Machine Learning and Data Analytics
Module: Optimization

Dr. Manojkumar C. Ramteke, Dr. Hariprasad Kodamana, Dr. Agam Gupta

Department of Chemical Engineering

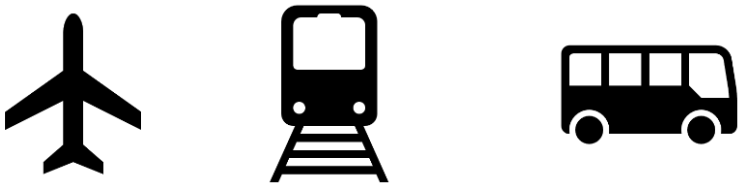
IIT Delhi

2024

What is Optimization?

In our day-to-day life, our entire decision making is subjected to some sort of optimization.

For instance, if you wanted to travel from Delhi to Mumbai, which travel mode you should select ? (SIMPLE INTIUTIVE)



Such optimization decision are taken intuitively (based on cost, time and comfort) in our mind

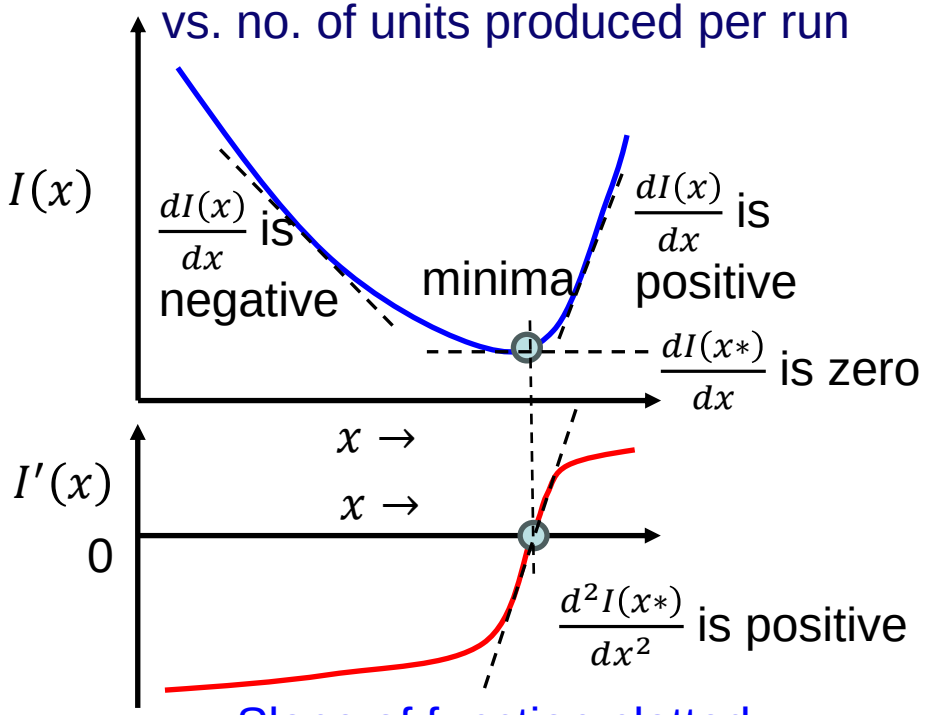
However, making such intuitive decision for engineering systems is difficult and one has to rely on suitable algorithms to simplify this decision making. This entire process is referred to as optimization and the algorithm used is referred to as optimization algorithm.

To solve such problems, a model (set of mathematical equations) of an underlying system is required which can give a state of system at any time for specific values of variables.

Concept of Optimization

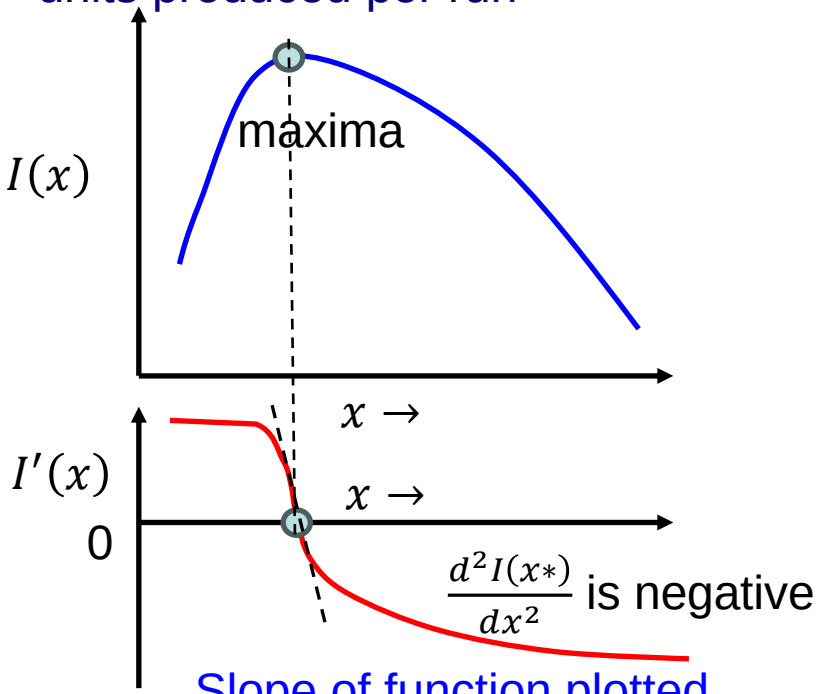
Suppose function I(x) varies with decision variable x, and we aim to find optimal value of I(x).

Function plotted with respect to variable: Variation in production cost vs. no. of units produced per run



Slope of function plotted with respect to variable

Function plotted with respect to variable: Variation in profit vs. no. of units produced per run



Slope of function plotted with respect to variable

Necessary Condition

Slope of I(x) vs. x = 0 at optimum

$$\frac{dI(x)}{dx} = 0$$

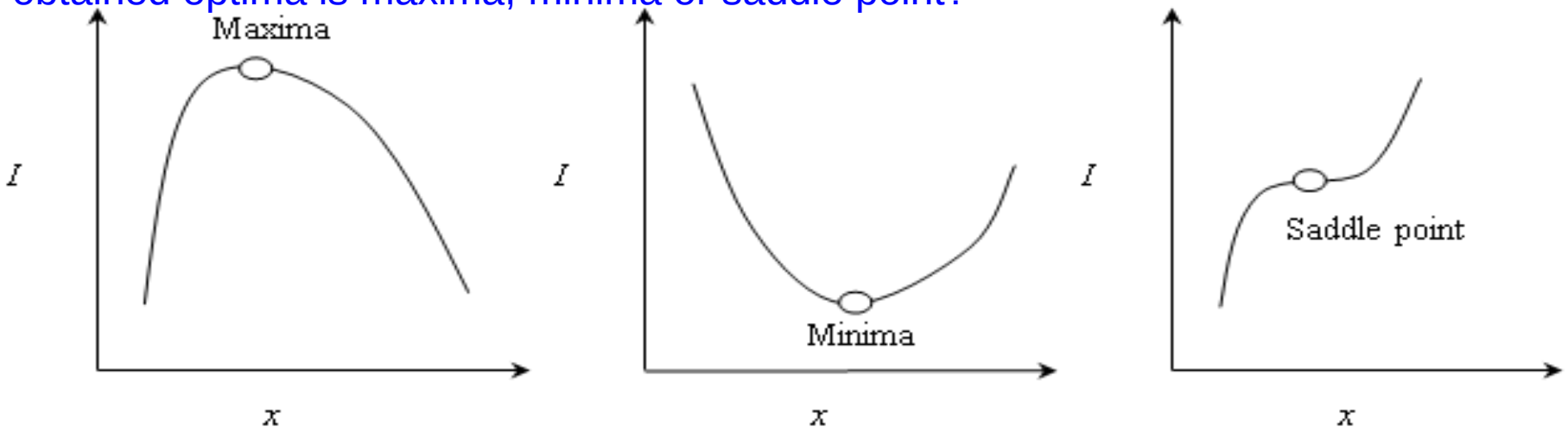
Sufficient Condition

Slope of I'(x) vs. x = positive for minimum
Slope of I'(x) vs. x = negative for maximum

$$\frac{d^2I(x)}{dx^2} > 0 \; for \; minima; \frac{d^2I(x)}{dx^2} < 0 \; for \; maxima;$$

Concept of Optimization

It may possible that first and second derivative are zero. How to decide whether the obtained optima is maxima, minima or saddle point?



Generalized Optimality Conditions: Continue differentiation for k no. of times till we get non-zero derivative (k should be > 2)

$$\frac{d^{k-1}I}{dx^{k-1}} = 0 \quad \begin{cases} \text{if } \dfrac{d^k I}{dx^k} > 0 \text{ and } k \text{ is even} \rightarrow \text{minima} \\ \text{if } \dfrac{d^k I}{dx^k} < 0 \text{ and } k \text{ is even} \rightarrow \text{maxima} \\ \text{if } \dfrac{d^k I}{dx^k} \neq 0 \text{ and } k \text{ is odd} \rightarrow \text{saddle point} \end{cases}$$

$$k = 2 \text{ if } \frac{d^k I}{dx^k} \neq 0; \text{ else } k = 3, 4, \ldots; \text{ continue till } \frac{d^k I}{dx^k} \neq 0$$

$$I = x^3$$

$$\frac{dI}{dx} = 3x^2 = 0, x = 0$$

$$\frac{d^2 I}{dx^2} = 6x\Big|_{x=0} = 0$$

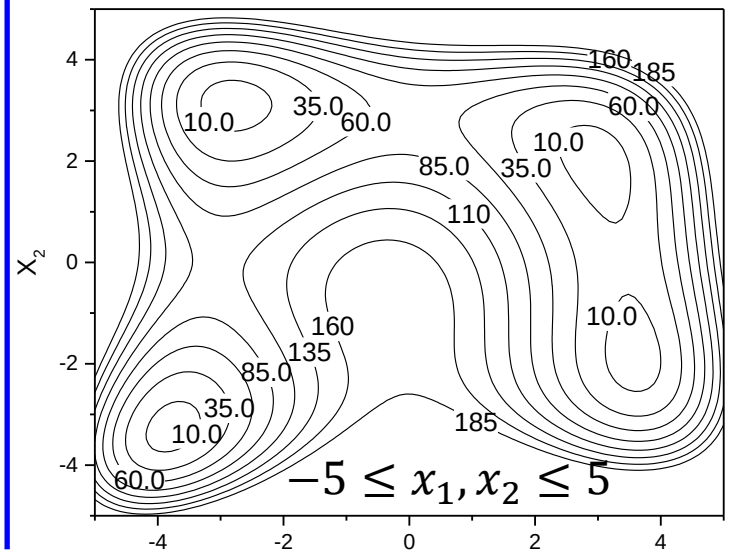
$$\frac{d^3 I}{dx^3} = 6\Big|_{x=0} \neq 0$$

Since k = 3, optimum is a saddle point

Concept of Optimization

For multi-variable objective function, $I(x_1, x_2)$, the necessary and sufficient conditions of optimality are as follows:

$$I(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$



Objective values are shown by contours

Necessary Condition:

Slope of $I(x_1, x_2)$ vs. $x_1 = 0$ and Slope of $I(x_1, x_2)$ vs. $x_2 = 0$ at optimum

$$\frac{\partial I(x_1, x_2)}{\partial x_1} = 0; \frac{\partial I(x_1, x_2)}{\partial x_2} = 0$$

Sufficient Condition:

Hessian Matrix $\nabla^2 I(x_1, x_2) = \begin{bmatrix} \frac{\partial^2 I(x_1, x_2)}{\partial x_1^2} & \frac{\partial^2 I(x_1, x_2)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 I(x_1, x_2)}{\partial x_1 \partial x_2} & \frac{\partial^2 I(x_1, x_2)}{\partial x_2^2} \end{bmatrix}$

Matrix comprising of double derivatives of the given function

If all eigen values, λ, of Hessian matrix ≥ 0 then the optimum is a minima

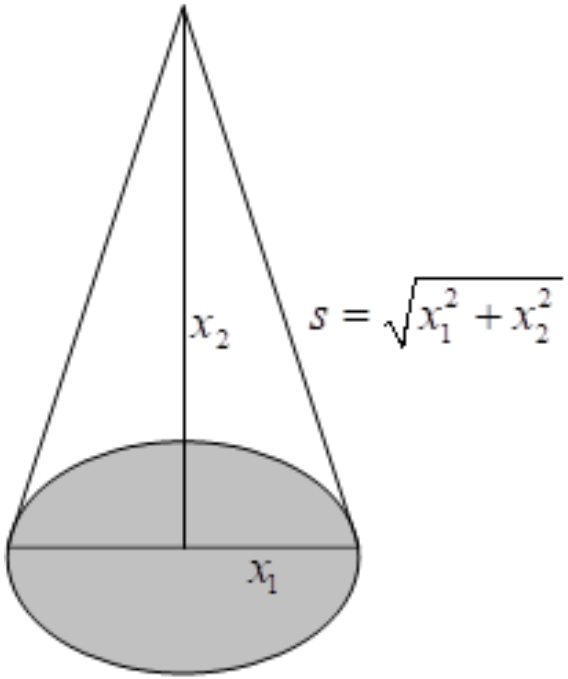
If all eigen values, λ, of Hessian matrix ≤ 0 then the optimum is a maxima

If some eigen values, λ, are positive and some are negative then the optimum is a saddle point

This can be proven from Taylor series Expansion.

