Algorithmic Foundations of Learning **Lecture 11**

Stochastic Oracle Model. Algorithmic Stability

Lecturer: Patrick Rebeschini *Version: December 8, 2021*

11.1 Introduction

During the past two lectures we introduced and analyzed various gradient methods. These methods can be applied to any first order oracle setting where *exact* subgradients at any given point can be computed. In many applications, however, an exact first order oracle may not exist, or a single call to the oracle may be very expensive, as in the case of machine learning. This justifies the introduction of *stochastic* oracle models and *randomized* algorithms, and the investigation of the interplay between optimization and randomness.

To see where the problem lies, computationally, let us recall the setting of linear predictors with ℓ_2 constraints described in Section 9.5 (this example will also serve to illustrate the theory that we will develop today).

Let $(X_1, Y_1), \dots, (X_n, Y_n) \in \mathcal{X} \times \{-1, 1\}$ be the training data, with $\mathcal{X} \subseteq \mathbb{R}^d$. Let $\varphi : \mathbb{R} \to \mathbb{R}_+$ be a given convex loss function. Let

$$\mathcal{A}_2 = \{x \in \mathbb{R}^d \to a(x) = w^\top x : w \in \mathcal{W}_2 \subseteq \mathbb{R}^d\},$$

where $\mathcal{W}_2$ is a convex set contained in a ball of radius $c_2^{\mathcal{W}}$, namely, $c_2^{\mathcal{W}} := \max_{w \in \mathcal{W}_2} \|w\|_2$. Let us assume that φ is γ_φ-Lipschitz and that $\mathcal{X}$ is contained in a ball of radius $c_2^{\mathcal{X}}$, namely, $c_2^{\mathcal{X}} := \max_{x \in \mathcal{X}} \|x\|_2$.
The risk minimization problem (the original problem we want to solve) reads

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & r(w) = \mathbf{E}\varphi(w^\top XY) \\ \text{subject to} \quad & w \in \mathcal{W}_2 \end{aligned} \tag{11.1}$$

Let $w_2^\star$ be a minimizer of this problem. The empirical risk minimization problem (a proxy to the original problem) reads

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & R(w) = \frac{1}{n}\sum_{i=1}^n \varphi(w^\top X_i Y_i) \\ \text{subject to} \quad & w \in \mathcal{W}_2 \end{aligned} \tag{11.2}$$

Let $W_2^\star$ be a minimizer of this problem.

Previously, we found that by running the projected subgradient method with $\eta_s \equiv \eta = \frac{2c_2^{\mathcal{W}}}{c_2^{\mathcal{X}}\gamma_\varphi\sqrt{t}}$, considering the time average $\overline{W}_t := \frac{1}{t}\sum_{s=1}^t W_s$, we can bound the excess risk as follows

$$\boxed{r(\overline{W}_t) - r(w_2^\star) \le \underbrace{R(\overline{W}_t) - R(W_2^\star)}_{\texttt{Optimization}_2} + \underbrace{\sup_{w \in \mathcal{W}_2}\{r(w) - R(w)\} + \sup_{w \in \mathcal{W}_2}\{R(w) - r(w)\}}_{\texttt{Statistics}_2}} \tag{11.3}$$

with

$$\boxed{\mathbf{E}\,\texttt{Statistics}_2 \le \frac{4c_2^{\mathcal{X}}c_2^{\mathcal{W}}\gamma_\varphi}{\sqrt{n}}} \qquad \boxed{\texttt{Optimization}_2 \le \frac{2c_2^{\mathcal{X}}c_2^{\mathcal{W}}\gamma_\varphi}{\sqrt{t}}}$$

This result *seems* appealing from a computational point of view, as it provides us with a principled approach to stop the projected gradient method after $t \sim n$ time steps (that is, as far as our guarantees go, we do not need to solve the optimization problem with an arbitrarily good accuracy but only with an accuracy that is of the same order as the statistical accuracy, which is a function of the number of training data n at our disposal). However, this result is only concerned with the *number of iterations* that our iterative algorithm needs to make in order to meet a prescribed level of error accuracy, and it does not say anything about the *computational cost* needed to meet this accuracy. Indeed, from a theoretical point of view, this result lies within the first order oracle model, which is only concerned with the number of calls to the oracle, and it does not say anything about the computational cost of the oracle.

