PTML 3: 8/04/2022

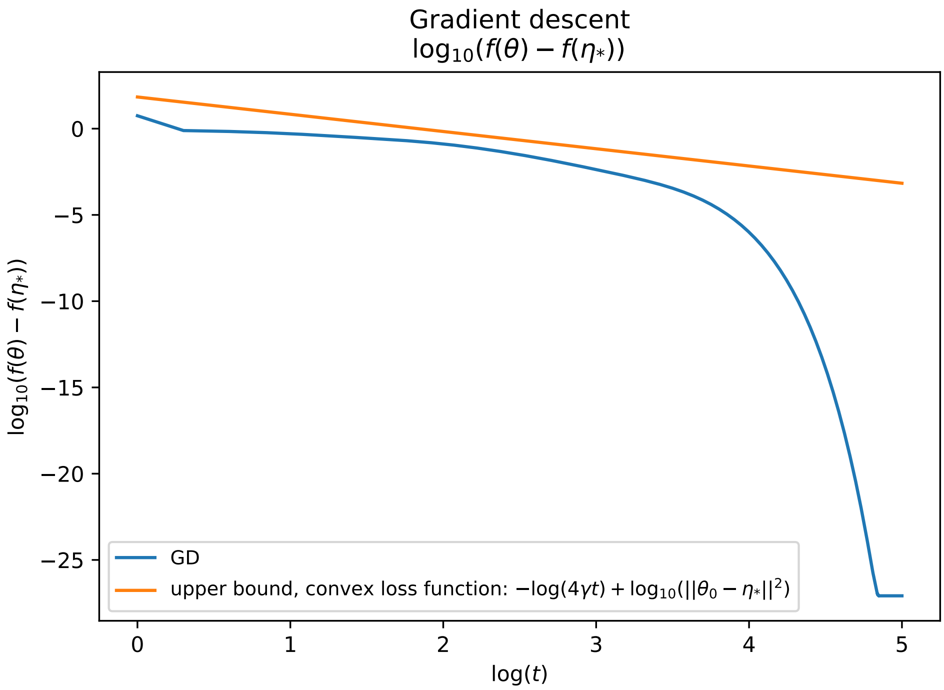

TABLE DES MATIÈRES

1 GRADIENT DESCENT ON A LEAST-SQUARES PROBLEM

1.1 Setting

In this exercise we will study gradient descent (GD) for a least-squares problem.

— $\mathcal{X} = \mathbb{R}^d$

— $\mathcal{Y} = \mathbb{R}$

— Design matrix : X

— Outputs : $y \in \mathbb{R}^n$.

We want to minimize the function f representing the empirical risk :

$$f(\theta) = \frac{1}{2n}||X\theta - y||^2 \tag{1}$$

We recall that the gradient and the Hessian write :

$$\begin{aligned} \nabla_\theta f &= \frac{1}{n}X^T(X\theta - y) \\ &= H\theta - \frac{1}{n}X^T y \end{aligned} \tag{2}$$

$$H = \frac{1}{n}X^T X \tag{3}$$

We note the gradient update $\theta_{t+1} = \theta_t - \gamma \nabla_{\theta_t} f$

We note η^* the minimizers of f. If H is not invertible, they might be not unique and all verify

$$\nabla_{\eta^*} f = 0 \tag{4}$$

This means that

$$H\eta* = \frac{1}{n}X^T y \tag{5}$$

If f is strongly convex, η^* is unique.

1.1.1 *Alternative formulation*

It is often convenient to note that the minimization of f is equivalent to the minimization of the quadratic function

$$g(\theta) = \frac{1}{2}\theta^T H\theta - b^T\theta \tag{6}$$

with $b = \frac{1}{n}X^T y$. Indeed,

$$\begin{aligned} f(\theta) &= \frac{1}{2n}||X\theta - y||^2 \\ &= \frac{1}{2n}\langle X\theta - y, X\theta - y\rangle \\ &= \frac{1}{2n}\Big(\langle X\theta, X\theta\rangle - 2\langle X\theta, y\rangle + \langle y, y\rangle\Big) \\ &= \frac{1}{2n}\Big(\theta^T X^T X\theta - 2(X^T y)^T\theta + ||y||^2\Big) \\ &= \frac{1}{2n}\Big(\theta^T X^T X\theta - 2(X^T y)^T\theta + ||y||^2\Big) \\ &= \frac{1}{2}\theta^T H\theta - \frac{1}{n}(X^T y)^T\theta + \frac{1}{2n}||y||^2 \\ &= g(\theta) + \frac{1}{2n}||y||^2 \end{aligned} \tag{7}$$

Hence, the gradients of f and g are identical, and minimizing g is equivalent to minimizing f.

