MATH-329 Nonlinear optimization
Exercise session 2: Gradient Descent

Instructor: Nicolas Boumal
TAs: Andreea Musat and Andrew McRae

Document compiled on September 11, 2024

1. Quadratic functions and condition number. Let $\mathcal{E} = \mathbb{R}^n$ with the usual inner product. Consider the quadratic function $f : \mathcal{E} \to \mathbb{R}$ defined by

$$f(x) = \frac{1}{2}x^\top Ax + b^\top x + c,$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric and nonzero, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}$.

1. Give an expression for the gradient of f. What is the set of critical points? Argue that it is nonempty if and only if b is in the image of A.

2. Show that if f is lower-bounded then b is in the image of A. *Hint: remember that we have* $\mathrm{im}(A) = \ker(A)^\perp$ *because A is symmetric. Apply f to vectors from the null space of A.*

From now we assume that b is in the image of A and we let $d \in \mathcal{E}$ be a vector such that $Ad = -b$.

3. For all $x \in \mathcal{E}$ find an expression for $f(x + d)$. Use it to deduce that f is lower-bounded if and only if A is positive semidefinite.

The last two questions showed that f is lower-bounded if and only if A is positive semidefinite and $b \in \mathrm{im}(A)$. We assume that these conditions hold; otherwise minimizing f would not make sense.

4. Argue that f attains its minimum value. What is the set of global minima? Under what condition is there a unique global minimum?

5. Does f admit local minima that are not global?

6. Show that ∇f is Lipschitz continuous. What is the smallest Lipschitz constant L?

We found that f is lower-bounded and has Lipschitz continuous gradients: that's all the properties we need to apply gradient descent with constant step-size. In Question 4, you should have found that global minima of f coincide with the solutions of a linear system of equations. This leads to a dual perspective: we could use standard linear algebra algorithms such as Gaussian elimination to minimize f... Or: we could apply optimization algorithms to f to solve the linear system. We adopt this second viewpoint here. To perform an iteration of gradient descent we only need to compute a matrix-vector product with A. If A is structured (for example if it is sparse) this operation can be done efficiently even when A is huge.

From now we consider the case where f has a unique global minimum. For a symmetric matrix A, we define the condition number $\kappa \geq 1$ as the ratio of its maximal to minimal eigenvalues, that is,

$$\kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

7. For $n = 2$ plot the level sets of f around its global optimum for $\kappa = 1$ and $\kappa = 5$. We can choose A diagonal, $b = 0$ and $c = 0$ for simplicity. For example:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{and} \qquad A = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}.$$

   In what situation do you expect gradient descent to work the best?

8. Write a script to run gradient descent with constant step-sizes $1/L$. Choose a random initial point. Try with other step-sizes, for example $1/2L$ and $2/L$. Plot the sequence of points that gradient descent outputs along with the level sets of f. What do you observe?

9. Can you improve the practical behavior of the algorithm with a linesearch method? In particular, can you solve the linesearch problem exactly?

Answer.

1. The gradient of f is given by

$$\nabla f(x) = Ax + b.$$

   The set of critical points is the set $\{x \in \mathbb{R}^n \mid \nabla f(x) = 0\}$, that is, the points $x \in \mathbb{R}^n$ such that

$$Ax = -b.$$

   This linear system has a solution if and only if b is in the image of A. Note that there is a unique solution when A is invertible.

2. Let u be a vector in the kernel of A. For all $t \in \mathbb{R}$ we have $f(tu) = tb^\top u + c$. If $b^\top u$ is nonzero then either $f(tu)$ or $f(-tu)$ will go to $-\infty$ when $t \to \infty$. So b is orthogonal to the kernel of A and we deduce that $b \in \mathrm{im}(A)$ (using the hint).

3. We let d be a vector such that $Ad = -b$. We find that

$$\begin{aligned} f(x+d) &= \frac{1}{2}(x^\top + d^\top)A(x+d) + b^\top(x+d) + c \\ &= \frac{1}{2}x^\top Ax + d^\top Ax + \frac{1}{2}d^\top Ad + b^\top x + b^\top d + c \\ &= \frac{1}{2}x^\top Ax + \frac{1}{2}d^\top Ad + b^\top d + c. \end{aligned}$$

   If A has a negative eigenvalue then clearly f is not lower-bounded. Indeed if we let $u \in \mathcal{E}, \lambda < 0$ such that $Au = \lambda u$ then we find that for all $t \in \mathbb{R}$

$$f(tu) = \frac{\lambda t^2}{2}\|u\|^2 + \frac{1}{2}d^\top Ad + b^\top d + c.$$

   From this we have $\lim_{t\to\infty} f(tu) = -\infty$. Conversely if A is positive semidefinite then the function $x \mapsto x^\top Ax$ is lower-bounded by 0. So we deduce that f is lower-bounded by $\frac{1}{2}d^\top Ad + b^\top d + c$.

