TECHNISCHE UNIVERSITÄT
BERGAKADEMIE FREIBERG
Die Ressourcenuniversität. Seit 1765.

Prof. Dr. Björn Sprungk
Faculty of Mathematics and Computer Science
Institute of Stochastics

Mathematics of machine learning

3. Statistical learning theory

Winter term 2024/25

Chapter 3: Statistical learning theory

Contents

3.1 Consistency and PAC-Learning

3.2 Uniform convergence

2.3 The VC dimension

2.4 Extensions of the VC dimension

Chapter 3: Statistical learning theory

What it's about?

- Tools and concepts to control the estimation error $\varepsilon_{\text{est}}(S)$:

  „How many training data do I need to make ε_{est} sufficiently small?“

- Understanding how ε_{est} relates to the „complexity“ of hypothesis classes $\mathcal{H}$

- The Vapnik–Chervonenkis dimension and the fundamental theorem of learning

Recalling statistical learning

Given

- a feature space $\mathcal{X}$ and a label space $\mathcal{Y}$,
- a distribution μ on $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$, which is unknown to us,
- a sample $s = \big((x_1, y_1), \dots, (x_m, y_m)\big)$ of $m \in \mathbb{N}$ independent realizations of $(X, Y) \sim \mu$

Learning

H is a mapign of h

We want to learn a hypothesis $h\colon \mathcal{X} \to \mathcal{Y}$ predictiong the label y given feature x via

- a chosen hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$,
- a learning algorithm $A\colon \mathcal{D}^m \to \mathcal{H}$ obtaining $h_s = A(s)$, in particular, we use the (biased) ERM-rule

$$h_s = \mathrm{ERM}_{\mathcal{H}}(s) \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_s(h) = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(h, (x_i, y_i)),$$

- based on a chosen loss function $\ell\colon \mathcal{H} \times \mathcal{D} \to [0, \infty)$.

Example: Learning a linear hypothesis

how many data required

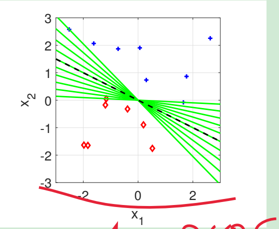
- Goal: Learning the true hypothesis $h^\dagger \colon \mathbb{R}^2 \to \{\pm 1\}$

$$Y = h^\dagger(X) = \operatorname{sgn}\left(w^\dagger \cdot X\right), \qquad w^\dagger = (1,2)^\top$$

where $X \sim \mathrm{U}[-3,3]^2$

↑ deg

$m = 15$ data points



gen error is high

- We choose the hypothesis class

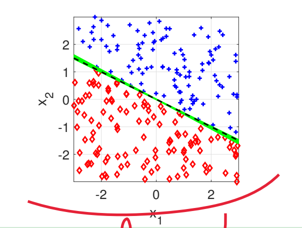
$$\mathcal{H} = \left\{h(x) = \operatorname{sgn}\left(w \cdot x\right) : w \in \mathbb{R}^2\right\}$$

so

$$\min_{h \in \mathcal{H}} \mathcal{R}_\mu(h) = 0$$

approximation error is 0

$m = 15^2$ data points



gen error is low here

- However, learning via the ERM rule with 0-1-loss, i.e.,

$$h_s \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_s(h) = \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{m} \left|\{i \colon h(x_i) \neq y_i\}\right|$$

can yield hypotheses h_s with high risk $\mathcal{R}_\mu(h_s)$ depending on m

The goal of Statistical Learning Theory is to understand and provide mathematical guarantees for how well a learning algorithm can generalize from a limited amount of data. such as:

1. How can we measure the model's performance on new, unseen data?
2. How can we minimize the error of the model on future data?
3. What conditions or assumptions ensure that our model will generalize well?

The purpose to use stats learning theory is:

1. Minimizing Generalization Error
2. Balancing Bias and Variance
3. Quantifying the Learning Process: Through concepts like PAC (Probably Approximately Correct) learning and VC (Vapnik–Chervonenkis) dimension, the theory provides ways to mathematically quantify how much data is needed and how complex a model can be to ensure good performance on new data.
4. Also we want to know how error **Estimation depend of the hypothesis class and the random traning data.**

Therefore we use this theory because due to neutral learning, this assumption of the error estimation is total random.

