Yale University CPSC 683
Stability in Machine Learning: Generalization, Privacy & Replicability

Lecture 1: Generalization in Machine Learning

Instructor: Alkis Kalavasis, alkis.kalavasis@yale.edu

In this first Lecture, we will cover the standard PAC learning framework for binary classification and discuss about the course plan for the upcoming Lectures.

1 What is Generalization in Machine Learning?

Generalization in statistical prediction is almost second nature to human reasoning. The work of Zhang et al. (2021) provides the following beautiful summary of what is generalization: Scientists, policymakers, and economists have capitalized on the empirical observation that *patterns in collected data often carry over when predicting future events or explaining unobserved phenomena*. This means that uncertain outcomes often follow patterns observed in past data. We call the process of identifying and leveraging these patterns *generalization*.

Towards formalizing and explaining generalization, Machine Learning theory is grounded on a statistical baseline. We assume that observations come from a fixed data generating process. Having collected the training data, we fit a model to the training set. Next, during testing, we evaluate the model by how well it performs on unseen generated data from the same process. This setting is extremely simple but it already raises interesting questions, which we are going to discuss today.

Consider an input (or feature) space $\mathcal{X}$ and a label space $\mathcal{Y}$. For this introduction, we will focus on the following classical learning problem. A binary classification problem is defined by a distribution $\mathcal{D}$ over labelled examples $(x, y) \in \mathcal{X} \times \{0, 1\}$.

We will need to important notions, that of a classifier and that of a learning algorithm.

Definition 1: Classifier and Learning Algorithm

- A classifier h is a mapping from $\mathcal{X}$ to $\mathcal{Y}$.

For the following we fix a training set size $n \in \mathbb{N}$:

- A deterministic learning algorithm (or learner) $A = A_n$ is a mapping from $(\mathcal{X} \times \mathcal{Y})^n$ to $\mathcal{Y}^{\mathcal{X}}$, which maps a training set S of size n to a classifier $A(S)$ from $\mathcal{X}$ to $\mathcal{Y}$.
- A randomized learning algorithm (or learner) $A = A_n$ is a mapping from $(\mathcal{X} \times \mathcal{Y})^n$ to $\Delta(\mathcal{Y}^{\mathcal{X}})$, which maps a training set S of size n to a distribution $A(S)$ over classifiers. [a] [b] [c]

[a] Here $\Delta(\cdot)$ denotes the probability simplex.
[b] In general, the training set S (which will contain labeled examples) will be drawn from some distribution $\mathcal{D}$. Hence, the randomness of the output of a deterministic learner is only due to the randomness of S. On the other side, a randomized learner has also internal randomness, which corresponds to eventually sampling a classifier from $\Delta(\mathcal{Y}^{\mathcal{X}})$.
[c] More formally, a learning algorithm A is a sequences of mappings $(A_n)_{n \in \mathbb{N}}$, where n is the training set size.

The learning algorithm does not know $\mathcal{D}$, but is able to collect a sample of n i.i.d. examples from $\mathcal{D}$. Then, during the training phase, the learner uses these n examples to build a classifier $\widehat{h}_n : \mathcal{X} \to \{0, 1\}$. The objective of the learner is to achieve small generalization error

$$R(\widehat{h}_n) = \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}} \left[\widehat{h}_n(x) \neq y \right] .$$

Observe that this quantity is a random variable depending on the randomness of the training set, used to obtain $\widehat{h}_n$. Making this quantity small intuitively means that the learning algorithm has extracted from the training data all the useful patterns hidden in $\mathcal{D}$ and can explain unobserved samples coming from $\mathcal{D}$.

2 VC Theory and Uniform Convergence

Since the data distribution $\mathcal{D}$ is unknown to the learner, our theory of learning must be expressed in terms of some restrictions on $\mathcal{D}$. To this end, the Probably Approximately Correct (PAC) model of learning (Vapnik, 2013; Valiant, 1984) introduces a concept class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ (a class of functions from $\mathcal{X}$ to $\{0,1\}$) and aims to understand learnability with respect to the this (known) class. Introducing this concept class is crucial since now we can state assumptions about $\mathcal{D}$.

