2), 189-201	3413	1979
The complexity of enumeration and reliability problems LG Valiant siam Journal on Computing 8 (3), 410-421	2579	1979
Cryptographic limitations on learning boolean formulae and finite automata M Kearns, L Valiant Journal of the ACM (JACM) 41 (1), 67-95	1318	1994
Random generation of combinatorial structures from a uniform distribution MR Jerrum, LG Valiant, VV Vazirani Theoretical computer science 43, 169-188	1218	1986

- "PAC Learning Theory" in 1984
- Turing Award winner in 2010

Four Key Ingredients of Learning Theory

The simplified objective of *statistical* learning theory:

$$\begin{aligned} \text{find} \quad & f \\ \text{subj. to} \quad & f \in \mathcal{F} \\ & \mathbb{E}_{(x,y)\sim D}\, \ell\left(x, y, f\right) \leq \varepsilon \end{aligned}$$

or

$$\min_{f\in\mathcal{F}} \mathop{\mathbb{E}}_{(x,y)\sim D} \ell\left(x, y, f\right)$$

- **Ingredient 1**: A distribution D (*e.g.*, a distribution over labeled images)
- **Ingredient 2**: Hypothesis space $\mathcal{F}$ (*e.g.*, linear functions, a set of resnet)
- **Ingredient 3**: A loss function ℓ (*e.g.*, 0-1 loss, L1 loss, cross-entropy loss)
- **Ingredient 4**: A learning algorithm (*e.g.*, GD)

Main Goal: Finding Conditions for Learnability

An Example

Conditions:

- D: *linearly separable* dog and cat image distribution
- $\mathcal{F}$: linear functions – encode prior of a data distribution
- ℓ: 0-1 loss for classification – represent task
- a learning algorithm: a gradient descent (GD) algorithm

Checking Learnability:
If we prove that the GD algorithm can find the true linear function with a "desired level" of loss, we say $\mathcal{F}$ is learnable. In this case, we say the GD algorithm is a "good" algorithm.

Contents from

CS229T/STAT231: Statistical Learning Theory (Winter 2016)

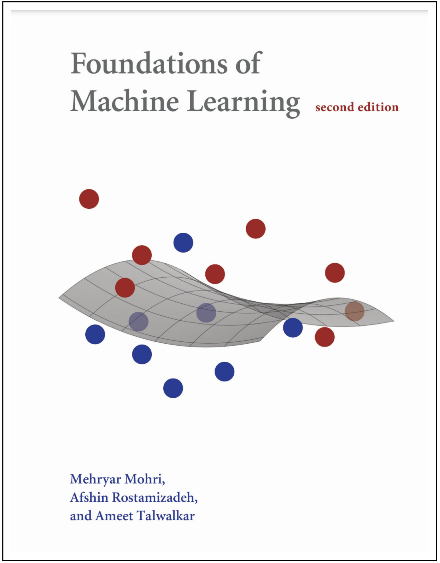
Percy Liang

Last updated Wed Apr 20 2016 01:36

These lecture notes will be updated periodically as the course goes on. The Appendix describes the basic notation, definitions, and theorems.

Contents

1



and various papers.

Why PAC Learning?

The key questions in machine learning:

- When can we learn?
- How many samples do we need to have a good model?

The PAC framework provides partial answers to these key questions.

Recall Four Key Ingredients of Learning Theory

- Distribution – setup / assumption
  - image distribution, language distribution
  - samples are independently drawn from the same distribution
- Loss – a goodness metric for a desired task
  - classification: 0-1 loss
  - regression: L1 loss
- Hypothesis space – prior on the distribution, what we will design!
  - convolution network: good for image classification
  - transformers: good for language modeling
- A learning algorithm – what we will design!
  - gradient descent (GD) method...

Assumption on Distributions

Assumption

We assume that labeled examples are independently drawn from the same (and unknown) distribution $\mathcal{D}$ over labeled examples $\mathcal{X} \times \mathcal{Y}$.

- "independent": not sequential data
- "unknown": yes, we don't know the true distribution
- "same": key for success
- A.K.A. the i.i.d. assumption
- The i.i.d. assumption is the standard setup.
- It is easily broken due to distribution shift.
- Online learning relaxes this assumption (under some conditions).

A Goodness Metric: Expected Error for Classification

Definition (expected error)

Given a hypothesis $h \in \mathcal{H}$ and an underlying distribution $\mathcal{D}$, the expected error is defined by

$$L(h) := \mathbb{P}\left\{h(x) \neq y\right\} = \mathbb{E}\left\{\mathbb{1}\left(h(x) \neq y\right)\right\},$$

where the probability is taken over $(x, y) \sim \mathcal{D}$ and $\mathbb{1}$ is the indicator function.

- Suppose the classification task. But, we can use any task-dependent loss.
- This expected error of h is sometimes called the *risk* of h or the *generalization error* of h.
- The indicator function is defined as follows:

$$\mathbb{1}\left(s\right) := \begin{cases} 1 & \text{if } s \text{ is true} \\ 0 & \text{if } s \text{ is false} \end{cases}.$$

A Goodness Metric: Empirical Error

Definition (empirical error)

Given a hypothesis $h \in \mathcal{H}$ and labeled samples $\mathcal{S} := ((x_1, y_1), \cdots, (x_n, y_n))$, the empirical error is defined by

$$\hat{L}(h) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(h(x_i) \neq y_i\right),$$

where $\mathbb{1}$ is the indicator function.

- This empirical error of h is sometimes called the *empirical risk* of h.

