SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Gradient Search Methods for Unconstrained Optimization

av

Adam Epstein

2019 - No K24

Gradient Search Methods for Unconstrained Optimization

Adam Epstein

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Lars Arvestad

2019

Abstract

Optimization consists of minimizing or maximizing an objective function over a certain domain. We cover minimization problems without loss of generality. Unconstrained optimization is when we have no constraints on our objective function. There are various ways to perform optimization. Numerical methods are superior for high complexity problems which are common in many applications. Gradient search methods use information of the derivatives to efficiently find a optimum.

The thesis treats unconstrained optimization with gradient search methods. The primary focus will be Gradient descent over convex functions. Applications in linear regression will be treated. Gradient descent will be compared with primary Newton-Raphson and the more advanced methods of Quasi-Newton and Conjugate direction. The comparison will cover convergence properties.

The convergence of Gradient descent is fast in the initial phase and slow in the end, this is due to a shrinking step size. The sequence of points generated by gradient descent converges in a bounded zigzag pattern if the conditions under the convergence theorem hold (Theorem 12). The convergence rate of gradient descent is highly dependent on the shape of the objective function. Newton-Raphson might not converge with an initial point far from optimum but converges fast with quadratic rate of convergence close to the optimum. In the Quasi-Newton and Conjugate direction methods we combine the benefits of Gradient descent and Newton-Raphson methods to benefit of both.

Acknowledgement

I thank Lars Arvestad for his support and ideas to improve the thesis.

Contents

List of Figures

1 Introduction

This thesis will give a theoretical foundation of Gradient methods for solving unconstrained optimization problems. Without loss of generality we will only consider minimization problems, maximization problems can be transformed into minimization problems by negating the objective function. Gradient methods are methods to find an extreme point of a function, if the function is differentiable the method always finds a local optimal solution; if we want a guaranteed global solution the function must be convex. All the quadratic functions treated in the thesis will be quadratic convex functions, quadratic convex is a special case of convex. Some theory hold for the special case of quadratic convex functions. More on quadratic functions in Section 2.3.

The gradient methods are numerical methods (Section 1.1.2). We are going to highlight important properties, so that one knows when to implement the different gradient methods. The primary focus will be the mathematical theory of Gradient descent, and compare to the multivariate Newton-Raphson. The properties of Quasi-Newton methods and Conjugate direction methods will be treated, these methods incorporate the benefits of both Gradient descent and Newton-Raphson.

1.1 Two problem solving strategies

The theory behind Analytic methods is fundamental for the Numerical methods [1]. We are going to focus on Numerical methods in this bachelor thesis, primarily Gradient descent. We will learn about the analytic foundations of our numerical methods.

1.1.1 Analytic methods

Analytic methods involve e.g. Calculus. Calculus is the part of mathematics that treats limits, integrals and derivatives. These methods are used to find exact solutions to problems. By using analytic methods we can learn the properties of the problem, make simplifications and transform the problem into a problem we can solve. Analytic methods are only feasible for small problems or problems with low complexity [1]. Many problems in real life

have high complexity.

1.1.2 Numerical methods

Numerical methods are approximate methods that use many easy steps iteratively to reach a solution, this enables us to solve complex problems. These methods are used when the analytic solutions is too time consuming, approximation is acceptable and when an analytical method is missing. Many problems in e.g. ordinary differential equations and partial differential equations do not posses any analytical solution methods .

1.2 About appendix

The theory placed in the appendix is more loosely connected to the "message" of the thesis, but more of a foundational nature. It is recommended to have a look in the appendix to get an overview of the foundational theory, to determine what is already clear and if there is something to learn now or later when reading the text. This gives an opportunity to further understanding of the theory in the main text. The appendix treats basic calculus, linear algebra and topology which is the theory about sets.

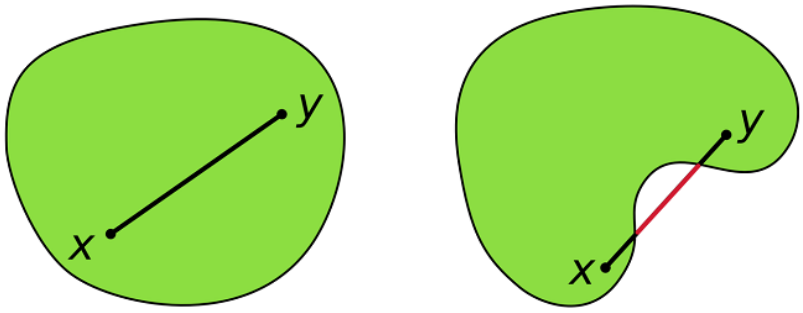
2 Convex theory

To find the minimum of an objective function, we can use gradient descent. But gradient descent only gives the global minimum if the function is convex, this is why we need to dive into the field of convex functions which are defined over convex sets. Quadratic functions will also be discussed as a special case of convex functions.

In appendix B we can find a section about the theory of sets (topology), which applies for some of the theory on convexity and some of the theory later in the text.

2.1 Convex set and Convex function

Definition 1 (Convex set)**.** *A set S in $\mathbb{R}^n$ is said to be convex if $x_1, x_2 \in S$ and $\lambda \in [0,1]$ then $\lambda x_1 + (1-\lambda)x_2 \in S$.*



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

Figure 1: The intuition is that a set is convex if we can draw a line between two points in the set, and the line remains in the set. Source: Wikimedia commons.

Definition 2 (Convex function)**.** *Let $f : S \to \mathbb{R}$, where S is a nonempty convex set in $\mathbb{R}^n$. The function f is said to be convex on S if*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

for each $x_1, x_2 \in S$ and for each $\lambda \in [0,1]$.

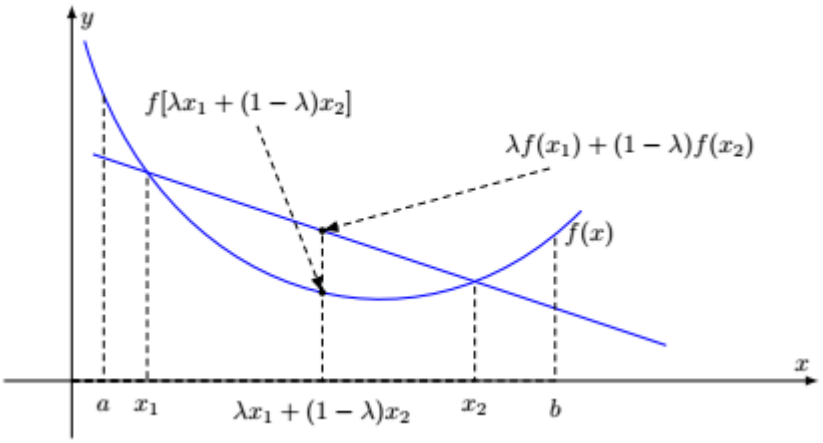


Figure 2: Convex function $f(x)$ over the intervall $[a, b]$, where $x_1, x_2 \in [a, b]$ and $\lambda \in [0, 1]$.

The intuitive meaning of a convex function is that we can draw a line between any two points on the function graph, where the line will lay above the function graph no matter which two point we chose.

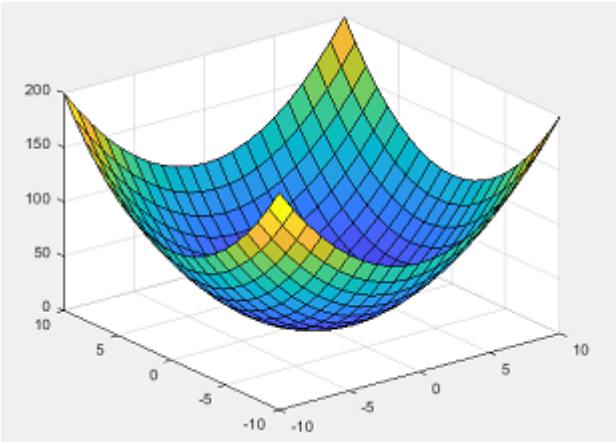


Figure 3: The Convex and quadratic function $f(x_1, x_2) = x_1^2 + x_2^2$.

2.2 Minimum property of convex function

First we define local and global optimum.

Definition 3 (Local/global minimum)**.** *Consider the problem of minimizing $f(x)$ over a domain S. Let $x' \in S$. If there exists a neighbourhood $N_\epsilon(x')$ around x' and $f(x') \leq f(x)$ for each $x \in N_\epsilon$, then x' is called a local minimum. If $f(x') \leq f(x)$ for all $x \in S$ then x' is called a global minimum.*

We will now prove a important property of convex functions: If our function is convex, then there is exactly one optimal solution x', in this case this is where the local optimal solution is equal to the global optimal solution.

Theorem 1 (Extremepoint of convex function)**.** *Let S be a nonempty convex set in $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be convex on S. Consider the problem to minimize $f(x)$ subject to $x \in S$. Suppose that $x' \in S$ is a local optimal solution to the problem. Then x' is the global optimal solution.*

Proof. Since x' is a local optimal solution, there exists a neighbourhood $N_\epsilon(x')$ around x' such that

$$f(x) \geq f(x') \text{ for each } x \in S \cap N_\epsilon(x'). \tag{1}$$

Now suppose that x' is not a global solution so that $f(x'') < f(x')$ for some x''. By the use of the definition of convexity for f we get

$$f(px'' + (1-p)x') \leq pf(x'') + (1-p)f(x') < pf(x') + (1-p)f(x') = f(x'),$$

where $p \in (0, 1)$.

