Learning theory from first principles

Lecture 4: Optimization for machine learning

Francis Bach

October 16, 2020

Class summary

-Gradient descent
-Stochastic gradient descent
-Generalization bounds through stochastic gradient descent
-Variance reduction

In this lecture, we present optimization algorithms based on gradient descent and analyze their performance, mostly on convex functions. See [1, 2] for further details.

1 Optimization in machine learning

- In supervised machine learning, we are given n i.i.d. samples (x_i, y_i), $i = 1, \dot{;} n$ of a couple of random variables (x, y) on $\mathcal{X} \times \mathcal{Y}$ and the goal is to find a predictor $f : \mathcal{X} \to \mathbb{R}$ with a small risk
$$\mathcal{R}(f) := \mathbb{E}[\ell(y, f(x))]$$
where $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a loss function. This loss is typically convex in the second argument (see Lecture 3), which is thus considered as a weak assumption.

- In the empirical risk minimization approach, we choose the predictor by minimizing the empirical risk over a parameterized set of predictors, potentially with regularization. For a parameterization $\{f_\theta\}_{\theta \in \mathbb{R}^d}$ and a regularizer $\Omega : \mathbb{R}^d \to \mathbb{R}$ (e.g., $\Omega(\theta) = \|\theta\|_2^2$ or $\Omega(\theta) = \|\theta\|_1$), this requires to minimize
$$F(\theta) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) + \Omega(\theta).$$
In optimization, the function $F : \mathbb{R}^d \to \mathbb{R}$ is called the *objective function.*

- In general, the minimizer has no closed form. Even when it has one (e.g., linear predictor and square loss), it could be expensive to compute for large problems. We thus resort to iterative algorithms.

- Solving optimization problems to high accuracy is computationally expensive, and the goal is not to minimize the training objective, but the error on unseen data.

  Then, which accuracy is satisfying in machine learning? If the algorithm returns $\hat{\theta}$ and $\theta_* \in \arg\min_\theta \mathcal{R}(f_\theta)$, we have the risk decomposition (where the approximation error due to the use of a specific set of models f_θ, $\theta \in \Theta$ is ignored):

  $$\mathcal{R}(f_{\hat{\theta}}) - \inf_{\theta \in \mathbb{R}^d} \mathcal{R}(f_\theta) = \underbrace{\Big\{\mathcal{R}(f_{\hat{\theta}}) - \hat{\mathcal{R}}(f_{\hat{\theta}})\Big\}}_{\leqslant \text{ estimation error}} + \underbrace{\Big\{\hat{\mathcal{R}}(f_{\hat{\theta}}) - \hat{\mathcal{R}}(f_{\theta_*})\Big\}}_{\leqslant \text{ optimization error}} + \underbrace{\Big\{\mathcal{R}(f_{\theta_*}) - \hat{\mathcal{R}}(f_{\theta_*})\Big\}}_{\leqslant \text{ estimation error}}.$$

  It is thus sufficient to reach an optimization accuracy of the order of the estimation error (usually of the order $O(1/\sqrt{n})$ or $O(1/n)$, see Lectures 2 and 3).

In this lecture, we will first look at the minimization without focusing on machine learning problems (Section 2), with both smooth and non-smooth optimization. We will then look at stochastic gradient descent in Section 4, which can be used to obtain bounds on both the training risk and the testing risk. We then briefly present variance reduction.

⚠ θ_* may mean different things in optimization and machine learning: minimizer of the regularized empirical risk, or minimizer of the expected risk. For the sake of clarity, we will use the notation η_* for the minimizer of empirical (potential regularized risk), that is, when we look at optimization problems, and θ_* for the minimizer of the expected risk, that is, when we look at statistical problems.

⚠ Sometimes, we mention solving a problem with *high* precision. This corresponds to a *low* optimization error.

2 Gradient descent

Suppose we want to solve, for a function $F : \mathbb{R}^d \to \mathbb{R}$, the optimization problem

$$\min_{\theta \in \mathbb{R}^d} F(\theta).$$

We assume that we are given access to certain "oracles": the *k-th-order oracle* corresponds to the access to: $\theta \mapsto (F(\theta), F'(\theta), \dots, F^{(k)}(\theta))$. All algorithms will call these oracles and thus their computational complexity will depend directly on the complexity of this oracle. For example, for least-squares with a design matrix in $\mathbb{R}^{n \times d}$, computing a single gradient of the empirical risk costs $O(nd)$. In this section, for the algorithms and proofs, we do not assume that the function F is the regularized empirical risk, but this situation will be our motivating example throughout.

In this section, we study the following first-order algorithm.

Algorithm 1 (Gradient descent (GD)) *Pick $\theta_0 \in \mathbb{R}^d$ and for $t \geqslant 1$, let*

$$\theta_t = \theta_{t-1} - \gamma_t F'(\theta_{t-1}),$$

for a well (potentially adaptively) chosen step-size sequence $(\gamma_t)_{t \geqslant 1}$.

There are many ways to choose the step-size γ_t, either constant, either decaying, either through a line search (see, e.g., `https://en.wikipedia.org/wiki/Line_search`). In practice, using some form of line search is strongly advantageous and is implemented in most applications. In this lecture, since we want to focus on the simplest algorithms and proofs, we will focus on step-sizes that depend explicitly on problem constants, and sometimes on the iteration number.

We first start with the simplest example, namely quadratic convex functions.

2.1 Simplest analysis: ordinary least-squares

We start with a case where the analysis is explicit: ordinary least squares (see Lecture 2 for the statistical analysis). Let $\Phi \in \mathbb{R}^{n \times d}$ be the design matrix and $y \in \mathbb{R}^n$ the vector of responses. Least-squares estimation amounts to finding a minimizer η_* of

$$F(\theta) = \frac{1}{2n} \|\Phi\theta - y\|_2^2.$$

⚠ A factor of $\frac{1}{2}$ has been added compared to Lecture 2 to get nicer looking gradients.

The gradient of F is $F'(\theta) = \frac{1}{n}\Phi^\top(\Phi\theta - y) = \frac{1}{n}\Phi^\top\Phi\theta - \frac{1}{n}\Phi^\top y$. Thus, denoting $H = \frac{1}{n}\Phi^\top\Phi \in \mathbb{R}^{d \times d}$, minimizers η_* are characterized by

$$H\eta_* = \frac{1}{n}\Phi^\top y.$$

Since $\frac{1}{n}\Phi^\top y \in \mathbb{R}^d$ is in the column space of H, there is always a minimizer, but unless H is invertible, the minimizer if not unique. But all minimizers η_* have the same function value $F(\eta_*)$, and we have, from a simple exact Taylor expansion (and using $F'(\eta_*) = 0$:

$$F(\theta) - F(\eta_*) = F'(\eta_*)^\top(\theta - \eta_*) + \frac{1}{2}(\theta - \eta_*)^\top H(\theta - \eta_*) = \frac{1}{2}(\theta - \eta_*)^\top H(\theta - \eta_*).$$

Two quantities will be important in the following developments, the largest eigenvalue L and the smallest eigenvalue μ of the Hessian matrix H. As a consequence of convexity of the objective, we have $0 \leqslant \mu \leqslant L$. We denote by $\kappa = \frac{L}{\mu} \geqslant 1$ the *condition number*.

Note that for least-squares, μ is the lowest eigenvalue of the non-centered empirical covariance matrix and that it is zero as soon as $d > n$, and, in most cases, very small. When adding a regularizer $\frac{\lambda}{2}\|\theta\|_2^2$ (like in ridge regression), then $\mu \geqslant \lambda$ (but then λ typically decreases with n, often between $\frac{1}{\sqrt{n}}$ and $\frac{1}{n}$, see Lecture 6 for more details).

Closed-form expression. Gradient descent iterates with fixed step-size $\gamma_t = \gamma$ can be computed in closed form:

$$\theta_t = \theta_{t-1} - \gamma F'(\theta_{t-1}) = \theta_{t-1} - \gamma\big[\frac{1}{n}\Phi^\top(\Phi\theta_{t-1} - y)\big] = \theta_{t-1} - \gamma H(\theta_{t-1} - \eta_*),$$

leading to

$$\theta_t - \eta_* = \theta_{t-1} - \eta_* - \gamma H(\theta_{t-1} - \eta_*) = (I - \gamma H)(\theta_{t-1} - \eta_*),$$

that is, we have a linear recursion, and we can unroll the recursion, and now write

$$\theta_t - \eta_* = (I - \gamma H)^t(\theta_0 - \eta_*).$$

We can now look at various measures of performance:

$$\begin{aligned} \|\theta_t - \eta_*\|_2^2 &= (\theta_0 - \eta_*)^\top (I - \gamma H)^{2t}(\theta_0 - \eta_*) \\ F(\theta_t) - F(\eta_*) &= (\theta_0 - \eta_*)^\top (I - \gamma H)^{2t} H(\theta_0 - \eta_*). \end{aligned}$$

The two optimization performance measures differ by the presence of the Hessian matrix H in the measure based on function values.

Convergence in distance to minimizer. If we hope to have $\|\theta_t - \eta_*\|_2^2$ going to zero, we need to have a single minimizer η_*, and thus H has to be invertible, that is $\mu > 0$. Given the form of $\|\theta_t - \eta_*\|_2^2$, we simply need to bound the eigenvalues of $(I - \gamma H)^{2t}$.

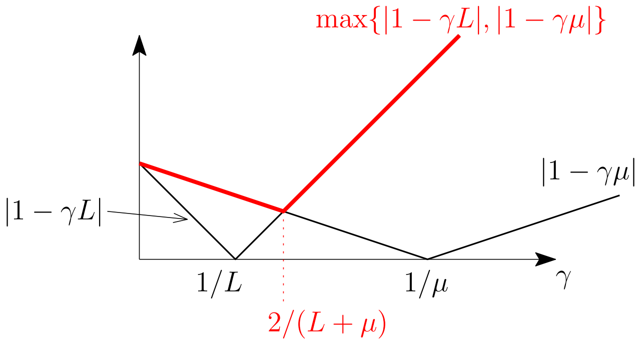
The eigenvalues of $(I - \gamma H)^{2t}$ are exactly $(1 - \gamma\lambda)^{2t}$ for λ an eigenvalue of H (which are all in the interval $[\mu, L]$).