The simplest and most standard assumption is that $\mathcal{D}$ is realizable with respect to $\mathcal{H}$, in the sense that

$$\inf_{h \in \mathcal{H}} R(h) = 0.$$

In words, $\mathcal{H}$ contains hypotheses with arbitrarily small generalization error[1]. We denote the class of realizable distributions by $\mathrm{RE}(\mathcal{H})$.

The fundamental result of PAC learning theory (Blumer et al., 1989; Vapnik, 2013; Vapnik and Chervonenkis, 1971) states that:

$$\inf_{\widehat{h}_n} \sup_{\mathcal{D} \in \mathrm{RE}(\mathcal{H})} \mathop{\mathbf{E}}_{S \sim \mathcal{D}^n} \left[R(\widehat{h}_n) \right] = \Theta \left(\min \left\{ \frac{\mathrm{VC}(\mathcal{H})}{n}, 1 \right\} \right) \text{ for all } n \in \mathbb{N}\,, \tag{1}$$

where the expectation is over the training set of size n drawn i.i.d. from $\mathcal{D}$ and the classifier $\widehat{h}_n$ is the output of the learning algorithm trained on S. Observe that the PAC model is truly worst-case since it quantifies the expected generalization of an algorithm against the worst-case possible distribution (which is allowed to change with the sample size). Recall that $\mathrm{VC}(\mathcal{H})$ is corresponds to the Vapnik-Chervonenkis (VC) dimension, which is defined as follows.

Definition 2: Shattering

Given a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a set of points $S = \{x_1, ..., x_n\} \subseteq \mathcal{X}$, we define the projection of $\mathcal{H}$ on S as

$$\mathcal{H}_S = \{(h(x_1), ..., h(x_n)) : h \in \mathcal{H}\} \subseteq \{0,1\}^n\,.$$

If $|\mathcal{H}_S| = 2^n$, we say that S is shattered by $\mathcal{H}$.

If a set S is shattered by $\mathcal{H}$, it means that the concept class has the expressive power to create all possible $2^{|S|}$ labelings of size $|S|$. Roughly speaking, being able to shattering very 'large' sets, it intuitively means that the concept class should be 'harder' to learn. The largest size that a class can shatter corresponds to its VC dimension.

Definition 3: VC Dimension

Given a class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, the VC dimension of $\mathcal{H}$ is the largest set size d that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of any size, then its VC dimension is ∞.

To give some examples, the VC dimension of thresholds over $\mathbb{R}$ is 1 but the VC dimension of the function class $x \mapsto \sin(\theta x)$ for $\theta \in \mathbb{R}$ is infinite. Going back to (1), the minimax formulation corresponds to how

[1]The notion of realizability is a little bit subtle. For instance, a stronger notion is that there exists an $h \in \mathcal{H}$ that realizes all the observed labels and achieves $R(h) = 0$. The infimum definition is weaker in the sense that it guarantees that for any $\epsilon > 0$, there exists some hypothesis h_ϵ such that $R(h_\epsilon) < \epsilon$. During the course, we will specify when each definition is used.

well the best classifier $\widehat{h}_n$ can perform against the worst-case possible realizable distribution $\mathcal{D}$. Note that $\widehat{h}_n$ does not have to belong to $\mathcal{H}$; if it does, it is called a *proper* learner, otherwise it is called *improper*. In fact, the algorithm that gets the optimal sample complexity is improper (Hanneke, 2016) and we know precisely when optimal proper learning is possible (Bousquet et al., 2020).

Hence, the classical theory of PAC learning gives a beautiful dichotomy on how fast the minimax generalization error goes to 0 with the number of samples n: *every concept class $\mathcal{H}$ has a rate that is either linear c/n or bounded away from zero, depending on the finiteness of the combinatorial parameter* $\mathrm{VC}(\mathcal{H})$.