We immediately see where the problem comes from in large-scale machine learning, where the number of data points at our disposal (n) is typically large. In this setting, a single computation of the subgradient is very costly, as we have

$$\partial R(w) = \frac{1}{n}\sum_{i=1}^{n} \partial_w \varphi(w^\top X_i Y_i)$$

which involves a sum of n term and is prohibitive when n is big. Here, the notation ∂_w indicates the subgradient of the function $w \to \varphi(w^\top X_i Y_i)$. For instance, if φ is differentiable and we denote by φ' its derivative, we have

$$\partial_w \varphi(w^\top X_i Y_i) = \varphi'(w^\top X_i Y_i) X_i.$$

In large-scale machine learning, one would like to apply gradient methods by only using a *subset* of the data to compute each subgradient.

Today we will see that running gradient methods by computing subgradient *approximations* that only use *one* data point per time step (i.e., with computational cost $O(1)$ per time step instead of $O(n)$ as required by the computation of the full gradient) is enough to meet the same level of error accuracy in expectation, namely, to get

$$\boxed{\mathbf{E}\,\texttt{Optimization}_2 \le \frac{2c_2^{\mathcal{X}} c_2^{\mathcal{W}} \gamma_\varphi}{\sqrt{t}}}$$

To get there, we introduce the general framework of stochastic oracles, where we accept some level of randomness in the first order oracle but require it to be unbiased. We will see that this general framework allows us to consider other ways to solve problem (11.1), without the need to go through the empirical risk minimization problem (11.2) and the decomposition (11.3).

We will then introduce the notion of *algorithmic stability*, and show how this notion can be used to directly analyze the excess risk (not only the estimation error) via *implicit* or *algorithmic* regularization, as opposed to *explicit* or *structural* regularization (which is what we have used so far in this course by considering an admissible set $\mathcal{A}$). In particular, we investigate the stability of stochastic gradient descent and the notion of early stopping.

11.2 Stochastic Oracle Model

We are given the following problem over $x \in \mathbb{R}^d$:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{C}, \end{array} \tag{11.4}$$

where f is a convex function, possibly not known, and $\mathcal{C}$ is a known convex subset of $\mathbb{R}^d$. Previously, we assumed that we had a first order oracle which, given $x \in \mathcal{C}$, would yield $g \in \partial f(x)$, a subgradient of f

evaluated at x. When analysing our algorithms in terms of oracle complexity, it was the number of calls to the oracle that was important. In many applications, however, an oracle may not exist, or a single call to the oracle may be very expensive. In such cases, we accept randomness in our oracle but require it to be unbiased.

Formally, a first order *stochastic* oracle model defines the framework where given $x \in \mathcal{C}$ an oracle yields back a random variable G that is an unbiased estimator of a subgradient of f at x, namely, $\mathbf{E}G \in \partial f(x)$. If X is a random variable, the oracle yields back a random variable G that is an unbiased estimator of a subgradient of f at X conditionally on X, namely, $\mathbf{E}[G|X] \in \partial f(X)$.

11.3 Projected Stochastic Subgradient Methods

We analyze the behavior of the projected gradient algorithm to solve problem (11.4) when we replace exact knowledge of subgradients of f with unbiased estimates of them. For a given initial point $X_1 \in \mathcal{C}$, possibly random, and a given collection of step sizes $\eta_1, \eta_2, \ldots$, the projected stochastic gradient method is defined by the sequence of random variables generated according to the following update.

```
Algorithm 1: Projected Stochastic Subgradient Method
Input: $X_1$, $\{\eta_s\}_{s\geq 1}$, stopping time $t$;
for $s = 1, \ldots, t$ do
    $\widetilde{X}_{s+1} = X_s - \eta_s G_s$, where $\mathbf{E}[G_s|X_s] \in \partial f(X_s)$,
    $X_{s+1} = \Pi_{\mathcal{C}}(\widetilde{X}_{s+1})$.
end
```

Remarkably, when the function f is γ-Lipschitz, then, in expectation, we recover the rate of convergence of the deterministic oracle (cf. Theorem 9.3). In fact, for the following result to hold, we do not need φ to be Lipschitz, but we need something weaker: we need to assume that the fluctuations of each subgradient are bounded in the following way $\mathbf{E}[\|G_s\|_2^2] \leq \gamma^2$.

