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

Lecture 2: Generalization and Stability in Machine Learning

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

In Lecture 1, we revisited the most prominent method to estimate the accuracy of a learning algorithm which is based on the theory of *uniform convergence* of empirical quantities to their mean (see e.g., Shalev-Shwartz and Ben-David (2014)). This theory provides ways to estimate the risk (or generalization error) of a learning system based on an empirical measurement of its accuracy and a measure of its complexity, such as the Vapnik-Chervonenkis (VC) dimension.

On the negative side, this theory cannot explain the success of various algorithms such as the k-NN algorithm: VC theory focuses on the size of the search space (the VC dimension of all concepts induced by k-NN is infinite) while other approaches focus on *how* the algorithm searches that space.

> **Insight 1: VC Theory does not Suffice for Generalization**
>
> Unlike VC-based approaches where the only property of the algorithm that matters is the size of the space to be searched, we are going to focus on *how the machine learning algorithm searches the space*. As we did in the end of Lecture 1, you should think of SGD and neural networks.

In Lecture 1, we used Uniform Convergence to show that the empirical loss of any hypothesis in the class is close to its population loss. Recall that uniform convergence makes no reference to the algorithm.

The general goal of Lecture 2 is to provide a collection of results of the flavor "Stability Implies Generalization"; here, stability is related to some *algorithmic property* and generalization means that the empirical loss will be close to the population one). We will discuss about the following four results-tools:

1. Uniform Stability Implies Generalization
2. Sample Compression Implies Generalization
3. Bounded Mutual Information Implies Generalization
4. Bounded Conditional Mutual Information Implies Generalization

The important difference compared to Lecture 1 is that now we will study generalization via *properties of the learning algorithm* and not of the *function class, i.e., the search space.*

> **Insight 2: Showing Generalization Bounds**
>
> If we fix some (binary) classifier we can show, using standard concentration bounds, that its performance on a sample is close to its performance on the underlying population. However, when we train an ML algorithm A using a dataset S to output a classifier h we cannot just use the fact that it has small loss on S to claim that its loss on the population is small because h depends on S. The above tools will allow us to get such results for algorithms A that satisfy some property among (1)-(4).

1 Local Rules and Generalization

The first stability-based results were obtained by Devroye, Rogers and Wagner in the seventies (see Rogers and Wagner (1978); Devroye and Wagner (1979a)). Rogers and Wagner first showed that the variance of the leave-one-out error can be upper bounded by what Kearns and Ron (1997) later called hypothesis stability. This quantity measures how much the function learned by the algorithm will change when one point in the training set is removed. The main distinctive feature of their approach is that, unlike VC-theory based approaches where the only property of the algorithm that matters is the size of the space to be searched, it focuses on how the algorithm searches the space (e.g., by making use of that the learning algorithm is 'local'). A standard example is nearest-neighbor-based algorithms: Let K be the set of classifiers implemented by the 1-NN algorithm (for details see Devroye et al. (2013)). Then $\mathrm{VC}(K) = \infty$. However, guarantees about the generalization error of k-NN are known, e.g., see Rogers and Wagner (1978) and Devroye and Wagner (1982) for a survey.

For instance, consider the multiclass classification setting, where (X, Y) is drawn from some distribution and $Y \in \{1, ..., K\}$. For a new point X', the estimate $\widehat{Y}$ of its label based on the k-NN algorithm is based on the most frequent label among the k nearest points of X'. More generally, if (X^j, Y^j) represents the j-th closest point to X', one can think of k-local rules as mappings

$$\widehat{Y} = g((X^1, Y^1), ..., (X^k, Y^k))$$

for some function g. For simplicity, we ignore ties in the above discussion. The loss is $R = \mathbf{Pr}[Y \neq \widehat{Y} | S_n]$, where S_n is the training set of size n. A natural estimate of R is the delete estimate $\widehat{R}_n = \frac{1}{n}\sum_{i=1}^{n} 1\{y_i \neq \widehat{y}_i\}$, where $\widehat{y}_i$ is the estimate of y_i from X_i and S_n with (X_i, y_i) deleted (here $k \leq n - 1$).

Theorem 1: Rogers and Wagner (1978)

For all distributions (X, Y) and any k-local rule,

$$\mathbf{E}(R - \widehat{R}_n)^2 \leq \frac{2k + 1/4}{n} + \frac{2k\sqrt{2k + 1/4}}{n^{3/2}} + \frac{k^2}{n^2}\,.$$

2 What is Algorithmic Stability in Machine Learning?

Inspired by the above semilar results, Bousquet and Elisseeff (2002) introduced a general stability framework for ML algorithms. In this section, we define notions of stability for learning algorithms from Bousquet and Elisseeff (2002) and show how to use these notions to derive generalization error bounds based on the empirical error and the leave-one-out error.

