EE 381V: Large Scale Optimization **Fall 2012**

Lecture 4 — September 9

Lecturer: Caramanis & Sanghavi *Scribe: Michael Motro, Janice Pan, Shun Zhang*

4.1 Recall from last lecture

4.1.1 Gradient descent

$$x^{(k+1)} = x^{(k)} - \eta^{(k)} \nabla f(x^{(k)}) \tag{4.1}$$

- $-\nabla f(x^{(k)})$ is the descent direction
- η is the step size

Definition 1. *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz if and only if*

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L||x - y||_2, \qquad \forall x, y \in \mathbb{R}^n \tag{4.2}$$

Theorem 4.1. *If f is L-Lipschitz and there exists an optimum x^*, then the gradient descent with a fixed step size $\eta < \frac{2}{L}$ will converge to a x^* (a stationary point) from any initial point.*

We want to be able to set bounds on:

- $f(x^{(k)}) - f^*$
- $\|x^{(k)} - x^*\|$

4.2 This lecture

- Strong convexity
- Upper bound on $\nabla^2 f(x)$
- Condition number
- Exact line search

First we explore the idea of strong convexity so we can characterize how far we are from the optimum as a function of $-\nabla f(x^{(k)})$. The notion of strong convexity will help us understand a lower bound on the eigenvalues of the Hessian, $\nabla^2 f(x)$. Then we will discuss how $\nabla^2 f(x)$ is upper bounded and how these bounds describe how well-conditioned an optimization problem. This is quantified in the condition number. Finally, we will introduce the exact line search algorithm, the goal of which is to select a step size η.

4.3 Strong convexity

Definition 2. *If there exist a constant $m > 0$ such that $\nabla^2 f \succeq mI$ for all $x \in S$, then the function $f(x)$ is a* strongly convex function *on S.*

When $m = 0$, we recover the basic inequality characterizing convexity; for $m > 0$, we obtain a better lower bound on $f(y)$ than that from convexity alone. The value of m reflects the shape of convex functions. Typically as shown in Figure (4.1) , a small m corresponds to a 'flat' convex function while a large m corresponds to a 'steep' convex function.

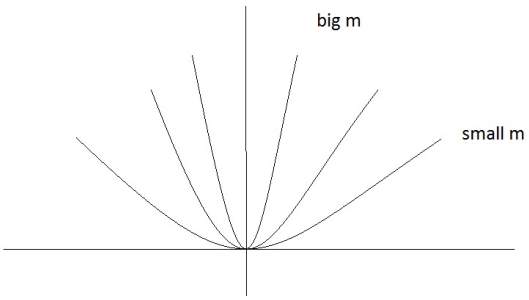

Figure 4.1. A strongly convex function with different parameter m. The larger m is, the steeper the function looks like.

Lemma 4.2. *If f is strongly convex on S, we have the following inequality:*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{m}{2}\|y - x\|^2 \tag{4.3}$$

for all x and y in S.

Proof: First, recall the Mean Value Theorem, which states that for f and f' continuous on [a,b] and f' differentiable on (a,b), there exists a number c in (a, b), $a < c < b$ such that

$$f(b) = f(a) + f'(a)(b - a) + \frac{f''(c)}{2}(b - a)^2. \tag{4.4}$$

Extending the Mean Value Theorem to higher-dimensional space, we have

$$f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(z)(y - x)$$

for some z on the line segment $[x, y]$. By the strong convexity assumption (see the definition for strong convexity above) the last term on the right-hand side is at least $\frac{m}{2}\|y - x\|_2^2$ □

4.4 Using Strong Convexity

Strong convexity has several interesting consequences. We will see that we can bound both $f^* - f(x)$ and $||x - x^*||_2$ in this section.