One More Assumption

Assumption

We assume that a distribution $\mathcal{D}$ is separable by some hypothesis $h^* \in \mathcal{H}$, i.e.,

$$L(h^*) = 0.$$

- Equivalently, we can consider a true hypothesis h^* from which a label $y = h^*(x)$ is generated; in this case, a distribution is only defined over $\mathcal{X}$.
- This assumption is strong but useful in some cases (*e.g.*, PAC conformal prediction).
- This assumption will be removed later (in a more general learning framework).

Approximately Correct

A Goodness Metric for Algorithms

Definition

Given $\varepsilon > 0$, we say that h is approximately correct if

$$L(h) \leq \varepsilon.$$

- ε is a user-defined parameter.
- Recall that L is an expected error.
- We want to find h that achieves a desired error level ε.
- h is learned from data; thus, h is also a random variable.

Probably Approximately Correct (PAC)

A Goodness Metric for Algorithms

Definition

Given $\varepsilon > 0$, $\delta > 0$, and $n \in \mathbb{N}$, we say that an algorithm $\mathcal{A}$ is probably approximately correct (PAC) if

$$\mathbb{P}\left\{L(\mathcal{A}(\mathcal{S})) \leq \varepsilon\right\} \geq 1 - \delta,$$

where $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{H}$ and the probability is taken over $\mathcal{S} := ((x_1, y_1), \ldots, (x_n, y_n)) \sim \mathcal{D}^n$.

- $S^* := \bigcup_{i=0}^{\infty} S^i$
- $\mathcal{S} \sim \mathcal{D}^n$: i.i.d. samples
- $\mathcal{A}$: a learning algorithm
- PAC is a property of an algorithm

PAC Learning Algorithm

Definition (simplified definition)

An algorithm $\mathcal{A}$ is a PAC-learning algorithm for $\mathcal{H}$ if for any $\varepsilon > 0$, $\delta > 0$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ separable by h^*, and for some minimum sample size n^* (which depends on $\varepsilon, \delta, \mathcal{D}$), the following holds with any sample size $n \geq n^*$:

$$\mathbb{P}\left\{L(\mathcal{A}(\mathcal{S})) \leq \varepsilon\right\} \geq 1 - \delta,$$

where $\mathcal{S} := ((x_1, y_1), \ldots, (x_n, y_n)) \sim \mathcal{D}^n$.

- Please check out the original PAC learning definition.
- The algorithm should satisfy the PAC guarantee for any $\mathcal{D}$ and h^*.
- If $\mathcal{D}$ is "complex" (thus h^* is complex), we need more samples.
- If ε (or δ) is small, we need more samples.

Example: A Learning Bound for a Finite Hypothesis Set I

Learning Setup:

- $\mathcal{H}$: a *finite* set of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$
  - *e.g.*, a set of experts
- $\mathcal{D}$: a distribution is separable by $h^* \in \mathcal{H}$
- $\mathcal{S}$: labeled examples
- $\mathcal{A}$: an algorithm that satisfies $\hat{L}(\mathcal{A}(\mathcal{S})) = 0$
  - *i.e.*, $\mathcal{A}$ returns a "consistent" hypothesis.
  - Here, the algorithm exploits the fact on the separability!

Example: A Learning Bound for a Finite Hypothesis Set II

Theorem

For any $\varepsilon > 0$, $\delta > 0$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ separable by h^*, we have

$$L(\mathcal{A}(\mathcal{S})) \leq \frac{1}{m}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)$$

with probability at least $1 - \delta$.

- $\mathcal{A}$ is a PAC learning algorithm.
- Sample complexity?

$$m \geq \frac{1}{\varepsilon}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)$$

  - See? As $\mathcal{H}$ gets complex and as ε and δ are smaller, we need more samples.
- **key**: A union bound over the events of each hypothesis.

Example: A Learning Bound for a Finite Hypothesis Set III

Lemma (a union bound)

Let $A_1, \ldots, A_K$ *be* K *different events (which might not be independent). Then,*

$$\mathbb{P}\left\{\bigcup_{k=1}^{K} A_k\right\} \leq \sum_{k=1}^{K} \mathbb{P}\{A_k\}.$$

- Recall the definition of a measure.

Example: A Learning Bound for a Finite Hypothesis Set IV

Proof Sketch:

Let $\mathcal{H}_\varepsilon := \{h \in \mathcal{H} \mid L(h) > \varepsilon\}$. Then, we have

$$\begin{aligned}
\mathbb{P}\left\{L(\mathcal{A}(\mathcal{S})) > \varepsilon\right\} &\le \mathbb{P}\left\{\exists h \in \mathcal{H}_\varepsilon, \hat{L}(h) = 0\right\} && (1)\\
&= \mathbb{P}\left\{\bigvee_{h \in \mathcal{H}_\varepsilon} \hat{L}(h) = 0\right\} \\
&\le \sum_{h \in \mathcal{H}_\varepsilon} \mathbb{P}\left\{\hat{L}(h) = 0\right\} && (2)\\
&\le \sum_{h \in \mathcal{H}_\varepsilon} (1-\varepsilon)^m && (3)\\
&\le |\mathcal{H}|(1-\varepsilon)^m.
\end{aligned}$$

- (1): we may want a (stronger) "uniform convergence" but data-agnostic bound
- (2): union bound due to the finite hypotheses
- (3): a special case of the "point" binomial tail bound due to the i.i.d. assumption, $\mathbb{1}\{h(x) \neq y\}$ is a Bernoulli random variable with a parameter of ε, and $m\hat{L}(h)$ is the sum of m Bernoulli random variables.

Next

Relax assumptions:

- What if we have an infinite hypothesis set?
- What if $\mathcal{D}$ is not separable?

We will explore a more general learning bound.