Illustrative Example

Consider a simple system of a conical container having radius, x_1, of the base and height, x_2, as shown in Figure. Objective *I* is to minimize or maximize the volume, *V*, with constraints, $x_1 + x_2 = 10$.



$$\max or \min I(x_1, x_2) \equiv \frac{\pi x_1^2 x_2}{3}$$

Subjected to $x_1 + x_2 = 10$

From constraint, $x_1 = 10\text{-}x_2$

$$\max or \min I(x_2) \equiv \frac{\pi x_2 (10 - x_2)^2}{3}$$

To obtain maximum or minimum, derivative of the objective function with respect to variable x_2 should be equated to zero

$$\frac{dI}{dx_2} = 0$$

Illustrative Example

$$\frac{dI}{dx_2} = 0 = \frac{d}{dx_2}\left[\frac{\pi x_2 (10 - x_2)^2}{3}\right] = \frac{\pi}{3}\frac{d}{dx_2}\left[x_2\left(100 - 20x_2 + x_2^2\right)\right]$$

$$= \frac{\pi}{3}\frac{d}{dx_2}\left(100x_2 - 20x_2^2 + x_2^3\right) = \frac{\pi}{3}\left(100 - 40x_2 + 3x_2^2\right) = 0$$

$$\Rightarrow 3x_2^2 - 40x_2 + 100 = 0$$

$$\therefore x_2 = \frac{40 \pm \sqrt{1600 - 4\times 3\times 100}}{6} = \frac{40 \pm 20}{6} = 3.3333 \text{ or } 10$$

$$and\ I^{opt} = \frac{\pi x_2}{3}(10 - x_2)^2 = 155.06\Big|_{x_2^{opt}=3.3333}, \text{ or } 0\Big|_{x_2^{opt}=10}$$

Whether the obtained optimum is maximum, or minimum is decided by second derivative

$$\frac{d^2 I}{dx_2^2} = \frac{d}{dx_2}\left[\frac{\pi}{3}\left(100 - 40x_2 + 3x_2^2\right)\right] = \frac{\pi}{3}\left(-40 + 6x_2\right)$$

$$i.e.,\ \frac{d^2 I}{dx_2^2}\Big|_{x_2^{opt}=3.3333} = -\frac{20\pi}{3} < 0 \rightarrow \text{the selected optimum is a maxima}$$

$$and,\ \frac{d^2 I}{dx_2^2}\Big|_{x_2^{opt}=10} = \frac{20\pi}{3} > 0 \rightarrow \text{the selected optimum is a minima}$$

Generalized Optimization Formulation

Objective Functions: These are selected based on specific characteristics of the interest (e.g., maximising customer satisfaction).

$$\min or \max I_i(x_1, x_2, \cdots, x_n) = f_i(x_1, x_2, \cdots, x_n); i = 1, 2, \cdots, m$$

Inequality Constraints:

$$g_j(x_1, x_2, \cdots, x_n) > 0; j = 1, 2, \cdots, J$$

Equality Constraints:

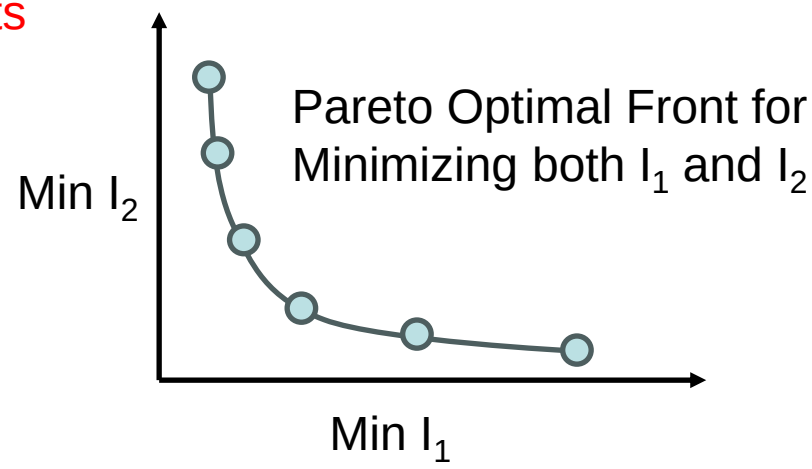
$$h_k(x_1, x_2, \cdots, x_n) = 0; k = 1, 2, \cdots, K; where\ K < n$$

These are selected based on various restrictions based on safety, design, operational practices or social requirements

Bounds:

$$x_l^{Low} \leq x_l \leq x_l^{High}; l = 1, 2, \cdots, n$$

These are selected based on the given operating ranges of the variables



1) In single objective optimization, m = 1. Usually, single optimal solution is obtained.
2) In multi-objective optimization, m > 1. Typically, multiple equally good solutions are obtained. All these solutions are referred as Pareto optimal solutions. Decision maker selects one of these as operating solution based on his experience.

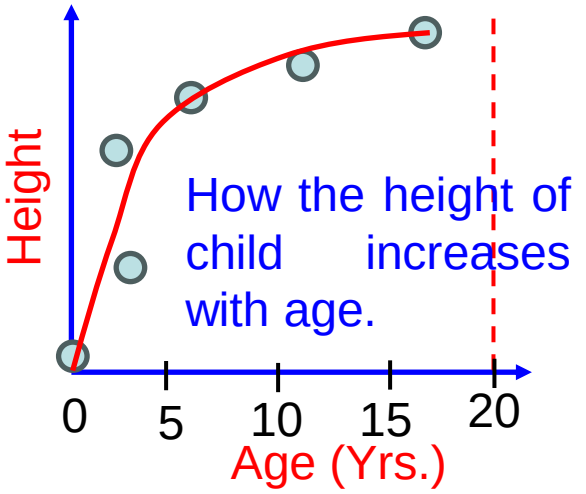
Use of Optimization in Machine Learning

1) In machine learning, a specific form of model is assumed initially. For instance, either linear ($y_{i,estimated} = \theta_1 x_{1,i} + \theta_2 x_{2,i} + \theta_3 x_{3,i} + \cdots + \theta_M x_{M,i}$) or polynomial equation ($y_{i,estimated} = \theta_1 + \theta_2 z_i + \theta_3 z_i^2 + \cdots + \theta_M z_i^{M-1}$) is assumed in regression. In Neural network and Logistic regression, a linear equation operated with non-linear activation function is assumed.
2) The parameters of this assumed model equations ($\theta_1, \theta_2, \theta_3, \cdots, \theta_M$) are then tuned using the available data (number of data points N > M). This step is called Training. The trained model is then tested on unknown data to confirm the suitability of the developed model (using metrics such as R^2, RMSE, etc.). Once the suitability of the trained model is confirmed, it is used for prediction or classification problems.
3) The machine learning problem in which the new data is required to be put in specific class (i.e., the outputs are discrete such as whether the customer will buy the SUV or not) is referred as classification. However, the numerical output value is predicted for new data point in the prediction problem (e.g., Predicting the value of old car based on its age, kilometers travelled, make, etc.).
4) All machine learning models are trained using optimization. For instance, in regression and neural network, the model training involves optimizing the M parameters ($\theta_1, \theta_2, \theta_3, \cdots, \theta_M$) of the assumed model to minimize the discrepancy between the model predicted output and actual output for all N data points.

$$\sum_{i=1}^{N} \left(y_{i,actual} - y_{i,estimated}\right)^2$$

Curve Fitting Optimization (Regression)

1) In curve fitting optimization, the objective is to obtain the best fit of the model equation for the given data points.
2) For all curve fitting exercises, the number of data points should be greater than or equal to the number of coefficients used in the model equation.
3) An example is to fit a polynomial form of the model equation comprising of M coefficients to N data points, with N ≥ M.



$$y_{i,estimated} = \theta_1 + \theta_2 z_i + \theta_3 z_i^2 + ... + \theta_M z_i^{M-1}$$

$$y_{i,estimated} = \sum_{j=1}^{M} \theta_j x_{j,i};\ x_{1,i} = 1, x_{2,i} = z_i, x_{3,i} = z_i^2, \ldots, x_{M,i} = z_i^{M-1}$$

Measured

Estimated

The objective function is to minimize the sum of square errors between the *measured* outputs and the model prediction

$$\min I = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} \theta_j x_{j,i} \right)^2$$

Partially differentiate the objective function with respect to each of the coefficients and equate these to zero to obtain *M* equations for the *M* coefficients

$$\frac{\partial I}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} \theta_j x_{j,i} \right)^2 = 0; \qquad j = 1, 2, \ldots, M$$

$$\frac{\partial I}{\partial \theta_j} = \sum_{i=1}^{N} 2 \left(y_i - \sum_{j=1}^{M} \theta_j x_{j,i} \right) \times -x_{j,i} = 0$$

$$\frac{\partial^2 I}{\partial \theta_j^2} = \sum_{i=1}^{N} 2 x_{j,i}^2 = +ve$$

$$\sum_{i=1}^{N} y_i x_{j,i} - \theta_1 \sum_{i=1}^{N} x_{1,i} x_{j,i} - \theta_2 \sum_{i=1}^{N} x_{2,i} x_{j,i} - \ldots - \theta_M \sum_{i=1}^{N} x_{M,i} x_{j,i} = 0; \ j = 1, 2, \ldots, M$$

Curve Fitting Optimization

$$\sum_{i=1}^{N} y_i x_{j,i} - \theta_1 \sum_{i=1}^{N} x_{1,i} x_{j,i} - \theta_2 \sum_{i=1}^{N} x_{2,i} x_{j,i} - \ldots - \theta_M \sum_{i=1}^{N} x_{M,i} x_{j,i} = 0;\ j = 1, 2, \ldots, M$$

These *M* simultaneous equations can be solved to estimate the *M* coefficients, $\theta = [\theta_1, \theta_2, \ldots\ldots, \theta_M]$

In a vector form

$$\left(\boldsymbol{X}^T \boldsymbol{y}\right) - \left(\boldsymbol{X}^T \boldsymbol{X}\right)\boldsymbol{\theta} = 0; or\ \boldsymbol{\theta} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T \boldsymbol{y}\right)$$

X has N rows (no. of data points) and M columns (no. of features)

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_M \end{bmatrix}; \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix}; \boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{2,1} & . & . & . & x_{M,1} \\ x_{1,2} & x_{2,2} & . & . & . & x_{M,2} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ x_{1,N} & x_{2,N} & . & . & . & x_{M,N} \end{bmatrix} = \begin{bmatrix} 1 & (z)_1 & . & . & . & (z^{M-1})_1 \\ 1 & (z)_2 & . & . & . & (z^{M-1})_2 \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ 1 & (z)_N & . & . & . & (z^{M-1})_N \end{bmatrix}$$

An analytical closed form solution is obtained in this case. However, in several regressions such as LASSO (Least absolute shrinkage and selection operator) and Logistic regression obtaining such closed form solution is not possible and one has to rely on numerical optimization methods.

Illustrative Example

The Table below represents four experimental data points, i = 1, 2, 3 and 4 (N = 4). Determine a suitable parameters of empirical model, $y = a_0 + a_1 z + a_2 z^2$, to represent the data. Determine the coefficients if the curve is required to pass through the origin (0, 0).

i	**1**	**2**	**3**	**4**
$\mathbf{z_i}$	2	4	6	8
$\mathbf{y_i}$	0.078	0.111	0.153	0.209

$$\boldsymbol{\theta} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}; \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_4 \end{bmatrix}; \boldsymbol{X} = \begin{bmatrix} 1 & (z)_1 & (z^2)_1 \\ 1 & (z)_2 & (z^2)_2 \\ . & . & . \\ . & . & . \\ 1 & (z)_4 & (z^2)_4 \end{bmatrix}$$

Illustrative Example

$$\left(\boldsymbol{X}^T\boldsymbol{X}\right)=\begin{bmatrix}1 & 1 & \ldots & 1\\ (z)_1 & (z)_2 & \ldots & (z)_4\\ (z^2)_1 & (z^2)_2 & \ldots & (z^2)_4\end{bmatrix}\times\begin{bmatrix}1 & (z)_1 & (z^2)_1\\ 1 & (z)_2 & (z^2)_2\\ . & . & .\\ . & . & .\\ 1 & (z)_4 & (z^2)_4\end{bmatrix}=\begin{bmatrix}4 & \sum_{i=1}^{4}(z)_i & \sum_{i=1}^{4}(z^2)_i\\ \sum_{i=1}^{4}(z)_i & \sum_{i=1}^{4}(z^2)_i & \sum_{i=1}^{4}(z^3)_i\\ \sum_{i=1}^{4}(z^2)_i & \sum_{i=1}^{4}(z^3)_i & \sum_{i=1}^{4}(z^4)_i\end{bmatrix}$$

$$=\begin{bmatrix}4 & 20 & 120\\ 20 & 120 & 800\\ 120 & 800 & 5664\end{bmatrix}$$

$$\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}=\begin{bmatrix}7.75 & -3.375 & 0.3125\\ -3.375 & 1.6125 & -0.15625\\ 0.3125 & -0.15625 & 0.015625\end{bmatrix}$$

Illustrative Example

$$\left(\boldsymbol{X}^T\boldsymbol{y}\right)=\begin{bmatrix}1 & 1 & \ldots & 1\\ (z)_1 & (z)_2 & \ldots & (z)_4\\ \left(z^2\right)_1 & \left(z^2\right)_2 & \ldots & \left(z^2\right)_4\end{bmatrix}\times\begin{bmatrix}y_1\\ y_2\\ .\\ .\\ .\\ y_4\end{bmatrix}=\begin{bmatrix}\sum_{i=1}^{4} y_i\\ \sum_{i=1}^{4} z_i y_i\\ \sum_{i=1}^{4} z_i^2 y_i\end{bmatrix}=\begin{bmatrix}0.551\\ 3.1900\\ 20.9720\end{bmatrix}$$

$$\boldsymbol{\theta}=\begin{bmatrix}a_0\\ a_1\\ a_2\end{bmatrix}=\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{y}\right)=\begin{bmatrix}7.75 & -3.375 & 0.3125\\ -3.375 & 1.6125 & -0.15625\\ 0.3125 & -0.15625 & 0.015625\end{bmatrix}\times\begin{bmatrix}0.551\\ 3.1900\\ 20.9720\end{bmatrix}=\begin{bmatrix}0.05775\\ 7.375\times10^{-3}\\ 1.4375\times10^{-3}\end{bmatrix}$$

If the curve is required to pass through the origin, then the coefficient, a_0, is forced to be zero.