The goal of statistical learning theory

- We study and bound the estimation error

$$\varepsilon_{\text{est}} = \varepsilon_{\text{est}}(\mathcal{H}, s) = \underline{\mathcal{R}_\mu(h_s) - \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)}$$

  which is due to not being able to compute (and minimize) $\mathcal{R}_\mu$ just its empirical approximation $\mathcal{R}_s$

  no determinist approach

- In particular, we want to know

  1. How does $\varepsilon_{\text{est}}(\mathcal{H}, s)$ depend on the chosen class $\mathcal{H}$?
  2. How does $\varepsilon_{\text{est}}(\mathcal{H}, s)$ depend on the amount $m = |s|$ of training data ?

- To this end, we have to take a probabilistic or statistical approach, because due to neutral learning assumption the estimation error ε_{est} is random

$$\varepsilon_{\text{est}}(\mathcal{H}, S) = \mathcal{R}_\mu(h_S) - \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h), \qquad S \sim \mu^m.$$

In machine learning, **consistency means that if we have enough training data, our model will eventually make fewer mistakes and predict accurately** on all examples in the dataset. Essentially, a consistent algorithm can perfectly fit the training data as the data grows.

A learning algorithm is **consistent if**, for any number of training examples, **the error on the training set (called empirical error and the estimation error) is zero or goes to zero** as the data grows. This means the algorithm’s predictions match the actual labels of the data.The goal is to have an algorithm A that, as the sample size $m \to \infty$ (more data), produces hypotheses that get closer and closer to the best possible hypothesis in H, in terms of expected risk over μ.In this set the universally consistent is that as the data gets bigger m->infinity, we would have our error estimation = 0.

The universal consistency says that the probability of having an estimation error larger than the tolerance rate or the allowable error goes to zero. which means our hypothesis, the more the data we get the risk and the error with this method (universal consistency) approaches zero. regardless of the data distribution mu.in this case we could say with enough data and regardless of the data distribution, the algorithm's performance will be closed to the best possible performance with probability.

PAC learning or probably approximately Correct, is a type of learning that we want our model's performance to reach its highest vantage point by calculating the probability of the Generalization error to be smaller than the summation of the estimate error and the maximum allowable error, which this probability is going to be bigger and greater than or the success probability (which can be calculated by 1 - failure probability)

Here the generalization error could be soo low which close to the best possible in order ot have a qualified trained hypothesis.

In other words, we’re probably getting an error rate approximately as low as the best we could hope for in our hypothesis space, which means our learning process is working well in general.

So when we say, for example, that $1-\delta=0.95$, this means we’re 95% confident (or have a 95% success probability) that our model's generalization error will be close enough to optimal. This success probability provides assurance that the model will perform well on new, unseen data. which is generalization error task.

If the $1-\delta$ is low, then the generalization error (which again quatifies how well the hypothesis is near the optimal true value (not the predicted the actual value) which is again bigger says that our model qutifies very bad. which is not Probably Approximately Correct.

PAC learning provides a framework to ensure that, with a high probability, our model will perform close to the best possible on unseen data.

3.1 Consistency and PAC-Learning

correct one ✓

- **Hope:** With increasing amount of training data $|S| = m \to \infty$ we shall learn a hypothesis $h_S = A(S)$ which performs arbitrarily close to the best one among all $h \in \mathcal{H}$

$$\underbrace{\mathcal{R}_\mu(A(S))}_{\text{gen error}} \xrightarrow[m\to\infty]{} \underbrace{\inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)}_{\text{app error (deterministic)}} \qquad \iff \qquad \varepsilon_{\text{est}}(S) \xrightarrow[m\to\infty]{} 0$$

- Since S and, thus, $\varepsilon_{\text{est}}(S)$ are random we may ask for convergence in probability:

↳ basic

Definition 3.1:

