**Indian Institute of Technology Kanpur**
**CS773 Online Learning and Optimization**

*Scribe:* Pawan Kumar
*Instructor:* Purushottam Kar
*Date:* September 8, 2016

**LECTURE**
# 11

## Optimal Rates for Subgradient Descent

## 1 Introduction

This lecture focuses on finding the optimal rates for subgradient descent and a brief introduction to conjugate gradient method. It presents methods for finding the subgradient of non-differentiable functions, a few properties of subgradient, proving the optimality of subgradient descent. All optimal rates for convex, smooth, strongly convex, smooth as well as strongly convex functions are also given. We try to solve the linear regression/linear system of equations using gradient descent(steepest descent). We also intuitively describe why conjugate gradient works better and reach the optima in less number of steps.

## 2 Min-Max Rate

Best rate of subgradient descent is found by min-max rate. let $\mathcal{A}$ denote the set of all algorithms, $\mathcal{F}$ denote the set of all functions. Then,

$$\epsilon_k(a, f) = f(a, k) = f^*$$

where, $a \in \mathcal{A}$, $f \in \mathcal{F}$, $\epsilon_k(a, f)$ denote how close we are to optima in $k$ steps, $f(a, k) = f^*$ denote the excess error i.e. function value of algorithm $a$ after it has run for $k$ iterations.

We can write the finding the Min-Max rate as:

$$min_{a\in\mathcal{A}} max_{f\in\mathcal{F}} \epsilon_k(a, f) \geq \frac{\left\|X^* - X^0\right\|_2 .G}{\sqrt{k}} = \frac{D_0.G}{\sqrt{k}}$$

It essentially means that for all functions $f \in \mathcal{F}$, find the best algorithm $a \in \mathcal{A}$, such that the above equation holds. Note that the function $f$ can be any function(even the worst of all) and we are only assuming bounded gradient property.

**Remark 11.1.** Instead of $max_{f\in\mathcal{F}}$, If we choose expected value of the function, then as long as the probability distribution is fixed, we can find the min-max rate, otherwise the problem becomes hard.

**Remark 11.2.** Meaning of bounded gradient : Suppose you receive an arbitrary $g^t \in f(x^t)$. For bounded gradient, $\|g\|_2 \leq G$.

**Remark 11.3.** For convex functions, the presence of $\mathbf{0}$ in subgradient is necessary and sufficient condition for optima.

# 3 Objective Function

Consider the function

$$f(x) = (max_{i\in[k+1]}x_i) + \frac{1}{2}\|\mathbf{x}\|_2^2$$

where $i \in [k+1]$ stands for $1 \leq i \leq k+1 i.e.$ consisting of natural numbers and $\mathbf{x} \in \mathbb{R}^n$

For $j > k+1, x_j^* = 0$.

If we want to change the maximum value, then we need to change ever value of $x_i$.

*Observation*: To reduce the value of objective function, we must distribute the values uniformly. For example: If we are given $x_i^{*^2} + x_j^{*^2} = x_i^2 + x_j^2$, then distributing uniformly i.e. $x_i^* = x_j^* = \frac{x_i + x_j}{2}$ will minimize it.

# 4 Finding the subgradient of non-differentiable functions

Consider the function

$$f(x) = max\{g_1(x), g_2(x)\}$$

where $g_i$ are convex and differentiable. The function $f$ looks like as shown in the figure 1 on next page. Clearly, $f$ is non-differentiable.

## 4.1 Calculating subgradient

Wikipedia-Subgradient For all $x$, such that $g_1(x) > g_2(x)$,

$$\partial f(x) = \{\Delta g_1(x)\}$$

Note that it satisfies our convexity property $f(y) \geq f(x) + \langle g^t, y - x\rangle$ $\langle \mathbf{x}, \mathbf{y}\rangle$ Similarly, For all $x$, such that $g_1(x) < g_2(x)$,

$$\partial f(x) = \{\Delta g_2(x)\}$$

Finally, for all $x$ such that $g_1(x) = g_2(x)$,

$$\partial f(x) = \{\Delta g_1(x)\}, \{\Delta g_2(x)\} + \text{convex combination of } \{\Delta g_1(x)\}, \{\Delta g_2(x)\}$$

Therefore,

$$\partial f(x) = \text{convex combination of } \{\Delta g_1(x)\}, \{\Delta g_2(x)\}$$

**Remark 11.4.** If $g_1$ and $g_2$ are non-differentiable themselves, then replace $\Delta g_1$ and $\Delta g_2$ with $\partial g_1(x)$ and $\partial g_2(x)$ respectively. Hence, the expression becomes:

$$\partial f(x) = \text{convex combination of } \{\partial g_1(x), \partial g_2(x)\}$$

The convex combination of two or more points gives rise to convex hull.