We will first show that the inequality can be used to bound $f(x) - f^*$, which is the sub-optimality of the point x, in terms of $\|\nabla f(x)\|_2$. The righthand size is a convex quadratic function of y (for fixed x). Setting the gradient with respect to y equal to zero, we can find the $\tilde{y}$ that minimizes the right hand side.

$$\begin{aligned}\frac{\partial}{\partial x}(f(x) + \langle \nabla f(x), y - x\rangle + \frac{m}{2}\|y - x\|^2) &= 0\\ \nabla f(x) - m(y - x) &= 0\\ y = x - \frac{1}{m}\nabla f(x)\end{aligned}$$

So $\tilde{y} = x - (1/m)\nabla f(x)$ minimizes the righthand side. Plug this into the righthand side, we can derive the lower bound of $f(y)$, for arbitrary y.

$$\begin{aligned}f(y) &\geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{m}{2}\|y - x\|^2\\ &\geq f(x) + \langle \nabla f(x), \tilde{y} - x\rangle + \frac{m}{2}\|\tilde{y} - x\|^2\\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|^2\end{aligned}$$

By substituting y with x^*,

$$f^* \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|^2 \tag{4.5}$$

This allows us to realize how fast you get to a minimum as a function of gradient. If the gradient is small at a point, then the point is nearly optimal.

Similarly, we can also derive a bound on $\|x - x^*\|_2$, the distance between x and any optimal point x^*, in terms of $\|\nabla f(x)\|_2$:

$$\|x - x^*\|_2 \leq \frac{2}{m}\|\nabla f(x)\|_2 \tag{4.6}$$

where $x^* = \arg\min_x f(x)$.

Proof: We apply (4.2) with $y = x^*$ to obtain:

$$\begin{aligned}f^* = f(x^*) &\geq f(x) + \langle \nabla f(x), x^* - x\rangle + \frac{m}{2}\|x^* - x\|_2^2\\ &\geq f(x) - \|\nabla f(x)\|_2\|x^* - x\|_2 + \frac{m}{2}\|x^* - x\|_2^2,\end{aligned}$$

Since $f^* \leq f(x)$, the terms following $f(x)$ on the righthand side must be negative. We have

$$\begin{aligned} -\|\nabla f(x)\|_2\|x^* - x\|_2 + \frac{m}{2}\|x^* - x\|_2^2 &\leq 0 \\ \|x - x^*\|_2 &\leq \tfrac{2}{m}\|\nabla f(x)\|_2 \end{aligned}$$

from which (4.6) follows. One consequence of (4.6) is x^* is unique and the solution locates within a ball of radius of $\|\nabla f(x)\|^2$ around the optimal solution. □

4.4.1 Upper Bound on $\nabla^2 f(x)$

The inequality (4.3) implies that the sublevel sets contained in S are bounded, so in particular, S is bounded. Therefore the maximum eigenvalue of $\nabla^2 f(x)$, which is a continuous function of x on S, is bounded above on S. And there exists a constant M such that $\nabla^2 f(x) \preceq MI$ for all $x \in S$.

Lemma 4.3. *For any $x, y \in S$, if $\nabla^2 f(x) \preceq MI$ for all $x \in S$ then*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{M}{2}\|y - x\|^2 \tag{4.7}$$

Proof: The proof is analogous to the proof of (4.3). □

4.4.2 Condition Number

From the strong convexity inequality (4.3) and the inequality (4.7), we have:

$$mI \preceq \nabla^2 f(x) \preceq MI \tag{4.8}$$

Definition 3. *If $mI \preceq \nabla^2 f(x) \preceq MI$ for all $x \in S$, then the* **condition number** *of f is $k = \frac{m}{M}$.*

Note that the range of condition number is between 0 and 1. The condition number is thus an lower bound on the condition number of the matrix $\nabla^2 f(\mathbf{x})$, i.e., the ratio of its largest eigenvalue to its smallest eigenvalue.

Definition 4. *When the ratio is close to 1, we call it* **well-conditioned**. *When the ratio is close to 0, we call it* **ill-conditioned**.

When the ratio is exactly 1, it is the best case that only one step will lead to the optimal solution (i.e., there is no wrong direction).

Figure 4.2 shows an example of level sets for different functions. Their corresponding m and M are shown below, where $a > b$. Note that finding optimal solution in the 2nd or 3rd function is easier than doing that in the 1st function, as the step size can be easily determined when $m = M$.