Let $p > 0$ and sufficiently small then:

$$px'' + (1-p)x' \in S \cap N_\epsilon(x').$$

This contradicts equation (1), x' is a global optimal solution. □

We can always find the global solution using gradient search methods if the convergence conditions in Theorem 12 are fulfilled [2].

2.3 Quadratic functions

Definition 4 (Symmetric matrix)**.** *A symmetric matrix is a square matrix $Q \in \mathbb{R}^{n\times n}$ with the property that*

$$Q^T = Q$$

Definition 5 (Positive semidefinite)**.** *The symmetric matrix Q is positive semidefinite when the following hold:*

$x^T Qx \geq 0$.

In this thesis we want to work with convex functions, for the nice properties like in Theorem 1. Our quadratic functions $f(x) = \frac{1}{2}x^T Qx + c^T x, x \in \mathbb{R}^n$ will be convex if Q is positive semidefinite (Theorem 2), because of this we will assume that when we are using quadratic functions the matrix Q will always be positive semidefinite. Convex functions will not necessarily be quadratic, some theory will only be valid for the special case of quadratic functions.

We will use the concept of concave functions when we prove the next theorem.

Definition 6 (Concave function)**.** *The function $f : S \to \mathbb{R}$ is called concave on S if $-f$ is convex on S.*

Theorem 2. *The function $f(x) = \frac{1}{2}x^T Qx + c^T x$ is a convex function if and only if Q is positive semidefinite [3].*

Proof. First, suppose that Q is not positive semidefinite. Then there exists r such that $r^T Qr < 0$. Let $x = \theta r$. Then $f(x) = f(\theta r) = \frac{1}{2}\theta^2 r^T Qr + \theta c^T r$ is strictly concave ($f(\theta r) =: h(\theta) = \alpha\theta^2 + \theta\gamma, \alpha < 0$ and $\gamma \in \mathbb{R}$) on the subset $\{x|x = \theta r\}$, since $r^T Qr < 0$. Thus $f(x)$ is not a convex function.

Next, suppose that Q is positive semidefinite. For all $\lambda \in [0, 1]$, and for all x, y,

$$f(\lambda x + (1-\lambda)y) = f(y + \lambda(x-y))$$

$$= \frac{1}{2}(y + \lambda(x-y))^T Q(y + \lambda(x-y)) + c^T(y + \lambda(x-y))$$

$$= \frac{1}{2}y^T Qy + \lambda(x-y)^T Qy + \frac{1}{2}\lambda^2(x-y)^T Q(x-y) + \lambda c^T x + (1-\lambda)c^T y$$

$$\leq \frac{1}{2}y^TQy + \lambda(x-y)^TQy + \frac{1}{2}\lambda(x-y)^TQ(x-y) + \lambda c^Tx + (1-\lambda)c^Ty$$

$$= \frac{1}{2}\lambda x^TQx + \frac{1}{2}(1-\lambda)y^TQy + \lambda c^Tx + (1-\lambda)c^T$$

$$= \lambda f(x) + (1-\lambda)f(y)$$

this shows that $f(x)$ is a convex function. □

3 Gradients

The idea behind gradient descent is to search for the minimum of a function. We start with a point on the function and travel in the direction of steepest descent (this is the direction of the negative gradient, which is proven in Theorem 5). We run the gradient descent algorithm until we reach a minimum. More on Gradient descent in Section 6.

The gradient is only defined for differentiable functions so lets start by defining differentiable and then the gradient.

Definition 7 (Differentiability)**.** *Let $\bar{a}$ be an interior point in the domain $S \subseteq \mathbb{R}^n$ of a function $f : S \to \mathbb{R}$. We say that f is differentiable at $\bar{a}$ if there are constants $A_1, ..., A_n$ and a function $\rho(h)$ such that*

$$f(\bar{a}+h) - f(\bar{a}) = A_1h_1 + ... + A_nh_n + |h|\rho(h) \tag{2}$$

and

$\lim_{h\to 0} \rho(h) = 0.$

where h is a n dimensional vector.

If f is differentiable in every point $\bar{a} \in S$, we say f is differentiable [4].

Definition 8 (Gradient)**.** *For differentiable functions $f(x) = f(x_1, ..., x_n)$ we define the gradient of f, in the point x, as the vector:*

$\nabla f = (\frac{\partial f}{dx_1}(x), ..., \frac{\partial f}{dx_n}(x)).$

The gradient is the vector of partial derivatives.

Theorem 3. *If a function $f(x)$ is differentiable at a point $x = \bar{a}$, then the function is continuous at that point.*

Proof. From equation 2, we can conclude that:

$$f(\bar{a} + h) - f(\bar{a}) \to 0$$

as $h \to 0$. This means that f is continuous in $\bar{a}$. □

We know that the gradient is only defined for differentiable functions and that differentiability implies continuity, this means that we can only do gradient descent on functions that are continuous. This makes sense because the negative gradient follows the direction of steepest descent (Theorem 5), which is along the function surface. This process is not possible if the function graph is discontinuous.

4 Properties of gradients

To fully understand the algorithm of gradient descent and related algorithms we need to understand the different properties of gradients. We need to introduce the concept of directional derivatives and to prove properties of the gradients.

Definition 9 (Directional derivative)**.** *By the derivative of $f(x)$ in the point $\bar{a}$ with respect to the direction v, $|v| = 1$, we mean the limit:*

$f'_v(\bar{a}) = \lim_{t\to 0} \frac{f(\bar{a}+tv)-f(\bar{a})}{t}$.

Theorem 4. *If f is a differentiable function and v is the direction, $|v| = 1$ then*

$$f'_v(\bar{a}) = \nabla f \cdot v. \tag{3}$$

Proof. Let

$$u(t) = f(\bar{a} + tv), t \in \mathbb{R}.$$

This function describes the behaviour of f on the line $x = \bar{a} + tv$. We derive that

$$f'_v(\bar{a}) = \lim_{t\to 0} \frac{u(t) - u(0)}{t} = u'(0).$$

Using the chain rule we obtain

$u'(t) = \nabla f(\bar{a} + tv) \cdot v.$

Insert $t = 0$ and we get equation (3). □

Theorem 5. *The gradient $\nabla f(x)$ has the direction in which the function f has the steepest ascent in the point x.*

Proof. To do this we are going to use the Cauchy-Schwartz inequality and Theorem 4 to show that

$|f'_v(x)| = |\nabla f \cdot v| \leq |\nabla f| \cdot |v| = |\nabla f|.$

$|\nabla f \cdot v| \leq |\nabla f| \cdot |v|$ is only equal when the vectors $|\nabla f|$ and $|v|$ are parallel, i.e. the maximal slope for the directional derivative is the slope of the gradient. This means that the gradient is the direction of steepest ascent. In the case of steepest descent we are going use the negative gradient, because the magnitude of the gradient is the same, but the direction is the opposite. □

4.1 Level surface and level curves

Definition 10 (Level surface)**.** *Assume that $f : S \to \mathbb{R}$ is a function of n variables, and that $c \in \mathbb{R}$ is a constant. Then the set*

$$L_c = \{x \in S | f(x) = c\}$$

is called a level surface to f [5].

Level curves is where the function $f(x, y) = c$, c constant, in the special case with 2 variables, the function will be projected onto the $\mathbb{R}^2$-plane in the way showed in the image.

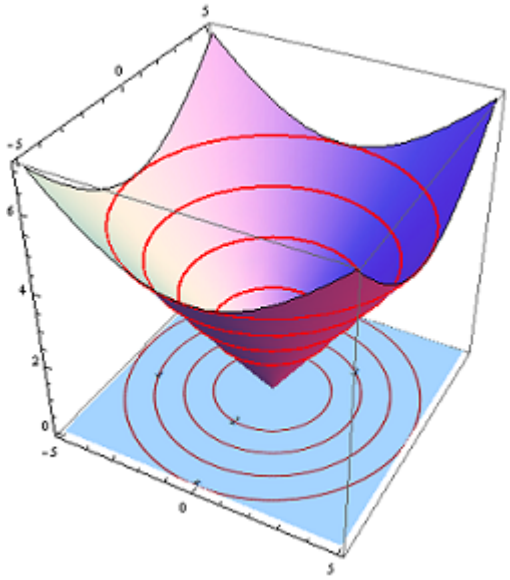


Figure 4: Level curves
Projection of function values on the function domain.

We will prove a theorem about how the gradient relates to the level curves, it is relevant because we can visualize the Gradient descent algorithm with the function being plotted as level curves, because then it is easier to spot patterns like the zigzag pattern (Section 6.5).

Theorem 6. *Assume that $f : S \to \mathbb{R}$ is a function of n variables and that f is differentiable in the point $\bar{a}$. If $f(\bar{a}) = c$ then the gradient is always normal to the level surface L_c in the following regard: If r is an differentiable curve which is on the level surface ($f(r(t)) = c$ for all t), and $r(t_0) = \bar{a}$ then*

r exist in the point $\bar{a}$ at $t = t_0$ then

$$\nabla f(\bar{a}) \cdot r'(t_0) = 0,$$

The tangent vector of the curve in the point $\bar{a}$ is normal to the gradient $\nabla f(\bar{a})$ in the point.