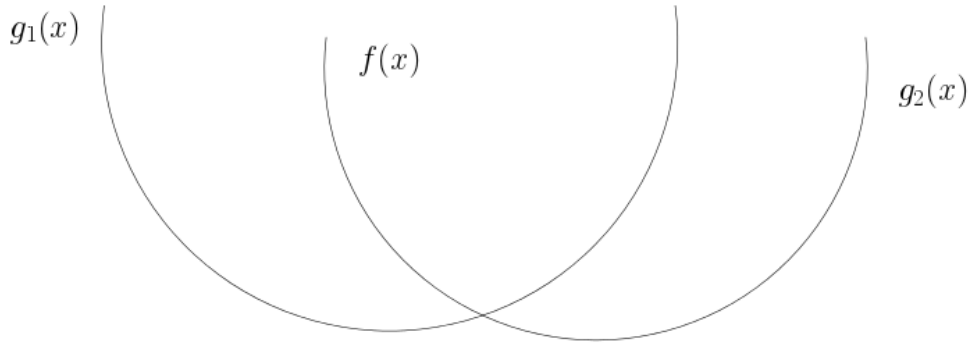


Figure 1: Showing Maximum of two functions.

### 4.2 Danskin's Theorem

Wikipedia-Danskins For our case, we have:

$$f(x) = max_i\{g_i(x)\}$$

Danskin's theorem states that, for any $x_0$, let $I_0 = \arg max_i\{g_i(x_0)\}$, then the subgradient $h$ is given by:

$$g \in \text{convex combination of } \{\cup_{i \in I_0} \partial g_i(x)\}$$

Since calculating this is hard, so we use rather this version of Danskin's theorem:

$$\text{choose } i^* \in I_0 \text{ and choose } h \in \partial g_{i^*}(x_0)$$

This gives the subgradient $h$ which is in $\partial f(x_0)$.

**Remark 11.5.** Subgradients are additive. Hence, if we have

$$g_1 \in \partial f_1(x), g_2 \in \partial f_2(x)$$

Then

$$g_1 + g_2 \in \partial(f_1 + f_2)(x)$$

## 5 Finding the optima of our objective function

Boyd and Vandenberghe (2004) We have

$$f(x) = (max_{i \in [k+1]} x_i) + \frac{1}{2} \|\mathbf{x}\|_2^2$$

where $i \in [k+1]$ stands for $1 \leq i \leq k+1 i.e.$ consisting of natural numbers and $\mathbf{x} \in \mathbb{R}^n$ Also, we have $\|\mathbf{x}\|_2 \leq \frac{1}{\sqrt{k+1}}$ so that the gradient is bounded (call it $G$, which is constant). Since, the objective function is non-differentiable, We want to find the subgradient of it. The subgradient will be the uniform convex combination of subgradients of both terms.

$$\partial f(x) \in \frac{1}{k+1} \sum_{i=1}^{k+1} e_i + \mathbf{x}$$

Algorithm 1: Initialization

1: $\mathbf{x}_0 = \mathbf{0}$ //Start with $\mathbf{0}$, if some other point is the starting point, then we can easily shift by $\frac{1}{2}\|x\|_2^2$

2: $D_0 = \frac{1}{\sqrt{k+1}}$

3: $G = 1 + \frac{1}{\sqrt{k+1}}$ //Adding the bounded gradients for both parts

where $e_i$ are standard unit vectors.

We want to make this subgradient $\mathbf{0}$ to find the optima. Let the value of $\mathbf{x}$ be $\mathbf{x}^*$ at optima. Therefore, $\mathbf{x}^*$ is given by

$$\mathbf{x}^* = (-\frac{1}{k+1}, -\frac{1}{k+1}, \cdots, -\frac{1}{k+1}, 0, 0, \cdots)$$

where first $k+1$ terms are $-\frac{1}{k+1}$ and rest are 0.

$$\mathbf{0} \in \partial f(x^*)$$

Note that $x^*$ is one of the optimum. Now, if we calculate the value of function $f$ at point $x^*$, we get

$$f(x^*) = -\frac{1}{k+1} + \frac{1}{2} \times (-\frac{1}{k+1})$$
$$-\frac{1}{2} \times (\frac{1}{k+1}) = f^*$$

## 5.1 Showing optimality

The initialization is done as shown above.