4. From Question 3 we know that for all $x \in \mathcal{E}$ we have

$$f(x+d) = \frac{1}{2}x^\top Ax + \frac{1}{2}d^\top Ad + b^\top d + c.$$

Taking $x = 0$ we find that $f(d) = \frac{1}{2}d^\top Ad + b^\top d + c$. We showed in Question 3 that f is lower-bounded by this quantity so this proves that f attains its minimum: all vectors d such that $Ad = -b$ are global minimizers. From Question 1 we know that they are exactly the critical points of f. Conversely, all global minimizers must be critical points. So we conclude that the set of global minima is the set of critical points, that is, the vectors d such that $Ad = -b$. This system of equation has a unique solution if and only if A is invertible. Since A is positive semidefinite this condition is equivalent to A being positive definite.

5. All local minima are global minima. Indeed, if x^* is a local minimizer then it is a critical point, and we know from the previous questions that critical points are global minima.

6. For all $x, y \in \mathbb{R}^n$ we have

$$\begin{aligned}\|\nabla f(x) - \nabla f(y)\| &= \|A(x-y)\| \\ &\leq \|A\|_2 \|x-y\|,\end{aligned}$$

where $\|A\|_2$ is the spectral norm (or operator norm) of A. Remember that the spectral norm is given by

$$\begin{aligned}\|A\|_2 &= \sup_{x \in \mathbb{R}^n,\, x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} \\ &= \sup_{x \in \mathbb{R}^n,\, \|x\|_2 = 1} \|Ax\|_2.\end{aligned}$$

When the matrix A is positive semidefinite the spectral norm is the maximal eigenvalue of A, that is, $\|A\|_2 = \lambda_{\max}(A)$.

There is no better Lipschitz constant than $\|A\|_2$. Indeed, suppose there exist $L < \|A\|_2$ such that for all $x, y \in \mathbb{R}^n$ we have

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

The set $\{x \in \mathbb{R}^n \mid \|x\| = 1\}$ (unit sphere) is compact, so there exist a vector $z \in \mathbb{R}^n$ with unit norm such that

$$\|Az\| = \|A\|_2.$$

(You may for example take the eigenvector associated to the largest eigenvalue of A). Now with $x = z$ and $y = 0$ we find that

$$\begin{aligned}\|\nabla f(x) - \nabla f(y)\| &= \|Az\| \\ &= \|A\|_2 \\ &\leq L,\end{aligned}$$

which contradicts the definition of L.

7. Figure 1 shows the level sets of f for two different values of κ. When $\kappa = 1$ the gradient is always pointing towards the global minimum. When the condition number is larger the gradient's direction can be completely off. For this reason we expect gradient descent to work better when κ is close to 1.

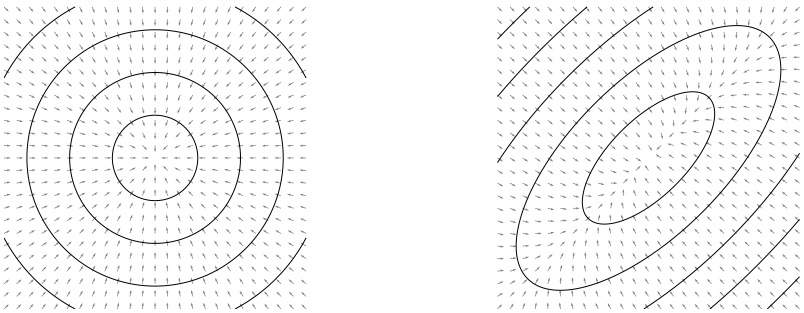

Figure 1: Level sets of the function f where A has condition numbers $\kappa = 1$ (left) and $\kappa = 5$ (right). Arrows represent the direction of the (normalized) gradient.

