Stochastic methods for optimization and sampling.

Spring 2026

Contents

Part I

Convergence rates for stochastic algorithms

Chapter 1

Introduction

The goal of this course is to study stochastic algorithms aimed at minimizing a function $F : \Theta \to \mathbb{R}$ for some subset $\Theta \subseteq \mathbb{R}^d$, $d \geq 1$

$$\min_{x \in \Theta} F(x).$$

The set Θ can be equal to $\mathbb{R}^d$ (unconstrained problem) or a strict subset of $\mathbb{R}^d$ (constrained problem). To solve this problem for instance when $\Theta = \mathbb{R}^d$, one of the most popular algorithms is Gradient Descent (GD). Starting from an initial guess x_0, this algorithm iteratively defines a sequence $x_0, x_1, \ldots, x_K$ where

$$x_{k+1} = x_k - \gamma_k \nabla F(x_k), \quad \forall k = 0, \ldots, K-1.$$

Here, $\gamma_k > 0$ is called a step size. Gradient Descent has numerous appealing optimization properties that we will precisely recall in Section 1.2 below.

A significant drawback of Gradient Descent arises when the evaluation of the gradient $\nabla F(x_k)$ is computationally costly. In such scenarios, each iteration of Gradient Descent can become exceedingly time-consuming, making the algorithm impractical for real-life applications. This can happen in the following practical cases of interest.

1. **Empirical risk minimization.** Suppose F can be decomposed as an average of N functions $f_i : \Theta \longrightarrow \mathbb{R}$, $i = 1, \ldots, N$, and we are interested in the problem

   $$\min_{x \in \Theta} F(x) = \min_{x \in \Theta} \frac{1}{N} \sum_{i=1}^{N} f_i(x), \tag{1.1}$$

   for some "very large" N (of the order of millions or billions). In this case, evaluating the gradient $\nabla F(x_k)$ requires computing N gradients $\{\nabla f_i(x_k) : i = 1, \ldots, N\}$, which can be prohibitive when N is large.

   Such objective functions naturally arise in supervised machine learning problems, especially when resorting to *empirical risk minimization.* Given N i.i.d. observations

   $$(X_1, Y_1), \ldots, (X_N, Y_N) \in \mathcal{X} \times \mathcal{Y},$$

   one typically looks for a predictor $\varphi : \mathcal{X} \to \mathcal{Y}$ such that, for a new observation (X_{N+1}, Y_{N+1}) with the same distribution as (X_1, Y_1), the predictor $\varphi(X_{N+1})$ would "predict well" Y_{N+1} from X_{N+1}. To quantify this, let $\ell : \mathcal{Y}^2 \to \mathbb{R}_+$ be a loss function, typically convex, and satisfying

$\ell(y, y) = 0$, $\forall y \in \mathbb{R}$. The quantity $\ell(Y_i, \varphi(X_i))$ represents the error incurred when predicting Y_i by $\varphi(X_i)$. For a parametrization $(\varphi_\theta)_{\theta\in\Theta}$, the method of empirical risk minimization selects the parameter $\theta \in \Theta$ as the minimizer of the empirical risk on the dataset:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, \varphi_\theta(X_i)), \tag{1.2}$$

which is a minimization problem of the form (1.1). Under suitable conditions, $\hat{\theta}$ will have good generalization properties.

Example: (Linear regression). We observe $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$ and there exists $\theta^* \in \mathbb{R}^d$ such that

$$\underbrace{y_i}_{\text{observed}} = \underbrace{x_i}_{\text{observed}}{}^\top \underbrace{\theta^*}_{\text{to learn}} + \xi_i, \qquad \forall i = 1, \ldots, n.$$

Here ξ_i's are i.i.d., centered noises, mutually independent of the $x_i's$. We aim to estimate θ^* or learn a predictor of the form $\varphi_\theta(x) = x^\top\theta$. In this context, the most classical estimator is the Ordinary Least Squares (OLS) estimator, which is well-suited in small dimensions (i.e. when $d < n$).

$$\begin{aligned}\hat{\theta}^{OLS} &= \arg\min_{\theta\in\mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^\top\theta)^2 \\ &= \arg\min_{\theta\in\mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \varphi_\theta(x_i)),\end{aligned}$$

where $\ell(a, b) = (a - b)^2$. This problem is an instance of (1.1), and more specifically (1.2).

2. **Function values given by an integral**. Suppose the function F can be written as

$$F(x) = \int_\Omega h(x, y)\, \mu(y)dy, \quad \forall x \in \Theta, \tag{1.3}$$

for some function $h : \Theta \times \Omega \to \mathbb{R}$ where $\Omega \subseteq \mathbb{R}^p$, $p \geq 1$ and some measure μ over Ω. The problem of evaluating the function F or its gradient ∇F at some point x might be intractable, even if $h(x, y)$ admits a simple expression for any $x \in \Theta$ and $y \in \Omega$. In such a case, it is impossible to apply the gradient descent method, and further techniques are required to minimize $F(x)$ for $x \in \Theta$.

Note that both cases are encompassed in the following more general formulation, that we will adopt throughout the course:

$$\min_{x\in\Theta} F(x) = \min_{x\in\Theta} \mathbb{E} f(x, \xi), \tag{1.4}$$

for some measurable function

$$\begin{aligned} f : \Theta \times \Xi &\to \mathbb{R} \\ (x, t) &\mapsto f(x, t) \end{aligned}$$

and a random variable ξ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in Ξ. We will assume that $\mathbb{E}(|f(x, \xi)|) < +\infty$.

Indeed, suppose first that ξ is the uniform random variable over the set $\Xi = \{1, \ldots, N\}$. Then for any $x \in \Theta$ the quantity $\mathbb{E}\left[f(x, \xi)\right]$ simplifies as

$$\mathbb{E}\left[f(x, \xi)\right] = \sum_{i=1}^{N} \mathbb{E}\big[f(x, \xi) \,\big|\, \xi = i\big]\, \mathbb{P}(\xi = i) = \sum_{i=1}^{N} f(x, i) \cdot \frac{1}{N} =: \frac{1}{N} \sum_{i=1}^{N} f_i(x),$$

where $f_i = f(\cdot, i)$, $\forall i = 1, \ldots, N$. We recover the formulation (1.1). As for the second formulation (1.3), we can note that

$$\int_\Omega h(x,y)\, d\mu(y) = \mathbb{E}\big[h(x,\xi)\big], \qquad \text{where } \xi \sim \mu.$$

From now on, we will therefore assume that our minimization problem is of the form (1.4)

$$\min_{x\in\Theta} F(x) = \min_{x\in\Theta} \mathbb{E} f(x,\xi).$$

The challenge of such minimization problems is that the law $\mathbb{P}_\xi$ of ξ may not be known and that we cannot necessarily evaluate F nor its gradient. However, we will make the key assumption that we can *sample* from $\mathbb{P}_\xi$, meaning that we can generate k i.i.d. copies $\xi_1, \ldots, \xi_k$ with distribution $\mathbb{P}_\xi$, *or* that we are at least given k i.i.d. observations $\xi_1, \ldots, \xi_k$ with distribution $\mathbb{P}_\xi$. In the case of empirical risk minimization (1.2), for instance, it is straightforward to generate $\xi \sim \mathrm{Unif}(\{1, \ldots, N\})$. In the second part of the course, we will cover some sampling techniques, which allow one to obtain such sequences $\xi_1, \ldots, \xi_k$ with probability distribution $\mathbb{P}_\xi$ under certain assumptions. Note that the assumption of i.i.d. samples (ξ_k) imposes in particular that this sequence does not depend on the optimization variable x.
This kind of optimization problem is ubiquitous in machine learning. Let us complement our above motivation examples with the following exercise in the setting of logistic regression.

Exercise (Maximum likelihood estimator for logistic regression).

We consider a classification problem defined by observations $(x_i, y_i)_{1\le i\le n}$ where $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$ for all i. We propose the following linear model for generating the data. Each observation is supposed to be independent and there exists a vector $w \in \mathbb{R}^p$ and $w_0 \in \mathbb{R}$ such that for all $i, (y_i, x_i)$ is a realization of the random variable (Y, X) whose law $\mathcal{D}$ satisfies

$$\mathbb{P}_{w,w_0}(Y = 1 \mid X) = \frac{\exp\left(X^\top w + w_0\right)}{1 + \exp\left(X^\top w + w_0\right)}$$

1. Show that $\forall i \in \{1, \ldots, n\}, \mathbb{P}\left(Y_i = y_i \mid x_i\right) = \frac{1}{1+\exp\left(-y_i\left(x_i^\top w + w_0\right)\right)}$.
2. Show that the maximum likelihood estimator is

$$(\hat{w}, \hat{w}_0) = \arg\min_{w,w_0} \sum_{i=1}^n \log\left(1 + \exp\left(-y_i\left(x_i^\top w + w_0\right)\right)\right)$$

3. Denote $f\left(w, w_0\right) = \sum_{i=1}^n \log\left(1 + \exp\left(-y_i\left(x_i^\top w + w_0\right)\right)\right)$. Compute $\nabla f\left(w, w_0\right)$.

In the exercise, we have $\xi_i = (x_i, y_i)$. Since we have n observations, it is possible to evaluate the objective function. However, when n is large, say millions or billions, this can be a tedious task.

1.1 Reminders on convexity and gradient-Lipschitzness

1.1.1 Convexity

We recall that a function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be *convex* if it satisfies

$$\forall x, y \in \mathbb{R}^d,\ \forall t \in [0,1] : \quad f((1-t)x + ty) \le (1-t)f(x) + tf(y).$$

Moreover, f is said to be μ-strongly convex (for some $\mu \geq 0$) if it satisfies

$$\forall x, y \in \mathbb{R}^d, \forall t \in [0,1]: \quad f((1-t)x + ty) \leq (1-t)f(x) + tf(y) - \frac{\mu\, t(1-t)}{2}\big\|x-y\big\|^2.$$

Throughout the course, we will consistently make use of the following properties

Proposition 1. *If f is convex and differentiable, then for any $a, b \in \mathbb{R}^d$, we have*

$$f(b) \geq f(a) + \big\langle \nabla f(a), b-a \big\rangle.$$

Moreover, if f is μ-strongly convex, then for any $a, b \in \mathbb{R}^d$, we have

$$f(b) \geq f(a) + \big\langle \nabla f(a), b-a \big\rangle + \frac{\mu}{2}\|a-b\|^2. \tag{1.5}$$

An important consequence of μ-strong convexity appears when taking $a = x^* \in \arg\min f$ in (1.5). Indeed, by injecting $\nabla f(x^*) = 0$, we obtain that for any $x \in \mathbb{R}^d$:

$$f(x) - f(x^*) \geq \frac{\mu}{2}\|x - x^*\|^2.$$

In other words, any bound on the behavior of $f(x) - f(x^*)$ automatically translates into a bound on $\|x - x^*\|^2$.

1.1.2 Gradient-Lipschitzness

We recall that a function $g : \mathbb{R}^d \to \mathbb{R}$ is said to be L-Lipschitz for some $L > 0$ if

$$\forall x, y \in \mathbb{R}^d: \quad \big|g(x) - g(y)\big| \leq L\|x-y\|.$$

In particular, f has an L-Lipschitz gradient if for any $x, y \in \mathbb{R}^d$, we have $\big|\nabla f(x) - \nabla f(y)\big| \leq L\|x-y\|$. In this case, we say that f is *L-smooth.* We will often use the following theorem

Proposition 2 (Taylor-Lagrange)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function with L-Lipschitz gradient. Then, for any $x, y \in \mathbb{R}^d$ we have*

$$\big|f(y) - f(x) - \langle \nabla f(x), y-x \rangle\big| \leq \frac{L}{2}\|x-y\|^2.$$

Proof (Taylor-Lagrange theorem). We have for any $x, y \in \mathbb{R}^d$:

$$\begin{aligned}
\big|f(y) - f(x) - \langle \nabla f(x), y-x \rangle\big| &= \left|\int_0^1 \big\langle \nabla f\big(t(y-x)+x\big), y-x \big\rangle dt - \langle \nabla f(x), y-x \rangle\right| \\
&= \left|\int_0^1 \Big\langle \nabla f\big(t(y-x)+x\big) - \nabla f(x),\ y-x \Big\rangle\, dt\right| \\
&\leq \int_0^1 \left|\Big\langle \nabla f\big(t(y-x)+x\big) - \nabla f(x),\ y-x \Big\rangle\right|\, dt \qquad \text{(triangle inequality)} \\
&\leq \int_0^1 \Big\|\nabla f\big(t(y-x)+x\big) - \nabla f(x)\Big\|\, \big\|\, y-x\big\|\, dt \qquad \text{(Cauchy-Schwarz)} \\
&\leq \int_0^1 L\, \big\|t(y-x)+x-x\big\|\, \big\|\, y-x\big\|\, dt \qquad (L\text{-Lipschitzness of } \nabla f) \\
&= L\|y-x\|^2 \int_0^1 t\, dt \ = \ \frac{L}{2}\|y-x\|^2.
\end{aligned}$$

□

Similarly, when applying the Taylor-Lagrange theorem with a minimizer $x^* \in \arg\min f$ of an L-smooth function f, we obtain that for any $x \in \mathbb{R}^d$:

$$f(x) - f(x^*) \leq \frac{L}{2}\|x - x^*\|^2.$$

This time, any bound on the behavior of $\|x - x^*\|^2$ automatically translates into a bound on $f(x) - f(x^*)$.

In conclusion, when f is both μ-strongly convex and L-smooth (i.e. ∇f is L-Lipschitz), the following relation holds

$$\frac{\mu}{2}\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{L}{2}\|x - x^*\|^2,$$

imposing that $f(x) - f(x^*)$ should have the same behavior as $\|x - x^*\|^2$, up to constants.

1.2 Gradient descent

The gradient descent method is among the most basic methods for minimizing a differentiable function f. However, it requires full access to the gradient of the function and is therefore not well-suited for our purposes. Still, it will constitute the basis for further developments later on in the course. The algorithm iteratively defines a sequence $(x_k)_{k\in\mathbb{N}}$ of points in $\mathbb{R}^n$ obtained by induction from $x_0 \in \mathbb{R}^n$ according to the following rule

Algorithm 1: Gradient Descent

```
    1. Pick $x_0 \in \mathbb{R}^d$
    2. Until termination condition, iterate
           $x_{k+1} = x_k - \gamma_k \nabla f(x_k).$
  where for all $k, \gamma_k$ is a positive coefficient.
```

The first part of the course will be focused on Stochastic Gradient Descent (SGD), which is a variant of the Gradient Descent (GD) algorithm. To prove convergence guarantees for SGD, we will rely on several techniques appearing in convergence proofs for GD, which we recall here for the sake of completeness.

Theorem 1 (Gradient Descent on convex/strongly convex functions)**.** *Let f be a convex differentiable function that has a minimizer x^* and whose gradient is L-Lipschitz continuous. The gradient method with constant step size $\gamma_k = \frac{1}{L}$ satisfies*

$$f(x_k) - f\left(x^*\right) \leq \frac{L\left\|x_0 - x^*\right\|^2}{2k}.$$

If moreover f is μ-strongly convex, then

$$
\begin{aligned}
f\left(x_k\right)-f\left(x^*\right) &\leq\left(1-\frac{\mu}{L}\right)^k\left(f\left(x_0\right)-f\left(x^*\right)+\frac{L}{2}\left\|x_0-x^*\right\|^2\right) \\
\left\|x_k-x^*\right\|^2 &\leq \frac{2}{\mu}\left(1-\frac{\mu}{L}\right)^k\left(f\left(x_0\right)-f\left(x^*\right)+\frac{L}{2}\left\|x_0-x^*\right\|^2\right).
\end{aligned}
$$

Proof of Theorem 1. As ∇f is L-Lipschitz, the Taylor-Lagrange inequality yields

$$
\begin{aligned}
f\left(x_{k+1}\right) &\leq f\left(x_k\right)+\left\langle\nabla f\left(x_k\right), x_{k+1}-x_k\right\rangle+\frac{L}{2}\left\|x_{k+1}-x_k\right\|^2 \\
&=f\left(x_k\right)+\left(-\gamma+\frac{L \gamma^2}{2}\right)\left\|\nabla f\left(x_k\right)\right\|^2 .
\end{aligned}
$$

As soon as $\gamma<2 / L$, we get $f\left(x_{k+1}\right) \leq f\left(x_k\right)$. Next, we have for any x,

$$
\left\langle\nabla f\left(x_k\right), x_{k+1}-x_k\right\rangle+\frac{L}{2}\left\|x_{k+1}-x_k\right\|^2=\left\langle\nabla f\left(x_k\right), x-x_k\right\rangle+\frac{L}{2}\left\|x-x_k\right\|^2-\frac{L}{2}\left\|x-x_{k+1}\right\|^2 . \tag{1.6}
$$

To see this, we can note that

$$
\begin{aligned}
&\left\langle\nabla f\left(x_k\right), x-x_k\right\rangle+\frac{L}{2}\left\|x-x_k\right\|^2-\frac{L}{2}\left\|x-x_k+x_k-x_{k+1}\right\|^2 \\
&=\left\langle\nabla f\left(x_k\right), x-x_k\right\rangle+\frac{L}{2}\left(\cancel{\left\|x-x_k\right\|^2}-\cancel{\left\|x-x_k\right\|^2}-\left\|x_k-x_{k+1}\right\|^2-2\left\langle x-x_k, \underbrace{x_k-x_{k+1}}_{=+\gamma \nabla f\left(x_k\right)}\right\rangle\right) \\
&=\cancel{\left\langle\nabla f\left(x_k\right), x-x_k\right\rangle}-\frac{L}{2}\left\|x_k-x_{k+1}\right\|^2-\frac{L}{2} 2 \gamma \cancel{\left\langle x-x_k, \nabla f\left(x_k\right)\right\rangle} \quad \text{by choosing } \gamma=\frac{1}{L} \\
&=-\frac{L}{2}\left\|x_k-x_{k+1}\right\|^2=-L\left\|x_k-x_{k+1}\right\|^2+\frac{L}{2}\left\|x_k-x_{k+1}\right\|^2 \\
&=\left\langle\nabla f\left(x_k\right), x_{k+1}-x_k\right\rangle+\frac{L}{2}\left\|x_{k+1}-x_k\right\|^2 .
\end{aligned}
$$

In the last line, we used that $\left\langle\nabla f\left(x_k\right), x_{k+1}-x_k\right\rangle=\left\langle-L(x_{k+1}-x_k), x_{k+1}-x_k\right\rangle$ since by definition, we have $x_{k+1}=x_k-\frac{1}{L} \nabla f(x_k)$.
Hence, using equation (1.6) and the convexity of f,

$$
\begin{aligned}
f\left(x_{k+1}\right) &\leq f\left(x_k\right)+\left\langle\nabla f\left(x_k\right), x_{k+1}-x_k\right\rangle+\frac{L}{2}\left\|x_{k+1}-x_k\right\|^2 \\
&=f\left(x_k\right)+\left\langle\nabla f\left(x_k\right), x^*-x_k\right\rangle+\frac{L}{2}\left\|x^*-x_k\right\|^2-\frac{L}{2}\left\|x^*-x_{k+1}\right\|^2 \\
&\leq f\left(x^*\right)+\frac{L}{2}\left\|x^*-x_k\right\|^2-\frac{L}{2}\left\|x^*-x_{k+1}\right\|^2 .
\end{aligned} \tag{1.7}
$$

We sum this inequality for $k \in\{0, \ldots, K-1\}$:

$$
\sum_{k=1}^K\left(f\left(x_k\right)-f\left(x^*\right)\right) \leq \frac{L}{2}\left\|x^*-x_0\right\|^2-\frac{L}{2}\left\|x^*-x_K\right\|^2 \leq \frac{L}{2}\left\|x^*-x_0\right\|^2 .
$$

Finally, since $f\left(x_{k+1}\right) \leq f\left(x_k\right)$ for all k,

$$f\left(x_K\right) - f\left(x^*\right) \leq \frac{L}{2K}\left\|x^* - x_0\right\|^2.$$

We now consider the case of strongly convex functions. By strong convexity, we have $f\left(x^*\right) \geq f\left(x_k\right) + \left\langle \nabla f\left(x_k\right), x^* - x_k\right\rangle + \frac{\mu}{2}\left\|x_k - x^*\right\|^2$. Therefore, coming back to (1.7), we get

$$\begin{aligned} f\left(x_{k+1}\right) &\leq f\left(x_k\right) + \left\langle \nabla f\left(x_k\right), x^* - x_k\right\rangle + \frac{L}{2}\left\|x^* - x_k\right\|^2 - \frac{L}{2}\left\|x^* - x_{k+1}\right\|^2 \\ &\leq f(x^*) - \frac{\mu}{2}\|x^* - x_k\|^2 + \frac{L}{2}\left\|x^* - x_k\right\|^2 - \frac{L}{2}\left\|x^* - x_{k+1}\right\|^2, \end{aligned}$$

hence

$$\begin{aligned} f\left(x_{k+1}\right) - f\left(x^*\right) + \frac{L}{2}\left\|x^* - x_{k+1}\right\|^2 &\leq \frac{L-\mu}{2}\left\|x^* - x_k\right\|^2 \\ &\leq \left(1 - \frac{\mu}{L}\right)\left(f\left(x_k\right) - f\left(x^*\right) + \frac{L}{2}\left\|x^* - x_k\right\|^2\right). \end{aligned}$$

The final result comes by iterating this inequality:

$$f\left(x_k\right) - f\left(x^*\right) + \frac{L}{2}\left\|x^* - x_k\right\|^2 \leq \left(1 - \frac{\mu}{L}\right)^k\left(f\left(x_0\right) - f\left(x^*\right) + \frac{L}{2}\left\|x^* - x_0\right\|^2\right).$$

We further obtain the last part of the theorem by using μ-strong convexity of f:

$$\begin{aligned} &\frac{\mu}{2}\|x_k - x^*\|^2 \leq f(x_k) - f(x^*) - \langle \underbrace{\nabla f(x^*)}_{=0}, x_k - x^*\rangle \\ \implies \quad &\|x_k - x^*\|^2 \leq \frac{2}{\mu}\left(1 - \frac{\mu}{L}\right)^k\left(f\left(x_0\right) - f\left(x^*\right) + \frac{L}{2}\left\|x_0 - x^*\right\|^2\right). \end{aligned}$$

□

1.3 How to compute gradients?

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. To run the GD algorithm, we would like to compute its gradient. By definition, $\nabla f(x)$ is the unique vector of $\mathbb{R}^n$ such that

$$f(x+h) = f(x) + \langle \nabla f(x), h\rangle + o(\|h\|),$$

for any norm $\|\cdot\|$ over $\mathbb{R}^n$.
There are several ways to compute a gradient. All should give the same result.

1.3.1 Using partial derivatives

We know that the gradient is the vector of all the partial derivatives. Hence, we can compute $\frac{\partial f}{\partial x_i}(x)$ for all i and reconstruct the vector.
Example. Let us consider the function $f(x) = \|Ax - b\|^2$ where $A \in \mathbb{R}^{m\times n}$. We can write

$$f(x) = \sum_{j=1}^{m}\left(\sum_{i=1}^{n} A_{j,i}x_i - b_j\right)^2$$

and so

$$\frac{\partial f}{\partial x_k}(x) = 2\sum_{j=1}^{m} A_{j,k}\left(\sum_{i=1}^{n} A_{j,i}x_i - b_j\right).$$

We recognize the components of the vector

$$\nabla f(x) = 2A^\top(Ax - b).$$

1.3.2 Using the definition

We compute $f(x+h)$ and isolate $f(x)$, a linear term in h and a negligible term.
Example. We consider $f(x) = \|Ax - b\|^2$.

$$\begin{aligned} f(x+h) = \|A(x+h) - b\|^2 &= \|Ax - b\|^2 + 2\langle Ax - b, Ah\rangle + \|Ah\|^2 \\ &= f(x) + 2\left\langle A^\top(Ax - b), h\right\rangle + o(h) \end{aligned}$$

thus, $\nabla f(x) = 2A^\top(Ax - b)$.

1.3.3 Using the chain rule

Let $g : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^p$. The chain rule states that the Jacobian matrix of the function $f \circ g$ at x is given by

$$J_{f\circ g}(x) = J_f(g(x)) \times J_g(x).$$

We recall that

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix}$$

is the unique linear map such that

$$g(x+h) = g(x) + J_g(x)h + o(h).$$

The chain rule allows us to combine simple functions to obtain complex functions. It is at the basis of automatic differentiation and the resolution of neural network models.
When $f : \mathbb{R}^m \to \mathbb{R}$ and $g(x) = Ax$ where A is a $m \times n$ matrix, the formula simplifies as

$$\nabla(f \circ A)(x) = A^\top \nabla f(Ax).$$

Example. We consider $f(x) = \|Ax - b\|^2$.
Let us remark that $f(x) = h(Ax)$ where $h(y) = \|y - b\|^2$.
Since $h(y+h) = \|y + h - b\|^2 = \|y - b\|^2 + 2\langle y - b, h\rangle + \|h\|^2$, we know that $\nabla h(y) = 2(y - b)$.
Using the chain rule, we get $\nabla f(x) = \nabla(h \circ A)(x) = A^\top \nabla h(Ax) = 2A^\top(Ax - b)$.

Chapter 2

Stochastic gradient

2.1 Algorithm

We recall that we are considering the optimization problem

$$\min_{x\in\Theta} F(x) = \min_{x\in\Theta} \mathbb{E}[f(x,\xi)],$$

and assume first that $\Theta = \mathbb{R}^d$. As previously mentioned, we assume we can *sample* from $\mathbb{P}_\xi$, meaning that we can generate k i.i.d. random variables $\xi_1, \dots, \xi_K \sim \mathbb{P}_\xi$ for any $K > 0$. The second part of the course (on sampling) will address methods for obtaining such sequences of random variables.

Since Gradient Descent might fail in such settings, as detailed in the previous section, new algorithms are required to solve the above minimization problem. This motivates us to introduce the Stochastic Gradient Descent algorithm (SGD). The idea is to modify the Gradient Descent algorithm by replacing a full gradient $\nabla F(x_k)$ at each step with a stochastic gradient $\nabla_x f(x_k, \xi_{k+1})$ for some ξ_{k+1} independent of x_k. Indeed, under mild conditions (see Proposition 20 in the Appendix), we may write

$$\nabla F(x) = \nabla \mathbb{E}[f(x,\xi)] = \mathbb{E}[\nabla f(x,\xi)], \quad \forall x \in \mathbb{R}^d.$$

This equality certifies that each stochastic gradient $\nabla_x f(x_k, \xi_{k+1})$ is centered around the optimal direction $\nabla F(x_k)$ that would be taken if we were using GD. While each update is now stochastic, thus less precise than a Gradient Descent step, computing $\nabla_x f(x_k, \xi_{k+1})$ can be considerably faster than computing $\nabla F(x)$. In practice, SGD may therefore lead to significant runtime acceleration compared to classical Gradient Descent.

Given a sequence of step sizes γ_k, the algorithm is given by

```
Algorithm 2: Stochastic Gradient Descent (SGD)
    1. Start from an initial guess $x_0 \in \mathbb{R}^d$
    2. Until termination condition, iterate
            Generate $\xi_k \sim \mathbb{P}_\xi$, independent of $(\xi_1, \dots, \xi_{k-1})$
            $x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$
```

where $\nabla f(x_k, \xi_{k+1})$ is the gradient of $\big(x \mapsto f(x, \xi_{k+1})\big)$ at x_k.

Strictly speaking, this is not a descent algorithm: $F(x_k)$ can go up and down, but decreases "on average". Moreover, the appropriate step size sequence $(\gamma_k)_k$ for SGD may not be the same as for GD.

Remark: If $\big(x \mapsto f\,(x, \xi_{k+1})\big)$ is not differentiable, one can use a subgradient of the function instead of its gradient.

