
Technical University of Crete
School of Electrical and Computer Engineering

Course: Convex Optimization

Instructor: Athanasios P. Liavas

Exercise 2 (110/500)

Report Delivery Date: November 16, 2021

In this exercise, we will solve simple unconstrained convex optimization problems.

A. We consider a simple quadratic optimization problem.

- (a) (10) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$. For fixed $\mathbf{x} \in \mathbb{R}^n$ and $m > 0$, let $g_{\mathbf{x}} : \mathbb{R}^n \rightarrow \mathbb{R}$ be defined as

$$g_{\mathbf{x}}(\mathbf{y}) := f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{m}{2} \|\mathbf{y} - \mathbf{x}\|_2^2. \quad (1)$$

- i. (5) Compute $\nabla g_{\mathbf{x}}(\mathbf{y})$.
- ii. (5) Compute the optimal point, $\mathbf{y}_* = \underset{\mathbf{y}}{\operatorname{argmin}} g_{\mathbf{x}}(\mathbf{y})$, and the optimal value, $g_{\mathbf{x}}(\mathbf{y}_*)$.

B. We proceed to the solution of convex quadratic problems. Our goal is to study the behavior of the gradient method, with **exact** and **backtracking** line search.

- (a) (40) Consider the problem

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{q}^T \mathbf{x}, \quad (2)$$

where $\mathbf{P} \in \mathbb{R}^{n \times n}$, $\mathbf{P} = \mathbf{P}^T \succ \mathbf{O}$ and $\mathbf{q} \in \mathbb{R}^n$ (indicative values of n are $n = 2, 50, 10^2, 10^3$.)

- i. A random positive definite matrix \mathbf{P} can be constructed in many ways. To fully control the condition number of the matrix, we can act as follows. Every positive definite \mathbf{P} can be expressed as

$$\mathbf{P} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \quad (3)$$

with $\mathbf{U}, \mathbf{\Lambda} \in \mathbb{R}^{n \times n}$, $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_n$,¹ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$, with $\lambda_i > 0$, for $i = 1, \dots, n$. The columns of \mathbf{U} are the eigenvectors of \mathbf{P} and the elements of the diagonal of $\mathbf{\Lambda}$ are the eigenvalues of \mathbf{P} .

- (a) To create a random orthonormal \mathbf{U} , create a random $(n \times n)$ matrix \mathbf{A} and then calculate its singular value decomposition

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{A});$$

Compute $\mathbf{U} * \mathbf{U}'$ and $\mathbf{U}' * \mathbf{U}$. What do you observe?

- (b) To construct the eigenvalues λ_i , for $i = 1, \dots, n$, select the smallest and largest, λ_{\min} and λ_{\max} , respectively. You can create the other eigenvalues in any way you wish. For example, you can create $(n - 2)$ random numbers, uniformly distributed in the interval $[\lambda_{\min}, \lambda_{\max}]$ using the command

$$\mathbf{z} = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) * \text{rand}(n - 2, 1); \quad (4)$$

and the vector of the n eigenvalues as

$$\text{eig_P} = [\lambda_{\min}; \lambda_{\max}; \mathbf{z}]; \quad (5)$$

Matrix $\mathbf{\Lambda}$ may be constructed as

$$\mathbf{\Lambda} = \text{diag}(\text{eig_P}); \quad (6)$$

The condition number of the problem is $\mathcal{K} := \frac{\lambda_{\max}}{\lambda_{\min}}$.

- ii. (5) Construct random vector \mathbf{q} and random positive definite matrix \mathbf{P} , with condition number \mathcal{K} (indicative values, $\mathcal{K} = 10, 10^2, 10^3$).
- iii. (5) Solve problem (2) using the closed-form solution, and compute the optimal point, \mathbf{x}_* , and the optimal value, $p_* = f(\mathbf{x}_*)$.
- iv. (15) Solve problem (2) using the gradient algorithm (with exact and backtracking line search). How many iterations are necessary for convergence, if $\mathcal{K} = 1$ and you use the exact line search?
- v. (5) For $n = 2$, construct a contour plot, with levels the values $f(\mathbf{x}_k)$. Then, on top of this plot, plot the trajectories of $\{\mathbf{x}_k\}$ produced by the two algorithms. (For problems with large condition number, you must observe the “zig-zag” effect.)

¹Matrices that satisfy this property are called **orthonormal**.

