DIDACTICA MATHEMATICA, Vol. 42 (2024) , pp. 77–86

APPLICATIONS OF THE GRADIENT DESCENT ALGORITHM IN OPTIMIZATION

Serghie Lucas

Abstract. This paper explores the applications of the Gradient Descent Algorithm in various optimization problems. First, we present different variants of the Gradient Descent Algorithm. The effectiveness of these methods is then demonstrated through several practical applications, showcasing their utility in solving complex problems. The results highlight the algorithm's performance in different contexts, offering insights into its implementation and potential areas for further research.

MSC 2000. 65K05.
Key words. Mathematical optimization, Gradient Descent, Lipschitz function, Smooth function.

1. GRADIENT DESCENT METHODS

1.1. Classical Method. In this section, we will analyze the well-known gradient descent method and study its convergent behavior. Similar to other optimization problems, we will aim to solve

$$\min_{x \in S} f(x).$$

In order to solve this problem we will need to take a closer look at the objective function $f : S \to \mathbb{R}$. Functions are typically abstract and don't provide much information on how to be approached.

Definition 1. For any differentiable convex function f with starting point x_1, the basic gradient descent method is defined as follows

$$x_{i+1} = x_i - \eta_i \nabla f(x_i) \qquad i = 1, 2, ...$$

The symbol η represents the step size, more often referred to as learning rate, which can vary with respect to i.

Choosing a good step size is an important part of the gradient descent algorithm since it assures convergence. There are multiple methods for choosing step sizes, each having advantages and disadvantages in various scenarios. We will dive deeper into some of these methods and see how

they ensure convergence, but it is important to note that the context of the problem has a great effect on the choice of the appropriate method therefore the methods presented below might not be the best choice in every case.

1.2. Gradient Descent applied to Lipschitz Functions. This subsection explores the application of Gradient Descent to Lipschitz functions, highlighting the importance of their properties in ensuring a stable and efficient optimization process.

THEOREM 1. Let us consider $f : S \to \mathbb{R}$ a differentiable, Lipschitz and convex over the domain function, with $D \geq ||x_1 - x^*||_2$ the upper-bound of the distance between the initial point and the optimal point. Let us compute n steps of the projected gradient descend with $x_1, \ldots, x_n$ the computed values and a step size of $\eta = \frac{D\beta}{\sqrt{n}}$. From this, we get that

$$f(\frac{1}{n}\sum_{i=1}^{n} x_i) - f(x^*) \leq \frac{D\beta}{\sqrt{n}}$$

Proof. We begin our proof by bounding the function values $f(x_i) - f(x^*)$

$$f(x_i) - f(x^*) \leq \nabla f(x_i)^T (x_i - x^*)$$

We will work only with the right-hand side which by the update rule is $= \frac{1}{\eta}(x_i - y_{i+1})^T(x_i - x^*)$, we can break it into $\frac{1}{2\eta}(||x_i - x^*||^2 + ||x_i - y_{i+1}||^2 - ||y_{i+1} - x^*||^2) = \frac{1}{2\eta}(||x_i - x^*||^2 - ||y_{i+1} - x^*||^2) + \frac{\eta}{2}||\nabla f(x_i)||^2$. By the Lipschitz condition $\leq \frac{1}{2\eta}(||x_i - x^*||^2 - ||y_{i+1} - x^*||^2) + \frac{\eta\beta^2}{2} \leq \frac{1}{2\eta}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{\eta\beta^2}{2}$

Now we apply the sum to both sides

$$\sum_{i=1}^{n} f(x_i) - f(x^*) \leq \frac{1}{2\eta}\sum_{i=1}^{n}(||x_i - x^*||^2 - ||x_{i+1} - x^*||^2) + \frac{n\eta\beta^2}{2}$$

By opening the sum and simplifying the repeating terms in the right-hand side $= \frac{1}{2\eta}(||x_1 - x^*||^2 - ||x_n - x^*||^2) + \frac{n\eta\beta^2}{2} \leq \frac{1}{2\eta}||x_1 - x^*||^2 + \frac{n\eta\beta^2}{2}$.

