Mid-term Exam 2015 – Part 2

Introduction to Optimization lecture
at Université Paris-Saclay

Anne Auger and Dimo Brockhoff

firstname.lastname@inria.fr

October/November, 2015

Abstract

This document details the exercises for the mid-term exam of the Introduction to Optimization lecture at Université Paris Saclay of 2015. The mid-term exam is published in two parts, one on discrete optimization and one on continuous optimization. This second part is published on October 30, 2015 and needs to be handed in two weeks later on **Friday, November 13 (before 11:59pm, Paris time)**.

In order to pass the final exam, 50% of the points of this mid-term exam need to be reached. It furthermore counts 1/3 of the final grade of the lecture.

Please hand in your solutions and in particular the source code by sending an e-mail to one of the lecturers (e-mail above). Use the keyword "mid-term exam" in the subject of the e-mail and do not forget to mention your full name.

1 Line Search in Descent Algorithms (30 Points)

The purpose of this exercice is to understand the interest of implementing a line search procedure in descent algorithms. Algorithm 1 (in Section 3) reminds the general scheme for a descent algorithm. It calls for a line-search procedure.

We start by considering a **gradient descent algorithm** where at each iteration, the descent direction corresponds to minus the gradient of the objective function $f : \mathbb{R}^n \to \mathbb{R}$ to be minimized, that is

$$\mathbf{d}_k = -\nabla f(\mathbf{x}_k) \ .$$

The result of the line-search procedure is called the step-size and is denoted as σ_k, that is

$$\sigma_k = \text{LineSearch}(\mathbf{x}_k, \mathbf{d}_k)$$

such that the update of the current solution $\mathbf{x}_k$ reads

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma_k \mathbf{d}_k \ .$$

We consider first a degenerated line-search procedure that consists in taking a constant step-size equal to σ. We will test the **gradient descent algorithm** with **constant step-size** on the functions

$$f_\alpha(\mathbf{x}) = \alpha \sum_{i=1}^{n} \mathbf{x}_i^2 \ , \alpha > 0 \ .$$

1. What is the optimum of f_α?

2. Compute the gradient of f_α.

3. Implement a function falpha that takes as input a vector $\mathbf{x}$ of $\mathbb{R}^n$, $\alpha \in \mathbb{R}$ and outputs $f_\alpha(\mathbf{x})$; a function gradientFalpha that takes as input $\mathbf{x} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$ and outputs the gradient of f_α.

4. Implement the gradient descent algorithm with fixed step-size. Save the sequence $\mathbf{x}_k$. Implement as stopping criteria: a maximum of iterations equal to 10^6 and $\|\nabla f(\mathbf{x}_k)\| \leq 10^{-12}$. We advise to write a function that takes as input the objective function, the gradient function, the initial search point, and the step-size σ and returns the sequence $\mathbf{x}_k$ (and implicitly the number of iterations to reach the stopping criterion).

5. Consider $\alpha = 1/2$ and $\sigma = 0.1$. For $n = 2$, plot the trajectory of the algorithm, that is, plot the evolution of the vectors $\mathbf{x}_k$ in the 2D-plane. We will consider two runs, the first one initialized at $(10, 10)$ and the second one at $(-5, 10)$. Comment what you observe and explain.

6. For $n = 10$, $\sigma = 0.1$, consider the functions f_α for $\alpha = 1/2$, $\alpha = 1/20$ and $\alpha = 1/200$. Initialize the algorithm at $\mathbf{x}_0 = (10, \ldots, 10)$.

Report the number of iterations needed to reach the stopping criterion of $\|\nabla f(\mathbf{x}_k)\| \leq 10^{-12}$. Perform the same experiments for $\sigma = 0.01$. Comment the results.

7. Explain the result theoretically. You can start by investigating what is the optimal step-size for the function f_α.

We will now compare the result of the gradient descent algorithm with fixed step-size and with the Armijo rule.

8. Implement the Armijo line search procedure. The Armijo rule is reminded in Algorithm 2 (in Section 3). Take $\beta = \theta = 1/2$. For the implementation, we suggest to return the found step-size σ and the number of calls to the function f.

9. Implement the gradient descent algorithm with Armijo rule as line search procedure.

10. Using the same settings as in Question 6, report the number of gradient calls and function calls needed to reach a gradient with norm smaller than 10^{-12}. Compare to the gradient descent with fixed step-size, conclude.

2 Gradient versus Newton direction in descent algorithms (20 Points)

We now consider the function

$$f_\alpha^{\text{elli}} = \frac{1}{2}\sum_{i=1}^{n} 10^{\alpha(\frac{i-1}{n-1})}\mathbf{x}_i^2 \ .$$

11. Compute the gradient and Hessian matrix of f_α^{elli}.

12. Compute the condition number of the Hessian matrix of f_α^{elli} (We remind that the condition number of a matrix corresponds to the ratio between the largest and smallest eigenvalue).

13. Implement the descent algorithm with the Newton direction as descent direction, that is

$$\mathbf{d}_k = -\text{Hess(f)}^{-1}\nabla f \ .$$

We will use the Armijo rule as line-search procedure.

14. Report for f^{elli}_α, $\alpha \in \{1,2,3\}$, dimension $n = 10$, initial search point $x_0 = (10,\dots,10)$and initial stepsize of $\sigma = 10$ the number of gradient calls and the number of function calls to reach $\|\nabla f\| \leq 10^{-12}$ for the descent algorithm with gradient and Newton as descent directions. Explain the results.

3 Algorithms

Algorithm 1: General framework for a descent algorithm to optimize $f : \mathbb{R}^n \to \mathbb{R}$. The descent direction and the LineSearch procedure depend on f.

```
Initialize x_0 ∈ R^n, k = 0
while stopping criteria not met do
    compute descent direction d_k
    x_{k+1} = x_k + LineSearch(x_k, d_k) d_k
    k = k + 1
end
```

Algorithm 2: Armijo rule

Input: descent direction $\mathbf{d}$, point $\mathbf{x}$, objective function $f(\mathbf{x})$ and its gradient $\nabla f(\mathbf{x})$, parameters $\sigma_0 = 10$, $\theta \in (0, 0.5)$ and $\beta \in (0,1)$
Output: step-size σ

```
Initialize σ: σ ← σ_0
while f(x + σd) > f(x) + θσ∇f(x)^T d do
    σ ← βσ
end
```