An important special case is the setting of Empirical Risk Minimization (1.2). Here we are given N data points, each of which is associated with a loss function $f_i, 1 \leq i \leq N$. A typical model in machine learning consists of minimizing the empirical risk given by

$$\min_x \frac{1}{N}\sum_{i=1}^{N} f_i(x).$$

When N is large, computing this sum or its gradient can be prohibitive. We can circumvent this by running stochastic gradient descent, whose pseudo-code can be written as follows in the present special case

Algorithm 3: SGD for Empirical Risk Minimization

```
1. Pick $x_0 \in \mathbb{R}^d$
2. Until termination, iterate:
        $i_k \sim \mathrm{Unif}(\{1,\dots,N\})$ independent of the past
        $x_{k+1} = x_k - \gamma_k \nabla f_{i_k}(x_k)$.
```

This leads to an algorithm with very low complexity per iteration, which is often used in practice. Indeed, each iteration only requires computing *one* gradient $\nabla f_{i_k}(x_k)$ selected uniformly at random among $\nabla f_1(x_k), \dots, \nabla f_N(x_k)$, rather than N gradients. This represents a considerable save of time—at the expense of less accurate updates, since each step is now stochastic rather than deterministic, but *centered* around the correct direction:

$$\mathbb{E}\left[\nabla f_{i_k}(x_k) \,\middle|\, x_k\right] = \sum_{i=1}^{N} \mathbb{E}\left[\nabla f_{i_k}(x_k) \,\middle|\, x_k, i_k = i\right] \mathbb{P}(i_k = i) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_k) = \nabla F(x_k).$$

Example: (Least Mean Squares). We are given a random variable $\xi = (X, Y)$ where $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}$. Least Mean Squares (LMS) is a regression problem in expectation

$$\min_{w \in \mathbb{R}^n} \frac{1}{2}\mathbb{E}\left[\left(Y - X^\top w\right)^2\right]$$

Show that the stochastic gradient on this problem can be written as

$$w_{k+1} = w_k - \gamma_k \left(X_{k+1}^\top w_k - Y_{k+1}\right) X_{k+1}.$$

2.1.1 Minibatch SGD

Another important variant of SGD is the *minibatch stochastic gradient descent* algorithm. This algorithm is extremely popular for solving optimization problems of the form

$$\min_{x \in \mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} f_i(x).$$

Instead of drawing one index i_k uniformly at random, as in SGD, the minibatch SGD algorithm defines a minibatch size $m \in \{1, \dots, N\}$ and draws m distinct indices $i_k^{(1)}, \dots, i_k^{(m)}$ uniformly at random from $\{1, \dots, N\}$. The subset $\{i_k^{(1)}, \dots i_k^{(m)}\}$ is called a minibatch.

In this case, we say that $\{i_k^{(1)}, \dots i_k^{(m)}\}$ is drawn without replacement from the uniform distribution over $\{1, \dots, N\}$, and we write $\{i_k^{(1)}, \dots i_k^{(m)}\} \sim \mathrm{Unif}_m(\{1, \dots, N\})$. The pseudo-code is given below.

```
Algorithm 4: Minibatch SGD
  Input: Minibatch size $m \in \{1, \dots, N\}$
    1. Pick $x_0 \in \mathbb{R}^d$
    2. Until termination, iterate:
         Draw $\left\{i_k^{(1)}, \dots, i_k^{(m)}\right\} \sim \mathrm{Unif}_m(\{1, \dots, N\})$ independently of the past
         $x_{k+1} = x_k - \gamma_k \cdot \frac{1}{m} \sum_{j=1}^{m} \nabla f_{i_k^{(j)}}(x_k).$
```

We immediately note that the classical SGD algorithm is a special case of the minibatch SGD algorithm where $m = 1$. In contrast, the classical GD algorithm corresponds to the minibatch SGD algorithm with $m = N$.

The minibatch SGD algorithm can also be rewritten as a special case of Algorithm 2. Specifically, letting $\xi_k := \{i_k^{(1)}, \dots i_k^{(m)}\}$ and defining $f\big(x, \{i_k^{(1)}, \dots i_k^{(m)}\}\big) := \frac{1}{m} \sum_{j=1}^{m} f_{i_k^{(j)}}(x)$ for any $x \in \mathbb{R}^d$, we obtain

$$x_{k+1} = x_k - \gamma_k \nabla f(x, \xi_{k+1}).$$

In practice, this algorithm is sometimes implemented in a slightly modified way. Instead of drawing a subset of m distinct indices uniformly at random each time, one rather shuffles the set of indices $\{1, \dots, N\}$ by drawing a random permutation of $\{1, \dots, N\}$ denoted by σ. This produces a re-ordered set of indices $(\sigma(1), \dots, \sigma(N))$. The first minibatch then consists of the first m indices $(\sigma(1), \dots, \sigma(m))$, the second minibatch of the next m indices $(\sigma(m+1), \dots, \sigma(2m))$, and so on. Once all the indices have been used, the set of indices is reshuffled, and the same procedure is repeated. A single pass through the entire set of indices is referred to as an *epoch*. The pseudo-code is given below (for simplicity, we assume that N/m is an integer here).

```
Algorithm 5: Epoch-based minibatch SGD
  Input: Minibatch size $m \in \{1, \dots, N\}$
    1. Pick $x_0 \in \mathbb{R}^d$
    2. Until termination, iterate:
         Draw a random permutation $\sigma$ of $\{1, \dots, N\}$, independent of the past.
         For $\ell = 1, \dots, N/m$
             $x_{k+1} = x_k - \gamma_k \cdot \frac{1}{m} \left(\nabla f_{\sigma((\ell-1)m+1)}(x_k) + \cdots + \nabla f_{\sigma(\ell m)}(x_k)\right).$
```

The properties of minibatch SGD will be studied in the exercise and practical sessions.

2.2 Convergence

We denote by $\mathbb{E}_k$ the expectation conditional on $(\xi_1, \dots, \xi_k)$. Note that x_k is measurable with respect to $(\xi_1, \dots, \xi_k)$. In this section, we study the convergence properties of the stochastic sequence $(x_k)_k$ produced by Algorithm 2.

2.2.1 Convex objective

Theorem 2 (Rates of SGD for convex functions)**.** *Suppose that:*

1. *$x \mapsto f(x, \xi)$ is convex and differentiable for all ξ,*
2. *$\exists C > 0, \forall x \in \mathbb{R}^d : \mathbb{E}\big(\big\|\nabla_x f(x, \xi)\big\|^2\big) \le C$,*
3. *There exists $x^* \in \arg\min F$*
4. *For any $k \in \mathbb{N} : \gamma_k = \frac{\gamma^*}{\sqrt{k+1}}$ for some $\gamma^* > 0$.*

The iterates $(x_k)_k$ of SGD defined in Algorithm 2 satisfy the convergence guarantee

$$\mathbb{E}\left[F\left(\bar{x}_k^\gamma\right) - F\left(x^*\right)\right] \le \frac{\|x_0 - x^*\|^2 + C\gamma^{*2}(1 + \log(k+1))}{4\gamma^*(\sqrt{k+2} - 1)} = O\left(\frac{\log(k)}{\sqrt{k}}\right),$$

where $\bar{x}_k^\gamma = \frac{\sum_{l=0}^k \gamma_l x_l}{\sum_{j=0}^k \gamma_j}$ is a convex combination of all previous iterates.

Before proving the theorem, some remarks are in order.

- We need convexity of $x \mapsto f(x, \xi)$ for each $\xi \in \Xi$. This implies that F defined by $F(x) = \mathbb{E}\big[f(x, \xi)\big]$ is convex in x (why?).
- Condition 2 imposes that F be $\sqrt{C}$-Lipschitz. Indeed,

  $$\|\nabla F(x)\| = \|\nabla \mathbb{E} f(x, \xi)\| = \|\mathbb{E} \nabla f(x, \xi)\| \le \left(\mathbb{E}\left[\|\nabla f(x, \xi)\|^2\right]\right)^{1/2} \le \sqrt{C}.$$

  Note that in the second equality, we have used Proposition 20 in the Appendix to swap the gradient and the expectation.
- For convex functions, the convergence rate of SGD is $O\big(\frac{\log(k)}{\sqrt{k}}\big)$ if the step sizes satisfy $\gamma_k = \frac{1}{\sqrt{k+1}}$.
- This rate is *not* a guarantee on $F(x_k) - F(x^*)$, but rather on the *expectation* $\mathbb{E}\left[F(\bar{x}_k^\gamma) - F(x^*)\right]$.
- If F is just convex, controlling $\mathbb{E}\left[F(\bar{x}_k^\gamma) - F(x^*)\right]$ does not give any guarantee on $\|\bar{x}_k^\gamma - x^*\|^2$ (why?).

Proof of Theorem 2. We can first check the following property of $\mathbb{E}[\nabla f(x, \xi)]$ for all x. Indeed,

$$\begin{aligned} &f(y, \xi) \ge f(x, \xi) + \langle \nabla f(x, \xi), y - x\rangle \\ &F(y) = \mathbb{E}[f(y, \xi)] \ge \mathbb{E}[f(x, \xi)] + \mathbb{E}[\langle \nabla f(x, \xi), y - x\rangle] = F(x) + \langle \mathbb{E}[\nabla f(x, \xi)], y - x\rangle. \end{aligned} \tag{2.1}$$

Now, we apply $\mathbb{E}_k$, defined as the expectation conditional on $(\xi_1, \ldots, \xi_k)$.

$$\begin{aligned}
\mathbb{E}_k\left[\left\|x_{k+1}-x^*\right\|^2\right] &= \mathbb{E}_k\left[\left\|x_k-x^*\right\|^2 + 2\left\langle x_{k+1}-x_k, x_k-x^*\right\rangle + \left\|x_{k+1}-x_k\right\|^2\right] \\
&= \left\|x_k-x^*\right\|^2 - 2\gamma_k\left\langle \mathbb{E}_k\left[\nabla f\left(x_k,\xi_{k+1}\right)\right], x_k-x^*\right\rangle + \gamma_k^2\mathbb{E}_k\left[\left\|\nabla f\left(x_k,\xi_{k+1}\right)\right\|^2\right] \\
&\leq \left\|x_k-x^*\right\|^2 + 2\gamma_k\left\langle \mathbb{E}_k\left[\nabla f\left(x_k,\xi_{k+1}\right)\right], x^*-x_k\right\rangle + \gamma_k^2 C \\
&\leq \left\|x_k-x^*\right\|^2 + 2\gamma_k\Big(F\left(x^*\right)-F\left(x_k\right)\Big) + \gamma_k^2 C.
\end{aligned}$$

We reorganize and apply total expectation:

$$\mathbb{E}\left[\gamma_k\Big(F\left(x_k\right)-F\left(x^*\right)\Big)\right] \leq -\frac{1}{2}\mathbb{E}\left[\left\|x_{k+1}-x^*\right\|^2\right] + \frac{1}{2}\mathbb{E}\left[\left\|x_k-x^*\right\|^2\right] + \frac{\gamma_k^2 C}{2}.$$

We sum for l between 0 and k :

$$\begin{aligned}
\mathbb{E}\left[\sum_{l=0}^{k}\gamma_l\Big(F\left(x_l\right)-F\left(x^*\right)\Big)\right] &\leq -\frac{1}{2}\mathbb{E}\left[\left\|x_{k+1}-x^*\right\|^2\right] + \frac{1}{2}\mathbb{E}\left[\left\|x_0-x^*\right\|^2\right] + \sum_{l=0}^{k}\frac{\gamma_l^2 C}{2} \\
&\leq \frac{1}{2}\mathbb{E}\left[\left\|x_0-x^*\right\|^2\right] + \sum_{l=0}^{k}\frac{\gamma_l^2 C}{2}.
\end{aligned}$$

Dividing by $\sum_{j=0}^{k}\gamma_j$, the result follows by convexity of F:

$$\mathbb{E}\left[F\left(\bar{x}_k^{\gamma}\right)-F\left(x^*\right)\right] \leq \frac{1}{\sum_{j=0}^{k}\gamma_j}\mathbb{E}\left[\sum_{l=0}^{k}\gamma_l\Big(F\left(x_l\right)-F\left(x^*\right)\Big)\right] \leq \frac{\mathbb{E}\left[\left\|x_0-x^*\right\|^2\right] + C\sum_{l=0}^{k}\gamma_l^2}{2\sum_{j=0}^{k}\gamma_j}.$$

Note that we get convergence of the algorithm as soon as $\sum_{j=1}^{k}\gamma_j \to +\infty$ and $\frac{\sum_{l=1}^{k}\gamma_l^2}{\sum_{j=1}^{k}\gamma_j} \to 0$. The fastest decay of $\frac{\sum_{l=1}^{k}\gamma_l^2}{\sum_{j=1}^{k}\gamma_j}$ turns out to be obtained when γ_k is proportional to $\frac{\gamma^*}{\sqrt{k+1}}$ (why?). Now, replacing γ_k with its value $\frac{\gamma^*}{\sqrt{k+1}}$ yields

$$\begin{aligned}
&\sum_{j=0}^{k}\gamma_j = \sum_{j=0}^{k}\frac{\gamma_0}{\sqrt{j+1}} \geq \sum_{j=0}^{k}\int_{j+1}^{j+2}\frac{\gamma_0}{\sqrt{t}}dt = \int_{1}^{k+2}\frac{\gamma_0}{\sqrt{t}}dt = \gamma_0\left[2\sqrt{t}\right]_1^{k+2} = 2\gamma_0\left(\sqrt{k+2}-1\right) \\
&\sum_{j=0}^{k}\gamma_j^2 = \sum_{j=0}^{k}\frac{\gamma_0^2}{j+1} \leq \gamma_0^2 + \sum_{j=1}^{k}\int_{j}^{j+1}\frac{\gamma_0^2}{t}dt = \gamma_0^2 + \int_{1}^{k+1}\frac{\gamma_0^2}{t}dt = \gamma_0^2(1+\ln(k+1)).
\end{aligned}$$

To compare sums and integrals in the two lines above, we used the fact that the functions $t \mapsto 1/t$ and $t \mapsto 1/\sqrt{t}$ are non-increasing. □

Remark: If we know in advance the number of iterations K to be performed, we can set a constant step size $\gamma_k = \frac{a}{\sqrt{K}}$ and obtain a guarantee $\mathbb{E}\left[F\left(\bar{x}_K^{\gamma}\right)-F\left(x^*\right)\right] \in O\left(\frac{1}{\sqrt{K}}\right)$ (why?).

2.2.2 Strongly convex objectives

When F is μ-strongly convex, we can show that taking $\gamma_k = \frac{a}{\mu(k+b)}$ gives an improved rate $\mathbb{E}\left[F\left(x_k\right) - F\left(x^*\right)\right] = O\left(\frac{1}{k}\right)$.

Theorem 3. *Suppose that:*

1. $x \mapsto f(x, \xi)$ *is convex and differentiable for all* $\xi \in \Xi$*,*
2. F *is* μ*-strongly convex and its gradient is* L*-Lipschitz,*
3. $\exists C > 0, \forall x \in \mathbb{R}^d : \mathbb{E}\left(\left\|\nabla_x f(x, \xi)\right\|^2\right) \leq C$*,*
4. $x^* \in \arg\min F$ *is attained,*
5. *For any* $k \in \mathbb{N} : \gamma_k = \frac{1}{\mu(k+1)}$.

The iterates of the stochastic gradient algorithm $x_{k+1} = x_k - \gamma_k \nabla f\left(x_k, \xi_{k+1}\right)$ *satisfy the convergence guarantee*

$$\mathbb{E}\left[\left\|x_k - x^*\right\|^2\right] \leq \frac{C}{\mu^2 k},$$

$$\mathbb{E}\left[F\left(x_k\right) - F\left(x^*\right)\right] \leq \frac{LC}{2\mu^2 k},$$

for any $k \geq 2$.

Before proving the theorem, some remarks are in order.

1. Compared to the purely convex case, this theorem only requires strong convexity of F (not of each of the $x \mapsto f(x, \xi)$ for $\xi \in \Xi$), and that ∇F be L-Lipschitz.
2. The step-size sequence now decays as $\gamma_k = O(\frac{1}{\mu k})$. (In the convex case, the decay was $\frac{1}{\sqrt{k}}$).
3. Convergence rate: $O(\frac{C}{\mu^2 k})$. (In the convex case it was $O(\frac{\log(k)}{\sqrt{k}})$).
4. We obtain guarantees on both $\mathbb{E}\left[F(x_k) - F(x^*)\right]$ and $\|x_k - x^*\|^2$ due to the strong convexity of F and Lipschitzness of ∇F.
5. The result above is valid for any $k \geq 2$. For $k = 1$, it turns out that the initial value must satisfy $\|x_1 - x^*\|^2 \leq C/\mu^2$ (see Proposition 3 below). This allows us to deduce the rates

$$\mathbb{E}\left[\left\|x_k - x^*\right\|^2\right] \leq \frac{16C}{\mu^2 k}, \quad \text{and} \quad \mathbb{E}\left[F(x_k) - F(x^*)\right] \leq \frac{8LC}{\mu^2 k},$$

for any $k \geq 1$.

Proposition 3. *Let* $F : \Theta \to \mathbb{R}$ *be* $\sqrt{C}$*-Lipschitz and* μ*-strongly convex. Then the diameter of* Θ *must satisfy*

$$\operatorname{diam}(\Theta) \overset{\text{Def}}{=} \max\{\|x - y\| : x, y \in \Theta\} \leq 4\sqrt{C}/\mu.$$

Proof of Proposition 3. Suppose for a contradiction that $\exists x, y \in \Theta$ such that $\|x - y\| >$

$4\sqrt{C}/\mu$. Then by strong convexity of F, we would have

$$\begin{aligned} F(x)-F(y) &\geq \langle \nabla F(y), x-y\rangle + \frac{\mu}{2}\|x-y\|^2 \\ &> -\big\|\nabla F(y)\big\|\big\|x-y\big\| + \frac{\mu}{2}\|x-y\|\frac{4\sqrt{C}}{\mu} \quad \text{(Cauchy-Schwarz)} \\ &\geq -\sqrt{C}\|x-y\| + 2\sqrt{C}\|x-y\| \\ &= \sqrt{C}\|x-y\|, \end{aligned}$$

which contradicts the $\sqrt{C}$-Lipschitzness of F. □

Proof of Theorem 3. Note that in this proof, we are allowed to swap gradients and expectations everywhere and write $\nabla \mathbb{E} f(x,\xi) = \mathbb{E}\nabla f(x,\xi)$ thanks to Proposition 20 in the Appendix. Compared to the convex case, we replace the inequality $F(y) \geq F(x) + \langle \mathbb{E}[\nabla f(x,\xi)], y-x\rangle$ from equation (2.1) by the stronger one

$$\begin{aligned} F(y) &\geq F(x) + \langle \nabla F(x), y-x\rangle + \frac{\mu}{2}\|y-x\|^2 && (2.2) \\ &= F(x) + \langle \mathbb{E}[\nabla f(x,\xi)], y-x\rangle + \frac{\mu}{2}\|y-x\|^2 \quad \text{by Proposition 20.} && (2.3) \end{aligned}$$

By equation (2.3), we obtain

$$\begin{aligned} \mathbb{E}_k\left[\left\|x_{k+1}-x^*\right\|^2\right] &= \left\|x_k-x^*\right\|^2 - 2\gamma_k\left\langle \mathbb{E}_k\left[\nabla f\left(x_k,\xi_{k+1}\right)\right], x_k-x^*\right\rangle + \gamma_k^2\mathbb{E}_k\left[\left\|\nabla f\left(x_k,\xi_{k+1}\right)\right\|^2\right] \\ &\leq \left(1-\mu\gamma_k\right)\left\|x_k-x^*\right\|^2 + 2\gamma_k\Big(F\left(x^*\right)-F\left(x_k\right)\Big) + \gamma_k^2 C. \end{aligned}$$

By equation (2.2), we obtain $F\left(x^*\right)-F\left(x_k\right) \leq -\frac{\mu}{2}\left\|x_k-x^*\right\|^2$, which further ensures that

$$\begin{aligned} \mathbb{E}_k\left[\left\|x_{k+1}-x^*\right\|^2\right] &\leq \left(1-2\mu\gamma_k\right)\left\|x_k-x^*\right\|^2 + \gamma_k^2 C \\ &= \left(1-\frac{2}{k+1}\right)\|x_k-x^*\|^2 + \frac{C}{\mu^2(k+1)^2}. \end{aligned}$$

Taking expectations on both sides, and using the law of total expectation, we obtain

$$\begin{aligned} \mathbb{E}\left[\left\|x_{k+1}-x^*\right\|^2\right] &\leq \left(\frac{k-1}{k+1}\right)\mathbb{E}\left[\left\|x_k-x^*\right\|^2\right] + \frac{C}{\mu^2(k+1)^2} \\ \text{i.e.}\quad k(k+1)\,\mathbb{E}\left[\left\|x_{k+1}-x^*\right\|^2\right] &\leq (k-1)k\,\mathbb{E}\left[\left\|x_k-x^*\right\|^2\right] + \frac{Ck}{\mu^2(k+1)}, \end{aligned}$$

where the second line follows from the first line by multiplying both sides by $(k+1)k$. We now define the sequence $u_k = (k-1)k\,\mathbb{E}\left[\|x_k-x^*\|^2\right]$ for any $k \in \mathbb{N}$. The previous equation can then be rewritten as

$$\begin{aligned} & u_{k+1} \leq u_k + \frac{Ck}{\mu^2(k+1)} \leq u_k + \frac{C}{\mu^2} \\ \implies \quad & u_{k+1} - u_k \leq \frac{C}{\mu^2}. \end{aligned}$$

Summing these inequalities, we obtain

$$\sum_{j=1}^{k-1} u_{j+1} - u_j \leq \sum_{j=1}^{k-1} \frac{C}{\mu^2} = \frac{C(k-1)}{\mu^2}$$
$$\text{i.e.} \quad u_k - u_1 \leq \frac{C(k-1)}{\mu^2}.$$

Recalling that $u_k = k(k-1)\mathbb{E}\left[\|x_k - x^*\|^2\right]$, which implies that $u_1 = 0$, we obtain

$$k(k-1)\mathbb{E}\left[\|x_k - x^*\|^2\right] \leq \frac{C(k-1)}{\mu^2}$$
$$\text{hence} \quad \mathbb{E}\left[\|x_k - x^*\|^2\right] \leq \frac{C}{\mu^2 k}$$

for any $k \geq 2$. For $k = 1$, the cancellation above by $k-1$ does not go through since $k-1=0$, but we have $\|x_1 - x^*\|^2 \leq 16\frac{C}{\mu^2}$ by Proposition 3. It follows that $\mathbb{E}\left[\|x_1 - x^*\|^2\right] \leq \frac{16C}{\mu^2}$.
The second part of the theorem follows from the Taylor-Lagrange inequality

$$F(x_k) - F\left(x^*\right) \leq \left\langle \nabla F\left(x^*\right), x_k - x^* \right\rangle + \frac{L}{2}\left\|x_k - x^*\right\|^2 = \frac{L}{2}\left\|x_k - x^*\right\|^2,$$

which implies $\mathbb{E}\left[F(x_k) - F(x^*)\right] \leq \frac{LC}{2\mu^2 k}$, as claimed.

□

2.3 Proximal stochastic gradient

The previous theorem is nice but it requires the objective to be at the same time Lipschitz continuous, strongly convex, and to have a Lipschitz gradient. Unfortunately, this very rarely happens in practice. Yet, if we replace "Lipschitz" with "locally Lipschitz", this issue disappears. The proof can be modified to manage a proximal term, potentially accounting for a projection onto a bounded domain. Also, we will write the proof for this case with the bounded variance condition $\mathbb{E}\left(\left\|\nabla f(x,\xi) - \nabla F\left(x_k\right)\right\|^2\right) \leq C$, which is less restrictive than bounded stochastic gradients $\mathbb{E}\left(\|\nabla f(x,\xi)\|^2\right) \leq C$. We consider the problem

$$\min_{x \in \mathcal{X}} \mathbb{E}(f(x,\xi)) + g(x),$$

where $f(\cdot,\xi)$ is differentiable for all ξ and g has a simple and easily computable *proximal operator*

$$\operatorname{prox}_g(x) = \arg\min_y g(y) + \frac{1}{2}\|x - y\|^2.$$

. For instance,

- $g(\theta) = \|\theta\|_1$ (LASSO penalization),
- $g(\theta) = \|\theta\|_2^2$ (Ridge penalization),
- $g(\theta) = \begin{cases} 0 & \text{if } \theta \in \Theta \\ +\infty & \text{otherwise} \end{cases}$ (characteristic function), equivalent to projecting onto Θ.

When g is the characteristic function of a set $\Theta \subseteq \mathbb{R}^d$, the optimization problem simplifies as

$$\min_{x\in\mathbb{R}^d} \mathbb{E}(f(x,\xi)) + g(x) = \min_{x\in\Theta} \mathbb{E}(f(x,\xi)) \quad \text{(why?)}.$$

Moreover, the proximal operator simplifies as follows

$$\begin{aligned} \forall x \in \mathbb{R}^d : \qquad \operatorname{prox}_g(x) &= \arg\min_{y\in\mathbb{R}^d}\ g(y) + \frac{1}{2}\|x-y\|^2 \\ &= \arg\min_{y\in\Theta}\ 0 + \frac{1}{2}\|x-y\|^2 \\ &= \arg\min_{y\in\Theta}\ \|x-y\|^2 \\ &\overset{\text{def}}{=} \operatorname{Proj}_\Theta(x). \end{aligned}$$

This is called the projection of x onto Θ (closest point to x in Θ).
The proximal stochastic gradient algorithm is given in the pseudo-code below

Algorithm 6: Proximal stochastic gradient algorithm

```
1. Start from $x_0 \in \Theta$
2. Until termination condition, iterate:
       Draw $\xi_{k+1} \sim \mathbb{P}_\xi$, independent of $(\xi_1, \dots, \xi_k)$
       $x_{k+1} = \operatorname{prox}_{\gamma_k g}\big(x_k - \gamma_k \nabla_x f\,(x_k, \xi_{k+1})\big)$
where $\gamma_k > 0, \forall k \in \mathbb{N}$
```

Before moving to this algorithm's convergence properties, we first define *proper functions.*

Definition 1. *A function $g : \mathbb{R}^d \to [-\infty, +\infty]$ is said to be proper if*

1. *g never takes the value $-\infty$ ($\forall x \in \mathbb{R}^d : g(x) > -\infty$)*
2. *g is not identically equal to $+\infty$ ($\exists x_0 \in \mathbb{R}^d : g(x_0) < +\infty$).*

We denote the domain of g by

$$\operatorname{dom}(g) = \{x \in \mathbb{R}^d : g(x) < +\infty\}.$$

The following theorem provides convergence guarantees for the proximal SGD algorithm.

Theorem 4. *Suppose that:*

1. *$(x \mapsto f(x,\xi))$ is convex and differentiable for all ξ,*
2. *g is a proper convex function,*
3. *F is μ-strongly convex and has an L-Lipschitz gradient,*
4. *there exists $C > 0$ such that $\mathbb{E}\left(\|\nabla f(x,\xi) - \nabla F\,(x)\|^2\right) \leq C$ for all $x \in \operatorname{dom} g$,*
5. *there exists $x^* \in \arg\min F + g$,*
6. *the sequence γ_k is deterministic, satisfies $\gamma_k = \frac{2}{\mu(k+1)}$.*

The iterates of the proximal stochastic gradient algorithm satisfy, for $k \geq 2$

$$\mathbb{E}\left[\left\|x_k - x^*\right\|^2\right] \leq \frac{8C}{\mu^2 k}.$$

Before proving the theorem, some remarks are in order.

1. Compared to the previous SGD algorithm, the condition $\mathbb{E}\big(\left\|\nabla_x f(x,\xi)\right\|^2 \big) \leq C$ is replaced with the weaker condition $\mathbb{E}\big(\left\|\nabla_x f(x,\xi) - \nabla F(x_k)\right\|^2 \big) \leq C$.
2. Formally, we can apply this theorem with $g = 0$. We then recover the rate of *non*-proximal SGD algorithm for strongly convex functions, with the *weaker* condition

$$\mathbb{E}\left(\left\|\nabla_x f(x,\xi) - \nabla F(x_k)\right\|^2\right) \leq C.$$

To prove Theorem 4, we will need the following property of the proximal operator.

