Lecture 11: *Mathematical Programming*

Scribe: *Janhavi Vikrama, Rakshanda Agarwal, Siddharth Agrawal, Varun Prakash*

Contents

1 Optimization

1.1 What is Optimization?

Optimization Problem is maximizing or minimizing some function $f(x)$ relative to some set of constraints, where the constraints represent a range of choices and boundations available in certain situations in real life. The function allows comparison of the different choices for determining which might be best.

1.2 Mathematical Programming

Before that we need to know about **finite-definite optimization**. It is the case where a choice corresponds to selecting the values of a finite number of real variables, called decision variables. For purposes of general discussion, such as now, the decision variables may be denoted by $x_1, ..., x_n$ and each allowable choice therefore identified with a point $x = (x_1, ..., x_n) \in R_n$.

Mathematical Programming is a synonym for finite-dimensional optimization. Its usage predates "Computer Programming," which actually arose from attempts at solving optimization problems on early computers.

1.3 Optimizing multi-variable functions

Let us consider a function of n-variables,

$$f(x) : R^n \rightarrow R \tag{1}$$

Case1: For n=1,$f(x)$: $R \rightarrow$ R be a a continuous function and for f′(x)=0 it has a maxima($f''(x) \leq 0$) and minima ($f''(x) \geq 0$) .

Case2: For n>1, assume that $f(x)$: $R \rightarrow$ R^n is C^2 i.e. it has a continuous second order partial derivative at a point $x*$ given by

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

$$\nabla^2 f(x*) > 0 \text{(positive semi-definite)} \tag{3}$$

Here,

$$\nabla f(x*) = [\frac{\partial f}{\partial x_1} \frac{\partial f}{\partial x_2} ..... \frac{\partial f}{\partial x_n}]^T \tag{4}$$

$$\nabla^2 f(x*) = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{5}$$

Suppose $C \in R$, b is a vector, A is symmetric positive definite.
For a function

$$f(x) = c + b^T x + 0.5 x^T A x \tag{6}$$

We get

$$\nabla f(x) = b + Ax \tag{7}$$

Say, $X = (v_1, v_2)^T$

$$f(x) = c + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{8}$$

$$f(x) = c + b_1x_1 + b_2x_2 + \frac{1}{2} \begin{bmatrix} x_1a_{11} + x_2a_{21} \\ x_1a_{12} + x_2a_{22} \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{9}$$

$$f(x) = c + b_1x_1 + b_2x_2 + \frac{1}{2}(x_1^2a_{11} + x_1x_2a_{21} + x_1x_2a_{12} + x_2^2a_{22}) \tag{10}$$

As symmetric $a_{12} = a_{21}$

$$f(x) = c + b_1x_1 + b_2x_2 + x_1x_2a_{21} + \frac{1}{2}(x_1^2a_{11} + x_2^2a_{22}) \tag{11}$$

From this we get,

$$\nabla f(x) = \begin{bmatrix} b_1 + x_2a_12 + x_1a_11 \\ b_2 + x_1a_12 + x_2a_22 \end{bmatrix} = b + Ax \tag{12}$$

Computing the Hessian:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1^2} & \frac{\partial f}{\partial x_1 \partial x_2} \\ \frac{\partial f}{\partial x_2 \partial x_1} & \frac{\partial f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = A \tag{13}$$

1.4 Gradient Descent

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient.

For a function f(x):$R^n \rightarrow$ R , the gradient descent equations are given by

$$x_{t+1} = x_t + alpha * f'(x) \text{ (for n=1)} \tag{14}$$

$$x_{t+1} = x_t + \nabla f(x) \text{ (for n >1)} \tag{15}$$

Here alpha is learning rate which is the size of these steps. With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point since the slope of the hill is constantly changing. With a very low learning rate, we can confidently move in the direction of the negative gradient since we are recalculating it so frequently. A low learning rate is more precise, but calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom. An example of gradient descent can be seen in Figure 1.