Theorem 11.1 (Projected stochastic subgradient method) *Let f be convex. Assume that $\mathbf{E}[\|G_s\|_2^2] \leq \gamma^2$ for any $s \in [t]$ and that $\mathbf{E}\|X_1 - x^\star\|_2^2 \leq b^2$. Then, the projected subgradient algorithm with $\eta_s \equiv \eta = \frac{b}{\gamma\sqrt{t}}$ satisfies*

$$\boxed{\mathbf{E}f\left(\frac{1}{t}\sum_{s=1}^{t} X_s\right) - f(x^\star) \leq \frac{\gamma b}{\sqrt{t}}}$$

Proof: By convexity and the properties of conditional expectations, for any $1 \leq s \leq t$ we have

$$f(X_s) - f(x^\star) \leq \partial f(X_s)^\top (X_s - x^\star) = \mathbf{E}[G_s|X_s]^\top (X_s - x^\star) = \mathbf{E}[G_s^\top (X_s - x^\star)|X_s].$$

Proceeding as in the proof of Theorem 9.3, we find

$$G_s^\top (X_s - x^\star) \leq \frac{1}{2\eta}(\|X_s - x^\star\|_2^2 - \|X_{s+1} - x^\star\|_2^2) + \frac{\eta}{2}\|G_s\|_2^2.$$

Taking the expectation, by the tower property we obtain

$$\mathbf{E}f(X_s) - f(x^\star) \leq \mathbf{E}G_s^\top (X_s - x^\star) \leq \frac{1}{2\eta}(\mathbf{E}\|X_s - x^\star\|_2^2 - \mathbf{E}\|X_{s+1} - x^\star\|_2^2) + \frac{\eta}{2}\mathbf{E}\|G_s\|_2^2,$$

and using the assumption $\mathbf{E}\|G_s\|_2^2 \le \gamma^2$ we obtain

$$\frac{1}{t}\sum_{s=1}^{t}(\mathbf{E}f(X_s) - f(x^\star)) \le \frac{1}{2\eta t}\left(\mathbf{E}\|X_1 - x^\star\|_2^2 - \mathbf{E}\|X_{t+1} - x^\star\|_2^2\right) + \frac{\eta}{2}\gamma^2 \le \frac{b^2}{2\eta t} + \frac{\eta\gamma^2}{2}.$$

Selecting $\eta = \frac{b}{\gamma\sqrt{t}}$ to minimize the right-hand side of the above inequality gives the final result, as

$$f\left(\frac{1}{t}\sum_{s=1}^{t} X_s\right) \le \frac{1}{t}\sum_{s=1}^{t} f(X_s)$$

by Jensen's inequality. ■

Theorem 11.1 is quite remarkable as it shows that, in expectation, the projected stochastic subgradient method yields the same convergence guarantees as the deterministic counterpart analyzed in Theorem 9.3. In particular, the oracle complexity is the same[1]: to get an accuracy ε, both methods requires $O(1/\varepsilon^2)$ calls to their respective oracles. The main advantage of the stochastic version lies in the fact that in some applications the *computational* complexity involved in having access to a stochastic oracle is much cheaper than in the deterministic case, as we will see below in machine learning.

Theorem 11.1 is an exception and in most cases, as it is natural to expect, we do pay a price by working with a stochastic oracle as opposed to an exact oracle. In particular, one can show that smoothness does not bring any improved rate of convergence in the stochastic oracle case. On the other hand, recall that in the deterministic setting (i.e., exact oracle) smoothness allows gradient descent to achieve a $O(1/\varepsilon)$ oracle complexity (Theorem 9.4) instead of the worse $O(1/\varepsilon^2)$ for the Lipschitz case (Theorem 9.3) — and this complexity can be further reduced to $O(1/\sqrt{\varepsilon})$ using accelerated methods.

However, one can combine the stochastic oracle model with variance reduction techniques to design algorithms that get overall computational savings. See **Problem 4.5** in the Problem Sheets, for instance.

