10-725/36-725: Convex Optimization **Spring 2015**

Lecture 5: Gradient Descent

Lecturer: Ryan Tibshirani *Scribes: Loc Do,2,3*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor. Some of the content in this note are borrowed from the Boyd & Vandenberghe's book.*

This lecture is about Gradient Descent, the first algorithm in a series of first-order methods for solving optimization problem.

5.1 Unconstrained minimization problems and Gradient descent

Let consider unconstrained, smooth convex optimization problems in the form of

$$\min f(x) \tag{5.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, twice differentiable, with $\mathbf{dom}\,(f) = \mathbb{R}$ (no constrains) Assume that the problem is solvable, we denote the optimal value, $f^* = \min_x f(x)$ and optimal solution as $x^* = \arg\min_x f(x)$.

To find x^*, one approach is to solve a system of equation $\nabla f(x^*) = 0$, which is often not easy to solve analytically. Hence, an iterative scheme is more preferred: computing a minimizing sequence of points $x^{(0)}, x^{(1)}, \ldots$ such that $f(x^{(k)}) \to f(x^*)$ as $k \to \infty$. The algorithm stops at some points when the residual error between current state and optimal value is within an acceptable tolerance, i.e. $f(x^{(k)}) - p^* \leq \epsilon$.

Gradient Descent is an iterative algorithm producing such a minimizing sequence $x^{(k)}$ by repeating

$$x^{(k)} = x^{(k-1)} - t_k \nabla f(x^{(k-1)}), \tag{5.2}$$

where $k = 1, 2, \ldots$ is iteration number, t_k is step size (or step length) at iteration k, initial $x_0 \in \mathbb{R}^n$ is usually given. We can prove that $f(x^{(k)}) < f(x^{(k-1)}$ by applying first-order approximation on the LHS as follows

$$f(x^{(k)}) = f(x^{(k-1)} - t_k \nabla f(x^{(k-1)})) \approx f(x^{(k-1)}) - t \nabla f(x^{(k-1)})^T \nabla f(x^{(k-1)}) \leq f(x^{(k-1)}) \tag{5.3}$$

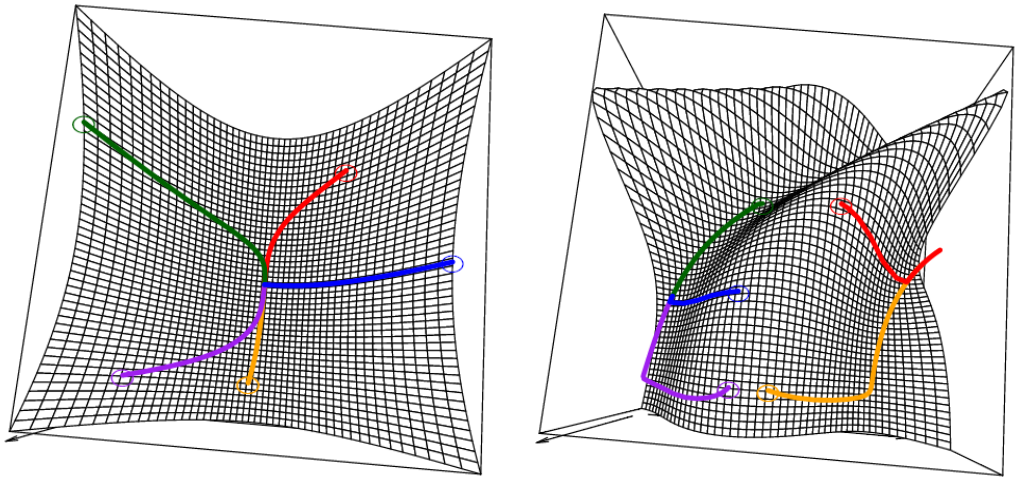
Basically, by following the points generated by Gradient Descent, we are guaranteed to reach closer to the optimal values of the functions. However, it depends on whether the function is convex or non-convex that we can reach global optimum or local optima (illustrated in Figure 5.1).