It is important to mention that we will mostly discuss about statistical learning theory ignoring *computational* aspects (such as what is the computational complexity of finding a hypothesis that generalizes well).

There is a crucial reason why this theory is satisfying. The idea is that learnability comes with a learning algorithm and that this algorithm is the most natural and actually applied algorithm, namely *Empirical Risk Minimization* (ERM). That is, the fact that every VC class is PAC learnable is witnessed by any algorithm that outputs a concept $h \in \mathcal{H}$ that is consistent with the input sample (i.e., it minimizes the training error).

This follows from the celebrated *uniform convergence theorem* of Vapnik and Chervonenkis (1971). Moreover, any ERM algorithm achieves the optimal uniform learning rate, up to lower order factors. We also mention that this landscape extends (perhaps surprisingly) to the agnostic setting. In the agnostic setting, no assumptions are made on $\mathcal{D}$ but the learning algorithm competes with the best predictor in the fixed class $\mathcal{H}$, i.e., the goal is to make the difference $R(\widehat{h}_n) - \inf_{h\in\mathcal{H}} R(h)$ as small as possible. For that, the linear rate of the realizable case is replaced by the slower rate of $\Theta(\min\{\sqrt{\mathrm{VC}(\mathcal{H})/n}, 1\})$.

Uniform convergence refers to the phenomenon where, given a sufficiently large sample from a population, the empirical losses of *all* concepts $h \in \mathcal{H}$ approximate their true population losses. The crucial point is that this happens *uniformly* over all functions in $\mathcal{H}$. This immediately implies an algorithmic principle: pick a concept in the class that minimizes empirical loss, which for a given training set S of size n, is defined as

$$\widehat{R}_S(h) = \frac{1}{n} \sum_{(x,y)\in S} 1\{h(x) \neq y\}\,.$$

Definition 4: Uniform Convergence

A concept class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ has the uniform convergence (UC) property (or is a uniform Glivenko-Cantelli class[a]) if there is a function $n_{UC} : (0,1)^2 \to \mathbb{N}$ such that for any $\epsilon, \delta \in (0,1)$ and any $\mathcal{D}$, if S is a training set of size $n \geq n_{UC}(\epsilon,\delta)$ drawn i.i.d. from $\mathcal{D}$, then with probability $1-\delta$, it holds

$$\sup_{h\in\mathcal{H}} \left|\widehat{R}_S(h) - R(h)\right| \leq \epsilon\,.$$

[a]The name comes from the famous Glivenko-Cantelli theorem in probability theory where the empirical CDF F_n of a CDF F satisfies $\|F_n - F\|_\infty = \sup_x |F_n(x) - F(x)| \to 0$ almost surely.

Some remarks are in order. Observe that, in the above definition, we do not need to assume that $\mathcal{D} \in \mathrm{RE}(\mathcal{H})$. In fact, the standard proof of the fundamental theorem of PAC learning goes by showing that UC $\Rightarrow$ Agnostic PAC Learning $\Rightarrow$ Realizable PAC Learning (the second implication is by definition). Note that n_{UC} for $\mathcal{H}$ will scale linearly with $\mathrm{VC}(\mathcal{H})$ and this will give the rate of the fundamental VC theorem, which we now state.

Theorem 1: Fundamental Theorem of Statistical Learning

Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$. Then the following are equivalent:

1. $\mathcal{H}$ has the uniform convergence property.
2. Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$.

3. $\mathcal{H}$ is agnostic PAC learnable.
4. $\mathcal{H}$ is PAC learnable (in the realizable setting).
5. Any ERM rule is a successful PAC learner for $\mathcal{H}$.
6. $\mathrm{VC}(\mathcal{H}) < \infty$.

Note that (1) is the quantitative version of realizable PAC learning.

2.1 A Proof of the Fundamental Theorem of PAC Learning

The proof is standard and appears in any introductory learning theory course or book (Shalev-Shwartz and Ben-David, 2014). I will just sketch the main ideas.