1.1.2 *Positivity of H*

As $H = \frac{1}{n}X^T X$, H is symmetric. We recall that it is also positive semi-definite (matrice positive), meaning that all its eigenvalues are non-negative. Indeed, let λ be such an eigenvalue, with associated eigenvector u_λ.

$$\langle Hu, u\rangle = \langle \lambda u, u\rangle = \lambda||u||^2 \tag{8}$$

But we also have

$$\begin{aligned} \langle Hu, u\rangle &= \langle X^T Xu, u\rangle \\ &= \langle Xu, Xu, \rangle \\ &= ||Xu||^2 \\ &\geqslant 0 \end{aligned} \tag{9}$$

Hence,**all eigenvalues of H are non-negative**. We note μ the smallest eigenvalue of H.

1.1.3 *Smoothness of H*

We have also seen that the convergence garantees of gradient descent depend on the**smoothness** of H. Let L be the largest eigenvalue of L. We can show that f is L-smooth.
To do so, we use the fact that $\forall x \in \mathbb{R}^d$,

$$||Hx|| \leqslant L||x|| \tag{10}$$

This can be proven by decomposing x in a basis of $\mathbb{R}^d$ made of orthogonal eigenvectors of H. Then, for all θ and θ',

$$\begin{aligned} ||\nabla_\theta f - \nabla_{\theta'} f|| &= ||H(\theta - \theta')|| \\ &\leqslant |||H||| \times ||\theta - \theta'|| \end{aligned} \tag{11}$$

Which shows the L-smoothness of f.

1.1.4 *Condition number*

We note κ the condition number, $\kappa = \frac{L}{\mu}$. By convention, if $\mu = 0$, $L = +\infty$.

1.2 Gradient descent

As we have seen during the lectures, the convexity or strong convexity of the objective function f is determined by H.

— If H is positive-definite (matrice définie positive), meaning that $\mu > 0$, f is μ-strongly convex.

— Is H is simply positive semi-definite, for instance if $\mu = 0$, then we only know that f is convex.

1.2.1 *Strongly convex function*

If $\mu > 0$, f is μ-convex and we have seen that we have**exponential convergence** for a good choice of γ. With $\gamma = \frac{1}{L}$, we obtain an exponential convergence

$$||\theta_t - \eta^*||_2^2 \leqslant \exp(-\frac{2t}{\kappa})||\theta_0 - \eta^*||_2^2 \tag{12}$$

Here, t represents the number of iterations. The characteristic convergence time is κ. We can also state that

$$\log(||\theta_t - \eta^*||_2^2) \leqslant -\frac{2t}{\kappa} + \log(||\theta_0 - \eta^*||_2^2) \tag{13}$$

Note that other choices of γ are possible, such as $\gamma = \frac{2}{\mu+L}$ [Bach, 2021,].

Exercice 1 : Use the files**TP_3 _GD_strongly_convex.py** and**TP_3_utils.py** in order to observe the exponential convergence for a strongly convex loss function.
You can generate different data. You should observe results like Figure 1 and 2.