### 5.1.1 Oracle Part

Oracle part gives $i^t$.

$$i^t = (min_{i \in [k+1]}\{i | x_i^t = max_{i \in [k+1]} x_i^t\}$$

The first co-ordinate which gives the maximum value; Since it wants to stall the algorithm i.e. it wants to prevent the algorithm from knowing which part the maximum is being taken.

$$\text{return}(e_{i^t} + \mathbf{x})(\text{which is a valid subgradient})$$

| Different functions with their optimal rate of convergence | | | |
|---|---|---|---|
| convex | smooth | strongly smooth | smooth and strongly-convex |
| $O(\frac{1}{\epsilon^2})$ | $O(\frac{1}{\epsilon^2})$ | $O(\frac{1}{\epsilon})$ | $O(\log(\frac{1}{\epsilon}))$ |

**Remark 11.6.** The optimal rate of convergence for smooth functions can be improved to $O(\frac{1}{\sqrt{\epsilon}})$ using accelerated gradient method.

Algorithm 2: Algorithmn

1: $\mathbf{x}_0 = \mathbf{0}$ //`Initializing x`
2: Oracle gives the gradient as $g_1 = (e_1)$
3: By span, we get $x^t = \text{span } (x^0, g^1, g^2 \cdots g^{t-1})$
4: $x^1 \in \text{span } (e_1)$
5: Now oracle returns $(e_2 + x)$ as the subgradient.
6: $g_2 = (e_2 + x^1)$
7: $x^1 = \text{span}(e_1, e_2)$
8: At any step $k$, only $(k-1)$ co-ordinates can be non-zero.
9: $x^k = \text{span}(e_1, e_2, \cdots e_k)$
10: $x^k_{k+1} = 0$ //`value of first` $k$ `co-ordinates has be greater than the` $(k+1)^{th}$ `co-ordinate, which is` $0$
11: $f(x^k) \geq 0$
12: $\Rightarrow f(x^k) - f(x^*) \geq \frac{1}{2(k+1)} \geq \frac{G.D_0}{\sqrt{(k+1)}}$

## 6 Linear regression/Linear system of equations

Consider the objective function:

$$min_w \|Y - x\mathbf{w}\|^2 = min_w(\|Y\|^2 + w^T x^T X w - 2Y^T x w)$$

Since it is a differentiable function, We can differentiate with respect to $w$ and equate it to $\mathbf{0}$ to get:

$$x^T x w - Y^T X = 0$$

Now, $x^T x$ is a symmetric and positive semi-definite matrix.

Whereas, in linear system of equations, we have: $Ax = b \Rightarrow Ax - b = 0$, where $A$ is a any rectangular matrix.

Hence, we have:

| Linear Regression/Linear system of equations | |
|---|---|
| $min_w \frac{1}{2} x^T A x - b^T x$<br>Consider this as $f(x)$ | $Ax - b = 0$<br>this is the linear system of equations |

**Remark 11.7.** If we take the gradient of function $f(x)$ and equate it to $\mathbf{0}$, then we shall get the equation $Ax - b = 0$.

We shall look at the cases in which $A$ is square, symmetric, and positive definite matrix. Let $A$ be $\mathbb{R}^{n \times n}$ matrix.

The given function is:

$$f(x^{t+1}) = \frac{1}{2}(x^{t+1})^T A x^{t+1} - b^T (x^{t+1})$$

If we apply chain rule to differentiate

$$f^{'}(x^{t+1})^T . \frac{\partial x^{t+1}}{\partial \alpha} = \mathbf{0}$$

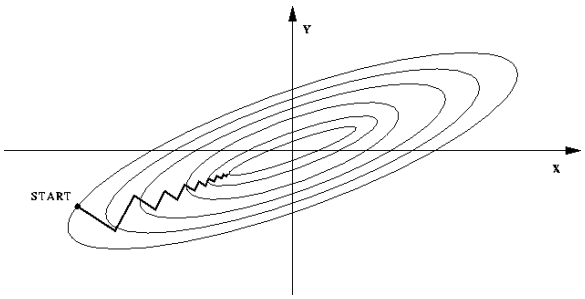


Figure 2: Working of Steepest descent algorithm. Notice the stair-case like behavior. Image Courtesy: `http://trond.hjorteland.com/thesis/img208.gif`

