EE 381V: Large Scale Optimization **Fall 2014**

Lecture 3 — September 4

Lecturer: Sanghavi *Scribe: Taewan Kim, Vatsal Shah, Shanshan Wu*

3.1 Recap

The previous lecture began with the first order conditions of optimality which gives a characterization of the optimal solution for an unconstrained optimization of a smooth convex objective function:

$$\hat{x} \text{ is a global minimum iff } \nabla f(\hat{x}) = 0.$$

Some examples on convex modelling, optimal inequalities in probability and matrix completion were briefly discussed to give us a flavor regarding the diverse set of problems that fall under the framework of convex optimization. The last lecture concluded with some basic key definitions of interior point, feasible directions, closure, convex cone, polar cone, tangent cone and normal cone; some of which will be revisited as they have been utilized while providing intuition for the gradient descent algorithm.

Definition 1. (Convex Cone) *A set $K \subseteq \mathbb{R}^n$ is called a* **convex cone**, *if $x_1, x_2 \in K$ implies that $\lambda_1 x_1 + \lambda_2 x_2 \in K,\ \forall \lambda_1, \lambda_2 \geq 0$.*

Definition 2. (Tangent Cone) *Let $C \subseteq \mathbb{R}^n$ be a nonempty set, and let $x \in C$. Then the* **tangent cone** *of C at x, denoted by $T_C(x)$, is defined as follows*

$$T_C(x) = closure(F_C(x)). \tag{3.1}$$

Definition 3. (Normal Cone) *Let $C \subseteq \mathbb{R}^n$ be a nonempty, convex set, and let $x \in C$. Then the* **normal cone** *of C at x, denoted by $N_C(x)$, is defined as follows*

$$N_C(x) = \{s :\ \langle s, y - x \rangle \leq 0,\ \forall y \in C\}. \tag{3.2}$$

3.2 Overview of this Lecture

In this lecture we formulate the optimality condition for the convex optimization problem, using the concepts of convex cone defined in the last lecture. Finding the optimal point based on the optimality condition may be cumbersome, so a practical way to approach the optimal point is then introduced: the gradient descent method. Discussions on how different step size impacts the convergence rate will be provided.

3.3 Optimality Conditions

In the previous lecture we introduce a sufficient condition for the convex optimization problem (constrained or unconstrained) to achieve optimality: if there exists an $x \in \mathcal{X}$, where $\mathcal{X}$ denotes the feasible set, such that $\nabla f_0(x) = 0$, then x is the optimum. A natural question then follows: what if we cannot find such an x, i.e., what if no point in $\mathcal{X}$ has zero gradient? The answer differs for unconstrained and constrained problems, so let us look at them separately.

- For an unconstrained optimization problem, if for $\nabla f_0(x) \neq 0$ all $x \in \mathbb{R}^n$, then $f(x)$ has no optimal point. This is easy to understand: for any $x \in \mathbb{R}^n$, we can always find an $\hat{x} = x - \epsilon \nabla f_0(x)$, where $\epsilon > 0$, that satisfies $f(\hat{x}) < f(x)$; in other words, no point in $\mathbb{R}^n$ can claim that it has reached the minimal point. Examples of convex functions are shown in Fig. **??**.

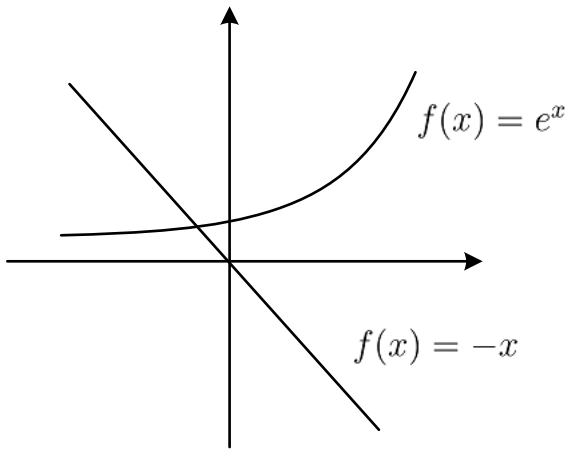

Figure 3.1. Examples of convex functions with no optimal point.