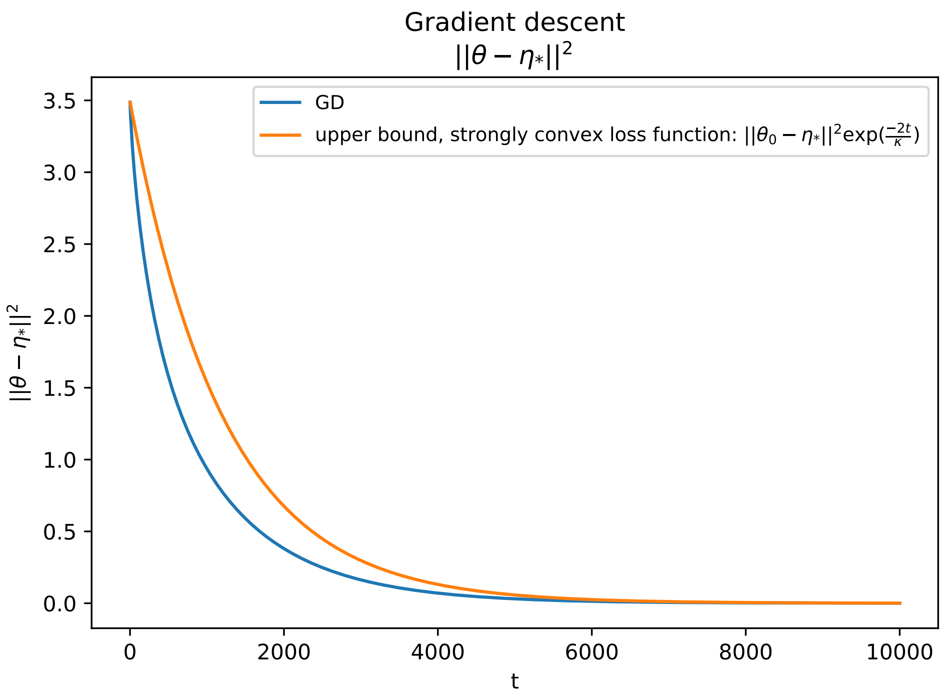

FIGURE 1 – GD, strongly convex loss function

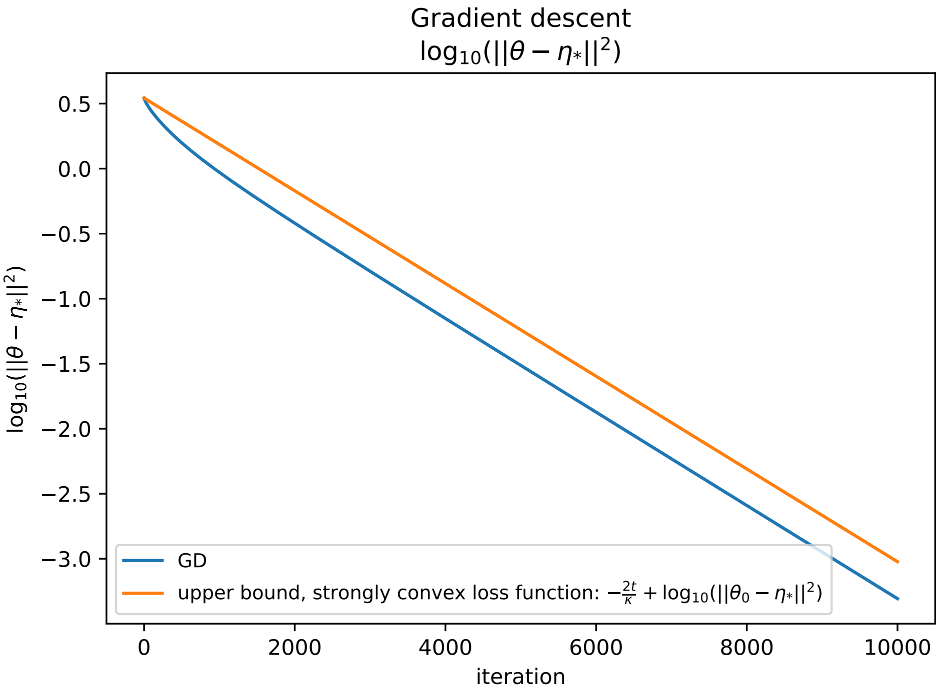

FIGURE 2 – GD, strongly convex loss function, semi-logarithmic scale

1.2.2 *Convex function*

If $\mu = 0$, we have seen that with $\gamma \leqslant \frac{1}{L}$,

$$f(\theta_t) - f(\eta^*) \leqslant \frac{1}{4t\gamma} ||\theta_0 - \eta^*||_2^2 \tag{14}$$

We can also state that

$$\log\left(f(\theta_t) - f(\eta^*)\right) \leqslant -\log(4t\gamma) + \log(||\theta_0 - \eta^*||_2^2) \tag{15}$$

We will study an example where X is not injective, hence H is not invertible. In such a setting, we can not use the OLS estimator in order to monitor convergence, as in the previous exercice. Instead, we will generate a random η^* and output vector $y \in \mathbb{R}^n$.

Exercice 2 : Use the files**TP_3 _GD_convex.py** and**TP_3_utils.py** in order to observe the exponential convergence for a convex loss function.

You can generate different data. You should observe results like Figure 3 and 4.