Observe that the implications $2 \to 3 \to 4$ and $2 \to 5$ are immediate. Hence, the interesting parts are $1 \to 2$ and $6 \to 1$. The implications $4 \to 6$ and $5 \to 6$ follow from the No-Free-Lunch Theorem (shortly, if $\mathcal{H}$ shatters some set of size $2m$, then we cannot learn $\mathcal{H}$ using m examples.).

Theorem 2: No-Free-Lunch Theorem

Let $\mathcal{H}$ be a hypothesis class of functions from $\mathcal{X}$ to $\{0,1\}$. Let m be a training set size. Assume that there exists a set $C \subset \mathcal{X}$ of size $2m$ that is shattered by $\mathcal{H}$. Then, for any learning algorithm A, there exist a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ and a classifier $h \in \mathcal{H}$ such that $R(h) = 0$ but with probability of at least $1/7$ over the choice of S we have that $R(A(S)) \geq 1/8$.

For the proof we refer to Shalev-Shwartz and Ben-David (2014). This result immediately implies that if the class has infinite VC dimension, then it is not PAC learnable. We now proceed to the other implications.

Idea for ($1 \to 2$) This essentially follows from the definitions of ERM and UC. Consider a training set S with the property that for any $h \in \mathcal{H}$, $|\widehat{R}_S(h) - R(h)|$ is small (say at most $\epsilon/2$). Such a set will be called $\epsilon/2$-good. Then it is not difficult to see that for any output h_{ERM} of the algorithm $\mathrm{ERM}_{\mathcal{H}}(S)$ with input S, the following will hold. First,

$$R(h_{\mathrm{ERM}}) \leq \widehat{R}_S(h_{\mathrm{ERM}}) + \epsilon/2\,,$$

by definition of the set S. Next, because of ERM, for any $h \in \mathcal{H}$,

$$\widehat{R}_S(h_{\mathrm{ERM}}) + \epsilon/2 \leq \widehat{R}_S(h) + \epsilon/2\,.$$

Again by the definition of S,

$$\widehat{R}_S(h) + \epsilon/2 \leq R(h) + \epsilon\,.$$

Chaining these inequalities, we get that $1 \to 2$ because (i) $R(h_{\mathrm{ERM}}) \leq \inf_{h \in \mathcal{H}} R(h) + \epsilon$ and (ii) n_{UC} measures the (minimal) sample complexity of obtaining the uniform convergence property, namely, how many examples we need to ensure that with probability of at least $1 - \delta$, the sample S would be ϵ-good.

Idea for $(6 \to 1)$ VC dimension controls the effective size of a class $\mathcal{H}$. We define the projection of $\mathcal{H}$ to a set $\{x_1, ..., x_m\}$ as $\mathcal{H}_{\{x_1,...,x_m\}} = \{(h(x_1), ..., h(x_m)) : h \in \mathcal{H}\}$. Typically, if $\mathrm{VC}(\mathcal{H}) = d$, then when projecting it to a finite set $C \subset \mathcal{X}$, its effective size $|\mathcal{H}_C|$ is $O(|C|^d)$ (and not $2^{|C|}$). This claim is often referred to as Sauer- Shelah-Perles lemma. Then one can show that uniform convergence holds whenever the hypothesis class has a "small effective size", i.e., when $|\mathcal{H}_C|$ grows polynomially with $|C|$.

To see this, we define the growth function as

$$\tau_{\mathcal{H}}(m) = \max_{C \subset \mathcal{X} : |C| = m} |\mathcal{H}_C|\,.$$

The SSP lemma reads as follows.

Lemma 1: Sauer- Shelah-Perles Lemma

Let $\mathcal{H}$ be a concept class with $\text{VC}(\mathcal{H}) \leq d < \infty$. For any sample size $m \in \mathbb{N}$, it holds that

$$\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}.$$

In particular, if $m > d+1$, then $\tau_{\mathcal{H}}(m) \leq (em/d)^d$ and if $m < d$, then $\tau_{\mathcal{H}}(m) = 2^m$.