Lemma 1. *Let $g : \mathbb{R}^d \to (-\infty, +\infty]$ be proper, convex, and assume that* $\operatorname{dom}(g)$ *has non-empty interior. Let $x \in \mathbb{R}^d$ and $\gamma > 0$, and define $p = \operatorname{prox}_{\gamma g}(x)$. Then for any $y \in \mathbb{R}^d$, it holds that*

$$\|y - p\|^2 \leq 2\gamma \big(g(y) - g(p)\big) + \|y - x\|^2 - \|p - x\|^2.$$

Proof Lemma 1. We recall that the subdifferential of a function $h : \mathbb{R}^d \to \mathbb{R}$ at point x is defined as the set

$$\partial h(x) = \left\{u \in \mathbb{R}^d : \forall y \in \mathbb{R}^d, \langle u, y - x\rangle \leq h(y) - h(x)\right\}.$$

If h is convex and differentiable at some $x \in \mathbb{R}^d$, then $\partial h(x) = \{\nabla f(x)\}$ (why?). Moreover, it always holds that x^* is a minimizer of h if, and only if, $0 \in \partial h(x^*)$ (why?). Now, define the function $\phi_x(y) = \frac{1}{2}\|x - y\|^2$ for any $y \in \mathbb{R}^d$. We will use the following property of the subdifferential of sums of functions without proof: If the interior of the domains of γg and ϕ_x have a non-empty intersection (which holds here by assumption), then

$$\partial(\gamma g + \phi_x)(y) = \partial \gamma g(y) + \partial \phi_x(y), \qquad \forall x, y \in \mathbb{R}^d,$$

where for any two sets $A, B \subseteq \mathbb{R}^d$, $A + B = \{a + b : a \in A, b \in B\}$. Interested readers are encouraged to consult [Roc15], Theorem 23.8, for a complete justification of this result that goes beyond the scope of this course.

Here, we have $\partial\phi_x(p) = \{\nabla\phi_x(p)\} = \{p - x\}$. Since p minimizes the function $\gamma g + \phi_x$, we have

$$\begin{aligned} 0 \in \partial(\gamma g + \phi_x)(p) \ &= \ \partial\gamma g(p) + \partial\phi_x(p) \ = \ \partial\gamma g(p) + \{p - x\} \\ \iff x - p \in \partial\gamma g(p) \ &\iff \ \forall y \in \mathbb{R}^d : \langle x - p, y - p\rangle \leq \gamma g(y) - \gamma g(p) \\ \iff \forall y \in \mathbb{R}^d :\ &\|y - p\|^2 \leq 2\gamma\big(g(y) - g(p)\big) + \|y - x\|^2 - \|p - x\|^2, \end{aligned}$$

by rearranging the terms. □

We can now move to the proof of the theorem.

Proof Theorem 4. We apply Lemma 1 with

$$y = x^*, \qquad p = x_{k+1}, \qquad \text{and} \qquad x = x_k - \gamma_k \nabla f(x_k, \xi_{k+1}) \overset{\text{Def}}{=} x_k - \delta.$$

We obtain

$$\begin{aligned} \left\|x^* - x_{k+1}\right\|^2 &\leq 2\gamma_k \big(g(x^*) - g(x_{k+1})\big) + \left\|x^* - x_k + \delta\right\|^2 - \left\|x_{k+1} - x_k + \delta\right\|^2 \\ &= 2\gamma_k \big(g(x^*) - g(x_{k+1})\big) + \|x^* - x_k\|^2 \ + \ 2\langle \delta, x^* - x_k \rangle \ + \ \cancel{\|\delta\|^2} \\ &\qquad\qquad - \|x_{k+1} - x_k\|^2 - 2\langle \delta, x_{k+1} - x_k \rangle - \cancel{\|\delta\|^2}. \end{aligned}$$

We first handle the term $\langle \delta, x^* - x_k \rangle$. Defining $\mathbb{E}_k = \mathbb{E}\left[\,\cdot\,|x_k\right]$, we get

$$\begin{aligned} \mathbb{E}_k \langle \delta, x^* - x_k \rangle = \gamma_k \mathbb{E}_k \big\langle \nabla f(x_k, \xi_{k+1}),\, x^* - x_k \big\rangle &= \gamma_k \big\langle \mathbb{E}_k \nabla f(x_k, \xi_{k+1}),\, x^* - x_k \big\rangle \quad \text{(why?)} \\ &= \gamma_k \big\langle \nabla F(x_k),\, x^* - x_k \big\rangle \qquad \text{by Proposition 20 in the Appendix} \\ &\leq \gamma_k \Big[F(x^*) - F(x_k) - \frac{\mu}{2}\left\|x_k - x^*\right\|^2\Big] \qquad \text{by strong convexity.} \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}_k \left\|x^* - x_{k+1}\right\|^2 &\leq 2\gamma_k \mathbb{E}_k \big(g(x^*) - g(x_{k+1})\big) + \|x^* - x_k\|^2 \ + \ 2\gamma_k \Big[F(x^*) - F(x_k) - \frac{\mu}{2}\|x_k - x^*\|^2\Big] \\ &\qquad - \mathbb{E}_k \|x_{k+1} - x_k\|^2 - 2\,\mathbb{E}_k \langle \delta, x_{k+1} - x_k \rangle \\ &= (1 - \gamma_k \mu)\left\|x^* - x_k\right\|^2 - \mathbb{E}_k \left\|x_{k+1} - x_k\right\|^2 - 2\,\mathbb{E}_k \langle \delta, x_{k+1} - x_k \rangle \\ &\qquad + 2\gamma_k\, \mathbb{E}_k \big((F + g)(x^*) - F(x_k) - g(x_{k+1})\big)\,. \end{aligned} \tag{2.4}$$

We recall that, by the Taylor-Lagrange inequality, we have

$$F\left(x_{k+1}\right) \leq F\left(x_k\right) + \big\langle \nabla F\left(x_k\right), x_{k+1} - x_k \big\rangle + \frac{L}{2}\left\|x_{k+1} - x_k\right\|^2.$$

Rearranging the terms, we can further link $F(x_k)$ with $F(x_{k+1})$ as follows:

$$\begin{aligned} -F\left(x_k\right) \leq -F\left(x_{k+1}\right) &+ \big\langle \nabla f\left(x_k, \xi_{k+1}\right), x_{k+1} - x_k \big\rangle \\ &+ \big\langle \nabla F\left(x_k\right) - \nabla f\left(x_k, \xi_{k+1}\right), x_{k+1} - x_k \big\rangle + \frac{L}{2}\left\|x_{k+1} - x_k\right\|^2. \end{aligned}$$

Subtracting $g(x_{k+1})$ on both sides and applying the conditional expectation $\mathbb{E}_k$, we obtain

$$\begin{aligned} -F(x_k) - g(x_{k+1}) &\leq -\mathbb{E}_k\left[F(x_{k+1}) + g(x_{k+1})\right] + \frac{1}{\gamma_k}\mathbb{E}_k \langle \delta, x_{k+1} - x_k \rangle \\ &\qquad + \mathbb{E}_k \Big[\big\langle \nabla F\left(x_k\right) - \nabla f\left(x_k, \xi_{k+1}\right), x_{k+1} - x_k \big\rangle + \frac{L}{2}\left\|x_{k+1} - x_k\right\|^2\Big] \\ &\leq -\big(F(x^*) + g(x^*)\big) + \frac{1}{\gamma_k}\mathbb{E}_k \langle \delta, x_{k+1} - x_k \rangle \\ &\qquad + \mathbb{E}_k \Big[\gamma_k \left\|\nabla F\left(x_k\right) - \nabla f\left(x_k, \xi_{k+1}\right)\right\|^2 + \frac{1}{4\gamma_k}\left\|x_{k+1} - x_k\right\|^2 + \frac{L}{2}\left\|x_{k+1} - x_k\right\|^2\Big] \\ &\leq -\big(F(x^*) + g(x^*)\big) + \frac{1}{\gamma_k}\mathbb{E}_k \langle \delta, x_{k+1} - x_k \rangle \\ &\qquad + \gamma_k C + \left(\frac{1}{4\gamma_k} + \frac{L}{2}\right)\mathbb{E}_k\left[\left\|x_{k+1} - x_k\right\|^2\right] \qquad \text{by assumption.} \end{aligned}$$

In the second inequality, we used the fact that x^* minimizes the function $F+g$ to handle the term $-\mathbb{E}_k\left[F(x_{k+1})+g(x_{k+1})\right]$, and we also used the inequality $\langle a,b\rangle \le \frac{\|a\|^2}{2}+\frac{\|b\|^2}{2}$ that holds true for any $a,b\in\mathbb{R}^d$, applied with

$$a=\sqrt{2\gamma_k}\left(\nabla F\left(x_k\right)-\nabla f\left(x_k,\xi_{k+1}\right)\right) \qquad \text{and} \qquad b=\frac{1}{\sqrt{2\gamma_k}}\big(x_{k+1}-x_k\big).$$

Plugging into (2.4), we obtain

$$\begin{aligned}
\mathbb{E}_k\left\|x^*-x_{k+1}\right\|^2 &= (1-\gamma_k\mu)\left\|x^*-x_k\right\|^2-\mathbb{E}_k\left\|x_{k+1}{-}x_k\right\|^2\\
&\quad+2\gamma_k\left[\gamma_k C+\left(\frac{1}{4\gamma_k}+\frac{L}{2}\right)\mathbb{E}_k\left[\|x_{k+1}-x_k\|^2\right]\right]\\
&=(1-\gamma_k\mu)\left\|x^*-x_k\right\|^2+2\gamma_k^2C+\Big(L\gamma_k-\frac{1}{2}\Big)\mathbb{E}_k\left[\|x_{k+1}-x_k\|^2\right]\\
&\le(1-\gamma_k\mu)\left\|x^*-x_k\right\|^2+2\gamma_k^2C,
\end{aligned}$$

since $\gamma_k L\le\frac{1}{2}$ for $k\ge 2$ by the choice of γ_k since $\mu\ge L$. We conclude as in the proof of Theorem 3.

□

2.4 Comparison of the results depending on the assumption

We have shown in this chapter several convergence results for the stochastic gradient algorithm, depending on the assumptions we made. The following table summarizes them.

	$\min_x \mathbb{E}[f(x,\xi)]$	$\min_x \mathbb{E}[f(x,\xi)]+g(x)$
Convex	yes	μ-strongly convex.
Lip ∇F	no	yes
Noise	$\mathbb{E}\left(\|\nabla f(x,\xi)\|^2\right)\le C$	$\mathbb{E}\left(\|\nabla f(x,\xi)-\nabla F(x)\|^2\right)\le C$
Step-size	$\gamma_k=\frac{\gamma_0}{\sqrt{k+1}}$	$\gamma_k=\frac{a}{\mu(k+b)}$
Rate	$\mathbb{E}\left[F\left(\bar{x}_k^{\gamma}\right)-F\left(x^*\right)\right]\in O\left(\frac{\ln(k)}{\sqrt{k}}\right)$	$\mathbb{E}\left[\|x_k-x^*\|^2\right]\in O\left(\frac{1}{k}\right)$

Chapter 3

Stochastic variance-reduced gradient

3.1 Motivation and algorithm

In this chapter, we will focus on minimizing an objective function expressed as a finite sum:

$$\min_x F(x) = \min_x \frac{1}{N} \sum_{i=1}^{N} f_i(x).$$

For large-sum problems, a stochastic algorithm like stochastic gradient descent (SGD) is often more efficient than gradient descent (GD) [Bot10]. Where GD requires N gradient computations at each step, SGD only requires one. Consider a μ strongly convex function $F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$ and a precision goal of ϵ. SGD needs about $1/\epsilon$ iterations, while GD requires $O(\ln(1/\epsilon))$. Hence, for $1/\epsilon < N \ln(1/\epsilon)$, SGD is the better choice.

Variance-reduced stochastic gradient methods attempt to get the best of both worlds by combining the advantages of these two algorithms: a cheap per-iteration cost and a linear convergence rate on strongly convex functions. The idea is to compute one full gradient from time to time while using stochastic gradients the rest of the time. During the phases where stochastic gradients are employed, it becomes possible to incorporate some information from the latest computed full gradient to correct the stochastic gradients and reduce their variance. The idea is to use the concept of control variates. The control variate we use is a periodically-computed full gradient. By carefully setting the period, we can mitigate the heavy cost of the computation of this gradient while significantly improving the quality of the stochastic gradients. Starting from an initial value w_0, the classical Stochastic Variance Reduced Gradient (SVRG) algorithm is given as follows:

Algorithm 7: Classical Stochastic variance-reduced gradient (SVRG)

```
Initial value: $w_0$
**Outer loop:** At step $k$:
    $x_0 = w_k$
    Compute and store $\nabla F(w_k)$
    **Inner loop:** For $t = 0, \dots, T-1$:
        Draw $i_t \sim \mathrm{Unif}(\{1, \dots, N\})$ independent of the past
        $g_t = \nabla f_{i_t}(x_t) - \big(\nabla f_{i_t}(w_k) - \nabla F(w_k)\big)$
        $x_{t+1} = x_t - \gamma g_t$
    $w_k = \frac{1}{T}\sum_{t=1}^{T} x_t$
```

In this course, we will study a slightly modified version of the algorithm above, which is more convenient to analyze mathematically[1]. Starting from a given input x_0, setting $w_0 = x_0$ and using a probability $p < 1$ of updating the control variate, Stochastic Variance Reduced Gradient (SVRG) is given as follows:

Algorithm 8: Stochastic variance-reduced gradient (SVRG)

```
**Initial value:** $x_0$ and set $w_0 = x_0$.
Until termination condition, iterate
    $i_{k+1} \sim \mathrm{Unif}(\{1, \dots, N\})$ independent of the past
    $g_{k+1} = \nabla F(w_k) + \nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(w_k)$
    $x_{k+1} = x_k - \gamma g_{k+1}$
    $w_{k+1} = \begin{cases} x_k & \text{with probability } p \\ w_k & \text{with probability } 1-p \end{cases}$
```

These two algorithms are very closely related: The main difference is that the first algorithm updates w_k exactly every T steps, while the second one has a small probability p of updating w_k at each iteration, which implies that it updates this parameter every $1/p$ steps on average. If $p = 1/T$, then the update occurs every T steps on average.

3.2 Convergence

We will need the following result on convex functions with a Lipschitz gradient.

Proposition 4. *Let f be a convex function with an L-Lipschitz gradient. For all x and y,*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|^2.$$

Proof. We fix a vector y and define the function $\phi : x \mapsto f(x) - \langle \nabla f(y), x - y \rangle$. It is readily seen that ϕ is convex and that $\nabla \phi(x) = \nabla f(x) - \nabla f(y)$. Hence $\nabla \phi(y) = 0$ and $y \in \arg\min \phi$.
Thus, using the Taylor Lagrange inequality

$$\begin{aligned} \phi(y) &\leq \phi\Big(x - \frac{1}{L}\nabla\phi(x)\Big) \leq \phi(x) + \Big\langle \nabla\phi(x), -\frac{1}{L}\nabla\phi(x) \Big\rangle + \frac{L}{2}\Big\| \frac{1}{L}\nabla\phi(x) \Big\|^2 \\ \phi(y) &\leq \phi(x) - \frac{1}{2L}\|\nabla\phi(x)\|^2. \end{aligned}$$

Recalling the definition of ϕ, we obtain $f(x) - \langle \nabla f(y), x - y \rangle \geq f(y) + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$. □

The previous lemma allows us to prove an important result, referred to as the Variance Reduction Lemma.

Lemma 2 (Variance reduction)**.** *Let $f_1, \dots f_N$ be N convex functions with L-Lipschitz gradients, and let $F = \frac{1}{N}\sum_{i=1}^N f_i$. Suppose F is minimized at $x^* \in \mathbb{R}^d$. We denote by $\mathbb{E}_I$ the expectation with respect to the random variable $I \sim \mathrm{Unif}(\{1, \dots, N\})$. We have*

$$\mathbb{E}_I \Big[\big\| \nabla f_I(y) - \nabla f_I(x^*) \big\|^2 \Big] \leq 2L\big(F(y) - F(x^*)\big).$$

[1] This version was proposed in [KHR20]

Note that in the lemma above, we can more simply rewrite the expectation as

$$\mathbb{E}_I \Big\| \nabla f_I(y) - \nabla f_I(x^*) \Big\|^2 = \frac{1}{N} \sum_{i=1}^{N} \Big\| \nabla f_i(y) - \nabla f_i(x^*) \Big\|^2 .$$

Proof of the Variance Reduction Lemma. By Proposition 4, we have

$$\begin{aligned} \mathbb{E}_I \Big\| \nabla f_I(y) - \nabla f_I(x^*) \Big\|^2 &\leq 2L\, \mathbb{E}_I \left[f_I(y) - f_I(x^*) - \langle \nabla f_I(x^*), y - x^* \rangle \right] \\ &= 2L \left(F(y) - F(x^*) - \langle \nabla F(x^*), y - x^* \rangle \right) \\ &= 2L \big(F(y) - F(x^*) \big). \end{aligned}$$

□

The Variance Reduction Lemma is a key result in the analysis of SVRG, as it explains why this algorithm performs a variance reduction. First, it is easy to see that each updating increment $-\gamma g_t$, where $g_t = \nabla f_{i_t}(x_t) - \big(\nabla f_{i_t}(w_k) - \nabla F(w_k) \big)$ is correctly centered around the direction of the steepest descent. More precisely, it holds that

$$\mathbb{E}_t\, g_t = \mathbb{E}_t \Big[\nabla f_{i_t}(x_t) - \big(\nabla f_{i_t}(w_k) - \nabla F(w_k) \big) \Big] = \nabla F(x_t) - \cancel{\nabla F(w_k)} + \cancel{\nabla F(w_k)} = \nabla F(x_t).$$

The Variance Reduction Lemma further guarantees that the variance of each update g_t decays when the iterates of the algorithm converge toward the minimizer. Indeed, using $\|a+b+c\|^2 \leq 3\|a\|^2 + 3\|b\|^2 + 3\|c\|^2$, the conditional variance of $\mathbb{V}_t(g_{t+1}) = \mathbb{V}(g_{t+1} | x_t, w_k)$ at time t satisfies

$$\begin{aligned} \mathbb{V}_t(g_{t+1}) = \mathbb{E}_t \big\| g_{t+1} - \mathbb{E}_t g_{t+1} \big\|^2 &= \mathbb{E}_t \big\| \nabla f_{i_{t+1}}(x_t) - \big(\nabla f_{i_{t+1}}(w_k) - \nabla F(w_k) \big) - \nabla F(x_t) \big\|^2 \\ &\leq 3\mathbb{E}_t \big\| \nabla f_{i_{t+1}}(x_t) - \nabla f_{i_{t+1}}(x^*) \big\|^2 + 3\mathbb{E}_t \big\| \nabla f_{i_{t+1}}(w_k) - \nabla f_{i_{t+1}}(x^*) \big\|^2 \\ &\quad + 3 \big\| \nabla F(x_t) - \nabla F(w_k) \big\|^2 \\ &\leq 6L \Big(F(x_t) - F(x^*) + F(w_k) - F(x^*) \Big) + 3L^2 \big\| x_t - w_k \big\|^2 \\ &\longrightarrow 0 \qquad \text{as } x_t, w_k \to x^*. \end{aligned}$$

Therefore, as the iterates x_t of the algorithm converge toward x^*, we can expect the update $-\gamma_{t+1} g_{t+1}$ to be centered around the correct direction $-\gamma_{t+1} \nabla F(x_t)$, but with a decaying variance. This allows us to use larger step sizes $\gamma_k := \gamma$ that do not vanish as $t \to \infty$, thereby accelerating the convergence. The convergence rate of SVRG is given by the following theorem.

Theorem 5. *Suppose that for all $i \in \{1, \ldots, N\}$, f_i is convex and differentiable, ∇f_i is L Lipschitz and F is μ-strongly convex. Denote by x^* the unique minimizer of F and suppose that $\gamma \leq \frac{1}{15L}$. The iterates of SVRG converge linearly as*

$$\mathbb{E} \left[\big\| x_k - x^* \big\|^2 \right] \leq c^k \Delta_0$$

where $c = \max(1 - \gamma\mu, 1 - p/2)$ and $\Delta_0 = \|x_0 - x^*\|^2 + \frac{24\gamma^2}{p}(F(x_0) - F(x^*))$.
Moreover, the expected cost of an iteration is $2 + pN$ stochastic gradients.

Before proceeding to the proof, some remarks are in order.

1. Now the rate has been improved to
$$\Delta_0\big[\max(1-\gamma\mu, 1-p/2)\big]^k \le \Delta_0 \exp\left(-k\min\left(\gamma\mu, \frac{p}{2}\right)\right).$$
2. The rate is exponentially fast in k (in comparison, the rate of SGD with strongly convex functions was $O(\frac{1}{\mu^2 k})$). However, the initial condition Δ_0 depends on the function F to optimize.
3. We can now take γ as a constant (for the classical SGD with strongly convex functions, it was $\gamma_k = \frac{\gamma^*}{k+1}$). In other words, due to the variance reduction, we can take larger steps!

Proof Theorem 5. We need to compute one full gradient $\nabla F(w_k)$—which requires N stochastic gradients—only when w_k is updated, which happens with probability p at each step. We also need to compute $\nabla f_{i_{k+1}}(x_k)$ and $\nabla f_{i_{k+1}}(w_k)$ at each iteration. The cost in terms of the number of stochastic gradients per iteration is therefore $2+pN$ in expectation at each iteration.

We now proceed to the convergence rate. Note that $\mathbb{E}_k[g_{k+1}] = \nabla F(w_k) + \nabla F(x_k) - \nabla F(w_k) = \nabla F(x_k)$.

$$\begin{aligned}
\left\|x_{k+1}-x^*\right\|^2 &= \left\|x_k-x^*\right\|^2 - 2\gamma\left\langle g_{k+1}, x_k-x^*\right\rangle + \gamma^2\left\|g_{k+1}\right\|^2 \\
\mathbb{E}_k\left[\left\|x_{k+1}-x^*\right\|^2\right] &= \left\|x_k-x^*\right\|^2 + 2\gamma\left\langle \nabla F(x_k), x^*-x_k\right\rangle + \gamma^2\mathbb{E}_k\left[\left\|g_{k+1}\right\|^2\right] \quad \text{(why?)} \\
\mathbb{E}_k\left[\left\|x_{k+1}-x^*\right\|^2\right] &\le (1-\gamma\mu)\left\|x_k-x^*\right\|^2 + 2\gamma\Big(F(x^*)-F(x_k)\Big) + \gamma^2\mathbb{E}_k\left[\left\|g_{k+1}\right\|^2\right]
\end{aligned}$$

We now control the noise term $\|g_{k+1}\|^2$. Using $\nabla F(x^*) = 0$ and the relation $\|a+b+c\|^2 \le 3\|a\|^2 + 3\|b\|^2 + 3\|c\|^2$ that holds true for any $a, b, c \in \mathbb{R}^d$, we obtain

$$\begin{aligned}
\mathbb{E}_k\left[\left\|g_{k+1}\right\|^2\right] &= \mathbb{E}_k\left[\left\|\nabla F(w_k) + \nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(w_k)\right\|^2\right] \\
&= \mathbb{E}_k\left[\left\|\nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(x^*) + \nabla F(w_k) - \nabla F(x^*) + \nabla f_{i_{k+1}}(x^*) - \nabla f_{i_{k+1}}(w_k)\right\|^2\right] \\
&\le 3\mathbb{E}_k\left[\left\|\nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(x^*)\right\|^2\right] + 3\mathbb{E}_k\left[\left\|\nabla F(w_k) - \nabla F(x^*)\right\|^2\right] \\
&\quad + 3\mathbb{E}_k\left[\left\|\nabla f_{i_{k+1}}(x^*) - \nabla f_{i_{k+1}}(w_k)\right\|^2\right] \\
&= 3\left\{2L\Big(F(x_k)-F(x^*)\Big) + 2L\Big(F(w_k)-F(x^*)\Big) + 2L\Big(F(w_k)-F(x^*)\Big)\right\} \\
&= 6L\Big(F(x_k)-F(x^*)\Big) + 12L\Big(F(w_k)-F(x^*)\Big).
\end{aligned}$$

In the second to last step, we used that for any $y \in \mathbb{R}^d$, we have $\mathbb{E}_k\left[\left\|\nabla f_{i_{k+1}}(y) - \nabla f_{i_{k+1}}(x^*)\right\|^2\right] \le 2L(F(y)-F(x^*))$ by the variance-reduction Lemma, and $\mathbb{E}_k\left[\left\|\nabla F(w_k) - \nabla F(x^*)\right\|^2\right] = \left\|\nabla F(w_k) - \nabla F(x^*)\right\|^2 \le 2L\left(F(y)-F(x^*)\right)$ by Proposition 4. Therefore, we have

$$\mathbb{E}_k\left[\left\|x_{k+1}-x^*\right\|^2\right] \le (1-\gamma\mu)\left\|x_k-x^*\right\|^2 + (2\gamma-6\gamma^2L)\left(F(x^*)-F(x_k)\right) + 12\gamma^2L\Big(F(w_k)-F(x^*)\Big) \tag{3.1}$$

Moreover, we recall that, conditional on x_k and w_k, we have $w_{k+1} = w_k$ with probability $1-p$ and $w_{k+1} = x_k$ with probability p. Therefore,

$$\mathbb{E}_k\left[F(w_{k+1}) - F(x^*)\right] = (1-p)\Big(F(w_k) - F(x^*)\Big) + p\Big(F(x_k) - F(x^*)\Big). \tag{3.2}$$

By equations (3.1) and (3.2), we can deduce that

$$\begin{aligned}
&\mathbb{E}_k\left[\left\|x_{k+1} - x^*\right\|^2 + \frac{24\gamma^2 L}{p}\Big(F(w_{k+1}) - F(x^*)\Big)\right] \\
&\leq (1-\gamma\mu)\left\|x_k - x^*\right\|^2 + (2\gamma - 6\gamma^2 L)\left(F\left(x^*\right) - F\left(x_k\right)\right) + 12\gamma^2 L\Big(F(w_k) - F(x^*)\Big) \\
&\quad + \frac{24\gamma^2 L}{p}\left\{(1-p)\Big(F(w_k) - F(x^*)\Big) + p\Big(F(x_k) - F(x^*)\Big)\right\} \\
&= (1-\gamma\mu)\left\|x_k - x^*\right\|^2 + \underbrace{(2\gamma - 30\gamma^2 L)}_{\geq 0}\underbrace{\left(F\left(x^*\right) - F\left(x_k\right)\right)}_{\leq 0} + \left(12 + \frac{24(1-p)}{p}\right)\gamma^2 L\Big(F\left(w_k\right) - F\left(x^*\right)\Big) \\
&\leq (1-\gamma\mu)\left\|x_k - x^*\right\|^2 + 0 + \left(1 - p/2\right)\frac{24\gamma^2 L}{p}\Big(F\left(w_k\right) - F\left(x^*\right)\Big) \\
&\leq \max(1-\gamma\mu, 1-p/2)\left\{\left\|x_k - x^*\right\|^2 + \frac{24\gamma^2 L}{p}\Big(F\left(w_k\right) - F\left(x^*\right)\Big)\right\}.
\end{aligned}$$

In the third step, we used that $2\gamma - 30\gamma^2 L \geq 0$ since $\gamma \leq \frac{1}{15L}$ by assumption. Now, taking total expectations on both sides, and defining $c = \max(1-\gamma\mu, 1-p/2)$ for any $k \in \mathbb{N}$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\left\|x_k - x^*\right\|^2 + \frac{24\gamma^2 L}{p}\Big(F\left(w_k\right) - F\left(x^*\right)\Big)\right] &\leq c\,\mathbb{E}\left[\left\|x_{k-1} - x^*\right\|^2 + \frac{24\gamma^2 L}{p}\Big(F\left(w_{k-1}\right) - F\left(x^*\right)\Big)\right] \\
&\leq c^k\,\mathbb{E}\left[\left\|x_0 - x^*\right\|^2 + \frac{24\gamma^2 L}{p}\Big(F\left(w_0\right) - F\left(x^*\right)\Big)\right]
\end{aligned}$$

by induction. The result follows. □

Chapter 4

Acceleration methods[1]

Acceleration is an important tool, especially for the training of neural networks [SMDH13]. The idea was first introduced by Polyak in 1964 under the name "heavy ball method" [Pol64]. It is inspired by the dynamics of a heavy ball rolling down the valley of the loss landscape. Since then other types of acceleration have been proposed and analyzed, with Nesterov acceleration being the most prominent example [Nes83]. In this section, we first give some intuition by discussing the heavy ball method for a simple quadratic loss. Afterwards we turn to Nesterov acceleration and give a convergence proof for L-smooth and μ-strongly convex objective functions that improves upon the bounds obtained for gradient descent.