- For a constrained optimization problem, if no interior point of the feasible set $\mathcal{X}$ has zero gradient, then the optimal point lies on the boundary of $\mathcal{X}$. As an example, consider the following convex optimization problem:

$$\begin{aligned} &\min \ f_0(x) \\ &\text{s.t. } g(x) \leq 0, \end{aligned} \tag{3.3}$$

where $f_0(x)$ and $g(x)$ are shown in Fig. **??**. Let x_1 and x_2 be the two boundary points of $\mathcal{X}$, then Fig. **??** indicates that x_1 is the optimum for the left case while x_2 is the optimum for the right case. Can we mathematically highlight the optimum point from

the two boundary points? The available parameters that we can use to characterize the boundary points would be

$$\begin{aligned} &f_0(x_1),\ \nabla f_0(x_1),\ g(x_1),\ \nabla g(x_1),\\ &f_0(x_2),\ \nabla f_0(x_2),\ g(x_2),\ \nabla g(x_2). \end{aligned} \tag{3.4}$$

A careful look at Fig. **??** would reveal that $\nabla f_0(x)$ and $\nabla g(x)$ at the optimal boundary point always have the opposite sign. In other words, for this simple 1-dimensional case, x is an optimum of the constrained problem if it satisfies the conditions:

$$g(x) = 0 \text{ and } \exists \lambda > 0 \text{ s.t. } \nabla f_0(x) + \lambda \nabla g(x) = 0. \tag{3.5}$$

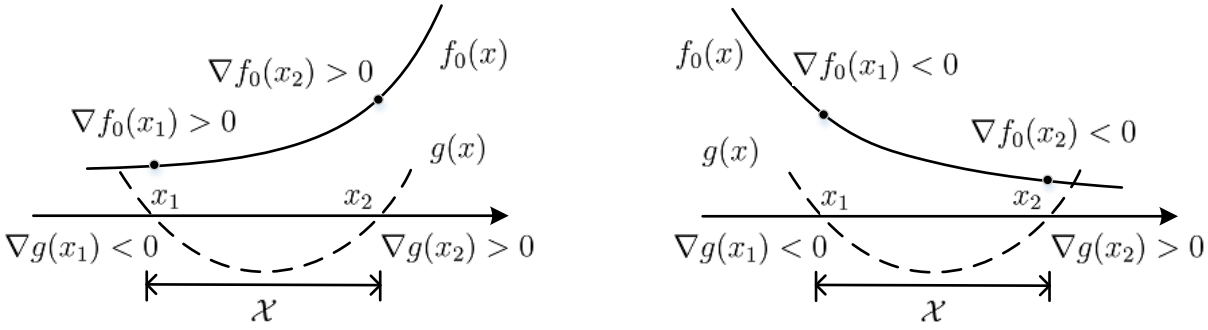

Figure 3.2. Examples of constrained optimization problems with optimal point on the boundary of the feasible set.

Now we want to generalize the above condition to characterize the optimal point for an n-dimensional case. Consider a general convex optimization problem

$$\begin{aligned} &\min\ f(x)\\ &\text{s.t. } x \in \mathcal{X}. \end{aligned} \tag{3.6}$$

By convexity, local and global optimality are equivalent. Thus, intuitively, x^* is optimal if and only if $f(x)$ increases along any direction that starts from x^* to the feasible set, i.e.,

$$x^* \text{ is optimum } \iff \text{ no feasible direction are descent.} \tag{3.7}$$

To describe the above condition mathematically, we have if x^* is optimal, then

$$\begin{aligned} &\langle \nabla f(x^*), x - x^* \rangle \geq 0,\ \forall x \in \mathcal{X}\\ \iff &\langle -\nabla f(x^*), x - x^* \rangle \leq 0,\ \forall x \in \mathcal{X}\\ \iff &-\nabla f(x^*) \in (T_{\mathcal{X}}(x^*))^\circ\\ \iff &-\nabla f(x^*) \in N_{\mathcal{X}}(x^*). \end{aligned} \tag{3.8}$$