Given a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and a loss ℓ we call a learning algorithm $A\colon \bigcup_{m\in\mathbb{N}} \mathcal{D}^m \to \mathcal{H}$ (universally) consistent if for any distribution μ on $\mathcal{D}$ we have with $S \sim \mu^m$

$$\varepsilon_{\text{est}}(\mathcal{H}, S) \xrightarrow[\underbrace{m\to\infty}_{\text{more data}}]{\mathbb{P}} 0 \qquad \iff \qquad \forall \epsilon > 0\colon \underbrace{\lim_{m\to\infty} \mathbb{P}_{\mu^m}}_{\text{tolerance bound}} \Big(\underbrace{\mathcal{R}_\mu(A(S)) - \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)}_{\varepsilon_{\text{est}}} > \underbrace{\boxed{\epsilon}}_{\text{tolerance rate}} \Big) = 0.$$

PAC learning (how many data enough) * **

- For finite data $m < \infty$ we can not expect to obtain the minimal risk $\mathcal{R}_\mu(h_S) = \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)$
- It suffices if h_S achieves a small risk with high probability:

PAC condition (Probably Approximately Correct)

For a tolerance bound $\epsilon > 0$ and failure probability $\delta \in (0,1)$ it holds
acceptable level of error

$$\mathbb{P}_{\mu^m}\left(\overset{e_{gen}}{\underbrace{\mathcal{R}_\mu(h_S)}} \le \underbrace{\inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)}_{\varepsilon_{app}} + \epsilon\right) \ge \underbrace{1-\delta}_{\text{success probability}} \tag{PAC}$$

or in short: $\mathbb{P}_{\mu^m}(\varepsilon_{\text{est}}(\mathcal{H}, S) \le \epsilon) \ge 1 - \delta.$

- **Question:** Can we achieve (PAC) for any $\epsilon > 0$ and any $\delta \in (0,1)$ given a sufficiently large sample size $m = m(\epsilon, \delta) < \infty$?

Definition 3.2: PAC-learnability

A hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is called (agnostic) PAC-learnable w.r.t. a given loss ℓ, if there exists

- a mapping $m_{\mathcal{H}} \colon (0,1)^2 \to \mathbb{N}$ and

- a learning algorithm A,

such that

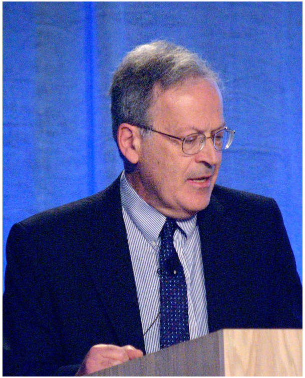
- for *any* data distribution μ on $\mathcal{X} \times \mathcal{Y}$ and

- *any* $\epsilon \in (0,1)$ and $\delta \in (0,1)$,

we satisfy the condition (PAC) for $h_S = A(S)$ and $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, i.e.,

$$\mathbb{P}_{\mu^m}\left(\mathcal{R}_\mu(A(S)) \leq \inf_{h \in \mathcal{H}} \mathcal{R}_\mu(h) + \epsilon\right) \geq 1 - \delta \qquad \underbrace{\forall m \geq m_{\mathcal{H}}(\epsilon, \delta).}_{\text{number of samples available}}$$

The smallest such mapping $m_{\mathcal{H}}$ is called sample complexity of $\mathcal{H}$ and we will always work with $m_{\mathcal{H}}$ being the sample complexity.



Leslie Valiant
(*1949)

PAC learnability and consistency

Proposition 3.3:

If a hypothesis class $\mathcal{H}$ is PAC-learnable w.r.t. a given loss ℓ by a learning algorithm A, then this learning algorithm is consistent for $\mathcal{H}$ and ℓ.

Remarks:

- The converse is not true, because the sample complexity $m_{\mathcal{H}}$ has to apply to any distribution μ.
- Sometimes, consistency is defined in the literature by

$$\lim_{m\to\infty} \mathbb{E}_{\mu^m}[\mathcal{R}_\mu(A(S))] = \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h) \tag{Con}$$

  in case of bounded loss functions ℓ.