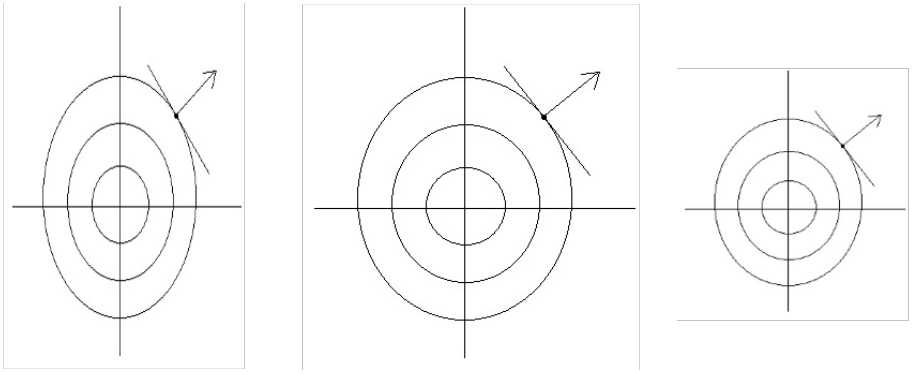

Figure 4.2. Functions with different m and M.

	1st function	2nd function	3rd function
m	a	b	a
M	b	b	a

It must be kept in mind that constants m and M are known only in the rare cases, so the the inequality cannot be used as a practical stopping criterion. It can be considered as *conceptual* stopping criterion; it shows that if the gradient of f at x is small enough, then the difference between $f(x)$ and f^* is small. If we terminate an algorithm when $\|\nabla f(x^k)\|_2 \leq \eta$, where η is chosen small enough to be smaller than $(m\epsilon)^{\frac{1}{2}}$, then we have $f(x^k) - f^* \leq \epsilon$, where ϵ is some positive tolerance.

Though these bounds involve the (usually) unknown constants m and M, they establish that the algorithm converges, even if the bound on the number of iterations required to reach a given accuracy depends on constants that are unknown.

4.4.3 Linear Convergence

As review, a function $f(x)$ can be considered linearly convergent to f^* if:

$$\lim_{k \to +\infty} \frac{(f(x^{(k+1)}) - f^*)}{(f(x^{(k)}) - f^*)} = c\,,\ c \in (0,1) \tag{4.9}$$

This is helpful as it specifies a steady rate of convergence, rather than guaranteeing convergence but with no specification as to how long it might take.

Theorem 4.4. *For $f(x)$ so that $mI \preceq \nabla^2 f(x) \preceq MI$, gradient descent with a step size of $\eta = \frac{1}{M}$ for k iterations will result in:*

$$f(x^{(k)}) - f^* \leq c^k (f(x^{(0)}) - f^*) \tag{4.10}$$

where $c = 1 - \frac{m}{M}$.

Proof:

1. $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{M}{2}\|y - x\|^2$ from lemma 4.3.
2. Use this equation for a single step from x to x^+, and recall $x^+ = x - \eta \nabla f(x)$:

$$\begin{aligned} f(x^+) &\leq f(x) + \langle \nabla f(x), -\eta \nabla f(x) \rangle + \frac{M}{2}\| - \eta \nabla f(x)\|^2 && (4.11) \\ &\leq f(x) - \eta \|\nabla f(x)\|^2 + \frac{M}{2}\eta^2 \|\nabla f(x)\|^2 && (4.12) \end{aligned}$$

3. Setting step size $\eta = \frac{1}{M}$:

$$f(x^+) \leq f(x) - \frac{1}{2M}\|\nabla f(x)\|^2 \quad (4.13)$$

4. $f(y) \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|^2$ from lemma 4.2, rearrange so:

$$(-\|\nabla f(x)\|^2) \leq 2m(f(y) - f(x)) \quad (4.14)$$

   Note that this holds true for $y = x^*$, the optimal point.

5.

$$\begin{aligned} f(x^+) &\leq f(x) - \frac{1}{2M}\|\nabla f(x)\|^2 \leq f(x) + \frac{1}{2M}(2m(f(x^*) - f(x)) && (4.15) \\ f(x^+) &\leq f(x) - \frac{m}{M} f(x) + \frac{m}{M} f(x^*) && (4.16) \\ f(x^+) - f(x^*) &\leq (1 - \frac{m}{M})(f(x) - f(x^*)) && (4.17) \end{aligned}$$

6. This is clearly linear convergence to $f(x^*)$ with a rate of convergence of $c = 1 - \frac{m}{M}$. Were this iteration repeated k times, the difference from the minimum would be multiplied by c each time, resulting in Equation 4.10.