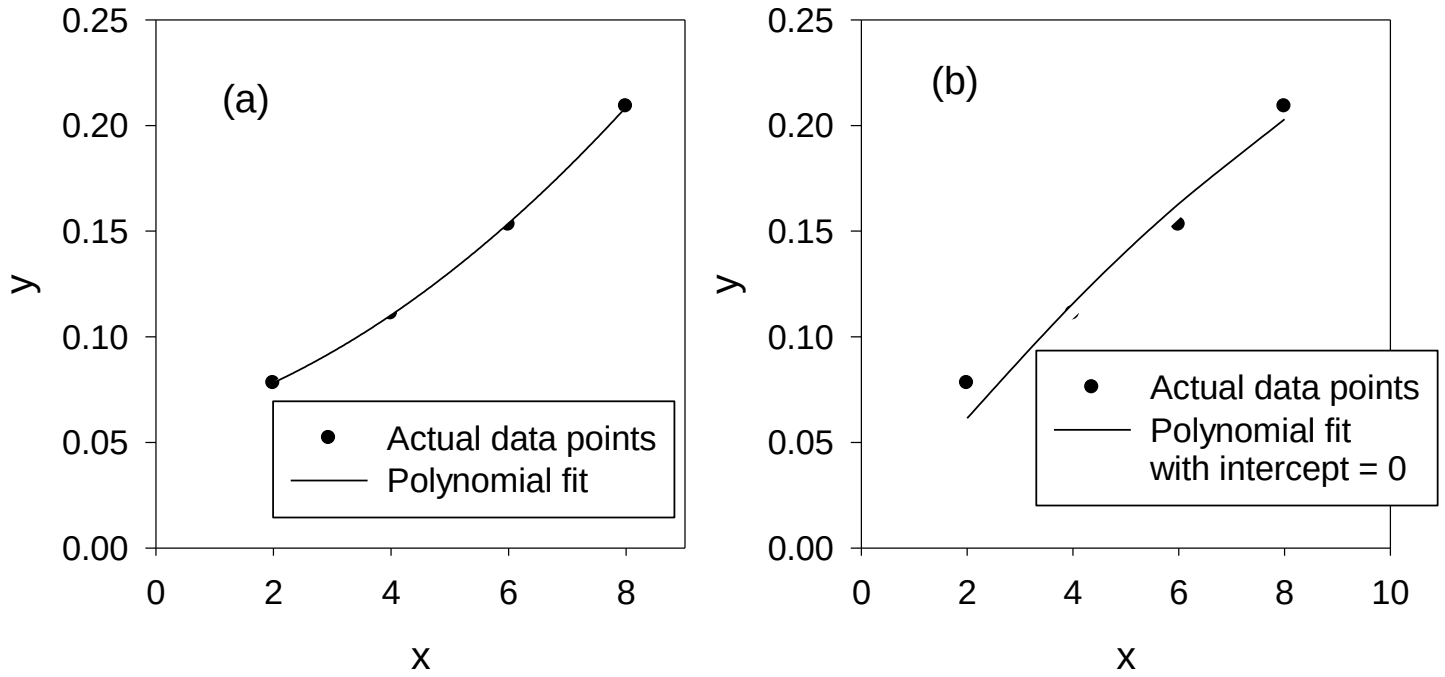
$$\boldsymbol{\theta}=\begin{bmatrix}a_1\\ a_2\end{bmatrix};\ \boldsymbol{y}=\begin{bmatrix}y_1\\ y_2\\ .\\ .\\ .\\ y_4\end{bmatrix};\ \boldsymbol{X}=\begin{bmatrix}(z)_1 & \left(z^2\right)_1\\ (z)_2 & \left(z^2\right)_2\\ . & .\\ . & .\\ (z)_4 & \left(z^2\right)_4\end{bmatrix}$$

Illustrative Example

$$\left(\boldsymbol{X}^T\boldsymbol{X}\right)=\begin{bmatrix}\sum_{i=1}^{4}\left(z^2\right)_i & \sum_{i=1}^{4}\left(z^3\right)_i\\ \sum_{i=1}^{4}\left(z^3\right)_i & \sum_{i=1}^{4}\left(z^4\right)_i\end{bmatrix}=\begin{bmatrix}120 & 800\\ 800 & 5664\end{bmatrix};\left(\boldsymbol{X}^T\boldsymbol{y}\right)=\begin{bmatrix}\sum_{i=1}^{4}z_iy_i\\ \sum_{i=1}^{4}z_i^2y_i\end{bmatrix}=\begin{bmatrix}3.19\\ 20.972\end{bmatrix}$$

$$\begin{bmatrix}a_1\\ a_2\end{bmatrix}=\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\times\left(\boldsymbol{X}^T\boldsymbol{y}\right)=\begin{bmatrix}120 & 800\\ 800 & 5664\end{bmatrix}^{-1}\times\begin{bmatrix}3.19\\ 20.972\end{bmatrix}=\begin{bmatrix}0.0325\\ -8.911\times10^{-4}\end{bmatrix}$$

(a) Polynomial fit and (b) fit of a polynomial passing through the origin



Multi-variable Gradient Based Techniques

1) These methods require the first- (or higher-) order derivatives of the function to direct the search. All methods used in this class commonly use the following search pattern:

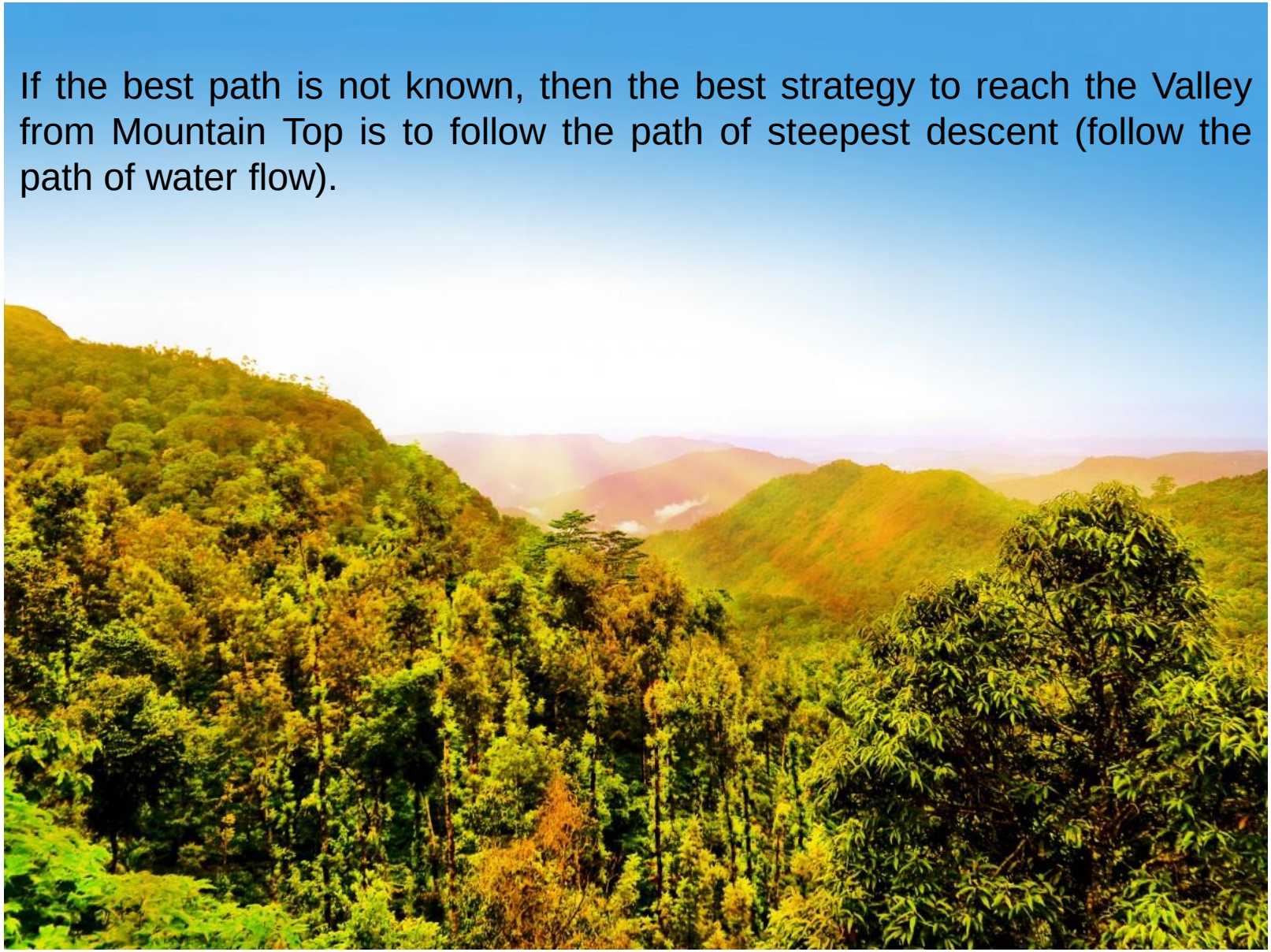
Generalized Form: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{s}(\boldsymbol{x}_k)$

Where, $\boldsymbol{x}_k$, is the current estimate of the optimum $\boldsymbol{x}^*$, $\boldsymbol{x}_{k+1}$ is the new estimate, α_k is the step size (learning rate) and $\boldsymbol{s}(\boldsymbol{x}_k)$ is the search direction of the kth step. In different techniques, α_k and $\boldsymbol{s}(\boldsymbol{x}_k)$ are calculated differently.

1) Advantage of these methods is the higher speed of convergence. The speed of convergence is extremely crucial factor while modeling the large size data which is typically the case in Machine Learning.
2) The most popular derivative based methods are: (a) Gradient Descent and (b) Newton's Method.
3) Almost all optimizers used in Machine learning are modified form of Gradient Descent.

Note that all Bold Letters are Vectors: $\boldsymbol{x}_k = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}_k ; \boldsymbol{s}(\boldsymbol{x}_k) = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}_{\boldsymbol{x}_k}$

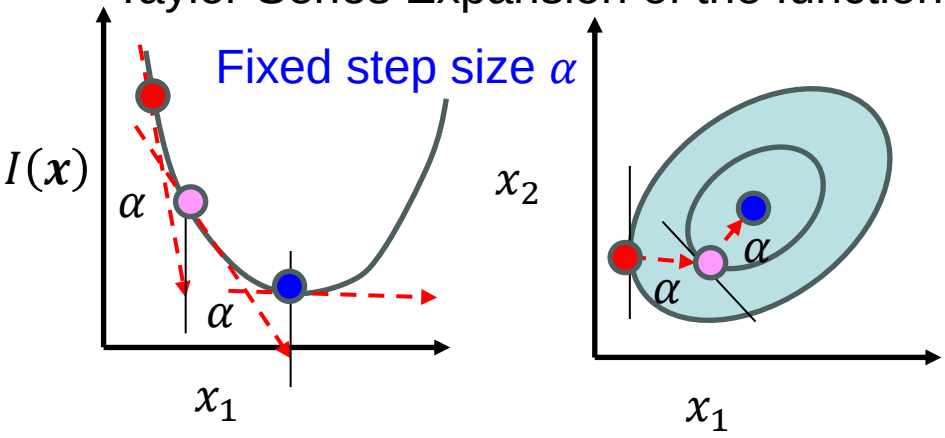
Gradient Descent

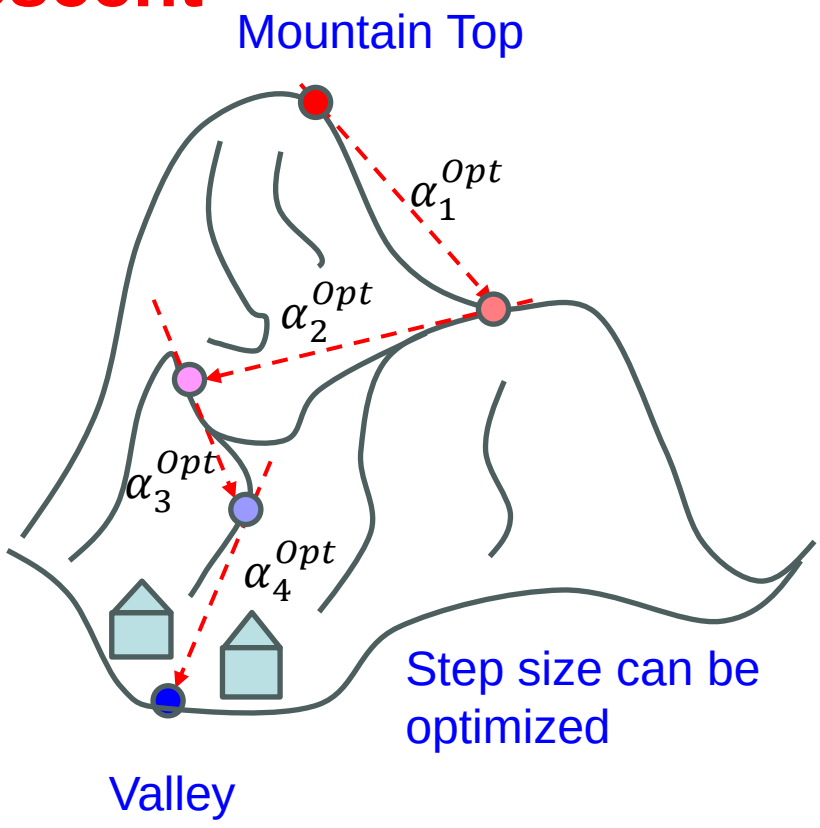


Free image taken from https://pxhere.com/en/photo/842624

Gradient Descent

1) The heuristics of water flowing from the mountain top to Valley is mathematically simulated in Gradient Descent for solving minimization problem.
2) This can be derived using First-Order Taylor Series Expansion of the function.