11.4 Projected Stochastic Mirror Descent

As in the deterministic setting Theorem 9.3 for gradient descent can be seen as a particular instance of Theorem 10.11 for mirror descent, also in the present case of a stochastic oracle it is possible to derive a general statement for stochastic mirror descent.

Algorithm 2: Projected Stochastic Mirror Descent

```
Input: $X_1$, $\{\eta_s\}_{s\ge 1}$, stopping time $t$;
for $s = 1, \dots, t$ do
    $\nabla\Phi(\widetilde{X}_{s+1}) = \nabla\Phi(X_s) - \eta_s G_s$, where $\mathbf{E}[G_s | X_s] \in \partial f(X_s)$,
        $X_{s+1} = \Pi_{\mathcal{C}}^{\Phi}(\widetilde{X}_{s+1})$.
end
```

Theorem 11.2 (Projected stochastic mirror descent) *Let f be convex. Assume that $\mathbf{E}[\|G_s\|_*^2] \le \gamma^2$ for any $s \in [t]$. Let Φ be a α-strongly convex mirror map on $\mathcal{C} \cap \mathcal{D}$ with respect to the norm $\|\cdot\|$. Assume*

[1] Note that different notions of accuracy are used, however, as we consider the expected value of the stochastic method.

that $X_1 \equiv x_1 \in \operatorname{argmin}_{x\in\mathcal{C}\cap\mathcal{D}} \Phi(x)$*. Then, projected mirror descent with* $\eta = \frac{c}{\gamma}\sqrt{\frac{2\alpha}{t}}$ *satisfies*

$$\boxed{\mathbf{E} f\left(\frac{1}{t}\sum_{s=1}^{t} X_s\right) - f(x^\star) \le c\gamma\sqrt{\frac{2}{\alpha t}}}$$

where $c^2 = \sup_{x\in\mathcal{C}\cap\mathcal{D}} \Phi(x) - \Phi(x_1)$*.*

Proof: See **Problem 3.4** in the Problem Sheets. ■

11.5 Back to Learning: Linear Predictors with ℓ_2 Constraints

In the case of linear predictors with ℓ_2 constraints the problem that we want to solve is the expected risk problem (11.1). The assumption here is that we know the loss function φ (indeed, we can choose it!) and the constraint set $\mathcal{C}$, but we do not know the distribution of (X, Y). We have access to n i.i.d. samples $(X_1, Y_1), \ldots, (X_n, Y_n)$ from this unknown distribution.

Armed with the concept of a stochastic oracle model, we can now attack the expected risk minimization problem in a computationally-efficient way, presenting with two approaches that yield subgradient evaluations at a cost $O(1)$.

11.5.1 Stochastic Gradient Method: Single Pass Through the Data

We can solve the expected risk minimization problem *directly*, by applying the stochastic gradient method directly to (11.1) for $t \le n$ steps, where at each step $s \in [n]$ we choose

$$G_s = \partial_w \varphi(W_s^\top X_s Y_s)$$

(note that the optimization is over $w \in \mathbb{R}^d$ now). This is an unbiased estimator of the subgradient of the expected risk r at W_s as, by the independence of W_s and (X_s, Y_s) (note that W_s is a function of (X_i, Y_i) for $i \in [s-1]$ and is independent of (X_s, Y_s)), we have

$$\mathbf{E}[\partial_w \varphi(W_s^\top X_s Y_s) | W_s] = \partial r(W_s).$$

The requirement that $t = n$ ensures that each data point (X_s, Y_s) is used only once in the algorithm. This requirement is necessary to ensure the unbiasedness of the subgradient estimators. Hence, the algorithm performs a single pass through the data. Such an approach is well-suited for an online learning algorithm in which a data is processed and discarded in a stream.

In this case, applying Theorem 11.1 we find, for any $t \le n$,

$$\boxed{\mathbf{E} r(\overline{W}_t) - r(w_2^\star) \le \frac{2 c_2^{\mathcal{X}} c_2^{\mathcal{W}} \gamma_\varphi}{\sqrt{t}}}$$

In other words, pursuing this approach we can directly provide a bound for the estimation error without using the decomposition (11.3) that we introduced in Lecture 1.

