Algorithms and Uncertainty, Winter 2018/19 Lecture 26 (5 pages)

Overfitting, Stability, and Regularization

Instructor: Thomas Kesselheim

Today, we will talk about *overfitting*, a familiar problem in machine learning. One is given a set of labeled data points. From this sample, one should predict labels for future data points. It can happen that, when trying to make the training error small, one actually adapts to the noise present in the data. This way, the true error becomes large.

Example 26.1. *Suppose that your data points are in* $[0,1]$*. The correct label for* x *is* $y = x+\nu$*, where* $\nu \sim \text{Normal}(0, 0.0025)$*. So,* ν *is random noise from a Normal distribution with mean* 0 *and variance* 0.0025*. The best prediction one can make is clearly given by hypothesis* h *defined as* $h(x) = x$*.*

Figure 1 gives an example of eight data points drawn from this distribution with $x \sim \text{Uniform}[0,1]$*. One could now be tempted to infer that the correct labeling is given by a function* h *that matches the value in all given points, for example a polynomial of degree seven. In this case, it is given by*

$$h(x) = 5940.33x^7 - 20262.6x^6 + 27659.7x^5 - 19294.7x^4 + 7302.01x^3 - 1476.7x^2 + 148.067x - 5.53035 \ .$$

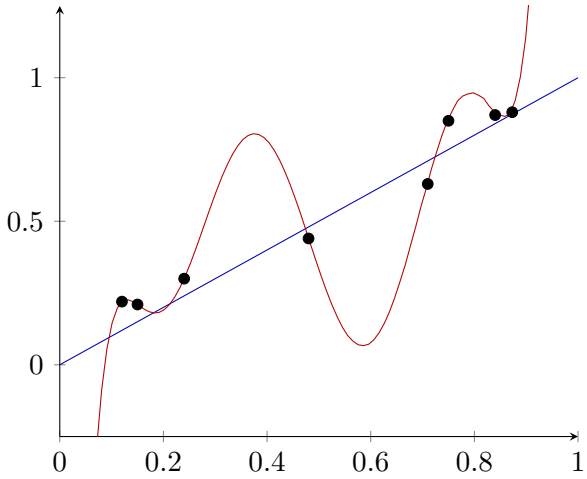

Figure 1: $x_i \sim \text{Uniform}[0,1]$, $y_i = x_i + \nu_i$, where $\nu_i \sim \text{Normal}(0, 0.0025)$. The blue line minimizes the true error; the red line is a degree-seven polynomial going exactly through the eight samples.

Today, we will first introduce a formal model, in which we can argue about overfitting. Afterwards, we will discuss regularization as a tool to avoid it and formally prove that it indeed does so.

1 Setting

There is a distribution $\mathcal{D}$ over pairs $z = (x, y) \in X \times Y$ such that y is the correct label for x. We are given a training set $S = \{z_1, \dots, z_m\}$, $z_i = (x_i, y_i) \in X \times Y$, of m samples drawn from $\mathcal{D}$ (so data points and their correct labels). We compute a hypothesis $h_S \colon X \to Y$, which predicts label $h_S(x)$ for point x.

An example to keep in mind is (linear) regression. In the easiest case, $X = \mathbb{R}$, $Y = \mathbb{R}$ and we use hypotheses h_S, which are straight lines or polynomial functions.

The error that hypothesis h makes on $z = (x, y)$ is defined by a loss function. We write $\ell(h, z)$ for this loss. In regression, one usually tries to minimize the squared error, so the loss function would be $\ell(h, z) = (h(x) - y)^2$.

Our goal is to have a small *true error*, which is defined as the expected loss of the hypothesis h_S that we compute on a point drawn from $\mathcal{D}$, that is

$$L_{\mathcal{D}}(h_S) = \mathbf{E}_{z\sim\mathcal{D}}\left[\ell(h_S, z)\right] \ .$$

This true error has two different sources. On the one hand, the hypothesis h_S might be incorrect on some samples in S. This is what we call the *training error*

$$L_S(h_S) = \frac{1}{m}\sum_{i=1}^{m} \ell(h_S, z_i) \ .$$

The amount by which the true error exceeds the training error

$$L_{\mathcal{D}}(h_S) - L_S(h_S)$$

is called the *generalization error*.

So, a learning algorithm overfits if it makes the training error too small at the cost of a high generalization error.