4.1 Momentum algorithm (a.k.a. heavy ball method)

Gradient descent or stochastic gradient descent may exhibit a slow convergence behavior in situations where the function to optimize is more convex along certain directions than others. Let us consider the simple example below:

$$F(u) = 6u_1^2 + \frac{1}{2}u_2^2, \qquad \forall u \in \mathbb{R}^2. \tag{4.1}$$

Running GD or SGD with too small step sizes is known to slow down convergence (why?). In order to speed up convergence, one might be tempted to increase the step size so as to take larger steps toward the function's minimizer. However, in the example above, increasing the step size may cause another issue: By taking too large steps, one may overshoot at every iteration (that is, go slightly too far each time). If the step size is even larger, overshooting may cause the trajectory to diverge from the optimizer. In fact, the loss landscape of this function looks like a ravine (the derivative is much larger in one direction than in the other) and ∇F mainly points toward the "opposite side of the ravine" rather than toward the optimizer. Therefore, the iterates oscillate back and forth in the first coordinate and make little progress in the direction of the valley along the second coordinate axis. This is illustrated by the blue trajectory in the figure below.

[1] This chapter is adapted from Chapter 10 of [PZ24]

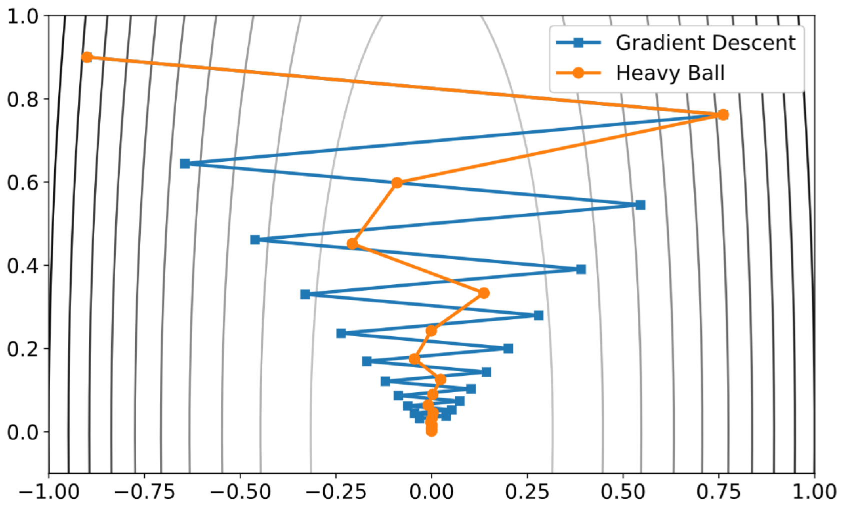


Figure 4.1: 20 steps of gradient descent and the heavy ball (momentum) method on the objective function (4.1) with step size $h = \alpha = h_*$ and $\beta = 1/3$.

To address this issue, the heavy ball method introduces a "momentum" term which can mitigate this effect to some extent. The idea is to choose the update not just according to the gradient at the current location, but to add information from the previous steps. After initializing $\boldsymbol{w}_0$ and, e.g., $\boldsymbol{w}_1 = \boldsymbol{w}_0 - \alpha \nabla f(\boldsymbol{w}_0)$, let for $k \in \mathbb{N}$

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha \nabla f(\boldsymbol{w}_k) + \beta(\boldsymbol{w}_k - \boldsymbol{w}_{k-1}). \tag{4.2}$$

This is known as Polyak's heavy ball method [Pol64]. Here $\alpha > 0$ and $\beta \in (0, 1)$ are hyperparameters (that could also depend on k) and in practice need to be carefully tuned to balance the strength of the gradient and the momentum term. Iteratively expanding (10.4.5) with the given initialization, observe that for $k \geq 0$

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha \left(\sum_{j=0}^{k} \beta^j \nabla f(\boldsymbol{w}_{k-j}) \right).$$

Thus, $\boldsymbol{w}_k$ is updated using an exponentially weighted average of all past gradients. Choosing the momentum parameter β in the interval $(0, 1)$ ensures that the influence of previous gradients on the update decays exponentially. The value of β determines the balance between the impact of recent and past gradients.

Intuitively, this (exponentially weighted) linear combination of the past gradients averages out some of the oscillation observed for gradient descent in Figure 4.1 in the first coordinate, and thus "smoothes out" the path. The partial derivative in the second coordinate, along which the objective function is very flat, does not change much from one iterate to the next. Thus, its proportion in the update is strengthened through the addition of momentum.

As mentioned earlier, the heavy ball method can be interpreted as a discretization of the dynamics of a ball rolling down the valley of the loss landscape. If the ball has positive mass, i.e. is "heavy", its momentum prevents the ball from bouncing back and forth too strongly. The following remark further elucidates this connection.

Remark: As pointed out, e.g., in [Qia99], for suitable choices of α and β, (4.2) can be interpreted as a discretization of the second-order ODE

$$m\boldsymbol{w}''(t) = -\nabla f(\boldsymbol{w}(t)) - r\boldsymbol{w}'(t) \tag{4.3}$$

This equation describes the movement of a point mass m under influence of the force field $-\nabla f(\boldsymbol{w}(t))$; the term $-\boldsymbol{w}'(t)$, which points in the negative direction of the current velocity, corresponds to friction, and $r > 0$ is the friction coefficient. The discretization

$$m\frac{\boldsymbol{w}_{k+1} - 2\boldsymbol{w}_k + \boldsymbol{w}_{k-1}}{h^2} = -\nabla f(\boldsymbol{w}_k) - \frac{\boldsymbol{w}_{k+1} - \boldsymbol{w}_k}{h}$$

then leads to

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \underbrace{\frac{h^2}{m - rh}}_{=\alpha} \nabla f(\boldsymbol{w}_k) + \underbrace{\frac{m}{m - rh}}_{=\beta} (\boldsymbol{w}_k - \boldsymbol{w}_{k-1})$$

and thus to (4.2), [Qia99]. Letting $m = 0$ in the equation above, we recover the gradient descent update. Hence, the positive mass corresponds to the momentum term.

4.2 Nesterov acceleration

Nesterov's accelerated gradient method (NAG) [Nes83, NTS15], is a refinement of the heavy ball method. After initializing $\boldsymbol{v}_0, \boldsymbol{w}_0 \in \mathbb{R}^n$, the update is formulated as the two-step process

$$\boldsymbol{v}_{k+1} = \boldsymbol{w}_k - \alpha\nabla f(\boldsymbol{w}_k) \tag{4.4}$$

$$\boldsymbol{w}_{k+1} = \boldsymbol{v}_{k+1} + \beta(\boldsymbol{v}_{k+1} - \boldsymbol{v}_k) \tag{4.5}$$

where again $\alpha > 0$ and $\beta \in (0, 1)$ are hyperparameters. Substituting the second line into the first we get

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k - \alpha\nabla f(\boldsymbol{w}_k) + \beta(\boldsymbol{v}_k - \boldsymbol{v}_{k-1})$$

Comparing with the momentum method (4.2), the key difference is that the gradient is not evaluated at the current position $\boldsymbol{v}_k$, but instead at the point $\boldsymbol{w}_k = \boldsymbol{v}_k + \beta(\boldsymbol{v}_k - \boldsymbol{v}_{k-1})$, which can be interpreted as an estimate of the position at the next iteration.

We next discuss the convergence for L-smooth and μ-strongly convex objective functions f. It turns out, that these conditions are not sufficient in order for the heavy ball method (4.2) to converge, and one can construct counterexamples [LRP16]. This is in contrast to NAG, as the next theorem shows. To give the analysis, it is convenient to first rewrite (4.4) and (4.5) as a three sequence updates: Let $\tau = \sqrt{\mu/L}, \alpha = 1/L$, and $\beta = (1-\tau)/(1+\tau)$. After initializing $\boldsymbol{w}_0, \boldsymbol{v}_0 \in \mathbb{R}^n$, (4.4) and (4.5) can also be written as $\boldsymbol{u}_0 = \big((1+\tau)\boldsymbol{w}_0 - \boldsymbol{v}_0\big)/\tau$ and for $k \in \mathbb{N}_0$

$$\boldsymbol{w}_k = \frac{\tau}{1+\tau}\boldsymbol{u}_k + \frac{1}{1+\tau}\boldsymbol{v}_k \tag{4.6}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{w}_k - \frac{1}{L}\nabla f(\boldsymbol{w}_k) \tag{4.7}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \tau \cdot (\boldsymbol{w}_k - \boldsymbol{u}_k) - \frac{\tau}{\mu}\nabla f(\boldsymbol{w}_k) \tag{4.8}$$

The following theorem yields convergence guarantees of the NAG algorithm in the case of strongly convex and smooth objective functions.

Theorem 6. *Let $n \in \mathbb{N}$ and $L, \mu > 0$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-smooth and μ-strongly convex. Further, let $\boldsymbol{v}_0, \boldsymbol{w}_0 \in \mathbb{R}^n$ and let $\tau = \sqrt{\mu/L}$. Let $(\boldsymbol{w}_k, \boldsymbol{v}_{k+1}, \boldsymbol{u}_{k+1})_{k=0}^{\infty} \subseteq \mathbb{R}^n$ be defined by (4.6), (4.7) and (4.8), and let $\boldsymbol{w}_*$ be the unique minimizer of f.*
Then, for all $k \in \mathbb{N}$, it holds that

$$\begin{aligned} \left\|\boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2 &\le \frac{2}{\mu}\left(1 - \sqrt{\frac{\mu}{L}}\right)^k \left(f\left(\boldsymbol{v}_0\right) - f\left(\boldsymbol{w}_*\right) + \frac{\mu}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{w}_*\right\|^2\right) \\ f\left(\boldsymbol{v}_k\right) - f\left(\boldsymbol{w}_*\right) &\le \left(1 - \sqrt{\frac{\mu}{L}}\right)^k \left(f\left(\boldsymbol{v}_0\right) - f\left(\boldsymbol{w}_*\right) + \frac{\mu}{2}\left\|\boldsymbol{u}_0 - \boldsymbol{w}_*\right\|^2\right) \end{aligned}$$

Proof. Define

$$e_k := f\left(\boldsymbol{v}_k\right) - f\left(\boldsymbol{w}_*\right) + \frac{\mu}{2}\left\|\boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2. \tag{4.9}$$

To prove the first claim of the theorem, it suffices to prove that $e_{k+1} \le c e_k$ for all $k \in \mathbb{N}_0$ where $c = 1 - \tau$. We start with the last term in the display above. By (4.8)

$$\begin{aligned} &\frac{\mu}{2}\left\|\boldsymbol{u}_{k+1} - \boldsymbol{w}_*\right\|^2 - \frac{\mu}{2}\left\|\boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2 \\ &\quad = \frac{\mu}{2}\left\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k + \boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2 - \frac{\mu}{2}\left\|\boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2 \\ &\quad = \frac{\mu}{2}\left\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\right\|^2 + \frac{\mu}{2}\cdot\left(2\left\langle \tau\cdot\left(\boldsymbol{w}_k - \boldsymbol{u}_k\right) - \frac{\tau}{\mu}\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{u}_k - \boldsymbol{w}_*\right\rangle\right) \\ &\quad = \frac{\mu}{2}\left\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\right\|^2 + \tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{w}_* - \boldsymbol{u}_k\right\rangle - \tau\mu\left\langle \boldsymbol{w}_k - \boldsymbol{u}_k, \boldsymbol{w}_* - \boldsymbol{u}_k\right\rangle. \end{aligned} \tag{4.10}$$

From (4.8) we have $\tau\boldsymbol{u}_k = (1+\tau)\boldsymbol{w}_k - \boldsymbol{v}_k$ so that

$$\tau\cdot\left(\boldsymbol{w}_k - \boldsymbol{u}_k\right) = \tau\boldsymbol{w}_k - (1+\tau)\boldsymbol{w}_k + \boldsymbol{v}_k = \boldsymbol{v}_k - \boldsymbol{w}_k \tag{4.11}$$

and using μ-strong convexity, we get

$$\begin{aligned} &\tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{w}_* - \boldsymbol{u}_k\right\rangle = \tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{w}_k - \boldsymbol{u}_k\right\rangle + \tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{w}_* - \boldsymbol{w}_k\right\rangle \\ &\le \left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k - \boldsymbol{w}_k\right\rangle - \tau\cdot\left(f\left(\boldsymbol{w}_k\right) - f\left(\boldsymbol{w}_*\right)\right) - \frac{\tau\mu}{2}\left\|\boldsymbol{w}_k - \boldsymbol{w}_*\right\|^2. \end{aligned}$$

Moreover,

$$\begin{aligned} &-\frac{\tau\mu}{2}\left\|\boldsymbol{w}_k - \boldsymbol{w}_*\right\|^2 - \tau\mu\left\langle\boldsymbol{w}_k - \boldsymbol{u}_k, \boldsymbol{w}_* - \boldsymbol{u}_k\right\rangle \\ &\qquad = -\frac{\tau\mu}{2}\left(\left\|\boldsymbol{w}_k - \boldsymbol{w}_*\right\|^2 - 2\left\langle\boldsymbol{w}_k - \boldsymbol{u}_k, \boldsymbol{w}_k - \boldsymbol{w}_*\right\rangle + 2\left\langle\boldsymbol{w}_k - \boldsymbol{u}_k, \boldsymbol{w}_k - \boldsymbol{u}_k\right\rangle\right) \\ &\qquad = -\frac{\tau\mu}{2}\left(\left\|\boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2 + \left\|\boldsymbol{w}_k - \boldsymbol{u}_k\right\|^2\right). \end{aligned}$$

Thus, (4.10) is bounded by

$$\begin{aligned} &\frac{\mu}{2}\left\|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\right\|^2 + \left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k - \boldsymbol{w}_k\right\rangle - \tau\cdot\left(f\left(\boldsymbol{w}_k\right) - f\left(\boldsymbol{w}_*\right)\right) \\ &- \frac{\tau\mu}{2}\left\|\boldsymbol{u}_k - \boldsymbol{w}_*\right\|^2 - \frac{\tau\mu}{2}\left\|\boldsymbol{w}_k - \boldsymbol{u}_k\right\|^2 \end{aligned}$$

which gives with $c = 1 - \tau$

$$\begin{aligned}\frac{\mu}{2}\left\|\boldsymbol{u}_{k+1}-\boldsymbol{w}_*\right\|^2 \leq c\frac{\mu}{2}\left\|\boldsymbol{u}_k-\boldsymbol{w}_*\right\|^2+\frac{\mu}{2}\left\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\right\|^2 \\ +\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle-\tau \cdot\big(f\left(\boldsymbol{w}_k\right)-f\left(\boldsymbol{w}_*\right)\big)-\frac{\tau\mu}{2}\left\|\boldsymbol{w}_k-\boldsymbol{u}_k\right\|^2 .\end{aligned} \tag{4.12}$$

To bound the first term in (4.9), we use L-smoothness and (4.7)

$$f\left(\boldsymbol{v}_{k+1}\right)-f\left(\boldsymbol{w}_k\right) \leq\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_{k+1}-\boldsymbol{w}_k\right\rangle+\frac{L}{2}\left\|\boldsymbol{v}_{k+1}-\boldsymbol{w}_k\right\|^2=-\frac{1}{2L}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2,$$

so that

$$\begin{aligned}f\left(\boldsymbol{v}_{k+1}\right)-f\left(\boldsymbol{w}_*\right)-\tau \cdot\big(f\left(\boldsymbol{w}_k\right)-f\left(\boldsymbol{w}_*\right)\big) &\leq(1-\tau)\big(f\left(\boldsymbol{w}_k\right)-f\left(\boldsymbol{w}_*\right)\big)-\frac{1}{2L}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2 \\ &=c \cdot\big(f\left(\boldsymbol{v}_k\right)-f\left(\boldsymbol{w}_*\right)\big)+c \cdot\big(f\left(\boldsymbol{w}_k\right)-f\left(\boldsymbol{v}_k\right)\big)-\frac{1}{2L}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2\end{aligned} \tag{4.13}$$

Now, (4.12) and (4.13) imply

$$\begin{aligned}e_{k+1} \leq& c e_k+c \cdot\big(f\left(\boldsymbol{w}_k\right)-f\left(\boldsymbol{v}_k\right)\big)-\frac{1}{2L}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2+\frac{\mu}{2}\left\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\right\|^2 \\ &+\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle-\frac{\tau\mu}{2}\left\|\boldsymbol{w}_k-\boldsymbol{u}_k\right\|^2\end{aligned}$$

Since we wish to bound e_{k+1} by ce_k, we now show that all terms except ce_k on the right-hand side of the inequality above sum up to a non-positive value. By (4.8) and (4.11)

$$\frac{\mu}{2}\left\|\boldsymbol{u}_{k+1}-\boldsymbol{u}_k\right\|^2=\frac{\mu}{2}\left\|\boldsymbol{v}_k-\boldsymbol{w}_k\right\|^2-\tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle+\frac{\tau^2}{2\mu}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2$$

Moreover, using μ-strong convexity

$$\begin{aligned}&\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle \\ &\quad \leq \tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle+(1-\tau)\left(f\left(\boldsymbol{v}_k\right)-f\left(\boldsymbol{w}_k\right)-\frac{\mu}{2}\left\|\boldsymbol{v}_k-\boldsymbol{w}_k\right\|^2\right).\end{aligned}$$

Thus, we arrive at

$$\begin{aligned}e_{k+1} \leq& c e_k+c \cdot\big(f\left(\boldsymbol{w}_k\right)-f\left(\boldsymbol{v}_k\right)\big)-\frac{1}{2L}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2+\frac{\mu}{2}\left\|\boldsymbol{v}_k-\boldsymbol{w}_k\right\|^2 \\ &-\tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle+\frac{\tau^2}{2\mu}\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2+\tau\left\langle\nabla f\left(\boldsymbol{w}_k\right), \boldsymbol{v}_k-\boldsymbol{w}_k\right\rangle \\ &+c \cdot\big(f\left(\boldsymbol{v}_k\right)-f\left(\boldsymbol{w}_k\right)\big)-c\frac{\mu}{2}\left\|\boldsymbol{v}_k-\boldsymbol{w}_k\right\|^2-\frac{\tau\mu}{2}\left\|\boldsymbol{w}_k-\boldsymbol{u}_k\right\|^2 \\ =& c e_k+\left(\frac{\tau^2}{2\mu}-\frac{1}{2L}\right)\left\|\nabla f\left(\boldsymbol{w}_k\right)\right\|^2+\frac{\mu}{2}\left(\tau-\frac{1}{\tau}\right)\left\|\boldsymbol{w}_k-\boldsymbol{v}_k\right\|^2 \\ \leq& c e_k\end{aligned}$$

where we used once more (4.11), and the fact that $\tau^2/(2\mu) - 1/(2L) = 0$ and $\tau - 1/\tau \leq 0$ since $\tau = \sqrt{\mu/L} \in (0, 1]$. □

Comparing the result for gradient descent with NAG, the improvement lies in the convergence rate, which is $1-\kappa^{-1}$ for gradient descent, and $1-\kappa^{-1/2}$ for NAG, where $\kappa = L/\mu$. In contrast to gradient descent, for NAG the convergence depends only on the square root of the condition number κ. For ill-conditioned problems where κ is large, we therefore expect much better performance for accelerated methods.

Finally, we mention that NAG also achieves faster convergence in the case of L-smooth and convex objective functions. While the error decays like $O\left(k^{-1}\right)$ for gradient descent, for NAG one obtains convergence $O\left(k^{-2}\right)$, see [Nes83, Nes13, WRJ21].

Chapter 5

Adaptive step-sizes

A major issue with the stochastic gradient method is that the step-size sequence should be determined beforehand and has no reason to be adapted the problem at stake. The following proposition seeks to address this issue.

Proposition 5 ([SZL13])**.** *Consider $x \in \mathbb{R}^d$ and a function $f : \mathbb{R}^d \times \Xi \to \mathbb{R}$ such that for any $\xi, (x \mapsto f(x,\xi))$ is differentiable with an L-Lipschitz gradient. Denote $x^+(\gamma) = x - \gamma \nabla f(x,\xi)$, where $\gamma \in \mathbb{R}^d_+$ does not depend on ξ.*

$$\inf_{\gamma \in \mathbb{R}^d_+} \mathbb{E}\left[f\left(x^+(\gamma), \xi\right)\right] \leq \mathbb{E}\left[f\left(x^+\left(\gamma^*\right), \xi\right)\right] \leq f(x) - \frac{1}{2L}\sum_{j=1}^{d} \frac{\left(\mathbb{E}\left[\nabla_j f(x,\xi)\right]^4\right)}{\mathbb{E}\left[\nabla_j f(x,\xi)^2\right]},$$

where $\gamma_j^* = \frac{1}{L}\frac{\left(\mathbb{E}\left[\nabla_j f(x,\xi)\right]\right)^2}{\mathbb{E}\left[\nabla_j f(x,\xi)^2\right]}$.

Proof. Denote $F(x) = \mathbb{E}[f(x,\xi)]$. By the Taylor Lagrange inequality

$$F\left(x^+(\gamma)\right) \leq F(x) - \langle \nabla F(x), \gamma \nabla f(x,\xi)\rangle + \frac{L}{2}\|\gamma \nabla f(x,\xi)\|^2$$

$$\mathbb{E}\left[F\left(x^+(\gamma)\right)\right] \leq F(x) - \sum_{j=1}^{d} \gamma_j \left(\nabla_j F(x)\right)^2 + \frac{L}{2}\sum_{j=1}^{d} \gamma_j^2 \mathbb{E}\left[\nabla_j f(x,\xi)^2\right].$$

Minimizing the right-hand side with respect to γ yields the result. □

This proposition shows that if we knew the law of ξ, we could design step sizes for the stochastic gradient method that would ensure a favorable decay of the objective function. Moreover, these step sizes would be adaptive to the local behavior of the function and decrease to 0 at the optimal rate. However, we cannot set step sizes as required by Proposition 5 because the law of ξ is not necessarily known.

In this chapter, we will study algorithms with adaptive step sizes: Adagrad, RMSProp and Adam. They even go beyond the previous proposition, by defining step sizes that depend on the whole history of stochastic gradients, ξ_{k+1} included.

For any three vectors $a, a', b \in \mathbb{R}^d$, we define

$$ab = \begin{pmatrix} a_1 b_1 \\ \vdots \\ a_d b_d \end{pmatrix}, \qquad \frac{a}{b} = \begin{pmatrix} a_1/b_1 \\ \vdots \\ a_d/b_d \end{pmatrix}, \qquad \|a\|_b^2 = \sum_{i=1}^{n} a_i^2 b_i, \qquad \langle a, a'\rangle_b = \sum_{i=1}^{d} a_i a_i' b_i.$$

if these quantities exist.

5.1 Adagrad

Adagrad has been introduced in [DHS11]. The algorithm is

```
Algorithm 9: Adagrad
    1. Pick $x_0 \in \mathbb{R}^d$.
    2. Until termination condition, iterate:
            Generate $\xi_{k+1}$ independent of the past
            $g_{k+1} = \begin{pmatrix} g_{k+1}(1) \\ \vdots \\ g_{k+1}(d) \end{pmatrix} = \nabla f_x\left(x_k, \xi_{k+1}\right)$
            $\gamma_{k+1}(j) = \frac{\alpha}{\sqrt{\sum_{s=0}^{k} g_{s+1}^2(j)}}, \ \forall j = 1, \ldots, d$
            $x_{k+1} = x_k - \gamma_{k+1} g_{k+1}$
```

Here, we defined $\gamma_k g_k = \begin{pmatrix} \gamma_k(1) g_k(1) \\ \vdots \\ \gamma_k(d) g_k(d) \end{pmatrix}$. The convergence guarantees of Adagrad are given in the theorem below.

Theorem 7. *Suppose that*

1. *$f(\cdot, \xi)$ is convex for all ξ*
2. *There exists $x^* \in \arg\min F$, where $F(x) = \mathbb{E}[f(x, \xi)]$*
3. *There exists $D > 0$ s.t. for all $k \geq 0$, for all $i \in \{1, \ldots, d\} : \left|x_{k,i} - x_i^*\right| \leq D$*
4. *For all $x, \xi : \|\nabla f(x, \xi)\| \leq G$.*

Then the iterates of Adagrad satisfy

$$\mathbb{E}\left[F\left(\bar{x}_K\right) - F\left(x^*\right)\right] \leq \frac{dG}{\sqrt{K}}\left(\frac{D^2}{\alpha} + 2\alpha\right)$$

where $\bar{x}_K = \frac{1}{K}\sum_{k=0}^{K-1} x_k$.

Proof of Theorem 7. Since the step sizes are no longer deterministic, more care is required when taking conditional expectations. Still, the proof will begin with similar arguments as in Theorem 2.2.

$$\begin{aligned} f\left(x_k, \xi_{k+1}\right) - f\left(x^*, \xi_{k+1}\right) &\leq \left\langle g_{k+1}, x_k - x^*\right\rangle \\ &\leq \frac{1}{2}\left\|x_k - x^*\right\|_{\gamma_{k+1}^{-1}}^2 - \frac{1}{2}\left\|x_{k+1} - x^*\right\|_{\gamma_{k+1}^{-1}}^2 + \frac{1}{2}\left\|g_{k+1}\right\|_{\gamma_{k+1}}^2 . \end{aligned}$$

To justify the last inequality, we can write

$$\frac{1}{2}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}} - \frac{1}{2}\Big\| \underbrace{x_{k+1} - x_k}_{=-\gamma_{k+1}g_{k+1}} + x_k - x^*\Big\|^2_{\gamma^{-1}_{k+1}} + \frac{1}{2}\left\|g_{k+1}\right\|^2_{\gamma_{k+1}}$$

$$= \cancel{\frac{1}{2}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}}} - \frac{1}{2}\left\|\gamma_{k+1}g_{k+1}\right\|^2_{\gamma^{-1}_{k+1}} - \cancel{\frac{1}{2}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}}} - \left\langle -\gamma_{k+1}g_{k+1}, x_k - x^*\right\rangle_{\gamma^{-1}_{k+1}} + \frac{1}{2}\left\|g_{k+1}\right\|^2_{\gamma_{k+1}}$$

$$= -\frac{1}{2}\left(\cancel{\sum_{j=1}^{d}\left(\gamma_{k+1}(j)g_{k+1}(j)\right)^2\gamma^{-1}_{k+1}(j)}\right) - \left(\sum_{j=1}^{d}\left(-\gamma_{k+1}(j)g_{k+1}(j)\right)\left(x_k(j) - x^*(j)\right)\gamma^{-1}_{k+1}(j)\right)$$

$$+\frac{1}{2}\left(\cancel{\sum_{j=1}^{d}\left(g_{k+1}(j)\right)^2\gamma_{k+1}(j)}\right)$$

$$= \left\langle g_{k+1}, x_k - x^*\right\rangle.$$

We now sum for k between 0 and $K-1$

$$\sum_{k=0}^{K-1} f\left(x_k, \xi_{k+1}\right) - f\left(x^*, \xi_{k+1}\right) \le \sum_{k=0}^{K-1}\left(\frac{1}{2}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}} - \frac{1}{2}\left\|x_{k+1} - x^*\right\|^2_{\gamma^{-1}_{k+1}}\right) + \sum_{k=0}^{K-1}\frac{1}{2}\left\|g_{k+1}\right\|^2_{\gamma_{k+1}}. \tag{5.1}$$

Note that we have not taken any expectations yet. The difference of the norms is nearly telescoping. We can use the following property of the norms

$$\|x\|^2_a - \|x\|^2_b = \sum_{i=1}^{d} x_i^2 a_i - \sum_{i=1}^{d} x_i^2 b_i = \sum_{i=1}^{d} x_i^2(a_i - b_i) = \|x\|^2_{a-b}, \quad \forall x, a, b \in \mathbb{R}^d.$$

We obtain

$$\sum_{k=0}^{K-1}\left(\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}} - \left\|x_{k+1} - x^*\right\|^2_{\gamma^{-1}_{k+1}}\right) = \sum_{k=0}^{K-1}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}} - \sum_{k=1}^{K}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k}}$$

$$= \left\|x_0 - x^*\right\|^2_{\gamma_1^{-1}} - \left\|x_K - x^*\right\|^2_{\gamma_K^{-1}} + \sum_{k=1}^{K-1}\left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k+1}} - \left\|x_k - x^*\right\|^2_{\gamma^{-1}_{k}}$$

$$= \left\|x_0 - x^*\right\|^2_{\gamma_1^{-1}} \underbrace{- \left\|x_K - x^*\right\|^2_{\gamma_K^{-1}}}_{\le 0} + \sum_{k=1}^{K-1}\left\|x_k - x^*\right\|^2_{\left(\gamma^{-1}_{k+1} - \gamma^{-1}_{k}\right)}$$

$$\le \sum_{j=1}^{d}\frac{\left|x_0(j) - x^*(j)\right|^2}{\gamma_1(j)} + \sum_{j=1}^{d}\sum_{k=1}^{K-1}\left|x_k(j) - x^*(j)\right|^2\left(\frac{1}{\gamma_{k+1}(j)} - \frac{1}{\gamma_k(j)}\right)$$

$$\le \sum_{j=1}^{d}\left[\frac{D^2}{\gamma_1(j)} + \sum_{k=1}^{K-1} D^2\left(\frac{1}{\gamma_{k+1}(j)} - \frac{1}{\gamma_k(j)}\right)\right] = \sum_{j=1}^{d}\frac{D^2}{\gamma_K(j)}$$

$$\le \frac{D^2 d G\sqrt{K}}{\alpha}. \tag{5.2}$$

In the second to last inequality, we used the fact that $\frac{1}{\gamma_{k+1}} - \frac{1}{\gamma_k} \ge 0$. In the last inequality, we used $\gamma_K(j) \ge \frac{\alpha}{G\sqrt{K}}$. This is due to the assumption that $\|\nabla f(x, \xi)\| \le G, \forall x \in \mathbb{R}^d, \forall \xi \in \Xi$, which ensures the desired lower bound on $\gamma_K(j)$ as justified below:

$$\gamma_K(j) = \frac{\alpha}{\sqrt{\sum_{s=0}^{K-1} g_{s+1}^2(j)}} \geq \frac{\alpha}{G\sqrt{K}}.$$

We now control the second part of (5.1): $\sum_{k=0}^{K-1} \|g_{k+1}\|_{\gamma_{k+1}}^2$. By definition of γ_k, if we denote $a_k^{(i)} = g_{k+1,i}^2 \geq 0$, then

$$\sum_{k=0}^{K-1} \|g_{k+1}\|_{\gamma_{k+1}}^2 = \alpha \sum_{k=0}^{K-1} \sum_{i=1}^{d} \frac{a_k^{(i)}}{\sqrt{\sum_{s=0}^{k} a_s^{(i)}}}.$$

Lemma 3. *Let (a_k) be a sequence of nonnegative numbers. Then*

$$\sum_{k=0}^{K-1} \frac{a_k}{\sqrt{\sum_{s=0}^{k} a_s}} \leq 2\sqrt{\sum_{s=0}^{K-1} a_s}$$

Proof. Denote $h_K = \sum_{k=0}^{K-1} \frac{a_k}{\sqrt{\sum_{s=0}^{k} a_s}}$. We will show the result by induction. Clearly, $h_1 = \sqrt{a_0} \leq 2\sqrt{a_0}$.