Taylor Series Expansion: $I(\boldsymbol{x}_{k+1}) = I(\boldsymbol{x}_k) + \nabla I(\boldsymbol{x}_k)^T(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \cdots$

For minimization problem, to have steepest decrease from $I(\boldsymbol{x}_k)$ to $I(\boldsymbol{x}_{k+1})$, one need to have largest (-negative) value of $\nabla I(\boldsymbol{x}_k)$.

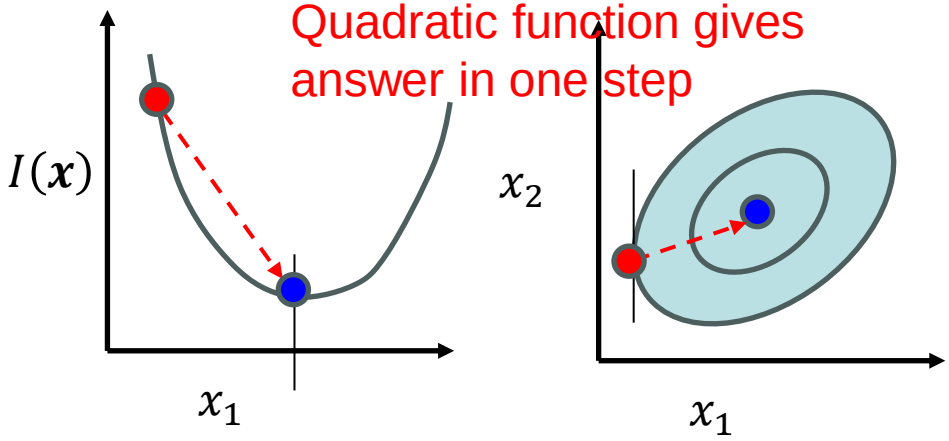
$$\boldsymbol{s}(\boldsymbol{x}_k) = -\nabla I(\boldsymbol{x}_k) \quad \Rightarrow \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla I(\boldsymbol{x}_k)$$

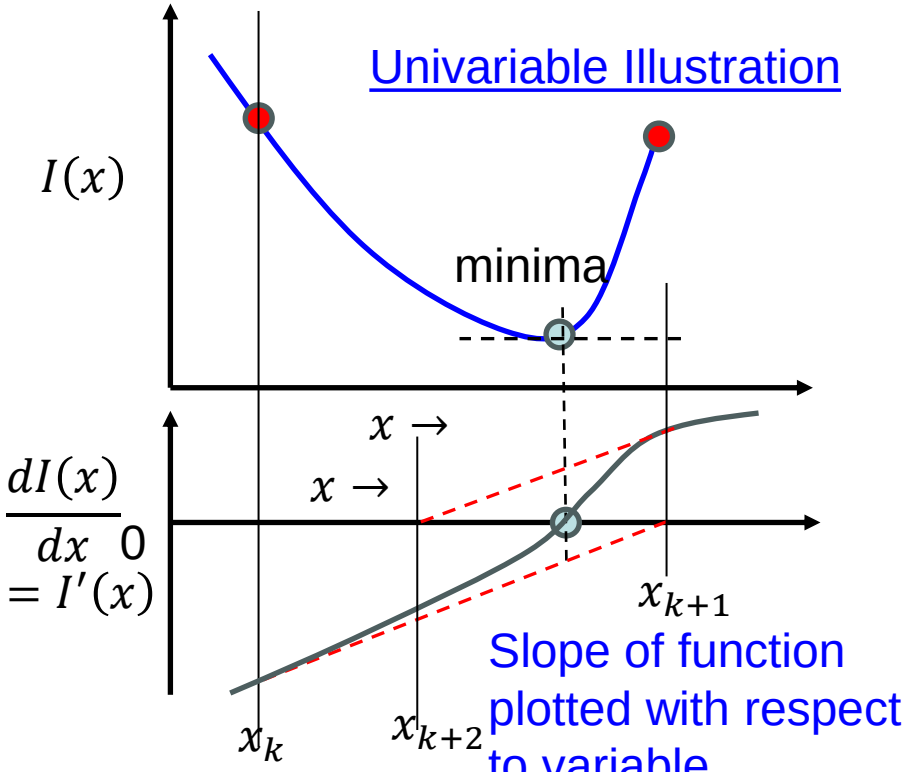
Step size need to obtained by optimization (by putting $\boldsymbol{x}_{k+1}$ in I and taking $\frac{dI(\boldsymbol{x}_{k+1})}{d\alpha_k} = 0$) or a fixed value is used

1) In Machine Learning application, fixed value of step size (learning rate) is used.

Newton's Method

1) It follows the search direction of slope of slope curve till zero line.
2) This can be derived using Second-Order Taylor Series Expansion of the function.





Second-Order Taylor Series Expansion:

$$I(\boldsymbol{x}_{k+1}) = I(\boldsymbol{x}_k) + \nabla I(\boldsymbol{x}_k)^T(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)^T \nabla^2 I(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \cdots$$

$$\frac{dI(x_{k+1})}{dx_{k+1}} = 0 + \nabla I(\boldsymbol{x}_k) + \nabla^2 I(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) = 0 \Longrightarrow \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\nabla^2 I(\boldsymbol{x}_k)]^{-1} \nabla I(\boldsymbol{x}_k)$$

$$\boldsymbol{s}(\boldsymbol{x}_k) = -[\nabla^2 I(\boldsymbol{x}_k)]^{-1} \nabla I(\boldsymbol{x}_k); \alpha_k = 1$$

Always, the step size of 1 is used and inverse of Hessian Matrix is required.

1) In Machine Learning application, Newton's method is not used as obtaining the inverse of Hessian matrix (N×N) of large size data (N) is computationally expensive.
2) Same expression of Gradient Descent can be obtained by putting $\nabla^2 I(\boldsymbol{x}_k) = 1/\alpha_k$.

Multi-variable Gradient Based Techniques

Gradient Descent

1) For a given point, $\boldsymbol{x}_0 = [x_1, x_2, \cdots, x_N]_0^T$, calculate the gradient

$$\nabla I(\boldsymbol{x}_0) = \begin{bmatrix} \partial I/\partial x_1 \\ \vdots \\ \partial I/\partial x_N \end{bmatrix}_{x_0}$$

2) Generate the new point as:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \alpha_0 \nabla I(\boldsymbol{x}_0)$$

3) Choose α_0 such that $I(\boldsymbol{x}_1) \rightarrow$ minimum or select the fixed user-defined value α.

Newton's Method

1) For a given point, $\boldsymbol{x}_0 = [x_1, x_2, \cdots, x_N]_0^T$, calculate the gradient

$$\nabla I(\boldsymbol{x}_0) = \begin{bmatrix} \partial I/\partial x_1 \\ \vdots \\ \partial I/\partial x_N \end{bmatrix}_{\boldsymbol{x}_0}$$

2) Generate the new point as:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - [\nabla^2 I(\boldsymbol{x}_0)]^{-1} \nabla I(\boldsymbol{x}_0)$$

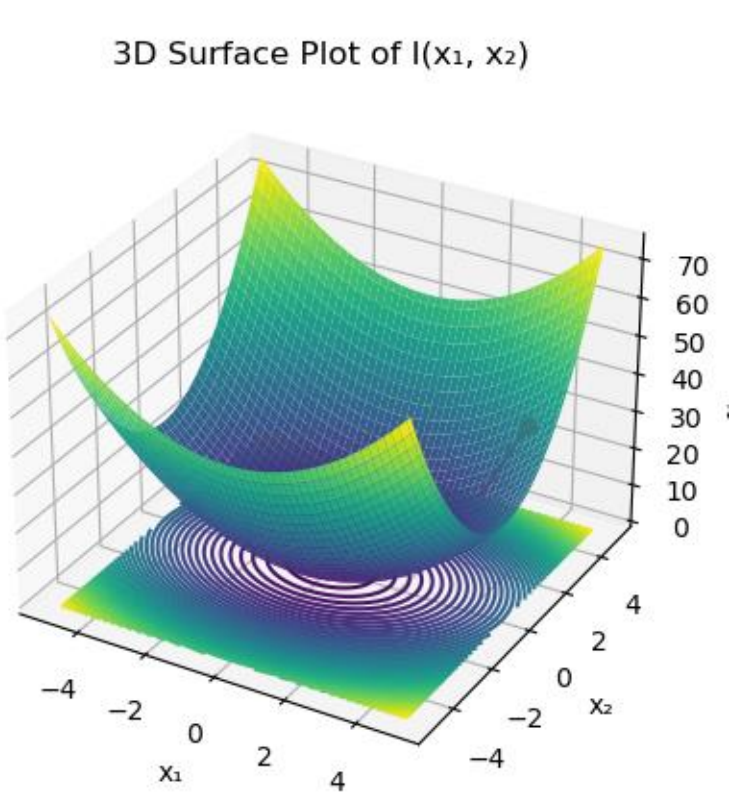
3) Choose $\boldsymbol{s}(\boldsymbol{x}_k)$ as (-) inverse of the Hessian into gradient and $\alpha_0 = 1$.

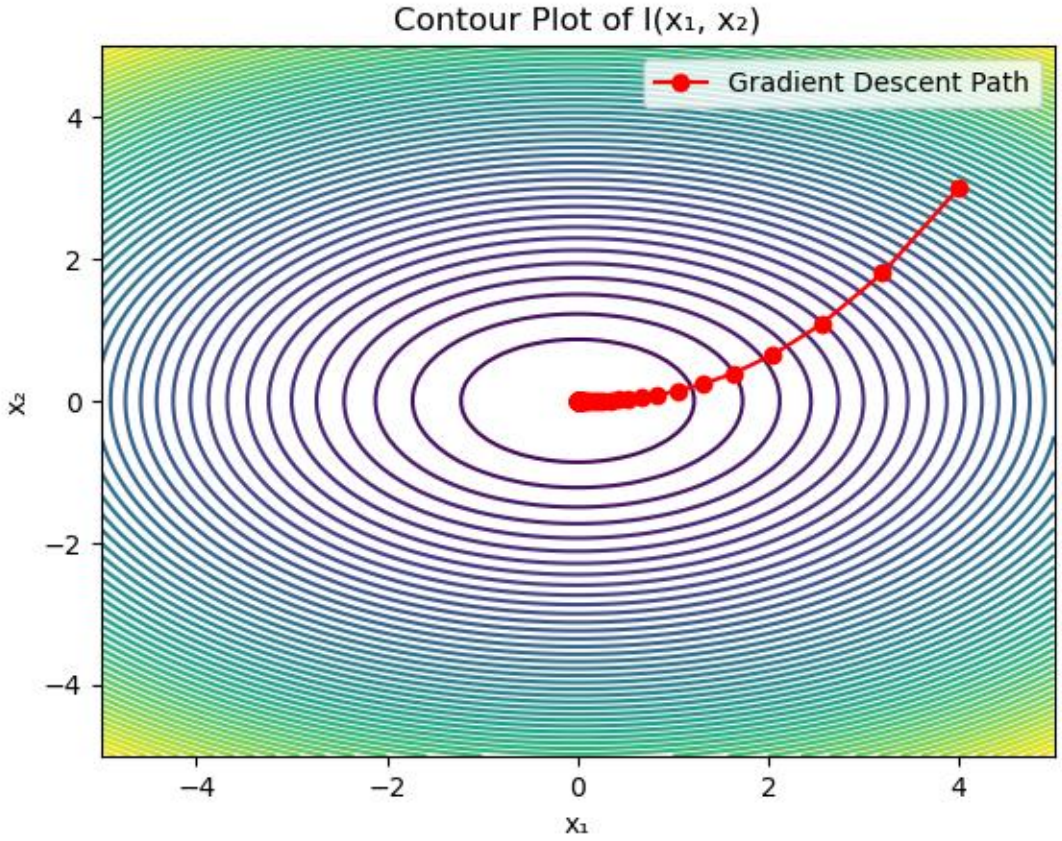
$$\boldsymbol{s}(\boldsymbol{x}_k) = -[\nabla^2 I(\boldsymbol{x}_k)]^{-1} \nabla I(\boldsymbol{x}_k); \alpha_k = 1$$

$$\boldsymbol{s}(\boldsymbol{x}_0) = -\begin{bmatrix} \partial^2 I/\partial x_1^2 & \cdots & \partial^2 I/\partial x_1 \partial x_N \\ \vdots & \ddots & \vdots \\ \partial^2 I/\partial x_N \partial x_1 & \cdots & \partial^2 I/\partial x_N^2 \end{bmatrix}_{\boldsymbol{x}_0}^{-1} \begin{bmatrix} \partial I/\partial x_1 \\ \vdots \\ \partial I/\partial x_N \end{bmatrix}_{\boldsymbol{x}_0}$$

Multi-variable Gradient Based Techniques

Example 1: Minimize $I = x_1^2 + 2x_2^2$, with the initial point $\boldsymbol{x}_0 = [x_1, x_2]_0^T = [2, 2]$ using the Gradient Descent with (a) fixed step size (α) of 0.2 and (b) with optimal step size calculated in each iterations (α_k^{opt}). Carry out two iterations for $-5 < x_1, x_2 < 5$.





Multi-variable Gradient Based Techniques

Example 1: Minimize $I = x_1^2 + 2x_2^2$, with the initial point $\boldsymbol{x}_0 = [x_1, x_2]_0^T = [2, 2]$ using the Gradient Descent with (a) fixed step size (α) of 0.2 and (b) with optimal step size calculated in each iterations (α_k^{opt}). Carry out two iterations for $-5 < x_1, x_2 < 5$.

