METHODS OF NONLINEAR OPTIMIZATION: HW#5

Gauss was born 50 years after Newton died, but I have a dream in which the young Newton meets Gauss, and both of them are perfectly fluent in the language of linear algebra and optimization we speak today. Newton is in his 20's and has not yet discovered the soon-to-be earthshaking $F = ma$ law. But he has access to the data compiled by a bunch of top experimental physicists for objects dropping from the top of Torre di Pisa, in the form of

$$(t_i = \text{time}, \;\; d_i = \text{distance}), \quad i = 1, \dots, n.$$

Newton would later see that the $F = ma$ law, together with $\int$, would yield $d(t) = -1/2gt^2 + v_0 t + s_0$, where g is the gravitational acceleration rate, v_0 the initial velocity and s_0 the initial position. For now, all he sees from the data is that a linear function would not fit the data well, and out of the blue he decides to use a quadratic function to 'explain' the data. That is, he seeks to find a quadratic function $f(t) = at^2 + bt + c$ such that

$$f(t_i) = d_i, \quad i = 1, \dots, n. \tag{0.1}$$

Note that this is an over-determined **linear** system: we are trying to find $a, b, c \in \mathbb{R}$ such that

$$\underbrace{\begin{bmatrix} t_1^2 & t_1 & 1 \\ t_2^2 & t_2 & 1 \\ \vdots & \vdots & \vdots \\ t_n^2 & t_n & 1 \end{bmatrix}}_{:=A} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{=x} = \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}}_{:=d},$$

which clearly has no solution in general, as there are n (> 3) equations with only 3 degrees of freedom. In linear algebra 101, we simply say there is no solution, unless $d \in \text{Image}(A)$, and call it a quit. Even if d is meant to be in $\text{Image}(A)$ – as Newton's $F = ma$ law implies, a tiny bit of noise in the experimental data would take the n-dimensional vector d out of the three-dimensional subspace $\text{Image}(A)$ and make the linear system inconsistent.

In my dream, Gauss suggests to Newton that he should instead aim for a solution $x \in \mathbb{R}^3$ such that $Ax \approx d$ in the sense that it minimizes $\|Ax - d\|_2$.[1] Equivalently, they solve

$$\min_{a,b,c} \sum_{i=1}^{n} (f(t_i) - d_i)^2 = \min_x \|Ax - d\|_2^2 = \min_x \underbrace{x^T A^T A x - 2 d^T A x + d^T d}_{:=\phi(x)}. \tag{0.2}$$

This is the celebrated *least square method*.

(1) For us, the least square method is a trivial case of a convex quadratic program, one that has no constraint. The KKT condition is simply the good old stationary condition $\nabla \phi(x) = 0$, and the convexity of ϕ makes it both necessary and sufficient for optimality.

Date: May 25, 2020.

[1] This is one of the most important ideas in statistics, namely one can gain useful insights by aggregating a lot of noisy data. See S. Stigler's "The Seven Pillars of Statistical Wisdom', Chapter 1.

What is even more nice about the least square method is that the stationary condition translates to a linear system, not the over-determined linear system (0.1), but the square linear system

$$A^T A x = A^T d.$$

(10 pts) Explain why the stationary condition is equivalent to the linear system above.

Therefore, we do not need any iterative method like gradient descent to solve the optimization problem (0.2), all we need is to solve a 3×3 linear system.

(2) (20 pts) Produce a computer demonstration in Matlab for this method.

(3) Newton came back to Gauss and told him that, while the time measurements t_i are very accurate, the distance measurements d_i are not only noisy, but every now and then get extremely erroneous. Such 'outliers' make the use of 'least square' rather unsuitable, as squaring a big number makes the number even bigger. Newton, after learning the overall spirit of the least square method, suggests to change $(\cdot)^2$ to $|\cdot|$ and reformulate the problem to:

$$\min_x \sum_{i=1}^n |f(t_i) - d_i| = \min_x \underbrace{\|Ax - d\|_1}_{:=\psi(x)}. \tag{0.3}$$

The KKT theory we have, which requires differentiability of the objective function, does not apply, as $\psi(x)$ above is not differentiable!

Now an interesting idea, not due to Gauss, saves us. It turns out the least L^1 regression problem (0.3) can be recast as a linear program.

Here is the trick for getting rid of the non-differentiable absolute value function in the problem: think of (0.3) as finding a, b, c, and "as tight as possible bounds" $s_i > 0$ so that $-s_i \le f(t_i) - d_i \le s_i$ and $s_1 + \cdots s_n$ is minimized, i.e. solve

$$\begin{aligned} &\min_{a,b,c,s_1,\ldots s_n} \sum_{i=1}^n s_i \\ &\text{s.t. } s_i \ge 0,\ -s_i \le f(t_i) - d_i \le s_i,\ \ i = 1, \ldots, n. \end{aligned} \tag{0.4}$$

This is a linear program, perhaps the simplest kind of convex optimization problem!

(40pts) Convince yourself that the unconstrained, but non-smooth, optimization problem (0.3) is equivalent to the LP in (0.4). Implement this method using CVX. Produce a computer demonstration for the method. Make sure you create data with outliers. And illustrate the advantage of least L^1 over the least square in the presence of outliers.

(4) (30 pts) Create a computer demonstration for the orthogonal regression problem and the algorithm developed in Section 10.3. Use the eigen-solver `eig()` in Matlab.

I will give some guidance in a email on how to create test data for the three computer demo.

Thomas Yu, Department of Mathematics, Drexel University, 3141 Chestnut Street, 206 Korman Center, Philadelphia, PA 19104, U.S.A.

E-mail address: yut@drexel.edu