By convexity, we can also say

$$f(\frac{1}{n}\sum_{i=1}^{n} x_i) - f(x^*) \leq \frac{1}{n}\sum_{i=1}^{n} f(x_i) - f(x^*)$$

Therefore we have

$$f(\frac{1}{n}\sum_{i=1}^{n} x_i) - f(x^*) \leq \frac{D^2}{2n\eta} + \frac{n\eta\beta^2}{2}$$

And finally by setting $\eta = D/\beta\sqrt{n}$

$$f(\frac{1}{n}\sum_{i=1}^{n} x_i) - f(x^*) \leq \frac{D\beta}{\sqrt{n}}$$

□

This shows that when applied to Lipschitz functions, Gradient Descent will guarantee that the function value of the average of points computed will approximate the function value of the optimal point with a maximum error of $\frac{D\beta}{\sqrt{n}}$

1.3. Gradient Descent applied to Smooth Functions. This subsection explores the application of Gradient Descent to Smooth functions. Building on the previous Lipschitz subsection, the properties of smooth functions provide more insight and control over the convergence of the algorithm.

THEOREM 2. Let f be a $\beta-$*smooth* function on $S \subseteq \mathbb{R}^n$, then $\forall x, y \in S$

$$|f(y) - f(x) - \nabla f(x)^T(y-x)| \leq \frac{\beta}{2}||y-x||^2$$

We begin by taking a deeper look at theorem 2. This theorem is very important since it allows us to choose any value for y. By combining this theorem with the iterative rule of the gradient descent algorithm we can choose $y = x - \frac{1}{\beta}\nabla f(x)$, which proves the fact that the update step decreases the function proportional to the squared norm of the gradient, which essentially means that the update step regulates itself as the function approaches the minimum.

$$f(y) - f(x) \leq -\frac{1}{2\beta}||\nabla f(x)||^2$$

Another useful property that $\beta - smooth$ functions bring is that we can bound the difference between $f(x) - f(x^*)$ by two quadratics.

THEOREM 3. Let $f : S \to \mathbb{R}$, where $S \subseteq \mathbb{R}^n$ be $\beta - smooth$, $\forall x \in S$

$$\frac{1}{2\beta}||\nabla f(x)||^2 \leq f(x) - f(x^*) \leq \frac{\beta}{2}||x - x^*||^2$$

Proof. $f(x) \leq f(x^*) + \nabla f(x)^T(x - x^*) + \frac{\beta}{2}||x - x^*||^2$. By $\nabla f(x^*) = 0$ we are left with $f(x) - f(x^*) \leq \frac{\beta}{2}||x - x^*||^2$ □

THEOREM 4. (Co-coercivity)
Let function $f : S \to \mathbb{R}$, where $S \subseteq \mathbb{R}^n$ to be $\beta - smooth$, $\forall x, y \in S$ we have

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{1}{\beta}||\nabla f(x) - \nabla f(y)||^2$$

Proof. We define the function $f_x(z) = f(z) - \langle f(x), z \rangle$ and similarly $f_y(z) = f(z) - \langle f(y), z \rangle$. In order to minimize these functions we apply the gradient with respect to z, and get (in the first equation): $\nabla f_x(z) = \nabla f(z) - \frac{\partial}{\partial z_i}\langle f(x), z \rangle$. By opening the dot product we get the following: $\nabla f_x(z) = \nabla f(z) - \frac{\partial}{\partial z_i}\sum_{i=1}^{i=n}(f(x)_i + z_i)$. Since $f(x)$ is constant when derivating with respect to z we get that the minimum of f is attained when: $\nabla f_x(z) = \nabla f(z) - \nabla f(x) = 0$ that is $x = z$. Equivalently, for the second equation, we get that the minimum of the function $f_y(z)$ is attained when $y = z$.