11.5.2 Stochastic Gradient Method: Multiple Passes Through the Data

We can solve the empirical risk minimization problem by applying the stochastic gradient method to (11.2) considering the dataset $S = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ as deterministic, or, equivalently, conditioning on the data. For any number of time steps t (not necessarily $t \le n$), where at each step $s \in [n]$, we choose

$$G_s = \partial_w \varphi(W_s^\top X_{I_{s+1}} Y_{I_{s+1}}),$$

where $I_2, I_3, I_4, \dots$ is a sequence of i.i.d. random variables (independent of the data random variables) uniformly distributed in $\{1, \dots, n\}$. This is an unbiased estimator of the subgradient of the empirical risk R at W_s as

$$\mathbf{E}[\partial_w \varphi(W_s^\top X_{I_{s+1}} Y_{I_{s+1}})|S, W_s] = \frac{1}{n}\sum_{i=1}^n \mathbf{E}[\partial \varphi(W_s^\top X_i Y_i)|S, W_s] = \partial R(W_s).$$

Since unlimited i.i.d. uniform samples are available, multiple passes over the data can be made.

Applying Theorem 11.1 we find, for any $t \ge 1$,

$$\boxed{\mathbf{E}\,\texttt{Optimization}_2 = \mathbf{E}R(\overline{W}_t) - R(W_2^\star) \le \frac{2c_2^{\mathcal{X}} c_2^{\mathcal{W}} \gamma_\varphi}{\sqrt{t}}}$$

11.6 Algorithmic Stability. Algorithmic/Implicit Regularization

As we saw in Section 11.5.1, the stochastic oracle model allows us to directly look at the *regularized* expected risk minimization problem (e.g., problem (11.1)) without going through the *regularized* empirical risk minimization problem (e.g., problem (11.2)), hence avoiding the need of the decomposition for the estimation error introduced in Lecture 1, namely (in its general formulation, for a generic algorithm A),

$$\begin{aligned}\underbrace{r(A) - r(a^\star)}_{\text{estimation error}} &= r(A) - R(A) + \underbrace{R(A) - R(A^\star)}_{\text{optimization error}} + \underbrace{R(A^\star) - R(a^\star)}_{\le 0} + R(a^\star) - r(a^\star) \\ &\le \underbrace{R(A) - R(A^\star)}_{\text{optimization error}} + \underbrace{\sup_{a\in\mathcal{A}}(r(a) - R(a)) + \sup_{a\in\mathcal{A}}(R(a) - r(a))}_{\text{statistics error}}.\end{aligned}$$

Here, we use the term "regularized" to indicate that we deal with constrained optimization problems. Recall that $a^\star$ is, by definition, a minimizer of the expected risk r over $a \in \mathcal{A}$, for a given set of admissible actions $\mathcal{A} \subseteq \mathcal{B}$ (in this course we assume that the minimizers always exists, i.e., the infima are attained). This form of regularization is *explicit* or *structural*, and allows us to derive error bounds for the statistical error that depend on notions of complexity of the set $\mathcal{A}$ (see Rademacher complexity, VC dimension, etc). This follows as in the above error decomposition we take the supremum over $a \in \mathcal{A}$ to simplify the problem and reduce the original form of randomness to a form that is amenable to computations. In fact, note that a generic algorithm A can, in principle, be a very nonlinear function of the data $Z_1, \dots, Z_n \in \mathcal{Z}$. On the other hand, for a deterministic $a \in \mathcal{A}$, the quantity $R(a) = \frac{1}{n}\sum_{i=1}^n \ell(a, Z_i)$ is a simple function of the data, an average sum of independent random variables.

Recall that

$$\underbrace{r(A) - r(a^{\star\star})}_{\text{excess risk}} = \underbrace{r(A) - r(a^\star)}_{\text{estimation error}} + \underbrace{r(a^\star) - r(a^{\star\star})}_{\text{approximation error}}$$

We will now introduce a technique, known as *algorithmic stability*, that shows that if a generic algorithm $A \in \mathcal{B}$ is stable upon perturbations of the data, then, in expectation, its *excess risk* (not the estimation error over an admissible set!) is bounded.

To achieve this, we first need a new error decomposition for the expected value of the excess risk. In the following, we denote $A^{\star\star}$ as any minimizer of the empirical risk R over $\mathcal{B}$.