Definition 1: Learning Algorithm

A learning algorithm A is a (potentially randomized) mapping from $(\mathcal{X} \times \mathcal{Y})^n$ to $\mathcal{Y}^{\mathcal{X}}$ which maps a training set S of size n to a function $A(S)$ from $\mathcal{X}$ to $\mathcal{Y}$.

In general, learning algorithms are randomized, so they map datasets to a distribution over functions ($A(S)$ is a distribution over $\mathcal{Y}^{\mathcal{X}}$) but for simplicity we will assume that A is deterministic and that all functions are measurable and all sets are countable. For deterministic machine learning algorithms, the only source of randomness comes from the training data.

We will denote datasets either with S or Z. Recall the following notation from Lecture 1:

Definition 2: Empirical and Population Loss

Given algorithm A and training set S, define the population loss on a fresh sample z as

$$R(A, S) = \mathop{\mathbf{E}}_{z} \ell(A(S), z)\,.$$

Also, let the empirical loss on $S = \{z_1 = (x_1, y_1), ..., z_n = (x_n, y_n)\}$ be

$$R_{emp}(A, S) = \frac{1}{n} \sum_{i=1}^{n} \ell(A(S), z_i)\,.$$

Note that the ERM algorithm on some class $\mathcal{H}$ is a mapping from datasets S to the function

$$\text{argmin}_{h \in \mathcal{H}} \sum_{z \in S} \ell(h, z)\,.$$

We may also use the notation $\ell(A(S), \cdot)$ to denote a vector of loss values of the predictor $A(S)$ over all possible examples.

Understanding the generalization properties of an algorithm A reduces to the following problem:

Problem 1: High-Probability Generalization Bounds for specific algorithm A

For any ϵ, upper bound the tail probability

$$\Pr_S \left[|R_{emp}(A, S) - R(A, S)| > \epsilon \right]\,.$$

We remark that given that an algorithm A generalizes in the above sense, does not mean that it is a good learning algorithm. It simply guarantees that with high probability, the performance of the algorithm in the training set will be reflected to the population loss, i.e., A outputs a model that generalizes to the underlying distribution, rather than overfitting its training data.

Let us first recall how this compares to uniform convergence and why then ERM is a natural approach.

Insight 3: Uniform Convergence

Contrast the above with

$$\Pr_S \left[\sup_{f \in \mathcal{H}} \left| \frac{1}{n} \sum_{x \in S} \ell(f, x) - \mathop{\mathbf{E}}_{z} \ell(f, z) \right| > \epsilon \right]\,.$$

In Lecture 1, we saw that when the above holds, then ERM is the way to go. In contrast, in this Lecture, we do not want to have a bound that holds uniformly over the whole space of possible functions $\mathcal{H}$ since we are interested in algorithms that *may not explore it*.

Given the above insight, we are looking to deal with algorithms that will not explore everything in a very complicated function space. An intuitive idea for neural network training is that SGD is somehow conservative in its updates exploring a small subset of potential neural networks. The notion of being 'conservative' will be captured by *algorithmic stability*.

There are many ways to define and quantify the stability of a learning algorithm. The natural way of making such a definition is to start from the goal: we want to get bounds on the generalization error of specific learning algorithm and we want these bounds to be tight when the algorithm satisfies the stability criterion.

As one may expect, the more restrictive a stability criterion is, the tighter the corresponding bound will be.

In the learning model we consider, the randomness comes from the sampling of the training set. We will thus consider stability with respect to changes in the training set. To model stability, we will consider only restricted changes such as the removal or the replacement of one single example in the training set.

Definition 3: Uniform Stability

An algorithm A has **uniform stability** β with respect to ℓ if

$$\forall S \in (\mathcal{X} \times \mathcal{Y})^n,\ \forall i \in [n],\ \|\ell(A(S), \cdot) - \ell(A(S_{-i}), \cdot)\|_\infty \leq \beta\,,$$

where $S_{-i} = \{z_1, ..., z_{i-1}, z_{i+1}, ..., z_n\}$ corresponds to the training set S with the i-th example removed.

Note that the above is extremely strong: it is worst-case over all possible datasets that the algorithm could encounter and worst-case over examples from the domain. Moreover, stability is tightly related to a loss function ℓ.

Definition 4: Hypothesis Stability

An algorithm A has **hypothesis stability** β with respect to ℓ if

$$\forall i \in [n],\ \mathop{\mathbf{E}}_{S,z} |\ell(A(S), z) - \ell(A(S_{-i}), z)| \leq \beta\,,$$

where $S_{-i} = \{z_1, ..., z_{i-1}, z_{i+1}, ..., z_n\}$ corresponds to the training set S with the i-th example removed.

Definition 5

An algorithm A has **pointwise hypothesis stability** β with respect to ℓ if

$$\forall i \in [n],\ \mathop{\mathbf{E}}_{S} |\ell(A(S), z_i) - \ell(A(S_{-i}, z_i)| \leq \beta\,,$$

where $S_{-i} = \{z_1, ..., z_{i-1}, z_{i+1}, ..., z_n\}$ corresponds to the training set S with the i-th example removed.