□

4.5 Exact Line Search

We've proven that using $\eta = \frac{1}{M}$ as the step size for gradient descent always provides an acceptable convergence. The problem with this method is that M is usually not known. There are several other methods for choosing the step size, including step sizes that vary for each iteration. Of the latter, the most straightforward method is exact line search, which calculates the optimal step size for every iteration. This results in each iteration having an optimization problem of its own.

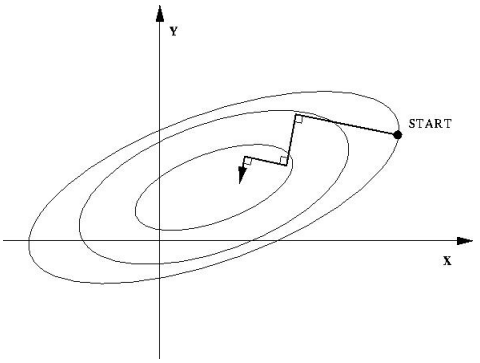

Figure 4.3. Exact Line Search

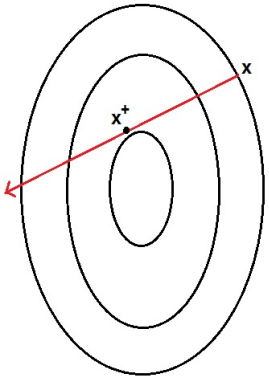

Figure 4.4. Exact Line Search, Single Iteration

Algorithm (Gradient descent with exact line search)

1. Set iteration counter $k = 0$, and make an initial guess x_0 for the minimum
2. Compute $\nabla f(x^{(k)})$
3. Choose $\eta^{(k)} = \arg\min_{\eta}\{f\left(x^{(k)} - \eta\nabla f(x^{(k)})\right)\}$
4. Update $x^{(k+1)} = x^{(k)} - \eta^{(k)}\nabla f(x^{(k)})$ and $k = k + 1$.
5. Go to 2 until $\|\nabla f(x^{(k)})\| < \epsilon$

An exact line search is used when the cost of the minimization problem with one variable is low compared to the cost of computing the search direction itself. However, the algorithm is not very practical.

4.5.1 Rate of Convergence for Exact Line Search

We can determine the rate of convergence by comparing exact line search to the previous fixed-step descent.

- $$\hat{x} = x - \eta_{ELS}\nabla f(x) \tag{4.18}$$
- $$x^+ = x - \frac{1}{M}\nabla f(x) \tag{4.19}$$

- $$f(\hat{x}) \text{ by definition } \leq f(x^+) \leq f(x) - \frac{1}{2M}\|f(x)\|^2 \tag{4.20}$$

$f(\hat{x})$ thus fulfills step 3 of the proof for theorem 4.4, the rest of the proof can be followed identically for this case. Therefore exact line search also converges linearly with rate $c \leq (1 - \frac{m}{M})$.

4.6 Example

Consider the strongly convex d-dimensional sphere,

$$f(x) = \sum_{i=1}^{d} x_i^2. \tag{4.21}$$

Let us take a simple case: $d = 2$, so we get

$$f(x) = x_1^2 + x_2^2. \tag{4.22}$$

Computing the gradient and the Hessian,

$$\nabla f(x) = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \tag{4.23}$$

$$\nabla^2 f(x) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \tag{4.24}$$

Now we want to use the exact line search method to find the optimal step size η, which minimizes

$$g(\eta) = f(x - \eta \nabla f(x)). \tag{4.25}$$

Plugging in, we find that

$$g(\eta) = (1 - 2\eta)^2 (x_1^2 + x_2^2), \tag{4.26}$$

Taking the derivative of g with respect to η and setting equal to 0, we get $\eta = 0.5$.

In the Figure (4.6), we show the results from using too small of a step size ($\eta = 0.005$), too large of a step size ($\eta = 1.05$), and the step size we found using exact line search ($\eta = 0.5$). For $\eta = 0.005$, the algorithm did not reach the optimum because it converged too slowly. For $\eta = 1.05$, the algorithm also did not reach the optimum because it diverged. For $\eta = 0.5$, the algorithm converged in 1 step.