Part (a)

$I = x_1^2 + 2x_2^2$ $\quad \frac{\partial I(x_1, x_2)}{\partial x_1} = 2x_1; \frac{\partial I(x_1, x_2)}{\partial x_2} = 4x_2$

Iteration 1: $\left[\frac{\partial I(x_1, x_2)}{\partial x_1}\right]_0 = 2 \times 2 = 4; \left[\frac{\partial I(x_1, x_2)}{\partial x_2}\right]_0 = 4 \times 2 = 8$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla I(\boldsymbol{x}_k)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 - \alpha \begin{bmatrix} \frac{\partial I(x_1, x_2)}{\partial x_1} \\ \frac{\partial I(x_1, x_2)}{\partial x_2} \end{bmatrix}_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.2 \times \begin{bmatrix} 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 0.8 \\ 1.6 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.4 \end{bmatrix}$$

Selecting the step size is crucial. Higher value may lead to oscillations around optimum point and low value takes large number of iterations to reach optimum.

Iteration 2: $\left[\frac{\partial I(x_1, x_2)}{\partial x_1}\right]_1 = 2 \times 1.2 = 2.4; \left[\frac{\partial I(x_1, x_2)}{\partial x_2}\right]_1 = 4 \times 0.4 = 1.6$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 - \alpha \begin{bmatrix} \frac{\partial I(x_1, x_2)}{\partial x_1} \\ \frac{\partial I(x_1, x_2)}{\partial x_2} \end{bmatrix}_1 = \begin{bmatrix} 1.2 \\ 0.4 \end{bmatrix} - 0.2 \times \begin{bmatrix} 2.4 \\ 1.6 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.4 \end{bmatrix} - \begin{bmatrix} 0.48 \\ 0.32 \end{bmatrix} = \begin{bmatrix} 0.72 \\ 0.08 \end{bmatrix}$$

The obtained solution is close to analytical optimal solution $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{opt} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Multi-variable Gradient Based Techniques

Part (b)

$I = x_1^2 + 2x_2^2$ $\qquad \frac{\partial I(x_1, x_2)}{\partial x_1} = 2x_1; \frac{\partial I(x_1, x_2)}{\partial x_2} = 4x_2$

Iteration 1: $\left[\frac{\partial I(x_1, x_2)}{\partial x_1}\right]_0 = 2 \times 2 = 4; \left[\frac{\partial I(x_1, x_2)}{\partial x_2}\right]_0 = 4 \times 2 = 8$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla I(\boldsymbol{x}_k)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 - \alpha_0 \begin{bmatrix} \frac{\partial I(x_1, x_2)}{\partial x_1} \\ \frac{\partial I(x_1, x_2)}{\partial x_2} \end{bmatrix}_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \alpha_0 \times \begin{bmatrix} 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 - 4\alpha_0 \\ 2 - 8\alpha_0 \end{bmatrix}$$

Additional Step in each iteration for Obtaining Optimum Step Size:

Putting in I: $I = x_1^2 + 2x_2^2 = (2 - 4\alpha_0)^2 + 2\,(2 - 8\alpha_0)^2$

Having additional step to obtain optimum step size α_k^{opt} is often cumbersome and therefore is not used in Machine Learning.

$$\frac{\partial I}{\partial \alpha_0} = -8 \times (2 - 4\alpha_0) - 32 \times (2 - 8\alpha_0) = 0 \Longrightarrow 10 - 36\alpha_0 = 0 \Longrightarrow \alpha_0^{opt} = 0.2777$$

$$\Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 = \begin{bmatrix} 2 - 4\alpha_0 \\ 2 - 8\alpha_0 \end{bmatrix} = \begin{bmatrix} 0.8892 \\ -0.2216 \end{bmatrix}$$

Multi-variable Gradient Based Techniques

Part (b) Continued

$I = x_1^2 + 2x_2^2$ $\quad \frac{\partial I(x_1,x_2)}{\partial x_1} = 2x_1; \frac{\partial I(x_1,x_2)}{\partial x_2} = 4x_2$ $\quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 = \begin{bmatrix} 0.8892 \\ -0.2216 \end{bmatrix}$

Iteration 2: $\left[\frac{\partial I(x_1,x_2)}{\partial x_1}\right]_1 = 2 \times 0.8892 = 1.7784; \left[\frac{\partial I(x_1,x_2)}{\partial x_2}\right]_1 = 4 \times -0.2216 = -0.8864$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla I(\boldsymbol{x}_k)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 - \alpha_1 \begin{bmatrix} \frac{\partial I(x_1,x_2)}{\partial x_1} \\ \frac{\partial I(x_1,x_2)}{\partial x_2} \end{bmatrix}_1 = \begin{bmatrix} 0.8892 \\ -0.2216 \end{bmatrix} - \alpha_1 \times \begin{bmatrix} 1.7784 \\ -0.8864 \end{bmatrix} = \begin{bmatrix} 0.8892 - 1.7784\alpha_1 \\ -0.2216 + 0.8864\alpha_1 \end{bmatrix}$$

Additional Step in each iteration for Obtaining Optimum Step Size:

Putting in I: $I = x_1^2 + 2x_2^2 = (0.8892 - 1.7784\alpha_1)^2 + 2\,(-0.2216 + 0.8864\alpha_1)^2$

$$\frac{\partial I}{\partial \alpha_1} = -3.5568 \times (0.8892 - 1.7784\alpha_1) + 3.5456 \times (-0.2216 + 0.8864\alpha_1) = 0$$

$$-3.9484 + 9.4682\alpha_1 = 0 \Longrightarrow \alpha_1^{opt} = 0.4170 \Longrightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_2 = \begin{bmatrix} 0.8892 - 1.7784\alpha_1 \\ -0.2216 + 0.8864\alpha_1 \end{bmatrix} = \begin{bmatrix} 0.1476 \\ 0.1480 \end{bmatrix}$$

The obtained solution is better than that obtained with fixed step size of 0.2 and is close to analytical optimal solution

Analytical Solution = $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{opt} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Multi-variable Gradient Based Techniques

Example 2: Minimize $I = x_1^2 + 2x_2^2$, with the initial point $\boldsymbol{x}_0 = [x_1, x_2]_0^T = [2, 2]$ using the Newton's Method. Carry out two iterations for $-5 < x_1, x_2 < 5$.

$$\frac{\partial I(x_1,x_2)}{\partial x_1} = 2x_1; \frac{\partial I(x_1,x_2)}{\partial x_2} = 4x_2; \frac{\partial^2 I(x_1,x_2)}{\partial x_1^2} = 2; \frac{\partial^2 I(x_1,x_2)}{\partial x_1 \partial x_2} = 0; \frac{\partial^2 I(x_1,x_2)}{\partial x_2^2} = 4$$

Iteration 1: $\left[\frac{\partial I(x_1,x_2)}{\partial x_1}\right]_0 = 2 \times 2 = 4; \left[\frac{\partial I(x_1,x_2)}{\partial x_2}\right]_0 = 4 \times 2 = 8$

$$\left[\frac{\partial^2 I(x_1,x_2)}{\partial x_1^2}\right]_0 = 2; \left[\frac{\partial^2 I(x_1,x_2)}{\partial x_1 \partial x_2}\right]_0 = 0; \left[\frac{\partial^2 I(x_1,x_2)}{\partial x_2^2}\right]_0 = 4$$

Global Optimum Solution is Obtained in One Step as the Function is Quadratic in nature. Often finding the inverse of Hessian Matrix is Difficult. Sometimes the method diverges.

$$\nabla^2 I(x_1,x_2)_0 = \begin{bmatrix} \partial^2 I(x_1,x_2)/\partial x_1^2 & \partial^2 I(x_1,x_2)/(\partial x_1 \partial x_2) \\ \partial^2 I(x_1,x_2)/(\partial x_1 \partial x_2) & \partial^2 I(x_1,x_2)/\partial x_2^2 \end{bmatrix}_0 = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}_0$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\nabla^2 I(\boldsymbol{x}_k)]^{-1} \nabla I(\boldsymbol{x}_k)$$

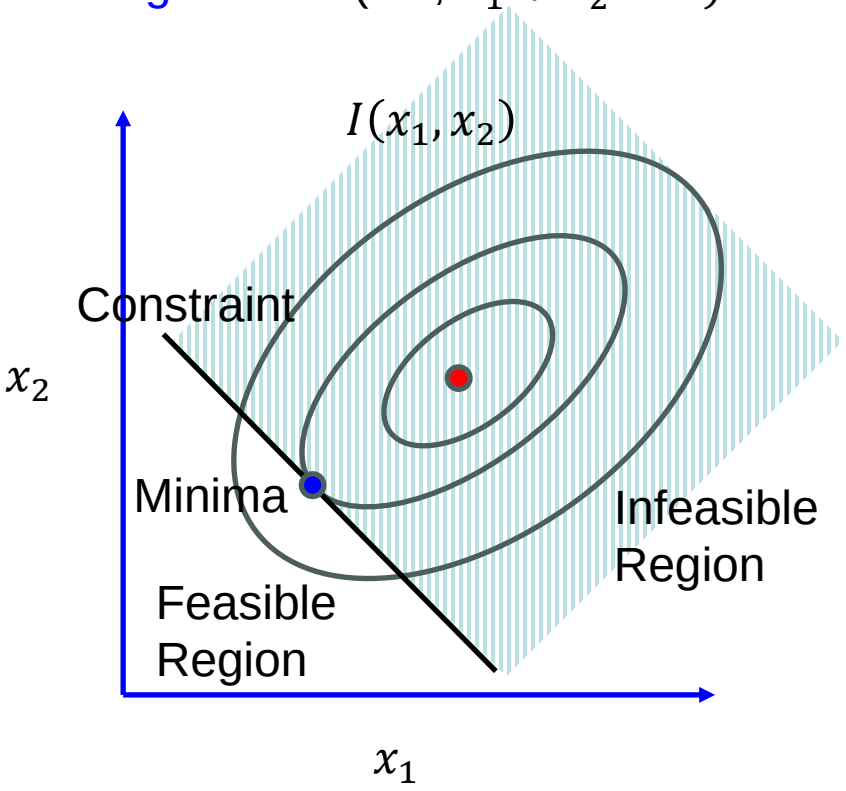
$$\Longrightarrow [\nabla^2 I(x_1,x_2)]_0^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/4 \end{bmatrix}$$

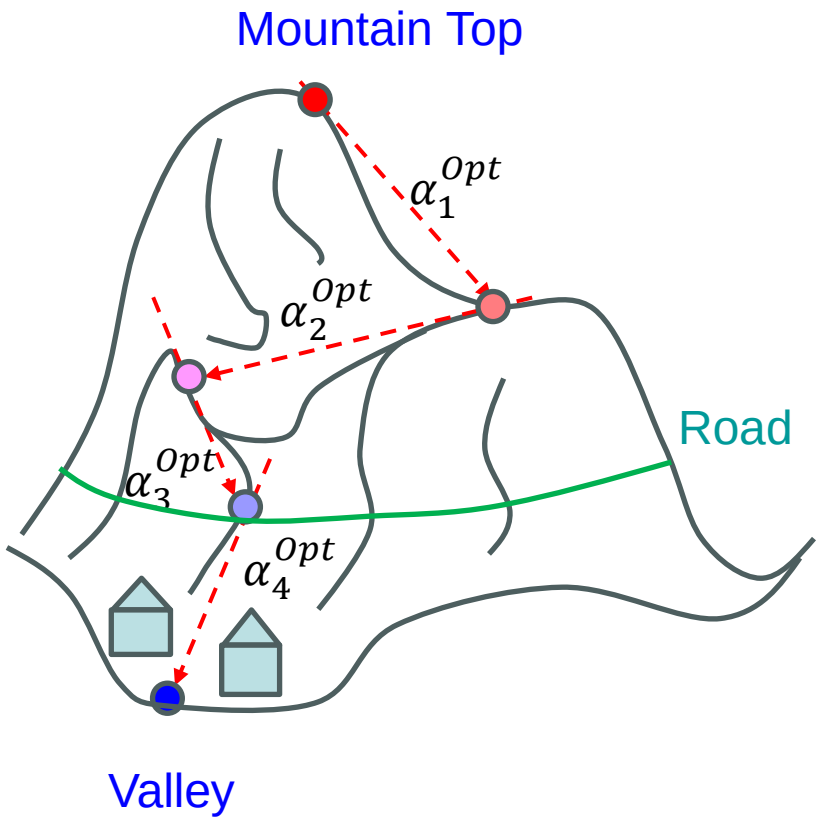
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 - [\nabla^2 I(x_1,x_2)]_0^{-1} \begin{bmatrix} \frac{\partial I(x_1,x_2)}{\partial x_1} \\ \frac{\partial I(x_1,x_2)}{\partial x_2} \end{bmatrix}_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1/2 & 0 \\ 0 & 1/4 \end{bmatrix} \times \begin{bmatrix} 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Second iteration is not needed.

Constrained Optimization Techniques

1) Unlike unconstrained optimization, real-life engineering processes often encounter constraints. For example, we might be interested to reach to road instead of Valley from Mountain Top (i.e., anything below the road is infeasible).
2) In Machine Learning (Regression and Neural Network), often the weights are regularized (i.e., $\theta_1 + \theta_2 < T$). This is included as constraint.





2) Commonly used Methods are a) Variable Elimination, b) Lagrange Multipliers, c) Kuhn Tucker Substitution, and d) Penalty Function.