We will only deal with uniformly stable algorithms but similar generalization bounds can be obtained under weaker assumptions.

Insight 4: The practical perspective

Practical methods have been developed to deal with instability of learning algorithms. In particular, Breiman introduced the *bagging* technique which is presented as a method to combine single classifiers in such a way that the variance of the overall combination is decreased. Recently, it was shown that bagging is an optimal PAC learner (Larsen, 2023).

3 Generalization of Uniformly Stable Algorithms

A stable algorithm has the property that removing one element in its learning set does not change much of its outcome. As a consequence, the difference between empirical and generalization error, if thought as a random variable, should have a small variance. If its expectation is small, stable algorithms should then be good candidates for their empirical error to be close to their generalization error. This assertion is formulated in the following theorem:

Theorem 2: Uniform Stability ⇒ Generalization

Let A be an algorithm with uniform stability β and assume that $0 \leq \sup_S \sup_z \ell(A(S), z) \leq M$. For any sample size $n \geq 1$ and confidence $\delta \in (0, 1)$, the following generalization bound holds with probability $1-\delta$ over the training set S:

$$R(A,S) \leq R_{emp}(A,S) + 2\beta + (4n\beta + M)\sqrt{\log(1/\delta)/n}\,.$$

This theorem gives tight bounds when the stability scales as $1/n$ ($1/n$ is the rate one should expect for the stability parameter $\beta = \{\beta_n\}$). We shortly mention that for classification tasks, the stability parameter is binary, i.e., $\beta = 0$ or $\beta = 1$. To deal with it, we usually consider algorithms with real-valued outputs. Then, the label predicted by such an algorithm is the sign of its real-valued output.

3.1 Proving that Uniform Stability Implies Generalization

We are now ready to prove the following result. It will be assumed that the algorithm A is symmetric with respect to S, i.e., it does not depend on the order of the elements in the training set.

Theorem 3: Uniform Stability Implies Generalization

Let A be a symmetric algorithm with uniform stability β and assume that $0 \leq \sup_S \sup_z \ell(A(S),z) \leq M$. For any sample size $n \geq 1$ and confidence $\delta \in (0,1)$, the following generalization bound holds with probability $1-\delta$ over the training set S:

$$R(A,S) \leq R_{emp}(A,S) + 2\beta + (4n\beta + M)\sqrt{\log(1/\delta)/n}\,.$$

Proof. Our goal is to apply McDiarmid's Inequality.
Fix $z_1, ..., z_n, z_i' \in Z$. Let $S = \{z_1, ..., z_n\}$ and define S^i to be obtained by S by removing z_i and adding z_i'.
Let $f : Z^n \to \mathbb{R}$ be any measurable function for which there exist constants c_i such that

$$\sup_{S \in Z^n, z_i' \in Z} |f(S) - f(S^i)| \leq c_i\,.$$

Then, it holds

$$\Pr_S[|f(S) - \mathbf{E}_S f| \geq \epsilon] \leq \exp\left(-2\epsilon^2 / \sum_i c_i^2\right).$$

Exercise 1

Prove McDiarmid's Inequality (for the proof, see Van Handel (2014).).

As an application of McDiarmid's inequality, let us find c_i for our functions of interest:

$$|R - R_{-i}| \leq \mathbf{E}_z |\ell(A(S),z) - \ell(A(S_{-i},z)| \leq \beta\,,$$

and

$$|R_{emp} - R_{emp,-i}| \leq \frac{\ell(A(S),z_i)}{n} + \frac{1}{n-1}\sum_{j\neq i} |\ell(A(S),z_j) - \ell(A(S_{-i},z_j)| \leq \beta + M/n\,.$$

This means that

$$|R - R^i| \leq |R - R_{-i}| + |R_{-i} - R^i| \leq 2\beta\,.$$

and

$$|R_{emp} - R_{emp}^i| \leq \frac{1}{n}\sum_{j\neq i} |\ell(A(S),z_j) - \ell(A(S_{-i},z_j)| + \frac{1}{n}\sum_{j\neq i} |\ell(A(S_{-i}),z_j) - \ell(A(S^i,z_j)| + \frac{1}{n}|\ell(A(S),z_i) - \ell(A(S^i),z_i')|.$$

The above is at most $2\beta + M/n$. Hence, let us set $f = R - R_{emp}$, which satisfies McDiarmid's inequality with $c_i = 4\beta + M/n$. Since by uniform stability and symmetry

$$\mathop{\mathbf{E}}_{S} R(A,S) - R_{emp}(A,S) \leq 2\beta\,,$$

we get the desired tail bound. The above inequality is shown as follows:

$$\mathop{\mathbf{E}}_{S} R(A,S) - R_{emp}(A,S) = \mathop{\mathbf{E}}_{S,z_i'} \ell(A(S),z_i') - \mathop{\mathbf{E}}_{S} R_{emp}(A,S)$$

and

$$\mathop{\mathbf{E}}_{S} R_{emp}(A,S) = \mathop{\mathbf{E}}_{S,z_i'} \frac{1}{n}\sum_i \ell(A(S),z_i)\,.$$