Proposition 11.3 *For any $A \in \mathcal{B}$ we have*

$$\boxed{\mathbf{E}\underbrace{r(A) - r(a^{\star\star})}_{\textit{excess risk}} \leq \mathbf{E}\ \underbrace{[r(A) - R(A)]}_{\textit{generalization error}}\ + \mathbf{E}\underbrace{[R(A) - R(A^{\star\star})]}_{\textit{optimization error}}}$$

Proof: We have

$$r(A) - r(a^{\star\star}) = r(A) - R(A) + R(A) - R(A^{\star\star}) + R(A^{\star\star}) - r(a^{\star\star}).$$

Note that $\mathbf{E}R(A^{\star\star}) \leq r(a^{\star\star})$, as for any $a \in \mathcal{B}$ we have $R(A^{\star\star}) \leq R(a)$ (as, by definition, $A^{\star\star}$ is a minimizer of the empirical risk R over $\mathcal{B}$) so that

$$\mathbf{E}R(A^{\star\star}) \leq \mathbf{E}R(a) = r(a),$$

which holds also for $a = a^{\star\star}$. ■

Proposition 11.3 breaks the problem into two components: the expected value of the optimization error (which we have already analyzed when the algorithm A is full-batched/stochastic gradient/mirror method), and a generalization error that directly involves the algorithm A and does not involve the supremum over an admissible set (note that Proposition 11.3 does not say anything about the presence of an admissible set!)

Remark 11.4 (On the generalization error) *Note that the generalization error is not the ultimate object of interest. We care about finding an algorithm A that minimizes the excess risk, not just the generalization error. The generalization error is trivially minimized by choosing a constant (i.e. independent of the data) algorithm: for any* fix $a \in \mathcal{A}$ *we have* $\mathbf{E}\,[r(a) - R(a)] = 0$.

Let $A \in \mathcal{B}$ be a given algorithm, function of the random variables $Z_1, \dots, Z_n$ and, possibly, function of other sources of randomness. For each $i \in [n]$, let $\widetilde{Z}_i$ be a resampled (independent) random variable coming from the same (unknown) data distribution. Let $\widetilde{A}(i)$ denote the output of the same algorithmic procedure when run on the perturbed dataset $\{Z_1, \dots, Z_{i-1}, \widetilde{Z}_i, Z_{i+1}, \dots, Z_n\}$ in which the i-th data is replaced by $\widetilde{Z}_i$. The following result shows that the generalization error is *equal* to the average mean deviation of the loss function evaluated at the perturbed outputs and. If the loss function is Lipschitz, the generalization error is upper bounded by the average mean deviation of the perturbed outputs with respect to a given norm.

Proposition 11.5 (Generalization error bound via algorithmic stability) *We have*

$$\boxed{\mathbf{E}[r(A) - R(A)] = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}[\ell(A, \widetilde{Z}_i) - \ell(\widetilde{A}(i), \widetilde{Z}_i)]}$$

In particular, if for any $z \in \mathcal{Z}$ the function $a \to \ell(a, z)$ is γ-Lipschitz with respect to the norm $\|\cdot\|$, we have

$$\boxed{\mathbf{E}[r(A) - R(A)] \leq \frac{\gamma}{n}\sum_{i=1}^{n} \mathbf{E}\|A - \widetilde{A}(i)\|}$$

Proof: As the resampled observation $\widetilde{Z}_i$ has the same distribution than Z_i and is independent of both A and $Z_1, \ldots, Z_n$, we have

$$\mathbf{E}\, r(A) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}\, \ell(A, \widetilde{Z}_i).$$

As the pair (A, Z_i) has the same distribution as the pair $(\widetilde{A}(i), \widetilde{Z}_i)$, the expectation of the empirical risk can be written as

$$\mathbf{E} R(A) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}\, \ell(A, Z_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}\, \ell(\widetilde{A}(i), \widetilde{Z}_i).$$

■

An algorithm A is stable if the deviation $\|A - \widetilde{A}(i)\|$ is small, so that the generalization error can be controlled via Proposition 11.5. In the next section we investigate the algorithmic stability of stochastic gradient descent, namely, $A = W_t$, and show that it decreases linearly with the number of data points n, in the case of smooth losses.