Thus all eigenvalues of $(I - \gamma H)^{2t}$ have magnitude less than

$$\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda|.$$

We can then have several strategies for choosing the step-size γ:

- Optimal choice: one can check that minimizing $\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda|$ is done by setting $\gamma = 2/(\mu + L)$, with an optimal value of $\frac{\kappa-1}{\kappa+1} = 1 - \frac{2}{\kappa+1} \in (0, 1)$. See "proof" below.



- Choice independent of μ: with the simpler (slightly smaller) choice $\gamma = 1/L$, we get $\max_{\lambda\in[\mu,L]} |1-\gamma\lambda| = (1-\frac{\mu}{L}) = (1-\frac{1}{\kappa})$, which is only sligthly larger than the value for the optimal choice.

With the weaker choice $\gamma = 1/L$, we get:

$$\|\theta_t - \eta_*\|_2^2 \leqslant \Big(1-\frac{1}{\kappa}\Big)^{2t}\|\theta_0-\eta_*\|_2^2,$$

which is often referred to as exponential, geometric, or also linear convergence.

⚠ The denomination "linear" is sometimes confusing and corresponds to a number of significant digits that grows linearly with the number of iterations.

We can further bound $\Big(1-\frac{1}{\kappa}\Big)^{2t} \leqslant \exp(-1/\kappa)^{2t} = \exp(-2t/\kappa)$, and thus the characteristic time of convergence is of order κ. We will often make the calculation $\varepsilon = \exp(-2t/\kappa) \Leftrightarrow t = \frac{\kappa}{2}\log\frac{1}{\varepsilon}$. Thus, for a relative reduction of squared distance to optimum of ε, we need at most $t = \frac{\kappa}{2}\log\frac{1}{\varepsilon}$ iterations.

For $\kappa = +\infty$, then the result remains true, but simply say that for all minimizers $\|\theta_t-\eta_*\|_2^2 \leqslant \|\theta_0-\eta_*\|_2^2$, which is a good sign (the algorithm does not move away from minimizers) but not indicative of any form of convergence. We will need to use a different criterion.

Convergence in function values. Using the same step-size as above, and using the upper-bound on eigenvalues of $(I-\gamma H)^{2t}$, we get

$$F(\theta_t) - F(\eta_*) \leqslant \Big(1-\frac{1}{\kappa}\Big)^{2t}[F(\theta_0)-F(\eta_*)].$$

When $\kappa < \infty$ (that is, $\mu > 0$), then we also obtain linear convergence for this criterion, but when $\kappa = \infty$, this is non-informative.

In order to obtain a convergence rate, we will need to bound the eigenvalues of $(I-\gamma H)^{2t}H$ instead of $(I-\gamma H)^{2t}$. The key difference is that for eigenvalues λ of H which are close to zero $(1-\gamma\lambda)^{2t}$ does not have a strong contracting effect, but they count less as they are multiplied by λ in the bound.

We can now make this trade-off precise, for $\gamma \leqslant 1/L$, as

$$\begin{aligned}\big|\lambda(1-\gamma\lambda)^{2t}\big| &\leqslant \lambda\exp(-\gamma\lambda)^{2t} = \lambda\exp(-2t\gamma\lambda)\\ &= \frac{1}{2t\gamma}2t\gamma\lambda\exp(-2t\gamma\lambda) \leqslant \frac{1}{2t\gamma}\sup_{\alpha\geqslant 0}\alpha\exp(-\alpha) = \frac{1}{2et\gamma} \leqslant \frac{1}{4t\gamma},\end{aligned}$$

where we used that $\alpha e^{-\alpha}$ is maximized over $\mathbb{R}_+$ at $\alpha = 1$ (as the derivative $e^{-\alpha}(1-\alpha)$).

This leads to

$$F(\theta_t) - F(\eta_*) \leqslant \frac{1}{4t\gamma} \|\theta_0 - \eta_*\|_2^2.$$

We can make the following observations:

- ⚠ The convergence results in $\exp(-t/\kappa)$ for invertible Hessians or $1/t$ in general are only upper-bounds! It is good to understand the gap between the bounds and the actual performance, as this is possible for quadratic objective functions.

  For the exponentially convergent case, the lowest eigenvalue μ dictates the rate for all eigenvalues. So if the eigenvalues are well-spread (or if only one eigenvalue is very small), there can be quite a strong discrepancy between the bound and the actual behavior.

  For the rate in $1/t$, the bound in eigenvalues is tight when $t\gamma\lambda$ is of order 1, namely when λ is of order $1/(t\gamma)$. Thus, in order to see an $O(1/t)$ convergence rate in practice, we need to have sufficiently many small eigenvalues, and as t grows, we often go to a local linear convergence phase where the smallest non zero eigenvalue of H kicks in. See simulations and exercice below.

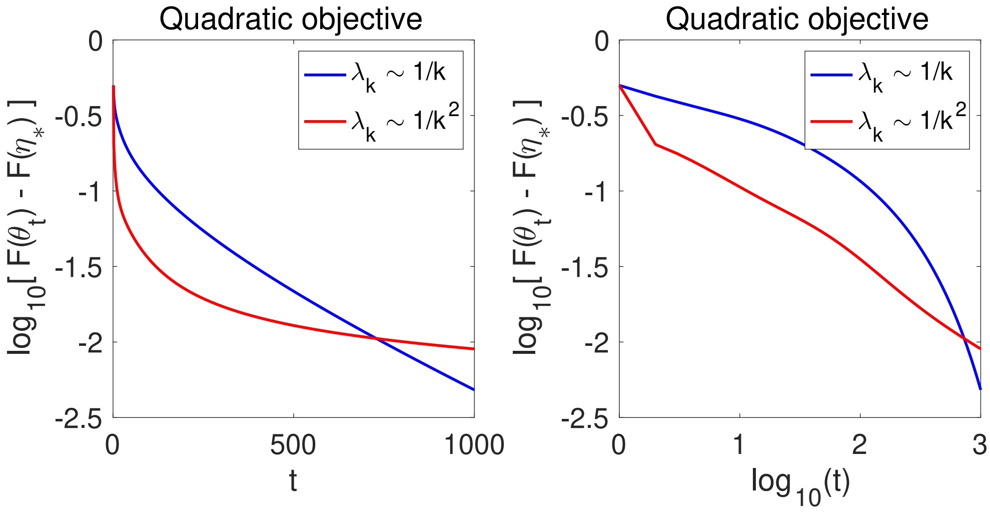
  – **Exercice**: Let μ_+ be the smallest non-zero eigenvalue of H. Show that gradient descent is linearly convergence with the contracting rate $(1 - \mu_+/L)$.

- Can an algorithm having the same access to oracles of F do better?

  If we have access to matrix-vector products with the matrix Φ, then conjugate gradient can be used with convergence rates in $\exp(-t/\sqrt{\kappa})$ and $1/t^2$ (see [3]). With only access to gradients of F (which is a bit weaker) Nesterov acceleration (see below) will also lead to the same convergence rates, which are then optimal (for a sense to be defined later).

- Can we extend beyond least-squares? The convergence results above will generalize to convex functions (see Section 2.2), but with less direct proofs. Non-convex objectives are discussed in Section 2.6

Experiments. We consider two quadratic optimization problems in dimension $d = 1000$, with two different decays of eigenvalues (λ_k) for the Hessian matrix H, one as $1/k$ (in blue below) and one in $1/k^2$ (in red below), and for which we plot the performance for function values, both in semi-logarithm plots (left) and full-logarithm plots (right). For slow decays (blue), we see the linear convergence kicking in, while for fast decays (red), the rates in $1/t$ dominate.



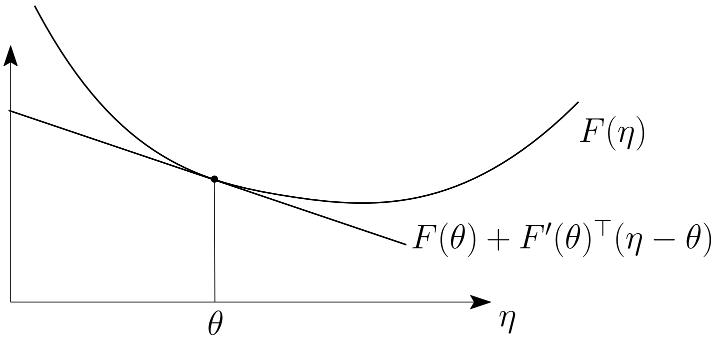
2.2 Convex functions

We now wish to analyze GD (and later its stochastic version SGD) in a broader setting. We will always assume convexity, although these algorithms are also used (and can sometimes also be analyzed) when this assumption does not hold (see Section 2.6). In other words, convexity is most often used for the analysis, not to define the algorithm.

Definition 1 (Convex function) *A differentiable function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is said* convex *if and only if*

$$F(\eta) \geq F(\theta) + F'(\theta)^\top(\eta - \theta), \qquad \forall \eta, \theta \in \mathbb{R}^d. \tag{1}$$

This corresponds to the function F being above its tangent at θ, as illustrated below.



If f is twice-differentiable, this is equivalent to requiring $F''(x) \succcurlyeq 0$, $\forall x \in \mathbb{R}^d$ (here $\succcurlyeq$ denotes the semidefinite partial ordering—also called Loewner order—characterized by $A \succcurlyeq B \Leftrightarrow A - B$ is positive semidefinite, see [4, 5]).

An important consequence that we will use a lot in this lecture is, for all $\theta \in \mathbb{R}^d$ (and usind $\eta = \eta_*$)

$$F(\eta_*) \geqslant F(\theta) + F'(\theta)^\top(\eta_* - \theta) \Leftrightarrow F(\theta) - F(\eta_*) \leqslant F'(\theta)^\top(\theta - \eta_*), \tag{2}$$

that is the distance to optimum in function values is upperbounded by a function of the gradient.

A more general definition of convexity is that $\forall x, y \in \mathbb{R}^d$ and $\alpha \in [0,1]$,

$$F(\alpha\eta + (1-\alpha)\theta) \leqslant \alpha F(\eta) + (1-\alpha)F(\theta).$$

- **Exercise**: show that if F is differentiable, this is equivalent to our definition. The following inequality appears frequently in the proofs involving convexity.