Then for any $\mathcal{D}$, if one manages to show that

$$\mathop{\mathbf{E}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}} |R(h) - \widehat{R}_S(h)|\right] \leq \frac{4+\sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\sqrt{2m}},$$

then we get, by Markov's inequality, that with high probability, $\mathcal{H}$ has the uniform convergence property given that $m \approx \text{VC}(\mathcal{H})$. Showing the above inequality is the non-trivial (and most beautiful) part of the proof and requires the *symmetrization* technique, which we now discuss.

We can write $R(h) = \mathbf{E}_{S'\sim\mathcal{D}^m}[\widehat{R}_{S'}(h)]$, where $S' = z'_1, ..., z'_m$ is a 'ghost' sample. Then we can upper bound the LHS by

$$\mathop{\mathbf{E}}_{S,S'\sim\mathcal{D}^m} \sup_{h\in\mathcal{H}} \left|\frac{1}{m}\sum_{i=1}^{m} \ell(h,z_i) - \ell(h,z'_i)\right|$$

where $z_1, ..., z_m$ and $z'_1, ..., z'_m$ are i.i.d. draws from $\mathcal{D}$. This follows from the fact that suprermum of expectation is smaller than expectation of supremum. Here $\ell(h,z)$ is the 0-1 loss of the classifier h on the example z. Since these two samples are i.i.d., nothing will change if we interchange the random sample z_i with z'_i. This allows us to write that for any fixed $\sigma \in \{-1,1\}^m$,

$$\mathop{\mathbf{E}}_{S,S'\sim\mathcal{D}^m} \sup_{h\in\mathcal{H}} \frac{1}{m}\left|\sum_{i=1}^{m} \sigma_i(\ell(h,z_i) - \ell(h,z'_i))\right|.$$

The above also holds if we sample each component of σ uniformly at random from the uniform distribution, i.e., we can also write

$$\mathop{\mathbf{E}}_{S,S'\sim\mathcal{D}^m} \mathop{\mathbf{E}}_{\sigma\sim U(\{-1,1\}^m)} \sup_{h\in\mathcal{H}} \frac{1}{m}\left|\sum_{i=1}^{m} \sigma_i(\ell(h,z_i) - \ell(h,z'_i))\right|.$$

Fixing two datasets S, S', let C be the instances of $\mathcal{X}$ appearing in them; we can consider the supremum over the projection set $\mathcal{H}_C$ (this reduces $\mathcal{H}$ to a finite class over the elements $x_1, x'_1, ..., x_m, x'_m$). Then, we have that

$$\mathop{\mathbf{E}}_{\sigma\sim U(\{-1,1\}^m)} \max_{h\in\mathcal{H}_C} \frac{1}{m}\left|\sum_{i=1}^{m} \sigma_i(\ell(h,z_i) - \ell(h,z'_i))\right|.$$

For any fixed h, consider the random variable $\theta_h = \frac{1}{m}\sum_{i=1}^{m} \sigma_i(\ell(h,z_i) - \ell(h,z'_i))$, where the randomness arises from σ. Note that θ_h is zero-mean and is a sum of independent random variables on $[-1,1]$. Then $\mathbf{Pr}[|\theta_h| > t] \leq \exp\left(-2mt^2\right)$ and so,

$$\mathbf{Pr}\left[\max_{h\in\mathcal{H}_C} |\theta_h| > t\right] \leq |\mathcal{H}_C| \cdot \exp\left(-2mt^2\right).$$

The result follows since we can control the tails of the above non-negative random variable. In fact, its expected value is bounded by

$$\mathbf{E}\left[\max_{h\in\mathcal{H}_C} |\theta_h|\right] \leq \frac{4+\sqrt{\log|\mathcal{H}_C|}}{\sqrt{2m}}.$$

3 Reducing Agnostic to Realizable PAC Learning without UC?

As we saw, uniform convergence implies agnostic PAC learning, which trivially implies learnability in the realizable setting. Moreover, finite VC dimension implies uniform convergence. Hence, this establishes a perhaps surprising equivalence between realizable and agnostic PAC learning.