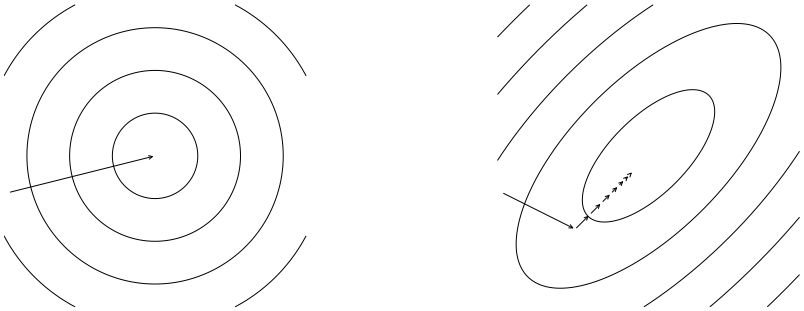

Figure 2: Gradient descent with ideal step-size $1/L$.

8. Figure 2 shows the iterates of gradient descent with the step-size $1/L$. When $\kappa = 1$ gradient descent converges to the optimum in one single iteration. Figure 3 shows the iterates when the step-size are not exactly $1/L$. When they are larger than $2/L$ gradient descent doesn't converge anymore.

Figure 3: Gradient descent when $\kappa = 5$. Step-sizes are $1/2L$ (left), $7/4L$ (middle) and $2/L$ (right).

9. Let $x_k \in \mathbb{R}^n$ be the current iterate and the gradient $\nabla f(x_k)$ be the direction in which we move from this point. A line-search algorithm would pick the step $\alpha_k \in \mathbb{R}$ for the next iterate

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

to have a cost as small as possible. Thus we aim at minimizing the function

$$g \colon \alpha \mapsto f(x_k - \alpha \nabla f(x_k)).$$

For all $\alpha \in \mathbb{R}$ we have

$$g(\alpha) = f(x) + \frac{\alpha^2}{2}\nabla f(x_k)^\top A \nabla f(x_k) - \alpha(x_k^\top A + b^\top)\nabla f(x_k)$$
$$= f(x) + \frac{\alpha^2}{2}\nabla f(x_k)^\top A \nabla f(x_k) - \alpha\|\nabla f(x_k)\|^2.$$

This is a second order polynomial which is minimized when

$$\alpha = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^\top A \nabla f(x_k)}.$$

The computation of this optimal step-size is cheap and speeds up convergence by a lot. In particular, whenever the gradient points towards the direction of the global minimum the algorithm is done in a single step.

■

2. The 2D Rosenbrock function. The Rosenbrock function (https://en.wikipedia.org/wiki/Rosenbrock_function) is a classical benchmark for testing optimization algorithms. Its original definition is the bivariate function given by

$$f(x, y) = (a - x)^2 + b(y - x^2)^2, \quad \text{with } a, b > 0.$$

1. Show that the Rosenbrock function has a unique global minimum $(x^*, y^*) = (a, a^2)$.

Restrict now to the case $a = 1$, $b = 100$. The minimizer is $(x^*, y^*) = (1, 1)$.

2. Compute the gradient of f.

3. Implement a fixed step-size gradient descent algorithm. Stopping criteria should include a maximum number of iterations and a tolerance on the gradient norm.

4. Argue that ∇f is not Lipschitz continuous.

Gradient descent does not have global convergence guarantees with a fixed step-size because ∇f is not Lipschitz continuous. However, the gradient is Lipschitz continuous in a compact neighborhood of the global minimum. So we expect the algorithm to converge to the minimum if we start sufficiently close, provided that the step-sizes are small enough. Consider for now the initial point $(x_0, y_0) = (1.2, 1.2)$.

5. Assess the first few iterations of your algorithm with step-size $\alpha = 10^{-2}$. Does it appear to be a good step-size?

6. The step-size $\alpha = 10^{-3}$ should work better. Run your algorithm for 10^5 iterations and plot the gradient norms. How close to the optimum do you get? Is starting closer to the optimum significantly improving the convergence speed? Try for example with $(x_0, y_0) = (-1.2, 1)$.

The convergence of gradient descent with fixed step-sizes is very slow for this problem. This is coming from the properties of f around the minimizer.

7. Compute the Hessian of f, that we denote by $\nabla^2 f$. Compute the eigenvalues of $\nabla^2 f$ at the minimizer (you can use the function `eig` in MATLAB). Can you diagnose the problem of gradient descent for this optimization problem?

8. Implement gradient descent with backtracking line-search (Algorithm 3.1 in Nocedal and Wright). Run it on the instances in Question 6 with $\bar{\alpha} = 1$, $\rho = 0.5$, $c = 10^{-4}$. Is adaptive step-sizing more efficient?