11.7 Stability for Stochastic Gradient Descent. Early Stopping

We now investigate the generalization error of stochastic gradient descent applied to solve the generic (uncostrained) empirical risk minimization problem:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad R(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, Z_i). \tag{11.5}$$

We consider the "multiple passes" version of stochastic gradient descent described in section 11.5.2, which, in the case of differentiable functions reads, for a given initial point $W_1 \in \mathbb{R}^d$,

$$W_{s+1} = W_s - \eta_s \nabla_w \ell(W_s, Z_{I_{s+1}}),$$

where the notation ∇_w denotes to the gradient of the function $w \to \ell(w, z)$, and where $I_2, I_3, I_4, \ldots$ is a sequence of i.i.d. random variables (independent of the data random variables) uniformly distributed in $\{1, \ldots, n\}$.

We now investigate the stability property of the algorithm W_t. To this end, for any $i \in [n]$ we let $\widetilde{W_t}(i)$ denote the output of stochastic gradient descent at time t applied to the problem (11.5) when the original dataset $\{Z_1, \ldots, Z_n\}$ is replaced by the "perturbed" dataset $\{Z_1, \ldots, Z_{i-1}, \widetilde{Z}_i, Z_{i+1}, \ldots, Z_n\}$, where $\widetilde{Z}_i$ is an independent random variable coming from the same (unknown) distribution of the data. We stress that $\widetilde{W_t}(i)$ is a purely theoretical device that we use in theory to address the stability property of the stochastic gradient descent algorithm. We note that W_t and $\widetilde{W_t}(i)$ are random variables that can be thought of as the output of the *same* deterministic function f_t applied to the random variables that define the models:

$$\begin{aligned} W_t &= f_t(Z_1, \ldots, Z_{i-1}, Z_i, Z_{i+1}, \ldots, Z_n, I_2, \ldots, I_t), \\ \widetilde{W_t} &= f_t(Z_1, \ldots, Z_{i-1}, \widetilde{Z_i}, Z_{i+1}, \ldots, Z_n, I_2, \ldots, I_t). \end{aligned}$$

In particular, note that W_t and $\widetilde{W_t}$ depend on the *same* random variables Z_j, for $j \neq i$, and on the *same* random variables I_s's.

Lemma 11.6 (Generalisation error bound for convex, Lipschitz, and smooth losses) *Assume that for any $z \in \mathcal{Z}$ the function $w \in \mathbb{R}^d \to \ell(w, z)$ is convex, γ-Lipschitz and β-smooth with respect to the Euclidean norm $\| \cdot \|_2$. Then, stochastic gradient descent with $\eta_s \equiv \eta$ satisfying $\eta\beta \leq 2$ yields, for any $t \geq 1$,*

$$\boxed{\mathbf{E}\|W_t - \widetilde{W}_t(i)\|_2 \leq \frac{2\eta\gamma}{n}(t-1)}$$

In particular,

$$\boxed{\mathbf{E}[r(W_t) - R(W_t)] \leq \frac{2\eta\gamma^2}{n}(t-1)}$$

Lemma 11.6 shows that the stability term $\mathbf{E}\|W_t - \widetilde{W}_t(i)\|_2$ for stochastic gradient descent *increases* linearly with time (it can be shown that the bound in the Lemma is tight). This result can be combined with a bound on the expected value of the optimization error, which *decreases* as a function of time, to yield a form of *implicit* or *algorithmic* regularization: we can find the optimal time that minimizes the upper bound given by Proposition 11.3. This technique is called *early stopping*, and it represents way to implement regularization. Other ways are: by means of solving a constrained problem, as we have seen so far in this course, or by means of solving an unconstrained problem with a penalty term in the objective function, as we will see next time with the Lasso algorithm for parameter estimation in high-dimensional statistics.

The proof of Lemma 11.6 relies on the fact that for a sufficiently small step size the gradient descent updates with smooth and convex function are non-expansive with respect to the Euclidean norm $\| \cdot \|_2$.