- vi. (5) Plot quantity $\log(f(\mathbf{x}_k) - p_*)$, produced by the two algorithms, versus k . What is the slope of this plot? What do you observe for different condition numbers?
- vii. (5) Using the convergence analysis results for strongly convex functions, and the values of $f(\mathbf{x}_0)$, p_* and ϵ , compute the minimum number of iterations that guarantees solution within accuracy ϵ , i.e., $f(\mathbf{x}_k) - p_* \leq \epsilon$, and compare it with the number of iterations performed by the algorithms. What do you observe?

C. We proceed to the solution of a convex unconstrained problem.

(60) Let $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$, with rows \mathbf{a}_i^T , for $i = 1, \dots, m$. Consider the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, defined as

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} - \sum_{i=1}^m \log(b_i - \mathbf{a}_i^T \mathbf{x}) = \mathbf{c}^T \mathbf{x} - \text{sum}(\log(\mathbf{b} - \mathbf{A}\mathbf{x})), \quad (7)$$

with $\mathbf{x} \in \mathbb{R}^n$ and $m > n$ (or $m \gg n$) (indicative value pairs $(n, m) = (2, 20), (50, 200)$. If you have patience and enough memory in the computer, you can set $(n, m) = (300, 800)$ or larger values).

Some observations are as follows (see Boyd–Vandenberghe, pages 141, 419–422, 458–459, 472, 492):

- The set $\text{dom} f$ contains only the points $\mathbf{x} \in \mathbb{R}^n$ for which the arguments of the logarithms are positive.

Prove that

- (a) (5) the set $\text{dom} f$ is convex.
- (b) (5) Function f is convex.
- Observe that if $b_i > 0$, for $i = 1, \dots, m$, then $\text{dom} f \neq \emptyset$, because $\mathbf{x} = \mathbf{0}$ is a feasible point (in the experiments, it makes sense to always use this convention).
- Function f may be unbounded from below. In this case, the problem has no solution (no need to do something for it - `cvx` will help you to identify these cases).
- If a solution \mathbf{x}_* exists, then it lies in the interior of $\text{dom} f$, because when we approach the boundary of $\text{dom} f$ the value of the function increases without

bound. This means that a necessary and sufficient condition for \mathbf{x}_* to be an optimal point is $\nabla f(\mathbf{x}_*) = \mathbf{0}$.

Our study will proceed as follows.

- (a) Minimize f using the `cvx`. If the problem has a solution, then `cvx` will compute it, otherwise it will display a message saying that the problem has no solution.
- (b) (5) If $n = 2$, then plot f and its level sets in the neighborhood of the optimum point. You can check whether a point \mathbf{x} belongs to the domain of f by checking the argument of the logarithm at this point. If a point \mathbf{x} belongs to $\mathbf{dom}f$, then you can compute the value of f at this point. Otherwise, you can give an arbitrarily large value to f at this point (for example, $f(\mathbf{x}) = 10^3$).
- (c) (20) Assuming that $\mathbf{x} = \mathbf{0}$ is a feasible point, minimize f using the gradient algorithm with backtracking line search, starting from $\mathbf{x}_0 = \mathbf{0}$. The implementation will have the following main difference from the baseline implementation.
 - In step $(k + 1)$, given the vectors \mathbf{x}_k and $\Delta\mathbf{x}_k$, and having set $t = 1$, before starting the backtracking, you should check whether the point $\mathbf{x}_k + t\Delta\mathbf{x}_k$ belongs to $\mathbf{dom}f$ or not. If it does not belong to $\mathbf{dom}f$, then you must put $t := \beta t$ (β is the backtracking parameter) and repeat this process until you find a point that belongs to $\mathbf{dom}f$.² When you arrive at a point that belongs to $\mathbf{dom}f$, then you can proceed to the typical backtracking line search.
- (d) (20) Following the analogous procedure, minimize function f using the Newton algorithm.
- (e) (5) Plot, using `semilogy`, quantities $(f_{\text{gradient}}(\mathbf{x}_k) - p_*)$ and $(f_{\text{newton}}(\mathbf{x}_k) - p_*)$, as a function of step k . What do you observe for small and large values of the pair (n, m) ?

²It may help you in the understanding of the behavior of the method if you print a message every time point $\mathbf{x} + t\Delta\mathbf{x}$ does not belong in $\mathbf{dom}f$. You may observe different behavior far away and close to the solution.