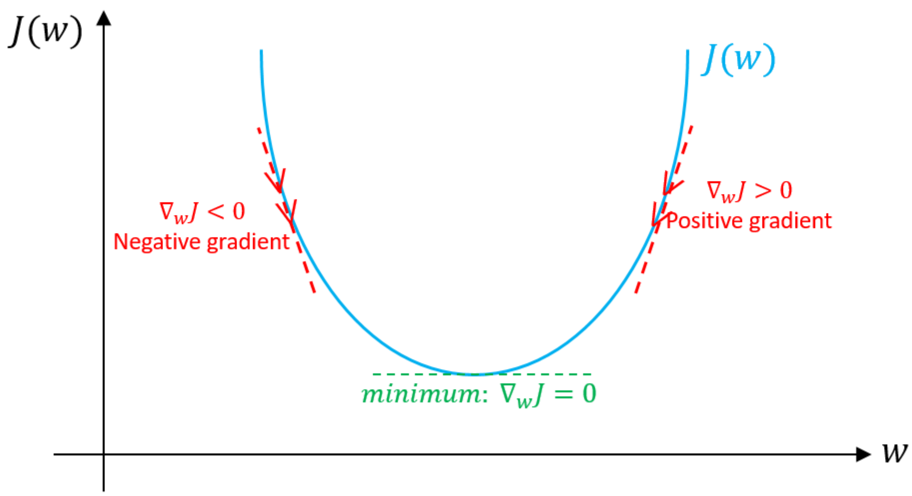

Figure 1: Gradient of Function $J(w)$ w.r.t. w.

2 Constrained Optimization

2.1 Introduction

Constrained optimization is the process of optimizing an objective function with respect to some variables in the presence of constraints on those variables. The objective function is either a cost function or energy function, which is to be minimized, or a reward function or utility function, which is to be maximized.

Constraints can be either hard constraints, which set conditions for the variables that are required to be satisfied, or soft constraints, which have some variable values that are penalized in the objective function if, and based on the extent that, the conditions on the variables are not satisfied.

2.2 General Form

A general constrained minimization problem may be written as follows:

$$\text{Minimize } f(x) \tag{16}$$

Subject To,

$$g(x) = c_i \quad for\, i = 1, ..., n \;\text{ Equality Contraint} \tag{17}$$

$$h(x) \geq d_j \;\; for\, j = 1, ..., m \;\; \text{Inequality Contraint} \tag{18}$$

Where equations (17) and (18) are constraints that are required to be satisfied (hard constraints), and equation (16) is the objective function that needs to be optimized subject to the constraints.

2.3 Example

Let us consider a simple example of Constrained Optimization Problem were our objective is to minimize the given function.

$$\text{Minimize} \sum_{t=0}^{2} u(t)^2 \tag{19}$$

Subject To,

$$x(t+1) = 2x(t) + u(t) \tag{20}$$

$$x(0) = 0 \tag{21}$$

$$x(3) = 10 \tag{22}$$

$$u(t) \geq 0 \tag{23}$$

Since, t only ranges from $0, ., .2$ let us generalize equation (20) for all values of t. Now our problem becomes,

$$\text{Minimize} \sum_{t=0}^{2} u(t)^2 \tag{24}$$

Subject To,

$$x(1) = 2x(0) + u(0) \tag{25}$$

$$x(2) = 2x(1) + u(1) \tag{26}$$

$$x(3) = 2x(2) + u(2) \tag{27}$$

$$x(0) = 0 \tag{28}$$

$$x(3) = 10 \tag{29}$$

$$u(t) \geq 0 \tag{30}$$

Lets simplify the equations by substituting the values of equation (25) and (28) in (26)

$$x(2) = 4x(0) + 2u(0) + u(1) = 2u(0) + u(1) \tag{31}$$

Similarly, substituting equation (29) and (31) in (27)

$$10 = 8x(0) + 4u(0) + 2u(1) + u(2) = 4u(0) + 2u(1) + u(2) \tag{32}$$

Our final simplified constrained optimization is :

$$\text{Minimize } \sum_{t=0}^{2} u(t)^2 \tag{33}$$

Subject To,

$$x(1) = u(0) \tag{34}$$

$$x(2) = 2u(0) + u(1) \tag{35}$$

$$4u(0) + 2u(1) + u(2) = 10 \tag{36}$$

$$u(t) \geq 0 \tag{37}$$

Finally, solving for these equations will provide us with the answer. This was an example of the constrained optimization problem.

