LINMA2471 : Optimization models and models : course 6 (28/10/2015)

Renaud Dufays, Antoine Durviaux & Leïla Van Keirsbilck

Octobre 2015

1 Reminders

Definition 1. *Let $\mathcal{C}$ be a closed convex set and f a differentiable function. We consider the constraint problem $\min_{x\in\mathcal{C}} f(x)$. A **stationary point** is a point x^* such that*

$$< \nabla f(x^*), x - x^* > \geq 0 \quad \forall x \in \mathcal{C}$$

Intuitively, that means that all possible errors $x - x^*$ are in opposite direction with $-\nabla f$. A schema of the situation is shown on figure 1.

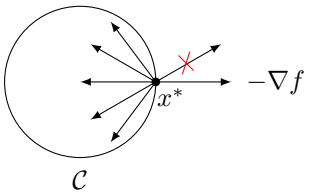

Figure 1: Example of a stationnary point x^*.

Example 1. $\mathcal{C} = \{x \in \mathbb{R}^n : x_i \geq 0\}$ (nonnegative orthant)

$$x^* \text{ stationnary iff } \sum_i [\nabla f(x^*)]_i \, [x_i - x_i^*] \geq 0 \quad \forall x \geq 0$$

$$x^* \text{ stationnary iff either } [\nabla f(x^*)]_i = 0$$
$$\text{or } [\nabla f(x^*)]_i > 0 \text{ and } x_i^* = 0 \; \forall \, i$$

Indeed, as $\sum_i [x_i - x_i^*] \geq 0$ for all $x \geq 0$, $[\nabla f(x^*)]_i$ must be ≥ 0. If not, we could choose a very large x_i for this component and have a negative sum.

Example 2. It also works easily for $\mathcal{C} = \{x | \sum_i x_i = 1\}$ which is a kind of budget constraint. In that case, it is possible to show that $[\nabla f(x^*)]_i = \lambda \; \forall i$. That means that all the gradient components are equal to each other, or economically speaking that the marginal costs are equal to each other. At the optimum, the marginal cost is equal for each component, it does not matter which one you lower.

Exampe 3. (not treated) The Eucledian bowl.

Projected gradient method ($f \in C_L^{1,1}$)

Given x_0, L, $k = 0$	
Repeat	$x_{k+1} = P_C[x_k - \frac{1}{L}\nabla f(x_k)]$
	$k \leftarrow k + 1$

2 Gradient mapping for constrained problems

Definition 2. *For some $M > 0$, the* ***gradient mapping*** $\mathbf{G_M^C f(x)}$ *is the unique vector satisfying*

$$x - \frac{1}{M} G_M^C f(x) = P_C\left[x - \frac{1}{M}\nabla f\right]$$

The role of the gradient mapping is similar to the one of the gradient in the non-constraint case. Notice that if we take $\mathcal{C} = \mathbb{R}^n$, then $G_M^{\mathbb{R}^n} f(x) = \nabla f(x)$. An illustration of the gradient mapping is given in figure 2

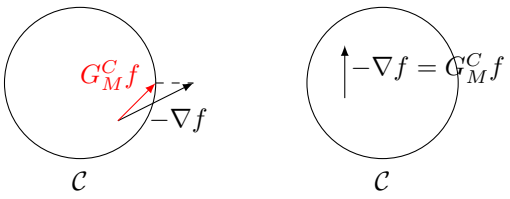

Figure 2: Illustration of gradient mapping.

The gradient method becomes, in the case of constraint problems :

Given x_0, L, $k = 0$	
Repeat	$x_{k+1} = P_C[x_k - \frac{1}{L} G_M^C f(x_k)]$
	$k \leftarrow k + 1$

Property 1. *For any $M > 0$, we have x^* stationary iff $G_M^C f(x^*) = 0$*

Property 2. *Given $f \in C_L^{1,1}$, and letting $x^+ = x - \frac{1}{L} G_L^C f(x)$ be the next step, we have*

$$f(x) - f(x^+) \geq \frac{\left\|G_L^C f(x)\right\|^2}{2L}$$

Theorem 1. *Using those properties, we obtain that for $f \in C_L^{1,1}$, the projected gradient method gives*

$$\min_{0 \leq i \leq N} \left\|G_L^C f(x_i)\right\| \leq \sqrt{\frac{2(f(x_0) - f(x^*))}{L(N+1)}}$$

We have a stronger result in the case of a convex f, as stated by the following theorem.