One of the goals of this course is to argue about learnability and generalization by avoiding uniform convergence arguments. As an example, in this section, we will ask whether the equivalence between realizable and agnostic PAC learning is fundamentally based on uniform convergence or if it can be shown in a more direct manner.

Vapnik and Chervonenkis (1971); Blumer et al. (1989)'s result is certainly a breakthrough in its own right, but its proof technique is too indirect to reveal any deeper connections between realizable and agnostic learning beyond the PAC setting. This led Hopkins et al. (2022) to raise the following natural question:

Is the equivalence of realizable and agnostic learning a fundamental property of learnability, or simply a happy coincidence derived from the uniform convergence?

Can we prove the equivalence between realizable and agnostic PAC learning result *without uniform convergence?* The answer is interestingly yes. This will be the content of this section, where we give a direct reduction from agnostic to realizable learning. The reduction is very simple and is presented right after:

- Input: Realizable PAC learner A, Unlabeled Sample Oracle O_U, Labeled Sample Oracle O_L.
- Algorithm:
  1. Draw an unlabeled sample $S_U \sim O_U$.
  2. Run A on all possible labelings of S_U to get
  $$C(S_U) = \{A(S_U, h(S_U)) : h \in H_{S_U}\}.$$
  3. Draw a labeled sample $S_L \sim O_L$.
  4. Output the element of $C(S_U)$ with the minimum training loss on the labeled sample S_L.

Note that as expected this algorithm is computationally inefficient since it requires the construction of the set $C(S_U)$ but also a search that set of hypotheses.

At its most basic, the correctness of the reduction is based on two observations.

- The set $C(S_U)$ almost certainly contains a near-optimal hypothesis. This will follow from the fact that A is a realizable PAC learner for $\mathcal{H}$.
- Second, we have to argue that Step 4 will actually find that near-optimal classifier. But since we know that it exists and since $C(S_U)$ has bounded size, we can employ standard uniform convergence for finite classes. This implies that if S_L is of order $\log |C(S_U)|$, ERM will find a hypothesis that is close to hypothesis that achieves $\min_{h \in \mathcal{H}} \mathbf{Pr}_{\mathcal{D}}[h(x) \neq y]$.

The key observation in this process is really that $C(S_U)$ 'acts like a cover' of $\mathcal{H}$ in the following weak sense we call non-uniform covering. In contrast to more classical notions, a non-uniform cover is a distribution over subsets of hypotheses that covers any fixed hypothesis in the class with high probability, but may fail to cover all hypotheses simultaneously.

Definition 5: Non-Uniform Cover

Let $\mathcal{H}$ be a hypothesis class over $\mathcal{X}$, $\mathcal{D}_X$ a marginal distribution over $\mathcal{X}$, and C a random variable over the power set $P(\mathcal{H})$. We call C a non-uniform (ϵ, δ)-cover of $\mathcal{H}$ with respect to $\mathcal{D}_X$ if

$$\forall h \in \mathcal{H} : \ \mathop{\mathbf{Pr}}_{C}[\exists h' \in C : \mathop{\mathbf{Pr}}_{x \sim \mathcal{D}_X}[h(x) \neq h'(x)] \leq \epsilon] \geq 1 - \delta \,.$$

Observe that if the quantifier $\forall h \in \mathcal{H}$ was inside $\mathbf{Pr}_C[\cdot]$, then the cover would be uniform since C would cover simultaneously all $h \in \mathcal{H}$. Here, in contrast, for every fixed hypothesis h in the class, C is very likely to contain a hypothesis close to h.

It is left to observe that $C(S_U)$ in Algorithm 1 is actually a non-uniform cover, but this is essentially immediate from the definition of realizable learning. Realizable learning promises that for every fixed hypothesis $h \in \mathcal{H}$, when A receives samples labeled by h it outputs a hypothesis close to h with high probability. $C(S_U)$ is generated by running A across all $h \in \mathcal{H}$, so this guarantee exactly translates to the above.

