EECS 127/227AT **Fall 2020**

Homework 11

Homework 11 is not due on Gradescope. However, this is a regular homework, so it is expected that you will work on it as you would with the earlier homeworks. Solutions will be posted on Monday 11/30 around noon.

1 Gradient-based backtracking line search

This question is related to backtracking line search. The relevant portions of the textbooks are Secs. 12.1, 12.2.1 and 12.2.2 of the textbook of Calaiore and El Ghaoui and Secs. 9.1 and 9.2 of the textbook of Boyd and Vandenberghe.

Recall that gradient-based backtracking line search to attempt to minimize a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with domain $\mathbb{R}^n$ works as follows. Pick $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$. Suppose the algorithm is at $x^{(k)} \in \mathbb{R}^n$ and $\nabla f(x^{(k)}) \neq 0$. We carry out the following iteration to determine the next step of the algorithm.

- Set $s := 1$
- While $f(x^{(k)} - s\nabla f(x^{(k)})) > f(x^{(k)}) - \alpha s\|\nabla f(x^{(k)})\|_2^2$, replace s by βs and repeat.
- Return $x^{(k)} - s\nabla f(x^{(k)})$ as $x^{(k+1)}$.

At each step of the overall algorithm this inner loop (in order to determine $x^{(k+1)}$ from $x^{(k)}$) is guaranteed to terminate in a finite number of iterations because, when $\nabla f(x^{(k)}) \neq 0$, there is an open interval $(0, \bar{s})$ (i.e. $\bar{s} > 0$) such that $f(x^{(k)} - s\nabla f(x^{(k)})) < f(x^{(k)}) - \alpha s\|\nabla f(x^{(k)})\|_2^2$ for all $s \in (0, \bar{s})$. See Figure 1 to understand why this is true—it is basically because $\alpha < 1$ (here $\bar{s} = \infty$ is also allowed). Since we are decreasing the step size by a factor $\beta < 1$ at each iteration of the inner loop, the step size will become less than $\bar{s}$ in a finite number of steps, at which point we will have figured out what $x^{(k+1)}$ is.

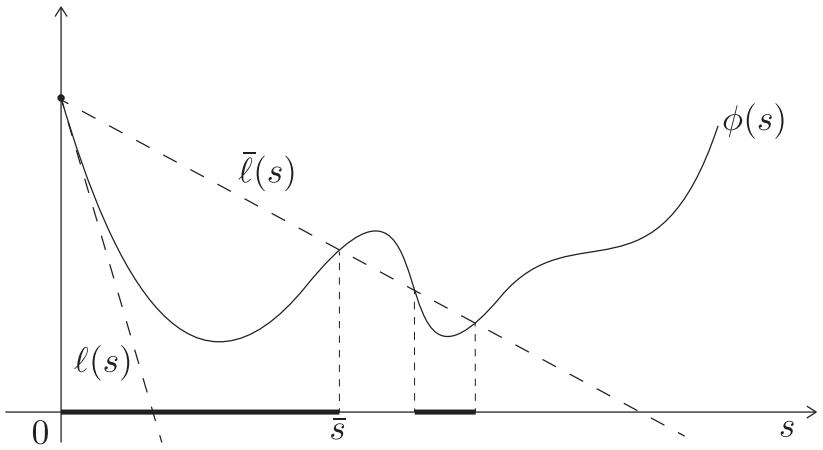

Figure 1: Backtracking line search. Here $l(s) := f(x^{(k)}) - s\|\nabla f(x^{(k)})\|_2^2$ and $\bar{l}(s) := f(x^{(k)}) - \alpha s\|\nabla f(x^{(k)})\|_2^2$. The abscissa is parametrized by s, so the graph is of $\phi(s) := f(x^{(k)} - s\nabla f(x^{(k)}))$

In class we considered the scenario where $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function with domain $\mathbb{R}^n$, all of whose sublevel sets are closed sets (one says that f is a closed function, or, what is the same thing, a

lower semicontinuous function). We then showed that gradient-based backtracking line search converges to a stationary point of f, i.e. a point where the gradient of f is zero. In particular, if f is a differentiable closed convex function with domain $\mathbb{R}^n$, achieving its minimum at some $x^* \in \mathbb{R}^n$, then gradient-based backtracking line search will converge to a global minimizer of f.