Proof. Because $r(t)$ is on the level surface L_c the function $u(t) = f(r(t)) = c$.

$u'(t_0) = \nabla f(r(t_0)) \cdot r'(t_0) = \nabla f(\bar{a}) \cdot r'(t_0) = 0.$ □

The derivative of the position vector is parallel to the level curve. We get by the scalar product that the derivative of the position vector is perpendicular to the gradient.

5 Properties of functions and hessian matrix

5.1 Properties of functions

Before we describe the gradient descent algorithm we want to describe two theorems, to understand some more theory behind the inner workings of the algorithm. The first theorem (Theorem 7) tells that the negative gradient actually successively decreases the cost function. The second theorem (Theorem 8) shows that all our minimum points have the property $\nabla f(x) = 0$.

Theorem 7 (Descent direction)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at x, and there exist a vector d such that $\nabla f(x)^T d \leq 0$, then there exists a $\delta > 0$ such that $f(x + \lambda \cdot d) < f(x)$ for each $\lambda \in (0, \delta)$, so that d is a descent direction of f at x.*

Proof. By the differentiability of f at x, we must have

$$f(x + \lambda d) = f(x) + \lambda \nabla f(x)^T d + \lambda |d| \alpha(x; \lambda d)$$

where $\alpha(x; \lambda d) \to 0$ as $\lambda \to 0$. Rearranging the terms and dividing by $\lambda, \lambda \neq 0$, we get:

$$\frac{f(x + \lambda d) - f(x)}{\lambda} = \nabla f(x)^T d + |d| \alpha(x; \lambda d).$$

Since $\nabla f(x)^T d < 0$ and $\alpha(x; \lambda d) \to 0$ as $\lambda \to 0$, then there exists a $\delta > 0$ such that $\nabla f(x)^T d + |d| \alpha(x; \lambda d) < 0$ for all $\lambda \in (0, \delta)$. □

Theorem 8 (Local minimum point)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at x. If x is a local minimum then $\nabla f(x) = 0$.*

Proof. Suppose that $\nabla f(x) \neq 0$. Then, letting $d = -\nabla f(x)$, we get $\nabla f(x)^T d = -|\nabla f(x)|^2 < 0$; and by theorem 7, there is $\delta > 0$ such that $f(x + \lambda d) < f(x)$ for $\lambda \in (0, \delta)$ contradicting the assumption that x is a local minimum. Hence, $\nabla f(x) = 0$. □

5.2 Hessian matrix

The Hessian matrix can be used to find out if a function is convex. If a function is convex, then we can always find the global minimum (Theorem 1). The Hessian is also going to be used in the gradient search method Newton-Raphson (Section 8.2).

Definition 11 (Hessian matrix)**.** *Let S be a nonempty set in $\mathbb{R}^n$ and let $f : S \to \mathbb{R}$. Then, f is said to be twice differentiable at $x' \in int(S)$ if there exist a vector $\nabla f(x')$, and an $n \times n$ symmetric matrix $H(x)$, called the Hessian matrix, and a function α such that:*

$$f(x) = f(x') + \nabla f(x')^T (x - x') + \frac{1}{2}(x - x')^T H(x')(x - x') + |x - x'|^2 \alpha(x - x') \quad (4)$$

for each $x' \in S$, where $lim_{x \to x'} \alpha(x'; x - x') = 0$. The function f is said to be twice differentiable on the open set $S' \subseteq S$ if it is twice differentiable at each point in S'.

We notice that for twice differentiable functions the Hessian is comprised of second order derivatives $\partial^2 f(x)/\partial^2 x_i x_j$ for $i = 1, ..., n$, $j = 1, ..., n$

$$H(x) = \begin{bmatrix} \partial^2 f(x)/\partial^2 x_1^2 & \partial^2 f(x)/\partial^2 x_1 x_2 & \dots & \partial^2 f(x)/\partial^2 x_1 x_n \\ \partial^2 f(x)/\partial^2 x_2 x_1 & \partial^2 f(x)/\partial^2 x_2^2 & \dots & \partial^2 f(x)/\partial^2 x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial^2 f(x)/\partial^2 x_n x_1 & \partial^2 f(x)/\partial^2 x_n x_2 & \dots & \partial^2 f(x)/\partial^2 x_n x_n \end{bmatrix}.$$

In equation 4 the right-handside expression is equal to the second order Taylor series expansion approximation if we exclude the rest term associated with α.

Our next theorem will develop the crucial connection between the Hessian matrix and convexity. It tells us about the global convexity of the function f, and its relation to the positive semidefinite Hessian matrix at each point.

Theorem 9. *Let S be a nonempty open convex set in $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be twice differentiable on S. Then, f is convex if and only if the Hessian matrix is positive semidefinite at each point in S.*

Proof. Suppose that f is convex and let $x' \in S$. We need to show that $x^T H(x')x \geq 0$ for each $x \in \mathbb{R}^n$. Since S is open, then, for any given $x \in \mathbb{R}^n$, $x' + \lambda x \in S$ for $|\lambda| \neq 0$ and sufficiently small. We can find two expressions:

$$f(x' + \lambda x) \geq f(x') + \lambda \nabla f(x)^T x, \tag{5}$$

$$f(x' + \lambda x) = f(x') + \lambda \nabla f(x')^T x + \frac{1}{2}\lambda^2 x^T H(x')x + \lambda^2 |x|^2 \alpha(x'; \lambda x). \tag{6}$$

Equation 5 is valid if and only if f is convex [2] and by the twice-differentiability of f we yield Equation 6. Subtracting Equation 6 from Equation 5, we get

$$\frac{1}{2}\lambda^2 x^T H(x')x + \lambda^2 |x|^2 \alpha(x'; \lambda x) \geq 0,$$

dividing by λ^2 and letting $\lambda \to 0$, it follows that $\frac{1}{2}x^T H(x')x \geq 0$. Conversely, suppose the Hessian matrix is positive semidefinite at each point in S. Consider x and x' in S. Then, by the mean value theorem [2], we have

$$f(x) = f(x') + \nabla f(x')^T (x - x') + \frac{1}{2}(x - x')^T H(x'')(x - x') \tag{7}$$

where $x'' = \lambda x' + (1 - \lambda)x$ for some $\lambda \in (0, 1)$. Note that $x'' \in S$ and, hence, by assumption, $H(x'')$ is positive semidefinite. Therefore $(x - x')^T H(x'')(x - x') \geq 0$ and from equation 7, we conclude that

$$f(x) \geq f(x') + \nabla f(x')^T (x - x').$$

Since the above inequality is true for each x, $x' \in S$, f is convex. This completes proof. □

The next theorem shows that the Hessian matrix is positive semidefinite at the local minimum points. We already proved the first part of the theorem in theorem 8.

Theorem 10. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at x. If x is a local minimum, then $\nabla f = 0$ and $H(x)$ is positive semidefinite.*

Proof. p. 133 in [2]. □

When we have a non-convex function, we can find the global minimum by comparing the functional values of the local minimums in the domain, and the find the point with the lowest functional value.

6 Gradient descent

Gradient descent is a part of gradient search methods. Search method use steps iteratively, proceeding from an initial approximation x_1 of the minimization point to successive points $x_2, x_3, \ldots$, until some stopping condition is satisfied. "The Gradient descent method is one of the most fundamental procedures for minimizing a differentiable function of several variables" [2]. The method gives essential insight into more advanced methods, methods like Newton-Raphson (Section 8.2), Quasi-Newton (Section 8.5) or Conjugate direction methods (Section 8.6). The more advanced methods are often an attempt to modify the gradient descent algorithm in such way that the new algorithm will have superior convergence properties [6].

6.1 Gradient descent algorithm

Let's describe the gradient descent algorithm. Given a point x, the steepest descent algorithm proceeds by performing a line search along the direction of $-\nabla f(x)$ to find a new point, the process is repeated until a stopping condition is reached. A summary of the method is given below.

1. Initialization step. Let $\epsilon > 0$ be the termination scalar. Choose a initial point x_1, let $k = 1$ and go to the main step.

2. Main step: If $|\nabla f(x_k)| < \epsilon$, stop; otherwise, let $d_k = -\nabla f(x_k)$, let λ_k be an optimal solution to the problem to minimize $f(x_k + \lambda d_k)$ subject to $\lambda \geq 0$. Let $x_{k+1} = x_k + \lambda_k d_k$, replace k by $k + 1$, and repeat the main step.

Gradient descent determines the next point on the function surface, which is in the direction of steepest decent. We can see that the above algorithm stops searching if $d_k = 0$ because $x_{k+1} = x_k$.

6.2 Gradient descent in linear regression

Gradient descent is widely used in machine learning [7]. We are going use gradient descent to determine a linear regression, linear regression is considered a machine learning algorithm [7], because the machine finds pattern in the data through an algorithm. Linear regression (Figure 5) is a method to find the best linear trend for data in $\mathbb{R}^n$. We can with some modifications even do a polynomial fit to a given data set [7].

