Sukkur Institute of Business Administration University
Department of Electrical Engineering.

Optimization Techniques

course lecture notes

Instructor
Dr. Muhammad Asim Ali

Spring 2020

Contents

Introduction

Optimization techniques find application in every facet of modern human life. From design to manufacturing and operation. It find application in a set of diverse fields ranging from agriculture, medicine, industry, transportation, stock markets , economics and now more recently to different applications of electrical engineering including signal processing.

Although optimization has been around since long time however Dantzig in 1947 proposed the first algorithm which automized the optimization problem. This algorithm popularly known as the "Simplex Algorithm". The specific field of optimization in known as *Linear Programming*.

The theory of optimization has evolved exponentially over the past century (for obvious economic implications) and has matured to become a technology.

As the literature on this theory points out the real problem in optimization is actually to decided whether a problem is optimizeable and how to formulate it mathematically so that it can be optimized *easily*.

Chapter 1

Introduction

Outline

The objective of this chapter is to familiarize the students with the fundamental concepts of mechanics which will form basis of pivotal concepts of Robotics. The topics included here are

Ⓐ Introduction
Ⓑ Vectors and Vector Spaces
Ⓒ Independence, Basis
Ⓓ The 4 Spaces
Ⓔ Rank and its implications.
Ⓕ Matrix Decomposition

1.1 Vectors Spaces and Subspaces

Definition: The space $\mathbb{R}^n$ consists of all column vectors $\mathbf{v}$ with n components.

The components of $\mathbf{v}$ are real numbers, which is the reason for the letter $\Re$. A vector whose n components are complex numbers lines in space $\mathbb{C}$.
The vector space $\mathbb{R}^2$ is represented by the usual xy plane. Each vector v in $\mathbb{R}^2$ has two components. The word space 'asks' us to think all possible vectors available in the plane. Each vector has x and y coordinates of a point in the plane. Similarly the vectors in $\mathbb{R}^3$ correspond to points (x, y, z) in three dimensional space. The

one-dimensional space $\mathbb{R}^1$ is a line (like x-axis). General vectors can be represented in a variety of notations. $\begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^3 \quad (1,0,1,1,0)$ is $\in\ \mathbb{R}^5 \quad \begin{bmatrix} 1+i \\ 1-i \end{bmatrix} \in \mathbb{C}^2$

1.2 Vector SubSpaces

Vector $\mathbb{R}^n$ is a set of all vectors such that

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \text{where} x_i \in \mathbb{R}^n, 1 \leq i \leq n$$

V is a linear subspace (subset) of $\mathbb{R}^n$ if following conditions are met include

- The subspace should include the all zero vector $\mathbf{0}$.
- If $\mathbf{v}$ is in a vector space then for any scalar $c \in \mathbf{R}$ cv is also in the vector space.
- If $\mathbf{v}$ and $\mathbf{w}$ are two vectors in a vector space than their sum i.e. $\mathbf{v} + \mathbf{w}$ is also in a vector space.

Some Examples

$$S_1 = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3 \right\}.$$

$$S_2 = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3 | x_1 > 0 \right\}.$$

$$S_3 = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3 | x_1 - 4x_2 + 5x_3 = 2 \right\}.$$

$$S_4 = \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 | y > x^2 \right\}.$$

$$S_5 = \left\{ \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 | 3x = 2y \right\}.$$

1.3 Matrices

Matrix simply speaking is an arrangement (array) of real and/or complex numbers. Matrix is a set of column vectors in $\mathbb{R}^m$ if we have n-columns then the dimension of matrix is $m \times n$.
Matrix can be used to describe and solve a plenty of problems, a set of linear equation

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

could be translated into a matrix form

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

transformation of vectors can be performed through matrix operation

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \quad \begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{bmatrix} \quad \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$

use of matrix as a convolution operation

$$\begin{bmatrix}y[0]\\ y[1]\\ \vdots\\ y[n-1]\end{bmatrix} = \begin{bmatrix}h[0] & 0 & 0 & \cdots & \cdots & 0\\ h[1] & h[0] & 0 & \cdots & \cdots & 0\\ h[2] & h[1] & \ddots & \cdots & \cdots & 0\\ \vdots & \ddots & \ddots & \cdots & h[0] & 0\\ \vdots & \ddots & \ddots & h[2] & h[1] & h[0]\end{bmatrix} \begin{bmatrix}x[0]\\ x[1]\\ \vdots\\ x[n-1]\end{bmatrix}$$

1.4 Vector Span

Span of a vector subspace $\mathbf{v}$ is a set of all linear combinations $\mathbf{x} = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n$, Formally speaking

$$\text{span}(\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n) = \left\{ c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n | c_i \in \mathbb{R} \text{ for } 1 \leq i \leq n \right\}$$

1.5 Linear Independence

We have $\mathbf{v}_1, \cdots, \mathbf{v}_n \in \mathbb{R}^n$, a linear combination $c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n$ with $c_i \in \mathbb{R}$, for a set of vectors $A = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n\end{bmatrix}$, column space is just the span of vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n$. If a set of vectors is linearly independent then it forms a basis for column space of A.
Likewise $A = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n\end{bmatrix}$ a set of vectors is called linearly dependent if and only if the vector equation $c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n = \mathbf{0}$ has a solution with at least $c_i = 0$.

Basis

A basis for a vector space is a sequence of vectors which are linearly independent and spans the space. The representation of any vector as combination of basis vectors is unique.

The columns of $\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ produce the 'standard basis' for $\mathbb{R}^2$.

The basis is not unique!