The above are summarized in two lemmas. First, we will show that the set $C(S_U)$ of outputs corresponding to running the realizable learner A across all possible labelings of the unlabeled sample S_U is a good approximation of the class $\mathcal{H}$.

Lemma 2

For any distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, there exists a hypothesis $h' \in C(S_U)$ with high probability such that

$$R(h') \leq \mathrm{OPT}_{\mathcal{D}} + \epsilon/2 \,.$$

If this lemma holds, the standard Chernoff concentration and a union bound over the elements in $C(S_U)$ implies that ERM will find a near-optimal classifier. It remains to argue about the above lemma. For this step, we will use the fact that $C(S_U)$ is a non-uniform cover: for any fixed $h \in \mathcal{H}$, $C(S_U)$ contains a hypothesis close to h with high probability.

Lemma 3

For any distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ and any hypothesis $h \in \mathcal{H}$, with high probability, there exists $h' \in C(S_U)$ such that $\mathbf{Pr}_{\mathcal{D}_X}[h(x) \neq h'(x)] \leq \epsilon/2$.

This lemma follows from the guarantees of a realizable PAC learner: algorithm A promises that for any $h \in \mathcal{H}$ and distribution $\mathcal{D}_X$, an $1 - \delta/2$ fraction of labeled samples $(S, h(S)) \sim \mathcal{D}^{n(\epsilon/2, \delta/2)}$ satisfy

$$\mathop{\mathbf{Pr}}_{\mathcal{D}_X \times h}[A(S, h(S))] = \mathbf{Pr}[h(x) \neq h'(x)] \leq \epsilon/2 \,.$$

Here h' is the output of A on $(S, h(S))$. Since $C(S_U)$ contains the output of A on any $h \in \mathcal{H}$, the result follows.

Essentially we have to apply the above Lemma for the hypothesis that realizes $\mathrm{OPT}_{\mathcal{D}}$.

4 Course Plan: Going Beyond VC Theory

In this course, we will try to revisit settings where VC theory and the uniform convergence (UC) tool are not sufficient to explain generalization and learnability. At the same time, the notion of stability as a property of the algorithm will be a key tool in going beyond the predictions of the classical theory. Let us see a particular motivation, which can provide some insight.

Insight 1: Generalization as a Property of Algorithms

Observe that in the UC definition, the supremum is taken over all $h \in \mathcal{H}$. Why this is bad? To see an example, think of perhaps the most standard learning theory problem: running stochastic gradient descent on the class of feed-forwards neural networks (NNs) with L layers and some nonlinear activation such as ReLU.

First, the VC dimension of these models grows at least linearly with the number of parameters in the network. Hence to achieve small excess risk or uniform convergence of sample averages to probabilities for discrete losses, the sample size must be large compared to the number of parameters in these networks. This behavior clearly does not explain *over-parameterization.*

However, there is a catch in the definition of uniform convergence: it is completely algorithm independent! While the space of possible NNs is very rich and complicated (having a VC dimension that scales with the number of parameters), maybe the standard training algorithm (say SGD) only visits a few of these networks. Hence, we do not want to have a bound that holds uniformly over the whole space of possible functions since we are interested in algorithms that may not explore it.

The plan for this course is the following:

Lectures 2, 3, 4. We aim to understand generalization as a property of the learning algorithm. In Lecture 2, we will introduce the notion of stability (Bousquet and Elisseeff, 2002) to do that and see how it implies generalization bounds. In Lecture 3, we will further examine the stability properties of SGD based on the work of Hardt et al. (2016) but also we will discuss about failures of uniform convergence, indicated by the work of Zhang et al. (2021). In Lecture 4, we will talk about failures of uniform convergence based on the work of Nagarajan and Kolter (2019) but also discuss domain adaptation (where the training and test distributions do not coincide) (Ben-David et al., 2006).

Lectures 5, 6, 7. Stability is quite rich of a notion. It can capture various fundamental notions of learning such as private learning. This will be the content of Lectures 5, 6 and 7. In terms of techniques, standard uniform convergence does not suffice to explain private learning. In Lecture 5, we will revisit one of the seminal papers studying private learning (Kasiviswanathan et al., 2011). In Lectures 5 and 6, we will deal with a more modern (and challenging) result: private learning is equivalent to online learning for binary classification (Alon et al., 2019; Bun et al., 2020).