Algorithm 3: Gradient Descent/Steepest Descent algorithm

1: $r^t = b - Ax^T$ //`Residual value`
2: $r^t = -\Delta f(x^t)$
3: $x^t + \alpha_t . r^t = x^{t+1}$
4: $x^1 \in$ span $(e_1)$
5: We want to find $\alpha_t$
6: $f(x^{t+1}) = \frac{1}{2}(x^{t+1})^T A x^{t+1} - b^T(x^{t+1})$
7: $\Rightarrow f(x^{t+1}) = \frac{1}{2}(x^t + \alpha r^t)^T A(x^t + \alpha r^t) - b(x^t + \alpha r^t)$
8: The function is $\frac{1}{2}(r^t)^T A r^t + \alpha (r^t)^T A x^T - \alpha b^T r^t + O(1)$
9: Differentiate with respect to $\alpha$ and equate to $\mathbf{0}$
10: $\alpha((r^t)^T A r^t) = (b - Ax^t)$, which in turn is $\left\|r^t\right\|_2^2$
11: $\Rightarrow \alpha = \frac{\left\|r^t\right\|_2^2}{(r^t)^T A r^t}$
12: therefore, $\alpha$ can be found out from here.

$$\Rightarrow \left\langle f^{'}(x^{t+1}), r^{t+1} \right\rangle = \mathbf{0}$$

The behavior is like a stairway.

**Remark 11.8.** The $\alpha$ found using above method is sadly not optimal.

**Remark 11.9.** If $A$ is identity matrix, then we reach optima in one step (As Shown in figure 3).

# 7 Congugate gradient method

Shewchuk (1994) Kar and Ullah (2016) In this section, we shall introduce conjugate gradient method briefly. The basic idea is: After the first step, don't trust the gradient. Conjugate gradient method reaches the optima in $d$ steps in $d$-dimensions. Moreover, if the number of distinct eigen values are $k$, then it reaches the optima in $k$ steps. See figure 4 for a comparison between steepest descent and conjugate gradient on next page.

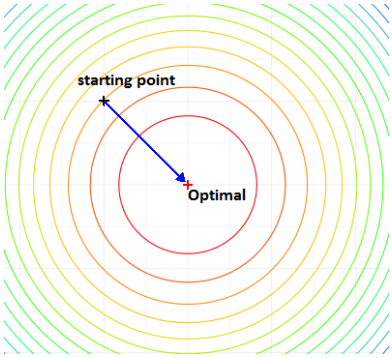


Figure 3: If $A$ is identity matrix, then we reach optima in one step. Image Courtesy: `https://blog.codecentric.de/files/2015/04/Simple.png`

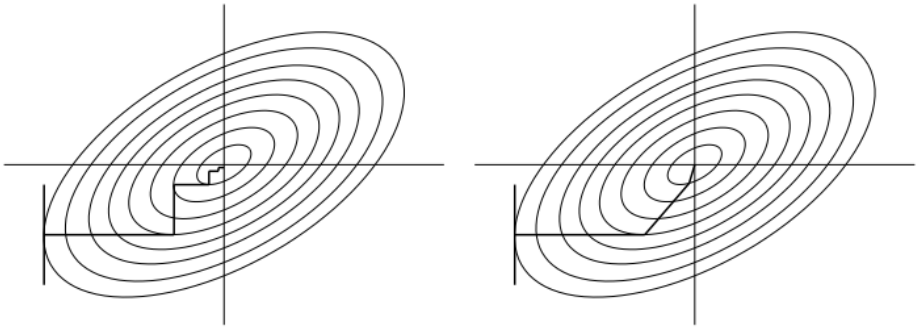

Figure 6.14: Steepest descent vs. conjugate gradient.

Figure 4: A comparison between steepest descent and conjugate gradient. Image Courtesy: `http://i.stack.imgur.com/zh1HH.png`

## References


Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Purushottam Kar and Md Enayat Ullah. Conjugate Gradient Method. pages 1–7, 2016. URL `http://web.cse.iitk.ac.in/users/purushot/courses/opt/2016-17-a/material/scribes/lec12.pdf`.

Jonathan Richard Shewchuk. An introduction to painless conjugate gradient method without agonizing pain. *School of Computer Science, Carnegie Mellon University, Pittsburgh*, pages 6–22, 1994.

Wikipedia-Danskins. Danskin's Theorem. URL `https://en.wikipedia.org/wiki/Danskin's_theorem`.

Wikipedia-Subgradient. Subderivative. URL `https://en.wikipedia.org/wiki/Subderivative`.