Proposition 1 (Jensen's inequality) *If $F : \mathbb{R}^d \to \mathbb{R}$ is convex and μ is a probability measure on $\mathbb{R}^d$, then*

$$F\Big(\int \theta d\mu(\theta)\Big) \leqslant \int F(\theta) d\mu(\theta).$$

In words: "the image of the average is smaller than the average of the images".

The class of convex functions satisfies the following stability properties (proofs left as an exercise):

- If $(F_j)_{j\in[m]}$ are convex and $(\alpha_j)_{j\in[m]}$ are nonnegative, then $\sum_{j=1}^m \alpha_j F_j$ is convex.
- If $F : \mathbb{R}^d \to \mathbb{R}$ is convex and $A : \mathbb{R}^{d'} \to \mathbb{R}^d$ is linear then $F \circ A : \mathbb{R}^{d'} \to \mathbb{R}$ is convex.

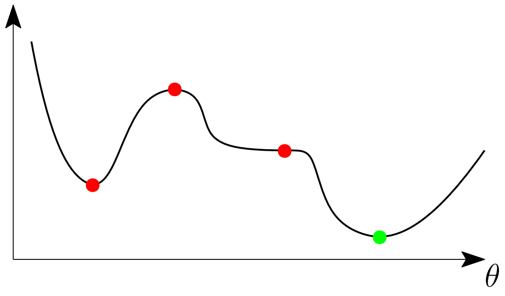
Example. Problems of the form in Eq. (1) are convex if the loss ℓ is convex in the second variable, $f_\theta(x)$ is linear in θ, and Ω is convex.

It is also worth emphasizing on the following property (immediate from the definition).

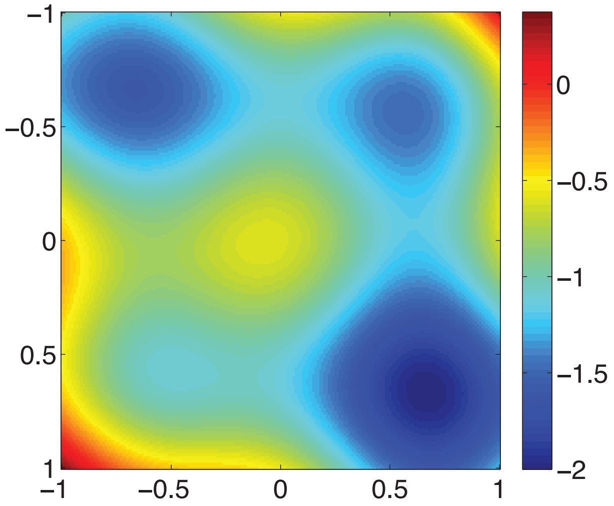
Proposition 2 *Assume that $F : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable. Then $\eta_* \in \mathbb{R}^d$ is a global minimizer of F if and only if*

$$F'(\eta_*) = 0.$$

This implies that for convex function, we only need to look for stationary points. This is *not* the case for potentially non-convex functions. For example, in one dimension below, all red points are stationary points which are not the global minimum (which is in green).



The situation is even more complex in higher dimensions. **Exercise**: identify all stationary points in the function in $\mathbb{R}^2$ depicted below.



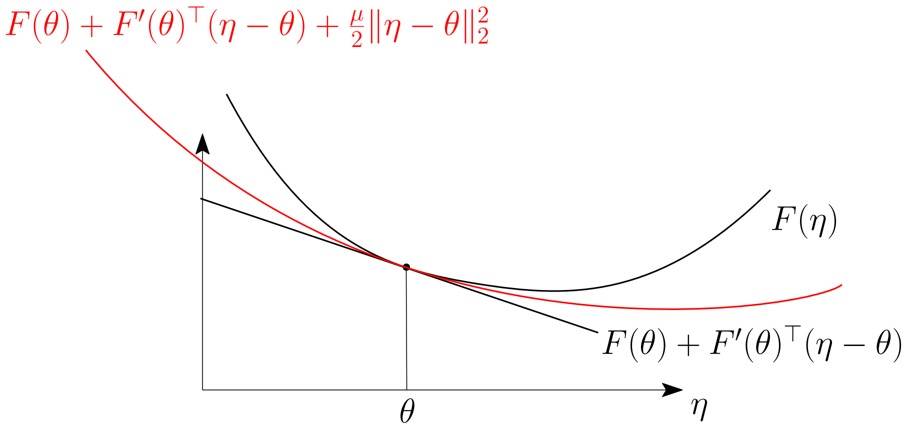
2.3 Analysis of GD for strongly convex and smooth functions

The analysis of optimization algorithms requires assumptions on the objective functions, like the ones introduced in this section. From these assumptions, additional properties are derived (typically inequalities), and then most convergence proofs look for a "Lyapunov function" (sometimes called a potential function) that goes down along the iterations. More precisely, if $V : \mathbb{R}^d \mapsto \mathbb{R}_+$ is such that $V(\theta_t) \leqslant (1-\alpha)V(\theta_{t-1})$, then $V(\theta_t) \leqslant (1-\alpha)^t V(\theta_0$ and we obtain linear convergence.

Definition 2 (Strong convexity) *A differentiable function F is said μ-strongly convex, with $\mu > 0$, if and only if*

$$F(\eta) \geq F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|_2^2, \quad \forall \eta, \theta \in \mathbb{R}^d$$

The function F is strongly-convex if and only if the function F is strictly above its tangent and the difference is at least quadratic in the distance to the point where the two coincide. This notably allows to defined quadratic lower bounds on F. See below.


$F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{\mu}{2}\|\eta - \theta\|_2^2$
$F(\eta)$
$F(\theta) + F'(\theta)^\top(\eta - \theta)$
θ
η

For twice differentiable functions, this is equivalent to $F'' \succcurlyeq \mu I$ (see [1]).

- **Exercise**: show that if F is convex, then $F + \frac{\mu}{2}\| \cdot \|_2^2$ is μ-strongly convex.
- In machine learning problems, with linear models, so that the empirical risk is convex, strong convexity most often comes from the regularizer (and thus μ decays with n), leading to condition numbers that grow with n.

This property implies that F admits a unique minimizer η_*, which is characterized by $F'(\eta_*) = 0$. Moreover, this guarantees that the gradient is large when a point is far from optimality:

Lemma 1 (Lojasiewicz inequality) *If F is differentiable and μ-strongly convex with minimizer η_*, then it holds*

$$\|F'(\theta)\|_2^2 \geq 2\mu(F(\theta) - F(\eta_*)), \quad \forall \theta \in \mathbb{R}^d.$$

Proof The right-hand side in Definition 2 is strongly convex in η and minimized with $\tilde{\eta} = \theta - \frac{1}{\mu}F'(\theta)$. Plugging this value into the bound and taking $\eta = \eta_*$ in the left-hand side we get

$$F(\eta_*) \geq F(\theta) - \frac{1}{\mu}\|F'(\theta)\|_2^2 + \frac{1}{2\mu}\|F'(\theta)\|_2^2 = F(\theta) - \frac{1}{2\mu}\|F'(\theta)\|_2^2.$$

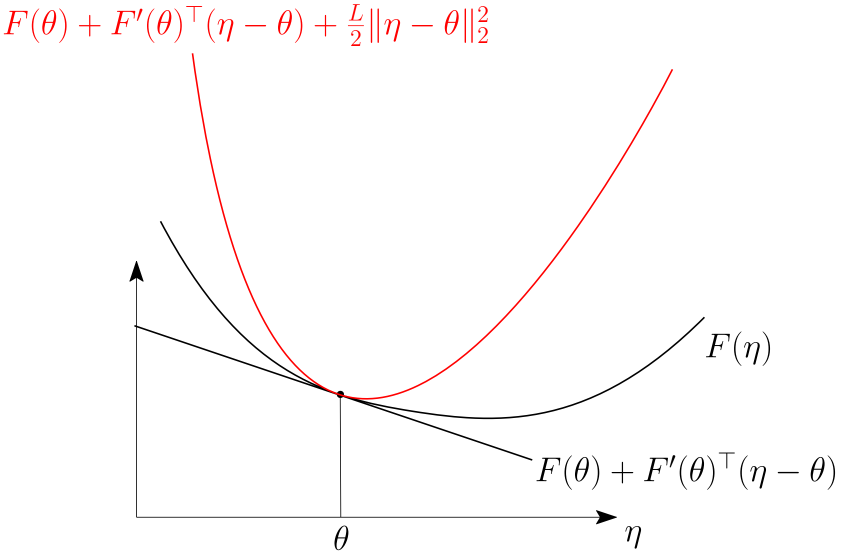
The conclusion follows by rearranging. ■

Definition 3 (Smoothness) *A differentiable function F is said L-smooth if and only if*

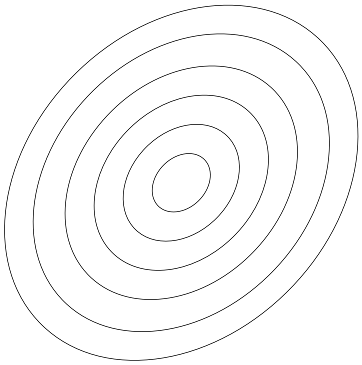
$$|F(\eta) - F(\theta) - F'(\theta)^\top(\eta - \theta)| \leqslant \frac{L}{2}\|\theta - \eta\|^2, \quad \forall \theta, \eta \in \mathbb{R}^d$$

This is equivalent to F having a L-Lipschitz gradient, i.e., $\|F'(\theta) - F'(\eta)\|_2^2 \leqslant \|\theta - \eta\|_2^2$, $\forall \theta, \eta \in \mathbb{R}^d$. For twice differentiable functions, this is equivalent to $-LI \preccurlyeq F''(\theta) \preccurlyeq LI$ (see [1]).

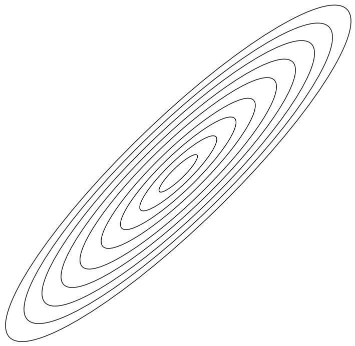
Note that when F is convex and L-smooth, we have a quadratic upper-bound which is tight at any given point (strong convexity implies the corresponding lower bound). See below.