In this question we consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x) = x_1^2 + x_2^2$, with $\mathrm{dom}(f) := \{(x_1, x_2) : x_1 > 1\}$.

We want to solve the optimization problem

$$p^* := \min_{x \in \mathbb{R}^2} f(x).$$

Note that the domain of the problem is the domain of f.

(a) Is f a convex function?

(b) What is p^*?

(c) Is there an optimal point for the optimization problem?

(d) Suppose we run the gradient-based backtracking algorithm with the initial condition $x^{(0)} := \begin{bmatrix} 2 & 2 \end{bmatrix}^T$. Does the algorithm converge? If so, what point does it converge to? Is this point optimal for the optimization problem? Does the sequence of values of the objective function at the points of the algorithm converge? If so, does it converge to the optimal value?

(e) How can your reconcile your findings with the theorem about the convergence of gradient-based backtracking line search mentioned above?

2 Gradient descent vs Newton's method

This question is related to gradient descent and Newton's method. The relevant portions of the textbooks are Secs. 12.1 and 12.2 of the textbook of Calaiore and El Ghaoui and Secs. 9.1, 9.2, 9.3 and 9.5 of the textbook of Boyd and Vandenberghe.

In this question, we will explore the performance (in terms of convergence properties) of first order and second-order optimization algorithms with the help of this jupyter notebook. Please complete the required parts of it before moving on.

Gradient descent is a first-order iterative optimization algorithm that uses the first derivative information to find the optimal value of a function.

Newton's method, applied in optimization settings, is a second-order iterative algorithm that effectively finds the solution of the equation that the first derivative of a function equals zero, which, for a convex function, is the same as finding its minimum.

To find the minimum of a function $f(x)$, we usually start with an initial guess x_0 and then iterate over till some stopping criterion is met.

Let the optimization problem at hand be

$$\min_{x \in \mathbb{R}^m} f(x)$$

Using gradient descent, the iteration step is

$$x_{n+1} = x_n - \nabla f(x_n), \quad \text{for } n = 0, 1, 2, \ldots$$

Using Newton's method, the iteration step is

$$x_{n+1} = x_n - [Hf(x_n)]^{-1} \nabla f(x_n), \quad \text{for } n = 0, 1, 2, \ldots$$

where $Hf(x_n)$ is the Hessian of $f(x)$ computed at x_n

a) Consider the paraboloid given by

$$f(x) = x_1^2 + x_2^2 - 8x_1 + 2x_2 + 17.$$

i. Find the expression for $\nabla f(x)$, the first derivative of $f(x)$.

ii. Find the expression for $Hf(x)$, the Hessian of $f(x)$.

iii. Compute the value of x^*, at which the optimum is achieved for $f(x)$.

iv. With an initial assumption $x_0 = \begin{bmatrix} 8 \\ 3 \end{bmatrix}$, perform 100 iterations of gradient descent and Newton's method with a step size = 0.9 in the Jupyter Notebook. Plot the path taken by x in the 100 steps towards optimum for both the algorithms.

v. What did you observe about the path taken by x towards optimum for both the algorithms in this case?

b) Consider the halfpipe, given by

$$f(x) = \cosh(\epsilon x_1^2 + x_2^2), \quad \text{where } \epsilon = 0.05, \text{ and } \cosh \text{ is the hyperbolic cosine function.}$$

(a) Find the expression for $\nabla f(x)$, the first derivative of $f(x)$.

(b) Find the expression for $Hf(x)$, the Hessian of $Hf(x)$.

(c) With an initial assumption $x_0 = \begin{bmatrix} -2 \\ 0.9 \end{bmatrix}$, perform 5000 iterations of gradient descent and Newton's method with a step size = 0.1, in the Jupyter Notebook. Plot the path taken by x in the 5000 steps towards optimum for both the algorithms.

