Optimization Methods		Date:	*06-Apr-2018*
Instructor: *Sujit Prakash Gujar*		Scribes:	Navya Khare Hemant Kasat Nitin Ramrakhiyani

Lecture 22: Constrained Non-linear Optimization

1 Recap

- In earlier lectures, we discussed about the two lemmas and the motivating theorem which states that if $d_0,\ d_1,\ \ldots,\ d_{n-1}$ are the directions generated through the conjugate gradient method, they are pair-wise Q-conjugate i.e. $d_i^T Q d_j\ =\ 0, \quad i \neq j$.
- It was noted that methods like conjugate gradient are known as Quasi-Newton methods as they employ the update rule similar to the Newton's method but approximate the Hessian H_k with another matrix B_k. Further, the value for B_{k+1} can be computed from B_k by either one update by a rank 1 matrix or updates by two rank 1 matrices. The update by a rank 1 matrix is suitable when the function is quadratic but when there are higher order terms, the update with two rank 1 matrices is more suitable.
- The BFGS method was also discussed which involves update by two rank one matrices. It was also noted that the BFGS also facilitates a closed form computation for inverse of the matrix B_k, avoiding the actual unstable matrix inverse computation.
- The Stochastic Gradient Descent (SGD) method was also discussed. The primary motivation for employing SGD is that earlier methods use all input points for updating minima values, which can be costly if there are a million input points. SGD allows computation of the update based on a randomly sampled single point or a batch of points.

2 Introduction

There are scenarios where optimization of non-linear functions is required subject to some equality constraints on the function variables.

$$\min_{x\ \in\ \mathbb{R}^n}\ f(x) \quad \text{subject to} \quad h_j(x)\ =\ 0,\ j\ =\ 1 \ldots l$$

To optimize for such scenarios the use of Lagrangian and Lagrange multipliers is proposed. We understand the underlying principle with help of an example (Example 1).

$$\min\ f(x)\ =\ x_1\ +\ x_2 \quad \text{subject to} \quad x_1^2\ +\ x_2^2\ =2$$

The above formulation requires minimizing the objective function $x_1\ +\ x_2$ given the fixation that we can only move on the curve specified by the constraint $x_1^2\ +\ x_2^2\ =2$. Movement at a point on a curve is generally along the tangent at that point. Figure 1 explains how the direction of movement on the curve (d), the gradient of the objective function ($\bigtriangledown f\ =\ [1\ 1]^T$) and gradient of the constraint function ($\bigtriangledown h\ =\ [2x_1\ 2x_2]^T$) are related.

It is straightforward to see that the direction of movement d is always going to be perpendicular to the gradient of the constraint function which is $\bigtriangledown h$ and hence we always have $d\ \perp\ \bigtriangledown h$ i.e. $d^T \bigtriangledown h\ =\ 0$. As seen in Figure 1, at point 1, which is a maxima for the function (objective value 2) given the constraint, both $\bigtriangledown f$ and $\bigtriangledown h$ are parallel and in the same direction. At point 2 however, $\bigtriangledown f$ is parallel to the direction of movement d and hence perpendicular to $\bigtriangledown h$. At point 3, which

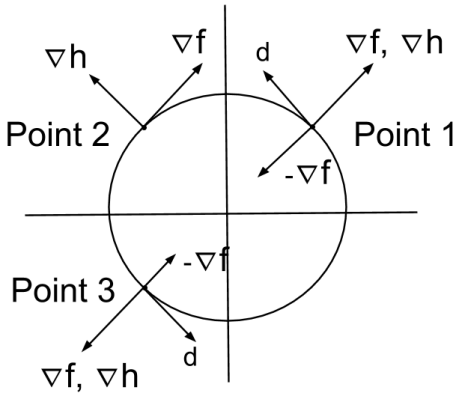


Figure 1: Diagram for Example 1

is the desired minima (objective value -2), both $\bigtriangledown f$ and $\bigtriangledown h$ are parallel and in opposite direction. So to get to the local minima, for a direction d and point x, if $d^T \bigtriangledown f(x) < 0$, there is scope for minimization and hence we keep moving in direction d. If $d^T \bigtriangledown f(x) > 0$, it is obvious to move in the opposite direction $-d$ such that $-d^T \bigtriangledown f(x) < 0$.

From the above discussion, we can infer that at local minima (x^*) as the function gradient and constraint gradient are parallel, $\bigtriangledown f(x^*) = \lambda \bigtriangledown h(x^*)$. We also have $d^T \bigtriangledown h = 0$. Therefore, $d^T \bigtriangledown f(x^*) = d^T \bigtriangledown h(x^*) = 0$.

