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 12, 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:

1 **Input:** Parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$, $x_0 \in \mathbb{R}^n$;
2 **for** $t = 0, 1, 2, \ldots$ **do**
3 $\quad g_t \leftarrow \nabla f(x_t)$;
4 $\quad$ **for** $k = 0, 1, 2, \ldots$ **do**
5 $\quad\quad$ If "sufficient decrease" condition holds

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

$\quad\quad$ set $\eta_t = \beta^k$ and **break**
6 $\quad$ **end**
7 $\quad$ Set $x_t \leftarrow x_{t-1} - \eta_t g_t$
8 **end**

Algorithm 1: Backtracking Line Search

(A) Show that condition 1 holds for whenever $\beta^k \in (0, 1/M]$.

(B) Show that $\eta_t \geqslant \min\{1, \beta/M\}$. Conclude that the loop in Line 4 aways terminates.

(C) Using part *b*, show that

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

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

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

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:

Input: Parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$, $x_0 \in \mathbb{R}^n$;
for $t = 0, 1, 2, \ldots$ **do**
  Set $g_t \overset{\text{unif}}{\sim} \mathcal{S}^{n-1}$;
  Set $\eta_t := \min_{\eta \geqslant 0} f(x_t - \eta g_t)$;
  Set $x_{t+1} \leftarrow x_t - \eta g_t$;
end

Algorithm 2: Random Direction Line Search

$\mathcal{S}^{n-1} := \{v \in \mathbb{R}^n : \|v\|_2^2 = 1\}$ denotes the unit sphere in $\mathbb{R}^n$. $g_t \overset{\text{unif}}{\sim} \mathcal{S}^{n-1}$ denotes the unique rotation invariant distribution on the unit sphere. For example, if $h \sim \mathcal{N}(0, I)$, then $h/\|h\| \overset{\text{unif}}{\sim} \mathcal{S}^{n-1}$

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

(C) Ammend the stated algorithm to use line search instead of solving for the exactly-optimal step size. To be clear, you don't have access to $\nabla f(x_t)$, all you are allowed to do at round t is the following:

(C.1) Sample *one* direction $g_t \overset{\text{unif}}{\sim} \mathcal{S}^{n-1}$.

(C.2) Making (finite) function evaluations of the form $f(x_t - \eta g_t)$ for $\eta \in \mathbb{R}$. Ideally, this should be at most logarithmic in problem parameters.

State *both* the number of iterations and the number of function evaluations. Are the rates qualitatively similar? *Hint:* Make the method a non-strict descent method, but accept that on some rounds, you might not make any progress, and you can let $x_{t+1} = x_t$. Just ensure that, with constant probability on each round, you make some progress.

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. Fix $n = 500$. We define the distribution over PSD matrix $\mathcal{D}(\epsilon)$:

Definition 0.1. $\mathcal{D}(\epsilon)$ is a distribution of matrix $\mathbf{M} = \mathbf{M}^\top$, defined as follows. Let $\mathbf{X} \in \mathbb{R}^{n \times n}$ denote a matrix with i.i.d $\mathcal{N}(0,1)$ entries. Generate a random vector $\mathbf{u}$ uniformly from the unit sphere. Define the matrix $\mathbf{M} = \frac{1}{\sqrt{2n}}(\mathbf{X} + \mathbf{X}^\top) + (1+\epsilon)\mathbf{u}\mathbf{u}^\top$.

Now, for each $\epsilon \in \mathcal{S} := \{1, .5, .2, .1, .05\}$, do the following

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

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

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

(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?