(d) What did you observe about the path taken by x towards optimum for both the algorithms in this case?

c) Which of the algorithms provides a more efficient path towards the optimum (x^*), starting from the same initial point? Justify your answer with proper reasoning.

Bonus: Change the step size in Jupyter Notebook for both the cases and notice the change in optimization paths. (Note: This will not be graded)

3 Descent Algorithms (Jupyter Notebook)

In this problem you will implement gradient descent and Newton's method and the perform some tests to compare the two descent method on two different functions.
Please complete the coding sections and comment on the outputs at this jupyter notebook.

4 Mirror Descent

This question studies a descent method called *mirror descent* which in a sense modifies the gradient descent step in a way that incorporates the local geometry of the function being optimized, as learned from the previous steps that the algorithm has visited. While this is not directly disucssed in the textbooks, it could be useful to consult Secs. 12.1, 12.2.1 and 12.2.2 of the textbook of Calaiore and El Ghaoui and Secs. 9.1, 9.2 and 9.3 of the textbook of Boyd and Vandenberghe.

Throughout this question, let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a convex differentiable function with domain $\mathbb{R}^n$. We seek an iterative first-order optimization algorithm for the unconstrained problem

$$p^* = \min_x f(x).$$

(a) Suppose that we have already seen the points $x_0, x_1, \ldots, x_k \in \mathbb{R}^n$, so we know the gradients $\nabla f(x_j)$ for $j = 0, 1, \ldots, k$. Let $\bar{x}_k$ be the average of these points; i.e.

$$\bar{x}_k := \frac{1}{k+1}\sum_{j=0}^{k} x_j.$$

Show that the function $\hat{f}_k : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\hat{f}_k(x) := f(\bar{x}_k) + \frac{1}{k+1}\sum_{j=0}^{k} \nabla f(x_j)^\top (x - x_j)$$

is a lower bound on f.

Since $\hat{f}_k$ is a lower bound on f, we might think to try minimizing $\hat{f}_k$ directly. However, since $\hat{f}_k$ is affine, the solution to $\min_x \hat{f}_k(x)$ is not very useful (the function is either unbounded below or constant.) Instead, we can try regularizing this minimization with a function $w : \mathbb{R}^n \longrightarrow \mathbb{R}$. We require w to be differentiable and 1-strongly convex; i.e.

$$w(y) \geq w(x) + \nabla w(x)^\top (y - x) + \frac{1}{2}\|y - x\|_2^2,$$

for all $x, y \in \mathbb{R}^n$. Our update rule is

$$x_{k+1} \coloneqq \arg\min_x \left(\hat{f}_k(x) + \frac{1}{\alpha(k+1)} w(x) \right),$$

where $\alpha > 0$ is the regularization parameter.

(b) The *Bregman divergence* associated with w is defined as

$$V_w(x, y) = w(y) - \nabla w(x)^\top (y - x) - w(x),$$

for pairs $x, y \in \mathbb{R}^n$. Note that, since w is convex, the Bregman divergence is nonnegative.

Assume that we begin at a stationary point of w, so that $\nabla w(x_0) = 0$. Show that

$$x_{k+1} = \arg\min_x \left(V_w(x_k, x) + \alpha \nabla f(x_k)^\top (x - x_k) \right).$$

Remark: If we choose $w(x) = \frac{1}{2}\|x\|_2^2$, then $V_w(x, y) = \frac{1}{2}\|x - y\|_2^2$. Setting the gradient to zero, we see

$$x_{k+1} = x_k - \alpha \nabla f(x_k).$$

That is, in this case the update rule is just gradient descent with step size α.

(c) Using the strong convexity of w, show that

$$\min_x \left(V_w(x_k, x) + \alpha \nabla f(x_k)^\top (x - x_k) \right) \geq -\frac{\alpha^2}{2} \|\nabla f(x_k)\|_2^2. \tag{1}$$

(d) Show that the Bregman divergence satisfies

$$-\nabla_y V_w(x, y)^\top (y - u) = V_w(x, u) - V_w(y, u) - V_w(x, y). \tag{2}$$

Remark: If $w(x) = \frac{1}{2}\|x\|_2^2$, then (2) tells us

$$2(y - x)^\top (u - y) = \|u - x\|_2^2 - \|u - y\|_2^2 - \|y - x\|_2^2.$$

This is just the law of cosines.