We now assume that $h_K \leq 2\sqrt{\sum_{s=0}^{K-1} a_s}$.

$$h_{K+1} = h_K + \frac{a_K}{\sqrt{\sum_{s=0}^{K} a_s}} \leq 2\sqrt{\sum_{s=0}^{K-1} a_s} + \frac{a_K}{\sqrt{\sum_{s=0}^{K} a_s}}.$$

Now, since the square root is concave, we have

$$\sqrt{b-a} \leq \sqrt{b} - \frac{a}{2\sqrt{b}}$$

as long as $b - a \geq 0$ and $b > 0$. Hence,

$$h_{K+1} \leq 2\left(\sqrt{\sum_{s=0}^{K} a_s} - \frac{a_K}{2\sum_{s=0}^{K} a_s}\right) + \frac{a_K}{\sqrt{\sum_{s=0}^{K} a_s}} = 2\sqrt{\sum_{s=0}^{K} a_s}.$$

By induction, the lemma is proved. □

We apply the lemma to our sequence of stochastic gradients to get:

$$\sum_{k=0}^{K-1} \|g_{k+1}\|_{\gamma_{k+1}}^2 \leq 2\alpha \sum_{i=1}^{d} \sqrt{\sum_{k=0}^{K-1} \left(g_{k+1}(i)\right)^2} \leq 2\alpha d G\sqrt{K}.$$

We combine the inequality with (5.2) to get

$$\sum_{k=0}^{K-1} f\left(x_k, \xi_{k+1}\right) - f\left(x^*, \xi_{k+1}\right) \leq \frac{dD^2 G\sqrt{K}}{\alpha} + 2\alpha d G\sqrt{K}.$$

Taking the expectation on both sides, and dividing by K, we can write:

$$\begin{aligned}
\frac{dD^2G}{\alpha\sqrt{K}} + 2\frac{\alpha dG}{\sqrt{K}} &\geq \frac{1}{K}\mathbb{E}\left[\sum_{k=0}^{K-1} f\left(x_k, \xi_{k+1}\right) - f\left(x^*, \xi_{k+1}\right)\right] \\
&= \frac{1}{K}\mathbb{E}\left[\sum_{k=0}^{K-1} \mathbb{E}_k\left[f\left(x_k, \xi_{k+1}\right) - f\left(x^*, \xi_{k+1}\right)\right]\right] \qquad \text{using } \mathbb{E}X = \mathbb{E}\big[\mathbb{E}(X|Y)\big]\forall X, Y \\
&= \frac{1}{K}\mathbb{E}\left[\sum_{k=0}^{K-1} F(x_k) - F(x^*)\right] \qquad \text{by definition of } F \\
&\geq \mathbb{E}\left[F(\bar{x}_K) - F(x^*)\right] \qquad \text{by convexity.}
\end{aligned}$$

□

5.2 RMSProp

A significant drawback of the Adagrad algorithm is that its step size, defined as

$$\gamma_{k+1}(j) = \frac{\alpha}{\sqrt{\sum\limits_{s=0}^{k} g_{s+1}^2(j)}}, \ \forall j = 1, \ldots, d$$

is always non-increasing over time:

$$\gamma_{k+1}(j) = \frac{\alpha}{\sqrt{\sum\limits_{s=0}^{k} g_{s+1}^2(j)}} \leq \frac{\alpha}{\sqrt{\sum\limits_{s=0}^{k-1} g_{s+1}^2(j)}} = \gamma_k(j).$$

This can cause severe issues in cases where the gradients g_{s+1} have large magnitudes close to the initialization point, but small magnitude around the targeted minimizer. Specifically, the Adagrad updating rule will aggressively reduce of the step sizes γ_k in early stages of the optimization process, but won't be able to increase them again to speed up convergence when needed in later stages. This can lead to a dramatic slow-down of convergence. To address this issue, the RMSProp algorithm (Root Mean Squared Propagation) slightly modifies the Adagrad algorithm by introducing a *forgetting rule* that gradually downweights old gradients and puts more emphasis on recent ones. More precisely, the RMSProp step sizes are given by

$$\gamma_{k+1} = \frac{\alpha}{\sqrt{(1-\beta)\sum_{s=0}^{k} \beta^{k-s} g_{s+1}^2}}, \tag{5.3}$$

for some $\beta \in (0, 1)$.

This idea is natural, as the oldest gradients are only informative in early stages of the process and might no longer be relevant as we move away from the initialization point. To see why the updating rule above implements this idea, let us consider the denominator in (5.3). The latest gradient g_{k+1}^2 appears in the denominator with the maximum re-weighting factor, equal to $\beta^0 = 1$. In contrast, the earliest gradient g_1^2 is re-weighted by β^k where $\beta < 1$, which decays exponentially fast as the time step k goes to ∞. This forgetting mechanism therefore prevents a naive accumulation of past gradients as was the case in the Adagrad algorithm. In particular, the step size γ_k can either increase or decrease over time, enabling the algorithm to better adjust to the local geometry of the function to optimize.

Note that the denominator in (5.3) can be computed recursively using a one-line update. Namely, letting $v_k = \sum_{s=0}^{k-1} \beta^{k-s} g_{s+1}^2$, we have

$$v_{k+1} = (1-\beta) g_{k+1}^2 + \beta v_k.$$

This allows one to only store one value v_k rather than the whole history $(g_1, \dots, g_k)$. Note also that in practice, a hyperparameter $\epsilon > 0$ of the order of 10^{-8} is introduced in the denominator to ensure we never divide by zero

$$\gamma_{k+1} = \frac{\alpha}{\epsilon + \sqrt{v_{k+1}}}.$$

This parameter generally has a negligible influence on the algorithm's updates in practice, and we will assume $\epsilon = 0$ in this course. The RMSProp algorithm is given as follows.

```
Algorithm 10: RMSProp
    1. Pick $x_0 \in \mathbb{R}^d$ and let $v_0 = 0$.
    2. Until termination condition, iterate:
            Generate $\xi_{k+1}$ independent of the past
            $g_{k+1} = \begin{pmatrix} g_{k+1}(1) \\ \vdots \\ g_{k+1}(d) \end{pmatrix} = \nabla f_x(x_k, \xi_{k+1})$
            $v_{k+1} = \beta v_k + (1-\beta) \nabla f(x_k, \xi_{k+1})^2$
            $\hat{v}_{k+1} = \frac{v_{k+1}}{1-\beta^{k+1}}$ (optional, we can also keep $\hat{v}_{k+1} = v_{k+1}$)
            $\gamma_{k+1}(j) = \frac{\alpha}{\sqrt{\hat{v}_k(j)}}, \forall j = 1, \dots, d$
            $x_{k+1} = x_k - \gamma_{k+1} g_{k+1}$
```

The convergence guarantees of RMSProp will be covered in the exercise and practical sessions.

5.3 Adam

We now introduce an algorithm that is often used when training neural network models: Adam, which stands for stochastic gradient with adaptive moment estimation [KB14]. This algorithm is simply a combination of the momentum RMSProp algorithm, studied earlier in the course. Its main ingredients are an adaptive estimation of the first and second moments of the stochastic gradient and coordinate-wise step sizes. The idea is to design an exponential moving average of previous gradients and square gradients to estimate its moments. Finally, instead of just using the estimate of $\nabla F(x)$ to set the step size, Adam uses it directly as a means of reducing the variance of the stochastic gradient. The algorithm uses parameters $\alpha > 0, \beta_1 \in [0,1]$, $\beta_2 \in [0,1]$ and $\epsilon > 0$. It is initialized with a fixed x_0 and $m_0 = v_0 = 0$. It is given by

Algorithm 11: Adam

```
Inputs: $\alpha > 0,\ \beta_1, \beta_2 \in [0,1],\ \epsilon > 0$.
    1. Pick $x_0 \in \mathbb{R}^d$, and let $m_0 = v_0 = 0$.
    2. Until termination condition, iterate:
        $\xi_{k+1} \sim \mathbb{P}_\xi$    independent of the past
        $m_{k+1} = \beta_1 m_k + (1-\beta_1) \nabla f(x_k, \xi_{k+1})$
        $\hat{m}_{k+1} = \frac{m_{k+1}}{1-\beta_1^{k+1}}$
        $v_{k+1} = \beta_2 v_k + (1-\beta_2) \nabla f(x_k, \xi_{k+1})^2$
        $\hat{v}_{k+1} = \max\left(\hat{v}_k, \frac{v_{k+1}}{1-\beta_2^{k+1}}\right)$
        $x_{k+1} = x_k - \frac{\alpha_k}{\epsilon + \sqrt{\hat{v}_{k+1}}} \hat{m}_{k+1}$.
```

Convergence guarantees of Adam are given in the theorem below.

Theorem 8. *Suppose that*

1. $f(\cdot, \xi)$ *is convex for all* ξ
2. $\exists x^* \in \arg\min F$
3. *For all* k*, for all* i*,* $\left|x_{k,i} - x_i^*\right| \leq D$
4. *For all* x, ξ*, for all* i*,* $\left|\nabla_i f(x,\xi)\right| \leq G$
5. $\alpha_k = \frac{\alpha_0}{\sqrt{k+1}}$
6. $\beta_1^2 < \beta_2 < 1, \epsilon = 0$

Then the iterates of Adam satisfy

$$\begin{aligned}\mathbb{E}\left[F(\bar{x}_K) - F(x^*)\right] &\leq \frac{dD^2}{2(1-\beta_1)\sqrt{1-\beta_2}} \frac{G}{\alpha_0 \sqrt{K}} + \frac{1+2\beta_1}{2(1-\beta_1)} \frac{\alpha_0 \sqrt{1+\ln(K)} G}{\sqrt{1-\beta_2}\sqrt{1-\frac{\beta_1^2}{\beta_2}}\sqrt{K}} \\ &\in O\left(\frac{\sqrt{\ln(K)}}{\sqrt{K}}\right)\end{aligned}$$

where $\bar{x}_K = \frac{1}{K}\sum_{k=0}^{K-1} x_k$.

This theorem will be proved in the exercise session.

5.4 Further reading

For further reading, we refer the interested reader to the excellent lecture notes by Olivier Fercoq [Fer21] `https://cermics.enpc.fr/~leclerev/courses/Saclay/fercoq/poly_optsto_fercoq.pdf` and the book [Bac21].

Part II

Sampling methods

Chapter 6

Sampling methods: Part 1

Notation

We will denote by $\mathbb{N} = \{1, 2, \dots\}$ the set of positive integers and by $\mathrm{Card}(E)$ the cardinality of a set E (i.e. the number of elements it contains). We will also use the following notation for classical probability distributions.

Name	**Notation**	**Density**
Uniform	$\mathrm{Unif}([a,b])$	$f(x) = \frac{1}{b-a},\ x \in [a,b]$
Bernoulli	$\mathrm{Ber}(p)$	$f(x) = p^x(1-p)^{1-x},\ x \in \{0,1\}$
Binomial	$\mathrm{Bin}(n,p)$	$f(k) = \binom{n}{k}p^k(1-p)^{n-k},\ k \in \{0,\dots,n\}$
Exponential	$\mathrm{Exp}(\lambda)$	$f(x) = \mathbf{1}_{x\geq 0}\lambda e^{-\lambda x}$
Laplace	$\mathrm{Lap}(\lambda)$	$f(x) = \frac{\lambda}{2}e^{-\lambda\|x\|},\ x \in \mathbb{R}$
Rademacher	$\mathrm{Rad}(p)$	$\mathbb{P}(X=1) = \mathbb{P}(X=-1) = 1/2$
Geometric	$\mathrm{Geom}(p)$	$f(n) = p(1-p)^{n-1},\ n \in \mathbb{N}$
Cauchy	$\mathrm{Cauchy}(\mu,\sigma)$	$f(x) = \frac{1}{\pi\sigma\left(1+\left(\frac{x-\mu}{\sigma}\right)^2\right)}$

Table 6.1: Summary of Probability Distributions

Remark: There are two conventions for $\mathrm{Geom}(p)$. In this course, we use the convention given above, which is equivalent to $\mathbb{P}(N > n) = (1-p)^n, \forall n \in \mathbb{N}$. Another convention is to consider N taking values in $\mathbb{N} \cup \{0\}$, with for all $n \in \mathbb{N} \cup \{0\} : \mathbb{P}(N = n) = p(1-p)^n$. The second variable simply results from the first by subtracting 1.

6.1 Simulating uniform random variables on $[0, 1]$

Numbers generated by a computer without external data are usually not random, since they are the result of a deterministic program. We aim to simulate numbers that *appear to be* random, that is, that will pass several statistical tests that truly random numbers must satisfy with high probability. They are referred to as pseudo-random numbers. These sequences are generally defined by recurrence (or are deduced from a sequence defined by recurrence). A canonical example is the historical (and simplistic) example of linear congruence generators defined by

$$X_{n+1} = 16807 X_n \pmod{2^{31} - 1}.$$

Here, $2^{-31} X_n$ approximately follows the uniform distribution on $[0, 1]$. The first term X_0 of the sequence is called the seed of the algorithm. There are also ways to introduce "true randomness" when generating pseudo-random numbers by using external elements such as the computer's clock, or physical phenomena (radioactivity, quantum phenomena, etc.) In general, the seed is fixed based on the number of milliseconds of the computer's clock at the moment the simulation starts. This method is only used for initializing the seed, while the following draws are deterministically deduced from it.

We will not focus on the problem of random number generation: We will assume from now on that we can simulate an arbitrarily long sequence of independent, uniform random variables on $[0, 1]$. We will now describe how to simulate variables following other laws.

Example: If $U \sim \mathrm{Unif}([0, 1])$, then $X = \mathbb{1}\{U \leq p\}$ simulates a Bernoulli with parameter p since X equals 0 or 1 and $\mathbb{P}[X = 1] = \mathbb{P}[U \leq p] = p$.

Simulating binomial variables: We use the fact that a binomial distribution with parameter $p \in [0, 1]$ and $n \geq 1$ is a sum of independent Bernoulli variables with parameter p. Thus, to simulate a random variable $X \sim \mathrm{Bin}(n, p)$, we start by drawing n independent uniform random variables $U_1, \dots, U_n$ on $[0, 1]$, then we construct the n Bernoulli variables $Y_k = \mathbb{1}_{[0,p]}(U_k)$, for $1 \leq k \leq n$ and finally set $X = Y_1 + \cdots + Y_n$.

6.2 Simulating random variables on $\mathbb{R}$

6.2.1 Discrete random variables

Proposition 6. *Let $x_1, \cdots, x_n \in \mathbb{R}$ be distinct and $p_1, \cdots, p_n > 0$ such that $p_1 + \cdots + p_n = 1$. We set $s_0 = 0$ and for all $1 \leq k \leq n$, $s_k = p_1 + \cdots + p_k$. Let $U \sim \mathrm{Unif}([0, 1])$ and*

$$X = \sum_{k=1}^{n} x_k \, \mathbb{1}\{s_{k-1} \leq U \leq s_k\}.$$

Then, X is a random variable following a discrete distribution over $\{x_1, \dots, x_n\}$ with probability parameters $(p_1, \dots, p_n)$:

$$\mathbb{P}(X = x_j) = p_j, \quad \forall j \in \{1, \dots, n\}.$$

Proof. For all $i \in \{1, \dots, n\}$, we have $\mathbb{P}(X = x_i) = \mathbb{P}\big(s_{i-1} \leq U \leq s_i\big) = s_i - s_{i-1} = p_i$. □

Discrete random variables

To simulate a discrete distribution over $\{x_1, \dots, x_n\}$ with parameters $(p_1, \dots, p_n)$:

```
$(s_i)_i$ = cumulative sums of $(p_i)_i$;
U = uniform();
i = 1;
while U > $s_i$, i = i + 1;
return $x_i$;
```

6.2.2 Calculating laws

If $X \overset{Law}{=} f(U_1, \ldots, U_n)$ where $U_1, \ldots, U_n \sim \text{Unif}([0,1])$ are iid, then we can simulate X:

Simulate X with law $f(U_1, \ldots, U_n)$
Apply f to uniform random variables $U_1, \ldots, U_n$

Example 1 (Uniform random variable on any interval)
If U is a uniform variable on $[0,1]$, then $a + (b-a)U$ is uniform over $[a,b]$. Indeed, we have $\mathbb{P}\big(a + (b-a)U \leq t\big) = \mathbb{P}\big(U \leq (t-a)/(b-a)\big) = (t-a)/(b-a)$.

Example 2 (Sample variables with density $2x$ over $[0,1]$)
We want to simulate a random variable with values in $[0,1]$ and density $2x$. If U_1 and U_2 are two independent uniform variables on $[0,1]$, then $\max(U_1, U_2)$ has the correct density (calculate the cumulative distribution function). Consequently, it suffices to take the maximum between two uniforms to simulate our random variable.

6.2.3 Change of variable

To compute the law of $f(U_1, \ldots, U_n)$, which is an image measure, it can be useful to use the change of variable formula

$$\int_a^b f(\varphi(t))\varphi'(t)dt = \int_{\varphi(a)}^{\varphi(b)} f(x)dx,$$

which we formally use by setting $x = \varphi(t)$ so that $dx = d(\varphi(t)) = \varphi'(t)dt$.

It will also be useful to recall the polar system of coordinates and polar change of variables. In Cartesian coordinates, any point z in $\mathbb{R}^2$ is represented by its coordinates (x,y) in the Euclidean plane. In the polar system of coordinates, any point z is rather represented by two numbers $(r,\theta) \in \mathbb{R}_+ \times [0, 2\pi)$, where the radius $r = \sqrt{x^2+y^2}$ represents the distance between z and the origin or *pole* $(0,0)$, and the angle θ is the angle between the first canonical vector $e_1 = (1,0)$ and the vector z. We therefore have the relations $(x,y) = (r\cos(\theta), r\sin(\theta))$. We can then use this relation to perform a change of variables in integrals:

Proposition 7 (Polar change of variables)**.** *For any integrable function $f : \mathbb{R}^2 \to \mathbb{R}$, we have*

$$\int_{\mathbb{R}^2} f(x,y)dxdy = \int_{\mathbb{R}_+ \times [0,2\pi)]} f\big(r\cos(\theta), r\sin(\theta)\big) r dr d\theta.$$

This change of variables is performed by replacing any occurrence of x by $r\cos(\theta)$ and any occurrence of y by $r\sin(\theta)$ and the differential $dxdy$ by $rdrd\theta$.

Example 3: If V is uniform over $[0, \pi/2]$, let us compute the law of $\sin^2(V)$. For any bounded measurable function g, we have

$$\begin{aligned}
\mathbb{E}[g(\sin^2(V))] &= \frac{2}{\pi}\int_0^{\pi/2} g(\sin^2(\theta))d\theta \\
&= \frac{2}{\pi}\int_0^1 g(x)d(\arcsin(\sqrt{x})) \\
&= \frac{2}{\pi}\int_0^1 g(x)\frac{1}{2\sqrt{x}}\frac{1}{\sqrt{1-\sqrt{x}^2}}dx
\end{aligned}$$

$$= \int_0^1 g(x) \frac{1}{\pi\sqrt{x(1-x)}} dx.$$

We deduce that if U is uniform over $[0,1]$, then $\sin^2(\pi U/2)$ follows the law on $[0,1]$, with density $x \mapsto 1/(\pi\sqrt{x(1-x)})$, that is the beta distribution with parameters 1/2 and 1/2. This distribution is also called *arcsine law*, because its cumulative distribution function involves the function $x \mapsto \arcsin x$.

Example 4: (Simulating uniform random variables on the unit disk).
In Cartesian coordinates, the uniform measure on the unit disk can be given by the expression $(1/\pi)\mathbb{1}_{x^2+y^2\leq 1} dx\,dy$, or in polar coordinates $(1/\pi)\mathbb{1}_{r\leq 1} r\,dr\,d\theta$. By setting $s = r^2$, we obtain $(1/2\pi)\mathbb{1}_{s\leq 1} ds\,d\theta$ since $ds = 2r\,dr$. Thus, for (x,y) uniform over the unit disk, $s = r^2$ and θ are independent and uniformly distributed respectively over $[0,1]$ and $[0,2\pi)$.

Proposition 8. *If U and V are two independent uniform variables on $[0,1]$, then*

$$\Big(\sqrt{U}\cos(2\pi V), \sqrt{U}\sin(2\pi V)\Big)$$

is uniform over the unit disk.

6.2.4 Mixtures of distributions

Let N, X_1, X_2 be three independent real random variables. Assume that N takes its values in $\{1,2\}$, and that X_1 and X_2 have densities. Let us compute the law of X_N.

Denote by f_1 and f_2 the densities of X_1 and X_2, respectively. For any bounded measurable function g, since $g(X_N) = g(X_1)\,\mathbb{1}_{N=1} + g(X_2)\,\mathbb{1}_{N=2}$, we have

$$\begin{aligned}
\mathbb{E}\big[g(X_N)\big] &= \mathbb{E}\big[g(X_1)\,\mathbb{1}_{N=1} + g(X_2)\,\mathbb{1}_{N=2}\big] \\
&= \mathbb{E}\big[\mathbb{1}_{N=1}\big]\mathbb{E}\big[g(X_1)\big] + \mathbb{E}\big[\mathbb{1}_{N=2}\big]\mathbb{E}\big[g(X_2)\big] \\
&= \mathbb{P}\big[N=1\big]\int_{\mathbb{R}} g(x) f_1(x) dx + \mathbb{P}\big[N=2\big]\int_{\mathbb{R}} g(x) f_2(x) dx \\
&= \int_{\mathbb{R}} g(x)\left(\alpha_1 f_1(x) + \alpha_2 f_2(x)\right) dx.
\end{aligned}$$

which shows that X_N has the density $x \mapsto \alpha_1 f_1(x) + \alpha_2 f_2(x)$. More generally, we have

Proposition 9. *If $N, X_1, \cdots, X_n$ are real random variables with N taking values in $\{1, \cdots, n\}$ with weights $\alpha_1, \ldots, \alpha_n \geq 0$ and independent of the others, then the random variable X_N follows the law*

$$\alpha_1 P_{X_1} + \ldots + \alpha_n P_{X_n}.$$

To simulate a random variable with law μ, where $\mu = \alpha_1\mu_1 + \cdots + \alpha_n\mu_n$, with $\alpha_1, \ldots, \alpha_n \geq 0$, $\alpha_1 + \cdots + \alpha_n = 1$, one can proceed as follows:

Simulating Mixtures of Distributions
We simulate a realization k of N taking values in $\{1, \cdots, n\}$ with probabilities $\alpha_1, \ldots, \alpha_n$, then simulate X with law μ_k.

For example, consider the law μ on $\mathbb{R}$ with density ρ relative to the Lebesgue measure, where $\rho(x) = e^{-x^2/2}/\sqrt{8\pi}$ if $x \notin [0,1]$ and $\rho(x) = (1/2) + e^{-x^2/2}/\sqrt{8\pi}$ if $x \in [0,1]$. The density ρ of μ is the half-sum of the uniform law on $[0,1]$ and the standard normal distribution. It follows that if Z_1, Z_2, J are independent variables, respectively uniform over $[0,1]$, standard normal, and uniform on $\{1,2\}$, then Z_J follows the law μ.

We can generalize the previous proposition for an infinite mixture of distributions:

Proposition 10. *If $(X_t)_{t\in\mathbb{R}}$ are real random variables with densities f_t, and if T is a real random variable with density g, and independent from $(X_t)_{t\in\mathbb{R}}$, then the random variable X_T has the following density*

$$x \mapsto \int_{\mathbb{R}} f_t(x)g(t)dt.$$

To simulate a random variable with density

$$x \mapsto \int_{\mathbb{R}} f_t(x)g(t)dt,$$

one can proceed as follows:

Simulating Continuous Mixtures of Distributions
We simulate a realization t of T, then simulate X with density f_t.

6.3 Inversion of the cumulative distribution function (CDF)

Let μ be a probability law on $\mathbb{R}$ (equipped with the Borel σ-algebra), and F its cumulative distribution function:

$$F(x) := \mu((-\infty, x]).$$

We recall that F is non-decreasing, right-continuous, has a left limit at every point, and tends toward 0 at $-\infty$ and toward 1 at $+\infty$.

Definition 2. *For $t \in (0,1)$, we define the **pseudo-inverse** G of F (also called the quantile function of μ) by*

$$G(t) := \inf\{x \in \mathbb{R}, F(x) \geq t\} < \infty.$$

In particular, we note that $G(t) < \infty$ for any $t \in (0,1)$ since $F(t) \to 1$ when $t \to \infty$ and $F(t) \to 0$ when $t \to -\infty$. We collect some useful properties below

Proposition 11.

1. *$G(t) \leq x \Leftrightarrow F(x) \geq t$*
2. *If F restricted to $]a,b[$ establishes a bijection from $]a,b[$ to $]0,1[$, (with inverse F^{-1}), then $G = F^{-1}$ on $]0,1[$.*

Proof.

1. By definition of G, if we have $F(x) \geq t$ then $G(t) \leq x$. Conversely, if $G(t) \leq x$, then for every $\varepsilon > 0$, we have $G(t) < x + \varepsilon$, which implies that $t \leq F(x+\varepsilon)$. As F is right-continuous, we can let ε tend to 0 and obtain $t \leq F(x)$.
2. For $t \in]0,1[$, $F^{-1}(t)$ is the unique real number x such that $F(x) = t$, and by monotonicity, it is therefore the smallest real number x such that $F(x) \geq t$. Thus, $F^{-1}(t) = G(t)$ by definition of G.

□

Proposition 12. *If U is a uniform random variable on $[0,1]$, then the random variable $G(U)$ follows the law μ.*

Proof. If U is uniform over $[0,1]$, U is almost surely in $(0,1)$, hence the random variable $G(U)$ is well-defined. We can now compute the CDF of the variable $G(U)$.

$$\mathbb{P}[G(U) \le x] = \mathbb{P}[U \le F(x)] = F(x).$$

Therefore, $G(X)$ follows the law μ. □

Inversion of the cumulative distribution function (CDF)
Consider a distribution on $\mathbb{R}$

- *We calculate its cumulative distribution function F,*
- *We calculate the function G*
- *We apply G to uniform random variables $G(U_k)$*

Corollary 1. *If $(X_n)_{n\ge1}$ are independent, uniform over $[0,1]$, then $(G(X_n))_{n\ge1}$ are independent random variables uniformly following the law μ.*

Proof. If the same measurable function is applied to a sequence of i.i.d. random variables, a sequence of i.i.d. random variables is obtained. As the function G is measurable, the result is immediate. □

The method presented here applies to any distribution with values in $\mathbb{R}$, however, it requires inverting the cumulative distribution function, which can be cumbersome (for example, for a Gaussian variable). We will see certain specific cases where this method can be used.

