The Gradient Descent Algorithm

George Michailidis

gmichail@ucla.edu

STAT 102B

Key take home message from Lecture 1.1

To solve the problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1}$$

we require:

1. Solve the gradient equation

$$\nabla f(x) = 0 \tag{2}$$

2. For the solution $x^\star$ of (2) evaluate the Hessian matrix of $f(x)$ at $x^\star$; i.e.,
   - if $H(x^\star) \succ 0$, then $x^\star$ is a minimum
   - if $H(x^\star) \prec 0$, then $x^\star$ is a maximum
   - otherwise, $x^\star$ is a saddle point

Remarks

- We focus on unconstrained optimization problems; i.e., $D = \mathbb{R}^n$
  Even if the original optimization problem is constrained (i.e., $x \in D$), we assume that it has been transformed to an unconstrained one (recall the method of Lagrange multipliers from calculus)
- To determine whether the Hessian is positive, negative or neither at $x^\star$, we use the eigenvalue decomposition and check if all the eigenvalues are positive, negative or both, respectively
- Many functions, can have both maxima, minima and saddle points

Example: Univariate function with both minima and maxima

Consider the function $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = x^4 - 4x^2$$

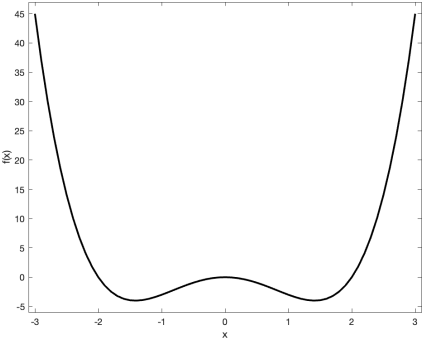


Figure 1: Plot of $f(x)$

Many statistical and machine learning problems lead to optimization problems that have unique global minima

A brief discussion of a broad class of objective functions that have unique global minima will be presented later on

Many statistical and machine learning problems lead to optimization problems that have unique global minima

A brief discussion of a broad class of objective functions that have unique global minima will be presented later on

Next, we introduce a general purpose algorithm (and its variants) that iteratively solves the gradient equation $\nabla f(x) = 0$

Problem Setup - I

Objective: Let $f : \mathbb{R}^n \to \mathbb{R}$

$$\min_{x \in \mathbb{R}^n} f(x) \tag{3}$$

We will study iterative descent algorithms

Outline of the algorithm:

1. Pick an initial point $x_0 \in \mathbb{R}^n$ (initial guess)
2. Successively generate sequence of $x_1, x_2, \cdots, x_k, x_{k+1} \cdots$ such that

$$f(x_{k+1}) \leq f(x_k) \qquad k = 1, 2, \cdots \tag{4}$$

Problem Setup - II

Illustration of the Setting

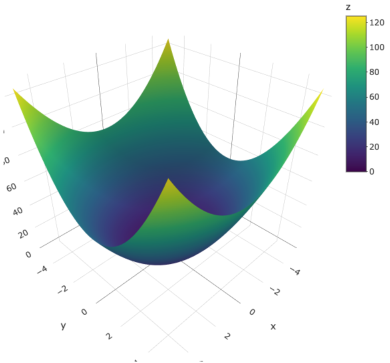


Figure 2: Plot of $f(x) = 3x_1^2 + 2x_2^2$

Problem Setup - III

Factors in designing iterative descent algorithms:

1. what direction to move: descent direction
2. how far to move in that direction: step size (in the machine learning literature, this factor is also referred to as the learning rate)
3. when to stop: stopping criterion

How to generate the iterates x_k?