- PAC learnability also implies (Con) and allows to quantify the (universal) speed of convergence (in probability) of $\mathcal{R}_\mu(A(S)) \to \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)$ as well as in (Con).

PAC learnability is not a given!

Theorem 3.4: ("No-Free-Lunch" theorem)

Let $\mathcal{X}$ be finite and $|\mathcal{Y}| = 2$. Further, let ℓ be the 0-1 loss and A be an arbitrary learning algorithm. Then, for any sample size $m < |\mathcal{X}|/2$, there exists a distribution μ on $\mathcal{X} \times \mathcal{Y}$ and a hypothesis $h^\star \colon \mathcal{X} \to \mathcal{Y}$ such that

$$\mathcal{R}_\mu(h^\star) = 0 \qquad \text{but} \qquad \mathbb{P}_{\mu^m}\left(\mathcal{R}_\mu(A(S)) \geq \frac{1}{8}\right) \geq \frac{1}{7}.$$

Proof: See Chapter 5 in "Understanding Machine Learning" (2014).

NFL therom states that no matter what learning algorithm we choose, there will always be a distribution where the algorithm performs poorly with a non-negligible probability. In another words, it states that no single learning algorithm is universally the best for all possible problems. In simple terms, if an algorithm works well for some tasks, it must perform poorly on others. There is no "one-size-fits-all" algorithm. Here it means that PAC learnability is not guaranteed for every learning problem. because it only works with arbitrary learning rule and structured distributions not total random. the number of data is less than half of the whole data selected. also There always exists at least one probability distribution . without assumptions about the structure of the data, learning is impossible.

Consequences of the NFL theorem

- **No universal learner:** There is no learning algorithm A which suceeds on all learning tasks.
- **Prior knowledge required:** We need to restrict to suitable hypotheses classes $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ for learnability

Corollary 3.5:

Let $\mathcal{X}$ be infinite, e.g., $\mathcal{X} = \mathbb{N}$ or $\mathcal{X} = [0,1]$, then the class of all hypotheses $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$ is not PAC-learnable for binary classification with 0-1 loss.

- But we need to **choose $\mathcal{H}$ with care**, because even simple binary hypothesis classes on $\mathcal{X} = [0,1]$, such as

$$\mathcal{H} = \{h_w(x) = \operatorname{sgn}(\sin(w\,x)) : w \in \mathbb{R}\},$$

can be not PAC-learnable.

PAC learning works only when we assume the target function is not arbitrary and that the data follows a structured probability distribution.

If the data were completely random, no learning algorithm would perform better than guessing.

Where is the problem?

- By the assumption of neutral learning we have $(X_i, Y_i) \sim \mu$ i.i.d. and thus

$$Z_i := \ell(h, (X_i, Y_i)) \quad \text{are i.i.d. with } \mathbb{E}[Z_i] = \mathcal{R}_\mu(h)$$

- Then, By the law of large numbers we have for **fixed** $h \in \mathcal{H}$

$$\mathcal{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} Z_i \xrightarrow[m\to\infty]{\mathbb{P}} \mathbb{E}[Z_i] = \mathcal{R}_\mu(h)$$

- However, this does **not** imply that, e. g., for the ERM rule

$$h_S = \operatorname*{argmin}_{h\in\mathcal{H}} \mathcal{R}_S(h) \xrightarrow[|S|\to\infty]{\mathbb{P}} \operatorname*{argmin}_{h\in\mathcal{H}} \mathcal{R}_\mu(h) \qquad \text{or} \qquad \mathcal{R}_\mu(h_S) \xrightarrow[|S|\to\infty]{\mathbb{P}} \inf_{h\in\mathcal{H}} \mathcal{R}_\mu(h)$$

- For this to hold, we need a uniform convergence

$$\sup_{h\in\mathcal{H}} |\mathcal{R}_S(h) - \mathcal{R}_\mu(h)| \xrightarrow[|S|\to\infty]{\mathbb{P}} 0$$

$\varepsilon_{gen} =$