

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 2, 2018

Problem 1: Backtracking Line Search

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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  $t = 0, 1, 2, \dots$  do  
3    $g_t \leftarrow \nabla f(x_t)$ ;  
4   for  $k = 0, 1, 2, \dots$  do  
5     If “sufficient decrease” condition holds  
6        $f(x_t - \beta^k g_t) \leq f(x_t) - \alpha \beta^k \cdot \|g_t\|^2$ , (1)  
7       set  $\eta_t = \beta^k$  and break  
6   end  
7   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 all $t \in (0, 1/M]$.
- (B) Show that $\eta_t \geq \min\{1, \beta/M\}$. Conclude that the loop in Line 4 always terminates.

(C) Using part *b*, show that

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

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

$$f(x_t - \eta_t g_t) - f(x_*) \leq C(\alpha, \beta, M, m) \cdot (f(x_t) - f(x_*)) \quad (3)$$

Problem 2: Random Descent Directions

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 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  $t = 0, 1, 2, \dots$  do
3   Set  $g_t \stackrel{\text{unif}}{\sim} \mathcal{S}^{n-1}$ ;
4   Set  $\eta_t := \min_{\eta \geq 0} f(x_t - \eta g_t)$ ;
5   Set  $x_{t+1} \leftarrow x_t - \eta g_t$ ;
6 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 . $g_t \stackrel{\text{unif}}{\sim} \mathcal{S}^{n-1}$ denotes the unique rotation invariant distribution on the unit sphere. For example, if $h \sim \mathcal{N}(0, I)$, then $h/\|h\| \stackrel{\text{unif}}{\sim} \mathcal{S}^{n-1}$.

(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

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

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

(C) Amend 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. 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 $t = 1, 2, \dots, 10$.

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

$$\min_{x \in \text{span}(x_1, \dots, x_s)} \mathbf{f}(x) - \min_x \mathbf{f}(x) \quad (5)$$

(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 actually attained by heavy ball/nesterov, and the “optimal” krylov algorithm,

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