Eq. **??** is in fact a general optimal condition for the convex optimization problem, regardless of whether it is a constrained or unconstrained, and whether the optimal point is in the interior or on the boundary of the feasible set. To see this, note that the sufficient condition for optimality that we introduced at the beginning of this lecture, i.e., $\nabla f(x) = 0$, is actually a special case of Eq. **??**. This is because if there exists an x^* with $\nabla f(x*) = 0$, then for all $x \in \mathcal{X}$, $\langle \nabla f(x^*), x - x^* \rangle = 0$, that means x^* would definitely satisfy Eq. **??**. On the other hand, for an unconstrained optimization problem with $\mathcal{X} = \mathbb{R}^n$, we have for any $x \in \mathcal{X}$, $N_{\mathcal{X}}(x) = \{0\}$. Therefore, Eq. **??** simply becomes $\nabla f(x^*) = 0$. This indicates that for unconstrained problems, $\nabla f(x^*) = 0$ is both the sufficient and necessary condition for x^* to be the optimal value, which is also consistent with the discussion we had at the beginning of this lecture.

Next we show through an example that Eq. **??** is precisely the geometric condition that the first-order KKT conditions are attempting to express. Consider a constrained optimization problem, as illustrated in Fig. **??**:

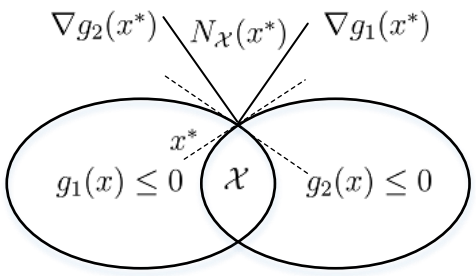

Figure 3.3. Example of optimal condition.

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } g_1(x) \leq 0 \\ &\quad\ g_2(x) \leq 0. \end{aligned} \tag{3.9}$$

In this case the feasible set is $\mathcal{X} = \{x : g_1(x) \leq 0,\ g_2(x) \leq 0\}$. We assume that the optimal point is on the boundary of $\mathcal{X}$. Then the normal cone at the point x^* is simply given by $N_{\mathcal{X}}(x^*) = \text{cone}(\nabla g_1(x^*), \nabla g_2(x^*))$, where $\text{cone}(x, y)$ denotes the cone formed by taking all the positive linear combinations of x and y. Accordingly, the optimal condition becomes $\exists \lambda_1 > 0,\ \lambda_2 > 0$ such that

$$-\nabla f(x^*) = \lambda_1 \nabla g_1(x^*) + \lambda_2 \nabla g_2(x^*). \tag{3.10}$$

This is exactly the first order KKT condition for optimality. In order to derive the KKT condition, we relied on the fact that the normal cone at x^* could be expressed using the gradients of the constraint functions. As we will see later, while this is commonly the case, it

is not always so. This is why some problems, even though convex, may fail to have Lagrange multipliers. In addition, evaluating the gradients of the convex functions may be sometimes cumbersome. Instead of computing the optimal value, we can take a more practical approach to find the optimal point: the descent methods. Before we go to the details of these methods, let us first give an definition for sublevel sets, a concept that is quite useful for visualizing the convex functions and analyzing the descent methods.

3.4 Sublevel Sets

A sublevel set S_L is defined as: $S_L = \{x : f(x) \leq L\}$. In Fig. **??** we show the sublevel sets by drawing its boundary, i.e., the contours of $f(x)$. Note that if $f(x)$ is convex function, then its sublevel sets are also convex, as indicated by Fig. **??**. However, the reverse is not true, i.e., if the sublevel sets of a function are convex, then the function could be non-convex. A simple example is $f(x) = \sqrt{||x||}$.