By the i.i.d. assumption and symmetry, let us rename any z_i as z_i' (the random variable $\ell(A(\{z_1,...,z_i,...,z_n\}),z_i)$ will have the same law as $\ell(A\{z_1,...,z_i',...,z_n\},z_i')$ and get for all $i \in [n]$:

$$\mathop{\mathbf{E}}_{S} R_{emp}(A,S) = \mathop{\mathbf{E}}_{S,z_i'} \frac{1}{n}\sum_i \ell(A(S),z_i) = \mathop{\mathbf{E}}_{S,z_i'} \ell(A(S^i),z_i')\,.$$

Now using uniform stability, we get the result. □

3.2 Applications of Stability

Local Algorithms A k-Local Rule is a classification algorithm that determines the label of an instance x based on the k closest instances in the training set. The simplest example of such a rule is the k-Nearest Neighbors (k-NN) algorithm which computes the label by a majority vote among the labels of the k nearest instances in the training set.

With respect to the $\{0,1\}$-loss function, the k-NN classifier has hypothesis stability

$$\beta \leq \frac{4}{n}\sqrt{k/2\pi}\,.$$

This was proven in Devroye and Wagner (1979a).

Regularization Uniform stability may appear as a strict condition. However, many existing learning methods exhibit a uniform stability which is controlled by the regularization parameter and can thus be very small (for details see Bousquet and Elisseeff (2002)).

3.3 Extensions of Algorithmic Stability

The main object of study in Lecture 2 is uniform stability, a property of the learning algorithm that captures how sensitive the (loss of the) algorithm is when changing a single element of its training set.

> **Definition 6: Uniform Stability (Revisited)**
>
> Algorithm A is said to have uniform stability β with respect to ℓ if for any pair of datasets $S, S' \in \mathcal{Z}^n$ that differ in a single element and every $z \in \mathcal{Z}$, $|\ell(A(S),z) - \ell(A(S'),z)| \leq \beta$.

Denote the generalization gap of A trained on S as

$$\Delta_S(A) = R(A,S) - R_{emp}(A,S)$$

Recall that R is the population loss while R_{emp} is the loss on the training set. For a uniformly stable algorithm, we have the following properties for any distribution $\mathcal{D}$ over Z and $\delta > 0$:

1. $|\,\mathbf{E}_{S\sim\mathcal{D}^n}[\Delta_S(A)]| \leq \beta$;
2. $\mathbf{E}_{S\sim\mathcal{D}^n}[(\Delta_S(A))^2] \leq \frac{1}{2n} + 6\beta$;
3. $\mathbf{Pr}_{S\sim\mathcal{D}^n}[\Delta_S(A) \geq (4\beta + 1/n)\sqrt{n\log(1/\delta)/2} + 2\beta] \leq \delta$.

Bound (3) does not lead to meaningful generalization bounds in many common settings where $\beta > 1/\sqrt{n}$. Feldman and Vondrak (2018) improve on (2) and (3) (using tools from differential privacy) by showing that

1. $\mathbf{E}_{S\sim\mathcal{D}^n}[(\Delta_S(A))^2] \leq \frac{2}{n} + 16\beta^2$;
2. $\mathbf{Pr}_{S\sim\mathcal{D}^n}[\Delta_S(A) \geq (2\beta + 1/n)\sqrt{\log(8/\delta)}] \leq \delta$.

Finally, Feldman and Vondrak (2019) show that

$$\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^n}\left[\Delta_S(A) \geq c(\beta\log n\log(1/\delta) + \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}})\right] \leq \delta.$$

We also note that differential privacy and VC dimension can be incorporated into the above stability framework (see Dagan and Feldman (2019)).

4 General Theories for Generalization

4.1 Generalization via Sample Compression

There are other ways to show that a specific algorithm A generalizes. Algorithms whose output essentially only depends on a few of the input data points (e.g., SVM or the median), as formalized by compression schemes (Littlestone and Warmuth, 1986), can be shown to generalize.

Definition 7: Sample Compression

A sample compression scheme takes a long list of samples and compresses it to a short sub-list of samples in a way that allows to invert the compression. Formally, a sample compression scheme for C with kernel size k and side information I, where I is a finite set, consists of two maps κ, ρ for which the following hold:

1. The compression map
$$\kappa : L_{\mathcal{H}}(\infty) \to L_{\mathcal{H}}(k) \times I$$
taking the training set (X, y) (labeled by a function in $\mathcal{H}$) and compressing it to $((Z, z), i)$ where $Z \subset X$ and $z = y|_Z$.
2. The reconstruction map
$$\rho : L_{\mathcal{H}}(k) \times I \to \{0,1\}^{\mathcal{X}}$$
which has the property that for all $(X, y) \in L_{\mathcal{H}}(\infty)$ it holds $\rho(\kappa(X, y))|_X = y$.