Row Reduced Echelon Form

It is well known fact that operations on the rows of a matrix do not affect the matrix. Through systematic manipulation of the rows of a matrix **A**.

> **R**=rref(**A**) of a matrix is unique. $\Rightarrow \mathbf{R}^T \neq \text{rref}(\mathbf{A^T})$.

How to obtain the original matrix **A** from **R**? There exists a set of linear operations such that $\mathbf{EA} = \mathbf{R}$, therefore $\mathbf{A} = \mathbf{E}^{-1}\mathbf{R}$. The first$r$ columns of $\mathbf{E}^{-1}$ are pivot rows of **A**.

$$\begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 4 \\ 1 & 1 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

What can we say about the dependence of column space?

1.6 Space of a Matrix

These are some the very important properties of the matrix which provide a very deep insight into the

Column Space of a matrix **A** i.e. C(**A**) contains all linear combinations of columns of **A**. This combination includes **Ax**. For a linear system $\mathbf{Ax} = \mathbf{b}$ to have a solution **x**, **b** must be in column space of **A**.

$2x + 3y = 6$ $4x - 6y = 12$ which is $\begin{bmatrix} 2 & 3 \\ 4 & -6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \end{bmatrix}$ Does **b** lie in the column space of **A**?	$2x + 3y = 6$ $4x - 6y = 12$ which is $\begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$ Does **b** lie in the column space of **A**?

If a matrix has dimensions $m \times n$ then each column has m components thus belongs to $\mathbb{R}^m$. Thus the column space $\mathrm{C}(\mathbf{A})$ is a $\mathbb{R}^m$ dimensional subspace.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Null Space of a matrix $\mathbf{A}$ i.e. $\mathrm{N}(\mathbf{A})$ consists of all the solutions of $\mathbf{Ax} = \mathbf{0}$. For a $m \times n$ matrix $\mathbf{A}$ the $\mathrm{N}(\mathbf{A})$ is in $\mathbb{R}^n$ dimensional subspace.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 8 & 6 & 16 \end{bmatrix} \Rightarrow \mathrm{rref}(\mathbf{A}) = \begin{bmatrix} \textcircled{1} & 0 & 2 & 0 \\ 0 & \textcircled{1} & 0 & 2 \end{bmatrix}$$

This simplification implies that x_1 and x_2 are pivot variables while x_3 and x_4 are free variables. Now solving

$$\begin{bmatrix} \textcircled{1} & 0 & 2 & 0 \\ 0 & \textcircled{1} & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which yields into following equalities

$$x_1 = -2x \qquad x_2 = -2y$$

letting $x = 1$ and $y = 0$, $x = \begin{bmatrix} -2 & 0 & 1 & 0 \end{bmatrix}^T$ or equivalently $x = 0$ and $y = 1$, $x = \begin{bmatrix} 0 & -2 & 0 & 1 \end{bmatrix}^T$.

just do a quick check to see if indeed $\mathbf{Ax} = \mathbf{0}$.

Row Space of a matrix is a subspace in $\mathbb{R}^n$ spanned by rows. The row space of $\mathbf{A}$ is $\mathrm{C}(\mathbf{A}^T)$. $\rightarrow$ It is column space of $\mathbf{A}^T$.

Left Null Space We solve $\mathbf{A}^T\mathbf{y} = \mathbf{0}$. The vector $\mathbf{y}$ goes on to the left side of $\mathbf{A}$ when the equation is written as $\mathbf{y}^T\mathbf{A} = \mathbf{0}^T$. Matrices $\mathbf{A}$ and $\mathbf{A}^T$ are different; therefore their column spaces and null spaces are also different.

The Four Spaces

For a Full rank matrix $\mathbf{A}$ with dimensions $m \times n$.

1. The column space is $\mathrm{C}(\mathbf{A})$, a subspace of $\mathbb{R}^m$.
2. The Null space is $\mathrm{N}(\mathbf{A})$, a subspace of $\mathbb{R}^n$.
3. The row space is $\mathrm{C}(\mathbf{A}^T)$, a subspace of $\mathbb{R}^n$.
4. The Left Null space is $\mathrm{C}(\mathbf{A}^T)$, a subspace of $\mathbb{R}^m$.

consider the following matrices

$\mathbf{A} = \begin{bmatrix} 1 & 4 \\ 2 & 7 \\ 3 & 5 \end{bmatrix}$ and $\mathbf{A}^T = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 7 & 5 \end{bmatrix}$ with $m = 3$ and $n = 2$. What is the column and row space of $\mathbf{A}$. The row space is all of $\mathbb{R}^2$.

The row space $\mathrm{C}(\mathbf{A}^T)$ and column space $\mathrm{C}(\mathbf{A})$ have same dimensions as the rank r. The Null space $\mathrm{N}(\mathbf{A})$ and the left null space $\mathrm{N}(\mathbf{A}^T)$ have dimensions $n - r$ and $m - r$ to make up full n and m.