3 Constrained Optimization: Linear Programming

3.1 Introduction

We have seen how to formulate the Constrained Optimization Problem. Formulating the problem is only half the battle. Once we have formulated the problem, we still need to know how to solve it. There are various ways to solve the Constrained Optimization Problems. One of the most popular method is Linear Programming. **Linear Programming** assumes that the given problem is a system of linear equations in n-dimensions, where n is the number of variables in the given formulation of the problem.

This quickly tells us that anything in 2 variables or 2 dimensions can be solved graphically. Let us go through an example of Linear Programming formulation in 2 variables to see how to go ahead with it.

3.2 Example

Consider the following formulation of the problem:

$$\text{Maximize } z = x_1 + 2x_2 \tag{38}$$

Subject to,

$$x_1 \leq 3 \tag{39}$$

$$x_1 + x_2 \leq 5 \tag{40}$$

$$x_1, x_2 \geq 0 \tag{41}$$

In the given example, we see that the maximum number of variables is 2. We can use graphs to solve this equation. First, we convert all the inequalities to equality. This would give us a line in 2 dimension. The lines can be seen as portioning the solution space, wherein one part satisfies the constraint and the other does not.

For example, when the inequality in equation (39) is converted to an equality, we get

$$x_1 = 3 \tag{42}$$

This is a vertical line passing through x_1=3. The left side of the line satisfies the constraint and the right side doesn't (Figure 2).

Similarly, we convert the rest of the inequalities and graph them. We get a graph like in Figure 3.

Now, once we have this, we find the intersection of all the feasible region. In this case, it bounded by (0,0) , (0,5), (3,2) ,(3,0). It has been represented by a black outline in Figure 3.

Once we have the vertices of the feasible region, we substitute them in the optimization function, equation (38) and find the value which maximizes the optimization function. In our case, value of z for each of the possible values of $(x1, x2)$ is as follows: $z = 0,\ z = 10,\ z = 7,\ z = 3$.

Clearly, z=10 is the maximum value, when x_1 is 0 and x_2 is 5. Therefore, the solution is $x_1 = 0,\ x_2 = 5$.

3.3 'n' greater than 2 variables

It becomes difficult to use the graphical method as the number of variables increase. Simply using the overlapping feasible region gets more complex with more variables. In such cases, Simplex method is used to solve such constraints. It is an iterative method which optimizes one variable at a time. Further details on simplex method can be found here: `https://personal.utdallas.edu/~scniu/OPRE-6201/documents/LP06-Simplex-Tableau.pdf`

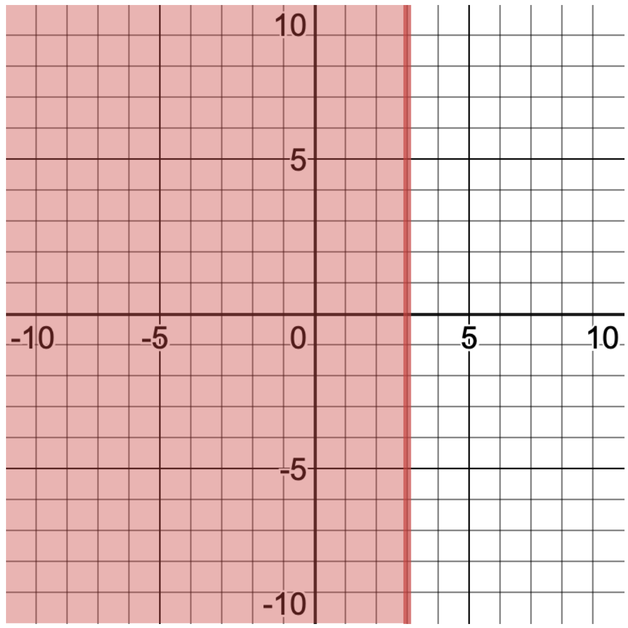

Figure 2: Plot of x_1 =3 with shaded feasible region.