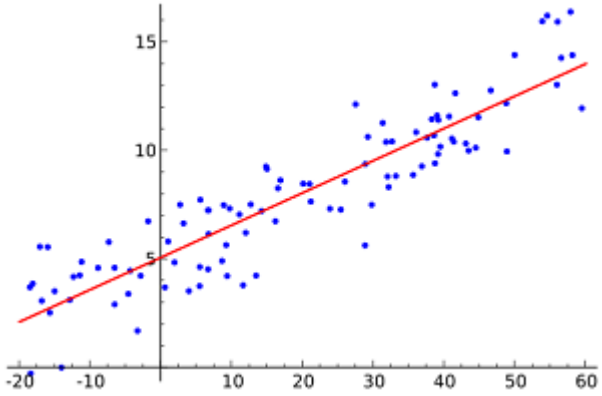


Figure 5: Linear regression (red), a trend line for data points (blue). Source: Wikimedia commons.

Let $x_i = x_{i1}, ..., x_{in}$ be the "input" variables and y_i the "output" variable for the data pair (x_i, y_i). Let i be data pair number, and n the number of input variables, x_i can for example be house size and $(n-1)$ other properties like location and y_i can be the house price.

In machine learning each data pair (x_i, y_i) is an observation called training example, because it is necessary to supply the data pairs to train the algorithm. When we train the algorithm with data it is called supervised learning in machine learning terminology [7].

We want to find an affine function $h : \mathbb{R}^n \to \mathbb{R}$, that fits data which is the objective of linear regression. There will be an error in the linear regression called ϵ_i, if the regression does not go through all points.

The linear regression model is given as:

$$h_\beta(x_i) = \beta_0 + \beta_1 x_{i1} + ... + \beta_n x_{in}, i = 1, 2, ..., n$$

$$y_i = h_\beta(x_i) + \epsilon_i, i = 1, 2, ..., n$$

the vectors are given as:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, x_i^T = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}$$

The matrix form for linear regression is $y = X\beta + \epsilon$.

6.3 Least squares

When we are doing gradient descent on linear regression we are trying to minimize the least squares function. The term Least squares function reflects that we are trying to minimize the vertical square distances from the regression line to the output variable y. The objective is to find the best possible solution for a regression.

Often we can't get an unique solution real life data in $\mathbb{R}^n$. If there is no solution we have an inconsistent system of linear equations, This means $X\beta \neq y$. We want to find the best possible solution by minimizing error: minimize $|y - X\beta|$.

Definition 12 (Least square solution)**.** *Consider the system:*

$$X\beta = y, \tag{8}$$

where X is an $n \times p$ matrix. A vector β' in $\mathbb{R}^p$ is called a least-square solution of this system if $|y - X\beta'| \leq |y - X\beta|$ for all β in $\mathbb{R}^p$ [8].

The least squares function is related to the error ϵ. This is the function we want to minimize for regression. Here is the Least square function $J(\beta)$:

$$J(\beta) = \frac{1}{2}\sum_{i=1}^{n}(h_\beta(x_i) - y_i)^2.$$

Our objective is to find β' by $\min_\beta J(\beta)$ and receiving the Least squares solution.

There is an analytic way of finding the least squares solution, we are going to derive it. $X\beta$ is the column space of X denoted $ColX$. $X\beta \neq y$ means that y is not in the column space. Let the closest point from y in $Col(X)$ be y', such that $y' = proj_{Col(X)}y$. Because $y' \in Col(X)$, the equation $X\beta = y'$ has a solution, let there be a $\bar{\beta}$ in $\mathbb{R}^n$ such that

$$X\bar{\beta} = y'.$$

$y - y' = y - X\bar{\beta}$ is ortogonal to $Col(X)$, so $y - X\bar{\beta}$ is orthogonal to each column of X. If a_j is any column of X, then $a_j \cdot (y - X\bar{\beta}) = 0$, and $a_j^T \cdot (y - X\bar{\beta})$. Since each row a_j^T is a row in X^T[9] we get

$$X^T(y - X\bar{\beta}) = 0,$$
$$X^T y = X^T X\bar{\beta}.$$

The expression $X^T y = X^T X\beta$, is called the normal equations of $X\beta = y$. If $X^T X$ is invertable ($det(X^T X) \neq 0$), we can provide a closed formula for the least squares solution [8]

$$\beta' = (X^T X)^{-1} X^T y. \tag{9}$$

6.4 Cost function plot

The cost function J is dependent on β, $J = J(\beta)$. The value of β determines how good the regression is, by minimizing $J(\beta)$ we get the best possible solution. There is a least squares solution β' such that $J(\beta') = \min_\beta J(\beta)$ which yields the minimal value of the cost function. In Figure 6 we give an example of a cost function plot, where we are applying gradient descent algorithm (Section 6.1) to reach the minimal value of the cost function. The gradient will find the exact global minimum if the function is convex (Theorem 1) if we use exact line search (Section 8.1.1) and the conditions under the convergence theorem (Theorem 12) are met.

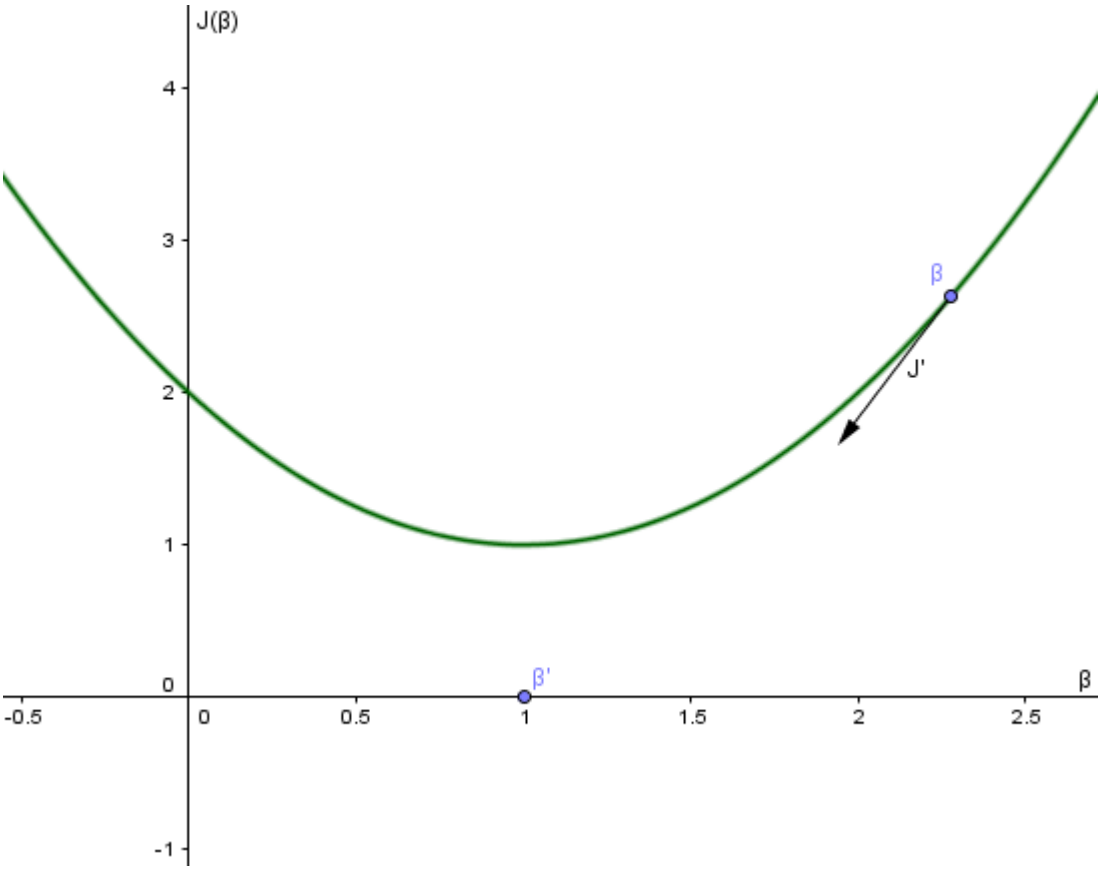


Figure 6: Minimization of the cost function by gradient descent (2D case).

6.5 Rate of convergence and zigzag pattern

6.5.1 Rate of convergence

Now if the conditions for convergence in Theorem 12 is meet, we want to treat the rate of convergence for Gradient descent. Gradient descent usually performs quite well during the early stages of the optimization process, depending on the point of initialization. However, as we approach a minimum point, the method behaves increasingly poorly because of smaller and smaller stepsize, which we will explore soon. The sequence of points $\{x_k\}$ generated by the algorithm will converge in a zigzag pattern (The zigzag feature is discussed in Section 6.5.2). These problems of poor convergence in the later stages of the algorithm can be explained by considering the following

expression of the function f.

$$f(x + \lambda d) = f(x) + \lambda \nabla f(x)^T d + \lambda |d| \alpha(x; \lambda d) \tag{10}$$

where $\alpha(x; \lambda d) \to 0$ as $\lambda d \to 0$

and d is a search direction. If x_k is close to a extreme-point with zero gradient, and f is continuously differentiable, then $|\nabla f(x)|$ will be small, making the term $\lambda \nabla f(x)^T d$ of small magnitude. This is a first order approximation of f, which is what gradient descent use since it calculates the first order derivatives. The error term $\lambda |d| \alpha(x; \lambda d)$ will have higher influence at the end of the algorithm. This means the steps size gets smaller and smaller.

6.5.2 Shape of level curves and zigzag pattern

We will determine more properties of the gradient descent algorithm, we are going to use a convex function where we can find global minimum (Theorem 1) using Gradient descent, if we assume that the conditions under (Theorem 12) hold.