3 Lagrangian and Lagrange Multipliers

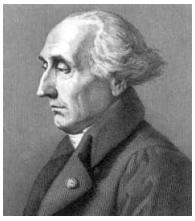
Writing generally, we need to minimize the function f and also maintain the notion of parallelism with the constraint gradient ($\bigtriangledown f = \bigtriangledown h$). So we combine both to write a joint unconstrained problem:

$$\min \mathcal{L}(x, \lambda) = f(x) + \sum_{j=1}^{l} \lambda_j h_j(x) \quad \text{where} \quad h_j(x) = 0, \forall j = 1 \dots l$$

The function $\mathcal{L}$ is called the *Lagrangian* and the multipliers λ_j are called the *Langrange multipliers*. The Lagrangian $\mathcal{L}$ is a function from $\mathbb{R}^{n+l}$ to $\mathbb{R}$ as the domain now covers the n dimensions of x as well as l values of the Lagrange multipliers. To now minimize the unconstrained non-linear problem posed by $\mathcal{L}$ we use the first-order necessary condition for function minimization. Therefore, $\bigtriangledown \mathcal{L}(x^*, \lambda^*) = 0$. We divide the first order necessary condition and write as:

$\bigtriangledown_x \mathcal{L}(x^*, \lambda^*) = 0$ (for first n variables) $\implies \bigtriangledown f(x^*) + \sum_{j=1}^{l} \lambda_j^* h_j(x^*) = 0$

$\bigtriangledown_\lambda \mathcal{L}(x^*, \lambda^*) = 0$ (for last l variables) $\implies h_j(x^*) = 0, \forall j = 1 \dots l$ (feasibility constraints)

Information Nugget (Source: Wikipedia)
Joseph-Louis Lagrange (Turin, 25 January 1736 - Paris, 10 April 1813) was an Italian Enlightenment Era mathematician and astronomer. He made significant contributions to the fields of analysis, number theory, and both classical and celestial mechanics. Lagrange was one of the creators of the calculus of variations, deriving the Euler-Lagrange equations for extrema of functionals. He also extended the method to take into account possible constraints, arriving at the method of Lagrange multipliers. 

The constraints discussed till now are primarily equality constraints. The Lagrangian formulation can also deal with inequality constraints and hence a problem of the form:

$$\min_{x \in \mathbb{R}^n} \; f(x) \quad \text{subject to} \quad g_i(x) \leq 0, \; i = 1 \dots m$$

We understand the Lagrangian formulation involving inequality constraints with an example (Example 2).

$$\min \; f(x) = (x_1 - 1.1)^2 + (x_2 - 1.1)^2 \quad \text{subject to} \quad g(x) : \; x_1^2 + x_2^2 - 1 \leq 0$$

Figure 2 explains how the direction of the gradient of the objective function ($\bigtriangledown f$) and gradient of the constraint function ($\bigtriangledown g$) are related. At the point of minima, the directions of $\bigtriangledown f$ and $\bigtriangledown g$ are opposite and parallel. This allows us to write:

$$\bigtriangledown f(x^*) = -\mu \bigtriangledown g(x^*) \implies \bigtriangledown f(x^*) + \mu \bigtriangledown g(x^*) = 0, \; \mu \geq 0. \tag{1}$$

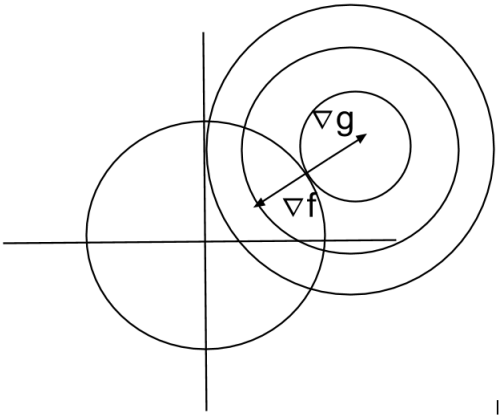


Figure 2: Diagram for Example 2

Now expressing the above using the Lagrangian formulation we get,

$$\min \; \mathcal{L}(x, \; \mu) = f(x) + \sum_{i=1}^{m} \mu_i g_i(x) \quad \text{where} \quad g_i(x) \leq 0, \; \forall i = 1 \dots m \; and \; \mu_i \geq 0$$

To now minimize the unconstrained non-linear problem posed by $\mathcal{L}$ we use the first-order necessary condition for function minimization. Therefore, $\bigtriangledown \mathcal{L}(x^*, \; \mu^*) = 0$. For the first n variables, we get the first order necessary condition as:

$$\bigtriangledown_x \mathcal{L}(x^*, \; \mu^*) = 0 \implies \bigtriangledown f(x^*) + \sum_{i=1}^{m} \mu_i^* g_i(x^*) = 0$$