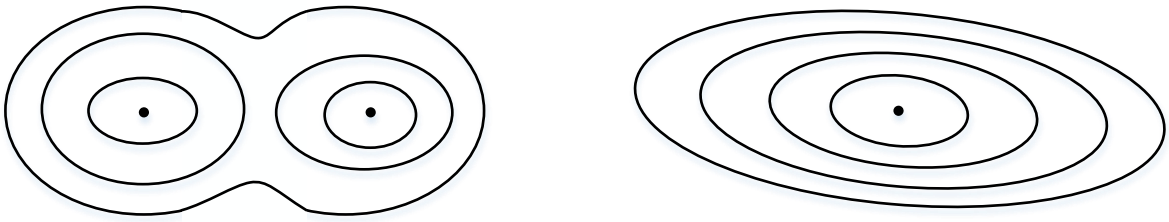

Figure 3.4. Sublevel sets for non-convex (left) and convex functions (right).

Sublevel sets give a convenient way to visualize the gradient of a function at every point, i.e., in which direction the function increase most rapidly, and how fast a function changes in a given direction. For example, as shown in Fig. **??**, the convex function changes fast (i.e. is steep) along the vertical direction and changes slowly (i.e. is flat) along the horizonal direction, since the contours are more close to each other in the vertical direction.

3.5 Gradient Descent Algorithm

The key idea for a descent algorithm for convex functions is the simple notion underlying the power of convexity:

$$\textit{Convexity} \;\Rightarrow\; \textit{local optimum} = \textit{global optimum}.$$

For nonconvex functions, the descent methods may converge to the local optimum instead of the global optimum.

This is illustrated in Figure ?? and Figure ?? [1]:

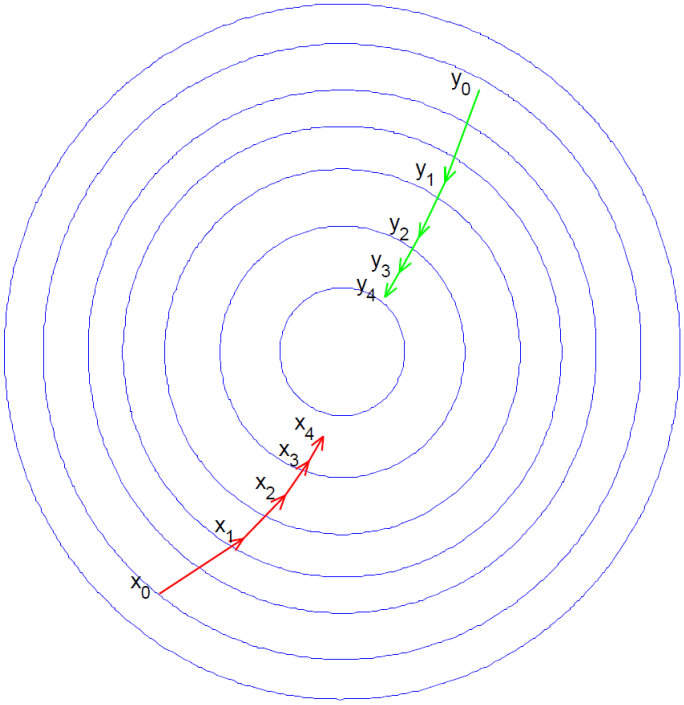

Figure 3.5. Gradient Descent on convex functions

Figure ?? is a contour of a convex function which has a unique minimum which is the global minimum and so the gradient descent algorithms converges to that unique minimum irrespective of the starting point.

Figure ?? is a contour of a non-convex function, $x^2+y^2(1-x)^3$, which has a two minimums: a local minimum and a global minimum. The gradient descent algorithm converges to either minima depending on the starting point. In the figure, we can see that depending on the initial point the x_n iterates go to the local minimum at $(x, y) = (0, 0)$ while the y_n iterates converge to the global minimum at$(x, y) = (2, 3)$.