The idea of iterative descent algorithms is to generate the next iterate x_{k+1} according to the rule

$$x_{k+1} = x_k + \eta_k d_k, \tag{5}$$

where $\eta_k \in \mathbb{R}_+$ is the step size (learning rate) and $d_k \in \mathbb{R}^n$ a descent direction

Descent direction design

Assume that f is differentiable; i.e., $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ exists everywhere

Then, by a first order Taylor expansion we get

$$f(x_{k+1}) = f(x_k + \eta_k d_k) \approx f(x_k) + \eta_k \left[\nabla f(x_k)\right]^\top d_k \tag{6}$$

Since we require $f(x_{k+1}) < f(x_k)$ we obtain that the following relationship should hold

$$\left[\nabla f(x_k)\right]^\top d_k < 0 \tag{7}$$

Step size design

Let $g(\eta) = f(x_k + \eta d_k)$

Then,

$$g'(\eta) = \nabla f(x_k + \eta d_k)^\top d_k \Longrightarrow g'(0) = \nabla f(x_k)^\top d_k < 0 \qquad (8)$$

by the design of the descent direction

Hence, there exists $\eta > 0$ such that

$$g(\eta) \approx g(0) + \eta g'(0) \Longrightarrow g(\eta) \leq g(0) \Longrightarrow f(x_k + \eta d_k) \leq f(x_k) \qquad (9)$$

Gradient Descent Algorithm - I

The most common choice for the descent direction is $-\nabla f(x)$, the negative of the gradient

Note that

$$\nabla f(x_k)^\top \left[-\nabla f(x_k) \right] = -\, ||\nabla f(x_k)||_2^2 < 0, \tag{10}$$

and hence $d_k = -\nabla f(x_k)$ is indeed a descent direction

Gradient Descent Algorithm - II

1. Select $x_0 \in \mathbb{R}^n$
2. While stopping criterion>tolerance do:
   - $x_{k+1} = x_k - \eta_k \nabla f(x_k)$
   - Calculate the value of the stopping criterion

Gradient Descent Algorithm - III

Common choices of stopping criteria:

1.
$$\|\nabla f(x_k)\|_2 \leq \text{tolerance}$$
If $\|\nabla f(x_k)\|_2 \approx 0$, then for all practical purposes, we have found an x_k that solves the gradient equation (2)

2.
$$|f(x_{k+1}) - f(x_k)| \leq \text{tolerance}$$
Improvements in function value are saturating

3.
$$\|x_{k+1} - x_k\|_2 \leq \text{tolerance}$$
Movement between iterates has become small

The tolerance parameter is usually set to a small value, such to 10^{-5} or 10^{-6}

Gradient Descent Algorithm - III

Common choices of stopping criteria:

1. $$\|\nabla f(x_k)\|_2 \leq \text{tolerance}$$

   If $\|\nabla f(x_k)\|_2 \approx 0$, then for all practical purposes, we have found an x_k that solves the gradient equation (2)

2. $$|f(x_{k+1}) - f(x_k)| \leq \text{tolerance}$$

   Improvements in function value are saturating

3. $$\|x_{k+1} - x_k\|_2 \leq \text{tolerance}$$

   Movement between iterates has become small

The tolerance parameter is usually set to a small value, such to 10^{-5} or 10^{-6}

Which one to choose? We will revisit this issue, since the best choice depends on additional properties of the f function

Illustration of GD through a 1-Dim Example - I

Let

$$f : \mathbb{R} \to \mathbb{R}, \text{ with } f(x) = 2x^2 + 2x - 16$$

We have that $\frac{df}{dx} = 4x + 2$ and $\frac{d^2f}{dx^2} = 4 > 0$, hence the function is twice differentiable