We will examine the following properties: How the shape of the level curves of the objective function will affect the convergence rate, and how pronounced the zigzag pattern will be. We will also show that the zigzag pattern will be bounded between two lines. We exemplify this by a quadratic convex function

$$f(x_1, x_2) = \frac{1}{2}(x_1^2 + \alpha x_2^2), \alpha > 1.$$

The reason we chose this quadratic function is that the $\frac{1}{2}$ will cancel out as we compute the gradient. The variables x_1^2, x_2^2 makes the function bivariate quadratic without adding too much complexity. The term α tells the skewness of the level curves of f. As α increases the level curves become more skewed, this results that the graph of the function become increasingly steep in the x_2 direction in relation to the x_1 direction. Given an initial point $x = (x_1, x_2)^T$, let us apply one iteration of Gradient descent to get a new

point $x_{new} = (x_{1new}, x_{2new})^T$. If $x_1 = 0$ and $x_2 = 0$, then the algorithm stops in the optimal point $x' = (0, 0)$. Hence, suppose that $x_1 \neq 0$ and $x_2 \neq 0$.

The gradient descent direction is given as the negative gradient of the objective function, $d = -\nabla f(x) = -(x_1, \alpha x_2)^T$. The successive point is given as, $x_{new} = (x + \lambda d)$, where λ solves the line search problem to minimize $\theta(\lambda) = f(x + \lambda d) = \frac{1}{2}[x_1^2(1 - \lambda)^2 + \alpha x_2^2(1 - \alpha\lambda)^2]$ subject to $\lambda \geq 0$. Let $\theta'(\lambda) = 0$, we obtain

$$\lambda = \frac{x_1^2 + \alpha^2 x_2^2}{x_1^2 + \alpha^3 x_2^2}$$

which yields

$$x_{new} = \left[\frac{\alpha^2 x_1 x_2^2(\alpha - 1)}{x_1^2 + \alpha^3 x_2^2}, \frac{x_1^2 x_2(1 - \alpha)}{x_1^2 + \alpha^3 x_2^2}\right].$$

Observe that $x_{1new}/x_{2new} = -\alpha^2(x_2/x_1)$, and let $x_1^0/x_2^0 = K \neq 0$, these two values are the inverse gradient of two straight lines. The sequence of values $\{x_1^k, x_2^k\}$ alternative between as the sequence $\{x^k\}$ converges to $x' = (0, 0)$. Gradient descent will converge under the conditions stated in Theorem 12 [2]. This means that the sequence zigzags between the pair of straight lines $x_2 = (1/K)x_1$ and $x_2 = (-K/\alpha)x_1$. The zigzag pattern will be more pronounced as α increases, as the strait line will align more narrowly. On the other hand, if $\alpha = 1$ then the contours or f are circular and we obtain optimum x' in one iteration.

Let's give an example of the zigzag pattern: in Figure 7 we are implementing gradient descent on a function, we are observing the level curves of the function. We can see that we approach the minimum of the function in a zigzag pattern and the stepsize gets smaller and smaller. In later chapters we are going to discover methods that have other patterns of convergence.

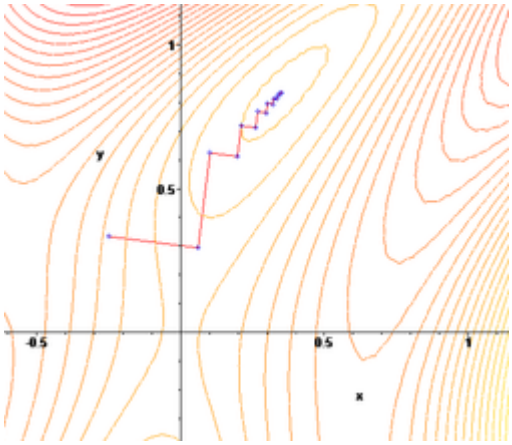


Figure 7: The zigzag convergence of gradient descent. The function where the gradient descent is applied is $f(x, y) = \sin(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3)\cos(2x + 2 - e^y)$. Source: Wikimedia commons.

7 Convergence theorem

The convergence theorem (Theorem 12) is used to show convergence for many algorithms, for example gradient search algorithms. The theorem in summary states that: if the sequence generated by the algorithm is contained in a compact set, then the Gradient descent algorithm (Section 6.1) converges to a point with zero gradient. We need to define point -to-set maps, and use Bolzano-Weierstrass theorem to prove the theorem.

Definition 13 (Algorithmic map)**.** *Given a point x_k and by applying the algorithm, we obtain a new point x_{k+1}. This map is generally a point-to-set map and assigns to each point in the domain X a subset of X. Thus, given the initial point x_1, the algorithmic map generates the sequence $x_1, x_2, \ldots$, where $x_{k+1} \in A(x_k)$ for each k. The transformation of x_k into x_{k+1} thorough the map constitutes an iteration of the algorithm.*

Theorem 11 (Bolzano-Weierstrass)**.** *Every bounded infinite subset of $\mathbb{R}^k$ has a limit point in $\mathbb{R}^k$.*

Proof. p.40 in [10]. □

Theorem 12 (Convergence)**.** *Let X be a nonempty closed set in $\mathbb{R}^n$, and let the nonempty set $\Omega \subset X$ be the solution set. Let $A : X \to X$ be a point-to-set*

map. Given $x_1 \in X$, the sequence $\{x_k\}$ is generated iteratively as follows:

- *If $x_k \in \Omega$ then stop; otherwise, let $x_{k+1} \in A(x_k)$, replace k by $k+1$, and repeat.*

Suppose that the sequence $x_1, x_2, ..$ produced by the algorithm is contained in a compact subset of X, and suppose that there exists a continuous function α, called the descent function, such that $\alpha(y) < \alpha(x)$ if $x \notin \Omega$ and $y \in A(x)$. If the map A is closed over the complement of Ω then either the algorithm stops in a finite number of steps with a point in Ω or it generates a infinite sequence $\{x_k\}$ such that:

1. *Every convergent subsequence of $\{x_k\}$ has a limit in Ω, that is, all accumulation points of $\{x_k\}$ belong to Ω.*
2. *$\alpha(x_k) \to \alpha(x)$ for some $x \in \Omega$.*

Proof. If at any iteration a point x_k in Ω is generated, then the algorithm stops. Now suppose that an infinite sequence $\{x_k\}$ is generated. Let $\{x_k\}_G$ be any convergent subsequence with limit $x \in X$. Since α is continuous, then, for $k \in G$, $\alpha(x_k) \to \alpha(x)$. Thus, for a given $\epsilon > 0$, there is a $k \in G$ such that

$$\alpha(x_k) - \alpha(x) < \epsilon \text{ for } k \geq K \text{ with } k \in G.$$

In particular for $k = K$, we get

$$\alpha(x_K) - \alpha(x) < \epsilon. \tag{11}$$

Now let $k > K$. Since α is a descent function, $\alpha(x_k) < \alpha(x_K)$, and, from (11), we get

$$|\alpha(x_k) - \alpha(x)| = \alpha(x_k) - \alpha(x_K) + \alpha(x_K) - \alpha(x) < 0 + \epsilon = \epsilon.$$

Since this is true for every $k > K$, and since $\epsilon > 0$ was arbitrary, then

$$\lim_{k\to\infty} \alpha(x_k) = \alpha(x). \tag{12}$$

We now show that $x \in \Omega$. By contradiction, suppose that $x \notin \Omega$, and consider the sequence $\{x_{k+1}\}_G$. This sequence is contained in a compact subset of X and, hence, has convergent subsequence $\{x_{k+1}\}_G$ with limit $\bar{x}$ in X. Noting (12), it is clear that $\alpha(\bar{x}) = \alpha(x)$. Since A is closed at x, and for $k \in \bar{G}$, $x_k \to x$, $x_{k+1} \in A(x_k)$, and $x_{k+1} \to \bar{x}$, then $\bar{x} \in A(x)$. Therefore, $\alpha(\bar{x}) < \alpha(x)$, contradicting the fact that $\alpha(\bar{x}) = \alpha(x)$. Thus, $x \in \Omega$ and part 1 of the theorem holds true. This, coupled with (12), shows that part 2 of the theorem holds true, and the proof is complete. □

8 Gradient methods for unconstrained optimization

Our primary focus so far has been to introduce Gradient descent, the theory behind, and the application to linear regression (Section 6.2). The theory includes topics like convex theory (Section 2), differentiability, the properties of gradients and how the Hessian matrix can be used to find out if a function is convex (Theorem 9) or if a point is a minimum point (Theorem 10).

Let's give a recap why we did this. Gradients require differentiability (Theorem 8), this means we are going to work with functions that have the differentiability property so we can use gradient based search methods. The gradient is aimed in the direction of steepest descent (Theorem 5) and is orthogonal to the level curves (Theorem 6) which gives us the zigzagpattern of gradient descent (Section 6.5).

We have used convex functions for the theoretical development because they lead to relatively easy to understand theorems and give a good intuition of the properties of various methods. We can for example always find a global minimum point (Theorem 1).

We now want to widen the scope and discuss some other gradient search methods, that are more effective, by taking account for the second order information of the function surface. There are two important aspects that we must consider when choosing a gradient search method:

- How should the direction d be chosen?
- How large step should be taken in the direction d from the on current point to the next?

Lets give the general template of how the unconstrained gradient search method looks like. We can see that it looks similar to the gradient descent algorithm, but in the gradient descent algorithm the direction is specified as $-\nabla f$.