Based on the remaining m variables, we specify the *Complementary Slackness* conditions, which state that $\mu_i^* g_i(x^*) = 0, \; \forall i = 1 \dots m$. If the constraint is satisfied with equality, the constraint is active, otherwise inactive. To understand the intuition behind these conditions, we consider the following cases:

- Case 1: When the constraint $g(x^*)$ is active.
  This case expresses that when the minima x^* is such that $g(x^*) = 0$ i.e. it is satisfied, we will have $\mu^* g(x^*) = 0$
- Case 2: When $g(x^*) < 0$ is valid.
  To satisfy $\bigtriangledown f(x^*) + \mu^* \bigtriangledown g(x^*) = 0$, discussed in 1 above, we always have $\mu^* = 0$ as $\bigtriangledown g(x^*) \neq 0$ and $\bigtriangledown f(x^*) = 0$.

So we can infer that at local minima, either $\mu_i^* > 0$ so $g(x^*) = 0$ or simply $\mu_i^* = 0$. Hence, the Complementary Slackness conditions hold for minima (x^*).

3.1 Karush-Kuhn-Tucker Conditions

In mathematical optimization, the Karush-Kuhn-Tucker (KKT) conditions, are first-order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. Allowing inequality constraints, the KKT approach to nonlinear programming generalizes the method of Lagrange multipliers, which allows only equality constraints. We discuss here the general framework for constrained non-linear optimization which deals with both equality as well as inequality constraints. The original problem is stated as follows:

$$\min_{x \in \mathbb{R}^n} f(x)$$
$$\text{such that}$$
$$g_i(x) \leq 0,\ i = 1 \dots m$$
$$h_j(x) = 0,\ j = 1 \dots l$$

Now expressing the above using the Lagrangian formulation we get,

$$\min\ \mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i=1}^{m} \mu_i g_i(x) + \sum_{j=1}^{l} \lambda_j h_j(x)$$

On applying the First Order necessary conditions we get the following results:

(I) First Order Conditions (FOC)

$\bigtriangledown_x \mathcal{L}(x^*, \mu^*, \lambda^*) = 0 \implies \bigtriangledown f(x^*) + \sum_{i=1}^{m} \mu_i^* \bigtriangledown g_i(x^*) + \sum_{h=1}^{l} \lambda_j^* \bigtriangledown h_j(x^*) = 0$

(II) Feasibility Conditions

$\bigtriangledown_\lambda \mathcal{L}(x^*, \mu^*, \lambda^*) = 0 \implies h_j(x^*) = 0,\ \forall j = 1 \dots l$

$g_i(x^*) \leq 0,\ \forall i = 1 \dots m$

(III) Complementary Slackness Conditions

$\bigtriangledown_\mu \mathcal{L}(x^*, \mu^*, \lambda^*) = 0 \implies \mu_i^* g_i(x^*) = 0,\ \forall i = 1 \dots m$

(IV) Dual Feasibility Conditions

$\mu_i^* \geq 0,\ \forall i = 1 \dots m$

Equations (I) to (IV) are collectively known as the *Karush-Kuhn-Tucker conditions.*

3.2 Necessity and Sufficiency of KKT Conditions

The above conditions are not necessary for minima and they become necessary only under certain specific scenarios such as those of Constraint Qualification (CQ). In this course we have skipped the discussion on CQ. As an example we consider, the following example which has a global minimum that does not satisfy the KKT conditions:

$$\min\ x \text{ such that } x^2 = 0$$

It is evident that the solution to the above problem is $x = 0$. Applying the KKT conditions for the above we get

$$1 + 2\lambda^* x^* = 0 \qquad \dots \text{based on (I)}$$
$$x^2 = 0 \qquad \dots \text{based on (II)}$$

The latter equation above implies $x^* = 0$ which violates the former equation and hence these conditions do not hold necessarily at minima.

They can be sufficient also under certain conditions which is when the function f is concave, the constraint g is convex and the constraint h is affine.

A slightly general set of conditions known as the *Fritz–John conditions* allow for accomodating examples of the above type. According to these conditions, the equation $1 + 2\lambda^* x^* = 0$ would get reformulated by introduction of a variable μ_0^* as $\mu_0^* + 2\lambda^* x^* = 0$ and by taking $x^* = 0$ and $\mu_0^* = 0$, the desired minima satisfies the conditions.

4 Water Filling Algorithm

Here we discuss the well-known Water Filling algorithm from signal and communication theory which involves calculation of the amount of power to be transmitted for maximum bit rate given a set of channels and limited amount of power.