Example:

$$\text{For some Matrix } \mathbf{A} \text{ it RREF is} \mathbf{R} = \begin{bmatrix} \textcircled{1} & 3 & 5 & 0 & 7 \\ 0 & 0 & 0 & \textcircled{1} & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Row space has dimension 2 (i.e. Rank), the first and fourth rows are the basis of Row space. Column space has dimension 2 as all columns can be obtained through linear combination of column 1 and 4. Null space has dimension $n - r$ i.e. $5 - 2 = 3$, there are **3** free variables. Left Null space has dimension $m - r$ i.e. $3 - 2 = 1$

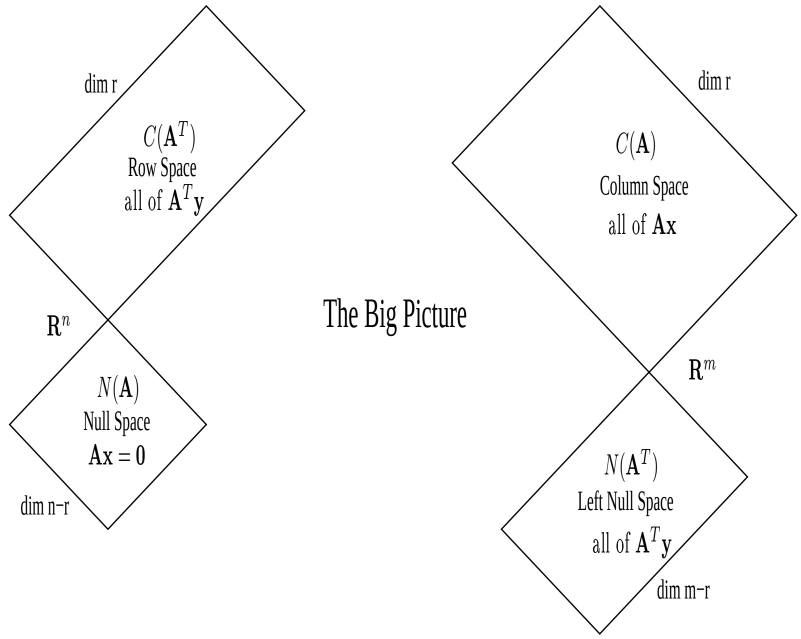


Figure 1.1: An illustration of relation between spaces span by columns and rows.

1.7 Rank of Matrix

For a set of m linear equations with n unknowns, the set of possible solutions depends on the rank of the matrix.

$r = m$	and	$r = n$	Square and Invertible	$\mathbf{Ax} = \mathbf{b}$ has one solution
$r = m$	and	$r < n$	Short and wide	$\mathbf{Ax} = \mathbf{b}$ has ∞ solutions
$r < m$	and	$r = n$	Tall and thin	$\mathbf{Ax} = \mathbf{b}$ has one or zero solution
$r < m$	and	$r < n$	Not Full rank	$\mathbf{Ax} = \mathbf{b}$ has zero or ∞ solution

The rref($\mathbf{A}$)=$\mathbf{R}$ will fall in the same category as the pivot columns happens first.

$$\text{Four Types} \quad \mathbf{R} = \begin{bmatrix}\mathbf{I}\end{bmatrix} \quad \begin{bmatrix}\mathbf{I} & \mathbf{F}\end{bmatrix} \quad \begin{bmatrix}\mathbf{I} \\ \mathbf{0}\end{bmatrix} \quad \begin{bmatrix}\mathbf{I} & \mathbf{F} \\ 0 & 0\end{bmatrix}$$

Case 1 and 2 have full rank $r = m$, case 1 and 3 have full rank $r = n$. Case 4 is the most general in theory and least common in practice.

1.8 Graphical Visualization of $\mathbf{Ax} = \mathbf{b}$

$$\mathbf{Ax} = \mathbf{b}$$

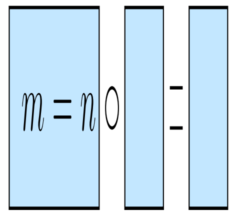
- $\mathbf{A}_{[m \times n]}$ can be square, tall (over determined) or fat (under-determined).
- Either rank[$\mathbf{A}$]=rank[$\mathbf{A}|\mathbf{b}$] (consistent)
  rank[$\mathbf{A}$]$\neq$ rank[$\mathbf{A}|\mathbf{b}$] (inconsistent)
- Either A is Full rank (i.e. rank[$\mathbf{A}$]=min(m,n))
  rank deficient (i.e. rank[$\mathbf{A}$]$<$ min(m,n))
- Looks like there are 12 possibilities but only 10 can exist.

	A- Full Rank	**A**- Not Full Rank
A- Square (consistent)	One Sol.	Infinite Sol.
A- Square (inconsistent)	Can't Happen	Infinite LS Sol.
A- Tall (consistent)	One Sol.	Infinite Sol.
A- Tall (inconsistent)	One LS Sol.	Infinite LS Sol.
A- Fat (consistent)	Infinite Sol	Infinite Sol.
A- Fat (inconsistent)	Can't Happen	Infinite LS Sol.

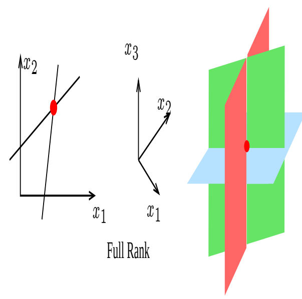
Square Matrix

$\mathbf{Ax} = \mathbf{b}$ consistent

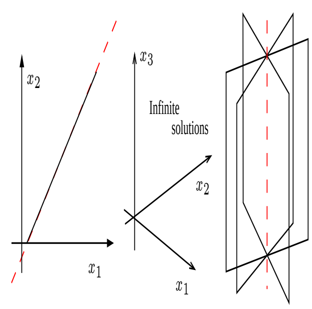
Rank($\mathbf{A}|\mathbf{b}$)=Rank($\mathbf{A}$)



rank($\mathbf{A}$)=r=m=n (full rank)

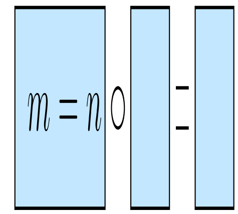


rank($\mathbf{A}$)=r< m=n (Not full rank)