Lecture 8. In this lecture, we will see two other different forms of stability: one-way perfect generalization and replicability (Impagliazzo et al., 2022; Bun et al., 2023; Kalavasis et al., 2023). We will see how these two notions are closely related to differential privacy. This indicates that many well-known stability notions share (perhaps surprisingly) many conceptual and technical similarities.

Lecture 9. Memorization is a big mystery in the ML community. In this lecture, we will try to understand how memorization is related to stability and generalization. We will present two important works (Feldman, 2020; Brown et al., 2021).

Lecture 10 In this lecture, we will see an important setting where VC theory fails. The topic of this lecture is a fundamental work of Bousquet et al. (2021). The goal is to answer the following important question: how quickly can a given class of concepts be learned from examples? The standard measure the performance of a supervised machine learning algorithm is its learning curve, i.e., the decay of the error rate as a function of the number of training examples. We will see how stability can be used to give surprising results.

Lectures 11, 12. In Lecture 11, we will see how stability and online learning are crucial to problems beyond prediction. We will focus on the classical problem of language identification and the recent reformulation of language generation (Gold, 1967; Kleinberg and Mullainathan, 2024). In Lecture 12, we will talk about consistent generation with breadth (Kalavasis et al., 2024b,a).

Lectures 13, 14. In this final pair of Lectures, there will be presentations for recent interesting papers, related to the course's topics.

References

Alon, N., Livni, R., Malliaris, M., and Moran, S. (2019). Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2006). Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526.

Bousquet, O., Hanneke, S., Moran, S., van Handel, R., and Yehudayoff, A. (2021). A theory of universal learning. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 532–541, New York, NY, USA. Association for Computing Machinery.

Bousquet, O., Hanneke, S., Moran, S., and Zhivotovskiy, N. (2020). Proper learning, helly number, and an optimal svm bound. In *Conference on Learning Theory*, pages 582–609. PMLR.

Brown, G., Bun, M., Feldman, V., Smith, A., and Talwar, K. (2021). When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*, pages 123–132.

Bun, M., Gaboardi, M., Hopkins, M., Impagliazzo, R., Lei, R., Pitassi, T., Sivakumar, S., and Sorrell, J. (2023). Stability is stable: Connections between replicability, privacy, and adaptive generalization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 520–527, New York, NY, USA. Association for Computing Machinery.

Bun, M., Livni, R., and Moran, S. (2020). An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 389–402. IEEE.

Feldman, V. (2020). Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959.

Gold, E. M. (1967). Language identification in the limit. *Information and control*, 10(5):447–474.

Hanneke, S. (2016). The optimal sample complexity of pac learning. *Journal of Machine Learning Research*, 17(38):1–15.

Hardt, M., Recht, B., and Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR.

Hopkins, M., Kane, D. M., Lovett, S., and Mahajan, G. (2022). Realizable learning is all you need. In *Conference on Learning Theory*, pages 3015–3069. PMLR.

Impagliazzo, R., Lei, R., Pitassi, T., and Sorrell, J. (2022). Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 818–831.

Kalavasis, A., Karbasi, A., Moran, S., and Velegkas, G. (2023). Statistical indistinguishability of learning algorithms. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15586–15622. PMLR.

Kalavasis, A., Mehrotra, A., and Velegkas, G. (2024a). Characterizations of language generation with breadth. *arXiv preprint arXiv:2412.18530*.

Kalavasis, A., Mehrotra, A., and Velegkas, G. (2024b). On the limits of language generation: Trade-offs between hallucination and mode collapse. *arXiv preprint arXiv:2411.09642*.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kleinberg, J. and Mullainathan, S. (2024). Language generation in the limit. In *Advances in Neural Information Processing Systems*, volume 37.

Nagarajan, V. and Kolter, J. Z. (2019). Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, V. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.