Interpretation of Gradient Descent. Now we consider a more formal interpretation of Gradient Descent. At each iteration, we want to move from our current point x to point y such that $f(y)$ is optimal. Since f is twice differentiable, apply the quadratic approximation on $f(y)$ to have

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(\theta(x - y) + y)(y - x) \tag{5.4}$$

where $\theta \in [0..1]$. By replacing $\nabla^2 f(\theta(x - y) + y)$ by $\frac{1}{t} I$, we can represent $f(y)$ as follows

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t} ||y - x||_2^2 = g(y) \tag{5.5}$$



(a) Convex function (b) Non-convex function

Figure 5.1: Gradient Descent paths with different starting points are illustrated in different colours. In the case of strictly convex function (lFigure a.), Gradient Descent paths starting from any points all lead to the global optimum. Conversely, in the case of non-convex function, different paths may end up at different local optima.

.

the first additive term is called linear approximation, and the second one is proximity term. Basically, the proximity term tell us that we should not go to far from x, otherwise results in large $f(y)$. To find optimal value of y, we solve the equality $\nabla g(y) = 0 \Leftrightarrow \nabla f(x) + \frac{1}{t}(y-x) = 0 \Leftrightarrow y = x - t\nabla f(x)$. Figure 5.2 illustrates this optimal movement.

5.2 How to choose step sizes

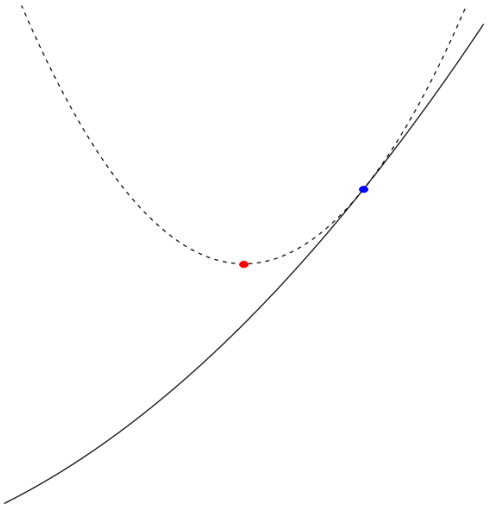
Recall that the update rule of Gradient Descent requires a step size t_k controlling the amount of gradient updated to the current point at each iteration. A naive strategy is to set a constant $t_k = t$ for all iterations. This strategy poses two problems. A too big t can lead to divergence, meaning the learning function oscillates away from the optimal point. A too small t takes longer time for the function to converge. A good selection of t can make the algorithm faster to converge, as illustrated in Figure 5.3. Hence, we need good strategies to select appropriate step sizes. Two examples of such approaches are **backtracking line search** and **exact line search**.

Backtracking line search The main idea of this strategy is to pick step sizes to reduce f approximately. The strategy is described as follows.

- First fix two parameters $0 < \beta < 1$ and $0 < \alpha \leq 0.5$.
- At each iteration, start with t = 1, and while

$$f(x - t\nabla f(x)) > f(x) - \alpha t||\nabla f(x)||_2^2, \tag{5.6}$$

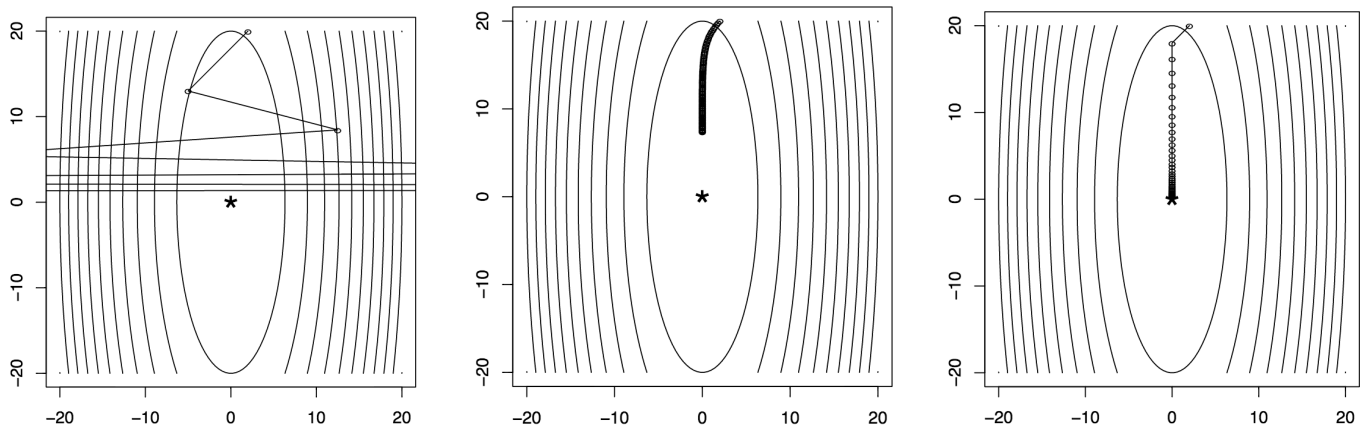
shrink $t = \beta t$. Else perform Gradient Descent update $x^+ = x - t\nabla f(x)$.