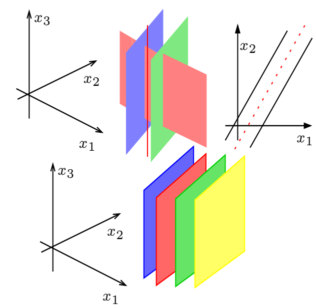


$\mathbf{Ax} \neq \mathbf{b}$ inconsistent

Rank($\mathbf{A}|\mathbf{b}$)≠Rank($\mathbf{A}$)



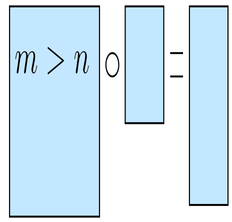
rank($\mathbf{A}$)=r< m=n (full rank)



Tall Matrix

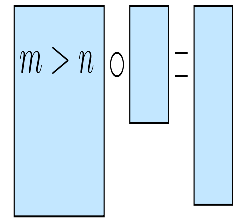
$\mathbf{Ax} = \mathbf{b}$ consistent

Rank($\mathbf{A}|\mathbf{b}$)=Rank($\mathbf{A}$)

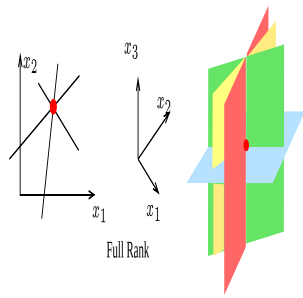


$\mathbf{Ax} \neq \mathbf{b}$ inconsistent

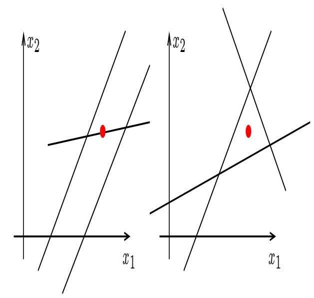
Rank($\mathbf{A}|\mathbf{b}$)$\neq$Rank($\mathbf{A}$)



rank($\mathbf{A}$)=r=n< m (full rank)

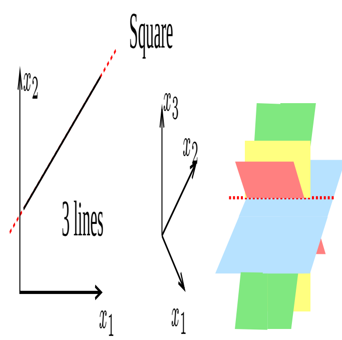


rank($\mathbf{A}$)=r=n< m (full rank)



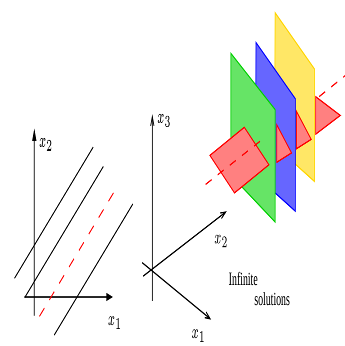
Rank($\mathbf{A}$)=r< n< m

(Not Full Rank)



Rank($\mathbf{A}$)=r< n< m

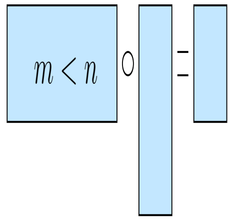
(Not Full Rank)



Fat Matrix

$\mathbf{Ax} = \mathbf{b}$ consistent

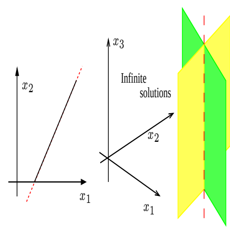
Rank($\mathbf{A}|\mathbf{b}$)=Rank($\mathbf{A}$)



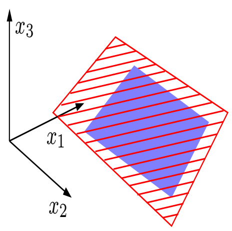
rank($\mathbf{A}$)=r=n< m (full rank)

Rank($\mathbf{A}$) $= r < n < m$

(Not Full Rank)

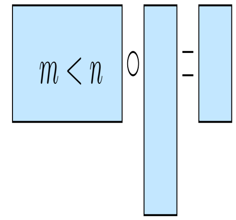


Rank($\mathbf{A}$) $= r < n < m$



$\mathbf{Ax} \neq \mathbf{b}$ inconsistent

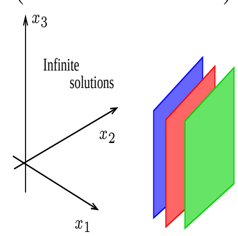
Rank($\mathbf{A}|\mathbf{b}$)$\neq$Rank($\mathbf{A}$)



rank($\mathbf{A}$) $= r < n < m$ (Not full rank)

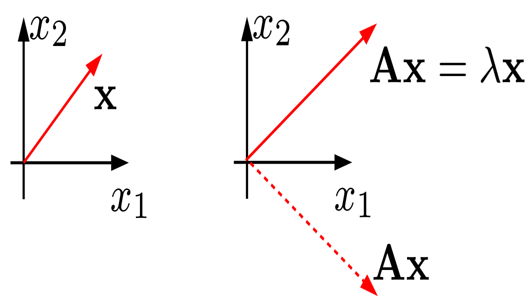
Rank($\mathbf{A}$) $= r < n < m$

(Not Full Rank)



1.9 Eigen-Values and Eigen-Vectors

Thus far we have been concerned with $\mathbf{Ax} = \mathbf{b}$, now we concern ourselves with $\mathbf{Ax} = \lambda\mathbf{x}$.