$F(\theta) + F'(\theta)^\top(\eta - \theta) + \frac{L}{2}\|\eta - \theta\|_2^2$
$F(\eta)$
$F(\theta) + F'(\theta)^\top(\eta - \theta)$
θ
η

When a function is both smooth and strongly convex, we denote by $\kappa = L/\mu \geqslant 1$ its condition number. See examples below: the condition number impacts the shapes of the level sets).

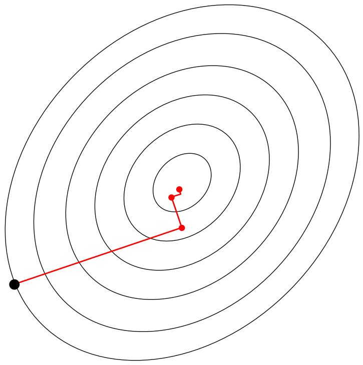


(small $\kappa = L/\mu$)

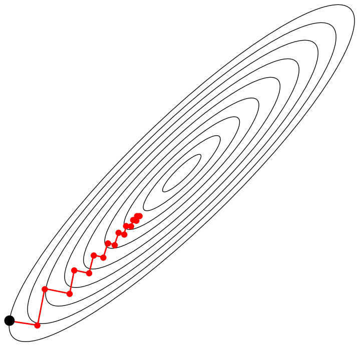


(large $\kappa = L/\mu$)

The performance of gradient descent will depend on this condition number (see steepest descent below, that is, gradient descent with exact line search): with small condition number (left), we get fast convergence, while for a large condition number (right), we get oscillations.



(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

- For machine learning problems, for linear predictions and smooth losses (square or logistic), then we have smooth problems. If we use a squared ℓ_2-regularizer $\frac{\mu}{2}$ regularizer, we get at μ-strongly convex problem. Note that when using regularization, as explained in Lectures 2 and 3, the value of μ decays with n, typically between $1/n$ and $1/\sqrt{n}$, leading to condition numbers between $\sqrt{n}$ and n.

  In this context, gradient descent on the empirical risk, is often called a "batch" technique.

In the next theorem, we show that gradient descent converges exponentially for such problems.

Theorem 1 (Convergence of GD for strongly convex functions) *Assume that F is L-smooth and μ-strongly convex. Choosing $\gamma_t = 1/L$, the iterates $(\theta_t)_{t \geq 0}$ of GD on F satisfy*

$$F(\theta_t) - F(\eta_*) \leq \exp(-t\mu/L)(F(\theta_0) - F(\eta_*)).$$

Proof By smoothness, we have the following descent property, with $\gamma_t = 1/L$,

$$\begin{aligned} F(\theta_t) &= F(\theta_{t-1} - F'(\theta_{t-1}/L) \leqslant F(\theta_{t-1}) + F'(\theta_{t-1})^\top(-F'(\theta_{t-1})/L) + \frac{L}{2}\|-F'(\theta_{t-1})/L\|_2^2 \\ &= F(\theta_{t-1}) - \|F'(\theta_{t-1})\|_2^2/L + \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2. \end{aligned}$$

Rearranging, we get

$$F(\theta_t) - F(\eta_*) \leqslant (F(\theta_{t-1}) - F(\eta_*)) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2.$$

Using Lemma 1, it follows

$$F(\theta_t) - F(\eta_*) \leqslant (1 - \mu/L)(F(\theta_{t-1}) - F(\eta_*)) \leqslant \exp(-\mu/L)(F(\theta_{t-1}) - F(\eta_*)).$$

We conclude by a recursion. ■

- We necessarily have $\mu \leqslant L$. The ratio $\kappa := L/\mu$ is called the *condition number*.
- If we only assume that the function is smooth and convex (not strongly convex), then GD with constant step-size $\gamma = 1/L$ also converges when a minimizer exists, but at a slower rate in $O(1/t)$. See proof below.
- Choosing the step-size only requires an upper bound L on the smoothness constant (in case it is over-estimated, the convergence rate only degrades slightly).
- Note that gradient descent is adaptive to strong convexity: the exact same algorithm applies to both strongly convex and convex cases, and the two bounds apply. This adaptivity is important in practice, as often, locally around the global optimum, the strong convexity constant converges to the minimal eigenvalue of the Hessian at η_*, which can very significantly larger than μ (the global constant).
- **Exercise**: compute all constants for ℓ_2-regularized logistic regression.
- **Fenchel conjugate**: given some convex function $F : \mathbb{R}^d \to \mathbb{R}$, an important tool is the Fenchel conjugate F^* defined as $F^*(\alpha) = \sup_{\theta\in\mathbb{R}^d} \alpha^\top\theta - F(\theta)$. This is crucial when dealing with convex duality (which we will not cover in this lecture); see [4] for details.

2.4 Analysis of GD for convex and smooth functions (♦)

In order to obtain the $1/t$ convergence rate without strong-convexity, we will need an extra property of convex smooth functions, sometimes called "co-coercivity". This is an instance of inequalities that we need to use to circumvent the lack of closed form for iterations.

Proposition 3 *If F is a convex L-smooth function on $\mathbb{R}^d$, then for all $\theta, \eta \in \mathbb{R}^d$, we have:*

$$\frac{1}{L}\|F'(\theta) - F'(\eta)\|_2^2 \leqslant \big[F'(\theta) - F'(\eta)\big]^\top(\theta - \eta).$$

Moreover, we have: $F(\theta) \geqslant F(\eta) + F'(\eta)^\top(\theta - \eta) + \frac{1}{2L}\|F'(\theta) - F'(\eta)\|^2$.

Proof We wil show the second inequality, which implies the first one by applying it twice with η and θ swapped, and summing them.

- Define $H(\theta) = F(\theta) - \theta^\top F'(\eta)$. The function $H : \mathbb{R}^d \to \mathbb{R}$ is convex with global minimum at η, since $H'(\theta) = F'(\theta) - F'(\eta)$, which is equal to zero for $\theta = \eta$. The function H is also L-smooth.
- We can apply the definition of smoothness: $H(\eta) \leqslant H(\theta - \frac{1}{L}H'(\theta)) \leqslant H(\theta) + H'(\theta)^\top(-\frac{1}{L}H'(\theta)) + \frac{L}{2}\|-\frac{1}{L}H'(\theta)\|_2^2$, which is thus less than $H(\theta) - \frac{1}{2L}\|H'(\theta)\|_2^2$
- This leads to $F(\eta) - \eta^\top F'(\eta) \leqslant F(\theta) - \theta^\top F'(\eta) - \frac{1}{2L}\|F'(\theta) - F'(\eta)\|_2^2$, which leads to the desired inequality by shuffling terms.

■

Following [6], the Lyapunov function that we will choose is

$$V_t(\theta_t) = t[F(\theta_t) - F(\eta_*)] + \frac{L}{2}\|\theta_t - \eta_*\|_2^2,$$

and our goal is to show that it decays along iterations. We get:

$$V_t(\theta_t) - V_{t-1}(\theta_{t-1}) \;=\; t[F(\theta_t) - F(\theta_{t-1})] + F(\theta_t) - F(\eta_*) + \frac{L}{2}\|\theta_t - \eta_*\|_2^2 - \frac{L}{2}\|\theta_{t-1} - \eta_*\|_2^2$$

In order to bound it, we use:

- We use $F(\theta_t) - F(\theta_{t-1}) \leqslant -\frac{1}{2L}\|F'(\theta_{t-1}\|_2^2$ like in the proof of Theorem 1
- We use $F(\theta_t) - F(\eta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*)$, as a consequence of convexity (function above the tangent at θ_{t-1}), as in Eq. (2).
- We get $\frac{L}{2}\|\theta_t - \eta_*\|_2^2 - \frac{L}{2}\|\theta_{t-1} - \eta_*\|_2^2 = -L\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + \frac{L\gamma^2}{2}\|F'(\theta_{t-1})\|_2^2$ by expanding the square.

This leads to, with $\gamma = 1/L$:

$$\begin{aligned} V_t(\theta_t) - V_{t-1}(\theta_{t-1}) &\leqslant t\Big[-\frac{1}{2L}\|F'(\theta_{t-1}\|_2^2\Big] + F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*) - L\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + \frac{L\gamma^2}{2}\|F'(\theta_{t-1})\|_2^2 \\ &= -\frac{t-1}{2L}\|F'(\theta_{t-1}\|_2^2 \leqslant 0, \end{aligned}$$

which leads to

$$t[F(\theta_t) - F(\eta_*)] \leqslant V_t(\theta_t) \leqslant V_0(\theta_0) = \frac{L}{2}\|\theta_0 - \eta_*\|_2^2,$$

and thus $F(\theta_t) - F(\eta_*) \leqslant \frac{L}{2t}\|\theta_0 - \eta_*\|_2^2$.

The proof above is on purpose mysterious: the choice of Lyapunov function seems arbitrary at first, but all inequalities lead to nice cancellations. These proofs are sometimes hard to design. For a very interesting line of work trying to automate these proofs, see `https://francisbach.com/computer-aided-analyses/`.

2.5 Beyond gradient descent (♦)

While gradient descent is the simplest algorithm with a simple analysis, there are multiple extensions that we will only briefly mention (see more details in [7, 8]):

- **Nesterov acceleration**: For convex functions, a simple modification of gradient descent allows to obtain better convergence rates. The algorithm is as follows, and is based on updating iterates:

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L}F'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{t-1}{t+2}(\theta_t - \theta_{t-1}).\end{aligned}$$

  This simple modification dates back to Nesterov in 1983, and leads to the following convergence rate $F(\theta_t) - F(\eta_*) \leqslant \frac{2L\|\theta_0 - \eta_*\|^2}{(t+1)^2}$.

