CO 672/CS 794: Optimization for Data Science
Fall 2018
Problem Set 1
S. Vavasis

Handed out: 2018-Sep-12.

Due: 2018-Sep-19 in lecture.

1. (a) Explain why the univariate function $f(x) = |x|$ is not L-smooth.

   (b) Find a function $g_L : \mathbf{R} \to \mathbf{R}$ such that g_L is L-smooth, and such that

   $$\max\{|(|x| - g_L(x))| : x \in \mathbf{R}\}$$

   tends to 0 as $L \to \infty$. [Hint: it follows from the mean value theorem that for a twice-differentiable univariate function, the smoothness constant is equal to the maximum absolute value of its second derivative.]

2. Consider a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x}$, where A is a symmetric $n \times n$ matrix. Prove that f is L-smooth, where L is equal to the operator norm of of A, $\|A\|_2$, which is defined as $\sup\{\|A\mathbf{x}\| : \|\mathbf{x}\| = 1\}$. [Hint: it follows from the definition of operator norm that $\|A\mathbf{x}\| \le \|A\|_2 \cdot \|\mathbf{x}\|$ for any A and $\mathbf{x}$. Show this first.]

3. A function $f : \mathbf{R}^n \to \mathbf{R}$ is said to be L-smooth over a set $S \subset \mathbf{R}^n$ if the inequality

   $$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|$$

   holds for all $\mathbf{x}, \mathbf{y} \in S$. Show that for the univariate function $f(x) = x^4$, there is no L such that f is L-smooth overall, but that it is L-smooth over the interval $[-1, 1]$ for some L.

4. Write a function in Matlab, Python/Numpy, Julia or R for gradient descent called `gradient_descent`. Your code should take the following input arguments:

   - A function fg invoked as `fval,gval = fg(x)` (or the equivalent in whichever language you use) that returns both $f(\mathbf{x})$ and $\nabla f(\mathbf{x})$ (the objective and its gradient),
   - An initial point $\mathbf{x}_0$,
   - Another function `eta` that is invoked as `eta(fg,x,fval,gval,i)` and returns the step-length parameter η given the objective function fg, the current iterate $\mathbf{x}$, the function value and gradient, and the iteration counter i.
   - A function `termtest` invoked as `termtest(fg,x,fval,gval,i)` that returns a boolean (i.e., true or false) and serves as the termination test. Its arguments are the same as those of `eta`.

Then apply your code to the $n = 1$ case with the objective function $f(x) = x^4$ initialized at $x_0 = 1$. Try it with three different values of the `eta` function: `eta1` that always returns the constant value $1/L$ for the value of L that you determined in Q3 for the interval $[-1, 1]$, `eta2` that returns $1/(20L)$, and `eta3` that returns $20/L$. Note: in all three cases, the `eta` function returns a hard-coded constant irrespective of its arguments. Note: you may use anonymous functions rather than fully explicit named functions as arguments to `gradient_descent`.

Hand in listings of your code(s). Run the code for 10,000 iterations with each of the three `eta` functions. What value is returned in each case? (Note: write a `termtest` function that returns "true" if the number of iterations exceeds 10,000.) Hand in a printout showing the invocations of `gradient_descent` and the results of these experiments.