The size of the compression scheme is $k + \log|I|$.

The side information I can be thought of as list decoding: the map ρ has a short list of possible reconstructions of a given dataset (X, y), and the information $i \in I$ indicates which element in the list is the correct one.

Littlestone and Warmuth showed that every compression scheme yields a natural learning procedure: Given a labeled sample $(S, h(S))$, the learner compresses it to $\kappa(S, h(S))$ and outputs the hypothesis $h = \rho(\kappa(S, h(S))$. They proved that this is indeed a PAC learner.

Theorem 4: Sample Compression Implies Generalization (Littlestone and Warmuth, 1986)

Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and let κ, ρ be a sample compression scheme for $\mathcal{H}$ of size k. Let $n \geq (k \log(1/\epsilon) + \log(1/\delta))/\epsilon$. Then, the learning map $(S, h(S)) \to \rho(\kappa(S, h(S)))$ is PAC learning $\mathcal{H}$ with n samples, generalization error ϵ and success probability $1 - \delta$.

The proof idea is as follows: given a training set $X = \{x_1, ..., x_n\}$, fix a subset T of size k and information $i \in I$: this choice induces a hypothesis $h_{T,i}(x) = \rho((x_S, c|_S), i)$. The random function $h_{T,i}$ is independent of the remaining dataset X_{-S}. Since the size of the scheme is k, the number of pairs (T, i) is of order n^k and the probability that $h_{T,i}$ that has ϵ-generalization error agrees with X_{-S} is of order $(1-\epsilon)^{n-k}$. Hence, by union bound, the probability that the scheme outputs a hypothesis with ϵ-loss (since the scheme should output a mapping such that $h|_X = c|_X$) is $n^k(1-\epsilon)^{n-k} \leq \delta$ and the sample complexity follows.

The other direction is more interesting. Moran and Yehudayoff (2016) show the following surprising result: VC classes have sample compression schemes of finite size (independent of the size of the training set).

Theorem 5: PAC Learning Implies Sample Compression (Moran and Yehudayoff, 2016)

If $\mathcal{H}$ has VC dimension d, then $\mathcal{H}$ has a sample compression scheme of size 2^d.

Exercise 2

Read the proof of Theorem 5 from Moran and Yehudayoff (2016).

It remains open to improve this bound (see also Chase et al. (2024)). Before moving on, we would like to mention that compression and generalization have close connections in many areas of research, such as memory and cognition, e.g., recall the chess experiment that we discussed in the class Frey and Adesman (1976). The study aimed to investigate how expertise in chess affects the ability to recall chessboard configurations. The methodology of the experiment was as follows:

- *Participants:* Chess players of varying skill levels, categorized as novices or experts.
- *Stimuli:* Two types of chessboard configurations were used:
  1. *Meaningful positions:* Derived from actual games (plausible configurations based on chess rules).
  2. *Random positions:* Created by placing pieces randomly on the board, breaking typical game patterns.
- *Procedure:* Participants were briefly shown a chess position for a limited exposure time. Afterward, they were asked to recreate the position from memory on a blank chessboard.

Experts significantly outperformed novices in recalling meaningful positions. For random positions, experts and novices showed similar levels of recall performance, indicating that the expertise advantage relies on the meaningfulness of the stimuli. Experts' superior performance was attributed to their ability to *compress* information, grouping related pieces into familiar patterns. In random positions, the absence of familiar patterns reduced their advantage.

4.2 Generalization via Mutual Information

The mutual information (MI) of two random variables is a measure of the *mutual dependence* between the two variables. It quantifies the amount of information (in units such as bits or nats) obtained about one random variable by observing the other random variable.

Definition 8: Mutual Information

For two random variables X, Y with joint probability distribution $P_{X,Y}$, let us denote by P_X and P_Y their marginals. The mutual information is defined as

$$I(X;Y) = \mathrm{KL}(P_{X,Y} \parallel P_X \otimes P_Y)\,.$$

Note that $I(X;Y)$ is equal to zero precisely when the joint distribution coincides with the product of the marginals. It is closely related to the entropy: $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ and is symmetric: $I(X;Y) = I(Y;X)$.

We will use the following notation:

1. $\mathcal{Z}$ will contain labeled examples $z = (x, y)$.
2. $\mathcal{H}$ is the space of functions.
3. $Z \sim \mathcal{D}^n$ is a dataset consisting of n i.i.d. labeled examples drawn from $\mathcal{D}$.