  For strongly convex functions, the algorithm has a similar form as for convex functions, but with all coefficients which are independent from t:

$$\begin{aligned}\theta_t &= \eta_{t-1} - \frac{1}{L}g'(\eta_{t-1}) \\ \eta_t &= \theta_t + \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}(\theta_t - \theta_{t-1}),\end{aligned}$$

  and the convergence rate is $F(\theta_t) - F(\eta_*) \leqslant L\|\theta_0 - \eta_*\|^2(1-\sqrt{\mu/L})^t$, that is the characteristic time to convergence goes from κ to κ. If κ is large (typically of order $\sqrt{n}$ or n for machine learning), the gains are substantial. In practice, this leads to significant improvements.

  Moreover, the last two rates are known to be optimal for the considered problems: for algorithms that access gradient and combine them linearly to select a new query point, it is not possible to have better dimension-independent rates. See [8] for more details.

- **Newton methods**: Given θ_{t-1}, the Newton method minimizes the second-order Taylor expansion around θ_{t-1}

$$F(\theta_{t-1}) + F'(\theta_{t-1})^\top(\theta - \theta_{t-1}) + \frac{1}{2}(\theta - \theta_{t-1})^\top F''(\theta_{t-1})^\top(\theta - \theta_{t-1}),$$

  which leads to $\theta_t = \theta_{t-1} - F''(\theta_{t-1})^{-1}F'(\theta_{t-1})$, which is an expansive iteration, as the running-time complexity is $O(d^3)$ in general to solve the linear system. It leads to local quadratic convergence: If $\|\theta_{t-1} - \theta_*\|$ small enough, for some constant C, we have $(C\|\theta_t - \theta_*\|) = (C\|\theta_{t-1} - \theta_*\|)^2$. See [4] for more details, and for conditions for global convergence.

  Note that for machine learning problems, quadratic convergence may be an overkill compared to the computational complexity of each iteration, since cost functions are averages of n terms and naturally have some uncertainty of order $O(1/\sqrt{n})$.

- **Proximal gradient descent (♦)**: Many optimization problems are said "composite", that is, the objective function F is the sum of a smooth function G and a non-smooth function H (such as a norm). It turns out that a simple modification of gradient descent allows to benefit from the fast convergence rates of smooth optimization.

For this, we need to first see gradient descent as a *proximal method.* Indeed, one may see the iteration $\theta_t = \theta_{t-1} - \frac{1}{L}G'(\theta_{t-1})$, as

$$\theta_t = \arg\min_{\theta \in \mathbb{R}^d} G(\theta_{t-1}) + (\theta - \theta_{t-1})^\top G'(\theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2,$$

where, for a L-smooth function G, the objective function above is an upper-bound of $G(\theta)$ which is tight at θ_{t-1}.

While this reformulation does not bring much for gradient descent, we can extend this to the composite problem, and consider the iteration

$$\theta_t = \arg\min_{\theta \in \mathbb{R}^d} G(\theta_{t-1}) + (\theta - \theta_{t-1})^\top G'(\theta_{t-1}) + \frac{L}{2}\|\theta - \theta_{t-1}\|_2^2 + H(\theta),$$

where H is left as is. It turns out that the convergence rates for $G + H$ are the same as smooth optimization, with potential acceleration [8, 9].

The crux is to be able to compute the step above, that is minimize with respect to θ functions of the form $\frac{L}{2}\|\theta - \eta\|_2^2 + H(\theta)$. When H is the indicator function of a convex set (which is equal to 0 inside the set, and $+\infty$ otherwise), we get projected gradient descent. When H is the ℓ_1-norm, that is $H = \lambda\|\cdot\|_2$, this can be shown to be soft-thresholding step, as for each coordinate $\theta_i = (|\eta_i| - \lambda/L)_+ \frac{\eta_i}{|\eta_i|}$ (proof left as an exercise).

2.6 Non-convex objective functions (♦)

For smooth potentially non convex objective functions, the best one can hope for is to converge to a stationary point θ such that $F'(\theta) = 0$. The proof below provides the weaker result that at least one iterate has a small gradient. Indeed, using the same Taylor expansion as the convex case (which is still valid), we get

$$F(\theta_t) \quad \leqslant \quad F(\theta_{t-1}) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2,$$

leading to, summing the inequalities above for all iterations between 1 and t, we get:

$$\frac{1}{t}\sum_{s=1}^{t} \|F'(\theta_{s-1})\|_2^2 \leqslant \frac{F(\theta_0) - F(\eta_*)}{t}.$$

Thus there has to be one s in $\{0, \dots, t-1\}$ for which $\|F'(\theta_s)\|_2^2 \leqslant O(1/t)$.

3 Gradient methods on non-smooth problems

We now relax our assumptions and only require Lipschitz continuity, in addition to convexity. The rates will be slower, but the extension to stochastic gradients easier.

Definition 4 (Lipschitz function) *A function $F : \mathbb{R}^d \to \mathbb{R}$ is said B-Lipschitz-continuous if and only if*

$$|F(\eta) - F(\theta)| \leqslant B\|\eta - \theta\|_2, \qquad \forall \theta, \eta \in \mathbb{R}^d.$$

- **Exercise**: show that if F is differentiable, this is equivalent to the assumption $\|F'(\theta)\|_2 \leqslant B, \forall \theta \in \mathbb{R}^d$. Without additional assumptions, this setting is usually referred to as *non-smooth* optimization.
- We can apply non-smooth optimization to objective functions which are not differentiable. For convex Lipschitz-continuous objectives, the function is almost everywhere differentiable. In points where it is not, then one can define the set of slopes of lower-bounding tangents as the *subdifferential*, and any element of it as a *subgradient*. The gradient descent iteration is then meant as using any subgradient instead of $F'(\theta_{t-1})$. The method is then referred to as the subgradient method (it is not a descent method anymore, that is, the function values may go up once in a while).

  The method can be in particular applied to the hinge loss.

3.1 Convergence rate of the subgradient method

Theorem 2 (Convergence of the subgradient method) *Assume that F is convex, B-Lipschitz and admits a minimizer η_* that satisfies $\|\eta_* - \theta_0\|_2 \leqslant D$. By chosing $\gamma_t = \frac{D}{B\sqrt{t}}$ then the iterates $(\theta_t)_{t\geq 0}$ of GD on G satisfy*

$$\min_{0\leqslant s\leqslant t-1} F(\theta_s) - F(\eta_*) \leqslant DB\frac{2+\log(t)}{\sqrt{t}}.$$

Proof We look at how θ_t approaches η_*, that is, we try to use $\|\theta_t - \eta_*\|_2^2$ as a Lyapunov function. We have:

$$\|\theta_t - \eta_*\|_2^2 = \|\theta_{t-1} - \gamma_t F'(\theta_{t-1}) - \eta_*\|_2^2 = \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma_t F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*) + \gamma_t^2\|F'(\theta_{t-1})\|_2^2.$$

Combining this with the convexity inequality $F(\theta_{t-1}) - F(\eta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1} - \eta_*)$ from Eq. (2), it follows (also using the boundedness of gradients):

$$\|\theta_t - \eta_*\|_2^2 \leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma_t[F(\theta_{t-1}) - F(\eta_*)] + \gamma_t^2 B^2.$$

and thus, by isolating the distance to optimum in function values:

$$\gamma_t(F(\theta_{t-1}) - F(\eta_*)) \leqslant \frac{1}{2}\Big(\|\theta_{t-1} - \eta_*\|_2^2 - \|\theta_t - \eta_*\|_2^2\Big) + \frac{1}{2}\gamma_t^2 B^2. \tag{3}$$

It is sufficient to sum these inequalities to get, for any $\eta_* \in \mathbb{R}^d$,

$$\frac{1}{\sum_{s=1}^t \gamma_s}\sum_{s=1}^t \gamma_s\left(F(\theta_{s-1}) - F(\eta_*)\right) \leqslant \frac{\|\theta_0 - \eta_*\|_2^2}{2\sum_{s=1}^t \gamma_s} + B^2\frac{\sum_{s=1}^t \gamma_s^2}{2\sum_{s=1}^t \gamma_s}.$$

The left-hand side is larger than $\min_{0\leqslant s\leqslant t-1}(F(\theta_s) - F(\eta_*))$ (trivially) and than $F(\bar\theta_t) - F(\eta_*)$ where $\bar\theta_t = (\sum_{s=1}^t \gamma_s\theta_{s-1})/(\sum_{s=1}^t \gamma_s)$ by Jensen's inequality.

The upper bound goes to 0 if $\sum_{s=1}^t \gamma_s$ goes to ∞ (to forget the initial condition, sometimes called the "bias") and $\gamma_t \to 0$ (to decrease the "variance" term). Let us choose $\gamma_s = \tau/\sqrt{s}$ for some $\tau > 0$. By using the series-integral comparisons below, we get the bound

$$\min_{0\leqslant s\leqslant t-1}(F(\theta_s) - F(\eta_*)) \leqslant \frac{1}{\sqrt{t}}\Big(D^2/\tau + \tau B^2(1+\log(t))\Big).$$

We choose $\tau = D/B$ (which is suggested by optimizing the previous bound when $\log(t) = 0$) which leads to the result. ■

In the proof, we used the following series-integral comparisons for decreasing functions:

$$\sum_{s=1}^{t} \frac{1}{\sqrt{s}} \geq \int_0^t \frac{ds}{\sqrt{s+1}} = \Big[2\sqrt{s+1}\Big]_0^t = 2\sqrt{t+1} - 2 \geqslant \frac{1}{2}\sqrt{t}$$

and

$$\sum_{s=1}^{t} \frac{1}{s} \leqslant 1 + \sum_{s=2}^{t} \frac{1}{s} \leqslant 1 + \int_0^t \frac{ds}{s} = 1 + \log(t).$$

- The previous proof scheme is very flexible. It can be extended in the following directions
  - No need to know in advance an upper-bound D on the distance to optimum, we then get with the same step-size $\gamma_t = \frac{D}{B\sqrt{t}}$ a rate of the form $\frac{BD}{\sqrt{t}}\Big(\frac{\|\theta_0 - \eta_*\|_2^2}{D^2} + (1 + \log(t))\Big)$.
  - Constrained minimization over a convex set (we then insert a projection step at each iteration);
  - Non-differentiable convex and Lipschitz objective functions (using sub-gradients, i.e. any vector satisfying Eq. (1) in place of $F'(\theta_t)$);
  - Non-Euclidean geometry (for instance multiplicative instead of additive updates), using "mirror descent".
  - Often the uniformly averaged iterate is used, as $\frac{1}{t}\sum_{s=0}^{t-1}\theta_s$. Convergence rates (without the $\log t$ factor) can be obtained using Abel summation formula (see `https://francisbach.com/integration-by-parts-abel-transformation/`).
  - Stochastic gradients, as seen below.