Proposition 11.7 (Non-expansivity of gradient update) *Let f be a β-smooth function, convex, and $\eta\beta \leq 2$ with $\eta > 0$. Then, for any $x, y \in \mathbb{R}$,*

$$\|x - y - \eta(\nabla f(x) - \nabla f(y))\|_2 \leq \|x - y\|_2.$$

Proof: When f is convex and β-smooth, the gradients are co-coercive, that is,

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \frac{1}{\beta}\|\nabla f(x) - \nabla f(y)\|_2^2.$$

The result then comes from the following

$$\begin{aligned}\|x-y-\eta(\nabla f(x)-\nabla f(y))\|_2^2 &= \|x-y\|_2^2 - 2\eta(\nabla f(x)-\nabla f(y))^\top(x-y) + \eta^2\|\nabla f(x)-\nabla f(y)\|_2^2 \\ &\leq \|x-y\|_2^2 - \Big(\frac{2\eta}{\beta}-\eta^2\Big)\|\nabla f(x)-\nabla f(y)\|_2^2 \leq \|x-y\|_2^2.\end{aligned}$$

■

We are now ready to present the proof of Lemma 11.5.

Proof:[Proof of Lemma 11.6] Fix $i \in [n]$. For any $t \geq 0$ let $\delta_t := \mathbf{E}\|W_t - \widetilde{W}_t(i)\|_2$. Define

$$\widetilde{U}_j := \begin{cases} Z_j & \text{if } j \neq i \\ \widetilde{Z}(i) & \text{if } j = i \end{cases}$$

Note that, by definition, $W_1 = \widetilde{W}_1(i)$ and

$$\begin{aligned} W_{t+1} &= W_t - \eta\nabla_w \ell(W_t, Z_{I_{t+1}}), \\ \widetilde{W}_{t+1}(i) &= \widetilde{W}_t(i) - \eta\nabla_w \ell(\widetilde{W}_t(i), U_{I_{t+1}}). \end{aligned}$$

By the law of total expectations (or tower property), by conditioning on I_{t+1} we find

$$\begin{aligned}
\delta_{t+1} &= \mathbf{E}\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 \\
&= \mathbf{E}\,\mathbf{E}[\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 | I_{t+1}] \\
&= \sum_{j\in[n]} \mathbf{E}[\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 | I_{t+1} = j]\mathbf{P}(I_{t+1} = j) \\
&= \mathbf{E}[\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 | I_{t+1} = i]\mathbf{P}(I_{t+1} = i) + \sum_{j\in[n]\setminus\{i\}} \mathbf{E}[\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 | I_{t+1} = j]\mathbf{P}(I_{t+1} = j).
\end{aligned}$$

For any $j \in [n]$ we have $\mathbf{P}(I_{t+1} = j) = 1/n$. By the independence of I_{t+1} and the data random variables, using the triangle inequality and the assumption $\|\nabla_w \ell(W_t, Z_i)\|_2 \leq \gamma$ we find

$$\mathbf{E}[\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 | I_{t+1} = i] = \mathbf{E}\|W_t - \widetilde{W}_t(i) + \eta\nabla_w \ell(\widetilde{W}_t(i), \widetilde{Z}_i) - \eta\nabla_w \ell(W_t, Z_i)\|_2 \leq \delta_t + 2\eta\gamma.$$

On the other hand, using Proposition 11.7 we find, for any $j \neq i$,

$$\mathbf{E}[\|W_{t+1} - \widetilde{W}_{t+1}(i)\|_2 | I_{t+1} = j] = \mathbf{E}\|W_t - \widetilde{W}_t(i) + \eta\nabla_w \ell(\widetilde{W}_t(i), Z_j) - \eta\nabla_w \ell(W_t, Z_j)\|_2 \leq \delta_t.$$

Putting everything together we finally obtain

$$\delta_{t+1} \leq \frac{\delta_t + 2\eta\gamma}{n} + \frac{(n-1)\delta_t}{n} = \delta_t + \frac{2\eta\gamma}{n},$$

which yields $\delta_t = 2\eta\gamma(t-1)/n$ using that $\delta_1 = 0$ as $W_1 = \widetilde{W}_1(i)$. The proof follows by using Proposition 11.5. ■

The results in Lemma 11.6 can be strengthened if one additionally assumes that the function f is strongly convex. See **Problem 3.6** in the Problem Sheets.