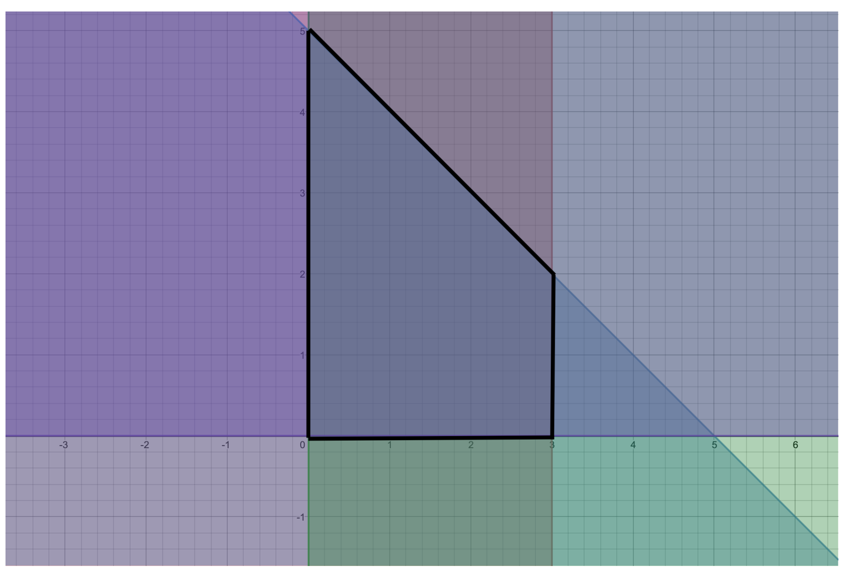

Figure 3: Plot of all the inequalities with shaded feasible region.

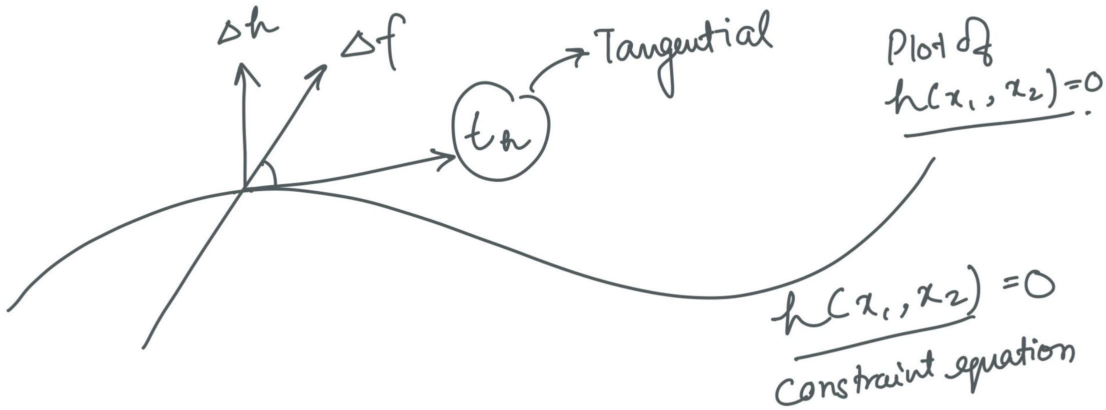

Figure 4: Plot of h with ∇f and ∇h.

4 General Constrained Optimization

The constrained optimization for linear program can be generalized for non linear objective and constraint functions. The aim of this problem is to optimize a function f subject to constraint function h.

Let there be a function $f(x1, x2)$ that maps from 2 dimensional space to a 1 dimensional space, and lets say we have to optimize the function subject to a constraint function $h(x1, x2) = 0$.

The function f will have a certain gradient function ∇f and the constraint function have a gradient function ∇h. ∇f might be perpendicular to ∇h, might be parallel or might have a component in the direction of ∇h. The maxima/minima of f subject to h will lie at a point on f where f is parallel to h, or $\nabla f = \lambda \nabla h$, let us denote the tangent to constraint function h by t_h (Figure 4).