Its global minimum is at $\frac{df}{dx} = 0 \Longrightarrow x = -\frac{1}{2}$

Illustration of GD through a 1-Dim Example - II

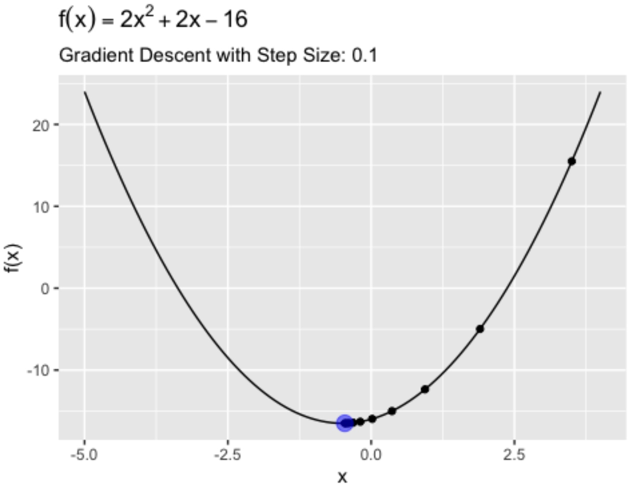


Figure 3: Iterates x_k, with $x_0 = 3.5, \eta = 0.1$

Illustration of GD through a 1-Dim Example - III

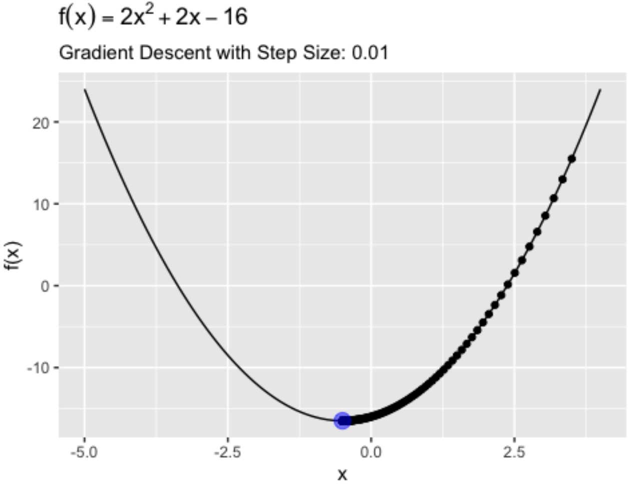


Figure 4: Iterates x_k, with $x_0 = 3.5, \eta = 0.01$

Illustration of GD though a 1-Dim Example - IV

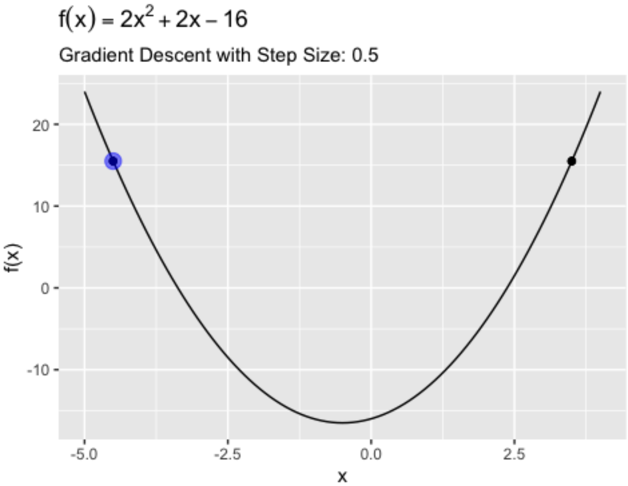


Figure 5: Iterates x_k, with $x_0 = 3.5, \eta = 0.5$

Illustration of GD through a 1-Dim Example - V

$\eta = 0.5$

Sequence of iterates:

Iterate	Value
x_0	3.5
x_1	-4.5
x_2	3.5
x_3	-4.5
...	...

The gradient descent algorithm does NOT converge

Illustration of GD through a 1-Dim Example - VI

Suppose that $\eta = 1$

Sequence of iterates:

Iterate	Value
x_0	3.5
x_1	-12.5
x_2	35.5
x_3	-108.5
x_5	323.5
...	...

The gradient descent algorithm DIVERGES

Illustration of GD through a 1-Dim Example - VII

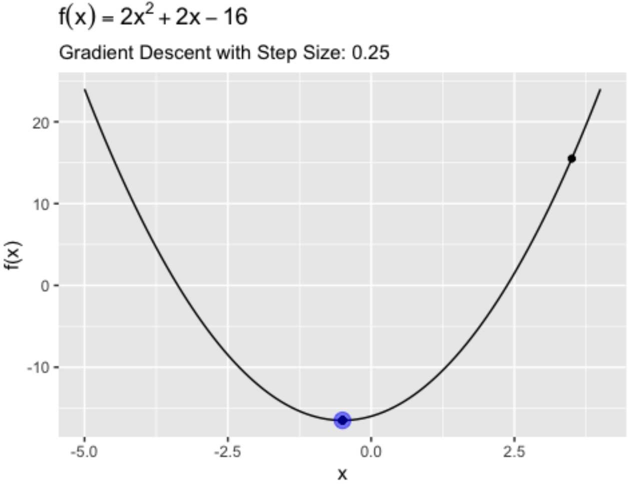


Figure 6: Iterates x_k, with $x_0 = 3.5, \eta = 0.25$

Illustration of GD through a 1-Dim Example - VIII

Suppose that $\eta = 0.249999$

Sequence of iterates:

Iterate	Value
x_0	3.5
x_1	-0.499984
x_2	-0.5

The gradient descent algorithm converges fast (2 steps)

Summary Regarding the Step Size based on the 1-Dimensional Example

The selection of the step size (learning rate) η matters!!