3.5.1 Descent Methods

Definition 4. *A sequence* $x_1, x_2, x_3, \ldots$ *descends if* $f(x_1) > f(x_2) > f(x_3) \cdots$

Let us consider the following equation:

$$x^+ = x + t\Delta x \tag{3.11}$$

[1]The contours in ?? and ?? were drawn using some part of the source code given at *http* : *//commons.wikimedia.org/wiki/File* : *Gradientdescent.svg*

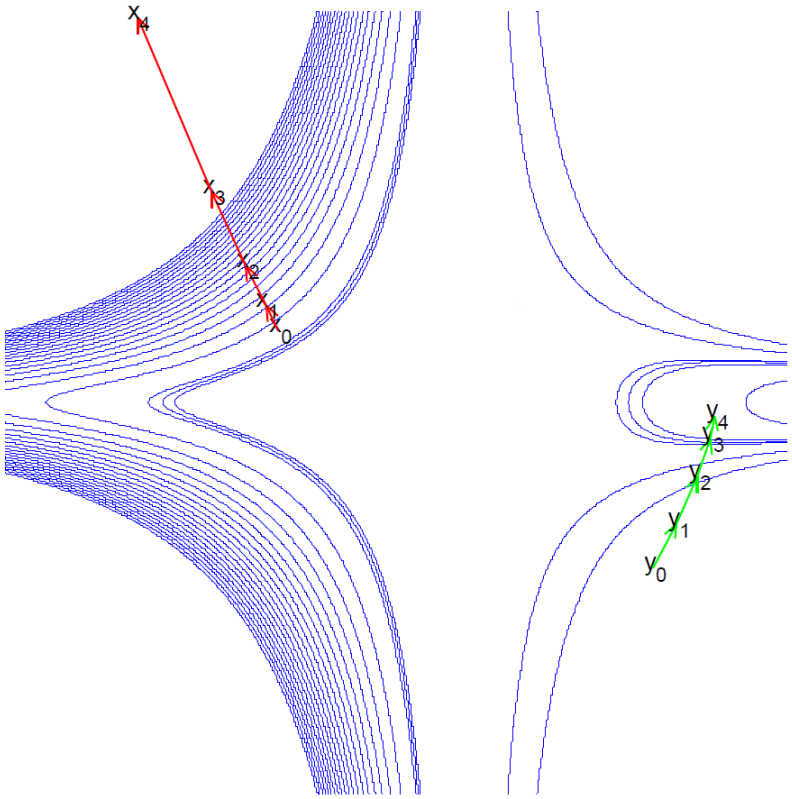

Figure 3.6. Gradient Descent on non-convex functions

In the above equation, x^+ represents the new point while x denotes the point under consideration, t is the step size and Δx is the direction vector.
Intuitively, at each iterate, we would like to ensure that the next step taken by this algorithm results in a smaller function value at the next iterate. Thus, a descent method is the one in which $f(x^+) < f(x)$ for every step.

3.5.2 Gradient descent

In the gradient descent algorithm, the descent direction is indicated by $\Delta x = -\nabla f(x)$ while the step size is $t = \eta$. The motivation for using $\Delta x = -\nabla f(x)$ lies in the fact that of all the directions available, $-\nabla f(x)$ represents the steepest direction of descent. Hence, the gradient descent is alternatively known as **steepest descent**. The update equation for the gradient descent algorithm can be represented using the following equation :

$$x^+ = x - \eta \nabla f(x) \tag{3.12}$$

In the above equation, the step size is assumed to be constant. Usually, in gradient descent algorithms

3.5.3 Advantages of gradient descent

1) Suitable for higher dimensions

   The gradient descent algorithm works for higher dimensions as well. The only problem may lie in computing the gradient in each dimension since the computations increase linearly with the number of dimensions.

2) Convergence

   The convergence of the gradient descent algorithm is almost always guaranteed with certain basic assumptions and proper values of η. Typically, the convergence can be guaranteed for low values of η but it may take up a lot of computational time depending on the objective function.

3) Extensions

   The gradient descent technique has its analogue in stochastic optimization where stochastic gradient descent is used for minimizing the objective function that is written as a sum of differentiable functions.

3.5.4 Issues in Gradient descent

1) $\nabla f(x)$ does not exist for some x or is difficult to compute.

   The gradient of the function may not exist at certain points . In that case, we can use subgradient descent algorithms which will be covered in upcoming lectures. The gradient maybe difficult to compute as well for certain convex functions. Coordinate descent techniques may be used in that case.