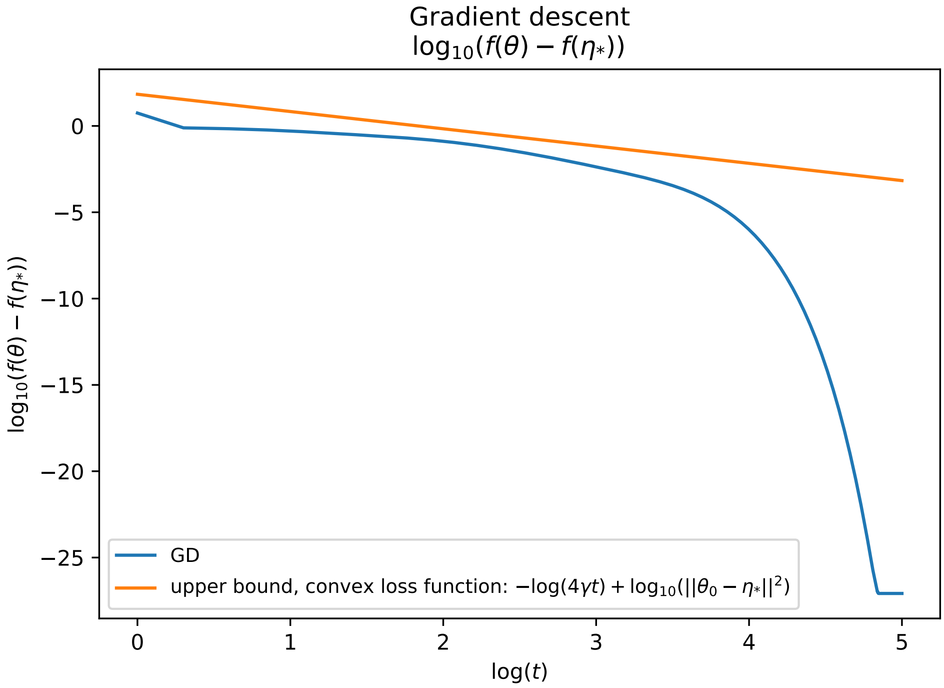

Figure 3 – GD, convex loss function, logarithmic scale.

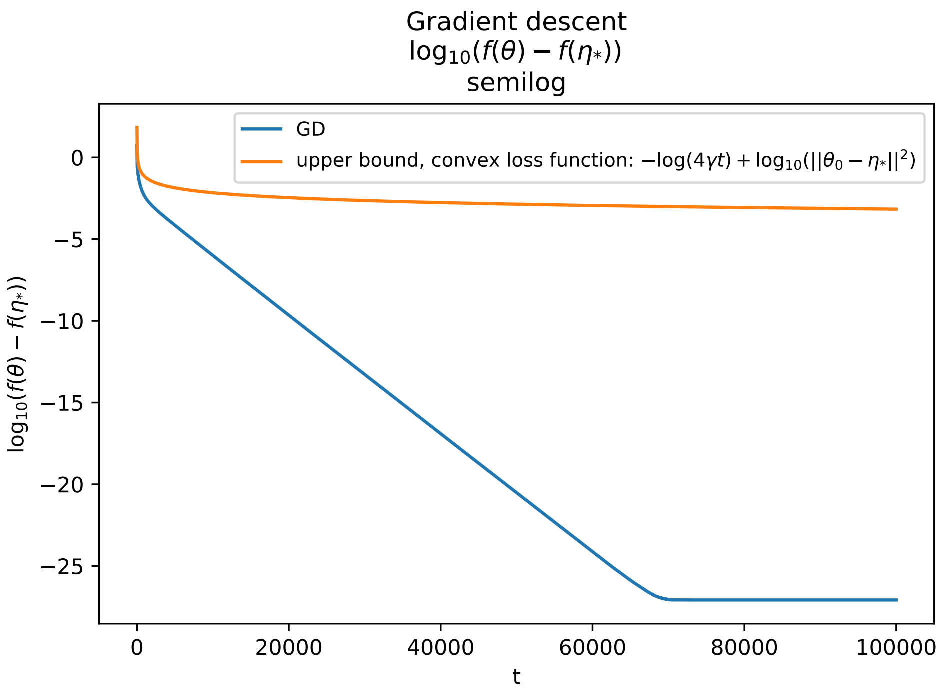

Figure 4 – GD, convex loss function, semi-logarithmic scale

It seems that with this function, we observe

- a phase of convergence approximately in the form of $\mathcal{O}(\frac{1}{t})$, since $\log_{10}\big(f(\theta) - f(\eta_*)\big)$ decreases approximately as $-\log(t)$ (figure 3).
- a phase of exponential convergence, approximately when $\log(t) \geqslant 4$ (the exponential convergence can be seen in figure 4, where $\log_{10}\big(f(\theta) - f(\eta_*)\big)$ is linear with t, with a negative slope.

Exercice 3 : Why do we have these two regimes one after the other ?

1.3 The heavy-ball method

When κ is very large, the convergence might become very slow. Some methods exist in order to speed it up, such as**Heavy-ball.** This method consists in adding a**momentum term** to the gradient update term, such as the iteration now writes

$$\theta_{t+1} = \theta_t - \gamma \nabla_{\theta_t} f + \beta(\theta_t - \theta_{t-1}) \tag{16}$$

The update $\theta_{t+1} - \theta_t$ is then a combination of the gradient $\nabla_{\theta_t} f$ and of the previous update $\theta_t - \theta_{t-1}$. The goal of this method it might balance the effet of oscillations in the gradient.

We will use these parameters :

$$\gamma = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2} \tag{17}$$

and

$$\beta = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\right)^2 \tag{18}$$

The heavy-ball method is called an*inertial method.* When f is a general convex function (not necessary quadratic), some generalizations exist, such as**Nesterov acceleration.**

1.3.1 *Impact on convergence rate*

Assuming $\mu > 0$, we will show that the characteristic convergence time with the heavy-ball momentum term is $\sqrt{\kappa}$ instead of κ.