- **Exercise**: compute all constants for ℓ_2-regularized logistic regression.

4 Convergence rate of SGD

For machine learning problems, at each iteration, the gradient descent algorithm requires to compute a "full" gradient $F'(\theta_{t-1})$ which could be costly. An alternative is to instead only compute *unbiased* stochastic estimations of the gradient $g_t(\theta_{t-1})$, i.e., such that $\mathbb{E}[g_t(\theta_{t-1})|\theta_{t-1}] = F'(\theta_{t-1})$, which could be much faster to compute.

Note that we need to condition over θ_{t-1} because θ_{t-1} encapsulates all the randomness due to past iterations, and we only require "fresh" randomness at time t.

Somewhat surprisingly, this unbiasedness does *not* need to be coupled with a vanishing variance: while there are always errors in the gradient, the use of a decreasing step-size will ensure convergence.

This leads to the following algorithm.

Algorithm 2 (Stochastic gradient descent (SDG)) *Choose step-size sequence* $(\gamma_t)_{t\geq 0}$*, pick* $\theta_0 \in \mathbb{R}^d$ *and for* t ≥ 0*, let*

$$\theta_{t+1} = \theta_t - \gamma_t g_t(\theta_t).$$

SGD in machine learning. There are two ways to use SGD for supervised machine learning:

- Empirical risk minimization: If $F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$ then at iteration t we can choose uniformly at random $i(t) \in \{1, \dots, n\}$ and define g_t as the gradient of $\theta \mapsto \ell(y_{i(t)}, f_\theta(x_{i(t)}))$. There exists "mini-batch" variants where at each iteration, the gradient is averaged over a random subset of the indices. We then converge to a minimizer η_* of the empirical risk.

  Note here that since we sample *with replacement*, a given function will be selected several times.

- Population risk minimization: If $F(\theta) = \mathbb{E}[\ell(y, f_\theta(x))]$ then at iteration t we can take a fresh sample (x_t, y_t) and define g_t as the gradient of $\theta \mapsto \ell(y_t, f_\theta(x_t))$, for which, if we swap the orders of expectation and differentiation, we get the unbiasedness. Note here that to preserve the unbiasedness, only a single pass is allowed (otherwise, this would create dependencies that would break it).

  Here, we *directly minimize the (generalization) risk.* The counterpart is that if we only have n samples, then we can only run n SGD iterations, and when n grows, the iterates will converge to a minimizer θ_* of the expected risk.

We can study the two situations above using the latter one, by considering the empirical risk as the expectation with respect to the empirical distribution of the data.

⚠ SGD is not a descent method

Under the same assumptions on the objective, we now study SGD . We assume the following:

- (H1) unbiased gradient: $\mathbb{E}[g_t(\theta_{t-1})|\theta_{t-1}] = F'(\theta_{t-1})$, $\forall t$,
- (H2) bounded gradient: $\|g_t(\theta_{t-1})\|_2^2 \leqslant B^2$, $\forall t$ almost surely

Theorem 3 (Convergence of SGD) *Assume that F is convex, B-Lipschitz and admits a minimizer θ_* that satisfies $\|\theta_* - \theta_0\|_2 \leqslant D$. Assume that the stochastic gradients satisfy (H1-2). Then, choosing $\gamma_t = (D/B)/\sqrt{t}$, the iterates $(\theta_t)_{t\geq 0}$ of SGD on F satisfy*

$$\mathbb{E}\Big[F(\bar{\theta}_t) - F(\theta_*)\Big] \leqslant DB\frac{2 + \log(t)}{\sqrt{t}}.$$

where $\bar{\theta}_t = (\sum_{s=1}^{t} \gamma_s \theta_{s-1})/(\sum_{s=1}^{t} \gamma_s)$.

Proof We follow essentially the same proof as in the deterministic case, adding some expectations at well chosen places.

$$\begin{aligned}
\mathbb{E}\Big[\|\theta_t - \theta_*\|_2^2\Big] &= \mathbb{E}\Big[\|\theta_{t-1} - \gamma_t g_t(\theta_{t-1}) - \theta_*\|_2^2\Big] \\
&= \mathbb{E}\Big[\|\theta_{t-1} - \theta_*\|_2^2\Big] - 2\gamma_t \mathbb{E}\Big[g_t(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\Big] + \gamma_t^2 \mathbb{E}\Big[\|g_t(\theta_{t-1})\|_2^2\Big] \\
&\leqslant \mathbb{E}\Big[\|\theta_{t-1} - \theta_*\|_2^2\Big] - 2\gamma_t \mathbb{E}\Big[F'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\Big] + \gamma_t^2 B^2.
\end{aligned}$$

Above, the subtle part is the expectation

$$\begin{aligned}\mathbb{E}\Big[g_t(\theta_{t-1})^\top(\theta_{t-1}-\theta_*)\Big] &= \mathbb{E}\Big[\mathbb{E}\Big[g_t(\theta_{t-1})^\top(\theta_{t-1}-\theta_*)\Big|\theta_{t-1}\Big]\Big] \\ &= \mathbb{E}\Big[\mathbb{E}\Big[g_t(\theta_{t-1})\Big|\theta_{t-1}\Big]^\top(\theta_{t-1}-\theta_*)\Big] = \mathbb{E}\Big[F'(\theta_{t-1})^\top(\theta_{t-1}-\theta_*)\Big].\end{aligned}$$

Thus, combining with the convexity inequality $F(\theta_{t-1}) - F(\theta_*) \leqslant F'(\theta_{t-1})^\top(\theta_{t-1}-\theta_*)$ from Eq. (2), it follows

$$\gamma_t\mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] \leqslant \frac{1}{2}\Big(\mathbb{E}\big[\|\theta_{t-1}-\theta_*\|_2^2\big] - \mathbb{E}\big[\|\theta_t-\theta_*\|_2^2\big]\Big) + \frac{1}{2}\gamma_t^2B^2. \tag{4}$$

Except for the expectations, this is the same bound that Eq. (3) so we can conclude as in the proof of Theorem 2, *mutatis mutandis.* We state our bound in terms of the average iterates because the cost of finding the best iterate could be high in comparison to that of evaluating a stochastic gradient. ■

- Many authors consider the projected version of the algorithm where after the gradient step, we orthogonally project onto the ball of radius D and center θ_0. The bound is then exactly the same.
- The result that we obtain, when applied to single pass SGD, is a generalization bound that is, after the n iterations, we have an excess risk proportion to $1/\sqrt{n}$, corresponding to the excess risk compared to the best predictor f_θ.

  This is to be compared to using results from Lecture 3 (uniform deviation bounds) and non-stochastic gradient descent. It turns out that the estimation error due to having n observation is exactly the same as the generalization bound obtained by SGD, but we need to add on top the optimization error proportional to $1/\sqrt{t}$ (with the same constants). The bounds match if $t = n$, that is, we run n iterations of gradient descent on the empirical risk. This leads to a running time complexity of $O(tnd) = O(n^2d)$ instead of $O(nd)$ using SGD, hence the strong gains in using SGD.
- The bound in $O(BD/\sqrt{t})$ is optimal for this class of problem. That is, as shown for example in [10], among all algorithms that can query stochastic gradients, having a better convergence rate (up to some constants) is impossible.
- As opposed to the deterministic case, the use of smoothness does not lead to significantly better results.

4.1 Strongly convex problems (♦)

We consider the regularized problem $G(\theta) = F(\theta) + \frac{\mu}{2}\|\theta\|_2^2$, with the same assumption as above, and started at $\theta_0 = 0$. The SGD iteration is then:

$$\theta_t = \theta_{t-1} - \gamma_t\big[g_t(\theta_{t-1}) + \mu\theta_{t-1}\big]. \tag{5}$$

We then have

Theorem 4 (Convergence of SGD for strongly-convex problems) *Assume that F is convex, B-Lipschitz and that $G + \frac{\mu}{2}\|\cdot\|_2^2$ admits a (necesary unique) minimizer θ_*. Assume that the stochastic gradient g satisfies (H1-2). Then, choosing $\gamma_t = 1/(\mu t)$, the iterates $(\theta_t)_{t\geq 0}$ of SGD from Eq. (5) satisfy*

$$\mathbb{E}\Big[F(\bar\theta_t) - F(\theta_*)\Big] \leqslant \frac{2B^2(1+\log t)}{\mu t}$$

where $\bar\theta_t = (\sum_{s=1}^t \gamma_s \theta_{s-1})/(\sum_{s=1}^t \gamma_s)$.