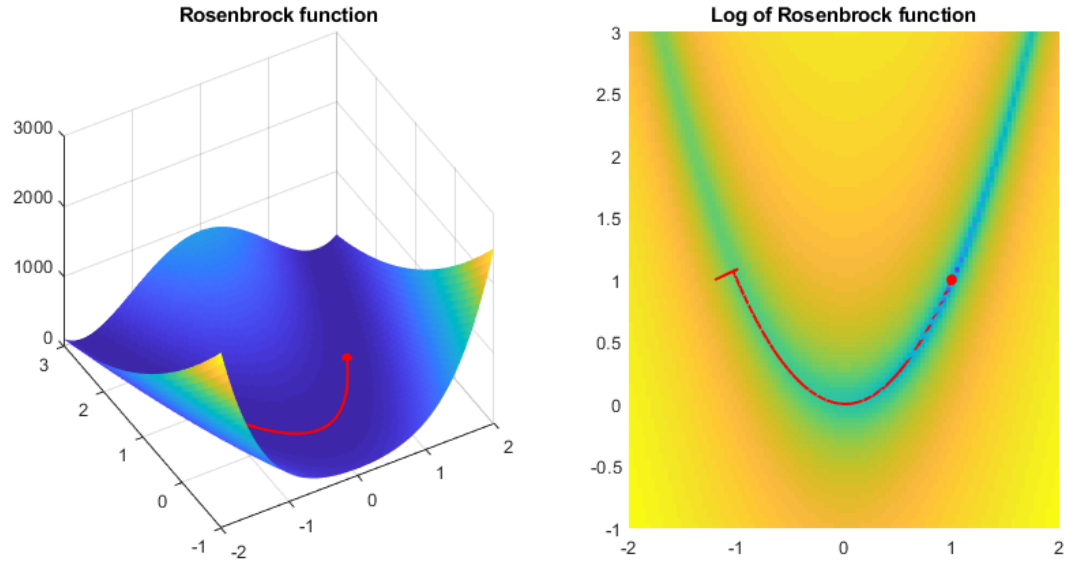

Figure 4: Surface plot of the Rosenbrock function and contour plot of the log of the Rosenbrock function (to highlight variations between small numbers). See in red the optimization path of GD with fixed step-size $\alpha = 10^{-3}$ started at $(x_0, y_0) = (-1.2, 1)$.

Answer.

1. The function f is nonnegative as a sum of squares. It is zero if and only if both terms are zero. We deduce that $f(x, y) = 0$ if and only if $x = a$ and $y = a^2$. This implies that there is a unique global minimum: (a, a^2).

2. The function is smooth and two dimensional so it is convenient to compute partial derivatives. We find that for all x, y we have

$$\nabla f(x, y) = \begin{bmatrix} 2(x - a) + 4bx(x^2 - y) \\ 2b(y - x^2) \end{bmatrix}.$$

   Notice that f has a unique critical point, which is the global minimum (a, a^2).

3. See lecture notes for pseudocode.

4. For all x we find that

$$\nabla f(x, 0) = \begin{bmatrix} 2(x - a) + 4bx^3 \\ -2bx^2. \end{bmatrix}$$

   and

$$\|\nabla f(x, 0) - \nabla f(0, 0)\| \geq 2bx^2.$$

   There is no constant L such that this quantity is bounded by $L|x|$ so we conclude that ∇f is not Lipschitz continuous. Alternatively, we could compute the Hessian and show that it is not upper-bounded.

5. We run gradient descent with step-size $\alpha = 10^{-2}$ starting from $(x_0, y_0) = (1.2, 1.2)$. We observe that the iterates diverge to infinity. After 8 iterations all entries of the iterates are Infs and the gradient norm is NaN. This poor behavior comes from the fact that

α is too large. The iterates generated by the algorithm jump over locations where the function value and gradient norm become larger and larger. As the gradients are not Lipschitz continuous there is no ideal step-size. However the gradient is always Lipschitz continuous in compact subsets of the search space. In particular it is Lipschitz continuous in compact neighborhoods of the global minimum. If we start close enough, we expect the method to converge if we choose the step-sizes carefully.