Let λ be an eigenvalue of H and u_λ a eigenvector for thie eigenvalue. We are interested in the evolution of $\langle \theta_t - \eta^*, u_\lambda \rangle$.

We note

$$a_t = \langle \theta_t - \eta^*, u_\lambda \rangle \tag{19}$$

Exercice 4 : Show that

$$a_{t+1} = (1 - \gamma\lambda + \beta)a_t - \beta a_{t-1} \tag{20}$$

Exercice 5 : **(Optional)** Compute the constant-recursive sequence a_t and show that $a_t \leqslant (\sqrt{\beta})^t C_\lambda$, where C_λ is a constant that depends on the initial conditions.

https://en.wikipedia.org/wiki/Constant-recursive_sequence

If u_i is a basis of orthogonal normed vectors with eigenvalues λ_i, we then have that

$$\begin{aligned} ||\theta_t - \eta^*||^2 &= \sum_{i=1}^{d} (\langle \theta_t - \eta^*, u_i \rangle)^2 \\ &\leqslant \sum_{i=1}^{d} (\sqrt{\beta})^{2t} C_{\lambda_i} \\ &= (\sqrt{\beta})^{2t} D \end{aligned} \tag{21}$$

with

$$D = \sum_{i=1}^{d} C_{\lambda_i} \tag{22}$$

We can now remark that

$$\begin{aligned}\sqrt{\beta} &= \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}} \\ &= \frac{1-\sqrt{\frac{\mu}{L}}}{1+\sqrt{\frac{\mu}{L}}} \\ &\leqslant 1-\sqrt{\frac{\mu}{L}} \\ &= 1-\frac{1}{\sqrt{\kappa}}\end{aligned} \tag{23}$$

Finally, as $1-\frac{1}{\sqrt{\kappa}} \leqslant \exp(-\frac{1}{\sqrt{\kappa}})$,

$$||\theta_t - \eta^*||^2 = \mathcal{O}(\exp(-\frac{2t}{\sqrt{\kappa}})) \tag{24}$$

Conclusion : with the heavy-ball momentum term, we changed the convergence rate of $\mathcal{O}(\exp(-\frac{2t}{\kappa}))$ to a convergence rate of $\mathcal{O}(\exp(-\frac{2t}{\sqrt{\kappa}}))$. This means that characteristic convergence time went from κ to $\sqrt{\kappa}$. If κ is large, which is the case we are interested in, this can be a great improvement.

Remember that $\kappa = \frac{L}{\mu}$, and that μ may be very small when n or d is large. For instance, in the case of Ridge regression, we have seen in the previous session that for instance, μ can be of order $\mathcal{O}(\frac{1}{\sqrt{n}})$ (see the computation of the optimal regularisation parameter). Hence, κ may be of order $\sqrt{n}$ or higher.

1.3.2 *Simulation*

Exercice 6 : Use the files**TP_3 _GD_strongly_convex _heavy_ball.py** to implement the Heavy-ball method and compare the convergence speed results to that of GD. You should obtain something like figures 5 and 6.