Variable Elimination

Consider a simple constrained optimization problem:

Objective: $\min I\,(x_1, x_2, \cdots, x_N)$

Subject to:

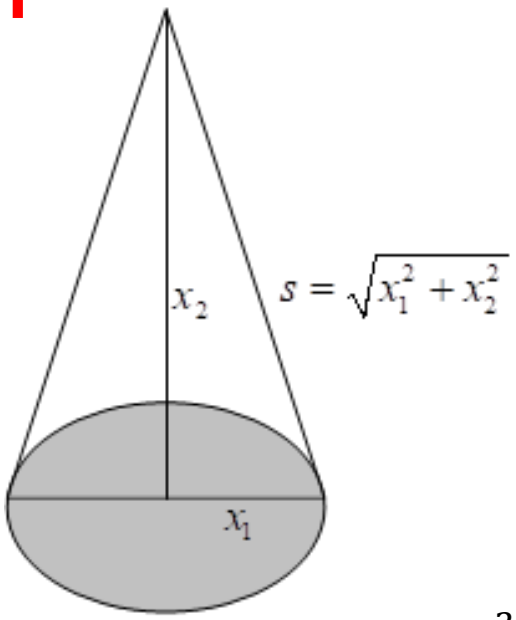
$$h_k(x_1, x_2, \cdots, x_N) = 0; k = 1, 2, \cdots, K; K < N$$

1) This problem comprises of K equality constraints.

2) In the variable elimination technique, (any) K of the N variables can be eliminated using the K constraint equations.

3) Thus, the modified optimization problem will have only (N — K) independent variables. The modified problem is then written as

Objective: $\min I\,(x_1, x_2, \cdots, x_{N-K})$



$$\max \text{ or } \min I\,(x_1, x_2) = \frac{\pi x_1^2 x_2}{3}$$

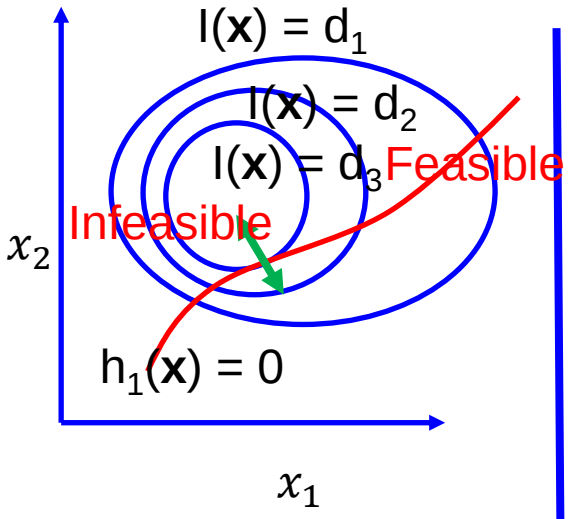
Subjected to $x_1 + x_2 = 10$

From constraint, $x_1 = 10 - x_2$

$$\max \text{ or } \min I\,(x_2) = \frac{\pi(10 - x_2)^2 x_2}{3}$$

4) It must be emphasized that such a substitution is easy only for simple forms of the equality constraints, mostly *linear*.

Lagrange Multipliers



At the intersection of contour and constraint line, gradient is same.

$$\nabla I(\boldsymbol{x}) = v_1 \nabla h_1(\boldsymbol{x})$$

Consider a simple constrained optimization problem:

Objective: $\min I(x_1, x_2, \cdots, x_N)$

Subject to: $h_k(x_1, x_2, \cdots, x_N) = 0; k = 1, 2, \cdots, K; K < N$

1) The constraints can be incorporated directly in the objective function to give a modified objective function, L. The modified problem is then written as

Objective: $\min L(\boldsymbol{x}, \boldsymbol{v}) = I(\boldsymbol{x}) - \sum_{k=1}^{K} v_k h_k(\boldsymbol{x})$

1) L is referred to as the Lagrange function and v_k are Lagrange multipliers. In this equation, the v_k can be either positive or negative.

2) To minimize I with K constraints, one must minimize L without constraints. Thus, for optimality, the necessary analytical conditions can be obtained as

$$\frac{\partial L}{\partial x_i} = 0; i = 1, 2, \cdots, N; \frac{\partial L}{\partial v_k} = h_k(x) = 0; k = 1, 2, \cdots, K$$

Kuhn-Tucker Conditions

Consider a simple constrained optimization problem:

Objective: $\min I(x_1, x_2, \cdots, x_N)$

Real-life problems often involve inequality constraints ($\theta_1 + \theta_2 < T$ in regression regularization)

Subject to: $h_k(x_1, x_2, \cdots, x_N) = 0; k = 1, 2, \cdots, K; K < N$ (Equality)

$g_j(x_1, x_2, \cdots, x_N) \geq 0; j = 1, 2, \cdots, J$ (Inequality)

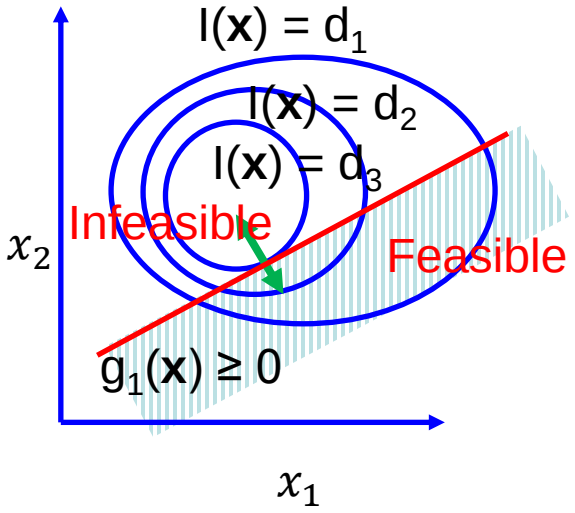
1) The constraints can be incorporated directly in the objective function to give a modified objective function, L. The modified problem is then written as

Objective: $$L(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{u}) = I(\boldsymbol{x}) - \sum_{k=1}^{K} v_k h_k(\boldsymbol{x}) - \sum_{j=1}^{J} u_j g_j(\boldsymbol{x})$$

Necessary Conditions of optimality are given as follows:

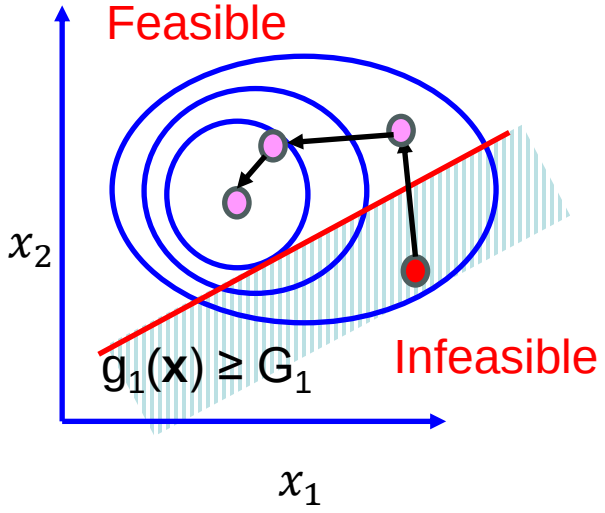
$$\nabla I(\boldsymbol{x}) - \sum_{k=1}^{K} v_k \nabla h_k(\boldsymbol{x}) - \sum_{j=1}^{J} u_j \nabla g_j(\boldsymbol{x}) = 0$$

$$h_k(x_1, x_2, \cdots, x_N) = 0; k = 1, 2, \cdots, K$$



$u_j g_j(\boldsymbol{x}) = 0; j = 1, 2, \cdots, J$ This is the special KKT (Karush-Kuhn-Tucker) condition

Penalty Function



Consider a simple constrained optimization problem:

Objective: $\min I(x_1, x_2, \cdots, x_N)$

Subject to:

$h_k(x_1, x_2, \cdots, x_N) = H_k; k = 1, 2, \cdots, K; K < N$ (Equality)

$g_j(x_1, x_2, \cdots, x_N) \geq G_J; j = 1, 2, \cdots, J$ (Inequality)

Objective: $\min F(\boldsymbol{x}) = I(\boldsymbol{x}) + \sum_{k=1}^{K+J} P_k$ — This term becomes dominating when constraint is violated.

$$P_k = \begin{cases} M & \leftarrow \text{Hard Penalty for Constraint Violation} \\ M \times \left(1 - \frac{h_k}{H_k}\right)^2 \ or \ M \times \left(1 - \frac{g_k}{G_k}\right)^2 & \leftarrow \text{Bracketed Penalty for Constraint Violation} \\ 0 & \leftarrow \text{No Constraint Violation} \end{cases}$$

M is a Very Large Number

1) Unlike, variable elimination, Lagrange multiplier and KKT multiplier which satisfy the constraints a priory, Penalty function may start with constraint violation which is eliminated over multiple optimization iterations.
2) Hard Penalty give strict constraint satisfaction but do not add any intelligence to algorithm. In Bracketed penalty, Penalty is proportional to the extent of constraint violation thus gives the intelligence to algorithm, but it does not lead to strict constraint satisfaction.

Multi-variable Gradient Based Techniques

Example 3: Minimize $I = x_1^2 + 2x_2^2$, with the initial point $\boldsymbol{x}_0 = [x_1, x_2]_0^T = [2, 2]$ using the Gradient Descent with fixed step size (α) of 0.2 with a constraint $x_1 + x_2 = 3$. Carry out two iterations for $-5 < x_1, x_2 < 5$ using variable elimination method.

$$I = x_1^2 + 2x_2^2 = (3 - x_2)^2 + 2x_2^2 \qquad \frac{dI(x_2)}{dx_2} = -2(3 - x_2) + 4x_2$$

Iteration 1:

$$\left[\frac{dI(x_2)}{dx_2}\right]_0 = -2 \times (3 - 2) + 4 \times 2 = 6$$

$$[x_2]_1 = [x_2]_0 - \alpha \left[\frac{dI(x_2)}{dx_2}\right]_0 = [2] - 0.2 \times [6] = 0.8$$

Iteration 2:

$$\left[\frac{dI(x_2)}{dx_2}\right]_1 = -2 \times (3 - 0.8) + 4 \times 0.8 = -1.2$$

$$[x_2]_2 = [x_2]_1 - \alpha \left[\frac{dI(x_2)}{dx_2}\right]_1 = [0.8] - 0.2 \times [-1.2] = 1.04 \quad \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.96 \\ 1.04 \end{bmatrix}$$

The obtained solution is close to analytical optimal solution $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{opt} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

Multi-variable Gradient Based Techniques

Example 4: Minimize $I = x_1^2 + 2x_2^2$, with the initial point $\boldsymbol{x}_0 = [x_1, x_2]_0^T = [2, 2]$ using the Gradient Descent with fixed step size (α) of 0.2 with a constraint $x_1 + x_2 = 3$. Carry out one iterations for $-5 < x_1, x_2 < 5$ using Lagrange Multiplier method.

$$I = x_1^2 + 2x_2^2 \Longrightarrow L(x_1, x_2, v_1) = x_1^2 + 2x_2^2 - v_1(x_1 + x_2 - 3)$$

$$\frac{\partial L(x_1, x_2, v_1)}{\partial x_1} = 2x_1 - v_1 = 0; \frac{\partial L(x_1, x_2, v_1)}{\partial x_2} = 4x_2 - v_1 = 0; \frac{\partial L(x_1, x_2, v_1)}{\partial v_1} = x_1 + x_2 - 3 = 0$$

Analytical Solution

$$2x_1 - v_1 = 0$$
$$4x_2 - v_1 = 0$$
$$x_1 + x_2 - 3 = 0$$
$$\begin{bmatrix} x_1 \\ x_2 \\ v_1 \end{bmatrix}_{opt} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}$$

Iteration 1: Obtain the point $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ in infeasible region using GD.

$$\left[\frac{\partial I(x_1, x_2)}{\partial x_1}\right]_0 = 2x_1 = 2 \times 2 = 4; \left[\frac{\partial I(x_1, x_2)}{\partial x_2}\right]_0 = 4x_2 = 4 \times 2 = 8$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 - \alpha \begin{bmatrix} \frac{\partial I(x_1, x_2)}{\partial x_1} \\ \frac{\partial I(x_1, x_2)}{\partial x_2} \end{bmatrix}_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.2 \times \begin{bmatrix} 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 0.8 \\ 1.6 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 0.4 \end{bmatrix}$$

