Problem Set 2 for EE227C (Spring 2018): Convex Optimization and Approximation

Instructor: Moritz Hardt

Email: hardt+ee227c@berkeley.edu

Graduate Instructor: Max Simchowitz

Email: msimchow+ee227c@berkeley.edu

March 1, 2018

Problem 1: Backtracking Line Search

Let $f : \mathbb{R}^n \to \mathbb{R}$ be an m-strongly convex, M-smooth (and thus differentiable) function with global minimum x^*. Consider the following algorithm:

Initialize with an arbitrary $x_0 \in \mathbb{R}^n$, and fix parameters $\alpha \in (0, 1/2), \beta \in (0, 1)$. Then at each step $t = 1, 2, \ldots,$ do the following:

(a) Let $g_t = \nabla f(x_t)$.

(b) For $k = \{0, 1, \ldots\}$ in sequence, check if the following "sufficient decrease" condition holds:

$$f(x - tg_t) \leqslant f(x) - \alpha\beta^k \cdot \|g_t\|^2 \tag{1}$$

Assuming that this condition holds for some k (you will show this), set $\eta_t = \beta^k$.

(c) Set $x_t \leftarrow x_{t-1} - \eta_t g_t$

(A) Show that condition 1 holds for all $t \in (0, 1/M]$.

(B) Show that $\eta_t \geqslant \min\{1, \beta/M\}$. Conclude that step (b) of the above algorithm aways terminates.

(C) Using part b, show that

$$f(x_t - \eta_t g_t) \leqslant f(x) - \alpha \min\{1, \frac{\beta}{M}\} \|\nabla f(x_t)^2\| \tag{2}$$

(D) Show that there is a constant $C = C(\alpha, \beta, M, m) < 1$ such

$$f(x_t - \eta_t g_t) - f(x) \leqslant C(\alpha, \beta, M, m) \cdot (f(x_t) - f(x_t)) \tag{3}$$

Problem 2: Random Descent Directions

Let $f : \mathbb{R}^n \to \mathbb{R}$ be an m-strongly convex, M-smooth (and thus differentiable) function with global minimum x^*. Consider the following algorithm: Initialize with an arbitrary $x_0 \in \mathbb{R}^n$. Then at each step $t = 1, 2, \dots$, do the following:

(a) Choose $g_t \overset{\text{unif}}{\sim} \mathcal{S}^{n-1}$ (equivalently, g_t has the distribution of $\frac{g}{\|g\|}$, where $g \sim \mathcal{N}(0, I_n)$).

(b) Compute a step size $\eta_t := \min_{\eta \geqslant 0} f(x_{t-1} - \eta g_t)$.

(c) set $x_t \leftarrow x_{t-1} - \eta g_t$

(A) Prove that the above algorithm is a (non-strict) descent method; that is $f(x_t)$ is non-increasing in t. Also prove that unless $x_t = x_*$, $f(x_{t+1}) < f(x_t)$ with probability 1/2.

(B) Prove that there exists a numerical constant C such that, if

$$t \geqslant T(\epsilon) := Cn \cdot \frac{M}{m} \log(\frac{f(x_0) - f(x^*)}{\epsilon}) \,, \tag{4}$$

then $\text{Exp}[f(x_t) - f(x^*)] \leqslant \epsilon$.

(C) Ammend the stated algorithm to use line search instead of solving for the exactly-optimal step size. Are the rates qualitatively similar?

Problem 3: Sh*t about Quadratics

In this problem, you are going to test the sharpness of our upper and lower bounds for quadratics on a randomly generated instance. Let $\mathcal{S} = \{1, .5, .2, .1, .05\}$ and $n = 500$. Now, for each $\epsilon \in \mathcal{S}$, generate the random matrix $\mathbf{M}$ as follows:

(a)Generate an $n \times n$ random wigner matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$,

$$\mathbf{W} = \frac{1}{\sqrt{2n}}(\mathbf{X} + \mathbf{X}^\top)$$

where $\mathbf{X} \in \mathbb{R}^{n \times n}$ is a matrix with i.i.d standard normal entries.

(b) Generate $\mathbf{u}$ uniformly on the unit sphere, and set $\mathbf{M} = \mathbf{W} + (1 + \epsilon)\mathbf{u}\mathbf{u}^\top$.

(c) Now, for each $\epsilon \in \mathcal{S}$, do the following:

(A) Conduct trials $t = 1, 2, \dots, 10$.

(A.1) Generate $\mathbf{M}$ as above, and a random vector $\mathbf{v}$ uniformly on the unit sphere.

(A.2) Set $\gamma = 2\lambda_{\max}(\mathbf{W}) - \lambda_2(\mathbf{W})$, and define the matrix $\mathbf{N} = \gamma I - \mathbf{M}$. Definally, define the function $\mathbf{f}(x) = \min_x x^T \mathbf{N} x - 2\langle \mathbf{v}, x \rangle$.

(A.3) Setting $x_0 = 0$, run gradient descent, a heavy-ball method or nesterov method to solve $\min_x \mathbf{f}(x)$ for a good number of iterations (use your discretion). You may compute the eigenvalues of $\mathbf{N}$ to tune your step parameters.

(A.4) For both gradient descent and heavy-ball, record for each trial iteration s, the difference between $\mathbf{f}(x_s) - \min_x \mathbf{f}(x)$ for each iteration.

(A.5) Using the step sizes, largest/smallest eigenvalues of $\mathbf{N}$, and the initial point $x_0 = 0$, compute a worst case upper bound for $\mathbf{f}(x_s) - \min_x \mathbf{f}(x)$ for each iteration s of gradient descent and the heavy ball method.

(A.6) Run gradient descent, but this time compute the optimality gap unising "best" iterate in the Krylov space. THat is, compute

$$\min_{x \in \operatorname{span}(x_1, \ldots, x_s)} \mathbf{f}(x) - \min_x \mathbf{f}(x) \tag{5}$$

(A.7) After each trial, you should have a list of 5 values for each iterate s: an upper bound for gradient descent, the rate actually attained by gradient descent, an upper bound for heavy ball/nesterov, the rate actualy attained by heavy ball/nesterov, and the "optimal" krylov algorith,

(B) For each of the lists above, average all 10 trials and plot them on the same plot. How sharp are the upper bounds?