6. With $\alpha = 10^{-3}$ the iterates of gradient descent seem to converge to the global minimizer. However, starting from $(x_0, y_0) = (1.2, 1.2)$ we need approximately $3 \cdot 10^4$ iterations to find a point where the gradient norm is less than 10^{-6}. If we start closer with $(x_0, y_0) = (-1.2, 1)$ we save only a few thousands of iterations for the same gradient norm tolerance. The convergence is overall very slow, even with a good initial guess.

7. We compute the second order partial derivatives and find that for all x, y we have

$$\nabla^2 f(x, y) = \begin{bmatrix} 12bx^2 - 4by + 2 & -4bx \\ -4bx & 2b \end{bmatrix}.$$

With the parameters $a = 1$ and $b = 100$ we compute the eigenvalues of this matrix at the global minimizer using the function `eig`. We find

$$\begin{cases} \lambda_1 \simeq 1001.601 \\ \lambda_2 \simeq 0.399 \end{cases} \qquad \text{so} \qquad \frac{\lambda_1}{\lambda_2} = \kappa\left(\nabla^2 f(a, a^2)\right) \simeq 2508.$$

In Problem 1 we defined the condition number as the ratio between the largest and the smallest eigenvalue of the Hessian. This number plays an important role in the local convergence behavior of gradient descent. In particular, it indicates how elongated the isolines are around the minimizer. This confirms our intiution that the optimum lies at the bottow of a very narrow and steep valley (see Figure 4). In this situation, the strategy of taking the steepest descent as an update direction will cause the iterates to oscillates from one side of the valley to the other, making very little progress. Later in the course, we shall see how second-order algorithms, like the Newton method and the trust regions method, drastically improve the local convergence behavior in situations of bad conditioning.

8. See Algorithm 3.1 in Nocedal and Wright for pseudocode. We use the standard parameters $\bar{\alpha} = 1$, $\rho = 0.5$, $c = 10^{-4}$ (you may experiment with other parameters). We run this algorithm with the initial points proposed in Question 6 and observe convergence to the global optimum. The number of iterations to reach the same gradient norm (or function value) is approximately reduced by a factor 3 (see Figure 5). However the convergence is still linear and overall slow because of the large condition number.

■

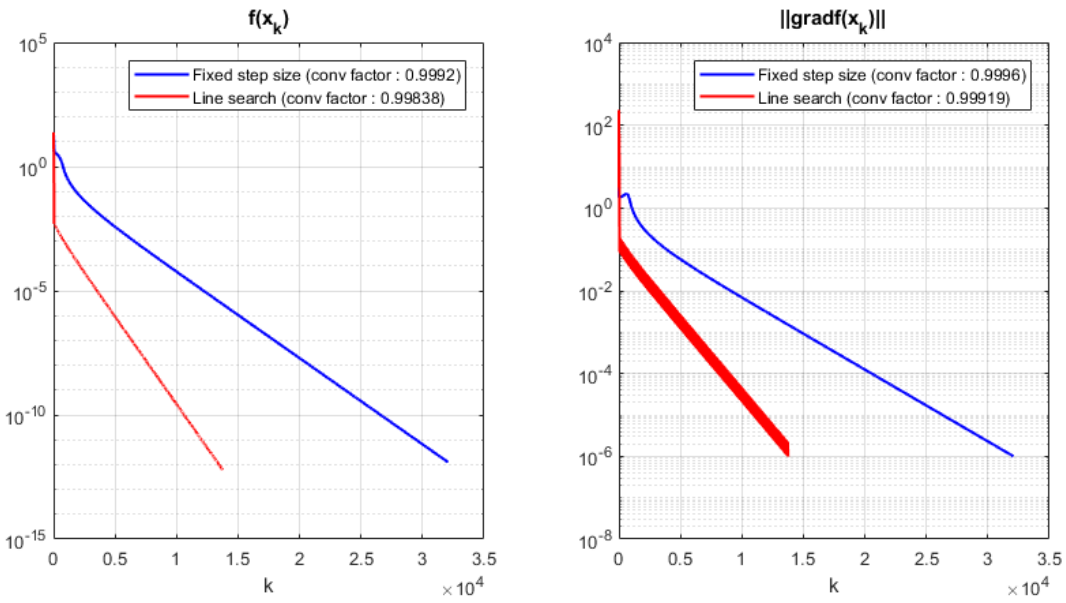

Figure 5: Function values and gradient norms of the iterations of gradient descent with fixed step size $\alpha = 10^{-3}$ (blue) and backtracking line search (red). Initial point: $(x_0, y_0) = (-1.2, 1)$.