CIE 6010 / MDS 6118
Matlab Problem: L1 Regularization

Two Optimization Models

We consider recovering a desired signal $\hat{x} \in \mathbb{R}^n$ that approximately satisfies an under-determined linear system $Ax = b \in \mathbb{R}^m$, where $m < n$, with the help of a regularization function $\phi_0(Dx)$ where

$$\phi_0(y) = \|y\|_1 = \sum_i |y_i| \tag{1}$$

is sparsity promoting, and D is a finite difference matrix of either the zeroth order (i.e., identity) or the first order (or even higher order). Clearly, $\phi_0(y)$ is a non-smooth function, not differentiable whenever y has a zero element.

We first consider the optimization model

$$\min_{x \in \mathbb{R}^n} \; \phi_0(Dx), \text{ s.t. } Ax = b, \tag{2}$$

which can be converted into a linear program and be solved by the Matlab function `linprog` (as we did in Assignment 1). The under-determined linear system $Ax = b \in \mathbb{R}^m$ permits infinitely many solutions when A is of full row-rank. The regularization term $\phi_0(Dx)$ helps pick the one with the property that Dx is sparse (even though x itself may not be sparse).

The second model is an unconstrained minimization model

$$\min_{x \in \mathbb{R}^n} \; \phi_\sigma(Dx) + \frac{\mu}{2}\|Ax - b\|_2^2, \tag{3}$$

where, with a small parameter $\sigma > 0$,

$$\phi_\sigma(y) = \sum_i \sqrt{y_i^2 + \sigma} \tag{4}$$

is a smooth, differentiable approximation to $\phi_0(y)$, which reduces to $\phi_0(y)$ when $\sigma = 0$. The parameter $\mu > 0$ balances the two terms: (i) the regularization term and (ii) the data fidelity term.

The above two optimization models, and their corresponding algorithms, have their advantages and disadvantages when applied to solving different problems. In this project, different D-matrices will lead to a varying level of difficulty.

Assignment

Write two Matlab functions

```
x = myL1reg0(A, b, D);      [x,iter] = myL1reg1(A, b, D);
```

The first function uses `linprog` to solve the optimization problem (2) with input A, b and D. In the second function, you solve model (3) by the steepest descent (or gradient) method with a back-tracking line search subject to the Armijo condition. The parameters μ and σ in (3) need to be carefully chosen via experimentation (μ around 0.1 should be good and σ should be smaller). From the second iteration on, initialize back-tracking with a BB step (see the attached slides for a detailed explanation). In addition to solution x, you also output the number of iterations taken.

Further Details

Download and open the zip file `L1_Reg.zip` which contains a test script `test1_L1.m` and two p-files that are corresponding solvers from the instructor. Without your solvers, the test script will run the instructor's p-files. You are to write and put your solvers in the same folder.

Once you have written and debugged your solvers, run the test script `test1_L1.m` for both solver types 0 and 1, which will compare your solvers with the instructor's. The test script will generate plots in which the blue lines are the "true signals" and red dots represent computed solutions. Data matrices A are randomly generated and the "observed data" (or the right-hand sides) b are constructed via the formula $b = Ax^*$ where x^* represents "true signals". Your computed solutions are supposed to closely match the true signals with very small errors.

You may try random initial points, but more informed initialization is to use the solution of the least squares problem (without regularization): $\min_x \|Ax - b\|_2^2$, which gives $x_0 = A^T(AA^T)^{-1}b$.

Terminate the iteration when either

$$\|\nabla f(x^k)\| \leq 10^{-2}\|\nabla f(x^0)\|$$

or

$$|f(x^k) - f(x^{k+1})| \leq 10^{-8} f(x^k).$$

Requirements

Run the test script (updated version) using both types of solvers (LP and SD) on both types of D matrices (identity and the first-order finite difference). Run data size $m = 500$ (or larger m if your computer can effectively handle it). Submit the following items:

- A copy of your codes.
- Matlab screen printout for the above runs.
- Two output figures from your solvers (2 plots in each figure).
- A short summary (typed, 20-lines or less) about your observations on potential advantages and disadvantages of the two solvers (LP and SD) on different problems.

Extra

It will be interesting to see what happens if you disable BB steps and always initialize $\alpha = 1$ in the back tracking.