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?

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?