(e) Show for any $u \in \mathbb{R}^n$ that

$$\alpha \nabla f(x_k)^\top (x_k - u) \leq \frac{\alpha^2}{2} \|\nabla f(x_k)\|_2^2 + V_w(x_k, u) - V_w(x_{k+1}, u). \tag{3}$$

(f) Let $x^* = \arg\min_x f(x)$, and assume that $\|\nabla f(x)\|_2 \leq \rho$ everywhere. Recall we define $\bar{x}_k = \frac{1}{k+1} \sum_{j=0}^k x_k$.
Conclude that

$$f(\bar{x}_T) \leq f(x^*) + \frac{\alpha \rho^2}{2} + \frac{V_w(x_0, x^*)}{\alpha(T+1)}.$$

In particular, if we set $\alpha = \varepsilon/\rho^2$ and $T + 1 = 2\rho^2 V_w(x_0, x^*)/\varepsilon^2$, we get

$$f(\bar{x}_T) \leq \varepsilon$$

for arbitrary $\varepsilon > 0$.

5 Gradient descent on a graph

This question studies a gradient descent method to solve an optimization problem where the variables are thought of as being parametrized by the vertices of an undirected graph. The relevant portions of the textbooks are Secs. 12.1, 12.2.1 and 12.2.2 of the textbook of Calaiore and El Ghaoui and Secs. 9.1, 9.2 and 9.3 of the textbook of Boyd and Vandenberghe.

We are given an undirected simple graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges.

(a) Consider the problem of assigning weights x_i to each vertex $i \in V$ such that adjacent vertices get similar weights, and the sum of weights is close to 1. That is, we want the solution to the optimization problem

$$x^* := \arg\min_{x \in \mathbb{R}^n} \sum_{(i,j) \in E} (x_i - x_j)^2 + 2\lambda \left(\sum_{i \in V} x_i - 1 \right)^2$$

where $\lambda \in \mathbb{R}$ is a constant. Show that this optimization problem is equivalent to

$$x^* = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top (L + \lambda \mathbb{1}\mathbb{1}^\top) x - \lambda \mathbb{1}^\top x$$

where L is the Laplacian matrix for G and $\mathbb{1}$ is the all-ones vector in $\mathbb{R}^n$.

(b) What is the optimal x^*?

(c) Suppose we use gradient descent with step size $\eta > 0$ to find the optimal x^*. Write the gradient descent step; i.e., express x_{k+1}, the $(k+1)$th step of gradient descent, in terms of x_k, L, η, and λ.

(d) Show that $x_{k+1} - x^* = (I - \eta(L + \lambda \mathbb{1}\mathbb{1}^\top))(x_k - x^*)$.

(e) Let $\lambda_1 \geq \ldots \geq \lambda_n$ be the eigenvalues of L, and assume λ is given such that $\lambda_1 \geq n\lambda \geq \lambda_{n-1}$. Show that $\|x_k - x^*\|_2 \leq \rho^k \|x_0 - x^*\|_2$ for $\rho := \max\{|1 - \eta\lambda_{n-1}|, |1 - \eta\lambda_1|\}$, where x_0 is the starting point of the gradient descent.

(f) Assuming that $\eta > 0$ is small enough that $0 < \rho < 1$, find the number of time steps needed to converge to some $\varepsilon > 0$ around x^* as a function of η, assuming $\|x_0 - x^*\|_2 > \varepsilon$. That is, find $t(\eta)$ such that $\|x_k - x^*\|_2 \leq \varepsilon$ for $k \geq t(\eta)$.

(g) Find the optimal step size, i.e. the solution to

$$\eta^* = \arg\min_{\{\eta > 0 : 0 < \rho < 1\}} t(\eta).$$

What is the corresponding $t(\eta^*)$?