1. Find a start point x_1. Let $k = 1$.
2. Find a search direction d_k.
3. If $|d_k| \leq \epsilon$ stop.
4. Find t_k from $\min_{t \geq 0} f(x_k + t_k d_k)$ with line search.
5. Let $x_{k+1} = x_k + t_k d_k$, set $k = k + 1$ return to 2.

We will discuss Step 4 to understand how we determine the step size once we have determined the direction of travel. Either we can have a fixed step size or a dynamic step size that depends on where on the function surface we are located, Line search is one of the more dynamic approaches.

8.1 Line search methods

When we are using gradient search methods, we need to determine the step length in every iteration, to do so we are performing a line search. There are exact and inexact line search methods, exact line search finds an exact optimal solution while inexact line search find a rough estimate of the optimal solution.

Very often in practice it is too expensive to perform an exact line search because of excessive function evaluations, even if we terminate with a small accuracy tolerance $\epsilon > 0$. On the other hand, if we sacrifice accuracy, then we might impair on the convergence of the overall algorithm that iteratively employs such a line search.

If we adopt a line search that guarantees a sufficient degree of accuracy, this might be sufficient for the algorithm to converge while improving efficiency. In summary we want a good step size for our algorithm, with an acceptable trade-off between accuracy and efficiency. Step size in machine learning is called learning rate. We will start off by showing an exact line search method, which finds exactly the minimal point.

8.1.1 Exact line search

We say that an iterative method has the property of quadratic termination if our algorithm reaches the exact optimal point of a quadratic function $f(x)$ in a finite number of steps, exact line search have this property if we are working with convex functions [11] which we are.

If we insert $x_{k+1} = x_k + t_k d_k$ in the objective function for an unknown value t, we get a function dependent on t, $\phi_k(t) = f(x_{k+1}) = f(x_k + td_k)$. The objective is to solve the problem

$$\min \phi_k(t), \text{ when } t \geq 0.$$

If $f(x)$ is a differentiable convex quadratic function then $\phi_k(t)$ will have the same property [11].

8.1.2 Inexact line search: Armijo's Rule

One inexact line search method for finding an acceptable step size is Armijo's rule. Armijo's Rule is driven by two parameters, $0 < \epsilon < 1$ and $\alpha > 1$, which will manage the acceptable step length from being too small or too large. Suppose we are minimizing a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$, in a direction $d \in \mathbb{R}^n$, where $\nabla f(x)^T d < 0$, by theorem 7 this is a descent direction. Define the line search function $\theta : \mathbb{R} \to \mathbb{R}$ as $\theta(\lambda) = f(x + \lambda d)$ for $\lambda \geq 0$. Then we can get a first order approximation of θ at $\lambda = 0$ given by $\theta(0) + \lambda\theta'(0)$

$$\bar{\theta}(\lambda) = \theta(0) + \lambda\epsilon\theta'(0), \text{ where } \lambda \geq 0.$$

A step is considered "acceptable", provided that $\theta(\lambda) \leq \bar{\theta}(\lambda)$. However, to prevent λ from being too small, Armijo's Rule also requires that $\theta(\alpha\lambda) > \bar{\theta}(\alpha\lambda)$.

8.1.3 Inexact line search: Newtons method

We can use Newton method (Section 8.2) with or without line search, lets look at Newton's method for line search. Newton's method is based on exploiting the quadratic approximation of the function $f : \mathbb{R} \to \mathbb{R}$ in a given point x. The quadratic approximation is given by $p(x)$

$$p(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2$$

the point x_{k+1} is chosen such that the derivative of $p'(x) = 0$, we want to find the minimum of p. This yield

$$f'(x_k) + f''(x_k)(x_{k+1} - x_k) = 0,$$
$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k).}$$

This procedure is terminated when $|x_{k+1} - x_k| < \epsilon$, where ϵ is a termination scalar. This process can only be applied for twice differentiable functions. The process is only well defined when $f''(x_k) \neq 0$ for each k.

8.2 Newton-Raphson method

The Newton-Raphson method use information about the second order derivatives to find the minimum point, this is more effective when we have a quadratic function. Let $y_k(x)$ be the second order approximation of $f \in C^2$, in a suitable neighbourhood of the current point x_k, we evaluate $f(x)$ and its first and second order derivatives at $x = x_k$, this gives

$$y_k(x) = f(x_k) + (x - x_k)^T g_k + \frac{1}{2}(x - x_k)^T H(x_k)(x - x_k) \qquad (13)$$

where $H(x_k)$ is the Hessian of the function in the point x_k, and g_k the gradient vector of $f(x)$ evaluated in x_k. We are searching for a stationary point (minimum) to $y_k(x)$ hence a point x where $\nabla y_k(x) = 0$. We get

$$\begin{aligned} 0 = \nabla y_k(x) &= g_k + H(x_k)(x - x_k), \\ H(x_k)(x - x_k) &= -g_k, \\ x - x_k &= -H(x_k)^{-1} g_k, \\ x &= x_k - H(x_k)^{-1} g_k. \end{aligned}$$

The Newton-Raphson method uses x as the next current point giving the iterative formula

$$x_{k+1} = x_k - H(x_k)^{-1} g_k.$$

Our direction vector points from one point to the next

$$d_k = x_{k+1} - x_k = -H(x_k)^{-1} g_k. \tag{14}$$

We can modify this equation

$$d_k = -\lambda_k H(x_k)^{-1} g_k \tag{15}$$

where λ_k is determined by a line search from x_k in the direction $-D_k g_k$.

8.3 Convergence and speed of convergence for Newton-Raphson method

8.3.1 Convergence and divergence

The Newton-Raphson method diverges when $-H(x_k)^{-1} g_k$ is a direction of ascent. We need to look at the convergence criterion for Newton-Raphson's method to find a direction of descent. We know that Gradient descent choices

the direction of steepest descent. The Newton-Raphson method also chooses a direction of descent when the angle between the direction vector of Newton-Raphson and Gradient descent is less than 90 degrees. We can formulate this by the scalar product

$$(H(x_k)^{-1}g_k)^T g_k > 0,$$

$$g_k^T H(x_k)^{-1} g_k > 0. \tag{16}$$

This is a result of the symmetry of the Hessian $H(x_k)^{-1} = (H(x_k)^{-1})^T$. Equation 16 is satisfied at all points where $g_k \neq 0$ if $H(x_k)$ is Positive definite. Unfortunately, if x_k is not close x', it might happen that D_k is not positive definite, the method fails to converge in this case [12]. Here is an example from reference [12] where the algorithm fails to converge.

8.3.2 Example of divergence

Minimize

$$f(x_1, x_2) = x_1^4 - 3x_1x_2 + (x_2 + 2)^2$$

starting at the point $\bar{x_1} = [0, 0]$.

Solution We can find the next point by this formula

$$x_{k+1} = x_k - \lambda_k H(x_k)^{-1} g_k, \tag{17}$$

λ_k can be obtained by line search.

The gradient vector and the Hessian matrix of $f(x)$ are given by

$$g_k(x) = [4x_1^3 - 3x_2, -3x_1 + 2(x_2 + 2)], H(x_1, x_2) = \begin{pmatrix} 12x_1^2 & -3 \\ -3 & 2 \end{pmatrix}.$$

Evaluated at $[0, 0]$ we get

$$g_1 = [0, 4], H_1 = \begin{pmatrix} 0 & -3 \\ -3 & 2 \end{pmatrix},$$

$$H_1^{-1} = -\frac{1}{9}\begin{pmatrix} 2 & 3 \\ 3 & 0 \end{pmatrix}, -H_1^{-1}g_1 = \begin{pmatrix} \frac{4}{3}, 0 \end{pmatrix}.$$

By using the Equation 17 we find the next functional value

$f(\bar{x_2}) = f\left(\frac{4}{3}\lambda_1, 0\right) = \frac{256}{81}\lambda_1^4 + 4.$

We do not have to do any line search to find out that the minimizing value is $\lambda_1 = 0$, this means the algorithm stops in the point because the new direction is an increasing direction. If we use equation

$x_{k+1} = x_k - H(x_k)^{-1}g_k,$

without the line search factor we obtain

$f(\bar{x_2}) = \frac{256}{81} + 4 > f(\bar{x_1})$

this is also an increasing value, which means both methods fails.

8.3.3 Convergence speed

For a convex, quadratic function, $f(x) = \frac{1}{2}x^TQx + c^Tx$ we get $\nabla f(x) = Qx + c$ and $H = Q$ which yield that $y_k(x) = f(x)$ where y_k is defined in equation 13 [11]. This means that the Newton-Raphson direction of descent point right toward the optimum and Newton-Raphson method solve a convex quadratic problem in one iteration [11]. Newton-Raphson obtain a property called quadratic or second order convergence (Theorem 13).

Theorem 13. *(Quadratic convergence) Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously three times differentiable. Newton's algorithm is defined as $x_{k+1} = x_k - H(x_k)^{-1}\nabla f(x_k)$. Let $\bar{x}$ be such that $\nabla f(\bar{x}) = 0$ and $H(x)^{-1}$ exist. Let the starting point x_1 be sufficiently close to $\bar{x}$ so that proximity implies that there exist $k_1, k_2 > 0$ with $k_1k_2|x_1 - \bar{x}| < 1$ such that*

1. *$|H(x_k)^{-1}| \leq k_1$[2]. And by the Taylor series expansion of ∇f,*
2. *$|\nabla f(\bar{x}) - \nabla f(x_k) - H(x_k)(\bar{x} - x_k)| \leq k_2|\bar{x} - x_k|^2$.*

Then the algorithm converges with quadratic rate of convergence to $\bar{x}$.