Blue point is x, red point is

$$x^+ = \underset{y}{\operatorname{argmin}} \; f(x) + \nabla f(x)^T(y-x) + \frac{1}{2t}\|y-x\|_2^2$$

Figure 5.2: Solid curve depicts f. Dashed curve depicts g, which is second-order approximation of f. Proximity term prevents y having lower values.



(a) t is too large (after 8 steps) (b) t is too small (after 100 steps) (c) t is good (after 40 steps)

Figure 5.3: Consider the function $f(x) = \frac{10x_1^2+x_2^2}{2}$.

.

This approach will eventually terminate. For small enough t, we may have

$$f(x - t\nabla f(x)) \approx f(x) - t||\nabla f(x)||_2^2 < f(x) - \alpha t||\nabla f(x)||_2^2 \tag{5.7}$$

Parameter α controls the decrease in f of the prediction based on the linear approximation. β controls how much exhaustive the search for t is. As suggested in B&V book, $\alpha \in 0.01, 03]$ and $\beta \in [0.1, 0.8]$.

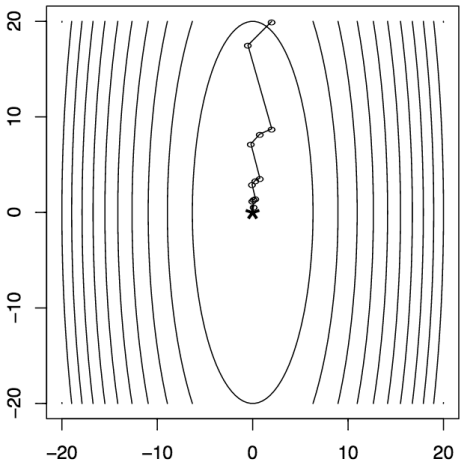


Figure 5.4: Backtracking line search with $\alpha = \beta = 0.5$. We can see the step side is adjusted appropriately after every iteration (large step size at early iterations and keep decreasing when getting closer to the optimal values).

Exact line search Backtrack line search is often known as inexact line search since it selects step size to reduce f approximately. We can also choose step size to do the best as we can along the direction of negative gradient. This strategy is called exact line search.

$$t = \arg\min_{s \geq 0} f(x - s\nabla f(x)) \tag{5.8}$$

Usually Equation 5.8 is not possible to solve exactly. Approximations to the solution are not more efficient than backtracking.

5.3 Convergence analysis

Assume that f is convex, differentiable with **dom (f)** $= \mathbb{R}^n$ and Lipschitz gradient with constant $L > 0$. We have the following theorem.

Theorem 5.1 *Gradient descent with fixed step size $t \leq 1/L$ satisfies*

$$f(x^{(k)}) - f^* \leq \frac{||x^{(0)} - x^*||_2^2}{2tk} \tag{5.9}$$

Proof: Since ∇f is Lipschitz continuous with constant $L > 0$, we have

$$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}||y - x||_2^2, \forall x, y \tag{5.10}$$

(Note: Proof of the inequality in Equation 5.10 is part of homework 1) Assume that y is computed from x using Gradient descent update rule

$$y = x^+ = x - t\nabla f(x) \Leftrightarrow \nabla f(x) = \frac{x - x^+}{t} \tag{5.11}$$

Substitute Equation 5.11 to Equation 5.10, we have

$$\begin{aligned} f(x^+) &\leq f(x) + \nabla f(x)^T(-t\nabla f(x)) + \frac{L}{2}t^2||\nabla f(x)||_2^2 \\ &= f(x) - t||\nabla f(x)||_2^2 + \frac{Lt^2}{2}||\nabla f(x)||_2^2 \\ &= f(x) - (1 - \frac{Lt}{2})t||\nabla f(x)||_2^2 \end{aligned} \tag{5.12}$$