Let there be n channels and x_i be the amount of power being sent on the i^{th} channel, $i = 1, 2, \ldots, n$. The bit-rate for the channel is given by $\sum_{i=1}^{n} \log(\alpha_i + x_i)$ such that $\sum_{i=1}^{n} x_i = 1$

The optimization problem is to select x such that Bit-rate is maximum, which is equivalent to minimizing negative of bit-rate. We can write the Lagrangian form of the problem as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda h(x) + \mu g_i(x)$$

$$\text{where } f(x) = -\sum_{i=1}^{n} \log(\alpha_i + x_i)$$

$$g_i(x) : -x_i \leq 0 \quad \text{i=1,2..,n}$$

$$h(x) : \sum_{i=1}^{n} x_i = 1$$

We now solve the above unconstrained function by applying the KKT conditions:

First Order Conditions:

$$\bigtriangledown_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0$$

On solving this we get

$$\lambda^* = \frac{1}{\alpha_i + x_i} + \mu^* \quad \forall i = 1,\ 2,\ \ldots\ n$$

Feasibility Conditions:

$$\sum_{i=1}^{n} x_i = 1$$

$$-x_i \leq 0 \quad \forall i = 1,\ 2,\ \ldots\ n$$

Complementary Slackness Conditions:

$$\mu_i^* x_i^* = 0 \quad \forall i = 1,\ 2,\ \ldots\ n$$

Dual Feasibility Conditions:

$$\mu_i^* \geq 0 \quad \forall i = 1,\ 2,\ \ldots\ n$$

To get the value for x we solve by exploring two possible cases:
Case-1:
If $\lambda^* < \frac{1}{\alpha_i} \quad \forall i = 1, 2, ..n$ holds then from the first order necessary conditions we get,
$\lambda^* = \frac{1}{\alpha_i + x_i} + \mu^* \implies \lambda^* \geq \frac{1}{\alpha_i + x_i}$
This implies that x_i is strictly greater than 0 i.e. $x_i > 0$
But then for the Complementary Slackness conditions to hold we must have $\mu_i^* = 0$

Case-2:
If $\lambda^* \geq \frac{1}{\alpha_i} \quad \forall i = 1, 2, ..n$ and $x_i^* > 0$ hold, then the first order necessary conditions will imply $\mu_i^* < 0$. But this is infeasible as the Dual Feasibility conditions of $\mu_i^* \geq 0, \forall i$ hold.
Hence, $x_i = 0 \quad \forall i = 1, \ 2, \ \ldots \ n$

Both these cases can be written as:

$$x_i^* = \begin{cases} \frac{1}{\lambda^* - \alpha_i} & \text{if} \quad \lambda < \frac{1}{\alpha_i} \\ 0 & \text{else} \end{cases}$$

which can further be stated as :

$$x_i^* = \max(0, \ \frac{1}{\lambda^*} - \alpha_i) \quad \forall i = 1, \ 2, \ \ldots \ n$$

We refer to Figure 3 ,to understand the intuition behind the above solution. We understand the graph in Figure 3 as a one-dimensional bucket and we need to fill one litre of water (based on our $h(x)$). The water here is the power. We begin with the deepest of the channels i.e. the one with the lowest α_i and keep filling till we hit the next deepest channel. We keep filling till we run out of power.

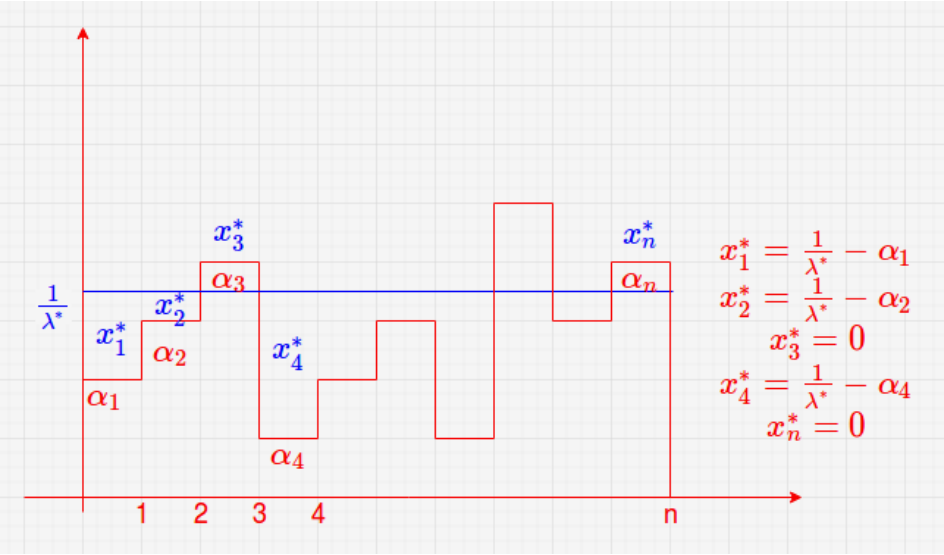


Figure 3: Water Filling Algorithm - Intuition