In this particular case all the set of vectors $\mathbf{x}$ which when multiplied with a matrix $\mathbf{A}$ yield $\lambda\mathbf{x}$. where λ is an eigen-value. The eigenvalues of can be computed as

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \mathbf{0} \tag{1.1}$$

(1.1) is known as the '*characteristic polynomial*' of the matrix $\mathbf{A}$. If this polynomial has a non-zero solution, $\mathbf{A} - \lambda\mathbf{I}$ is non-invertible. The determinant of $\mathbf{A} - \lambda\mathbf{I}$ must be equal to zero. This is how we recognize an eigen-value λ.

The characteristic polynomial $\det(\mathbf{A} - \lambda\mathbf{I})$ involves only λ and not $\mathbf{x}$. When $\mathbf{A}$ is $n \times n$, the polynomial has degree n. For each Eigen-value λ solve $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$ or equivalently $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ to find an eigen-vector $\mathbf{x}$. The eigen-vectors essentially make up the null space of $\mathbf{A} - \lambda\mathbf{I}$.

Example: Consider $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ is an already singular (zero-determinant). Find it's λ's and $\mathbf{x}$'s.

Subtract λ from the diagonal to find $\mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{bmatrix}$ i.e. $(\lambda^2 - 5 - \lambda) = 0$ which leads to two solutions i.e. $\lambda = 0, 5$.

$(\mathbf{A} - 0\mathbf{I})\mathbf{x} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ yields into eigen-vector $\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$ for $\lambda_1 = 0$.

$(\mathbf{A} - 5\mathbf{I})\mathbf{x} = \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ yields into eigen-vector $\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ for $\lambda_2 = 5$.

Eigen-values fill up the null-space $\mathbf{A}\mathbf{x} = \mathbf{0}$. If $\mathbf{A}$ is invertible, 0 is not an eigen-value, $\mathbf{A}$ is shifted by multiple of $\mathbf{I}$ to make it singular.

Compute the determinant of $\mathbf{A} - \lambda\mathbf{I}$, with λ subtracted along the diagonal, the determinant starts with with λ^n or $-\lambda^n$. It is polynomial in degree n.

Find the roots of this polynomial by solving $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. The $n-$roots are the $n-$ eigen-values of A. They make $\mathbf{A} - \lambda\mathbf{I}$ singular.

For each eigen-value λ, solve $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$ to find eigen-vector $\mathbf{x}$.

The eigen-values of $\mathbf{A}^k$ for any positive integer k are $\lambda_1^k, \cdots, \lambda_n^k$.

A matrix $\mathbf{A}$ is invertible if every eigen-value is non-zero.

If matrix $\mathbf{A}$ is invertible then its eigen-values are $\frac{1}{\lambda_1}, \cdots, \frac{1}{\lambda_n}$.

For an $n \times n$ matrix without a set of n independent eigen-vectors it is not possible to have basis.
other important properties to note:

$$\lambda_1 + \lambda_2 + \cdots + \lambda_n = \text{trace} = a_{11} + a_{22} + \cdots + a_{nn}$$
$$\det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \cdots \lambda_n$$

Some more facts

1. Shuffling the rows of a matrix would change the Eigen-values.
2. The product of $n-$eigenvalues equals the determinant.
3. The sum of $n-$eigenvalues equals the sum of n diagonal entreies.
4. The sum of the main diagonal is called the trace of a matrix.
5. The eigen-values of $\mathbf{A}^2$ and $\mathbf{A}^{-1}$ are λ^2 and λ^{-1} with same eigen-vectors.

1.10 Diagonalization of a Matrix

The combination of eigen-values and eigen-vectors can be expressed in a matrix notation as $\mathbf{AX} = \mathbf{X}\Lambda$, where $\mathbf{X}$ is the set of eigen-vectors and Λ is diagonal matrix with eigen-values λ on the diagonal.

$$\mathbf{AX} = \mathbf{A}\begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \lambda_1\mathbf{x}_1 & \cdots & \lambda_n\mathbf{x}_n \end{bmatrix}\begin{bmatrix} \lambda_1\mathbf{x}_1 & \cdots & \lambda_n\mathbf{x}_n \end{bmatrix}$$
$$\begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n \end{bmatrix}\begin{bmatrix} \lambda_1 & & \\ & \ddots & \lambda_n \end{bmatrix} = \mathbf{X}\Lambda$$

If an $n \times n$ matrix $\mathbf{A}$ has $n-$linearly independent eigen-vectors $\mathbf{x}_1$,$\mathbf{x_2}$,$\cdots$, $\mathbf{x_n}$, put them on the columns of an eigen-vector matrix $\mathbf{X}$. Then $\mathbf{X}^{-1}\mathbf{AX}$ is an eigen-value matrix Λ.

$$\mathbf{AX} = \mathbf{X}\Lambda \quad \text{is} \quad \mathbf{X}^{-1}\mathbf{AX} \quad \text{or} \quad \mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$$

The important application of this feature:

$$\begin{aligned}\mathbf{A}^k &= (\mathbf{X}\Lambda\mathbf{X}^{-1})(\mathbf{X}\Lambda\mathbf{X}^{-1})\cdots(\mathbf{X}\Lambda\mathbf{X}^{-1}) \\ &= (\mathbf{X}\Lambda^k\mathbf{X}^{-1})\end{aligned}$$

Suppose if the eigen-vector matrix $\mathbf{\Lambda}$ is fixed and we can change the eigen vector matrix $\mathbf{X}$ we get a whole family of $\mathbf{X\Lambda X}^{-1}$ all with same eigen-values $\mathbf{\Lambda}$. All those matrices with same $\mathbf{\Lambda}$ are called **Similar**.
A matrix is said to be a **Symmetric Matrix** if it has n real valued eigen-values λ_i and $n-$orthonormal eigen-vectors $\mathbf{q}_1,\mathbf{q}_2,\cdots,\mathbf{q}_n$.
Every real symmetric matrix $\mathbf{S}$ can be diagonalized as $\mathbf{S} = \mathbf{Q\Lambda Q}^{-1} = \mathbf{Q\Lambda Q}^T$.
Every matrix is said to be positive semi-definite matrix if for any non-zero vector $\mathbf{x}$ the product $\mathbf{x}^T\mathbf{Ax} \geq 0$.