Note that $(1 - \frac{Lt}{2})t \geq \frac{t}{2}$ for $t \in (0, \frac{1}{L}]$, we can bound Equation 5.12

$$f(x^+) \leq f(x) - \frac{t}{2}||\nabla f(x)||_2^2 = f(x) - \frac{t}{2}||\frac{x - x^+}{t}||_2^2 = f(x) - \frac{1}{2t}||x - x^+||_2^2 \tag{5.13}$$

By the convexity of f, we can write

$$f(x^*) \geq f(x) + \nabla f(x)^T(x^* - x) \Leftrightarrow f(x) \leq f(x^*) + \nabla f(x)^T(x - x^*) \tag{5.14}$$

Hence, we can substitute Equation 5.14 to Equation 5.13

$$\begin{aligned} f(x^+) &\leq f(x^*) + \nabla f(x)^T(x - x^*) - \frac{1}{2t}||x^+ - x||_2^2 \\ &= f(x^*) + (\frac{x - x^+}{t})^T(x - x^*) - \frac{1}{2t}||x^+ - x||_2^2 \\ &= f(x^*) + \frac{1}{t}(x - x^+)^T(x - x^*) - \frac{1}{2t}||x^+ - x||_2^2 \\ &= f(x^*) + \frac{1}{2t}(2(x - x^+)^T(x - x^*) - ||x^+ - x||_2^2) \\ &= f(x^*) + \frac{1}{2t}(2(x - x^+)^T(x - x^*) - ||x^+ - x||_2^2 + ||x - x^*||_2^2 - ||x - x^*||_2^2) \\ &= f(x^*) + \frac{1}{2t}(-(||x - x^*||_2^2 - 2(x - x^+)^T(x - x^*) + ||x^+ - x||_2^2) + ||x - x^*||_2^2) \\ &= f(x^*) + \frac{1}{2t}(-||x^+ - x^*||_2^2 + ||x - x^*||_2^2) \end{aligned} \tag{5.15}$$

Move $f(x^*)$ to the LHS, we have

$$f(x^+) - f(x^*) \leq \frac{1}{2t}(-||x^+ - x^*||_2^2 + ||x - x^*||_2^2) \tag{5.16}$$

Summing all the LHS over different k, we have

$$\begin{aligned} \sum_{i=1}^{k} f(x^{(i)}) - f(x^*) &= \frac{1}{2t}\sum_{i=1}^{k}(||x^{(i-1)} - x^*||_2^2 - ||x^{(i)} - x^*||_2^2) \\ &= \frac{1}{2t}(||x^0 - x^*||_2^2 - ||x^{(k)} - x^*||_2^2) \leq \frac{1}{2t}||x^0 - x^*||_2^2 \end{aligned} \tag{5.17}$$

Since $f(x^{(k)}) \leq \ldots \leq f(x^{(1)})$, we have

$$k(f(x^{(k)}) - f(x^*)) \leq \sum_{i=1}^{k} f(x^{(i)}) - f(x^*) \leq \frac{1}{2t}||x^0 - x^*||_2^2 \tag{5.18}$$

Divide both sides to k, we have Theorem 5.1. ■

A couple of things to note from this convergence analysis:

- We say Gradient Descent has convergence rate $O(1/k)$. In other words, to get $f(x^{(k)}) - f^* \leq \epsilon$, we need $O(1/\epsilon)$ iterations.
- Equation 5.12 recalls us the stopping condition in Backtracking line search when $\alpha = 0.5, t = \frac{1}{L}$. Hence, Backtracking line search with $\alpha = 0.5$ plus condition of Lipschitz gradient will guarantee us the convergence rate of $O(1/k)$.

5.3.1 Convergence analysis for Backtracking line search

With the same assumption as above, namely f is convex, differentiable with **dom (f)** $= \mathbb{R}^n$ and Lipschitz gradient with constant $L > 0$. We have the following theorem.