Theorem 2. *Let $f \in F_L^{1,1}$. For any iterate x_N and any x,*

$$f(x_N) - f(x^*) \leq \frac{M \left\|x_0 - x^*\right\|^2}{2N}$$

The projected gradient method is quite slow because it is in $\mathcal{O}(\frac{1}{N})$ but also because projection can be complicated. Indeed, projection can be hard to compute if $\mathcal{C}$ is too complex.

Example 3. If $\mathcal{C} = \{x \geq 0\}$ this is easy because $[P_C(x)]_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{if } x_i < 0 \end{cases}$

Example 4. If $\mathcal{C} = \{x | Ax = b\}$ is a subspace, this is expensieve. We have indeed to solve a linear system, which is $\mathcal{O}(n^3)$.

Example 5. If $\mathcal{C} = \{x | Ax \leq b\}$ is a polyhedron, this is even more expensive : the projection problem has to be written as a minimization of the distance (quadratic programming). We could also use a theorem that separates the problem on each facet of the polytope.

3 Acceleration gradient [Nesterov 1983]

Given $f \in C_L^{1,1}$, x_0, L, $k = 0$ and $x_{-1} = x_0$	
Repeat	$y_k = x_k + \beta_k(x_k - x_{k-1})$
	$x_{k+1} = P_C[y_k - \frac{1}{L}\nabla f(y_k)]$
	$k \leftarrow k + 1$

The first step is called an extrapolation step whereas the second is called a gradient step. Note that y_k is not always in $\mathcal{C}$. The idea is to recycle the previous work because $(x_k - x_{k-1})$ will be close to the gradient.

Theorem 3. *For $\beta_k = \frac{k-1}{k+2}$, we have*

$$f(x_N) - f(x^*) \leq \frac{2L \|x_0 - x^*\|^2}{(N+1)^2}$$

It can be proved that this rate of $\frac{1}{N^2}$ is not improvable. Accelerated gradient is thus a sublinear method (slower than linear). While the gradient method is really robust, the accelerated method is extremely sensitive. Note that β_k goes to 1 as k goes to infinity. An example is the Huber function : on the linear part, the steps increase in a quadratic way.

Linear convergence (to zero) : $1, \rho, \rho^2, \rho^3, \rho^4, \rho^5, ...$ with $\rho < 1$. If we restrict the class of functions to **strongly convex**, we can get linear convergence.

Definition 3. *Given $\mu > 0$, f is μ-**strongly convex** iff*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - \frac{\mu}{2}\lambda(1-\lambda)\|x - y\|^2$$

A strongly-convex function can be viewed as a function that has no flat part.

Property 3. *If $f \in C^2$ then f is μ-strongly convex iff for all x*

$$\lambda_{min}(\nabla^2 f(x)) \geq \mu$$

Lemma 1. *For any strongly convex function, we have*

$$\|x - x^*\|^2 \leq \frac{2}{\mu}[f(x) - f(x^*)]$$

That means that if I decrease the error on the function value, automatically I decrease the distance to the solution. Using previous results, we obtain :

$$\|x_N - x^*\|^2 \leq \frac{2}{\mu}\frac{L\|x_0 - x^*\|^2}{2N} = \frac{L}{\mu}\frac{\|x_0 - x^*\|^2}{N}$$

$\frac{L}{\mu}$ is called the condition number.

Remark 1. *The accelerated gradient method gives better results if we restart the method after a certain number of iteration. For example, if the squared norm of the error is divided by 5 after $N = 10$ iterations, it will be divided by 10 after $N = 20$ iterations. But if we restart the method after 10 iterations and make 10 new iterations, the square norm of the error will be divided by 25 although we made a total of 20 iterations in both cases. A result that is not proven here is that the optimal number of iterations after which the method should be restarted is $\frac{L}{\mu}e$. In this case, the method is linearly convergent and we have :*

$$\|x_N - x^*\|^2 \leq \left(1 - \frac{1}{e\frac{L}{\mu}}\right)\|x_0 - x^*\|^2$$