- if all the columns of a matrix are independent the matrix is positive semi-definite.
- for a semi-definite matrix all eigen-values are positive.

1.11 Singular Value Decomposition

Singular Value Decomposition is one of the most significant milestones in linear algebra. $\mathbf{A}$ is a $m \times n$ matrix square or rectangular. Its rank is r. It is possible to diagonalize $\mathbf{A}$ but not as $\mathbf{S\Lambda S}^{-1}$. The eigen-vectors in $\mathbf{S}$ are not always orthogonal, there are not always enough eigen-vectors. $\mathbf{Ax} = \lambda\mathbf{x}$ requires $\mathbf{A}$ to be a square matrix. The singular vector of $\mathbf{A}$ solve all those problems in a perfect way.
The price of this is that we have to calculate a set of two singular vectors $\mathbf{u}$'s and $\mathbf{v}$'s. The $\mathbf{u}$'s are eigen-vectors of $\mathbf{AA}^T$ and the $\mathbf{v}$'s are the eigen-vectors of $\mathbf{A}^T\mathbf{A}$. Since both those matrices are symmetric, there eigen-vectors can be chosen to be orthonormal.

Fundamental Concept

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \text{is same as} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a & a & c & c & e & e \\ a & a & c & c & e & e \\ a & a & c & c & e & e \\ a & a & c & c & e & e \\ a & a & c & c & e & e \\ a & a & c & c & e & e \end{bmatrix} \text{is same as} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} a & a & c & c & e & e \end{bmatrix}$$

$\mathbf{A}$ is diagonalized $\quad \mathbf{A}\mathbf{v}_1 = \sigma_1 \mathbf{u}_1 \quad \mathbf{A}\mathbf{v}_2 = \sigma_2 \mathbf{u}_2 \quad \cdots \quad \mathbf{A}\mathbf{v}_r = \sigma_r \mathbf{u}_r$

The eigen-vectors $\mathbf{v}_1, \mathbf{v}_2 \cdots \mathbf{v}_r$ are in the row space of $\mathbf{A}$. The vectors $\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_r$ are in the column space of $\mathbf{A}$. The singular $\sigma_1, \sigma_2, \cdot, \sigma_r$ are all positive numbers. When $\mathbf{v}$'s and $\mathbf{u}$'s go into the columns of $\mathbf{U}$ and $\mathbf{V}$, we have $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The σ's can go into the diagonal matrix Σ.
So just as $\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i$ led to the diagonalization of $\mathbf{AS} = \mathbf{S\Lambda}$, the equations $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$ can be expressed in the matrix notation as $\mathbf{AV} = \mathbf{U\Sigma}$.

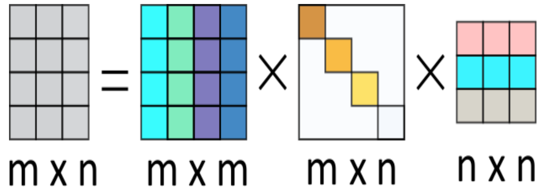
$$\underset{m\times n}{\begin{bmatrix} \\ \mathbf{A} \end{bmatrix}} \underset{n\times r}{\begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r \end{bmatrix}} = \underset{m\times r}{\begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_r \end{bmatrix}} \underset{r\times r}{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}}$$

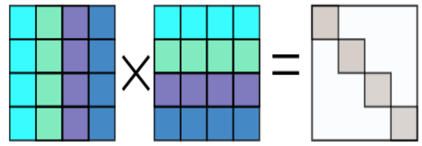
In short-hand notation

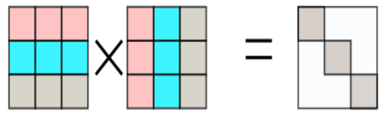
$$\mathbf{A} = \mathbf{U\Sigma V}^T = \mathbf{u}_1 \sigma_1 \mathbf{v}_1^T + \cdots + \mathbf{u}_r \sigma_r \mathbf{v}_r^T$$

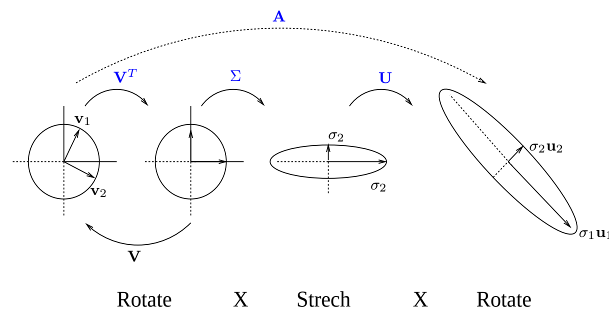
which is equal to splitting the matrix $\mathbf{A}$ into r matrices of rank 1.