Theorem 5.2 *Gradient descent with backtracking line search satisfies*

$$f(x^{(k)}) - f^* \leq \frac{||x^{(0)} - x^*||_2^2}{2t_{min}k}, \tag{5.19}$$

where $t_{min} = min\{1, \beta/L\}$

It is obvious to see that Backtracking line search arrives at the similar rate with fixed step size. The only difference is the amount of penalty is now controlled by t_{min}, which again depends on β. $\beta = 1$ returns the exact rate of fixed step size. Not too small β would return a comparable result to fixed step size.

5.3.2 Convergence analysis under strong convexity

Recall that a function f is strongly convex for some $m > 0$ if $f(x) - \frac{m}{2}||x||_2^2$ is convex. Using the same assumption as above, Lipschitz gradient, plus strong convexity, we have the following theorem

Theorem 5.3 *Gradient descent with fixed step size* $t \leq \frac{2}{m+L}$ *or with backtracking line search satisfies*

$$f(x^{(k)}) - f^* \leq c^k \frac{L}{2}||x^{(0)} - x^*||_2^2, \tag{5.20}$$

where $0 < c < 1$.

(Note: Proof of this theorem is part of homework 2.)

Theorem 5.3 gives us an exponentially fast converge rate ($O(c^k)$). In other word, we can say that to get $f(x^{(k)} - f^* \leq \epsilon$, we only need $O(log(1/\epsilon))$ iterations. Hence, sometimes this rate is known as linear convergence.

Constant c depends adversely on condition number L/m. Higher condition number will lead to slower rate.

5.3.3 Example of the conditions

Let consider $f(\beta) = \frac{1}{2}||y - X\beta||_2^2$.

Lipschitz continuity of ∇f:

- $\nabla^2 f(x) \preceq LI$
- As $\nabla^2 f(\beta) = X^T X$, we have $L = \sigma^2_{\max}(X)$ (i.e. largest eigen value of $X^T X$)

Strong convexity of f:

- $\nabla^2 f(x) \succeq mI$
- As $\nabla^2 f(\beta) = X^T X$, we have $m = \sigma^2_{\min}(X)$
- If X is wide, X is $n \times p$ with $p > n$(more columns than rows, i.e. more features than observations), $\sigma^2_{\min}(X) = 0$ ($X^T X$ is rank deficient), and f can't be strongly convex.
- If $\sigma_{min}(X) > 0$, we can have very large condition number $L/m = \sigma_{\max}(X)/\sigma_{\min}(X)$

5.3.4 Practicalities

Now we know that Gradient Descent can eventually lead to convergence under some assumptions after a large number of iterations. A good stopping condition in practice is that $||\nabla f(x)||_2$ is small (Recall that $||\nabla f(x)||_2 = 0$ at solution x^*). If f is strongly convex with parameter m, then we can stop when

$$||\nabla f(x)||_2 \leq \sqrt{(2m\epsilon)} \rightarrow f(x) - f^* \leq \epsilon \tag{5.21}$$

Pros and Cons of Gradient Descent:

- Pro: simple idea (iteratively moving toward the steepest descent), each iteration is cheap (need to recompute the gradients)
- Pro: very fast for well-conditioned, strongly convex problems.
- Con: often slow, because interesting problems are not strongly convex or well-conditioned.
- Con: can't handle non-differentiable functions.

5.4 Summary

Gradient Descent belongs to a group of first-order method, basically, algorithms that update $x^{(k)}$ in the following linear space

$$x^{(0)} + \text{span}\{\nabla f(x^{(0)}), \ldots, \nabla f(x^{(k-1)})\} \tag{5.22}$$

The algorithm has $O(1/\epsilon)$ rate over problem class of convex, differentiable functions with Lipschitz continuous gradients.

Theorem 5.4 *For any iteration $k \leq (n-1)/2$ and any starting point $x^{(0)}$, there is a function f in the problem class such that any first-order method satisfies*

$$f(x^{(k)}) - f^* \geq \frac{3L||x^{(0)} - x^*||_2^2}{32(k+1)^2} \tag{5.23}$$

This theorem sets a lower bound on the rate, telling us that we cannot do better than $O(1/k^2)$. However, we can tweak the algorithm to have better rate of $O(1/\sqrt{\epsilon})$ (e.g. using acceleration techniques). Besides, there are other algorithms that allow us to solve more complex functions, i.e. non-differentiable functions.