By taking the left-hand side we can break it down into [**?**]

$$\langle \nabla f(x) - \nabla f(y), y - x \rangle = -\langle \nabla f(x), x - y \rangle - \langle \nabla f(y), x - y \rangle$$

We then add $f(y) and - f(x)$ to both terms

$$= (f(y) - f(x) - \langle \nabla f(x), x - y \rangle) - (f(y) - f(x) - \langle \nabla f(y), x - y \rangle)$$

By taking each term separately we can rewrite them using $f_x(z)$ and $f_y(z)$

(In the case of the first term):

$$f(y) - f(x) - \langle \nabla f(x), x - y \rangle = f(y) - \langle \nabla f(x), y \rangle - (f(x) - \langle \nabla f(x), x \rangle)$$

$$= f_x(y) - f_x(x)$$

Since $f_x(x)$ is the optimal value of f we can use the left inequality of 3 to get

$$f_x(y) - f_x(x) \geq \frac{1}{2\beta} ||\nabla f_x(y)||^2$$

Which by $||a - b|| = ||b - a||$ can be rewritten as

$$f_x(y) - f_x(x) \geq \frac{1}{2\beta} ||\nabla f(y) - \nabla f(x)||^2$$

Similarly, we can apply the same operations on the second term and get:

$f_y(x) - f_y(y) \geq \frac{1}{2\beta} ||\nabla f(x) - \nabla f(y)||^2$

By adding the two terms we finally get:

$$\langle \nabla f(x) - \nabla f(y), y - x \rangle \geq \frac{1}{\beta} ||\nabla f(x) - \nabla f(y)||^2$$

□

THEOREM 5. Let $f : S \to \mathbb{R}$, where $S \subseteq \mathbb{R}^n$ be a $\beta - smooth$ convex function. By applying gradient descent with step size $\eta = \frac{1}{\beta}$ we have that

$$f(x_i) - f(x^*) \leq \frac{2\beta ||x_1 - x^*||^2}{i - 1}$$

Proof. First, we need to prove that $||x_s - x^*||$ is decreasing with s.

$$||x_{s+1} - x^*||^2 = ||x_s - \frac{1}{\beta} \nabla f(x_s) - x^*||^2 = \langle x_s - \frac{1}{\beta} \nabla f(x_s) - x^*, x_s - \frac{1}{\beta} \nabla f(x_s) - x^* \rangle$$

$$= ||x_s - x^*||^2 - \langle x_s - x^*, \frac{1}{\beta} \nabla f(x_s) \rangle - \langle \frac{1}{\beta} \nabla f(x_s), (x_s - x^*) \frac{1}{\beta} \nabla f(x_s) \rangle$$

$$= ||x_s - x^*||^2 - \frac{2}{\beta} \nabla f(x_s)^T (x_s - x^*) + \frac{1}{\beta^2} ||\nabla f(x_s)||^2$$

By 4 and $\nabla f(x^*) = 0$ we have that $\nabla f(x_s)^T(x_s - x^*) \geq \frac{1}{\beta}||\nabla f(x_s)||^2$

$$\leq ||x_s - x^*||^2 - \frac{1}{\beta^2}||\nabla f(x_s)||^2$$

And since $\frac{1}{\beta^2}||\nabla f(x_s)||^2 > 0$

$$||x_{s+1} - x^*||^2 \leq ||x_s - x^*||^2$$

We will use the above facts to finish the proof. By **??** we have

$$f(x_{i+1}) - f(x_i) \leq -\frac{1}{2\beta}||\nabla f(x_i)||^2$$

By moving the $f(x_i)$ to the other side and denoting $\delta_i = f(x_i) - f(x^*)$ we get

$$\delta_{i+1} \leq \delta_i - \frac{1}{2\beta}||\nabla f(x_i)||^2$$

We will use the following convexity property $f(x^*) \geq f(x_s) + \nabla f(\mathbf{x_s})^T(x^* - x_s)$. By rearranging and applying the norm we get $||f(x_s) - f(x^*)|| \leq |\nabla f(\mathbf{x_s})^T(x_s - x^*)|$. We can apply the Cauchy-Schwarz inequality on the modulus to get $||f(x_s) - f(x^*)|| \leq ||\nabla f(\mathbf{x_s})||\ ||(x_s - x^*)||$ which is equivalent to $\delta_s \leq ||\nabla f(\mathbf{x_s})||\ ||(x_s - x^*)||$. By dividing $||(x_s - x^*)||$ and squaring we get $\frac{1}{||(x_1 - x^*)||^2}\delta_s^2 \leq ||\nabla f(\mathbf{x_s})||^2$ (since $||(x_1 - x^*)|| \geq ||(x_s - x^*)||$)