2) Selecting the best step size

   One of the most important parameter to control in gradient descent is the step sizes $\eta > 0$. Very small values of η will cause our algorithm to converge very slowly and thus the computational time required is very large. On the other hand, larger values of η could cause our algorithm to overshoot the minima and may lead to oscillations around the optimum. Thus, there exists a tradeoff in choosing the best value of η between larger time required for computation and oscillations around the minima. This is illustrated in figure **??** where for lower values of step size the function takes a long time to converge whereas for higher values the function oscillates around the optimum. As a result, it becomes very important to pick the best value of η which will deal with this trade-off. This will be mainly covered in the next lecture.

3.5.5 Convergence of Gradient Descent

Although, the importance of step sizes in gradient descent method can be explained intuitively with simple figure examples (as shown in Figure **??**), it is better to have a formal

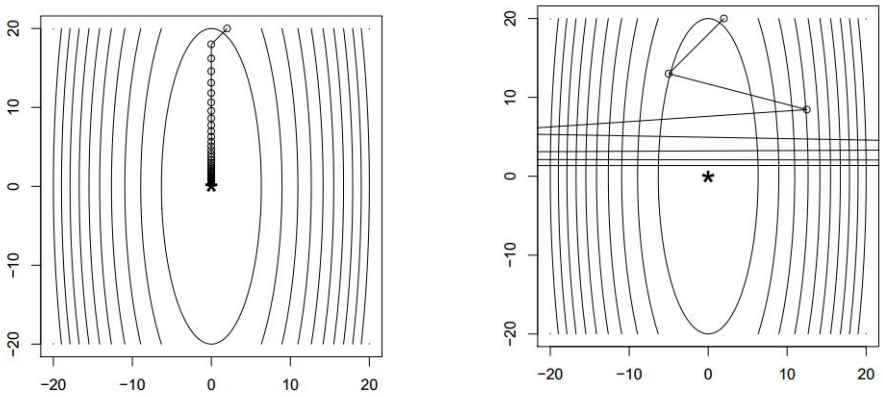

Figure 3.7. (left) a too small step size that leads to slow convergence; (right) a too large step size that leads to oscillations

condition on a step size to guarantee the convergence. There exists a theorem which specifies the condition of a constant step size to have a converging behavior $x^{(k)} \to x^*$ regardless of the initial point. Before presenting the theorem, a useful condition of a function is required to be defined.

Definition 5. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called L-Lipschitz if and only if*

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L||x - y||_2, \qquad \forall x, y \in \mathbb{R}^n \tag{3.13}$$

We denote this condition by $f \in C_L$, where C_L is the class of L-Lipschitz functions.

An intuitive meaning of the Lipschitz condition is how fast the derivative of function f can change, and the Lipschitz constant L is an upper bound of the slope of $\nabla f(x)$. To understand this Lipschitz condition better, consider a simple example of 1-dimensional function which is L-Lipschitz with $L = 1$.

Following function is a special case of *Huber loss function*[2] which is defined as:

$$\psi(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - \frac{1}{2} & \text{if } |x| \geq 1 \end{cases} \tag{3.14}$$

Graph of the *Huber loss function* is plotted in Figure **??**, and it is easy to see that $|\psi'(x) - \psi'(y)| \geq |x - y|$ for $\forall x, y \in \mathbb{R}$. So the function ψ is 1-Lipschitz which can be denoted as $\psi \in C_1$.

Based on this definition of L-Lipschitz function, we can introduce a lemma. It is necessary in proving the theorem about a condition of step size which guarantees the convergence of gradient descent method.