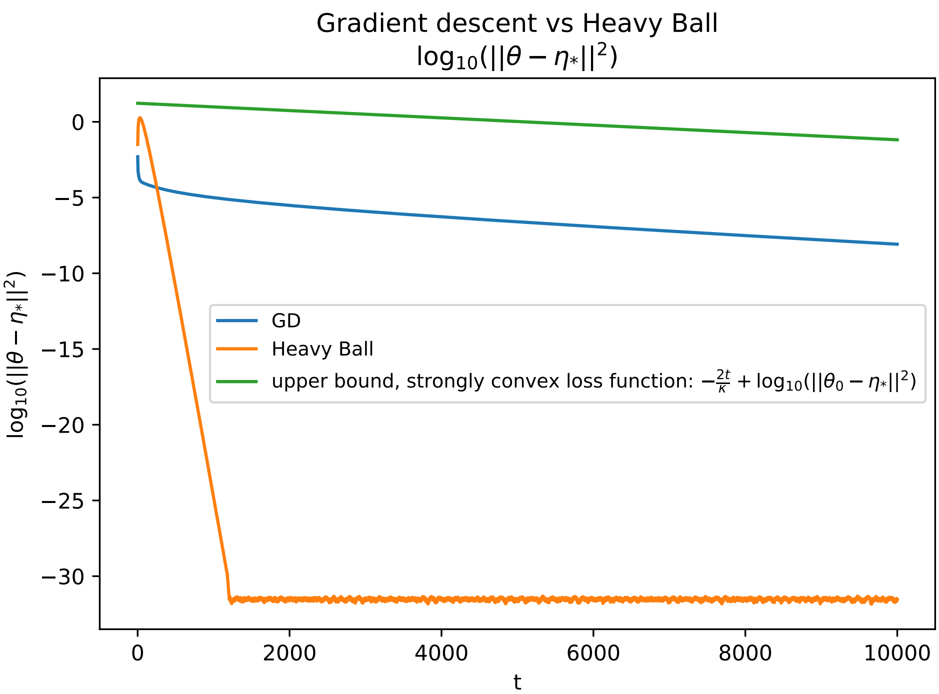

FIGURE 5 – Heavy-ball vs GD

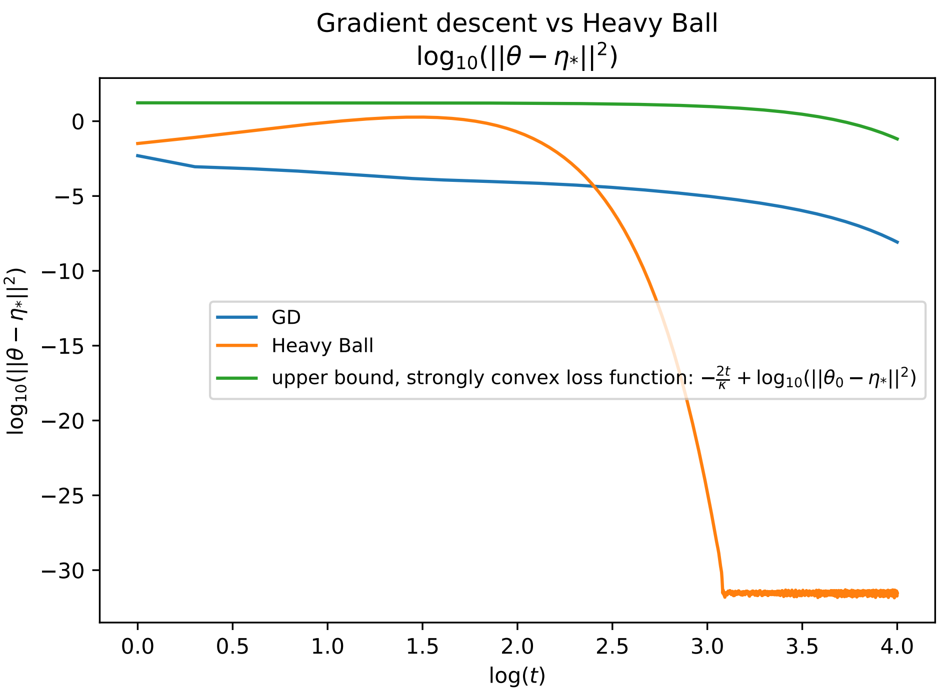

FIGURE 6 – Heavball vs GD, logarithmic scale

1.4 Line search

Considering an fixed iteration step θ_t, we note

$$\alpha(\gamma) = \theta_t - \gamma \nabla_{\theta_t} f \tag{25}$$

1.4.1 *Exact line search*

The**exact line seach** method attempts to find the optimal step γ^*, at each iteration. This means, given the position θ_t, the parameter γ that minimizes the function defined by

$$\begin{aligned} g(\gamma) &= f(\theta_t - \gamma \nabla_{\theta_t} f) \\ &= f(\alpha(\gamma)) \end{aligned} \tag{26}$$

We note that

$$\begin{aligned} \nabla_{\alpha(\gamma)} f &= H\alpha(\gamma) - \frac{1}{n} X^T y \\ &= H(\theta_t - \gamma \nabla_{\theta_t} f) - \frac{1}{n} X^T y \\ &= \nabla_{\theta_t} f - \gamma H \nabla_{\theta_t} f \end{aligned} \tag{27}$$

We can derivate g with respect to γ.

$$\begin{aligned} g'(\gamma) &= \langle \nabla_{\alpha(\gamma)} f, -\alpha'(\gamma) \rangle \\ &= -\langle \nabla_{\theta_t} f - \gamma H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \\ &= -\langle \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle + \langle \gamma H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \\ &= -||\nabla_{\theta_t} f||^2 + \gamma \langle H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \end{aligned} \tag{28}$$

In order to cancel the derivative, we must have that

$$\gamma^* = \frac{||\nabla_{\theta_t} f||^2}{\langle H\nabla_{\theta_t} f, \nabla_{\theta_t} f\rangle} \tag{29}$$

We note that this is correct if $\nabla_{\theta_t} f \neq 0$. If $\nabla_{\theta_t} f = 0$, this means that $\theta_t = \eta^*$, as f is convex.

This computation may then be done at each iteration.