2 Stable Learning Algorithms do not Overfit

In order to bound the generalization error of a learning algorithm, we first rewrite it in a different form. It computes a hypothesis h_S based on a training set S. If we let S be a set of size m being drawn from $\mathcal{D}$ and I being drawn uniformly from $\{1, \ldots, m\}$, then we can write the expected training error as

$$\mathbf{E}\left[L_S(h_S)\right] = \mathbf{E}\left[\frac{1}{m}\sum_{i=1}^{m} \ell(h_S, z_i)\right] = \mathbf{E}\left[\ell(h_S, z_I)\right] \ .$$

We can write the expected true error as the expected loss on a fresh sample z' drawn from $\mathcal{D}$

$$\mathbf{E}\left[L_{\mathcal{D}}(h_S)\right] = \mathbf{E}\left[\ell(h_S, z')\right] \ .$$

There is also a different way to write the true error. Given samples $z_1, \ldots, z_m$ and z', let S^i denote the set $z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_m$. That is, we replace z_i by z'. As z_i and z' are both drawn from $\mathcal{D}$, they are identically distributed and we can swap their roles. Therefore for all i

$$\mathbf{E}\left[\ell(h_S, z')\right] = \mathbf{E}\left[\ell(h_{S^i}, z_i)\right] \ .$$

As this holds for any i, we can also use the random variable I from above. So, in combination

$$\mathbf{E}\left[L_{\mathcal{D}}(h_S)\right] = \mathbf{E}\left[\ell(h_{S^I}, z_I)\right] \ .$$

This means, that the expected generalization error can also be rewritten as

$$\mathbf{E}\left[L_{\mathcal{D}}(h_S) - L_S(h_S)\right] = \mathbf{E}\left[\ell(h_{S^I}, z_I) - \ell(h_S, z_I)\right] \ .$$

So, the expected generalization error can only be large if it ever happens that S^i and S (which differ only in a single point) lead to very different hypothesis. If this is guaranteed not to occur, we call the algorithm stable.

Definition 26.2. *A learning algorithm is* universally δ-replace-one stable *if for all S, i, and z' we have*

$$\ell(h_{S^i}, z_i) - \ell(h_S, z_i) \leq \delta \ .$$

Clearly, if the learning algorithm is universally δ-replace-one stable then the expected generalization error is at most δ. The great advantage of stability is that it does not talk anymore about distributions and statistical properties but rather a property of the algorithm. If the algorithm's output does not change a lot when replacing one point in the input by a different one, then the expected generalization error is small.

3 Regularized Risk Minimization

A good rule of thumb to avoid overfitting is to prefer "easier" hypotheses. One way to ensure this is to use *regularization*. Rather than choosing h_S so that $L_S(h)$ is minimized, we penalize "extreme" hypotheses and instead minimize $L_S(h) + R(h)$. As we will show, such regularized risk minimization is often universally δ-replace-one stable for small values of δ.

3.1 Assumptions

For our analysis of regularization, we will use the same set of assumptions and techniques that we already used in the context of online convex optimization. To this end, we will assume that hypotheses are defined by d-dimensional real vectors. For notational simplicity, we will not differentiate between the vectors and the associated hypotheses. We assume that the set of all hypotheses $\mathcal{H}$ is a convex subset of $\mathbb{R}^d$. Furthermore, there is a norm $\|\cdot\|$ defined on the set $\mathcal{H}$, which assigns a "length" to every vector in $\mathcal{H}$.

Recall the definition of strong convexity.

Definition 26.3. *Let $\sigma \geq 0$. A differentiable function F is σ-strongly convex if for all $\mathbf{u}, \mathbf{v}$*

$$F(\mathbf{u}) \geq F(\mathbf{v}) + \langle \nabla F(\mathbf{v}), \mathbf{u} - \mathbf{v} \rangle + \frac{\sigma}{2}\|\mathbf{u} - \mathbf{v}\|^2$$

A function is convex if it is 0-strongly convex.

Convexity requires the function F to stay above its tangent; strong convexity with $\sigma > 0$ additionally requires it to move away from it (see Figure 2 for an illustration).

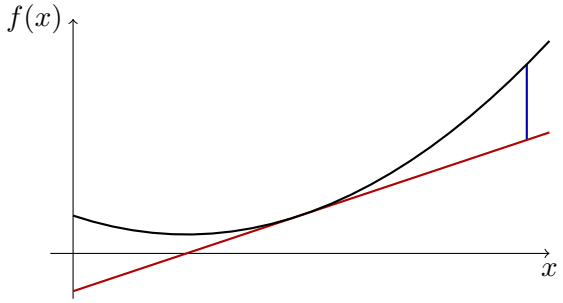

Figure 2: A strongly convex function moves away from the tangent (tangent drawn in red, distance drawn in blue).

We assume that the loss function is convex for every fixed z. That is, $h \mapsto \ell(h, z)$ is convex. Furthermore, it has to fulfill a Lipschitz condition by requiring that for all $h, h' \in \mathcal{H}$

$$\ell(h, z) - \ell(h', z) \leq \rho \|h - h'\| \ .$$

We will combine these loss functions with σ-strongly convex regularization and show that the resulting learning algorithm is universally δ-replace-one stable, where δ depends on σ, ρ, and the size of the training set m.