- A very small step size, results in a lot of iterations for the gradient descent algorithm to converge to the minimum
- A very large step size may result in the gradient descent algorithm to diverge
- A really good choice of the step size makes the gradient descent algorithm converge in fewer iterations

In the illustrative example, the step size η was kept fixed for all iterations (e.g., $\eta = 0.1$ throughout)

What if we started with a reasonable large value and gradually reduce the step size i.e., use a variable step size η_k across iterations

Before we discuss strategies to select the step size, we illustrate the rationale behind the gradient descent algorithm

For those interested, the technical explanation can be found in the Appendix

Illustration of the GD Rationale

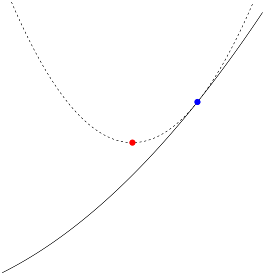


Figure 7: Blue point is x_k; Red point is x_{k+1}, the minimizer of the function $h(z) = f(x_k) + [\nabla f(x_k)]^\top (z - x_k) + \frac{1}{\eta}||z - x_k||_2^2$

The GD algorithm finds the minimum of a local quadratic approximation of the $f(x)$ function at the current update x_k; the approximation depends on the step size selected

Illustration of GD on a function in $\mathbb{R}^2$

Let $f : \mathbb{R}^2 \to \mathbb{R}$, with

$$f(x) \equiv f(x_1, x_2) = 3x_1^2 + 0.5x_2^2 + 2x_1x_2$$

We have that

$$\nabla f(x) = \begin{bmatrix} 6x_1 + 2x_2 \\ x_2 + 2x_1 \end{bmatrix}$$

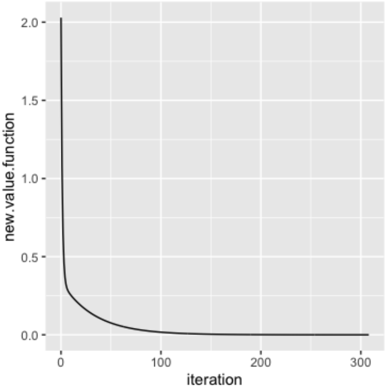
Some algebra gives that the global minimum is $x_{\text{min}} = (0, 0)$, since

$$\nabla^2 f(x) = \begin{bmatrix} 6 & 2 \\ 2 & 1 \end{bmatrix}$$

a positive definite matrix (its determinant is positive)

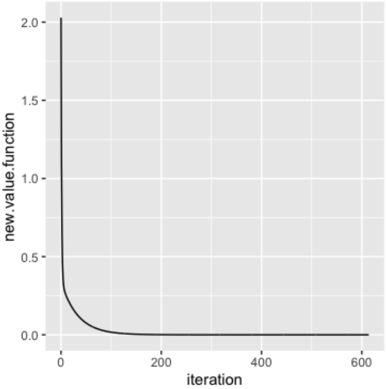
Results for the 2-Dim Example Function - I

$\eta = 0.05$ and tolerance $= 0.000001$
$\hat{x}_{\text{min}} = (-0.004873299, 0.013892708)$, so not quite close to the theoretical value!!