6.3.1 Example: Simulation of exponential variables

Proposition 13. *If U_n is a uniform variable on $[0,1]$, and $\lambda > 0$, then $-(1/\lambda)\log(U_n)$ is a sequence of independent random variables with an exponential distribution of parameter λ.*

Proof. We recall that the cumulative distribution function of an exponential variable is

$$F(t) = 1 - e^{-\lambda t}, \quad t \ge 0.$$

Thus, if we invert the relation we obtain

$$u = 1 - e^{-\lambda t} \Leftrightarrow t = -\frac{1}{\lambda}\log(1-u).$$

Hence, we deduce that if U follows a uniform distribution then $-1/\lambda \log(1-U)$ follows an exponential distribution. If U follows a uniform law on $[0,1]$ then so does $1-U$. The proposition follows. □

Alternatively, we verify that for any $x \ge 0$,

$$\mathbb{P}\Big[-\frac{1}{\lambda}\log(X_n) \le x\Big] = \mathbb{P}\Big[X_n \ge e^{-\lambda x}\Big] = 1 - \exp\big(-\lambda x\big),$$

which proves that $-(1/\lambda)\log(X_n)$ is an exponential variable of parameter λ.

6.4 Rejection Method

The rejection method aims to simulate a random variable X whose density f is difficult to sample from. It proceeds by drawing samples from an auxiliary density g and accepting or rejecting them. The density g should possess the following desirable properties

1. It is easy to simulate Y with density g
2. There exists a constant $c > 0$ such that $f \leq cg$
3. It is easy to evaluate $\dfrac{f(x)}{cg(x)}$ for any $x \in \mathbb{R}^d$.

Consider now two independent sequences of random variables:

(a) $(Y_n)_{n\geq 1}$ i.i.d. with density g;

(b) $(U_n)_{n\geq 1}$ i.i.d. with uniform distribution on $[0,1]$.

Rejection Method
Simulate $(Y_n)_n$ iid with density g and $(U_n)_n$ iid uniform multiple times and output the first Y_n for which $cU_ng(Y_n) \leq f(Y_n)$.

More formally, output Y_N where the random index N is defined by

$$N = \min\left\{n \geq 0 : \; U_n \leq \frac{f(Y_n)}{cg(Y_n)}\right\}.$$

Concretely, we keep on sampling Y_n and U_n until the condition $cU_ng(Y_n) \leq f(Y_n)$ is met, and discard all the samples that failed to satisfy this condition. A rejection therefore represents a waste of computational power; to optimize the algorithm's speed, it is essential to minimize the number of rejections by reducing the constant c.

A variable Y corresponds to a proposed sample and U to a coin toss to decide whether to accept or reject this proposal. We denote by r the acceptance ratio function for the coin toss, namely

$$r(y) = \frac{f(y)}{cg(y)} \quad \text{if } g(y) > 0 \qquad \text{and} \qquad r(y) = 0 \text{ otherwise.}$$

Furthermore, since f and g are densities, we necessarily have $c \geq 1$:

$$1 = \int_{\mathbb{R}} f(x)dx \leq \int_{\mathbb{R}} cg(x)dx = c\int_{\mathbb{R}} g(x)dx = c$$

The following result guarantees that the rejection method generates samples with density f.

Proposition 14 (Rejection Algorithm)**.** *Let $N = \inf\left\{n \geq 1, U_n \leq r\left(Y_n\right)\right\}$ be the first time a proposed sample is accepted. Then*

1. $N \sim \mathrm{Geom}(1/c)$. In particular, $N < \infty$ almost surely, therefore the random variable $X = Y_N$ is well defined almost surely.

2. The variable X has density f and the variables N and X are independent.

Remark: We recall that the geometric distribution has been defined in the Notation Section.

Proof. We start by showing that N is a.s. finite. The random variable N takes values in $\mathbb{N}\cup\{+\infty\}$. For all $n \in \mathbb{N}$, the fact that the pairs (Y_i, U_i) are i.i.d. yields that

$$\mathbb{P}(N > n) = \mathbb{P}\left(U_1 > r\left(Y_1\right), \ldots, U_n > r\left(Y_n\right)\right) = \mathbb{P}\left(U_1 > r\left(Y_1\right)\right)^n.$$

Since the variables Y_1 and U_1 are independent, their joint distribution is the product of the marginals. By the Fubini-Tonelli theorem (see Theorem 1 in the Appendix) we obtain

$$\mathbb{P}\left(U_1 > r\left(Y_1\right)\right) = \mathbb{E}\left[\mathbb{1}_{U_1 > r(Y_1)}\right] = \int_{\mathbb{R}} \left(\int_0^1 \mathbb{1}_{u > r(y)} du\right) g(y) dy,$$

and since g and f are densities, we can further simplify this expression as

$$P\left(U_1 > r\left(Y_1\right)\right) = \int_{\mathbb{R}} (1 - r(y)) g(y) dy = 1 - \int_{\mathbb{R}} r(y) g(y) dy = 1 - \frac{1}{c},$$

which yields

$$\mathbb{P}(N > n) = \left(1 - \frac{1}{c}\right)^n.$$

This shows that N follows a geometric distribution with parameter $1/c$ and in particular that this variable is a.s. finite. The variable $X = Y_N$ is therefore a.s. well defined. Now we denote by F the distribution function associated with the density f. Then,

$$\mathbb{P}(X \leq x, N = n) = \mathbb{P}\left(U_1 > r\left(Y_1\right), \ldots, U_{n-1} > r\left(Y_{n-1}\right), U_n \leq r\left(Y_n\right), Y_n \leq x\right)$$

Since the n draws are i.i.d., this can be rewritten as

$$\mathbb{P}(X \leq x, N = n) = \mathbb{P}\left(U_1 > r\left(Y_1\right)\right)^{n-1} \mathbb{P}\left(U_n \leq r\left(Y_n\right), Y_n \leq x\right).$$

For the first term, the previous inequalities give

$$\left(\mathbb{P}\left(U_1 > r\left(Y_1\right)\right)\right)^{n-1} = \left(1 - \frac{1}{c}\right)^{n-1}.$$

For the second term, a comparable calculation yields

$$\mathbb{P}\left(U_n \leq r\left(Y_n\right), Y_n \leq x\right) = \mathbb{E}\left[\mathbb{1}_{U_n \leq r(Y_n)} \mathbb{1}_{Y_n \leq x}\right] = \int_{\mathbb{R}} \left(\int_0^1 \mathbb{1}_{u \leq r(y)} du\right) \mathbb{1}_{y \leq x} g(y) dy$$

that is,

$$\mathbb{P}\left(U_n \leq r\left(Y_n\right), Y_n \leq x\right) = \int_{\mathbb{R}} \mathbb{1}_{y \leq x} r(y) g(y) dy = \int_{-\infty}^x r(y) g(y) dy = \frac{1}{c} \int_{-\infty}^x f(y) dy = \frac{F(x)}{c}.$$

Therefore, we obtain

$$\mathbb{P}(X \leq x, N = n) = \left(1 - \frac{1}{c}\right)^{n-1} \times \frac{F(x)}{c}.$$

By sigma-additivity, we have

$$\mathbb{P}(X \leq x) = \sum_{n=1}^{\infty} \mathbb{P}(X \leq x, N = n) = \frac{F(x)}{c} \sum_{n=1}^{\infty} \left(1 - \frac{1}{c}\right)^{n-1} = F(x),$$

which proves that X indeed has the desired law. Finally, the fact that $\mathbb{P}(X \leq x, N = n) = \mathbb{P}(X \leq x)\mathbb{P}(N = n)$ clearly shows the independence of the variables X and N.

□

Remark: since $N \sim \mathrm{Geom}(1/c)$, its expectation is c. Consequently, it takes on average c trials to obtain a single sample from the target distribution f. Therefore, the pair (g, c) should be chosen so that c is as close to 1 as possible or equivalently so that the density g resembles f as closely as possible.

Toy Example. Suppose we want to simulate X with density $f(x) = 3x^2$ on $[0, 1]$. Since $f(x) \leq 3$, we can choose g as the uniform law on $[0, 1]$ and take $c = 3$, which gives the acceptance ratio

$$r(y) = \frac{f(y)}{cg(y)} = y^2.$$

The algorithm proceeds by generating a pair of independent uniform variables (Y_1, U_1). If $U_1 \leq Y_1^2$, we accept Y_1 as a valid sample for X; otherwise, we reject the sample and repeat the process. On average, three attempts are required to generate a single sample from f. This example is given for illustration purposes only: In this specific case, it would be more efficient to use the inversion method. By setting $X = U^{1/3}$ where U is uniformly distributed over $[0, 1]$, we can directly sample from the desired distribution f.

To adapt Proposition 14 to higher dimensions, it suffices to replace each occurrence of $\mathbb{P}(X \leq x, N = n)$ (resp. $\mathbb{P}(X \leq x)$) with $\mathbb{P}(X \in A, N = n)$ (resp. $\mathbb{P}(X \in A)$) for any Borel set A of $\mathbb{R}$. In other words, the rejection method remains valid if f and g are densities on $\mathbb{R}^d$.

Example (simulation of a conditional law).
Let B be a Borel set contained in the square $[0, 1]^2$. Suppose we want to simulate points uniformly in B. The set B can be complicated, but we assume that we can easily evaluate the indicator function $\mathbb{1}_B(x)$ for any $x \in [0, 1]^2$ (that is, we can easily *check* whether x belongs to B or not).
Applying the rejection algorithm amounts to generating points Y_n uniformly in $[0, 1]^2$ until the first time N when one of them falls into B. By the preceding result, the variable $X = Y_N$ is then uniform over B. Indeed, denoting by $\lambda(B)$ the Lebesgue measure of B (i.e., its area) and defining $C = [0, 1]^2$, we have

$$f(x, y) = \frac{\mathbb{1}_B(x, y)}{\lambda(B)} \text{ and } g(x, y) = \mathbb{1}_C(x, y) \Rightarrow c = \frac{1}{\lambda(B)} \text{ and } r(x, y) = \mathbb{1}_B(x, y).$$

Notably, we do not need to know the constant c to apply the rejection algorithm!

Generalization. The previous example's underlying idea can be generalized to other important settings: Suppose that $f(x) = c_1 f_u(x)$ where c_1 is an unknown normalization constant and f_u can be evaluated at every point (the index "u" standing for "unnormalized"). Such a case frequently arises in Bayesian statistics and statistical physics. We assume there exists a density g that is easy to sample from and to evaluate at every point, and verifying $f_u(x) \leq c_2 g(x)$ with a *known* constant $c_2 > 0$. Then one can apply the rejection method with $c = c_1 c_2$: Even if c is unknown, it is possible to evaluate $r(y) = f(y)/(cg(y))$ since it holds that $r(y) = f_u(x)/\big(c_2 g(y)\big)$, which we can compute. This technique is frequently used in some Monte Carlo methods such as the Metropolis-Hastings algorithm, which we will study later on.

Simulation of Uniform Variables on the Unit Disk: We will use the rejection method, observing that the uniform law on the unit disk is the uniform law on the square $[-1, 1]^2$, conditioned by the variable taking its values on the disk.
The uniform law on $[-1, 1]^2$ is easily simulated by taking two independent variables (X, Y), uniformly distributed over $[-1, 1]$. In this example, the auxiliary law μ is written

$$\mu(dx, dy) = \frac{1}{4} \mathbb{1}_{[-1,1]}(x)\, \mathbb{1}_{[-1,1]}(y) dx dy$$

and $\eta = \mu(\cdot|A)$ where

$$A = \{(x^2 + y^2 \leq 1)\}.$$

Thus, we reject with probability $1 - \pi/4 \approx 0.21$, which corresponds to around 21% of the proposed samples.

6.5 Simulating Gaussian Random Variables

Definition 3. *A standard normal random variable is a real random variable with density*

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

A normal random variable is a random variable $X = \mu + \sigma Z$ where $\mu, \sigma \in \mathbb{R}$, and Z is a standard normal random variable. We denote $X \sim \mathcal{N}(\mu, \sigma^2)$. If $\sigma \neq 0$, then the density of X is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

It can be tricky to simulate a standard normal variable using the inversion method since the cumulative distribution function of $N(0,1)$ does not have a simple expression in terms of "usual" functions. In this Section, we provide several methods to simulate one-dimensional Gaussian random variables and Gaussian vectors.

6.5.1 Using the rejection method

To apply the rejection method, we can choose the Laplace distribution with parameter λ as an auxiliary law, whose density with respect to the Lebesgue measure given by

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}.$$

We can easily simulate this distribution by outputting a product of an exponential variable with parameter $\lambda > 0$ (studied above) by a Rademacher random variable with parameter $1/2$ (i.e. a uniform variable on $\{-1, 1\}$). Indeed,

$$\begin{aligned}\mathbb{E}(\phi(sE)) &= \frac{1}{2}\mathbb{E}(\phi(-E)) + \frac{1}{2}\mathbb{E}(\phi(E)) \\ &= \frac{1}{2}\int_0^\infty \phi(-x)\lambda e^{-\lambda x} dx + \frac{1}{2}\int_0^\infty \phi(x)\lambda e^{-\lambda x} dx.\end{aligned}$$

Thus, we obtain the ratio of densities

$$\rho(x) = \frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}} \frac{1}{\lambda} e^{\lambda|x| - x^2/2},$$

whose maximum is attained for $x = \pm\lambda$ (to see this, study the logarithm of ρ). We recall that c should be an upper bound on ρ to apply the rejection method. Therefore, we must have $c \geq \sqrt{\frac{2}{\pi}} \frac{e^{\lambda^2/2}}{\lambda}$, which suggests choosing $c = \sqrt{\frac{2}{\pi}} \frac{e^{\lambda^2/2}}{\lambda}$.

To optimize the method, c should be taken as small as possible to minimize the probability of rejecting a sample, equal to $1 - 1/c$. The optimal choice of λ that minimizes $c = \sqrt{\frac{2}{\pi}} \frac{e^{\lambda^2/2}}{\lambda}$ is $\lambda = 1$, yielding $c \approx 1.31$ and a rejection probability of about 24%.

6.5.2 Using the Box-Müller method

Proposition 15 (Box and Müller Method)**.** *If U, V are independent uniforms on $[0,1]$, then the random variables defined by*

$$\sqrt{-2\log(U)}\cos(2\pi V) \text{ and } \sqrt{-2\log(U)}\sin(2\pi V)$$

are two independent standard normal random variables.

Proof. We show that if (X, Y) is a pair of independent standard normal variables, then

$$(X,Y) \overset{\text{(d)}}{=} (\sqrt{-2\log(U)}\cos(2\pi V), \sqrt{-2\log(U)}\sin(2\pi V)),$$

where U and V are independent uniform random variables on $[0,1]$. To do so, we will use successive changes of variables in the law of (X,Y):

$$(1/2\pi)e^{-(x^2+y^2)/2}dxdy.$$

In polar coordinates (see Proposition 7), this becomes

$$(1/2\pi)e^{-r^2/2}rdrd\theta,$$

or setting $s = r^2/2$,

$$(1/2\pi)e^{-s}dsd\theta.$$

We recognize the law of two independent variables: $S \sim \text{Exp}(1)$ and $\theta \sim \text{Unif}([0,2\pi])$. To conclude, it suffices to recall that an exponential variable can be obtained from a uniform variable U by $-\log(U)$. Using the change of variable $x = r\cos(\theta) = \sqrt{2s}\cos(\theta)$ and $y = \sqrt{2s}\sin(\theta)$, we deduce that if $U, V \sim \text{Unif}([0,1])$ are independent, then $\sqrt{-2\log(U)}\cos(2\pi V)$ and $\sqrt{-2\log(U)}\sin(2\pi V)$ are two independent standard normal variables. □

6.5.3 Gaussian random variables in higher dimensions

Recall that to simulate a standard normal random variable, one can use the Box-Müller method: If $U, V \sim \text{Unif}([0,1])$ are independent, then the random variables defined by

$$\sqrt{-2\log(U)}\cos(2\pi V) \text{ and } \sqrt{-2\log(U)}\sin(2\pi V)$$

are two independent standard normal random variables. Then, by multiplying by σ and adding μ, we obtain a normal random variable with mean μ and variance σ^2.

Definition 4. *A random vector $X = (X_1, \dots, X_n)$ is a* ***Gaussian vector*** *if any linear combination of its components is a Gaussian random variable. The mean μ of X and its covariance matrix Σ characterize the law of X, and we write $X \sim \mathcal{N}(\mu, \Sigma)$.*

We recall that the mean vector μ is

$$\mathbb{E}[X] = \begin{pmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_n] \end{pmatrix},$$

and the covariance matrix Σ is the $n \times n$ matrix

$$\begin{aligned}\mathbb{V}ar(X) &= \mathbb{E}\left[(X - \mathbb{E}[X]) \cdot (X - \mathbb{E}[X])^T\right] \\ &= \mathbb{E}\left[\begin{pmatrix} X_1 - \mathbb{E}[X_1] \\ \vdots \\ X_n - \mathbb{E}[X_n] \end{pmatrix} \cdot \begin{pmatrix} X_1 - \mathbb{E}[X_1], \cdots, X_n - \mathbb{E}[X_n] \end{pmatrix}\right],\end{aligned}$$

(i.e., $\Sigma_{i,j} = \mathrm{cov}(X_i, X_j)$).

Warning: A Gaussian vector is not simply a vector with Gaussian components (called "marginals"). For example, the following vector has Gaussian marginals but is not a Gaussian vector:

$$(X, bX) \quad \text{for } X \sim N(0,1) \text{ and } b \sim \mathrm{Rad}(1/2) \text{ such that } b \perp\!\!\!\perp X.$$

This vector is *not* a Gaussian vector since the sum of its coordinates is $(1+b)X$ and takes the value 0 with probability $1/2$ (in fact, $(1+b)X$ follows the mixture distribution $\frac{1}{2}\delta_0 + \frac{1}{2}N(0,4)$).

To show that a vector is a Gaussian vector, one can:

- Express it as a linear combination of other Gaussian vectors,
- Calculate its density: If $\det(\Sigma) \neq 0$, the density of X with respect to the Lebesgue measure over $\mathbb{R}^n$ is
$$f(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left(-\frac{(x-\mu)^T \Sigma^{-1}(x-\mu)}{2}\right),$$
where $(x-\mu)$ and $(x-\mu)^T$ denote a column vector and a row vector, respectively.
- Another possibility is to calculate the characteristic function. The characteristic function of a Gaussian vector X with mean μ and covariance matrix Σ is given by
$$\varphi_X(z) = \exp(i\langle z, \mu\rangle - \langle z, \Sigma z\rangle),$$
which is the main tool to show the equality of distributions.

Proposition 16. *1. If $X \sim \mathcal{N}(\mu, \Sigma)$ (column-wise), then $AX + b \sim \mathcal{N}(A\mu + b, A\Sigma A^T)$.*

2. If $X \sim \mathcal{N}(\mu, \Sigma)$, then for all $1 \leq i, j \leq n$, X_i and X_j are independent if and only if $\Sigma_{i,j} = 0$.

Proof. It suffices to calculate the characteristic function. □

Thus, one can first simulate independent Gaussian variables to obtain a vector $X \sim \mathcal{N}(0, I_n)$. Then, one can use the Cholesky decomposition of Σ: Since Σ is symmetric and non-negative, there exists a lower-triangular matrix A such that $AA^T = \Sigma$. It suffices to multiply by A and we obtain $\mu + AX \sim \mathcal{N}(\mu, AA^T)$.

6.6 Monte Carlo Method

The Monte Carlo method is a canonical method to approximate the expectation $\mathbb{E}[X]$ of an (integrable) random variable X. It proceeds by simulating a large number N of i.i.d. random variables $X_1, \ldots, X_N$ with the same law as X, and applying the law of large numbers:

$$\text{If } \mathbb{E}|X| < \infty, \text{ then } \quad \frac{X_1 + \ldots + X_N}{N} \underset{n\to\infty}{\longrightarrow} \mathbb{E}[X] \;\; a.s.$$

Similarly, for a measurable function φ such that $\varphi(X)$ is integrable, the random variables $\varphi(X_1), \ldots, \varphi(X_N)$ are i.i.d. with the same law as $\varphi(X)$, so the law of large numbers ensures that

$$\frac{\varphi(X_1) + \ldots + \varphi(X_N)}{N} \underset{N\to\infty}{\longrightarrow} \mathbb{E}[\varphi(X)] \quad a.s.$$

In particular, for a measurable set A, the previous equality applied to the indicator function $\varphi = \mathbb{1}_A$ gives us that

$$\frac{1}{N}\text{Card}\left(\{n \text{ between 1 and } N \text{ such that } X_n \in A\}\right) \underset{N\to\infty}{\longrightarrow} \mathbb{E}[\varphi(X)] = \mathbb{P}[X \in A] \quad a.s.$$

(Here, $\text{Card}(E)$ denotes the number of elements in a set E). This aligns with the intuition of a probability $\mathbb{P}[X \in A]$: It is the proportion of X falling into A when drawing X a large number of times.

Monte Carlo Method:
We simulate $X_1, \ldots, X_N \sim X$ i.d.d. Then

$$\frac{\varphi(X_1) + \ldots + \varphi(X_N)}{N}$$

is an approximation of $\mathbb{E}[\varphi(X)]$.

Example: Suppose we know the measure $|B|$ of a domain $B \subset \mathbb{R}^n$ and aim to estimate the measure $|A|$ of a domain $A \subset B$. We can consider a point X uniformly distributed in B and observe that

$$\mathbb{P}[X \in A] = |A|/|B|.$$

To estimate $|A|$, it is therefore sufficient to estimate $\mathbb{P}[X \in A]$ which is done by calculating the proportion

$$\frac{1}{N}\text{Card}\left(\{n \text{ between 1 and } N \text{ such that } X_n \in A\}\right)$$

for X_n points drawn uniformly in $|B|$.

6.6.1 Confidence Interval

Definition 5 (Confidence interval)**.** *Suppose we have*

1. *A family of probability distributions $\{\mathbb{P}_\theta : \theta \in \Theta\}$, for $\Theta \subseteq \mathbb{R}$,*
2. *A real number $\alpha \in (0, 1)$*
3. *A sample $X_1, \ldots, X_n \overset{iid}{\sim} \mathbb{P}_\theta$ for some unknown $\theta \in \Theta$.*

A random interval $\big[a(X_1, \ldots, X_n),\, b(X_1, \ldots, X_n)\big]$ is a confidence interval for θ of level $1 - \alpha$ if

$$\mathbb{P}_\theta\Big(\big[a(X_1, \ldots, X_n),\, b(X_1, \ldots, X_n)\big] \ni \theta\Big) = 1 - \alpha,$$

where $a(X_1, \ldots, X_n)$ and $b(X_1, \ldots, X_n)$ only depend on the observations $(X_1, \ldots, X_n)$.
It is an asymptotic confidence interval for θ with asymptotic level $1 - \alpha$ if

$$\mathbb{P}_\theta\Big(\big[a(X_1, \ldots, X_n),\, b(X_1, \ldots, X_n)\big] \ni \theta\Big) \underset{n\to\infty}{\longrightarrow} 1 - \alpha.$$

To obtain an asymptotic confidence interval, we can control the estimation error using the central limit theorem. If $\varphi(X)^2$ is integrable, then

$$\sqrt{N}\left(\frac{\varphi(X_1)+\ldots+\varphi(X_N)}{N}-\mathbb{E}[\varphi(X)]\right)$$

converges in distribution to a centered normal random variable with variance $\sigma^2 = \mathrm{Var}(\varphi(X))$. Specifically,

$$\mathbb{P}\left[\left|\frac{\varphi(X_1)+\ldots+\varphi(X_N)}{N}-\mathbb{E}[\varphi(X)]\right| \le \frac{a\sigma}{\sqrt{N}}\right] \underset{N\to\infty}{\longrightarrow} \mathbb{P}\left[|Z|\le a\right],$$

where Z is a standard normal. This leads to the following confidence interval: With probability $\mathbb{P}\left[|Z|\le a\right]$, the true value $\mathbb{E}[\varphi(X)]$ asymptotically belongs to the random interval centered at $\frac{1}{N}\big(\varphi(X_1)+\ldots+\varphi(X_N)\big)$ with half-width $\frac{a\sigma}{\sqrt{N}}$.
When the standard deviation σ is unknown, we can estimate or at least upper bound it and use the estimated value to construct a confidence interval. For instance, to estimate a probability $\mathbb{P}[X\in A]$, the variance is that of a Bernoulli, which can always be bounded by $1/4$. It follows that, with asymptotic probability at least $\mathbb{P}\left[|Z|\le a\right]$, the true value $\mathbb{P}[X\in A]$ belongs to the random interval centered at

$$\frac{1}{N}\mathrm{Card}\Big(\{n \text{ between 1 and } N \text{ such that } X_n\in A\}\Big)$$

and with a half-width of $\frac{a}{2\sqrt{N}}$.

Example: Coming back to the previous example, where the area $|A|$ of a domain $A\subset B\subset\mathbb{R}^n$ is estimated, an asymptotic confidence interval for $\mathbb{P}[X\in A]=|A|/|B|$ is centered at

$$\frac{1}{N}\mathrm{Card}\Big(\{n \text{ between 1 and } N \text{ such that } X_n\in A\}\Big)$$

and with a half-width of $\frac{0.98}{\sqrt{N}}$. To convert this into a confidence interval for $|A|$, it suffices to multiply it by $|B|$.

6.7 Importance Sampling

Importance sampling is a refinement of the Monte Carlo method. Its aim is still to estimate

$$I=\mathbb{E}[\varphi(X)]=\int\varphi(x)f(x)dx.$$

Such an integral can be difficult to approximate using Monte Carlo if the function φ takes large values where the density f is very low since it is unlikely to observe a random variable X with density f in such regions. In this case, the classical Monte Carlo estimator

$$\hat{I}_n=\frac{1}{n}\sum_{i=1}^{n}\varphi\left(X_i\right)$$

does not perform well: Unless n is very large, we will estimate I by a value close to 0 even if I is large.

Examples:

1. *Rare events*: Suppose the variable X follows a standard normal distribution, and we want to estimate the probability

$$\mathbb{P}(X > 6) = 1 - \Phi(6) = \mathbb{E}\left[\mathbb{1}_{X>6}\right] = \int_{\mathbb{R}} \mathbb{1}_{x>6} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \approx 10^{-9}.$$

   Unless n is at least of the order of a billion, we will likely observe $\hat{I}_n = 0$.

2. A more subtle example: Let m be a real number, $X \sim \mathcal{N}(m, 1)$, and $\varphi(x) = \exp\left(-mx + m^2/2\right)$. For all m, we have

$$I = \mathbb{E}[\varphi(X)] = \int_{\mathbb{R}} \varphi(x) f(x) dx = \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx = 1.$$

   However, around 95% of the X_i's fall within the interval $[m-2, m+2]$ while $\varphi(m) = \exp\left(-m^2/2\right)$ quickly tends to 0 as m increases. Thus, as m increases, $\hat{I}_n$ decays to 0, whereas the true value of the integral is $I = 1$ regardless of the value of m. The Monte-Carlo estimator therefore becomes unreliable as $m \to \infty$ and should be replaced with a more accurate estimator.

Importance sampling is a modification of the Monte Carlo method where points are simulated according to an auxiliary density g rather than the density f of X. The density g is chosen to achieve a compromise between the regions of space where φ is large and where the density f is high. To account for the fact that the simulation law is g rather than f, it then suffices to introduce a correction factor in the final empirical average.

Mathematically, it is simply a rewriting of I in the form

$$I = \mathbb{E}[\varphi(X)] = \int \varphi(x) f(x) dx = \int \frac{f(y)}{g(y)} \varphi(y) g(y) dy = \int w(y) \varphi(y) g(y) dy = \mathbb{E}[w(Y)\varphi(Y)],$$

where Y has density g and $w(y) = f(y)/g(y)$ is a re-weighting (or likelihood ratio) accounting for the change of law. We need to guarantee that we do not divide by 0, or equivalently, that $g(y) > 0$ as soon as $f(y)\varphi(y) > 0$. The auxiliary density g should satisfy the following requirements:

(a) It is easy to simulate according to the density g,

(b) It is easy to calculate the likelihood ratio $w(y) = f(y)/g(y)$ for any y.