Proof The beginning of the proof is essentially the same as for convex problems, leading to (with the new terms in blue):

$$\begin{aligned}\mathbb{E}\Big[\|\theta_t - \theta_*\|_2^2\Big] &= \mathbb{E}\Big[\|\theta_{t-1} - \gamma_t(g_t(\theta_{t-1}) + \mu\theta_{t-1}) - \theta_*\|_2^2\Big] \\ &= \mathbb{E}\Big[\|\theta_{t-1} - \theta_*\|_2^2\Big] - 2\gamma_t \mathbb{E}\Big[(g_t(\theta_{t-1}) + \mu\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\Big] + \gamma_t^2 \mathbb{E}\Big[\|g_t(\theta_{t-1}) + \mu\theta_{t-1}\|_2^2\Big].\end{aligned}$$

From the iterations, we see that $\theta_t = (1-\gamma_t\mu)\theta_{t-1} + \gamma_t\mu\big[-\frac{1}{\mu}g_t(\theta_{t-1})\big]$ is a convex combination of gradients divided by $-\mu$, and thus $\|g_t(\theta_{t-1}) + \mu\theta_{t-1}\|_2^2$ is always less than $4B^2$. Thus

$$\mathbb{E}\Big[\|\theta_t - \theta_*\|_2^2\Big] \leqslant \mathbb{E}\Big[\|\theta_{t-1} - \theta_*\|_2^2\Big] - 2\gamma_t \mathbb{E}\Big[G'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)\Big] + 4\gamma_t^2 B^2.$$

Therefore, combining with the strong convexity inequality $G(\theta_{t-1}) - G(\eta_*) + \frac{\mu}{2}\|\theta_{t-1} - \theta_*\|_2^2 \leqslant G'(\theta_{t-1})^\top(\theta_{t-1} - \theta_*)$ it follows

$$\gamma_t \mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] \leqslant \frac{1}{2}\Big((1 - \gamma_t\mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \mathbb{E}\|\theta_t - \theta_*\|^2\Big) + 2\gamma_t^2 B^2,$$

and thus, now using the specific step-size choice:

$$\begin{aligned}\mathbb{E}[F(\theta_{t-1}) - F(\theta_*)] &\leqslant \frac{1}{2}\Big((\gamma_t^{-1} - \mu)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \gamma_t^{-1}\mathbb{E}\|\theta_t - \theta_*\|^2\Big) + 2\gamma_t B^2, \\ &= \frac{1}{2}\Big(\mu(t-1)\mathbb{E}\|\theta_{t-1} - \theta_*\|^2 - \mu t \mathbb{E}\|\theta_t - \theta_*\|^2\Big) + \frac{2B^2}{\mu t}.\end{aligned}$$

Thus, summing between all indices between 1 and t, and using the bound $\sum_{s=1}^t \frac{1}{s} \leqslant 1 + \log t$. ■

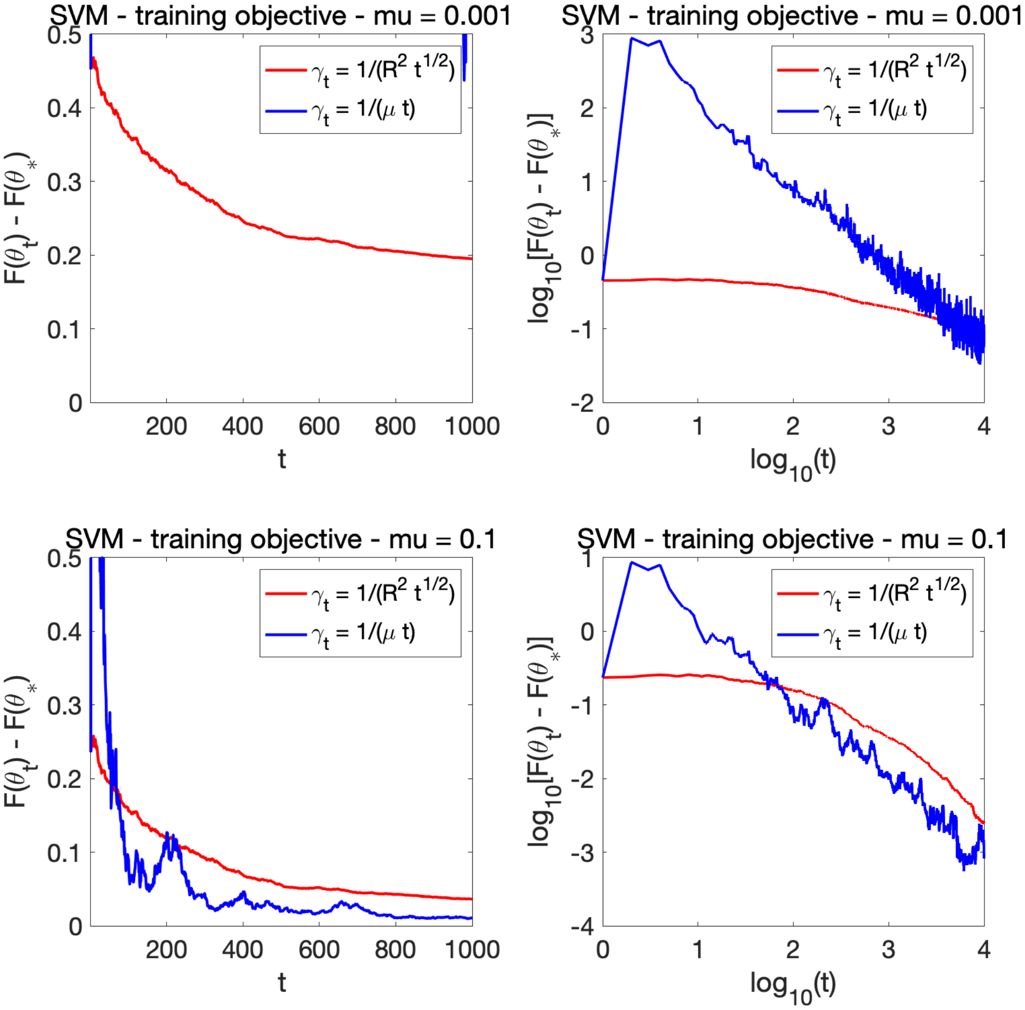
- For smooth problems, we can show a similar bound of the form $O(\kappa/t)$. For quadratic problems, constant step-sizes can be used with averaging, leading to improved convergence rates [11].
- The bound in $O(B^2/\mu t)$ is optimal for this class of problem. That is, as shown for example in [10], among all algorithms that can query stochastic gradients, having a better convergence rate (up to some constants) is impossible.
- We note that for the same regularized problem, we could use a step size proportion to $DB/\sqrt{t}$ and obtain a bound proportional to $DB/\sqrt{t}$, which looks worse thatn $B^2/(\mu t)$, but can in fact be better.

  Note also the loss of adaptivity: the step-size now depends on the difficulty of the problem (this was not the case for deterministic gradient descent).

  See experiments below for illustrations.

⚠ Check homogeneity

Experiments. We consider a simple binary classification problem with linear predictors and features with ℓ_2-norm bounded by R. We consider the hinge loss with a square ℓ_2 regularizer $\frac{\mu}{2}\|\cdot\|_2^2$. We measure the excess training objective. We consider two values of μ, and compare the two step-sizes $\gamma_t = 1/(R^2\sqrt{t})$ and $\gamma_t = \frac{1}{\mu t}$. We see that for large enough μ, the strongly-convex step-size is better. This is not the case for small μ.



The experiments above highlight the danger of a step-size equal to $1/(\mu t)$. In practice, it is often preferable to use $\gamma_t = \frac{1}{B^2\sqrt{t}+\mu t}$.

4.2 Variance reduction (♦♦)

We consider a finite sum $F(\theta) = \frac{1}{n}\sum_{i=1}^n f_i(\theta)$, where each f_i is R^2-smooth (for example logistic regression with features bounded by R in ℓ_2-norm), and which is such that F is μ-strongly convex.

Using SGD, the convergence rate can be shown to to $O(\kappa/t)$, with iterations of complexity $O(d)$, while for GD, the convergence rates is $O(\exp(-t/\kappa))$, but each iteration has complexity $O(nd)$. We now present a result allowing to get exponential convergence with an iteration cost which is $O(d)$.

The idea is to use a form of *variance reduction*, which is made possible by keeping in memory past gradients. We denote by $z_i^{(t)} \in \mathbb{R}^d$ the version of gradient i stored at time t.

The SAGA algorithm [12] is as follows.

- At every iteration, an index $i(t)$ is selected uniformly at random in $\{1, \dots, n\}$, and we perform the iteration $\theta_t = \theta_{t-1} - \gamma\Big[f'_{i(t)}(\theta_{t-1}) + \dfrac{1}{n}\displaystyle\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\Big]$ with $z_{i(t)}^{(t)} = f'_{i(t)}(\theta_{t-1})$ and all others z_i^t left unchanged (i.e., the same as $z_i^{(t-1)}$).

  The idea behind variance reduction is that if the random variable $z_{i(t)}^{(t-1)}$ (only considering the source of randomness coming from $i(t)$ is positively correlated with $f'_{i(t)}(\theta_{t-1})$, then the variance is reduced, and larger step-sizes can be used.

  As the algorithm converges, then $z_i^{(t)}$ converges to $f'_i(\eta_*)$, and then the update tends to have zero variance, and thus a constant step-size allows to obtain convergence. The key is then to show *simultaneously* that θ_t converges to η_* and that all $z_i^{(t)}$ converge to $f'_i(\eta_*)$, all at the same speed.

Theorem 5 (Convergence of SAGA) *If initializing with* $z_i^{(0)} = f'_i(\theta_0)$, *we have*

$$\mathbb{E}\big[\|\theta_t - \eta_*\|_2^2\big] \leqslant \Big(1 - \min\{\frac{1}{3n}, \frac{3\mu}{4R^2}\}\Big)^t\big(1 + \frac{n}{4}\big)\|\theta_0 - \eta_*\|_2^2.$$

Proof The proof consists in finding a Lyapunov function that decays along iterations.