[2]First defined by Huber, Peter J in 1964. *Huber, Peter J. "Robust estimation of a location parameter." The Annals of Mathematical Statistics 35.1 (1964): 73-101*

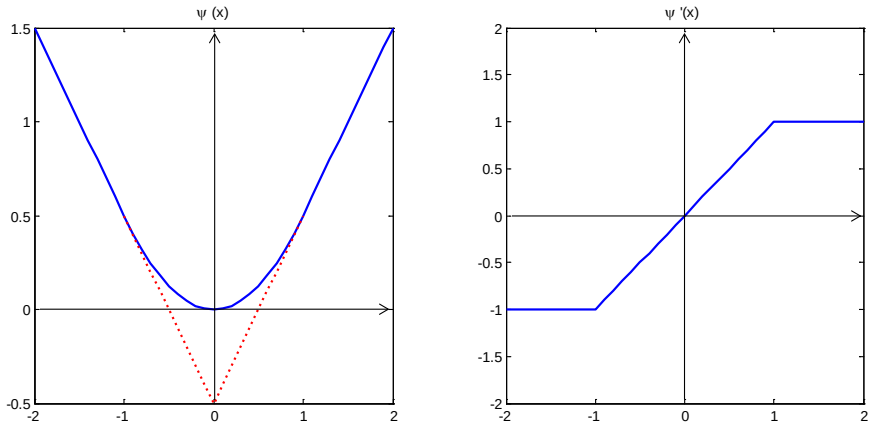

Figure 3.8. Graph of basic *Huber loss function*(left) and its derivative(right)

Lemma 3.1. *If* $f \in C_L$, *then* $|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2}||y - x||^2$

Proof: Let's define a new function g as $g(x) = \frac{L}{2}x^T x - f(x)$. Then g is apparently differentiable and $\nabla g(x) = Lx - \nabla f(x)$.

Claim 1: $(\nabla g(x) - \nabla g(y))^T (x - y) \geq 0 \quad \forall x, y$

This can be shown as follows:

$$\begin{aligned}
&(\nabla g(x) - \nabla g(y))^T (x - y) \\
&\qquad = (Lx - \nabla f(x) - Ly + \nabla f(y))^T (x - y) \\
&\qquad = L\|x - y\|^2 - (\nabla f(x) - \nabla f(y))^T (x - y) \\
&\qquad \geq L\|x - y\|^2 - \|\nabla f(x) - \nabla f(y)\|\|x - y\| \quad (\because \textit{Cauchy} - \textit{Schwartz inequality}) \\
&\qquad \geq L\|x - y\|^2 - L\|x - y\|^2 \quad (\because f \text{ is } L\text{-Lipschitz}) \\
&\qquad = 0
\end{aligned}$$

So, the Claim 1 is proved and let's show that $g(x)$ is convex.

First, define a new function h as $h(t) = g(x + t(y - x))$, and the derivative of h is $h'(t) = \nabla g(x + t(y - x))^T (y - x)$ by Chain Rule. Then for $t > 0$,

$$\begin{aligned}
h'(t) - h'(0) &= \nabla g(x + t(y - x))^T (y - x) - \nabla g(x)^T (y - x) \\
&= \frac{1}{t} (\nabla g(x + t(y - x)) - \nabla g(x))^T t(y - x) \\
&\geq \frac{1}{t} \times 0 = 0 \quad (\because \text{Claim 1})
\end{aligned}$$

Therefore, $h'(t) \geq h'(0)$ for $t \geq 0$.

$$\begin{aligned}
\Rightarrow\ g(y) &= h(1) \\
&= h(0) + \int_0^1 h'(t)dt \\
&\geq h(0) + \int_0^1 h'(0)dt \\
&= g(x) + \nabla g(x)^T(y-x)
\end{aligned}$$

This inequality is an equivalent definition of a convex function, so g is a convex function.

Now, let's rewrite this inequality in f.

$$\begin{aligned}
& g(y) \geq g(x) + \nabla g(x)^T(y-x) \\
\Leftrightarrow\ & \frac{L}{2}y^Ty - f(y) - \frac{L}{2}x^Tx + f(x) \geq (Lx - \nabla f(x))^T(y-x) \\
\Leftrightarrow\ & \frac{L}{2}\|x-y\|^2 \geq f(y) - f(x) + \nabla f(x)^T(y-x)
\end{aligned}$$

Since $\langle a, b\rangle = a^Tb$, this concludes the proof. □

The following Theorem **??** provides a proper condition of step size for gradient descent when the step size is assumed to be fixed as a constant value η.