A recent line of work has studied generalization using mutual information and related quantities (Russo and Zou, 2016; Raginsky et al., 2016, etc.). Specifically, for a (possibly randomized) algorithm $A : \mathcal{Z}^n \to \mathcal{H}$ and a dataset $Z \sim \mathcal{D}^n$, we consider the quantity $I(A(Z); Z)$. This measures how much information the output of A contains about its input.

The following is shown in Russo and Zou (2016); Xu and Raginsky (2017).

Theorem 6: Bounded Mutual Information Implies Generalization (Xu and Raginsky, 2017)

If $\ell : \mathcal{H} \times \mathcal{Z} \to [0, 1]$, $A : \mathcal{Z}^n \to \mathcal{H}$ and $Z \sim \mathcal{D}^n$, then

$$|\mathbb{E}\ell(A(Z), Z) - \ell(A(Z), \mathcal{D})| \leq \sqrt{\frac{2}{n} \cdot I(A(Z); Z)}\,.$$

Note that this bound controls the expected generalization error. There exist algorithms that can make the RHS trivially zero (e.g., use the constant function independently of the dataset). This algorithm however has very bad generalization error. The above inequality does not guarantee anything about the quality of the algorithm, it only allows to argue that the empirical and the population losses will be close.

The key idea behind this result is the following Lemma.

Lemma 1

Consider random variables (X, Y) with joint distribution $P_{X,Y}$. Consider an independent copy $\bar{X}$ of X and an independent copy $\bar{Y}$ of Y such that $P_{\bar{X},\bar{Y}} = P_X \otimes P_Y$. If $f(\bar{X}, \bar{Y})$ is σ-subgaussian with respect to $P_{\bar{X},\bar{Y}}$, then

$$|\,\mathbf{E}[f(X, Y)] - \mathbf{E}[f(\bar{X}, \bar{Y})]| \leq \sqrt{2\sigma^2 I(X;Y)}\,.$$

Exercise 3

Prove Lemma 1.

To use this result, we set $X = S$ (a drawn dataset) and Y to be a classifier drawn from the algorithm $P_{h|S}$. Note that $P_{X,Y} = \mathcal{D}^n \times P_{h|S}$. Now we create the independent copies: $\bar{X}$ will be a new dataset and $\bar{Y}$ will be an independent copy of Y (which is independent of $\bar{X}$ and is drawn from the marginal). Hence the expected generalization error can be written as

$$\mathbf{E}[f(\bar{X}, \bar{Y})] - \mathbf{E}[f(X, Y)]$$

where $f(s, h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, z_i)$.

Bounds on mutual information can be obtained from differential privacy or from bounds on the entropy of the output of A. Specifically, if A is ε-differentially private, then $I(A(Z); Z) \leq \frac{1}{2}\varepsilon^2 n$ (McGregor et al., 2010; Bun and Steinke, 2016).

Unfortunately, mutual information can easily be infinite even in settings where generalization is easy to prove. Bassily et al. (2018); Nachum et al. (2018) showed that *any* proper and consistent learner for threshold functions A must have $I(A(Z); Z) \geq \Omega\left(\frac{\log\log|\mathcal{Z}|}{n^2}\right)$ when $Z \sim \mathcal{D}^n$ for some worst-case distribution $\mathcal{D}$. In contrast, the VC dimension of threshold functions is 1, which implies strong uniform convergence bounds even for infinite domains Z.

4.3 Generalization via CMI

The issue with the mutual information approach is that even a single data point has infinite information content if the distribution is *continuous*. This makes it difficult to bound the mutual information – an algorithm revealing a single data point is not an issue for generalization, but it is for mutual information.

Steinke and Zakynthinou (2020) introduce the conditional mutual information (CMI) framework for reasoning about the generalization properties of machine learning algorithms. CMI is a quantitative property of an algorithm A. (Note that it does *not* depend on the loss function of interest; compare this to uniform stability.)

The definition of CMI will be inspired by the symmetrization technique, which we used in Lecture 1 in order to show that finite VC dimension implies uniform convergence. Intuitively, CMI measures how well we can "recognize" the input (i.e., training data) given the output (i.e., trained model) of the algorithm. Recognizing the input is formalized by considering a set consisting of $2n$ data points – namely the n input data points mixed with n "ghost" data points (as we did in the symmetrization trick) – and measuring how well it is possible to distinguish the true inputs from their ghosts.

