Modern Optimization

Jean-Luc Bouchot

School of Mathematics and Statistics
Beijing Institute of Technology
jlbouchot@bit.edu.cn

Spring 2021

Outline

Definition 2.1

A **local numerical optimization** *algorithm is an iterative algorithm where*

$$x^{k+1} = x^k + \alpha_k d_k$$

assuming a starting point x^0 is provided.
The algorithm is characterized by

- *A choice of direction d_k at each iteration.*
- *A choice of step size α_k at each iteration.*

Example 2.1

We have in our mathematical journey already seen some iterative local optimization algorithms:

- Gradient descent: assumes the objective function of an unconstrained problem is differentiable and choose the steepest descent direction: $d_k = -\nabla f_0(x^k)$.
- Newton-like algorithms: assumes a twice differentiable function and pick $d_k = -\nabla^2 f_0(x^k)^{-1} \nabla f_0(x^k)$.
- Quasi-Newton type: approximate the (inverse) Hessian, pick $d_k = -B_k \nabla f_0(x_k)$ where $B_k \approx \nabla^2 f_0(x_k)^{-1}$ (SR1 and BFGS are great candidates)

Example 2.2

They are various ways of selecting the step size

- Constant step – Works in the convex settings, if you know a lot about your function. It should be avoided in most cases
- α_k satisfies the Goldstein conditions – We'll talk about it later. Roughly speaking, it makes sure that the next step decreases the objective value sufficiently.
- α_k satisfies the (weak/strong) Wolfe conditions – We'll talk about it later. Roughly speaking, it makes sure that we decrease the function sufficiently, and that decrease at the next point is not as big as at the previous.
- Backtracking α_k: go somewhat far from x^k and reduce slightly the step size until enough decrease is noticed.

We will give more details to why these strategies work fine in later chapters.

Remark 2.1

This is not the whole story and we will scratch only parts of the problem:

- Line search methods: define a search direction and find a good step size along this direction
- Trust region methods: define a search region and find a good direction within this region.
- One may accelerate the updates ...

Definition 2.2 (Globally convergent algorithms)

An algorithm is said to be **globally convergent** *if*

$$\|\nabla f_0(x^k)\| \to 0.$$

Example 2.3

Note that globally convergence only means convergence to a stationary point. As a counter example think of
$$x \mapsto x^3.$$

Proposition 2.1

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. The steepest decrease from a point x^k is done in the direction of the negative gradient.

Definition 2.3 (Vanilla gradient descent: the convex case)

The vanilla gradient descent is characterized by the following iterations

$$x^{k+1} = x^k - \gamma \nabla f_0(x^k),$$

for a constant $\gamma > 0$.
Note: From now on, we will write ∇f_k *or even* g_k *for the gradient evaluated at point* x^k.

Exercise 2.1

Some care should be taken though. Consider the univariate function $f(x) = \frac{1}{2}x^2$ and the step size $\gamma = 2$.
Show that for any given starting point x^0, the vanilla gradient descent will not converge to the optimal point.
This shows that some care should be taken when using the gradient descent method.

Proposition 2.2

Let f_0 *be a convex function. Then, using a fixed stepsize* $\gamma > 0$ *and starting at any initial point* x^0 *will yield an error averaged over* K *steps in the sequence of iterates fulfilling*

$$\sum_{k=0}^{K} \Big(f(x^k) - f(x^*) \Big) \leq \frac{\gamma}{2} \sum_{k=0}^{K} \|g_k\|^2 + \frac{1}{2\gamma} \|x^0 - x^*\|^2.$$

Remark 2.2

It is important to remark:

- We cannot hope much more than this. All we have used here is convexity and differentiability.
- The dependence on $\|x^0 - x^*\|$ makes sense: the further away you start the longer you'll have to work.
- This gradient isn't quite the nicest thing ever.