An important remark is that if we note $\theta^*_{t+1} = \theta_t - \gamma^* \nabla_{\theta_t} f = \alpha(\gamma^*)$, then equation 28 shows that

$$\langle \nabla_{\theta^*_{t+1}} f, \nabla_{\theta_t} f\rangle = 0 \tag{30}$$

Two optimal directions of the gradient updates are**orthogonal**. Importantly, this is true in the general case, not only for least-squares.

1.4.2 *Simulation*

Exercice 7 : Use**TP_3 _GD_strongly_convex _line_search.py** in order to implement the exact line search method. You should obtain something like figures 7 and 8

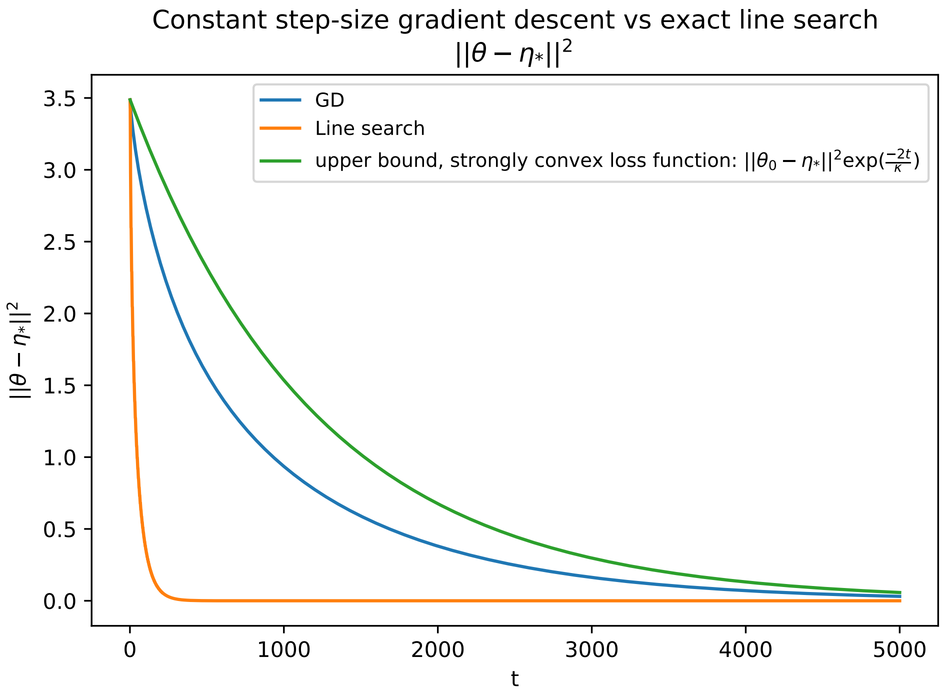

Figure 7 – Line search vs constant step-size gradient descent

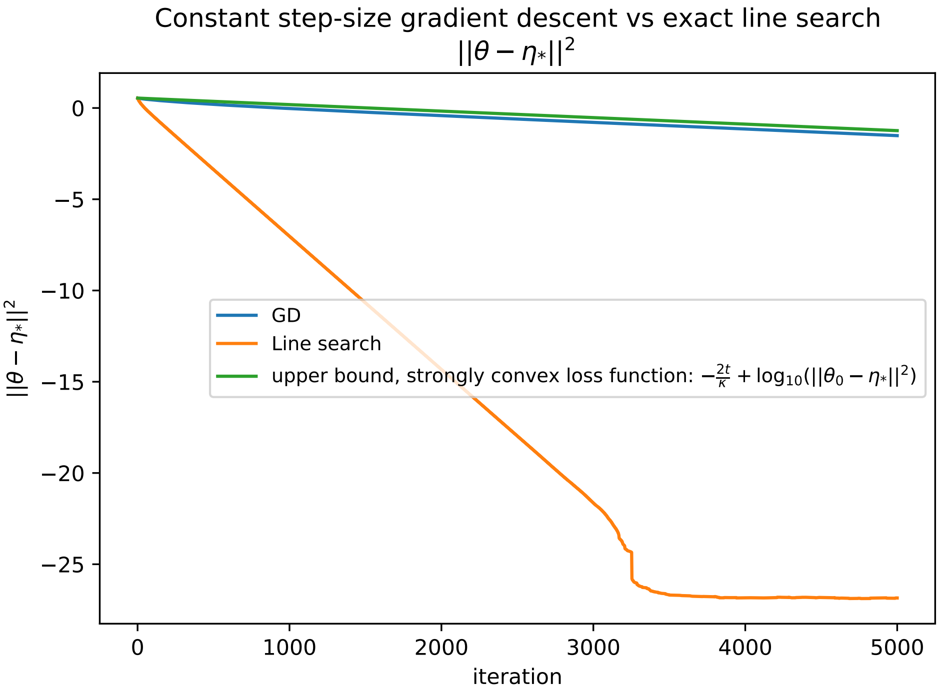

Figure 8 – Line search vs constant step-size gradient descent, semi logarithmic scale.