Now we can replace this in the previous equation and get

$$\delta_{i+1} \leq \delta_i - \frac{1}{2\beta||(x_1 - x^*)||^2}\delta_s^2$$

We denote $w = \frac{1}{2\beta||(x_1 - x^*)||^2}$ and get

$$\delta_s \geq w\delta_s^2 + \delta_{s+1} \iff \frac{1}{\delta_{s+1}} \geq w\frac{\delta_s}{\delta_{s+1}} + \frac{1}{\delta_s}$$

Since $\delta_s > \delta_{s+1}$

$$\implies \frac{1}{\delta_{s+1}} - \frac{1}{\delta_s} \geq w$$

By applying $\sum_{s=1}^{s=n-1}$ and using telescopic cancellation we get

$$\frac{1}{\delta_n} \geq \frac{1}{\delta_n} - \frac{1}{\delta_1} \geq w(n-1)$$

□

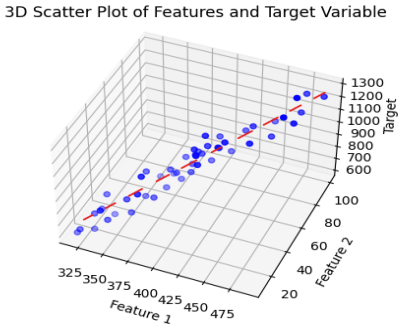
This shows that the gradient descent algorithm reaches a faster rate of convergence when applied to $\beta - smooth$ functions.

2. APPLICATIONS

The source code for the applications can be found here:
https://github.com/LucasSerghie/Gradient-Descent—Optimization-Analysis.

This section will explore several real-world examples and applications of Gradient Descent. The following material presents an application focused on comparing the use of different step-size strategies for Gradient Descent, representing visually and intuitively the content of the past sections. The aim is to explore the benefits of applying the algorithm to functions with $\beta - smooth$ properties and using the Lipschitz constant for computing the step size. In order to demonstrate this, we will compute three different step sizes, each with its own advantages and disadvantages.

As for training sets, we will choose two distinct sets, one artificially generated and one containing real data representing the admission rate for various universities. Both of our training sets are linear, such that gradient descent can be successfully applied to them.

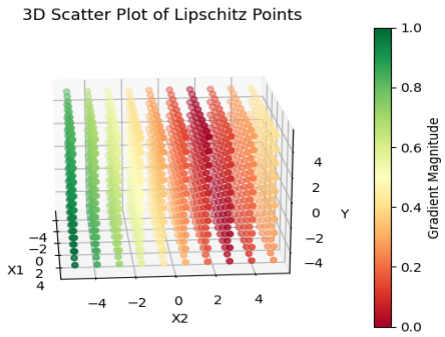


Now that our training data is prepared, we can apply the algorithm to it. When it comes to computing the Lipschitz constant we have to compute the maximum variation in gradient over the distance between

the two corresponding points. This is very hard to compute for all the points in a function, so my proposal was to create a "grid" of a limited number of points for which we would compute the gradient, then compute for each point in the grid the formula with every other point in the grid, and store the maximum value, which would closely approximate the Lipschitz constant.

```
def compute_lipschitz_constant(m, X, y, a, b, num_points=10):
    x1_vals = np.linspace(a, b, num_points)
    x2_vals = np.linspace(a, b, num_points)
    y_vals = np.linspace(a, b, num_points)
    X1, X2, Y = np.meshgrid(x1_vals, x2_vals, y_vals)
    points = np.vstack([X1.ravel(), X2.ravel(), Y.ravel()]).T

    errors_matrix = np.apply_along_axis(lambda point: predict(X, point) - y, 1, points)
    gradients = np.apply_along_axis(lambda err: gradientMSE(m, X, err), 1, errors_matrix)

    grad_diff_matrix = np.linalg.norm(gradients[:, np.newaxis, :] - gradients[np.newaxis, :, :], axis=-1)
    point_diff_matrix = np.linalg.norm(points[:, np.newaxis, :] - points[np.newaxis, :, :], axis=-1)

    point_diff_matrix[point_diff_matrix == 0] = np.inf

    lipschitz_matrix = grad_diff_matrix / point_diff_matrix
    max_lipschitz = np.max(lipschitz_matrix)

    return max_lipschitz, points, gradients
```

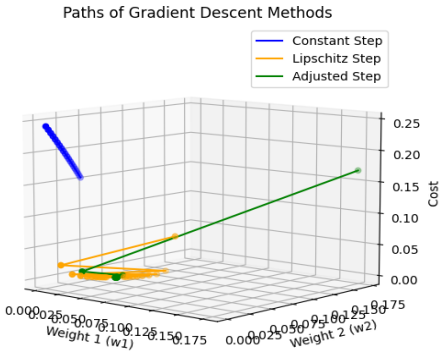
This approximation can be easily understood by the graph below.