Example 26.4. *For regression with polynomials of degree at most $d-1$, we can identify hypotheses with coefficient vectors in $\mathbb{R}^d$. That is h is identified with $(a_0, \ldots, a_{d-1}) \in \mathbb{R}^d$ and $h(x) = \sum_{i=0}^{d-1} a_i x^i$. The usual goal in regression is to minimize the mean squared error. This is represented by the loss function*

$$\ell(h, z) = (h(x) - y)^2 \ .$$

For every fixed z, this function is convex.

Let us consider two hypotheses $h = (a_0, \ldots, a_{d-1})$ and $h' = (a'_0, \ldots, a'_{d-1})$. From the convexity of $\ell(\cdot, z)$, one can derive that if $z = (x, y) \in [0, 1]^2$ and $a_i \in [-\alpha, \alpha]$, then also

$$\ell(h, z) - \ell(h', z) \leq (2\alpha d + 1)\sqrt{d}\|h - h'\|_2 \ .$$

That is, if $x, y \in [0, 1]$ and $a_i \in [-\alpha, \alpha]$ for all i, the Lipschitz condition is fulfilled with $\rho = (2\alpha d + 1)\sqrt{d}$ in the ℓ_2-norm.

We have seen before that the function R with $R(h) = \frac{1}{2\eta}\sum_{i=1}^{d} a_i^2$ is $\frac{1}{\eta}$-strongly convex with respect to the ℓ_2-norm. In this context, the regularization function is known as Tikhonov regularization. Regression with Tikhonov regularization is also called ridge regression.

3.2 Stability with Strongly Convex Regularization

Whenever the regularizer is strongly convex, we can show a bound on universal replace-one stability.

Theorem 26.5. *Using a σ-strongly convex regularizer, regularized risk minimization with m samples is universally $\frac{2\rho^2}{m\sigma}$-replace-one-stable if the loss functions are convex and fulfill the Lipschitz condition with parameter ρ.*

We will make use of the following lemma, which we proved in the context of online convex optimization.

Lemma 26.6. *Let $F\colon S \to \mathbb{R}$ be a σ-strongly convex differentiable function over S with respect to a norm $\|\cdot\|$. Let $\mathbf{w} \in \arg\min_{\mathbf{v} \in S} F(\mathbf{v})$. Then, for all $\mathbf{u} \in S$*

$$F(\mathbf{u}) - F(\mathbf{w}) \geq \frac{\sigma}{2}\|\mathbf{u} - \mathbf{w}\|^2 \ .$$

Proof of Theorem 26.5. We use Lemma 26.6 on the fact that h_S minimizes $F(h) := \frac{1}{m}\sum_{j=1}^{m} \ell(h, z_j) + R(h)$, whereas h_{S^i} minimizes $F^i(h) := \frac{1}{m}\sum_{j=1, j\neq i}^{m} \ell(h, z_j) + \ell(h, z') + R(h)$.

So, we get

$$F(h_{S^i}) - F(h_S) \geq \frac{\sigma}{2}\|h_{S^i} - h_S\|^2$$

and

$$F^i(h_S) - F^i(h_{S^i}) \geq \frac{\sigma}{2}\|h_S - h_{S^i}\|^2 \ .$$

So, in combination

$$F(h_{S^i}) - F(h_S) + F^i(h_S) - F^i(h_{S^i}) \geq \sigma \|h_{S^i} - h_S\|^2$$

By the definitions of F and F^i, this is equivalent to

$$\frac{1}{m}\ell(h_{S^i}, z_i) - \frac{1}{m}\ell(h_{S^i}, z') - \frac{1}{m}\ell(h_S, z_i) + \frac{1}{m}\ell(h_S, z') \geq \sigma \|h_{S^i} - h_S\|^2 \ .$$

By the Lipschitz condition, we have

$$\ell(h_{S^i}, z_i) - \ell(h_S, z_i) \leq \rho \|h_{S^i} - h_S\| \quad \text{and} \quad \ell(h_{S^i}, z') - \ell(h_S, z') \leq \rho \|h_{S^i} - h_S\| \ .$$

So

$$2\rho \|h_{S^i} - h_S\| \geq m\sigma \|h_{S^i} - h_S\|^2 \ ,$$

and therefore

$$\|h_{S^i} - h_S\| \leq \frac{2\rho}{m\sigma} \ .$$

So also

$$\ell(h_{S^i}, z_i) - \ell(h_S, z_i) \leq \rho \|h_{S^i} - h_S\| \leq \frac{2\rho^2}{m\sigma} \ .$$

□

In this bound, it is important that $\delta = \frac{2\rho^2}{m\sigma}$ vanishes as m grows.