In practice quadratic convergence yields that if we start close enough to a solution our algorithm will double the number of correct digits for the current point x_k relative the optimal solution for each iteration [13]. This makes sense because near to the extreme point (in a minimizing problem) the function is often approximatly convex [12]. When the function is convex we can yield important information from the second order derivatives. The property of local convexity is utilized in Section 8.6.

8.4 Comparison Newton-Raphson and Gradient descent

We can note that when the Hessian is non invertible, we can't use the Newton-Raphson method, because we can't find new directions in the gradient search. The only difference between the Newton-Raphson and Gradient descent algorithm is the search direction (see general template for unconstrained gradient search method, in introduction of Section 8). For convex functions the Newton-Raphson method usually gives better search directions than gradient descent because it incorporates second derivatives which holds more information about the function. The Newton-Raphson method converges rapid when x_k is near the optimal point (Theorem 13), but might not converge far from the optimal point (Section 8.3). Gradient descent convergence fast far from the optimal point and slow close to the optimal point (Section 6.5). We want to combine the two algorithms to get something that combines the benefits of the two separate algorithms. We can utilize Quasi-Newton methods to achieve this, we shall discuss the DFP (Davidon-Fletcher-Powell) method.

8.5 A Quasi-Newton method: The Davidon-Fletcher-Powell method

The part of Newton's method that requires the most computer power is the computation of the inverse Hessian matrix, $H(x_k)^{-1}$. In Quasi-Newton methods we decrease the computational load by replacing the Hessian with a more easy computed approximation $D_k(x)$. We obtain a new direction vector $d_k = -D_k g_k$ instead of $d_k = -H(x_k)^{-1} g_k$. D_k is symmetric and positive definite just like the Hessian. The matrix D_k is updated each iteration, we get a new iteration algorithm:

$$x_{k+1} = x_k - D_k g_k,$$

if we use line search

$$x_{k+1} = x_k - \lambda_k D_k g_k.$$

We define $D_1 = I$ where I is the unit matrix, this means that the first step the algorithm uses the same direction as gradient descent, which is the negative gradient direction. The slow convergence of gradient descent near the optimal point x' is overcome by choosing a sequence D_k in such that D_k becomes approximately equal to $H(x_k)^{-1}$ as x_k approaches x'. The disadvantage of DFP is that the quadratic convergence from Newtons method is lost [13], being replaced by a convergence called super linear, super linear convergence have the quadratic termination property, which is when we reach the exact optimal solution in a finite number of steps [13].

Theorem 14. *If the DFP is used to minimize the quadratic function $f(x)$ with n variables and H being positive and symmetric (Hessian matrix), then $D_{k+1} = H(x_k)^{-1}$ and the exact minimum is reached in, at most, n iterations.*

Proof. p. 113-115 [12]. □

8.5.1 DFP algorithm

This is the Quasi-Newton algorithm DFP for finding minimum of an objective function using gradient search.

1. Set $d_k = -D_k g_k$ with $D_1 = I$. Where d_k is the direction of search from the current point x_k.
2. Perform a line search to find $\lambda'_k \geq 0$, where λ' is the value of λ_k that minimizes $f(x_k + \lambda_k d_k)$.
3. Set $\sigma_k = \lambda'_k \cdot d_k$.
4. $x_{k+1} = x_k + \sigma_k$ yielding the new current point.
5. Evaluate $f(x_{k+1})$ and g_{k+1}, noting that g_{k+1} is orthogonal to σ_k, hence

$$\sigma^T g_{k+1} = 0$$

σ_k is tangential to the level hyper surface while and g_{k+1} is orthogonal to the level hyper surface.

6. Set $\gamma_k = g_{k+1} - g_k$.

7. $D_{k+1} = D_k + A_k + B_k$ where

$$A_k = \frac{\sigma_k \sigma_k^T}{\sigma_k^T \gamma_k}$$

$$B_k = \frac{-D_k \gamma_k \gamma_k^T D_k}{\gamma_k^T D_k \gamma_k}.$$

8. Set $k = k + 1$ return to step 1.

9. Stop when either $|d_k| < \epsilon$ or when the components of d_k is less than some prescribed amount. The creators of the algorithm Fletcher and Powell recommend that the calculations be continued for at least n iterations in order to avoid false minimum.

We always find minimum for DFP algorithm in n iterations where n is the number of variables of the objective function. Let us state and prove another property of DFP, to get a better knowledge of the Quasi-Newton algorithm.

Definition 14 (Square root of matrix)**.** *A matrix B is said to be a square root of A if the matrix product BB is equal to A.*

Theorem 15. *In the DFP method, D_k is positive definite for all k.*

Proof. The proof is inductive. First , $D_1 = I$, which is positive definite. Now assume the theorem is true for $k = K$; we shall prove that it is true for $k = K+1$. In step 2, the direction of the search is "downhill", i.e. $g_K d_K < 0$ and hence $\lambda'_K > 0$ Define the vectors

$$p = D_K^{\frac{1}{2}} \theta \text{ and } q = D_K^{\frac{1}{2}} \gamma_K$$

where θ is an arbitrary non zero vector θ. The matrix $D_K^{\frac{1}{2}}$ exist [12], where D_k is a symmetric positive definite matrix, the proof is omitted. From equations in step 7,we find

$$\theta^T D_{K+1}\theta = \theta^T D_K\theta + \frac{(\theta^T\sigma_K)^2}{\sigma_K^T\gamma_K} - \frac{(\theta^T D_K\gamma_K)^2}{\gamma_K^T D_K\gamma_K}$$

$$= p^2 + \frac{(\theta^T\sigma_K)^2}{\sigma_K^T\gamma_K} - \frac{(p^Tq)^2}{q^2}$$

$$= \frac{p^2q^2 - (p^Tq)^2}{q^2} + \frac{(\theta^T\sigma_K)^2}{\sigma_K^T\gamma_K}$$

$$\geq \frac{(\theta^T\sigma_K)^2}{\sigma_K^T\gamma_K}.$$

Where we have used Cauchy-Schwartz inequality ($p^2q^2 \geq (p^Tq)^2$). Now we get

$$\sigma_K^T\gamma_K = \sigma_K^T g_{K+1} - \sigma_K^T g_K,$$

$$\sigma_K^T\gamma_K = -\sigma_K^T g_K \text{ using equalities in step 5 of DFP algorithm,}$$

$$= \lambda'_K g_K^T D_K g_K > 0 \text{ using equalites in step 1,3}$$

since $\lambda'_K > 0$ and D_k is positive definite. Hence $\theta^T D_{K+1}\theta > 0$ for all non zero θ, i.e. D_{K+1} is positive definite, and the induction is complete. □

8.6 Conjugate direction methods

Conjugate direction methods are used for the same reason as Quasi-Newton methods, they are a intermediate between Gradient descent and Newton-Raphsons method. The methods are motivated to accelerate the slow convergence rate of gradient decent close to optimum while avoiding the information requirement associated with evaluation, storage and inversion of the

Hessian matrix as required in Newton-Raphsons method. Conjugate direction methods are created for purely quadratic problems

$$\text{minimize } \frac{1}{2}x^T Qx - b^T x$$

where Q is an $n \times n$ matrix. The methods that are used for quadratic problems can be used, by approximation, for other more general problems, since we can assume that around the solution point the function is approximately quadratic [6], convergence around that point is similar as in a quadratic problem. The conjugate gradient algorithms are considered among the best general purpose methods available [6] (1989).

Definition 15 (Conjugate vectors)**.** *Given a symmetric matrix Q, two vectors d_1 and d_2 are said to be conjugate with respect to Q, if $d_1^T Q d_2 = 0$.*

For the applications on convex functions that we consider we also want Q to be positive definite.

Theorem 16. *If Q is positive definite and the set of non zero vectors $d_0, d_1, d_2, \ldots, d_k$ are conjugate, then these vectors are linearly independent.*

Proof. p. 239 in [6]. □

The conjugate gradient algorithm is a conjugate direction method. In the conjugate gradient algorithm the linear independence of the vectors are important because, in step k of the algorithm one evaluates the current negative gradient and a linear combination of the previous direction vectors which makes the algorithm formulae simpler than in the DFP Quasi-Newton method.

Theorem 17 (Conjugate direction)**.** *Let $d_0, d_1, \ldots, d_{n-1}$ be a set of non zero conjugate vectors. For any $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated according to*

$$x_{k+1} = x_k + \alpha_k d_k, k \geq 0$$

with

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}$$

and

$$g_k = Qx_k - b$$

converges to the unique solution, x' of $Qx = b$ after n steps, such that, $x_n = x'$.

Proof. p. 240 in [6]. □

We have the same property as the Quasi-Newton methods, convergence in at most n steps.