The importance sampling estimator is given by

$$\tilde{I}_n = \frac{1}{n} \sum_{i=1}^{n} w\left(Y_i\right) \varphi\left(Y_i\right),$$

where the Y_i are i.i.d. with density g.

Proposition 17 (Importance Sampling). *If $\mathbb{E}|w(Y)\varphi(Y)| < \infty$, then the estimator $\tilde{I}_n$ is unbiased and convergent, meaning that $\mathbb{E}\left[\tilde{I}_n\right] = I$ and*

$$\tilde{I}_n = \frac{1}{n} \sum_{i=1}^{n} w\left(Y_i\right) \varphi\left(Y_i\right) \xrightarrow[n \to \infty]{a.s.} \mathbb{E}[w(Y)\varphi(Y)] = I.$$

Furthermore, if $\mathbb{E}\left[w(Y)^2\varphi(Y)^2\right] < \infty$, *then*

$$\sqrt{n}(\tilde{I}_n - I) \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}(0, s^2),$$

where

$$s^2 = Var(w(Y)\varphi(Y)) = \int w(y)^2\varphi(y)^2 g(y)dy - I^2 = \mathbb{E}\left[w(X)\varphi(X)^2\right] - I^2$$

Note. As before, the variance s^2 is naturally estimated by

$$\tilde{s}_n^2 = \frac{1}{n}\sum_{i=1}^{n} w\left(Y_i\right)^2 \varphi\left(Y_i\right)^2 - \tilde{I}_n^2$$

from which one can deduce asymptotic confidence intervals.

The variance s^2 of $\tilde{I}_n$ has to be compared with the variance $\sigma^2 = \mathbb{E}\left[\varphi^2(X)\right] - I^2$ of $\hat{I}_n$. Therefore, it "suffices" to choose the weighting w, that is, the instrumental density g, so that the term $\mathbb{E}\left[w(X)\varphi^2(X)\right]$ is as small as possible. This problem has an explicit solution. Unfortunately, this solution involves the quantity we are trying to approximate and cannot be used in practice.

Proposition 18 (Optimal Sampling Law)**.** *For any density g s.t.* $\mathbb{E}\left[w(Y)^2\varphi(Y)^2\right] < \infty$, *we have*

$$s^2 = Var\big(w(Y)\varphi(Y)\big) \geq \mathbb{E}[|\varphi(X)|]^2 - I^2 = \left(\int |\varphi(x)|f(x)dx\right)^2 - \left(\int \varphi(x)f(x)dx\right)^2,$$

the lower bound being reached for the density $g^\star$ defined by

$$g^\star(y) = \frac{|\varphi(y)|f(y)}{\int |\varphi(y)|f(y)dy}.$$

Proof. Since any random variable has a non-negative variance, it follows that

$$s^2 = \mathbb{E}\left[w(Y)^2\varphi(Y)^2\right] - I^2 = \mathbb{E}\left[(w(Y)|\varphi(Y)|)^2\right] - I^2 \geq \mathbb{E}[w(Y)|\varphi(Y)|]^2 - I^2 = \mathbb{E}[|\varphi(X)|]^2 - I^2.$$

Equality in the preceding inequality only occurs if

$$\mathbb{E}\left[(w(Y)|\varphi(Y)|)^2\right] = \mathbb{E}[w(Y)|\varphi(Y)|]^2 \quad \text{i.e.} \quad \mathbb{V}\left[w(Y)|\varphi(Y)|\right] = 0,$$

in other words, if the random variable $w(Y)|\varphi(Y)|$ is almost surely constant. Since the variable Y has density g, there exists a constant c such that for any y satisfying $g(y) > 0$, we have

$$w(y)|\varphi(y)| = c \iff g(y) = \frac{|\varphi(y)|f(y)}{c} \iff g(y) = \frac{|\varphi(y)|f(y)}{\int |\varphi(y)|f(y)dy},$$

the last equivalence resulting from the fact that g is a density. □

If φ has a constant sign, the variance obtained using the proposal density $g^\star$ is zero, which means that a single draw from the density $g^\star$ exactly yields the value of I! Indeed, if $Y \sim g^\star$, then

$$\tilde{I}_1 = w(Y)\varphi(Y) = \frac{f(Y)}{g^\star(Y)}\varphi(Y) = \int \varphi(x)f(x)dx = I.$$

Of course, there are two obstacles. First, it is not guaranteed that one can sample from $g^\star$. Second, even if we could, the likelihood ratio $w(Y) = I/\varphi(Y)$ involves the value of I which we are precisely trying to approximate.

Even though the previous proposition is mainly of theoretical interest, it demonstrates that the auxiliary density g must strike a compromise between φ and f: The density g should ideally take large values where the product $|\varphi(x)|f(x)$ is highest.

Chapter 7

Markov chain Monte Carlo (MCMC) methods

7.1 Reminders on Markov Chains

Let $(X_n)_n$ be a sequence of random variables taking values in a finite set $E = \{1, 2, \ldots, d\}$, called the state space.

Definition 6. *We say that (X_n) is a homogeneous Markov chain if for all $n \geq 1$ and any sequence $(x_0, x_1, \ldots, x_{n-1}, x, y)$ from E such that $\mathbb{P}(X_0 = x_0, \ldots, X_{n-1} = x_{n-1}, X_n = x) > 0$, the following equalities holds:*

$$\begin{aligned}\mathbb{P}\left(X_{n+1} = y \mid X_0 = x_0, \ldots, X_{n-1} = x_{n-1}, X_n = x\right) &\overset{(1)}{=} \mathbb{P}\left(X_{n+1} = y \mid X_n = x\right)\\ &\overset{(2)}{=} \mathbb{P}\left(X_1 = y \mid X_0 = x\right).\end{aligned}$$

In other words, conditional on the present, the next step is independent of the past. Said differently, any information about the past is unnecessary to predict the future state given the current state. A general Markov chain only needs to satisfy equality (1). A *homogeneous* Markov chain needs to satisfy (1) and (2). The probability of transitioning from state x to state y is then called the transition probability

$$P(x, y) = \mathbb{P}\left(X_1 = y \mid X_0 = x\right)$$

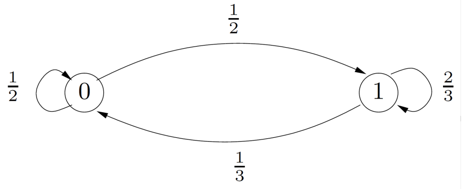
and the matrix of transition probabilities for the chain is the $d \times d$ matrix $P = [P(x, y)]_{1 \leq x, y \leq d}$, which satisfies the following properties:

- Coefficient bounds: $\forall (x, y) \in E^2,\ 0 \leq P(x, y) \leq 1$.
- Row sums: For all $x \in E$, we have $\sum_{y \in E} P(x, y) = 1$.

Furthermore, specifying the initial distribution, i.e. $\mathbb{P}(X_0 = x)$ for all $x \in E$, the joint distribution of the random vector $(X_0, \ldots, X_n)$ can be expressed in terms of the transition probabilities $P(x, y)$ since we have:

$$\begin{aligned}\mathbb{P}(X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) &= \mathbb{P}(X_0 = x_0)\,\mathbb{P}\left(X_1 = x_1 \mid X_0 = x_0\right) \ldots \mathbb{P}\left(X_n = x_n \mid X_{n-1} = x_{n-1}\right)\\ &= \mathbb{P}(X_0 = x_0)\,P(x_0, x_1) \ldots P(x_{n-1}, x_n).\end{aligned}$$

We can also associate a transition graph to any Markov chain as follows: The vertices of the graph are the states of E, and we draw an edge labeled $P(x, y)$ from x to y if $P(x, y) > 0$. This representation is convenient when E is not too large, or when the matrix P is very sparse (i.e. it only has a few non-zero coordinates, or equivalently, each state is connected with only a few other states).



The probabilities of going from one state to another in exactly n steps are entirely determined by the transition matrix. This is known as the Chapman-Kolmogorov equations, given below.

Notation. The probability of going from state x to state y in exactly n steps is denoted as:

$$P(x,y)^{(n)} = \mathbb{P}\left(X_n = y \mid X_0 = x\right)$$

and the n-step transition matrix is defined as:

$$P^{(n)} = \left[P(x,y)^{(n)}\right]_{(x,y)\in E^2}.$$

We also use the convention that $P^{(0)} = I_d$, where I_d denotes the identity matrix of size $|E|$.

Proposition 19 (Chapman-Kolmogorov Equations)**.** *For all $n \geq 0$, the n-step transition matrix is the n-th power of the Markov chain's transition matrix, that is:*

$$P^{(n)} = P^n.$$

Remark. It follows that for any pair of natural numbers (n_1, n_2):

$$P^{(n_1+n_2)} = P^{n_1+n_2} = P^{n_1} \times P^{n_2} = P^{(n_1)} \times P^{(n_2)}.$$

This equation is often referred to as the Chapman-Kolmogorov relation: Going from x to y in $(n_1 + n_2)$ steps amounts to going from x to some x' in n_1 steps, and from x' to y in n_2 steps.

Notation. The initial state X_0 may also be random. We denote the distribution of X_0 as a row vector of size $|E| = d$:

$$\mu_0 = \left[\mu_0(1), \dots, \mu_0(d)\right] = \left[\mathbb{P}\left(X_0 = 1\right), \dots, \mathbb{P}\left(X_0 = d\right)\right].$$

Similarly, we will denote the distribution of X_n as a row vector:

$$\mu_n = \left[\mathbb{P}\left(X_n = 1\right), \dots, \mathbb{P}\left(X_n = d\right)\right]$$

Corollary 2 (Marginal Distribution of the Chain)**.** *Let (X_n) be a Markov chain with initial distribution μ_0 and transition matrix P. Then for any $n \geq 0$, the distribution of X_n is:*

$$\mu_n = \mu_0 P^n.$$

Proof. The relation is clear for $n = 0, 1$. Now suppose that for some $n \in \mathbb{N}$, we have $P^{(n)} = P^n$. Then for any $x, y \in E$

$$\begin{aligned} P^{(n+1)}(x,y) &= \mathbb{P}(X_{n+1} = y \,|\, X_0 = x) \\ &= \sum_{z \in E} \mathbb{P}(X_{n+1} = y \,|\, X_n = z, X_0 = x)\mathbb{P}(X_n = z \,|\, X_0 = x) \\ &= \sum_{z \in E} P(z,y)P^n(x,z) \\ &= P^{n+1}(x,y), \end{aligned}$$

so the property also holds for $n + 1$. □

For a sequence of random variables (X_n) taking values in the finite set E, the convergence in distribution simply corresponds to the convergence of the row vector μ_n of size d, that is, to the convergence of each of its d components. Since $\mu_n = \mu_0 P^n$, a sufficient condition for the convergence in distribution of (X_n) is therefore that the sequence (P^n) converges.

Ideally, we would like the distribution of (X_n) to converge to a distribution independent of the initial distribution μ_0, a phenomenon known as "forgetting the initial condition". Let's mention two pathological situations:

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2 \tag{7.1}$$

In the first case, we have $P^{2n} = I_2$ and $P^{2n+1} = P$, so neither (P^n) nor $\mu_n = \mu_0 P^n$ converge, except in the very particular case of $\mu_0 = [1/2, 1/2]$. This problem is due to a periodicity phenomenon.

In the second case, we have $\mu_n = \mu_0 Q^n = \mu_0$, which trivially converges, but the initial distribution is not forgotten over time. This time, a problem of communication between states arises.

To avoid such issues, we will focus on irreducible and aperiodic chains.

Definition 7. *(Irreducible and aperiodic Markov chains)*

1. *A chain is said to be irreducible if all states communicate with each other: One can go from any x to any y in E in finitely many steps.*

$$\forall (x,y), \exists n = n(x,y), \quad P^n(x,y) > 0.$$

2. *A chain is said to be aperiodic if*

$$\forall x \in E, \quad d(x) = \text{GCD}\left\{n \geq 1, P^n(x,x) > 0\right\} = 1.$$

The notation GCD stands for the greatest common divisor. The quantity $d(x) \in \mathbb{N} \cup \{\infty\}$ is called the period of state x. It can be shown that when a chain is irreducible, all states have the same period. Consequently, for an irreducible chain, a sufficient condition for aperiodicity is that there exists a state on which it can loop, i.e. an index x such that $P(x,x) > 0$. For an irreducible chain, a criterion for aperiodicity is as follows:

$$\forall (x,y) \in E^2, \ \exists n_0 = n_0(x,y), \ \forall n \geq n_0, : P^n(x,y) > 0.$$

Notation:

- A distribution π over E is written as a row vector, and a function $\varphi : E \to \mathbb{R}$ as a column vector. The quantity $\pi\varphi$ consequently represents the expectation of the random variable $\varphi(X)$ when X follows the distribution π:

$$\pi\varphi = \sum_{x\in E} \pi(x)\varphi(x) = \mathbb{E}[\varphi(X)].$$

- The total variation distance between two probability distributions μ and ν on E is:

$$\|\nu - \mu\|_{TV} = \frac{1}{2}\sum_{x\in E} |\nu(x) - \mu(x)|.$$

Definition 8 (Invariant probability measure)**.** *Let π be a probability measure over E and P be a transition matrix. The measure π is said to be an invariant (or stationary, or equilibrium) measure if $\pi = \pi P$.*

Suppose we start from a probability distribution μ over E at time 0. It is easy to see that the measure over E at time 1 is μP, that is, the law of X_1. Thus, being an invariant measure π (i.e. satisfying $\pi P = \pi$) means that if the initial measure is $\mu_0 = \pi$, then the measure μ_n remains equal to π at any time n: π does not change over time (hence the term "equilibrium law"). More precisely, starting from this measure at a given time, everything that leaves state x is equal to everything that arrives at state x.

Theorem 9 (Convergence of Irreducible and Aperiodic Chains)**.** *Suppose (X_n) is an irreducible Markov chain with transition matrix P over a finite state space E. Then*

1. *There exists a unique invariant probability measure π for this chain, that is $\pi P = \pi$. This measure is such that $\pi(x) > 0$ for every x.*

2. *For any function $\varphi : E \to \mathbb{R}$, regardless of the distribution of X_0,*

$$\frac{1}{n}\sum_{k=1}^{n} \varphi(X_k) \xrightarrow[n\to\infty]{a.s.} \pi\varphi.$$

3. *Furthermore, if $(X_n)_n$ is aperiodic, then the distribution μ_n of X_n converges to π at a geometric rate: there exist $C > 0$ and $\alpha \in (0,1)$ such that, for any initial distribution μ_0,*

$$\|\mu_n - \pi\|_{TV} \leq C|\alpha|^n.$$

Proof of Theorem 1. The proof of this theorem goes beyond this course's scope. The interested reader is encouraged to study the references below.

1. [Fre17], Theorem 3.3.

2. [Nor98] Theorem 1.10.2.

3. [Fre17], Theorem 4.9, or [LP17] Theorem 4.9.

□

Although we have an exponentially fast convergence rate, the parameter $|\alpha|$ can be arbitrarily close to 1 and C arbitrarily large if the state space E is unfavorably large. In such a case, this result then has no practical implication.

Remarks:

1. The terms *invariant distribution, stationary distribution*, or *equilibrium distribution* are used interchangeably. Assuming the chain is irreducible, the invariant distribution π is unique and defined as follows: Letting $T_x^+ = \inf\{n > 0, X_n = x\}$ denote the first return time of the chain to x, we have the following expression for this stationary distribution:
$$\pi(x) = \frac{1}{\mathbb{E}[T_x^+ | X_0 = x]}$$
(The proof of this claim can be found in [LP17], Proposition 1.19).

2. The ergodic theorem for the law of large numbers applies in the case of Markov chains. Following the previous remark, the equilibrium measure π can be interpreted as the proportion of time spent by a trajectory in each state. For example, take $\varphi = \mathbf{1}_{x_0}$ where x_0 is a fixed state, and apply the ergodic theorem:
$$\frac{1}{n}\sum_{k=1}^{n} \mathbf{1}_{x_0}(X_k) = \frac{|\{k \in \{1, \ldots, n\}, X_k = x_0\}|}{n} \xrightarrow[n\to\infty]{a.s.} \pi\mathbf{1}_{x_0} = \pi(x_0). \tag{7.2}$$

3. Coming back to the example of the transition matrix P in (7.1), which is irreducible, the unique stationary law is $\pi = [1/2, 1/2]$. According to the theorem, this chain thus verifies a law of large numbers, namely
$$\frac{1}{n}\sum_{k=1}^{n} \varphi(X_k) \xrightarrow[n\to\infty]{a.s.} \frac{1}{2}(\varphi(1) + \varphi(2)),$$
but certainly no convergence in distribution. The reason is intuitively clear: Averaging over a trajectory of the chain eliminates the periodicity phenomenon, which is not the case with convergence in distribution.

Definition 9 (Reversibility)**.** *Let π be a probability measure and (X_n) a Markov chain with transition matrix P. The chain is said to be reversible for π if it satisfies*
$$\pi(x)P(x,y) = \pi(y)P(y,x) \quad \forall (x,y) \in E \times E.$$

Interpretation: Suppose the initial measure is some probability measure π. The quantity $P(x,y)$ represents the *proportion* of the mass $\pi(x)$ moving from x to y between time 0 and time 1. The condition defining reversibility means that under the law π, any pair of states x and y exchange exactly the same amount of mass at any time. In particular, this probability measure will not change over time and is therefore invariant (why?).

Lemma 4 (Reversibility ⇒ Stationarity)**.** *Let (X_n) be a Markov chain with transition matrix P. If the chain is reversible for the measure π, then π is invariant, i.e., $\pi P = \pi$.*

Proof. If the chain is reversible for the measure π, then for any y

$$(\pi P)(y)=\sum_{x\in E}\pi(x)P(x,y)=\pi(y)\sum_{x\in E}P(y,x)=\pi(y).$$

and π is indeed stationary.
Among Markov chains, reversible chains thus constitute a privileged framework of study. We will focus our attention on them in the remainder of the course.
Remark: (Time Reversibility). The term reversibility comes from the following phenomenon: If $X_0\sim\pi$, then we can write the joint law as follows

$$\begin{aligned}\mathbb{P}\left(X_0=x_0,X_1=x_1,\ldots,X_n=x_n\right)&=\pi\left(x_0\right)P\left(x_0,x_1\right)\ldots P\left(x_{n-1},x_n\right)\\&=P\left(x_1,x_0\right)\ldots P\left(x_n,x_{n-1}\right)\pi\left(x_n\right)\\&=\pi\left(x_n\right)P\left(x_n,x_{n-1}\right)\ldots P\left(x_1,x_0\right)\\&=\mathbb{P}\left(X_0=x_n,X_1=x_{n-1},\ldots,X_n=x_0\right)\\&=\mathbb{P}\left(X_n=x_0,X_{n-1}=x_1,\ldots,X_0=x_n\right).\end{aligned}$$

The time-reversed chain therefore has the same distribution:

$$\mathcal{L}\left(X_0,\ldots,X_n\right)=\mathcal{L}\left(X_n,\ldots,X_0\right)$$

In other words, when presented with a sequence of states for a reversible chain at equilibrium, one cannot tell in which direction time is flowing. In contrast, an example of an irreducible Markov chain that does not have a reversible distribution is as follows. Consider the transition matrix

$$P=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}. \tag{3.6}$$

Its unique stationary distribution π is the uniform distribution $(\frac{1}{3},\frac{1}{3},\frac{1}{3})$. It is readily seen that the chain is not reversible for π since one can clearly see in which direction time flows.

7.2 Reminders on the Bayesian Framework

Monte Carlo methods play an important role in Bayesian statistics, particularly because one often needs to compute integrals of the form $\mathbb{E}[\varphi(X)]=\int\varphi(x)f(x)dx$ for some density f and some function φ. In this section, we briefly recall why such integrals constantly arise in the Bayesian framework.

In Statistics, we observe a data set $\mathbf{X}=(X_1,\ldots,X_N)$ where the law of each X_i depends on a parameter θ. In the frequentist approach, θ is unknown but assumed to have a fixed value, and the goal is to estimate it from the observations $\mathbf{X}$ (e.g. by maximum likelihood). In the Bayesian framework, θ itself is considered as a random variable following a given law (called prior), with the observations $\mathbf{X}=(X_1,\ldots,X_N)$ refining this law.
More formally, let π be the density of the prior law of $\boldsymbol{\theta}$ (or prior) and $f(\mathbf{x}|\theta)$ the conditional density of $\mathbf{X}$ given $\boldsymbol{\theta}=\theta$ (or likelihood). For simplicity, we assume that all these densities are defined with respect to the Lebesgue measure. By Bayes' rule, the posterior density of $\boldsymbol{\theta}$ given the observation $\mathbf{X}$ (or posterior) is then

$$\pi(\theta|\mathbf{X})=\frac{f(\mathbf{X}|\theta)\pi(\theta)}{f(\mathbf{X})}=\frac{f(\mathbf{X}|\theta)\pi(\theta)}{\int f(\mathbf{X}|t)\pi(t)dt} \tag{7.3}$$

Since we are looking for a density with respect to the variable θ, everything that does not depend on θ acts as a normalization constant, hence the use of the symbol $\propto$ ("proportional to"). Thus, the notation

$$\pi(\theta|\mathbf{X}) \propto f(\mathbf{X}|\theta)\pi(\theta)$$

means that there exists a normalization constant $c(\mathbf{X})$, not involving θ, such that

$$\pi(\theta|\mathbf{X}) = c(\mathbf{X})f(\mathbf{X}|\theta)\pi(\theta).$$

Example: Gaussian model. Assume that $\boldsymbol{\theta} \sim \mathcal{N}(0,1)$ and that conditional on $\boldsymbol{\theta} = \theta$, the observations $\mathbf{X} = (X_1, \ldots, X_N)$ are i.i.d. with distribution $\mathcal{N}(\theta, 1)$. Then for any $\mathbf{X} = (X_1, \ldots, X_N) \in \mathbb{R}^N$, the likelihood is written as

$$f(\mathbf{X}|\theta) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i-\theta)^2}{2}} = \frac{1}{(2\pi)^{N/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{N}(X_i-\theta)^2\right\}.$$

The expression of the posterior density follows

$$\pi(\theta|\mathbf{X}) = \frac{\frac{1}{(2\pi)^{N/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{N}(X_i-\theta)^2\right\} \times \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\theta^2}{2}\right\}}{f(\mathbf{X})} \propto \exp\left\{-\frac{1}{2}\left((N+1)\theta^2 - 2N\bar{X}_N\theta\right)\right\}.$$

The term in the exponential is quadratic in θ. This suggests rewriting this expression as a Gaussian density to deduce the posterior law:

$$\pi(\theta \mid \mathbf{X}) \propto \exp\left\{-\frac{N+1}{2}\left(\theta - \frac{N}{N+1}\bar{X}_N\right)^2\right\}$$

hence

$$\mathcal{L}(\boldsymbol{\theta} \mid \mathbf{X}) = \mathcal{N}\left(\frac{N}{N+1}\bar{X}_N, \frac{1}{N+1}\right)$$

The posterior distribution is a normal distribution with mean $\frac{N\bar{X}_N}{N+1}$ (which depends on the observations) and variance $\frac{1}{N+1}$ (which does not depend on the observations). This distribution is random, as it depends on $\mathbf{X}$. Its density is given by

$$\pi(\theta \mid \mathbf{X}) = \sqrt{\frac{N+1}{2\pi}} \exp\left\{-\frac{N+1}{2}\left(\theta - \frac{N}{N+1}\bar{X}_N\right)^2\right\}$$

Compared to the prior distribution (i.e., the standard Gaussian), the posterior distribution is therefore roughly centered around the empirical mean $\bar{X}_N$ of the observed data and is much more concentrated around this mean than the prior distribution was around 0. In other words, the observations $X_1, \ldots, X_N$ have provided information about the unknown and random parameter $\boldsymbol{\theta}$. Now, we could be interested in computing an expectation with respect to this posterior distribution

$$\mathbb{E}[\varphi(\boldsymbol{\theta}) \mid \mathbf{X}] = \int \varphi(\theta)\pi(\theta \mid \mathbf{X})d\theta = \frac{\int \varphi(\theta)f(\mathbf{X} \mid \theta)\pi(\theta)d\theta}{\int f(\mathbf{X} \mid \theta)\pi(\theta)d\theta}. \tag{7.4}$$

For instance, the Bayes estimator for the L_2 loss or quadratic loss is defined as the posterior mean, namely

$$\hat{\theta}_N(\mathbf{X}) = \mathbb{E}[\boldsymbol{\theta} \mid \mathbf{X}] = \int \theta \pi(\theta \mid \mathbf{X}) d\theta = \frac{\int \theta f(\mathbf{X} \mid \theta)\pi(\theta) d\theta}{\int f(\mathbf{X} \mid \theta)\pi(\theta) d\theta}$$

Example. In the previous example, since knowing $\mathbf{X}$, the variable $\boldsymbol{\theta}$ follows a Gaussian distribution with mean $\frac{N\bar{X}_N}{N+1}$ and variance $\frac{1}{N+1}$, the Bayes estimator for the quadratic loss is simply

$$\hat{\theta}_N(\mathbf{X}) = \frac{N}{N+1}\bar{X}_N$$

Contrary to what this toy example might suggest, calculating the integral is generally challenging. Thus, we naturally resort to Monte Carlo methods. It suffices to know how to simulate $\theta_1, \ldots, \theta_n \sim \pi(\theta)$ i.i.d. and evaluate the quantity $\varphi(\theta) f(\mathbf{x} \,|\, \theta)$ for any θ. In this case,

$$\frac{1}{n}\sum_{i=1}^{n} \varphi\left(\boldsymbol{\theta}_i\right) f\left(\mathbf{X} \mid \boldsymbol{\theta}_i\right) \xrightarrow[n\to\infty]{a.s.} \int \varphi(\theta) f(\mathbf{X} \mid \theta)\pi(\theta) d\theta.$$

Estimating the denominator of (7.4) corresponds to the special case where $\varphi = 1$ in the equation above and is therefore treated in the same way. Overall, the Monte-Carlo estimator of $I = \mathbb{E}[\varphi(\boldsymbol{\theta}) \mid \mathbf{X}]$ is written as

$$\hat{I}_n = \frac{\sum_{i=1}^{n} \varphi\left(\boldsymbol{\theta}_i\right) f\left(\mathbf{X} \mid \boldsymbol{\theta}_i\right)}{\sum_{i=1}^{n} f\left(\mathbf{X} \mid \boldsymbol{\theta}_i\right)}$$

In particular, the Bayes estimator $\hat{\theta}_N(\mathbf{X}) = \mathbb{E}[\boldsymbol{\theta} \mid \mathbf{X}]$ for N observations $\mathbf{X} = (X_1, \ldots, X_N)$ itself has a Monte-Carlo estimator based on n simulations $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$:

$$\hat{\theta}_N^n(\mathbf{X}) = \frac{\sum_{i=1}^{n} \boldsymbol{\theta}_i f\left(\mathbf{X} \mid \boldsymbol{\theta}_i\right)}{\sum_{i=1}^{n} f\left(\mathbf{X} \mid \boldsymbol{\theta}_i\right)}.$$

7.3 Metropolis-Hastings algorithm

The purpose of the Metropolis-Hastings algorithm is to simulate a probability measure π on a state space E by simulating a Markov chain whose invariant measure is π. Under favorable assumptions, this Markov chain will thus converge in distribution to π (ergodicity). This algorithm is particularly used in cases where the measure π is not exactly known, but **known up to a constant**.
The general framework is as follows: the probability measure π is written

$$\pi(dx) = \frac{f(x)\lambda(dx)}{\int_E f(x)\lambda(dx)},$$

where

- f is a measurable function from $E \mapsto \mathbb{R}_+$
- λ is a positive reference measure (like the Lebesgue measure)

The normalization constant $\int_E f(x)\lambda(dx)$ is not known. We generally write that the measure π is proportional to $f(x)\lambda(dx)$:

$$\pi \propto f(x)\lambda(dx).$$

This case naturally arises in the Bayesian framework, where the posterior π has the form (7.3)

$$\pi(\theta|\mathbf{X}) = \frac{f(\mathbf{X}|\theta)\pi(\theta)}{f(\mathbf{X})} = \frac{f(\mathbf{X}|\theta)\pi(\theta)}{\int f(\mathbf{X}|t)\pi(t)dt}$$

where the numerator can be evaluated but the denominator involves computing an integral, which is generally intractable.