1.4.3 *Backtracking line search*

In many practical situations, it is not possible to compute explicitely the optimal step γ^*. Or it could be possible, but too expensive computationnally.

In such situations, it is possible to compute an approximation of γ^*, for instance using**backtracking line search.** This method attempts to find a good γ by trying several decreasing values until a sufficient decrease in f after the gradient update is obtained.

https://en.wikipedia.org/wiki/Backtracking_line_search

2 RIDGE REGRESSION

2.1 Setting

We recall that when doing Ridge regression, we minimize the regularized risk

$$f(\theta) = \frac{1}{2n}||Y - X\theta||_2^2 + \frac{\nu}{2}||\theta||_2^2 \quad (31)$$

As in 6, it is convenient to note that this risk minimization is equivalent to the minimization of a quadratic function

$$g(\theta) = \frac{1}{2}\theta^T G\theta - b^T\theta \quad (32)$$

with

$$G = H + \nu I_d \quad (33)$$

and

$$b = \frac{1}{n}X^T y \quad (34)$$

Indeed using 7,

$$\begin{aligned} f(\theta) &= \frac{1}{2n}\|X\theta - y\|^2 + \frac{\nu}{2}\|\theta\|_2^2 \\ &= \frac{1}{2}\theta^T H\theta - \frac{1}{n}(X^T y)^T\theta + \frac{\nu}{2}\langle\theta,\theta\rangle \\ &= \frac{1}{2}\theta^T(H + \nu I_d)\theta - \frac{1}{n}(X^T y)^T\theta + \frac{1}{2n}\|y\|^2 \end{aligned} \tag{35}$$

We note that G is a symmetric definite-positive matrix.

2.2 Simulations

We assume that $d > n$. This means that H is not invertible. Indeed, as $X \in \mathbb{R}^{n,d}$ is of rank at most n, its columns are not linearly independent, and is not injective. There exists $u_0 \in \mathbb{R}^d$ such that $u_0 \neq 0$ and $Xu_0 = 0$. Then, $Hu_0 = X^T X u_0 = 0$ and the smallest eigenvalue of H is 0. Finally, the smallest eigenvalue of G is ν.

Exercice 8 : Implement GD on a Ridge regression problem, using**TP_3_GD_strongly_convex_ridge.py**

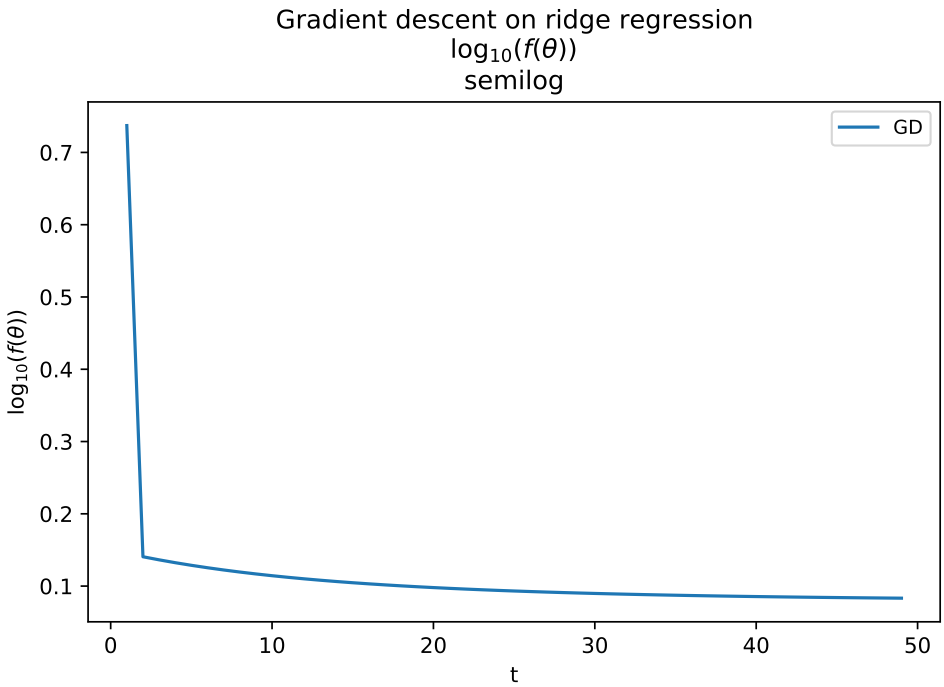

FIGURE 9 – Gradient descent on ridge regression

RÉFÉRENCES

[Bach, 2021] Bach, F. (2021). Learning Theory from First Principles Draft. Book Draft, page 229.