Optimization Theory and Algorithms

# Comparison of three 1st-order methods

## Optimization Model

We again consider solving the L1 regularization problem as an unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} \; f(x) \equiv \phi_\sigma(Dx) + \frac{\mu}{2}\|Ax - b\|_2^2, \tag{1}$$

where

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

and $D$ is a finite difference matrix of either the zeroth order (i.e., identity) or the first order. The parameters $\sigma > 0$ and $\mu > 0$ will be fixed.

The objective function $f(x)$ is strictly convex and Lipschitz continuously differentiable.

## Assignment

Write three Matlab functions of the form

```
[x, iter] = myL1reg2*(A, b, D);
```

where the asterisk $\star$ represents the letters $a$, $b$ or $c$. The output `x` is the computed solution, and `iter` is the number of iterations taken by the algorithm implemented in the function. In addition to the input $A, b$ and $D$, the model parameters $\sigma$ and $\mu$ are passed to your functions from the test script via global variables; that is, in each of the functions you need to declare: `global sigma mu`.

Your 3 functions should implement 3 different algorithms.

- `myL1reg2a.m` uses a BB-step back-tracking line search with the Armijo condition. It is nothing but your `myL1reg1.m` with minor modifications necessary for this project.
- `myL1reg2b.m` is the simple steepest descent method with a constant step $\alpha$ which will be obtained from the test script via the declaration: `global alpha` in this function.
- `myL1reg2c.m` implements the Nesterov's optimal 1st-order (also called acceleration) method. The same constant step $\alpha$ will be passed into this function via the declaration: `global alpha`.

## Further Details

Download and open the zip file `yzcode2.zip` which contains 3 p-files that are corresponding solvers from the instructor. Also download the test script `test2_L1.m`, which can test several run cases. For example, for `runcase = 0` the test script will run the instructor's 3 p-files without your solvers (try that first). You are to write and put your solvers in the same folder. Once you have written and debugged your solvers, run the test script `test2_L1.m` 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 whenever

$$\frac{\|\nabla f(x^k)\|}{\|\nabla f(x^0)\|} \le 10^{-2} \quad \textbf{and} \quad \frac{|f(x^k) - f(x^{k-1})|}{f(x^{k-1})} \le 10^{-8}.$$

## Requirements

Run the test script `test2_L1.m` with $runcase = 1, 2, 3, 4$ (you might change some of the $m$ values if your computer or code is too slow). Submit the following items:

- A copy of your code `myL1reg2c.m`.
- Matlab screen printout for the above 4 run cases (1 to 4).
- Output figure from your solver `myL1reg2c.m` on run case number 4.
- A short summary (typed, half a page or less) about your experiments and observations that you consider to be the most relevant and most important.