7.3.1 Finite State Space

Let π be a probability measure on a finite state space E, known only up to an unknown multiplicative constant. Our goal is to define a Markov chain $(X_n)_{n\geq 0}$ for which π is an invariant measure. Under the assumption of irreducibility and aperiodicity of the chain, the measure μ_n of the Markov chain will converge toward π regardless of the initialization μ_0 (see Theorem 9).

The Metropolis-Hastings algorithm requires an auxiliary transition matrix $Q : E \times E \to [0, 1]$ from which one can easily generate a Markov chain (whose invariant measure is not necessarily π). The algorithm then modifies Q to obtain a Markov chain with invariant measure π. The pseudo code of the Metropolis-Hastings algorithm is given below.

Algorithm 12: Metropolis-Hastings algorithm

```
1. Initialize $X_0$ according to any initial law;
2. For $i$ from 1 to $n$,
       Draw $y \sim Q(X_{i-1}, \cdot)$
       If $uniform() < \frac{\pi(y)Q(y,X_{i-1})}{\pi(X_{i-1})Q(X_{i-1},y)}$ then $X_i = y$
       Otherwise, $X_i = X_{i-1}$
3. Output $(X_0, \ldots, X_n)$.
```

Definition 10. *The Markov chain $(X_n)_{n\geq 0}$ defined by Metropolis-Hastings from a measure π on E and a transition matrix Q is the homogeneous Markov chain with transition matrix*

$$P(x,y) = \begin{cases} Q(x,y)\min\left(1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right) & \text{if } x \neq y \\ 1 - \sum_{z\neq x} P(x,z) & \text{if } x = y \end{cases} .$$

The probability $P(x,y)$ of moving from x to y (for $y \neq x$) is therefore the probability of moving from x to y according to the transition matrix Q multiplied by a factor $\min\left(1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right)$ called the acceptance-rejection probability.

A jump from x can be simulated as follows: Choose a candidate y to jump to according to the law given by $Q(x,\cdot)$, and jump to y with probability $\min\left(1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right)$, otherwise stay at x.

Important remark: The probability of keeping or rejecting the jump involves the *ratio* $\pi(x)/\pi(y)$. In particular, it suffices to know π on E up to a constant.

Theorem 10. *Assume that $\forall x \in E, \pi(x) > 0,$, and $\forall x, y \in E, Q(x,y) > 0 \iff Q(y,x) > 0$. Let $(X_n)_n$ be the output of M.H. with inputs Q and $C\pi$.*

1. *The sequence of random variables $(X_n)_n$ forms a Markov chain. Its transition matrix $P = \big(P(x,y)\big)_{x,y\in E}$ is given by*

$$P(x,y) = \begin{cases} Q(x,y)\min\left(\dfrac{\pi(y)\,Q(y,x)}{\pi(x)\,Q(x,y)},1\right) & \textit{if } x \neq y \\ 1-\sum_{z\neq x} P(x,z) & \textit{if } x = y. \end{cases}$$

2. *The measure π is an invariant probability measure for $(X_n)_n$.*
3. *If Q is irreducible, then so is P.*

Item 2 and 3 above ensure the convergence of the empirical measure μ_n of $(X_n)_n$ toward π by the ergodic theorem (see Theorem 9). Therefore, one can approximate π by simulating a trajectory $(X_n)_n$ using the Metropolis-Hastings algorithm and computing the empirical histogram of the trajectory (see equation (7.2)).

Proof of Theorem 10.

1. We check that P is the transition matrix of the chain (X_n) constructed by the Metropolis-Hastings algorithm. At each step i, let Y_i be the random variable drawn according to the law $Q(X_{i-1},\cdot)$ and $U_i \sim \mathrm{Unif}([0,1])$ a uniform random variable independent of Y_i and thus of X_{i-1}. For any pair $(x,y) \in E$ with $x \neq y$, we have

$$\begin{aligned} \mathbb{P}(X_{n+1}=y|X_n=x) &= \mathbb{P}\Bigg(Y_{n+1}=y \text{ and } U_{n+1} \leq \left(\frac{\pi(Y_{n+1})Q(Y_{n+1},X_n)}{\pi(X_n)Q(X_n,Y_{n+1})}\right)\Bigg| X_n = x\Bigg) \\ &= \mathbb{P}\Bigg(Y_{n+1}=y \text{ and } U_{n+1} \leq \left(\frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right)\Bigg| X_n = x\Bigg) \\ &= \min\left(1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right) Q(x,y), \end{aligned}$$

using the fact that U_{n+1} is independent of X_n and Y_{n+1}.

2. It suffices to show that π is reversible for P, i.e. that $\pi(x)P(x,y) = \pi(y)P(y,x)$ for any $x,y \in E$. The property holds if $x = y$.
For $x \neq y$, we have $\pi(x)P(x,y) = 0 = \pi(y)P(y,x)$ if $Q(x,y) = Q(y,x) = 0$, and

$$\begin{aligned} \pi(x)P(x,y) &= \pi(x)Q(x,y)\min\left(1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right) \\ &= \min\big(\pi(x)Q(x,y), \pi(y)Q(y,x)\big) \\ &= \pi(y)Q(y,x)\min\left(\frac{\pi(x)Q(x,y)}{\pi(y)Q(y,x)},1\right) \\ &= \pi(y)P(y,x) \end{aligned}$$

if $Q(x,y)$ and $Q(y,x)$ are non-zero.

3. We have $P(x,y) > 0$ as soon as $Q(x,y) > 0$ since we assumed that $\pi(x) \neq 0$. Therefore, all trajectories achievable under Q are achievable under P, and the irreducibility of Q implies the irreducibility of P.

□

Example: We want to simulate the measure π such that $\pi(k) \propto \frac{1}{(k+1)^2}$ on the space $\{1, \cdots, n\}$. We use the auxiliary Markov chain defined as follows: At each step, if the Markov chain is at state k, it jumps to one of the neighboring integers $k-1$ and $k+1$ with equal probability. When the chain is at a boundary (that is, at 1 or n), it remains at the same location with probability $1/2$ or moves to the neighboring integer with probability $1/2$. The matrix Q associated with this Markov chain is then written as

$$Q = \begin{pmatrix} 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ & 1/2 & \ddots & \ddots & \\ & & \ddots & 0 & 1/2 \\ & & & 1/2 & 1/2 \end{pmatrix}$$

Note that the Markov chain with kernel Q is irreducible. For example, let's compute

$$P(1,2) = Q(1,2)\min(1, \frac{\pi(2)Q(2,1)}{\pi(1)Q(1,2)}) = \frac{1}{2}\min(1, \frac{2^2}{3^2}) = \frac{4}{18}$$

It can be verified that for all $1 \geq k \leq n-1$:

$$P(k,k+1) = \frac{1}{2}\frac{(k+1)^2}{(k+2)^2}$$

and for all $2 \leq k \leq n$,

$$P(k,k-1) = \frac{1}{2}$$

In this example, we note that the calculations are greatly simplified by the fact that the matrix Q is symmetric.

7.3.2 Continuous State Space

Consider now a target measure π over $E = \mathbb{R}$ with density f. Even though E is now infinite, we can still define the Metropolis-Hastings algorithm. Proving its convergence is more involved than in the finite case and goes beyond the scope of the course. The interested reader is encouraged to consult [RS94] Theorem 3 for proof of convergence of the Metropolis-Hastings algorithm in general state spaces.

In the case $E = \mathbb{R}^d$, one must use a Markov chain on an uncountable space. A family of density functions $(q(x,y))_{x,y\in\mathbb{R}}$ indexed by $x \in \mathbb{R}$ is given, meaning that for every $x \in \mathbb{R}$, there is a probability measure $q(x,y)dy$, called the *proposal kernel.* This family of densities will allow us to choose a candidate for the algorithm's evolution.

Algorithm 13: Metropolis-Hastings Method (continuous case)

```
1. Initialize $X_0$ according to any initial law;
2. For $i$ from 1 to $n$,
        Draw $Y$ according to the law $q(X_{i-1}, y)dy$
        If $uniform() < \frac{f(y)q(Y,X_{i-1})}{f(X_{i-1})q(X_{i-1},Y)}$ then $X_i = Y$
        Otherwise, $X_i = X_{i-1}$
3. Output $(X_0, \ldots, X_n)$.
```

Special Case: Metropolis-Hastings via Random Walk Consider the special case where the exploration kernel $q(x, y)$ corresponds to a single law with density $g : \mathbb{R} \to \mathbb{R}$, shifted by x

$$q(x, y) = g(y - x).$$

This is referred to as the *random walk* case: If $q(x, y)dy$ denotes the law of the location to which one jumps from x, then g represents the law of the jump size $y - x$.

More specifically, it can be noted that for a fixed x, the law of a variable Y distributed according to the measure $q(x, y)dy = g(y - x)dy$ is that of the random variable $x + Z$ where Z is distributed according to the density g. Instead of changing the exploration kernel as the state x evolves, one can thus always use the same density g, which indicates the size of the next jump. Moreover, we also have the following simplification

$$\frac{q(y, x)}{q(x, y)} = \frac{g(x - y)}{g(y - x)} = \frac{g(-Z)}{g(Z)}$$

Algorithm 14: Metropolis-Hastings (Random Walk Case)

```
1. Initialize $X_0$ by simulating according to any initial distribution.
2. For $i$ from 1 to $n$,
       Simulate $Z$ according to the density $g$.
       Set $Y = X_{i-1} + Z$.
       If uniform() $< \frac{f(Y)}{f(X_{i-1})} \frac{g(-Z)}{g(Z)}$ then set $X_i = Y$.
       Otherwise, set $X_i = X_{i-1}$.
3. Output $(X_0, \ldots, X_n)$.
```

Symmetric Case The symmetric case $Q(x, y) = Q(y, x), \forall x, y \in E$ is interesting as it yields a simplified acceptance/rejection probability $\pi(y)/\pi(x)$. In the continuous case, a symmetric kernel means that $q(x, y) = q(y, x)$, which simplifies the acceptance/rejection probability to $f(y)/f(x)$.

Example: To simulate a law with density $f(x)$ proportional to $\exp(-|x|^4)$ using a Gaussian exploration kernel (jumping from x following a standard Gaussian centered at x), the exploration kernel is

$$q(x, y) = g(y - x)$$

where g is the Gaussian density, which is symmetric. Thus, we are in the case of a symmetric random walk, and we can write (noting that $f(y)/f(x) = \exp(|x|^4 - |y|^4)$)

Algorithm 15: Metropolis-Hastings (Symmetric Random Walk Case)

```
1. Initialize $X_0$ by simulating according to any initial distribution.
2. For $i$ from 1 to $n$,
       Simulate $Z$ according to a standard normal distribution.
       Set $Y = X_{i-1} + Z$.
       If uniform() $< \exp(|X_{i-1}|^4 - |Y|^4)$ then set $X_i = Y$.
       Otherwise, set $X_i = X_{i-1}$.
3. Output $(X_0, \ldots, X_n)$.
```

7.3.3 Conclusions and Questions Not Addressed Here

The Metropolis-Hastings algorithm is used to approximate a target law π which is known up to a constant. We have justified its convergence in the case of a finite state space.
However, we have not quantified the algorithm's convergence rate, which would allow us to know after how many iterations of the Markov chain the simulated variable is well distributed according to the invariant law. In the case of Metropolis-Hastings algorithms, this convergence rate will depend on the auxiliary Markov chain Q chosen to define the algorithm.

7.4 Gibbs Sampler

We will consider two different versions of the Gibbs sampler, referred to as *random-scan* and *deterministic-scan* Gibbs sampler. The random-scan version is a special case of Metropolis-Hastings where all transitions are accepted, but it requires much more knowledge about the target distribution π.

Consider, for example, the case of a continuous state space $\mathbb{R}^d$ or a subset of $\mathbb{R}^d$. The target density is written $\pi(x) = \pi(x_1, \ldots, x_d)$. For any index ℓ between 1 and d, we denote by

$$x_{-\ell} = (x_1, \ldots, x_{\ell-1}, x_{\ell+1}, \ldots, x_d)$$

the $d-1$-dimensional vector obtained by removing the ℓ-th coordinate of x. We slightly abuse notation and write the joint density as $\pi(x) = \pi(x_\ell, x_{-\ell})$. The notation $\pi\left(\cdot \mid x_{-\ell}\right)$ denotes the density conditional on $x_{-\ell}$, namely the density of x_ℓ when the remaining $(d-1)$ ones are frozen. We also denote by $\pi(x_{-\ell})$ the joint density of these $(d-1)$ coordinates.

The crucial assumption is as follows: For any ℓ and any $(d-1)$-tuple $x_{-\ell}$, we know how to simulate according to the conditional density $\pi\left(\cdot \mid x_{-\ell}\right)$. The Gibbs sampler simply updates one coordinate at a time, by drawing it from the probability density conditional on all the remaining coordinates. Its pseudo-code is given below.

Algorithm 16: random-scan Gibbs sampler

```
1. Draw $X_0 \in \mathbb{R}^d$ according to some initial measure $\mu_0$
2. Until termination condition, iterate:
       Set $x = X_k$
       Draw $\ell \sim \mathrm{Unif}(\{1, \ldots, d\})$ independent of the past
       Draw $x'_\ell \sim \pi(\,\cdot\,|x_{-\ell})$
       Define $X_{k+1} = \big(x_1, \ldots, x_{\ell-1}, x'_\ell, x_{\ell+1}, \ldots, x_d\big)$
3. Output $(X_0, \ldots, X_n)$.
```

Lemma 5 (random-scan Gibbs Sampler)**.** *The random-scan Gibbs sampler corresponds to a Metropolis-Hastings algorithm where all transitions are accepted, that is $r(x, y) = 1$ at each step. Consequently, the chain (X_n) is π-reversible and π is invariant.*

Proof. We use the notation introduced in the Metropolis-Hastings framework. The previous algorithm comes down to proposing $Y = y = \left(x'_\ell, x_{-\ell}\right)$ starting from $X_n = x = (x_\ell, x_{-\ell})$ with the transition density

$$q(x, y) = \frac{1}{d}\pi\left(x'_\ell \mid x_{-\ell}\right) = \frac{1}{d} \times \frac{\pi\left(x'_\ell, x_{-\ell}\right)}{\pi\left(x_{-\ell}\right)} = \frac{1}{d} \times \frac{\pi(y)}{\pi\left(x_{-\ell}\right)}.$$

Similarly, the transition from y to x has a density given by

$$q(y,x) = \frac{1}{d} \times \frac{\pi(x)}{\pi\left(x_{-\ell}\right)}.$$

With this established, the Metropolis-Hastings ratio therefore simplifies as

$$r(x,y) = \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)} = 1$$

meaning that all proposed transitions are accepted. □

This method is called random-scan Gibbs sampling since the index ℓ is drawn uniformly at random at each step. This is not completely satisfactory: If some coordinate ℓ is drawn twice in a row, the first draw does not contribute to the evolution of the chain (why?). A more commonly used Gibbs sampling algorithm is the *systematic or deterministic scan* version. This method consists in drawing the d coordinates successively.

Algorithm 17: Deterministic/Systematic scan Gibbs sampler

```
1. Draw $X_0 \in \mathbb{R}^d$ according to some initial measure $\mu_0$
2. Until termination condition, iterate:
       Set $x = X_k$
       For $\ell = 1, \dots, d$:
           Draw $x'_\ell \sim \pi\big(\cdot \mid (x'_1, \dots, x'_{\ell-1}, x_{\ell+1}, \dots x_d)\big)$
       Define $X_{k+1} = x' = (x'_1, \dots, x'_d)$
3. Output $(X_0, \dots, X_n)$.
```

Lemma 6 (Gibbs Sampler)**.** *The systematic scan Gibbs sampler admits π as its invariant distribution.*

Proof. Denote by Q_ℓ the transition kernel corresponding to the update of the ℓ-th coordinate. This kernel does not have a density with respect to the Lebesgue measure over $\mathbb{R}^d$ since $(d-1)$ coordinates remain unchanged. It is written

$$Q_\ell(x, dy) = \mathbf{1}_{y_{-\ell}=x_{-\ell}} \delta_{x_{-\ell}}\left(dy_{-\ell}\right) \pi\left(y_\ell \mid x_{-\ell}\right) dy_\ell.$$

In the proof of Lemma 5, we have seen that for any pair (x,y),

$$\pi(dx)Q_\ell(x,dy) = \pi(dy)Q_\ell(y,dx)$$

both terms being 0 if $y_{-\ell} \neq x_{-\ell}$. Let $R(x,dy)$ be the transition kernel of the systematic scan Gibbs sampler. It amounts to composing the transition kernels Q_ℓ introduced above, i.e., $R = Q_1 \dots Q_d$. We want to show that $\pi R = \pi$, but we have seen that $\pi Q_\ell = \pi$ for every ℓ, hence the result is clear. □

Remarks:

1. The deterministic scan Gibbs sampler is the version used in practice.
2. If π has a density, then the kernel R of the Gibbs sampler (implicitly: systematic scan) has a density: for any pair $(x,y) \in \mathbb{R}^d \times \mathbb{R}^d$, it is

$$R(x,y) = \pi\left(y_1 \mid x_2, \dots, x_d\right) \pi\left(y_2 \mid y_1, x_3, \dots, x_d\right) \cdots \pi\left(y_d \mid y_1, \dots, y_{d-1}\right)$$

3. Unlike the case of random-scanning, the chain (X_n) is generally not π-reversible. In dimension 2, it would indeed be necessary that for all pairs (x_1, x_2) and (y_1, y_2)

$$\pi(x_1, x_2)R\left((x_1, x_2), (y_1, y_2)\right) = \pi(y_1, y_2)R\left((y_1, y_2), (x_1, x_2)\right)$$

   that is, according to the previous expression of the kernel R,

$$\pi(x_1, x_2)\pi(y_1|x_2)\pi(y_2|y_1) = \pi(y_1, y_2)\pi(x_1|y_2)\pi(x_2|x_1)$$

   or equivalently,

$$\pi(y_1|x_2)\pi(y_2) = \pi(y_1)\pi(y_2|x_1)$$

   This relation is verified if the components of π are independent, but there is no general reason for it to be the case. It is sufficient to see that the left-hand side depends on x_2, unlike the right-hand side.

4. It is crucial not to update all coordinates simultaneously. Otherwise, the Markov chain might fail to converge to the desired law.

5. The invariance of π for the Gibbs sampler is not a sufficient condition to ensure its convergence. The proof of convergence of the Gibbs sampler can be found in [RS94], Theorem 2.

Bibliography

[Bac21] Francis Bach. Learning theory from first principles. *Draft of a book, version of Sept*, 6:2021, 2021.

[Bot10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010.

[Che06] Steve Cheng. Differentiation under the integral sign with weak derivatives. *tech. report*, 2006.

[Dan] Raphaël Danchin. Cours d'intégration l3 de mathématiques.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[Fer21] Olivier Fercoq. Stochastic optimization, 2021.

[Fre17] Ari Freedman. Convergence theorem for finite markov chains. *Proc. REU*, 2017.

[GLP14] François Golse, Yves Laszlo, and Frank Pacard. Analyse réelle. *Polycopié de l'École Polytechnique*, 2014.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KHR20] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In *Algorithmic Learning Theory*, pages 451–467. PMLR, 2020.

[LP17] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[LRP16] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[Nes83] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl akad nauk Sssr*, volume 269, page 543, 1983.

[Nes13] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[Nor98] James R Norris. *Markov chains.* Number 2. Cambridge university press, 1998.

[NTS15] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pages 1376–1401. PMLR, 2015.

[Pol64] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[PZ24] Philipp Petersen and Jakob Zech. Mathematical theory of deep learning. *arXiv preprint arXiv:2407.18384*, 2024.

[Qia99] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[Roc15] Ralph Tyrell Rockafellar. Convex analysis:(pms-28). 2015.

[RS94] Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic processes and their applications*, 49(2):207–216, 1994.

[SMDH13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[SZL13] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International conference on machine learning*, pages 343–351. PMLR, 2013.

[WRJ21] Ashia C Wilson, Ben Recht, and Michael I Jordan. A lyapunov analysis of accelerated methods in optimization. *Journal of Machine Learning Research*, 22(113):1–34, 2021.

Appendix

Proposition 20 below provides conditions allowing one to swap expectation and gradient.

Proposition 20. *Let $f : \Theta \times \Xi \to \mathbb{R}$ be a measurable function and let ξ be a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in Ξ. We suppose that $\mathbb{E}(|f(x,\xi)|) < +\infty$. Let $F(x) = \mathbb{E}f(x,\xi)$. Assume that*

1. *$(x \mapsto f(x,\xi))$ is convex and differentiable for all $\xi \in \Xi$,*
2. *F is C^1 i.e. F is differentiable with a continuous gradient,*
3. *There exists $C > 0$ such that $\mathbb{E}\left(\left\|\nabla f(x,\xi) - \nabla F(y)\right\|^2\right) \leq C$ for all $x \in \operatorname{dom} g$, and $y \in \Theta$.*

Then for any $x \in \Theta$, it holds that $\nabla F(x) = \mathbb{E}\nabla f(x,\xi)$.

Proof. For any $x \in \Theta$, we show that $\frac{\partial F}{\partial x_j}(x) = \mathbb{E}\left[\frac{\partial f}{\partial x_j}(x,\xi)\right]$. To do so, we apply the theorem of differentiation under the integral sign stated in Proposition 21. We first check the assumptions of this theorem.

1. For each $x \in \Theta$, $\xi \mapsto f(x,\xi)$ is an integrable function of ξ, since we have by the triangle inequality and Jensen's inequality

$$\begin{aligned}\mathbb{E}\|\nabla f(x,\xi)\| &\leq \mathbb{E}\|\nabla f(x,\xi) - \nabla F(y)\| + \|\nabla F(y)\| \\ &\leq \mathbb{E}^{1/2}\Big(\big\|\nabla f(x,\xi) - \nabla F\left(y\right)\big\|^2\Big) + \|\nabla F(y)\| \\ &\leq \sqrt{C} + \|\nabla F(y)\| < +\infty.\end{aligned}$$

2. For almost all $\xi \in \Xi$, the derivative $\frac{\partial f(x,\xi)}{\partial x_i}$ exists for all $x \in \Theta$, by assumption.

3. To check the last assumption, fix $i \in \{1,\dots,d\}$ and $x = (x_1,\dots,x_d) \in \Theta$ and $\xi \in \Xi$, and consider the function f_i obtained by letting only coordinate x_i vary, while every other parameter is fixed:

$$f_i(t,\xi) \mapsto f\big((x_1,\dots,x_{i-1},\, x_i + t,\, x_{i+1},\dots,x_d),\xi\big).$$

Since Θ is open, this function is defined on a neighborhood $(-\epsilon,\epsilon)$ of 0, and it is also convex by assumption (why?). Therefore, $f_i'(\cdot,\xi)$ is non-decreasing, so that, over $[-\epsilon/2,\epsilon/2]$, we can upper bound $f_i'(t,\xi)$ by a quantity that does not depend on x. Indeed, let $t \in [-\epsilon/2,\epsilon/2]$, then we have by convexity

$$f_i'\Big(-\frac{\epsilon}{2},\xi\Big) \leq f_i'(t,\xi) \leq f_i'\Big(\frac{\epsilon}{2},\xi\Big) \quad\implies\quad \big|f_i'(t,\xi)\big| \leq \big|f_i'\Big(\frac{\epsilon}{2},\xi\Big)\big| + \big|f_i'\Big(-\frac{\epsilon}{2},\xi\Big)\big| =: \varphi(\xi).$$

This ensures integrability. To see this, consider that for any $y \in \Theta$, we have

$$\begin{aligned}\mathbb{E}\Big|f_i'\Big(\frac{\epsilon}{2},\xi\Big)\Big| &\leq \mathbb{E}\Big|f_i'\Big(\frac{\epsilon}{2},\xi\Big) - \frac{\partial F}{\partial x_i}(y)\Big| + \Big|\frac{\partial F}{\partial x_i}(y)\Big| \\ &\leq \mathbb{E}\Big\|\nabla f_i\Big(\frac{\epsilon}{2},\xi\Big) - \nabla F(y)\Big\| + \Big\|\nabla F(y)\Big\| \\ &\leq \mathbb{E}^{1/2}\left[\Big\|\nabla f_i\Big(\frac{\epsilon}{2},\xi\Big) - \nabla F(y)\Big\|^2\right] + \Big\|\nabla F(y)\Big\| \quad \text{by Jensen's inequality} \\ &\leq \sqrt{C} + \Big\|\nabla F(y)\Big\| < \infty.\end{aligned}$$

Similarly, we can show that $\mathbb{E}\big|f_i'\big(-\frac{\epsilon}{2},\xi\big)\big| < \infty$, which yields

$$\mathbb{E}\varphi(\xi) \leq \mathbb{E}\big|f_i'\big(\frac{\epsilon}{2},\xi\big)\big| + \mathbb{E}\big|f_i'\big(-\frac{\epsilon}{2},\xi\big)\big| < \infty.$$

Therefore, we have found a function φ independent of x such that φ is integrable and $\left|\frac{\partial f(x,\xi)}{\partial x_i}\right| \leq \varphi(\xi)$ for all $x \in (-\epsilon/2,\epsilon/2)$.

We can now apply Proposition 21 below with $\Theta' = (-\epsilon/2,\epsilon/2)$ to obtain

$$\frac{\partial}{\partial x_i}\mathbb{E}f(x,\xi) = \mathbb{E}\left[\frac{\partial f(x,\xi)}{\partial x_i}\right].$$

Since this holds for any $i \in \{1,\dots,d\}$, we can conclude that

$$\nabla F(x) = \mathbb{E}\left[\nabla f(x,\xi)\right].$$

□

Note that the condition that there exists $C > 0$ such that $\mathbb{E}\left(\left\|\nabla f(x,\xi) - \nabla F(y)\right\|^2\right) \le C$ for all $x \in \operatorname{dom} g$, and $y \in \Theta$ is implied by the stronger one $\mathbb{E}\left(\left\|\nabla f(x,\xi)\right\|^2\right) \le C$ (why?).

We now state the Differentiation under the Integral Sign Theorem, also called the Leibniz rule.

Proposition 21 ([Che06], Theorem A.1)**.** *Let Θ be an open subset of $\mathbb{R}^d$, and Ξ be a measure space. Suppose that the function $f : \Theta \times \Xi \to \mathbb{R}$ satisfies the following conditions:*

1. *$f(x,\xi)$ is an integrable function of ξ for each $x \in \Theta$.*
2. *For almost all $\xi \in \Xi$, the derivative $\frac{\partial f(x,\xi)}{\partial x_i}$ exists for all $x \in \Theta$.*
3. *There is an integrable function $\varphi : \Xi \to \mathbb{R}$ such that $\left|\frac{\partial f(x,\xi)}{\partial x_i}\right| \le \varphi(\xi)$ for all $x \in \Theta$.*

Then

$$\frac{\partial}{\partial x_i}\int_{\Xi} f(x,\xi)\,d\xi = \int_{\Xi}\frac{\partial f(x,\xi)}{\partial x_i}\,d\xi.$$

Proof. See [GLP14] Theorem 5.10, or [Dan] “Théorème de dérivation sous le signe intégral”. □

Theorem 1 (Fubini-Tonelli)**.** *Let $\Omega_1 \subseteq \mathbb{R}^{N_1}$ and $\Omega_2 \subseteq \mathbb{R}^{N_2}$ be two non-empty open sets and let $f : \Omega_1 \times \Omega_2 \to [0,+\infty)$ be a measurable function. Then:*

(a) For almost every $x_2 \in \Omega_2$, the function $x_1 \mapsto f(x_1,x_2)$ from $[0,+\infty[$ is measurable;

(b) The function $x_2 \mapsto \int_{\Omega_1} f(x_1,x_2)dx_1 \in [0,+\infty)$, defined almost everywhere on Ω_2, is measurable;

(c) We have the equality in $[0,+\infty)$:

$$\iint_{\Omega_1\times\Omega_2} f(x_1,x_2)dx_1dx_2 = \int_{\Omega_1}\left(\int_{\Omega_2} f(x_1,x_2)dx_2\right)dx_1 = \int_{\Omega_2}\left(\int_{\Omega_1} f(x_1,x_2)dx_1\right)dx_2.$$

Proof. See [GLP14], Theorem 6.6. □