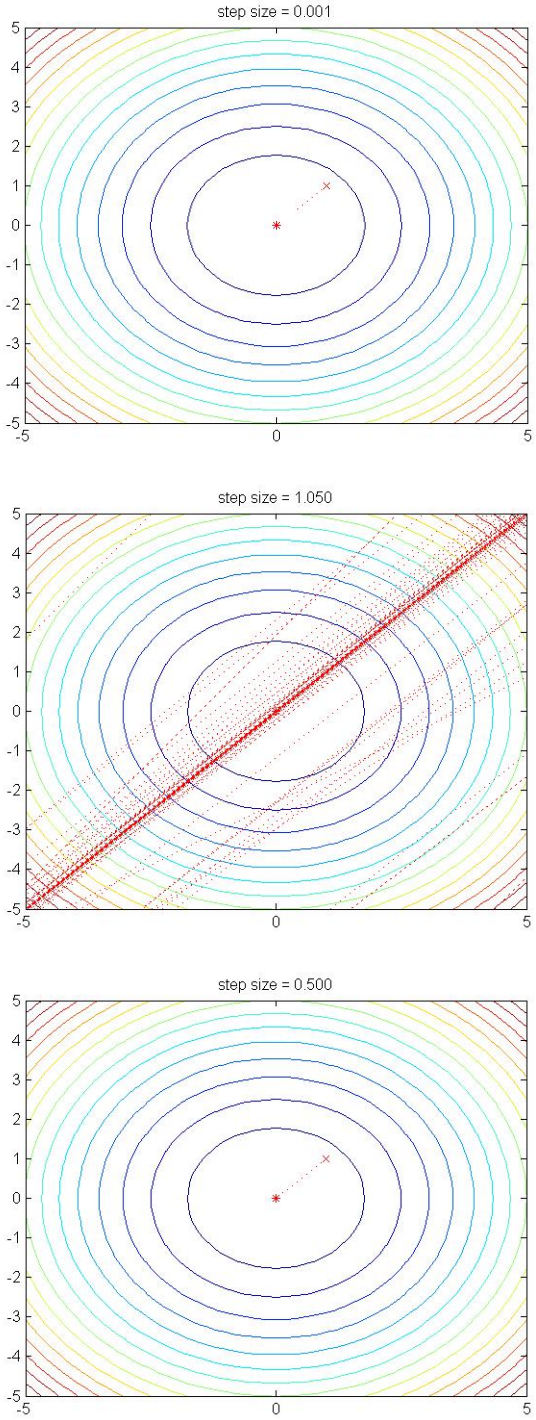

Figure 4.5. $\eta = 0.005$, $\eta = 1.05$, $\eta = 0.5$

MATLAB code to generate the figures:

```
% starting point
x0 = [1,1];

% for plotting contours
[x1,x2] = meshgrid(-5:0.2:5,-5:0.2:5);
f = x1.^2 + x2.^2;

% define various stepsizes
etasmall = 0.0005; % too small
etalarge = 1.05; % too large
etaELS = 0.5; % exact line search step size

etas = [etasmall, etaELS, etalarge];
maxiterations = 1000;
for i = 1:numel(etas)
    clear x;
    x(1,:) = x0;
    convFlag = false;
    eta = etas(i);
    j = 1;
    while j <= maxiterations && ~convFlag;
        j = j + 1;
        x(j,:) = x(j-1,:) - eta * gradientf(x(j-1,:));
        if isequal(x(j,:),zeros(1,2))
            convFlag = true;
        end
    end
    if ~convFlag
        fprintf('\nFor eta = %d, algorithm did not reach optimum.\n',eta);
    else
        fprintf('\nFor eta = %d, algorithm converged in %d steps.\n',eta,j-1);
    end
    figure;
    [c,h] = contour(x1,x2,f,15);
    hold on, plot(x(:,1),x(:,2),'r:');
    plot(x0(1),x0(2),'rx');
    plot(0,0,'r*');
```

```
    plottitle = sprintf('step size = %.3f',eta);
    title(plottitle);
    hold off;
end

function gradf = gradientf(x)
gradf = 2*x;
```