The supersample is randomly partitioned into the input and the ghost samples. We then measure how much information the output reveals about this partition using mutual information, where we take the supersample to be known (i.e., we condition on the supersample and the unknown information is how it is partitioned). We now state the formal definition of CMI:

Definition 9: Conditional Mutual Information

Let $A : \mathcal{Z}^n \to \mathcal{H}$ be a randomized algorithm. Let $\mathcal{D}$ be a probability distribution on $\mathcal{Z}$ and let $Z' \in \mathcal{Z}^{n \times 2}$ consist of $2n$ samples drawn independently from $\mathcal{D}$. Let $S \in \{0,1\}^n$ be uniformly random and independent from Z' and the randomness of A.
Define $\mathcal{Z}'_S \in \mathcal{Z}^n$ by $(\mathcal{Z}'_S)_i = Z'_{i,S_i+1}$ for all $i \in [n]$ – that is, Z'_S is the subset of Z' indexed by S.
The *conditional mutual information (CMI) of A with respect to $\mathcal{D}$* is

$$\mathsf{CMI}_{\mathcal{D}}(A) := I(A(Z'_S); S | Z').$$

The *(distribution-free) conditional mutual information (CMI) of A* is

$$\mathsf{CMI}(A) := \sup_{z' \in \mathcal{Z}^{n \times 2}} I(A(z'_S); S).$$

The above quantity captures the following: Given the supersample Z', how much information does the output of A trained in Z'_S reveal about S?

The following result shows that CMI controls the gap between the empirical and the generalization error.

Theorem 7: Bounded CMI Implies Generalization (Steinke and Zakynthinou, 2020)

Let $A : \mathcal{Z}^n \to \mathcal{H}$ and $\ell : \mathcal{H} \times \mathcal{Z} \to [0,1]$. Let $\mathcal{D}$ be a distribution on $\mathcal{Z}$ and define $\ell(w, \mathcal{D}) = \mathbf{E}_{Z \sim \mathcal{D}} \ell(w, Z)$ and $\ell(w, z) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, z_i)$ for all $w \in \mathcal{H}$ and $z \in \mathcal{Z}^n$. Then

$$\left| \mathop{\mathbf{E}}_{Z \sim \mathcal{D}^n, A} \ell(A(Z), Z) - \ell(A(Z), \mathcal{D}) \right| \le \sqrt{\frac{2}{n} \cdot \mathsf{CMI}_{\mathcal{D}}\left(A\right)}, \tag{1}$$

A definition is useful if it can be used to capture various existing approaches; this is exactly the case for CMI, as we will review below.

Compression Schemes If an algorithm $A : \mathcal{Z}^n \to \mathcal{H}$ has a compression scheme of size k, then $\mathsf{CMI}\left(A\right) \le O(k \cdot \log n)$. Intuitively, this is in agreement with the fact that an algorithm revealing k of the input points and nothing about the rest would have a CMI of k bits.

The argument goes intuitively as follows: Let Z'_S be the random training set chosen by S. Since A is based on sample compression, $I(A(Z'_S); S) = I(\rho(\kappa(Z'_S)); S)$ (let us ignore the information set for simplicity). By the data-processing inequality, this information is at most $I(\rho(\kappa(Z'_S)); S) \le I((Z'_S)_{K(Z'_S)}; S)$, where $K(Z'_S)$ is the set of indices chosen by the compression algorithm κ on input Z'_S. This means that the information revealed is of order k bits. Since Z' is fixed and S is random, this information is at most $k \log(2n)$, since the possible subsets are at most $\binom{2n}{k}$.

Uniform Convergence & VC Dimension Bounded VC dimension is a necessary and sufficient condition for uniform convergence and is hence a sufficient condition for generalization (recall Lecture 1). Note that CMI is a property which depends on the algorithm, whereas the VC dimension is a property of the output space; this appears to cause an incompatibility between the two methods. However, one can show the following: for any hypothesis class of bounded VC dimension, there always *exists* some ERM algorithm $A : \mathcal{Z}^n \to \mathcal{H}$ with bounded CMI:

Theorem 8

Let $\mathcal{Z} = \mathcal{X} \times \{0,1\}$ and let $\mathcal{H} = \{h : \mathcal{X} \to \{0,1\}\}$ be a hypothesis class with VC dimension d. Then, there exists an empirical risk minimizer $A : \mathcal{Z}^n \to \mathcal{H}$ for the 0-1 loss such that $\mathsf{CMI}\left(A\right) \le d \log n + 2$.

Note that it is not true that *every* empirical risk minimizer for a class of bounded VC dimension has bounded CMI.

Distributional Stability & Differential Privacy Finally, we show that distributional stability implies CMI bounds. Differential privacy is the most well-known form of distributional stability and its generalization properties are well-established (Dwork et al., 2015; Bassily et al., 2016; Jung et al., 2019).

Theorem 9

Let $A : \mathcal{Z}^n \to \mathcal{H}$ be a randomized algorithm. The following conditions imply that $\mathsf{CMI}\left(A\right) \le \varepsilon n$.

(i) A is $\sqrt{2\varepsilon}$-differentially private (Dwork et al., 2006).

(ii) A is ε-TV stable (Bassily et al., 2016).[a]

[a] A randomized algorithm is ϵ-TV stable if for any pair of samples that differ on one element, $\mathrm{TV}(A(S), A(S')) \le \epsilon$.