Theorem 3.2. *If f is L-Lipschitz and $\exists$ an optimum x^*, i.e. $f^* = f(x^*) = \min_x f(x) > -\infty$, then the gradient descent algorithm with fixed step size satisfying $\eta < \frac{2}{L}$ will converge to a stationary point (or x^* when f is convex) from any initial point.*

Proof: Let $x^+ = x - \eta\nabla f(x)$. Using lemma (**??**), we can write

$$\begin{aligned}
f(x^+) &\leq f(x) + \langle \nabla f(x), x^+ - x\rangle + \frac{L}{2}\|x^+ - x\|^2 \\
&= f(x) - \eta\|\nabla f(x)\|^2 + \frac{\eta^2}{2}L\|\nabla f(x)\|^2 \\
&= f(x) - \eta(1 - \frac{\eta}{2}L)\|\nabla f(x)\|^2
\end{aligned}$$

This leads to:

$$\|\nabla f(x)\|^2 \leq \frac{1}{\eta(1-\frac{\eta L}{2})}(f(x) - f(x^+))$$

Now, let's denote $x^{(k)}$ a k^{th} value of x in gradient descent, where $x^{(0)}$ is an initial value of x. Then the above equation can be written as follows for $\forall k \geq 1$.

$$\|\nabla f(x^{(k)})\|^2 \leq \frac{1}{\eta(1-\frac{\eta L}{2})}(f(x^{(k)}) - f(x^{(k+1)}))$$

Summing these equations for $k = 1, \cdots, N$ gives:

$$\begin{aligned}\sum_{k=1}^{N} \|\nabla f(x^{(k)})\|^2 &\leq \frac{1}{\eta(1-\frac{\eta}{2}L)}(f(x^{(0)}) - f(x^{(N)}) \\ &\leq \frac{1}{\eta(1-\frac{\eta}{2}L)}(f(x^{(0)}) - f^*)\end{aligned}$$

The last inequality comes from the fact that f^* is an optimal value. Since RHS of the inequality is a finite value, the summation of N sequences is bounded by finite number. This implies that $\lim_{k\to\infty} \nabla f(x^{(k)}) = 0$. So, when $f(x)$ is a convex function, $x^{(k)} \to x^*$ because the derivative converges to 0. □

3.6 Conclusion

According to the Theorem **??**, convergence of the gradient method can be provided with additional assumptions on a function: L-Lipschitz and convex. Now, the natural next question is how fast this happens. Unfortunately, the Theorem **??** doesn't provide any information about the rate of convergence of $\nabla f(x^{(k)}) \to 0$. To handle this issue, we can use two natural metrics and investigate how fast do the metrics decrease to 0. The metrics are as follows:

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

Some gradient descent methods tend to use fixed step size for simplicity but the choice of appropriate step sizes is not easy. Simple example of a convex function with really flat shape near their minima (Figure **??**) shows a very slow rate of convergence.

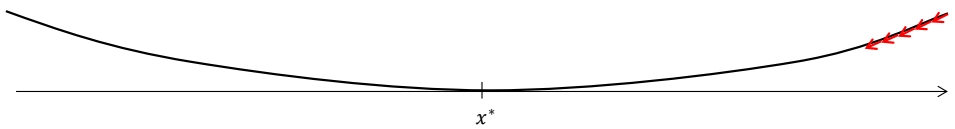

Figure 3.9. Visualize a 'flat' convex function, which is a bad function for constant step size

Figure (**??**) shows level sets of a higher dimensional case. According to the shape of level set, a direction of horizontal axis can have high Lipschitz constant L, and denote as "high L". Similarly, the vertical axis shows "low L" characteristic. As shown in the figure, very small values of η will approach to "low L" axis at certain point and always converges, but

possibly cause the algorithm to converge very slowly. On the other hand, a too large η could cause the algorithm to overshoot the minima and diverge.

So, an additional investigation on the speed of convergence, or the rate of convergence, should be studied to resolve this problem. In the next lecture, an answer to this question will be introduced: we will get bounds on $f(x^{(k)}) - f^*$ and $\|x^{(k)} - x^*\|$ with some additional assumptions on a function and an appropriate number to reveal the rate of convergence.