Step 1. We first try our "usual" Lyapunov function, making the differences $\|z_i^{(t)} - f'_i(\theta_*)\|_2$ appear, with the update $\theta_t = \theta_{t-1} - \gamma\Box_t$, with $\Box_t = \big[f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\big]$,

$$\begin{aligned}
\|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top \Box_t + \gamma^2\big\|\Box_t\big\|_2^2 \text{ by expanding the square,}\\
\mathbb{E}_{i(t)}\|\theta_t - \eta_*\|_2^2 &= \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + \gamma^2\mathbb{E}_{i(t)}\Big\|f'_{i(t)}(\theta_{t-1}) + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)} - z_{i(t)}^{(t-1)}\Big\|_2^2\\
&\qquad \text{using the unbiasedness of the stochastic gradient,}\\
&\leqslant \|\theta_{t-1} - \eta_*\|_2^2 - 2\gamma(\theta_{t-1} - \eta_*)^\top F'(\theta_{t-1}) + 2\gamma^2\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\big\|_2^2\\
&\quad +2\gamma^2\mathbb{E}_{i(t)}\Big\|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)} + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)}\Big\|_2^2 \text{ using } \|a+b\|_2^2 \leqslant 2\|a\|_2^2 + 2\|b\|_2^2.
\end{aligned}$$

In order to bound $\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\big\|_2^2$, we use co-coercivity to get:

$$\begin{aligned}
\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\eta_*)\big\|_2^2 &= \frac{1}{n}\sum_{i=1}^n \big\|f'_i(\theta_{t-1}) - f'_i(\eta_*)\big\|_2^2 \leqslant \frac{1}{n}\sum_{i=1}^n R^2[f'_i(\theta_{t-1}) - f'_i(\eta_*)]^\top(\theta_{t-1}-\theta_*) \\
&\leqslant R^2F'(\theta_{t-1})^\top(\theta_{t-1}-\eta_*). \qquad (6)
\end{aligned}$$

In order to bound $\mathbb{E}_{i(t)}\big\|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)} + \frac{1}{n}\sum_{i=1}^n z_i^{(t-1)}\big\|_2^2$, we can simply use the identity $\mathbb{E}_{i(t)}\|Z - \mathbb{E}_{i(t)}Z\|_2^2 \leqslant \mathbb{E}_{i(t)}\|Z\|_2^2$. We thus get

$$\begin{aligned}
\mathbb{E}_{i(t)}\|\theta_t - \eta_*\|_2^2 &\leqslant \|\theta_{t-1}-\eta_*\|_2^2 - 2\gamma(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) + 2\gamma^2 R^2(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) \\
&\quad +2\gamma^2\frac{1}{n}\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2, \\
&\leqslant \|\theta_{t-1}-\eta_*\|_2^2 - 2\gamma(1-\gamma R^2)(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) + 2\frac{\gamma^2}{n}\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2.
\end{aligned}$$

Step 2. We see the term $\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2$ appearing, so we try to study how it varies across iterations. We have:

$$\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2 = \sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2 + \big\|f'_{i(t)}(\eta_*) - f'_{i(t)}(\theta_{t-1})\big\|_2^2 - \big\|f'_{i(t)}(\eta_*) - z_{i(t)}^{(t-1)}\big\|_2^2$$

Taking expectations with respect to $i(t)$, we get

$$\begin{aligned}
\mathbb{E}_{i(t)}\Big[\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2\Big] &= (1-\frac{1}{n})\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2 + \frac{1}{n}\sum_{i=1}^n \big\|f'_i(\eta_*) - f'_i(\theta_{t-1})\big\|_2^2 \\
&\leqslant (1-\frac{1}{n})\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2 + R^2(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}),
\end{aligned}$$

where we use the bound in Eq. (6). Thus, for a positive number Δ to be chosen later,

$$\begin{aligned}
\mathbb{E}_{i(t)}\Big[\|\theta_t-\eta_*\|_2^2 + \Delta\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2\Big] &\leqslant \|\theta_{t-1}-\eta_*\|_2^2 - 2\gamma(1-\gamma R^2 - \frac{R^2\Delta}{2\gamma})(\theta_{t-1}-\eta_*)^\top F'(\theta_{t-1}) \\
&\quad +[2\frac{\gamma^2}{n\Delta} + (1-1/n)]\Delta\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2.
\end{aligned}$$

With $\Delta = 3\gamma^2$ and $\gamma = \frac{1}{4R^2}$, we get:

$$\mathbb{E}_{i(t)}\Big[\|\theta_t-\eta_*\|_2^2 + \Delta\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t)}\big\|_2^2\Big] \leqslant \Big(1-\min\{\frac{1}{3n},\frac{3\mu}{4R^2}\}\Big)\Big[\|\theta_{t-1}-\eta_*\|_2^2 + \Delta\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(t-1)}\big\|_2^2\Big].$$

Thus

$$\mathbb{E}\big[\|\theta_t-\eta_*\|_2^2\big] \leqslant \Big(1-\min\{\frac{1}{3n},\frac{3\mu}{4R^2}\}\Big)^t\Big[\|\theta_0-\eta_*\|_2^2 + \frac{3}{16R^4}\sum_{i=1}^n \big\|f'_i(\eta_*) - z_i^{(0)}\big\|_2^2\Big].$$

If initializing with $z_i^{(0)} = f_i'(\theta_0)$, we get the desired bound by using the Lipschitz-continuity of each f_i'. ■

We can make the following observations:

- The contraction rate after one iteration is $\Big(1 - \min\{\frac{1}{3n}, \frac{3\mu}{4R^2}\}\Big) \leqslant \exp\Big(\min\{-\frac{1}{3n}, \frac{3\mu}{4R^2}\}\Big)$. Thus, after an "effective pass" over the data, that is, n iterations, the contracting rate is $\exp\Big(\min\{-\frac{1}{3}, \frac{3\mu n}{4R^2}\}\Big)$. It is only an effective pass, because after we sample n indices with replacement, we will not see all functions.

  In order to have a contracting effect of ε, that is, having $\|\theta_t - \eta_*\|_2^2 \leqslant \varepsilon\|\theta_0 - \eta_*\|_2^2$, we need to have $\exp\Big(t \min\{-\frac{1}{3n}, \frac{3\mu}{4R^2}\}\Big) n \leqslant \varepsilon$, which is equivalent to

  $$t \geqslant \max\{3n, \frac{4R^2}{3\mu}\} \log \frac{n}{\varepsilon}.$$

  It just suffices to have $t \geqslant \Big(3n + \frac{4R^2}{3\mu}\Big) \log \frac{n}{\varepsilon}$, and thus the running time complexity is equal to d times the minimal number, that is

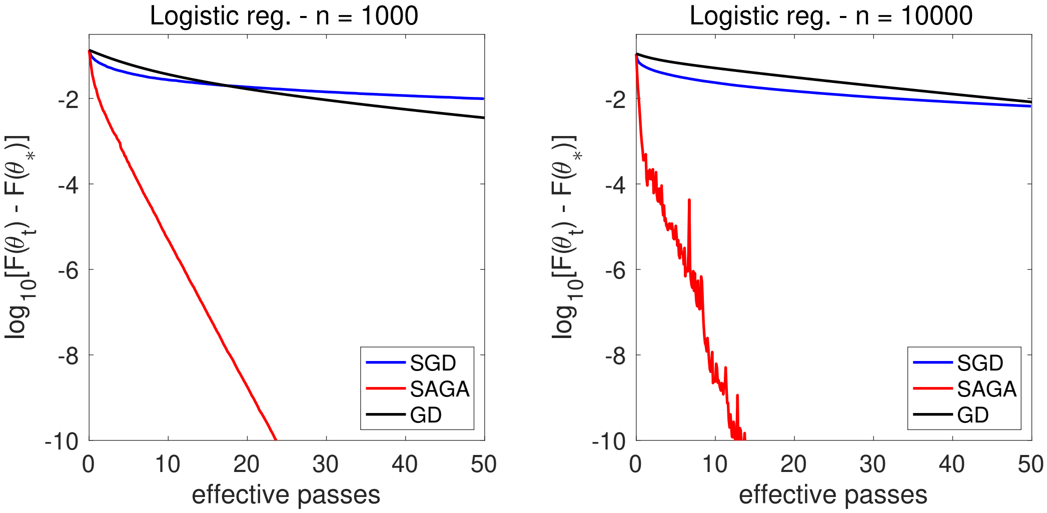
  $$d\Big(3n + \frac{4R^2}{3\mu}\Big) \log \frac{n}{\varepsilon}.$$

  This to be contrasted with batch gradient descent with step-size $\gamma = 1/R^2$ (which is the simplest step-size that can be computed easily), whose complexity is $dn\frac{R^2}{\mu} \log \frac{n}{\varepsilon}$. We replace the product of n and condition number $\frac{R^2}{\mu}$ by a sum, which is significant where κ is large.

- Multiple extensions of this result are available, such as a rate for non-strongly-convex functions, adaptivity to strong-convexity, proximal extensions, acceleration. It is also worth mentioning that the need to store past gradients can be alleviated (see [13] for more details).

- Note that these fast algorithms allow to get very small optimization errors, and that the best testing risks will typically obtained after a few (10 to 100) passes.

Experiments. We consider ℓ_2-regularized logistic regression and we compare GD, SGD and SAGA, all with their corresponding step-sizes coming from the theoretical analysis, with two values of n (left: small, right: large). We see that for early iterations, SGD dominates GS, while for larger numbers of iterations, GD is faster. This last effect is not seen for large numbers of observations (right). In the two cases, SAGA gets to machine precision after 50 effective passes over the data.



5 Conclusion

- We can now provide a summary of convergence rates below, with the main rates that we have seen in this lecture (and some that we have not seen). We separate between convex and strongly convex, and between smooth and non-smooth, as well as between deterministic and stochastic methods. Below, L is the smoothness constant, μ the strong convexity constant, B the Lipschitz constant and D the distance to optimum at the initialization.

	convex	strongly convex
nonsmooth	deterministic: $BD/\sqrt{t}$ stochastic: $BD/\sqrt{t}$	deterministic: $B^2/(t\mu)$ stochastic: $B^2/(t\mu)$
smooth	deterministic: LD^2/t^2 stochastic: $LD^2/\sqrt{t}$ finite sum: n/t	deterministic: $\exp(-t\sqrt{\mu/L})$ stochastic: $L/(t\mu)$ finite sum: $\exp(-\min\{1/n, \mu/L\}t)$

- Note that many important themes in optimization have been ignored, such as Frank-Wolfe methods, coordinate descent, duality. See [1, 2] for further details. See also Lectures 6 and 8 for optimization methods for kernel methods and neural networks.

Acknowledgements

These class notes have been adapted from the notes of many colleagues I have the pleasure to work with, in particular Lénaïc Chizat, Pierre Gaillard, Alessandro Rudi and Simon Lacoste-Julien.

References

[1] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[2] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[3] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 1996.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[5] Rajendra Bhatia. *Positive Definite Matrices*, volume 24. Princeton University Press, 2009.

[6] Nikhil Bansal and Anupam Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(1):1–32, 2019.

[7] Yurii Nesterov. *Introductory Lectures on Convex Optimization: a Basic Course.* Kluwer, 2004.

[8] Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep*, 76, 2007.

[9] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[10] Alekh Agarwal, Martin J. Wainwright, Peter L. Bartlett, and Pradeep K. Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, 2009.

[11] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems*, 2013.

[12] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 2014.

[13] Robert M. Gower, Mark Schmidt, Francis Bach, and Peter Richtarik. Variance-reduced methods for machine learning. Technical Report 2010.00892, arXiv, 2020.