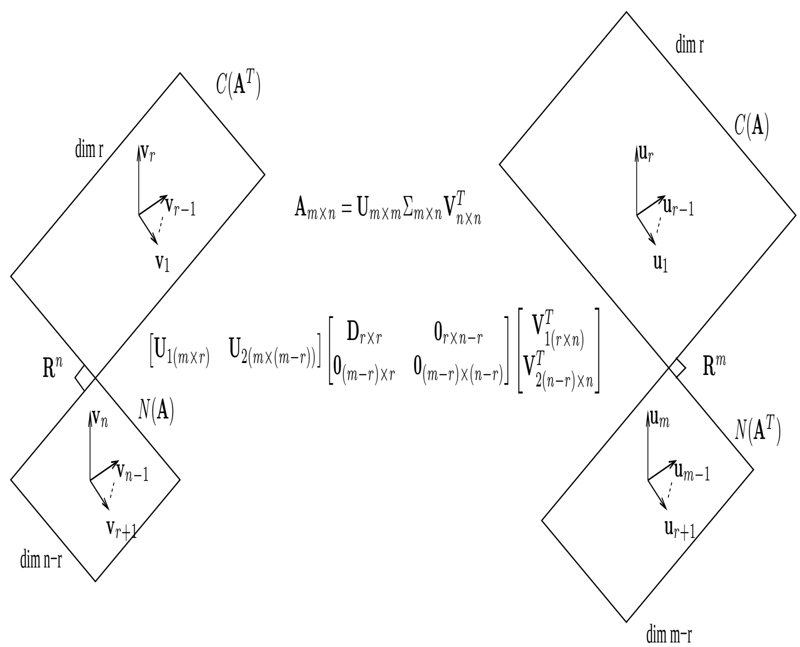
The $\mathbf{v}$'s and $\mathbf{u}$'s account for the row space and the column space of $\mathbf{A}$, we need $n-r$ more $\mathbf{v}$'s and $m-r$ $\mathbf{u}$'s from $N(\mathbf{A})$ and $N(\mathbf{A}^T)$. They can be orthonormal bases for those two null spaces and automatically orthogonal to the first r $\mathbf{v}$'s and $\mathbf{u}$'s.











Combine all those **u**'s and **v**'s in **V** and **U** now those matrices become square. We still have the $\mathbf{AV} = \mathbf{U\Sigma}$.

$$\underset{m\times n}{\begin{bmatrix} \mathbf{A} \end{bmatrix}} \underset{n\times n}{\overset{\mathbf{V}}{\begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_r & \mathbf{v}_{r+1} & \cdots & \mathbf{v}_n \end{bmatrix}}} = \underset{m\times m}{\overset{\mathbf{U}}{\begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_m \end{bmatrix}}} \underset{m\times n}{\overset{\mathbf{\Sigma}}{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \\ & & \end{bmatrix}}}$$

where Σ is a $m \times n$ matrix with $m - r$ new zero rows and $n - r$ new zero columns. The dimensions of $\mathbf{U}$ and $\mathbf{V}$ and $\mathbf{\Sigma}$ have changed and $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$ and $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$. $\mathbf{V}$ is now an orthogonal square matrix with inverse $\mathbf{V}^{-1} = \mathbf{V}^T$. So $\mathbf{AV} = \mathbf{U\Sigma}$.
When $\mathbf{U\Sigma V}^T$ (singular values) are the same as $\mathbf{S\Lambda S}^{-1}$ (eigen-values)?
We need orthogonal eigen-vectors in $\mathbf{S} = \mathbf{U}$ we need non-negative eigen-values $\mathbf{\Lambda} = \mathbf{\Sigma}$. So $\mathbf{A}$ must be a positive semi-definite (or definite) symmetric matrix $\mathbf{Q\Lambda S}^T$.

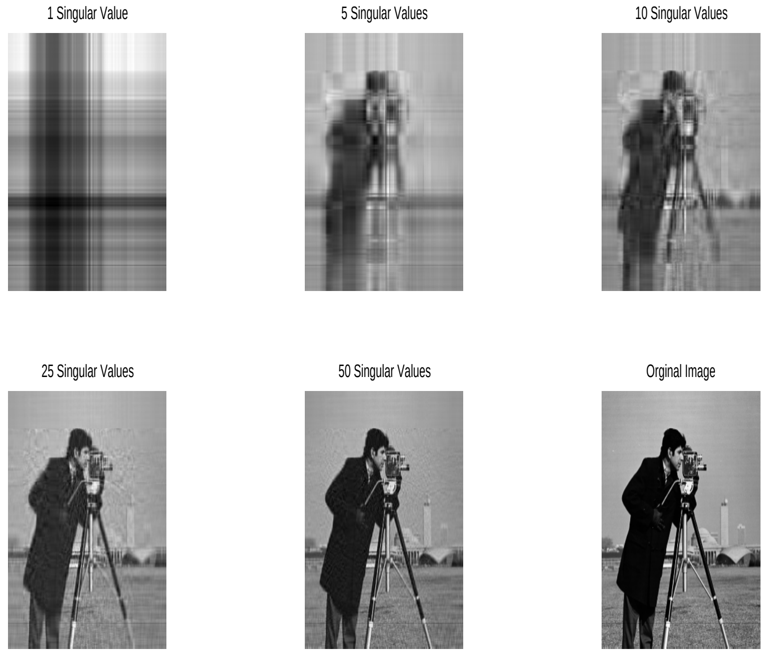
Applications of SVD

Image Compression   Reduction of Dimension   Feature Extraction

How to Calculate SVD

$$\begin{aligned} \mathbf{A}^T\mathbf{A} &= (\mathbf{U\Sigma V}^T)^T(\mathbf{U\Sigma V}^T) & \mathbf{AA}^T &= (\mathbf{U\Sigma V}^T)(\mathbf{U\Sigma V}^T)^T \\ &= \mathbf{V\Sigma U}^T\mathbf{U\Sigma V}^T & &= \mathbf{U\Sigma V}^T\mathbf{V\Sigma U}^T \\ &= \mathbf{V\Sigma}^2\mathbf{V}^T & &= \mathbf{U\Sigma}^2\mathbf{U}^T \\ \mathbf{A}^T\mathbf{AV} &= \mathbf{V\Sigma}^2 & \mathbf{AA}^T\mathbf{U} &= \mathbf{U\Sigma}^2 \end{aligned}$$

