Subgradient Method

Indian Institute of Technology, Hyderabad
ai20btech11006@iith.ac.in

April 28, 2022

Abstract

Non-differentiable functions are an important class of functions which often appear in optimization problems, a gradient descent method would fail to optimize such an objective because the gradient might not exist at all points that the method encounters. Methods such as interior point method work great, but it has its own limitations of being computationally inefficient. We will explore the subgradient method which is an iterative first-order method similar to gradient descent.

I. Introduction

Subgradient method is a simple algorithm used to minimize non-differentiable convex functions. This method is similar to vanilla gradient method which is used to optimize differentiable functions.

i. Algorithm

Lets say we have a nondifferentiable function $f : \mathbb{R}^n \to \mathbb{R}$. The update rule of subgradient method says

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \alpha_k \underline{g}^{(k)} \tag{1}$$

where $\underline{x}^{(k)}$ is the k^{th} iterate, α_k is the step step size at k^{th} iteration and $\underline{g}^{(k)}$ is any subgradient of f at $\underline{x}^{(k)}$. The subgradient $\underline{g}^{(k)}$ is any vector which satisfies

$$f(\underline{y}) \geq f(\underline{x}) + \underline{g}^T(\underline{y} - \underline{x}) \tag{2}$$

At a given pozint there can be more than one subgradients, we call the set of subgradients as subdifferential.

Theorem I.1. *If the function f is differentiable at $\underline{x}^{(k)}$ then $\underline{g}^{(k)}$ is equal to the gradient of f at $\underline{x}^{(k)}$*

Proof. Substitute $\underline{y} = \underline{x} + \lambda \underline{z},\ \lambda > 0$ in (2)

$$\frac{f(\underline{x} + \lambda \underline{z}) - f(\underline{x})}{\lambda} \geq \underline{g}^T \underline{z} \tag{3}$$

We can use the limit $\lambda \to 0$

$$\underline{\nabla} f(\underline{x})^T \underline{z} \geq \underline{g}^T \underline{z} \tag{4}$$

$$\underline{z}^T \left(\underline{\nabla} f(\underline{x}) - \underline{g} \right) \geq 0 \quad \forall \underline{z} \tag{5}$$

$$\therefore \underline{g} = \underline{\nabla} f(\underline{x}) \tag{6}$$

□

II. Example

Consider the following problem

$$min \quad max\{f_1, f_2, \ldots f_n\} \tag{7}$$

This problem is non differentiable but it can be solved using subgradient method. Lets consider a numerical example.

III. Convergence proof

Lets assume x^* is the minimizer of our objective function f. Assume that the norm of subgradients is bounded.

Using Lipschitz condition

$$|f(u) - f(v)| \leq G\|u - v\|_2 \tag{8}$$

for all u, v. Some versions of subgradient method work even when the gradient is not bounded.

IV. SVM using subgradient method

A support vector machine is used for two class classification. The objective is to maximize the slab thickness while still satisfying few constraints. A hard-margin SVM can be formuated as

$$\min_{\underline{w},b} \ \underline{w}^T\underline{w} \tag{9}$$

$$\text{s.t: } y_i(\underline{w}^T\underline{x_i} + b) \geq 1 \tag{10}$$

$$\tag{11}$$

We can transform this problem into

$$\min_{\underline{w},b} \ \underline{w}^T\underline{w} + \lambda \sum_i \max(0, 1 - y_i \left(\underline{w}^T\underline{x} + b\right) \tag{12}$$

Where λ is the trade off factor, the higher is the value of this parameter, the higher is the penalty of voilating the given constraints. This problem is essentially a soft-margin SVM. We can now solve this problem by subgradient method.

The first step is to calculate the subgradients,

$$\underline{g_{\underline{w}}} = 2 * \underline{w} + \lambda \sum_i y_i * \underline{x_i} \ \ for\, y_i \left(\underline{w}^T\underline{x} + b\right) < 1 \tag{13}$$

and

$$g_b = \lambda \sum_i y_i \ \ for\, y_i \left(\underline{w}^T\underline{x} + b\right) < 1 \tag{14}$$

In practice, we tend to use mini-batch subgradient method because of its several advantages such as computational efficiency, stable convergence and faster learning.

V. Results

i. Maximum of convex functions

I considered a problem of the type maximum of linear functions and compared different step size rules. The results obtained are as follows

1. Constant Step Size

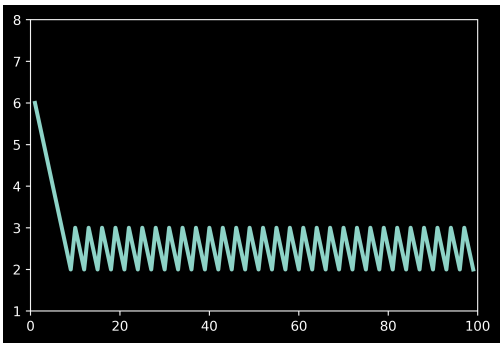

Figure 1: *Constant Step Size*

2. Constant Step Length

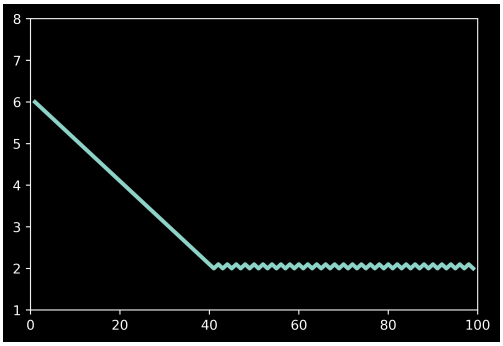

Figure 2: *Constant Step Length*

3. Square Summable But Not Summable

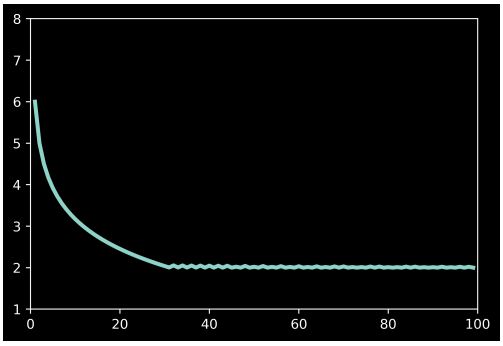

Figure 3: *Square Summable But Not Summable*

4. Non Summable Diminishing

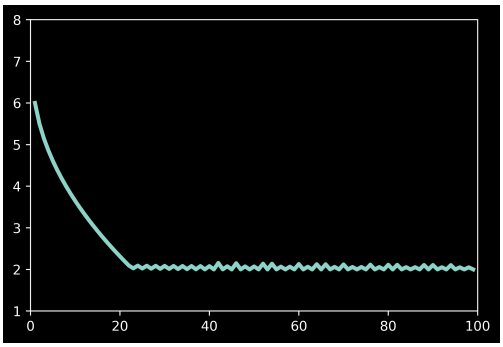

Figure 4: *Non summable diminishing*

VI. Preliminary References

1. Boyd, Stephen and Xiao, Lin and Mutapcic, Almir. Subgradient method [1]

2. Boyd, Stephen and Mutapcic, Almir. stochastic Subgradient method [2]