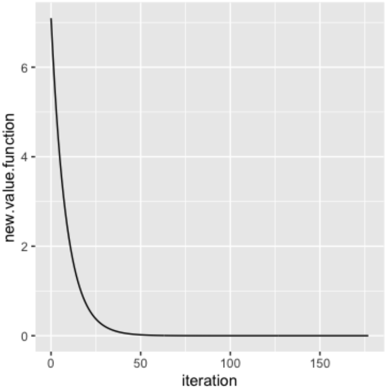
Results for the 2-Dim Example Function - II

$\eta = 0.05$ and tolerance $= 0.000000001$
$\hat{x}_{\text{min}} = (-4.896064e-05, 1.395761e-04)$, so practically the theoretical solution



Results for the Example Function - III

$\eta = 0.29$ and tolerance $= 0.000000001$
$\hat{x}_{\min} = (-4.527236e-05, -1.572929e-05)$, so practically the theoretical solution, but fewer iterations



Some Remarks

As with the 1-dim example, the step size matters a lot

In all the illustrative examples presented thus far, a fixed value of η was used

What strategies can we employ to select iterate dependent stepsizes; i.e., η_k instead of a fixed η

Selection Strategies for the Step Size η

1. Exact Line Search method
2. Backtracking line search method
3. Constant step size

Appendix

Rationale for the Gradient Descent Algorithm - I

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable such that $\nabla^2 f(\cdot)$ is positive definite for all $x \in \mathbb{R}^n$ and also f has a unique minimum

Then, a second order Taylor expansion of the function $f(x)$ around a point $a \in \mathbb{R}^n$ is given by

$$f(x) \approx f(a) + \left[\nabla f(a)\right]^\top (x - a) + \frac{1}{2}(x - a)^\top \nabla^2 f(a)(x - a)$$

Recall that this approximation is accurate for any x close to the prespecified point a

Rationale for the Gradient Descent Algorithm - II

Let us consider a point x_0 (the initialization of the gradient descent algorithm) as a (i.e., $a = x_0$) and we are interested in an update x_1 (close to x_0), such that

$$f(x_1) < f(x_0)$$

Since we are interested in minimizing f, we want at every iteration to have $f(x_{k+1}) < f(x_k)$

Then, the Taylor expansion formula gives

$$f(x_1) \approx f(x_0) + \left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2}(x_1 - x_0)^\top \nabla^2 f(x_0)(x_1 - x_0),$$

and since we want $f(x_1) < f(x_0)$ we obtain that

$$\left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2}(x_1 - x_0)^\top \nabla^2 f(x_0)(x_1 - x_0) < 0$$

Rationale for the Gradient Descent Algorithm - III

Then, the question becomes how to select x_1, assuming that $\nabla f(x_0) \neq 0$?

First, let us consider the following simplification. Note that since we have assumed that $\nabla^2 f(x)$ is positive definite in the domain of f, then there exists $\eta > 0$, such that

$$\nabla^2 f(x) \geq \frac{1}{\eta} > 0$$

and the Taylor expansion formula becomes

$$f(x_1) \approx f(x_0) + \left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2\eta}(x_1 - x_0)^\top (x_1 - x_0)$$

To find the optimal x_1, we will minimize the approximation with respect to x_1; i.e.,

$$\min_{x_1} \; \left[f(x_0) + \left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2\eta}\|x_1 - x_0\|_2^2\right]$$

Rationale for the Gradient Descent Algorithm - IV

To do so, we take the derivative with respect to x_1, set it to zero and solve for x_1; i.e.,

$$\frac{d}{dx_1}\left[f(x_0) + \left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2\eta}\|x_1 - x_0\|_2^2\right] = \nabla f(x_0) + \frac{1}{\eta}(x_1 - x_0)$$

Then,

$$\nabla f(x_0) + \frac{1}{\eta}(x_1 - x_0) = 0 \Longrightarrow x_1 = x_0 - \eta \nabla f(x_0)$$

It is easy to check that x_1 is indeed a minimum, since the second derivative of the approximation function with respect to x_1 is $\frac{1}{\eta} > 0$

Rationale for the Gradient Descent Algorithm - V

What about the requirement that $f(x_1) < f(x_0)$ or equivalently that

$$\left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2}(x_1 - x_0)^\top \nabla^2 f(x_0)(x_1 - x_0) < 0?$$

Plug in the optimal $x_1 = x_0 - \eta \nabla f(x_0)$ to obtain

$$\left[\nabla f(x_0)\right]^\top (x_1 - x_0) + \frac{1}{2}(x_1 - x_0)^\top \nabla^2 f(x_0)(x_1 - x_0) = -\frac{\eta}{2}\left[\nabla f(x_0)\right]_2^2 < 0$$