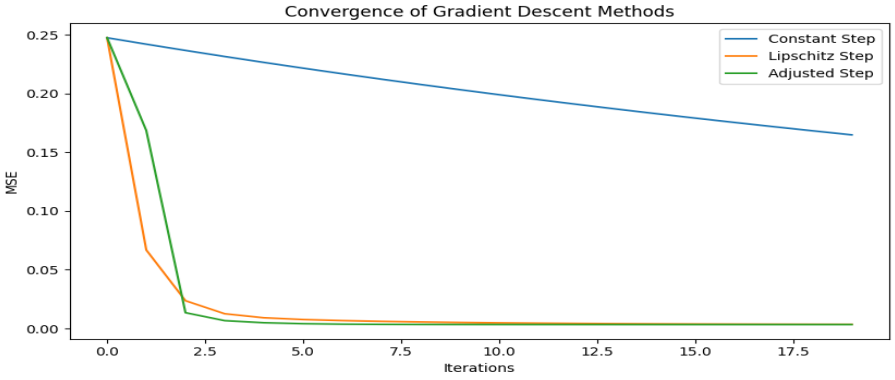
As an alternative to the Lipschitz and classical constant step sizes, we will also compute an updated step size computed with the help of a backtracking function.

```
def backtracking_line_search(X, W, y, grad, alpha, beta=0.5, sigma=0.1):
    while MSE(X.dot(W - alpha * grad) - y) > MSE(X.dot(W) - y) - sigma * alpha * np.linalg.norm(grad)**2:
        alpha *= beta
    return alpha
```

After computing all three step sizes we can complete the algorithm and compare the effectiveness of each.
Not surprisingly, the results are similar in both the generated training data and the real world data when our three different steps are applied.



In the above graph, we can notice the increased effectiveness of the Lipschitz and updated steps compared to the constant one which does not even converge since it is limited by the number of iterations the algorithm is allowed to perform.



Finally, the last graph represents the rate of convergence for each separate step, which is highlighted by color. We can clearly see the benefits of using the Lipschitz properties of the function when choosing a step size against any regular constant step size, as well as the faster convergence of the adjusted step size as compared to the arbitrary constant one. In conclusion, the results we got demonstrated an impact on the effectiveness of the linear regression algorithm highlighting the importance of step size in the optimization process.

REFERENCES

[1] Busenberg, S.N., Fischer, D.C. and Martelli, M., *Better Bounds for Periodic Solutions of Differential Equations in Banach Spaces*, Proc. Amer. Math. Soc., **98** (1986), 376–378.

[2] Cecconi, J., *La Disuguaglianza di Cavalieri per la k Area Secondo Lebesgue in un n Spazio*, Ann. Mat. Pura Appl., **4** (1956), 189–204.

[3] Cesari, L., *Surface area*, Annals of Mathematics Studies 35, Princeton Univ. Press, 1956.

[4] D'Ambrosio, U., *La disuguaglianza di Cavalieri per superficie del E_N definite sopra una varietà bi-dimensionale*, Bol. Soc. Mat. S. Paulo, **16** (1965), 105–113.

[5] Evans, L. and Gariepy, R., *Measure theory and fine properties of functions*, Studies in Advanced Mathematics, CRC Press, 1992.

[6] Hale, J.K., *Ordinary differential equations*, 2nd. Ed, Krieger, 1980.

[7] Lang, S., *Differential manifolds*, Addison-Wesley, 1972.

[8] Lasota, A. and Yorke, J., *Bounds for Periodic Solutions of Differential Equations in Banach Spaces*, J. Differential Equations, **10** (1971), 83–91.

Student at Babes Bolyai Cluj-Napoca, Romania

e-mail: `serghie.lucas@yahoo.com`