9 Conclusion

The thesis treats unconstrained optimization with gradient search methods, with the goal to minimize an objective function. To solve a general unconstrained minimization problem is hard, so we have chosen to focus on the areas of convex functions and the special case of quadratic functions. These functions only have one optimum which is the global optimum, which simplifies the optimization effort. To use gradient search methods differentiability of the function is required, so this is assumed. Applications of gradient search methods are displayed in the area of linear regression. Gradient descent is the gradient search method we discuss in greatest detail, because it is basic and gives good insight in the theory for other gradient search methods. The other methods that are discussed are: Newton-Raphson, Quasi-Newton and Conjugate direction methods. Stepsize is fixed or determined by line search. Direction is what differentiates the various methods and determines the convergence properties. The convergence of Gradient descent is fast in the initial phase and slow in the end, this is due to a shrinking step size. The sequence of points generated by gradient descent converges in a bounded zigzag pattern if the conditions under the convergence theorem hold (Theorem 12).

The convergence rate of gradient descent is highly dependent on the shape of the objective function. Newton-Raphson might not converge with an initial point far from optimum but converges fast with quadratic rate of convergence close to the optimum. The methods of Quasi-Newton and Conjugate direction are motivated to accelerate the slow convergence rate of gradient decent close to optimum while avoiding the information requirement associated with evaluation, storage and inversion of the Hessian matrix as required in Newton-Raphsons method.

References

[1] "Analytical vs Numerical methods," *Research Gate*. [Online]. Available: https://www.researchgate.net/post/What_are_the_advantages_of_numerical_method_over_analyatical_method2

[2] M. S. Bazarra, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*, 2nd ed. Pearson, 1993.

[3] Robert M. Freund, "Quadratic Functions, Optimization, and Quadratic Forms," 2004. [Online]. Available: https://ocw.mit.edu/courses/sloan-school-of-management/15-084j-nonlinear-programming-spring-2004/lecture-notes/lec4_quad_form.pdf

[4] A. Persson and L.-C. Böiers, *Analys i flera variabler*, 2005.

[5] T. Kolsrud, T. Lindström, and K. Hveberg, *Flervariabelanalys med linjär algebra*, 2012.

[6] D. G. Luenberger, *Linear and Nonlinear programming*, 1989.

[7] A. Ng, "CS229 Lecture notes," pp. 1–15. [Online]. Available: https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf

[8] Otto Bretscher, "Linear algebra with applications." Prentice Hall, 2009, ch. 5, pp. 222–224.

[9] D. C. Lay, *Linear Algebra and its Applications*, 3rd ed., 2006.

[10] W. Rudin, *Principles of mathematical analysis*, 1976.

[11] K. Holmberg, *Optimering*. Liber AB, 2010.

[12] G. Walsh, *Methods of Optimization*. Wiley, 1985.

[13] D. J. Faires and R. L. Burden, *Numerical methods*, 1993.

[14] L. Sadun, *Applied linear algebra*. American Mathematical Society, 2018.

A Basic definitions and theorems

A.1 Calculus

Definition 16 (Limit)**.** *Let f be a real-valued function defined on a subset D of the real numbers. Lets c be a limit point of D and let L be a real number. We say that:*

$\lim_{x\to c} f(x) = L$ *if for every $\epsilon > 0$ there exists a δ such that, for all $x \in D$, if $0 < |x - c| < \delta$, then $|f(x) - L| < \epsilon$.*

Definition 17 (Continuity)**.** *The function f is continuous at some point c of its domain if the limit of $f(x)$, as x approaches c thorough the domain of f, exists and is equal to $f(c)$*

$$\lim_{x\to c} f(x) = f(c).$$

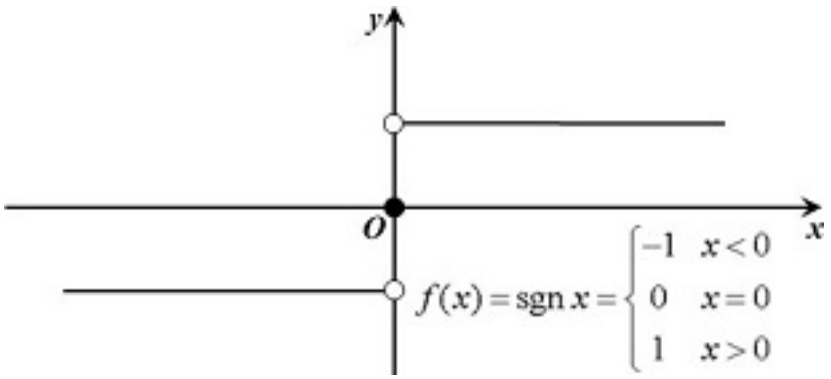


Figure 8: This is an example of a discontinuous function. Intuitively a function is continuous is when we can draw the function without lifting the drawing tool. Source: Wikimedia commons.

Definition 18 (Derivative, single variate)**.** *If $f'(c) = \lim_{x\to c} \frac{f(x)-f(c)}{x-c}$ exist then f has a derivative at the point c.*

Definition 19 (class C^n)**.** *Consider an open set on the real line and a function $C^n, n \in Z$ is said to be of class C^n when if the derivatives $f^{(1)}, ..., f^{(n)}$ exist and are continuous.*

A.2 Linear algebra

Definition 20 (Scalar product). *[14]* $x, y \in \mathbb{R}^n$

$x \cdot y = x^T y = x_1 y_1 + ... + x_n y_n = \sum_{i=1}^{n} x_i y_i$.

alternatively:

$x \cdot y = |x||y| \cos(\theta)$

where θ is the angle between the two vectors, this definition only holds in $\mathbb{R}^2$ and $\mathbb{R}^3$.

Theorem 18 (The Cauchy-Schwarz inequality). *If $u = (u_1, ..., u_n)$ and $v = (v_1, ..., v_n)$ are vectors in $\mathbb{R}^n$, then $|u \cdot v| \leq |u| \cdot |v|$.*

Definition 21 (Subspace). *A subspace of $\mathbb{R}^n$ is any set H in $\mathbb{R}^n$ that has three properties:*

1. *The zero vector is in $\mathbb{R}^n$.*
2. *For each $u, v \in H$, the sum $u + v \in H$.*
3. *For each $u \in H$ and each scalar c, the vector $cu \in H$.*

The points 2 and 3 defines that a subspace is closed under addition and scalar multiplication.

Definition 22 (Linear combination). *Given vectors $v_1, v_2, \ldots, v_p$ in $\mathbb{R}^n$ and given scalars $c_1, c_2, \ldots, c_p$ the vector y, defined by*

$$y = c_1 v_1 + \cdots + c_p v_p$$

is called a linear combination of $v_1, v_2, \ldots, v_p$ with weights $c_1, c_2, \ldots, c_p$.

For example $\sqrt{3}v_1 + v_2$ is a linear combination, where v_1, v_2 are vectors in $\mathbb{R}^n$.

Definition 23 (Span). *If $v_1, v_2, \ldots, v_p$ are in $\mathbb{R}^n$, then the set of all linear combinations of $v_1, v_2, \ldots, v_p$ is denoted Span$\{v_1, v_2, \ldots, v_p\}$ and is called*

the subset of $\mathbb{R}^n$ *spanned by* $v_1, v_2, \ldots, v_p$. *That is,* $Span\{v_1, v_2, \ldots, v_p\}$ *is the collection of all vectors that can be written on the form*

$$c_1 v_1 + \cdots + c_p v_p$$

with $c_1, c_2, \ldots, c_p$ *scalars.*

The span is the set of all linear combinations.

Definition 24 (Column Space)**.** *The column space of an* $m \times n$ *matrix* X*, written as* $Col(X)$*, is the set of all linear combination of the Columns of* X*. If* $X = [a_1 \ldots a_n]$*, then*

$$Col(X) = Span\{a_1 \ldots a_n\}.$$

B Topology

Topology treats the properties of sets, this is for example of relevance because if we want to find the global minimum of a convex function by gradient descent we need the function domain to be convex.

Let S be an arbitrary set, where $S \subset [2]$.

Definition 25 (Closure)**.** *A point* x *is said to be in the closure of* S *(*cl$S$*), if* $S \cap N_\epsilon(x) \neq \varnothing$ *for every* $\epsilon > 0$.

Definition 26 (Closed)**.** *If* $S = \text{cl}S$ *then* S *is called closed.*

Definition 27 (Interior)**.** x *is said to be in the interior of* S*, denoted* int$S$*, if* $N_\epsilon(x) \subset S$ *for some* $\epsilon > 0$.

Definition 28 (Open)**.** *If* $S = intS$*, then* S *is called open.*

Definition 29 (Boundary)**.** *Boundary is denoted* dS*, if* $N_\epsilon(x)$ *contains one point in* S *and one outside for every* $\epsilon > 0$.

Definition 30 (Bounded)**.** *A set* S *is bounded if it is contained in a ball of a sufficiently large radius.*

Definition 31 (Compact)**.** *A compact set is both bounded and closed.*

Definition 32 (Distance function)**.** *A set X, whose elements we shall call points, is said to be a metric space if with any two points p and q of X where associated a real number $d(p,q)$, called the distance from p to q, such that*

1. $d(p,q) > 0$ *if* $p \neq q$; $d(p,p) = 0$.
2. $d(p,q) = d(q,p)$.
3. $d(p,q) \leq d(p,r) + d(r,q)$, *for any* $r \in X$.

Any function with these three properties is called a distance function, or metric [10].

Definition 33 (Neighbourhood)**.** *A neighbourhood is an open ball with centre p and radius $r > 0$ such that $N_r(p) = \{x | d(x,p) < r\}$.*