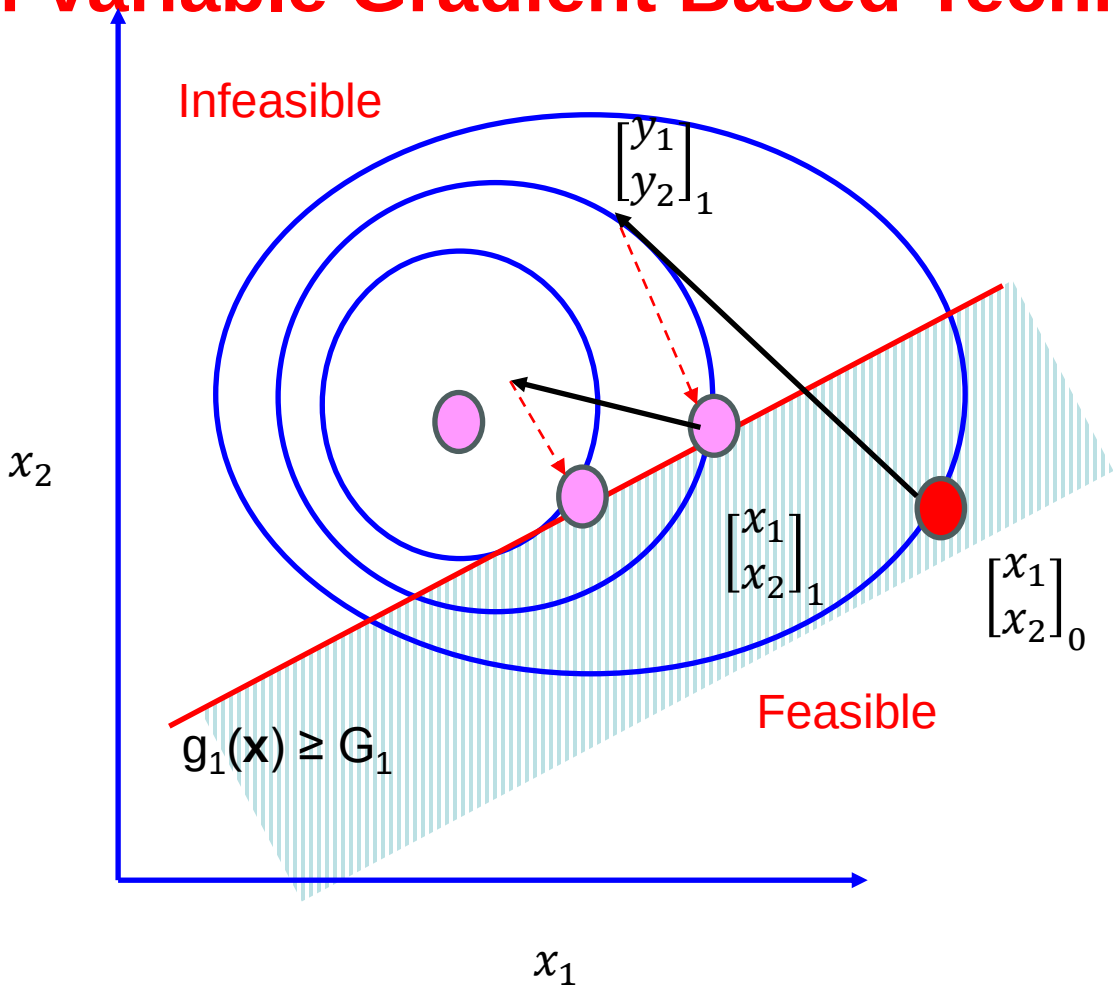
Obtaining the projection of $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ on feasible region (Projected GD)

$$\min F = [(y_1 - x_1)_1^2 + (y_2 - x_2)_1^2] = (1.2 - x_1)_1^2 + (0.4 + 3 - x_1)_1^2$$

$$\frac{dF}{dx_1} = -2(1.2 - x_1) - 2(3.4 - x_1) = 0 \Longrightarrow x_1 = 2.3 \Longrightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 = \begin{bmatrix} 2.3 \\ 0.7 \end{bmatrix}$$

The obtained solution is close to analytical optimal solution

Multi-variable Gradient Based Techniques



1) Projected Gradient Descent follows the same path as Gradient Descent and obtain new point $[y_1, y_2, \cdots, y_N]_1^T$ from $[x_1, x_2, \cdots, x_N]_0^T$.
2) If this point is infeasible, it is projected on a feasible region to obtain $[x_1, x_2, \cdots, x_N]_1^T$.
3) This projection is done by minimizing the Euclidian distance of $[y_1, y_2, \cdots, y_N]_1^T$ from feasible region to obtain the best $[x_1, x_2, \cdots, x_N]_1^T$.

Thank You

Concept of Optimization

Consider minimization problem. For any given function, $f(x)$, of a single variable, x, the optimality criteria can easily be derived using the Taylor series expansion for the change in f around any point, x^* , in the interval, x^{low} to x^{high} , if x changes by a small amount (a perturbation) from x^* to $x^* + \varepsilon$ as follows

$$f\left(x^* + \varepsilon\right) - f\left(x^*\right) = \varepsilon \frac{df}{dx}\Big|_{x=x^*} + \frac{\varepsilon^2}{2!}\frac{d^2 f}{dx^2}\Big|_{x=x^*} + ... + \frac{\varepsilon^n}{n!}\frac{d^n f}{dx^n}\Big|_{x=x^*} + O_{n+1}(\varepsilon)$$

If x^* is the local minimum, then ***all*** other points around x^* will have higher values of f and the LHS of equations will be positive for ***all*** values (positive as well as negative) of ε. One can, therefore, write the above equation as

$$\varepsilon \frac{df}{dx}\Big|_{x=x^*} + \frac{\varepsilon^2}{2!}\frac{d^2 f}{dx^2}\Big|_{x=x^*} + ... + \frac{\varepsilon^n}{n!}\frac{d^n f}{dx^n}\Big|_{x=x^*} + O_{n+1}(\varepsilon) \geq 0$$

Since ε is sufficiently small, the first term dominates over the others, and one has

$$\varepsilon \frac{df}{dx}\Big|_{x=x^*} \geq 0$$

Again, ε can either be positive or negative. The only way the above equation can be valid is when

$$\frac{df}{dx}\Big|_{x=x^*} = 0$$

Why Derivative should be Zero for Optimization?

If $\left.\frac{df}{dx}\right|_{x=x^*} = 0$ in $\varepsilon \left.\frac{df}{dx}\right|_{x=x^*} + \frac{\varepsilon^2}{2!}\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} + ... + \frac{\varepsilon^n}{n!}\left.\frac{d^n f}{dx^n}\right|_{x=x^*} + O_{n+1}(\varepsilon) \geq 0$

The second term will then dominate over the remaining terms. Further, the term, ε^2, can have only positive values. Thus, the inequality in above equation will hold true only if

For minimization $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} \geq 0$

Similarly, the condition for maximization can also be derived as:

$\left.\frac{df}{dx}\right|_{x=x^*} = 0$ and $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} \leq 0$

Since equality sign can be present in double derivative of minimization and maximization, the more refined optimality conditions are

Minimization: $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} > 0$ Maximization: $\left.\frac{d^2 f}{dx^2}\right|_{x=x^*} < 0$

Further, the second or higher derivative is zero then one has to continue till we get a non-zero derivative ⇒

$$f(x^* + \varepsilon) - f(x^*) = \frac{\varepsilon^k}{k!}\left.\frac{d^k f}{dx^k}\right|_{x=x^*} + O_{k+1}(\varepsilon)$$

From this equation, one can derive that if k is odd then obtained solution is saddle point. Else, it is positive or negative based on sign of nth order derivative

Condition of Optimality for Multi-variable Function

Consider a Taylor series expansion of the two-variable function, $f(x_1, x_2)$, at any point, x_1, x_2, with respect to a reference point, x^*_1, x^*_2:

$$f(x_1^* + \varepsilon_1, x_2^* + \varepsilon_2) = f(x_1^*, x_2^*) + \left[\frac{\partial f}{\partial x_1}\right]_{x_1^*, x_2^*} \varepsilon_1 + \left[\frac{\partial f}{\partial x_2}\right]_{x_1^*, x_2^*} \varepsilon_2 +$$

$$+ \frac{1}{2}\left[\varepsilon_1 \left[\frac{\partial^2 f}{\partial x_1^2}\right]_{x_1^*, x_2^*} \varepsilon_1 + 2\varepsilon_1 \left[\frac{\partial^2 f}{\partial x_1 \partial x_2}\right]_{x_1^*, x_2^*} \varepsilon_2 + \varepsilon_2 \left[\frac{\partial^2 f}{\partial x_2^2}\right]_{x_1^*, x_2^*} \varepsilon_2\right] + O_3(\varepsilon_1, \varepsilon_2)$$

In general: $$f(\boldsymbol{x}^* + \boldsymbol{\varepsilon}) = f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^T \boldsymbol{\varepsilon} + \frac{1}{2}\boldsymbol{\varepsilon}^T \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{\varepsilon} + O_3(\boldsymbol{\varepsilon})$$

$\nabla f(\boldsymbol{x}^*) \rightarrow$ N-component column vector of N first derivatives of f at x*

$\nabla^2 f(\boldsymbol{x}^*) \rightarrow$ N×N symmetric matrix of second derivatives of f at x*. It is also called as Hessian Matrix $H_f(x)$ of f(x).

If $\boldsymbol{x}^*$ is the local minimum, then any point in its vicinity will have a higher value of the function, *f* (as for the one-variable case) i.e. $[f(\boldsymbol{x}^*+\boldsymbol{\varepsilon}) - f(\boldsymbol{x}^*)] \geq 0$

$$\left[\frac{\partial f}{\partial x_1}\right]_{x_1^*, x_2^*} = 0; \left[\frac{\partial f}{\partial x_2}\right]_{x_1^*, x_2^*} = 0; \left[\left(\frac{\partial^2 f}{\partial x_1^2}\right)_{x_1^*, x_2^*} + 2\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)_{x_1^*, x_2^*} + \left(\frac{\partial^2 f}{\partial x_2^2}\right)_{x_1^*, x_2^*}\right] \geq 0$$

For Minimization ⇒ $$\nabla f(\boldsymbol{x}^*) = 0, \nabla^2 f(\boldsymbol{x}^*) \geq 0$$

Condition of Optimality for Multi-variable Function

$$\left[\frac{\partial f}{\partial x_1}\right]_{x_1^*, x_2^*} = 0; \left[\frac{\partial f}{\partial x_2}\right]_{x_1^*, x_2^*} = 0; \left[\left(\frac{\partial^2 f}{\partial x_1^2}\right)_{x_1^*, x_2^*} + 2\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)_{x_1^*, x_2^*} + \left(\frac{\partial^2 f}{\partial x_2^2}\right)_{x_1^*, x_2^*}\right] \geq 0$$

For Minimization $\Rightarrow$ $\nabla f\left(\boldsymbol{x}^*\right) = 0, \nabla^2 f\left(\boldsymbol{x}^*\right) \geq 0$

Above equation represent the fact that all the N eigenvalues, $\boldsymbol{\Lambda} \equiv [\lambda_1, \lambda_2, \ldots, \lambda_N]$, of the Hessian matrix $\mathbf{H}_f(x)$ are greater than or equal to zero.

If all eigen values are > 0, ≥ 0, < 0, ≤ 0, and some are greater and remaining are smaller than zero then the $\mathbf{H}_f(x)$ is positive definite, positive semidefinite, negative definite, negative semidefinite and indefinite, respectively

Condition of optimality

$$\nabla f\left(\boldsymbol{x}^*\right) = 0$$

If $\mathbf{H}_f(x)$ is positive definite $\Rightarrow$ Optimum is minimum

If $\mathbf{H}_f(x)$ is negative definite $\Rightarrow$ Optimum is maximum

If $\mathbf{H}_f(x)$ is indefinite $\Rightarrow$ Optimum is a saddle point

Multi-variable Gradient Based Techniques

Example: Minimize f(x) = $(3 - x_1^2)^2 + (2 - x_2^2)^2$, with the initial point $\boldsymbol{x}^{(0)} = \{x_1, x_2\}^{(0)} = \{1, 1\}$ using the gradient based techniques. Carry out two iterations.

Gradient Descent (with optimum step size): Iteration 1

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(0)} = \{1,1\}$

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \alpha \nabla f\left(\boldsymbol{x}^{(0)}\right)$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right) \Rightarrow \frac{\partial f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1} = -4 \times 1 \times \left(3 - 1^2\right) = -8$$

$$x_1^{(1)} = x_1^{(0)} - \alpha\left(\frac{\partial f}{\partial x_1}\right)^{(0)} = 1 - \alpha(-8) = 1 + 8\alpha$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right) \Rightarrow \frac{\partial f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_2} = -4 \times 1 \times \left(2 - 1^2\right) = -4$$

$$x_2^{(1)} = x_2^{(0)} - \alpha\left(\frac{\partial f}{\partial x_2}\right)^{(0)} = 1 - \alpha(-4) = 1 + 4\alpha$$

$$f(\boldsymbol{x}) = \left(3 - \left(1 + 8\alpha\right)^2\right)^2 + \left(2 - \left(1 + 4\alpha\right)^2\right)^2 = \left(2 - 16\alpha - 64\alpha^2\right)^2 + \left(1 - 8\alpha - 16\alpha^2\right)^2$$

$$\frac{df(\boldsymbol{x})}{d\alpha} = 2\left(2 - 16\alpha - 64\alpha^2\right)\left(-16 - 128\alpha\right) + 2\left(1 - 8\alpha - 16\alpha^2\right)\left(-8 - 32\alpha\right) = 0$$

Using trial and error procedure, the value of α is obtained as 0.0932

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \alpha \nabla f\left(\boldsymbol{x}^{(0)}\right) \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.0932 \begin{bmatrix} -8 \\ -4 \end{bmatrix} = \begin{bmatrix} 1.7456 \\ 1.3728 \end{bmatrix}$$

$$f\left(\boldsymbol{x}^{(1)}\right) = \left(3 - 1.7456^2\right)^2 + \left(2 - 1.3728^2\right)^2 = 0.0155$$

Multi-variable Gradient Based Techniques

Gradient Descent (with optimum step size): Iteration 2

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(1)} = \{1.7456, 1.3728\}$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right); \frac{\partial f(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right)$$

$$\Rightarrow \frac{\partial f\left(\boldsymbol{x}^{(1)}\right)}{\partial x_1} = -4 \times 1.7456 \times \left(3 - 1.7456^2\right) = 0.329$$

$$\Rightarrow \frac{\partial f\left(\boldsymbol{x}^{(1)}\right)}{\partial x_2} = -4 \times 1.3728 \times \left(2 - 1.3728^2\right) = -0.6337$$

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} - \alpha \nabla f\left(\boldsymbol{x}^{(1)}\right)$$

$$x_1^{(2)} = x_1^{(1)} - \alpha\left(\frac{\partial f}{\partial x_1}\right)^{(1)}$$

$$= 1.7456 - \alpha(0.329) = 1.7456 - 0.329\alpha$$

$$x_2^{(2)} = x_2^{(1)} - \alpha\left(\frac{\partial f}{\partial x_2}\right)^{(1)}$$

$$= 1.3728 - \alpha(-0.6337) = 1.3728 + 0.6337\alpha$$

Putting in the function:

$$f(\boldsymbol{x}) = \left(3 - \left(1.7456 - 0.329\alpha\right)^2\right)^2 + \left(2 - \left(1.3728 + 0.6337\alpha\right)^2\right)^2$$

$$= \left(-0.0471 + 1.1486\alpha - 0.1082\alpha^2\right)^2 + \left(0.1154 - 1.74\alpha - 0.4015\alpha^2\right)^2$$

$$\frac{df(\boldsymbol{x})}{d\alpha} = \begin{bmatrix} 2\left(-0.0471 + 1.1486\alpha - 0.1082\alpha^2\right)\left(1.1486 - 0.2164\alpha\right) \\ +2\left(0.1154 - 1.74\alpha - 0.4015\alpha^2\right)\left(-1.74 - 0.803\alpha\right) \end{bmatrix} = 0$$