We will formally define and study the above notions of stability later in the course.

Loss Stability/Uniform Stability A long line of work Rogers and Wagner (1978); Devroye and Wagner (1979b); Bousquet and Elisseeff (2002) has proven generalization bounds by showing that various algorithms have the property that their loss changes very little if a single input datum is replaced or removed (recall the first part of Lecture 2). Uniform stability (one of the strongest and most well-studied variants of loss stability) has the advantage that it readily yields high probability generalization bounds. However, one limitation of the loss stability approach is that the loss function is an integral part of the definition, whereas CMI do not depend on the loss function. It is of course natural to ask whether loss stability and CMI can be unified in some way. Steinke and Zakynthinou (2020) propose a variant of CMI (Evaluated CMI or eCMI) that takes the loss function into account and allows us to translate between the notions.

Here we define

$$\mathrm{eCMI}_{\mathcal{D}}(\ell(A)) = I(\ell(A(Z'_S), Z'); S|Z') .$$

By the data-processing inequality, $\mathrm{eCMI}_{\mathcal{D}}(\ell(A)) \leq \mathrm{CMI}_{\mathcal{D}}(A)$. It turns out given a uniformly stable algorithm A under loss ℓ, one can bound the evaluated CMI of an algorithm $\widetilde{A}$, which corresponds to A with some 'Gaussian perturbations' to its loss.

References

Bassily, R., Moran, S., Nachum, I., Shafer, J., and Yehudayoff, A. (2018). Learners that use little information. In Janoos, F., Mohri, M., and Sridharan, K., editors, *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 25–55. PMLR.

Bassily, R., Nissim, K., Smith, A., Steinke, T., Stemmer, U., and Ullman, J. (2016). Algorithmic stability for adaptive data analysis. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 1046–1059, New York, NY, USA. ACM.

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

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

Chase, Z., Chornomaz, B., Hanneke, S., Moran, S., and Yehudayoff, A. (2024). Dual vc dimension obstructs sample compression by embeddings. *arXiv preprint arXiv:2405.17120*.

Dagan, Y. and Feldman, V. (2019). Pac learning with stable and private predictions. *arXiv preprint arXiv:1911.10541*.

Devroye, L., Györfi, L., and Lugosi, G. (2013). *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media.

Devroye, L. and Wagner, T. (1979a). Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207.

Devroye, L. and Wagner, T. (1979b). Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604.

Devroye, L. and Wagner, T. J. (1982). 8 nearest neighbor methods in discrimination. *Handbook of Statistics*, 2:193–197.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. L. (2015). Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg. Springer-Verlag.

Feldman, V. and Vondrak, J. (2018). Generalization bounds for uniformly stable algorithms. *Advances in Neural Information Processing Systems*, 31.

Feldman, V. and Vondrak, J. (2019). High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pages 1270–1279. PMLR.

Frey, P. W. and Adesman, P. (1976). Recall memory for visually presented chess positions. *Memory & Cognition*, 4:541–547.

Jung, C., Ligett, K., Neel, S., Roth, A., Sharifi-Malvajerdi, S., and Shenfeld, M. (2019). A new analysis of differential privacy's generalization guarantees. *arXiv preprint arXiv:1909.03577*.

Kearns, M. and Ron, D. (1997). Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 152–162.

Larsen, K. G. (2023). Bagging is an optimal pac learner. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 450–468. PMLR.

Littlestone, N. and Warmuth, M. K. (1986). Relating data compression and learnability. Technical report.

McGregor, A., Mironov, I., Pitassi, T., Reingold, O., Talwar, K., and Vadhan, S. (2010). The limits of two-party differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE.

Moran, S. and Yehudayoff, A. (2016). Sample compression schemes for vc classes. *Journal of the ACM (JACM)*, 63(3):1–10.

Nachum, I., Shafer, J., and Yehudayoff, A. (2018). A direct sum result for the information complexity of learning. *arXiv preprint arXiv:1804.05474*.

Raginsky, M., Rakhlin, A., Tsao, M., Wu, Y., and Xu, A. (2016). Information-theoretic analysis of stability and bias of learning algorithms. *2016 IEEE Information Theory Workshop (ITW)*, pages 26–30.

Rogers, W. H. and Wagner, T. J. (1978). A finite sample distribution-free performance bound for local discrimination rules. *The Annals of Statistics*, pages 506–514.

Russo, D. and Zou, J. (2016). Controlling bias in adaptive data analysis using information theory. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1232–1240, Cadiz, Spain. PMLR.

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

Steinke, T. and Zakynthinou, L. (2020). Reasoning about generalization via conditional mutual information. In *Conference on Learning Theory*, pages 3437–3452. PMLR.

Van Handel, R. (2014). Probability in high dimension. *Lecture Notes (Princeton University)*, 2(3):2–3.

Xu, A. and Raginsky, M. (2017). Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2524–2533.