Generally, at any general point

$$\nabla(f) \neq \nabla(h) \tag{43}$$

or

$$\nabla(f) * t_h \neq 0 \tag{44}$$

But at min/max.

$$\nabla(f) = \nabla(h) \tag{45}$$

or

$$\nabla(f) * t_h = 0 \tag{46}$$

4.1 Steps to implement General Constrained Optimization

In order to solve the constrained optimization problem, we use the following methodology:

In order to get the optimized solution we solve the following equation:

$$\nabla f + \lambda \nabla h = 0 \tag{47}$$

We are given the objective function and the constraint function as follows:

$$Of : min(f(x1, x2)) \tag{48}$$

$$Cf : h(x1, x2) = 0 \tag{49}$$

Let F be the Lagrangian equation for the following functions:

$$F(x1, x2, \lambda) = f(x1, x2) + \lambda h(x1, x2) \tag{50}$$

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \frac{\partial F}{\partial \lambda} \end{bmatrix} = 0$$

$$\nabla F = \begin{bmatrix} \frac{\partial f}{\partial x_1} + \lambda \frac{\partial h}{\partial x_1} \\ \frac{\partial f}{\partial x_2} + \lambda \frac{\partial h}{\partial x_2} \\ h(x1, x2) \end{bmatrix} = 0$$

Here λ is called the Lagrangian multiplier and by solving the above Lagrangian equation we will get to the desired optimized coordinates.

4.2 Example

Get the optimal coordinates for the following situation:

$$f(x, y) = xy \tag{51}$$

$$h(x, y) = \frac{x^2}{8} + \frac{y^2}{2} - 1 = 0 \tag{52}$$

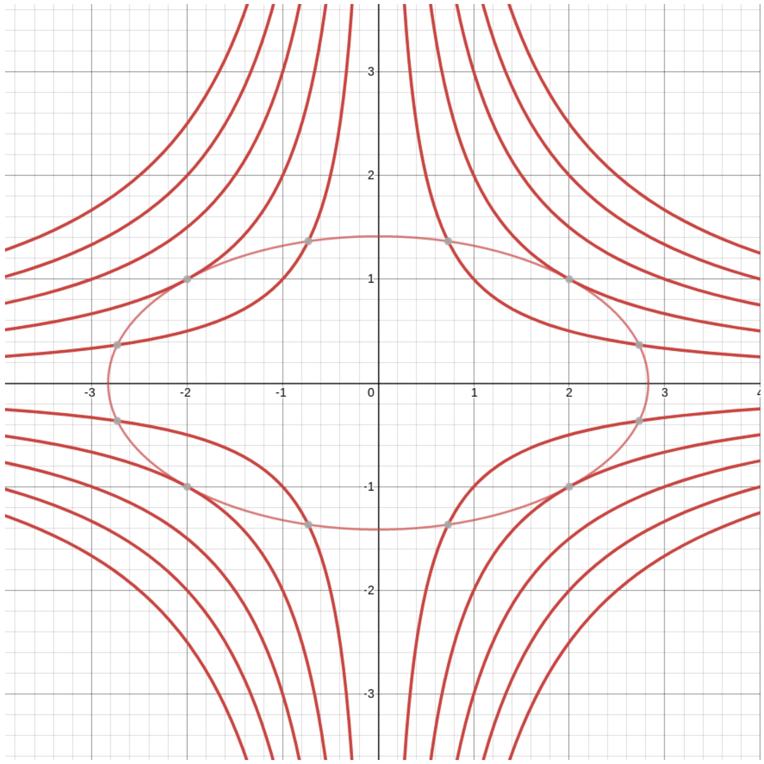

Figure 5: Contour plot of f and h.

Solution:
The Lagrangian equation is as follows:

$$F(x, y, \lambda) = xy + \lambda(\frac{x^2}{8} + \frac{y^2}{2} - 1) \tag{53}$$

$$\nabla F = \begin{bmatrix} y + \frac{\lambda h}{4} \\ x + \lambda y \\ \frac{x^2}{8} + \frac{y^2}{2} - 1 \end{bmatrix} = 0$$

We equate each term of the matrix as 0 and get 3 equations that we have to solve:

$$y = \frac{-\lambda x}{4} \tag{54}$$

$$x = -\lambda y \tag{55}$$

$$x^2 + 4y^2 = 8 \tag{56}$$

We solve (54), (55), (56) to get :
$(x, y) = (2, 1), (-2, 1), (-2, -1), (2, -1)$
If we draw the contour for $f(x, y) = C$ and $h(x, y)$, at $C = -2$, we will get the optimum solution for the objective equation.

As we can see from the Figure 5. that the solution of above equation represents the coordinates where $\nabla f = \nabla h$

5 References

- Constrained Optimization
- Plotting
- Khan Academy
- University of Washington Tutorial
- Gradient Descent