Using trial and error procedure, the value of α is obtained as 0.059

Multi-variable Gradient Based Techniques

Gradient Descent (with optimum step size): Iteration 2

$$f(\boldsymbol{x}) = \left(3-\left(1.7456-0.329\alpha\right)^2\right)^2 + \left(2-\left(1.3728+0.6337\alpha\right)^2\right)^2$$

$$= \left(-0.0471+1.1486\alpha-0.1082\alpha^2\right)^2 + \left(0.1154-1.74\alpha-0.4015\alpha^2\right)^2$$

$$\frac{df(\boldsymbol{x})}{d\alpha} = \begin{bmatrix} 2\left(-0.0471+1.1486\alpha-0.1082\alpha^2\right)\left(1.1486-0.2164\alpha\right) \\ +2\left(0.1154-1.74\alpha-0.4015\alpha^2\right)\left(-1.74-0.803\alpha\right) \end{bmatrix} = 0$$

Using trial and error procedure, the value of α is obtained as 0.059

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} - \alpha\nabla f\left(\boldsymbol{x}^{(1)}\right)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(2)} = \begin{bmatrix} 1.7456 \\ 1.3728 \end{bmatrix} - 0.059\begin{bmatrix} 0.0329 \\ -0.6337 \end{bmatrix} = \begin{bmatrix} 1.7436 \\ 1.4101 \end{bmatrix}$$

$$f\left(\boldsymbol{x}^{(2)}\right) = \left(3-1.7436^2\right)^2 + \left(2-1.4101^2\right)^2 = 1.746\times10^{-3}$$

The obtained optimum is (x_1 =1.7436, x_2 = 1.4101)

Take Home Assignment: Perform two iterations with fixed step size of 0.05

Multi-variable Gradient Based Techniques

Example: Minimize $f(x) = (3 - x_1^2)^2 + (2 - x_2^2)^2$, with the initial point $\boldsymbol{x}^{(0)} = \{x_1, x_2\}^{(0)} = \{1, 1\}$ using the gradient based techniques. Carry out two iterations.

Newton's Method: Iteration 1

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(0)} = \{1,1\}$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right)$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right)$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} = 12(x_1^2 - 1)$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} = 4(3x_2^2 - 2)$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} = 0$$

$$\frac{\partial f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1} = -4 \times 1 \times \left(3 - 1^2\right) = -8; \frac{\partial f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_2} = -4 \times 1 \times \left(2 - 1^2\right) = -4$$

$$\frac{\partial^2 f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1^2} = 12 \times (1 - 1) = 0; \frac{\partial^2 f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_2^2} = 4 \times (3 - 2) = 4; \frac{\partial^2 f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1 \partial x_2} = 0$$

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \left[H_f\left(\boldsymbol{x}^{(0)}\right)\right]^{-1} \nabla f\left(\boldsymbol{x}^{(0)}\right) \Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix}^{-1} \begin{bmatrix} -8 \\ -4 \end{bmatrix} = \text{undetermined}$$

$$\left|H_f\left(\boldsymbol{x}^{(0)}\right)\right| = \begin{vmatrix} 0 & 0 \\ 0 & 4 \end{vmatrix} = 0 \times 4 - 0 \times 0 = 0 \Rightarrow \left[H_f\left(\boldsymbol{x}^{(0)}\right)\right]^{-1} = \text{undetermined}$$

Newton's method cannot be applied for this problem for the given initial point.

Multi-variable Gradient Based Techniques

For illustration, different initial point is used

Newton's Method: Iteration 1

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(0)} = \{1.5, 1.25\}$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right)$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right)$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1^2} = 12(x_1^2 - 1)$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_2^2} = 4(3x_2^2 - 2)$$

$$\frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_2} = 0$$

$$\frac{\partial f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1} = -4 \times 1.5 \times \left(3 - 1.5^2\right) = -4.5; \quad \frac{\partial f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_2} = -4 \times 1.25 \times \left(2 - 1.25^2\right) = -2.1875$$

$$\frac{\partial^2 f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1^2} = 12 \times (1.5^2 - 1) = 15; \quad \frac{\partial^2 f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_2^2} = 4 \times (3 \times 1.25^2 - 2) = 10.75; \quad \frac{\partial^2 f\left(\boldsymbol{x}^{(0)}\right)}{\partial x_1 \partial x_2} = 0$$

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \left[H_f\left(\boldsymbol{x}^{(0)}\right)\right]^{-1} \nabla f\left(\boldsymbol{x}^{(0)}\right)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{(1)} = \begin{bmatrix} 1.5 \\ 1.25 \end{bmatrix} - \begin{bmatrix} 15 & 0 \\ 0 & 10.75 \end{bmatrix}^{-1} \begin{bmatrix} -4.5 \\ -2.1875 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.25 \end{bmatrix} - \begin{bmatrix} 0.0667 & 0 \\ 0 & 0.093 \end{bmatrix} \begin{bmatrix} -4.5 \\ -2.1875 \end{bmatrix} = \begin{bmatrix} 1.80015 \\ 1.4534 \end{bmatrix}$$

$$f\left(\boldsymbol{x}^{(1)}\right) = \left[3 - 1.80015^2\right]^2 + \left[2 - 1.4534^2\right]^2 = 0.07049$$

Optimum solution = {1.8, 1.4534}

Constrained Optimization Techniques

Example: Minimize f(x) = $(3 - x_1^2)^2 + (2 - x_2^2)^2$, with the initial point $\boldsymbol{x}^{(0)} = \{x_1, x_2\}^{(0)} = \{1, 1\}$ and constraint $x_1 + x_2 = 2$

Variable Elimination Method

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(0)} = \{1,1\}$

$$x_1 + x_2 = 2$$

Replace x_2 by $(2\text{-}x_1)$

$$f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left[2 - \left(2 - x_1\right)^2\right]^2 = \left(3 - x_1^2\right)^2 + \left(-2 + 4x_1 - x_1^2\right)^2$$

$$f\left(\boldsymbol{x}^{(0)}\right) = \left(3 - 1^2\right)^2 + (1)^2 = 5$$

$$\frac{df}{dx_1} = -4x_1\left(3 - x_1^2\right) + 2\left(-2 + 4x_1 - x_1^2\right)\left(4 - 2x_1\right)$$

$$\frac{d^2 f}{dx_1^2} = -12(1 + x_1^2) + 2\left(4 - 2x_1\right)\left(4 - 2x_1\right) - 4\left(-2 + 4x_1 - x_1^2\right)$$

$$\frac{df\left(x_1^{(0)}\right)}{dx_1} = -4 \times 1 \times \left(3 - 1^2\right) + 2\left(-2 + 4 \times 1 - 1^2\right)\left(4 - 2 \times 1\right) = -4$$

$$\frac{d^2 f\left(x_1^{(0)}\right)}{dx_1^2} = -12 \times 0 + 2\left(4 - 2 \times 1\right)\left(4 - 2 \times 1\right) - 4\left(-2 + 4 \times 1 - 1^2\right) = 4$$

Constrained Optimization Techniques

Variable Elimination Method

$$x_1^{(1)} = x_1^{(0)} - \frac{f'\left(x_1^{(0)}\right)}{f''\left(x_1^{(0)}\right)} = 1 - \frac{(-4)}{4} = 2.0000$$

$$x_2^{(1)} = 2 - x_1^{(1)} = 2 - 2.0000 = 0.0000$$

$$f\left(\mathbf{x}^{(1)}\right) = \left(3 - 2.0000^2\right)^2 + \left(2 - 0.0000^2\right)^2 = 5.0000$$

Optimum = ($x_1 = 2$, $x_2 = 0$)

Constrained Optimization Techniques

Example: Minimize f(x) = $(3 - x_1^2)^2 + (2 - x_2^2)^2$, with the initial point $\boldsymbol{x}^{(0)} = \{x_1, x_2\}^{(0)} = \{1, 1\}$ and constraint $x_1 + x_2 = 2$

Lagrange Multiplier Method

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(0)} = \{1,1\}$

$x_1 + x_2 = 2$

$$L(\boldsymbol{x}) = f(\boldsymbol{x}) - \sum_{k=1}^{K} v_k h_k = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2 - v_1\left(x_1 + x_2 - 2\right)$$

$$\frac{\partial L(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right) - v_1 = 0$$

$$\frac{\partial L(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right) - v_1 = 0$$

$$h_1 = \left(x_1 + x_2 - 2\right) = 0$$

Constrained Optimization Techniques

Lagrange Multiplier Method

$$\frac{\partial L(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right) - v_1 = 0 \Rightarrow x_1\left(3 - x_1^2\right) = -\frac{v_1}{4}$$

$$\frac{\partial L(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right) - v_1 = 0 \Rightarrow x_2\left(2 - x_2^2\right) = -\frac{v_1}{4}$$

$$\left(x_1 + x_2 - 2\right) = 0$$

$$x_1\left(3 - x_1^2\right) = x_2\left(2 - x_2^2\right)$$

$$x_1\left(3 - x_1^2\right) - \left(2 - x_1\right)\left(2 - \left(2 - x_1\right)^2\right) = 0$$

Using a trial-and-error procedure, we get $x_1 = 1.59$, $x_2 = 2 - 1.59 = 0.41$

$$f\left(\boldsymbol{x}^{(1)}\right) = \left(3 - 1.59^2\right)^2 + \left(2 - 0.41^2\right)^2 = 3.5785$$

Function value is slightly better than that obtained using Newton's method with variable elimination

Constrained Optimization Techniques

Example: Minimize f(x) = $(3 - x_1^2)^2 + (2 - x_2^2)^2$, with the initial point $\boldsymbol{x}^{(0)} = \{x_1, x_2\}^{(0)} = \{1, 1\}$ and constraints $x_1 + x_2 = 2$, $x_1 - 1 \geq 0$

Kuhn Tucker Multiplier Method

Given $f(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2$ at $\boldsymbol{x}^{(0)} = \{1,1\}$

$$h_1(\boldsymbol{x}) = x_1 + x_2 - 2 = 0$$
$$g_1(\boldsymbol{x}) = x_1 - 1 \geq 0$$

$$L(\boldsymbol{x}) = f(\boldsymbol{x}) - v_1 h_1(\boldsymbol{x}) - u_1 g_1(\boldsymbol{x}) = \left(3 - x_1^2\right)^2 + \left(2 - x_2^2\right)^2 - v_1(x_1 + x_2 - 2) - u_1(x_1 - 1)$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_1} = -4x_1\left(3 - x_1^2\right); \frac{\partial f(\boldsymbol{x})}{\partial x_2} = -4x_2\left(2 - x_2^2\right); \frac{\partial h(\boldsymbol{x})}{\partial x_1} = 1; \frac{\partial h(\boldsymbol{x})}{\partial x_2} = 1; \frac{\partial g(\boldsymbol{x})}{\partial x_1} = 1; \frac{\partial g(\boldsymbol{x})}{\partial x_2} = 0$$

$$-4x_1\left(3 - x_1^2\right) - v_1 - u_1 = 0 \quad (1)$$

$$-4x_2\left(2 - x_2^2\right) - v_1 = 0 \quad (2)$$

$$h_1(\boldsymbol{x}) = (x_1 + x_2 - 2) = 0 \quad (3)$$

$$u_1 g_1(\boldsymbol{x}) = u_1(x_1 - 1) = 0 \quad (4)$$

From (1) and (2)

$$u_1 = 4x_2\left(2 - x_2^2\right) - 4x_1\left(3 - x_1^2\right)$$

$$x_2 = 2 - x_1$$ From (3)

$$u_1 g_1(\boldsymbol{x}) = \left[4x_2\left(2 - (2 - x_1)^2\right) - 4x_1\left(3 - x_1^2\right)\right](x_1 - 1) = 0$$

Constrained Optimization Techniques

Kuhn Tucker Multiplier Method

$$-4x_1\left(3-x_1^2\right)-v_1-u_1=0 \quad (1)$$

$$-4x_2\left(2-x_2^2\right)-v_1=0 \quad (2)$$

$$h_1(\boldsymbol{x})=\left(x_1+x_2-2\right)=0 \quad (3)$$

$$u_1g_1(\boldsymbol{x})=u_1\left(x_1-1\right)=0 \quad (4)$$

From (1) and (2)

$$u_1=4x_2\left(2-x_2^2\right)-4x_1\left(3-x_1^2\right)$$

$$x_2=2-x_1$$

From (3)

$$u_1g_1(\boldsymbol{x})=\left[4x_2\left(2-\left(2-x_1\right)^2\right)-4x_1\left(3-x_1^2\right)\right]\left(x_1-1\right)=0$$

If $g_1 = 0$ and $u_1 \neq 0$ $\Rightarrow x_1 = 1, u_1 = -4$ $\Rightarrow x_2 = 1, f = 5$

If $u_1 = 0$, g_1 may or may not be zero, following equations need to be solved to give

$$u_1=\left[4(2-x_1)\left(2-\left(2-x_1\right)^2\right)-4x_1\left(3-x_1^2\right)\right]=0$$

$\Rightarrow x_1 = 1.59, x_2 = 0.41, u_1 = 0$ $\Rightarrow f = 3.5785$

$h_1(\mathbf{x}) = x_1 + x_2 - 2 = -0.05 \approx 0$ (discrepancy arise due to round off error)

$g_1(\mathbf{x}) = x_1 - 1 = 1.59 - 1 = 0.59 \geq 0$