Application of SVD



Chapter 2

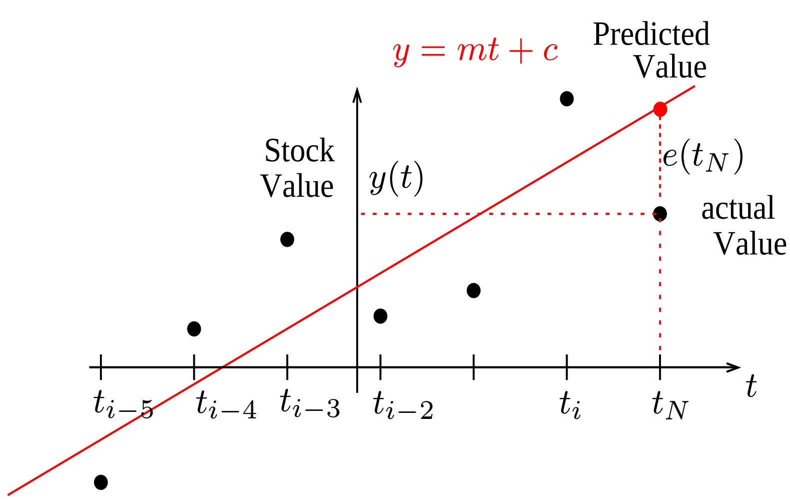
Least Square Solution

Outline

The objective of this chapter is to study a fundamental set of un-constrained optimization techniques. We also study some of the well-known adaptive algorithms used to solve such problems **online**. The fundamental concepts covered in this chapter are

- Ⓐ Least Square Solution
- Ⓑ Weighted Least Squares
- Ⓒ Constrained Least Squares
- Ⓒ LMS Algorithm
- Ⓓ RLS Algorithm
- Ⓖ Lagrange Multipliers

This is an all too familiar problem

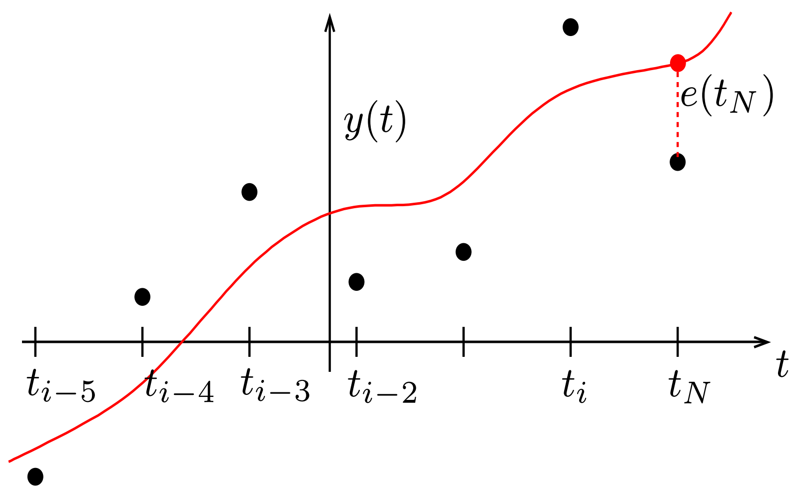


the statisticians know this as a linear regression problem for engineers this is curve fitting i.e. to find a function that 'best' fits to the data i.e. find the values of m and c to minimize $J = \sum_{i=1}^{N} e^2(t_i)$. This problem can be reformulated in terms of linear algebra as

$$\begin{bmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_N & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y2 \\ \vdots \\ y_N \end{bmatrix}$$

and in general form as polynomial

$$y(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_m t^m$$
$$e(t_i) = y(t_i) - y_i$$

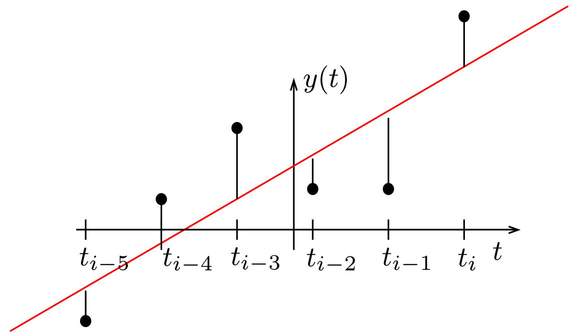


$$\begin{bmatrix} e(t_1) \\ e(t_2) \\ \cdots \\ e(t_N) \end{bmatrix} = \begin{bmatrix} 1 & t_1 & \cdots & t_1^m \\ 1 & t_1 & \cdots & t_1^m \\ \vdots & \vdots & \cdots & \vdots \\ 1 & t_N & \cdots & t_N^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} - \begin{bmatrix} y_1 \\ y2 \\ \cdots \\ y_N \end{bmatrix}$$

Traditional Least Squares

$$\mathbf{Ax} = \mathbf{b} + \mathbf{e}$$

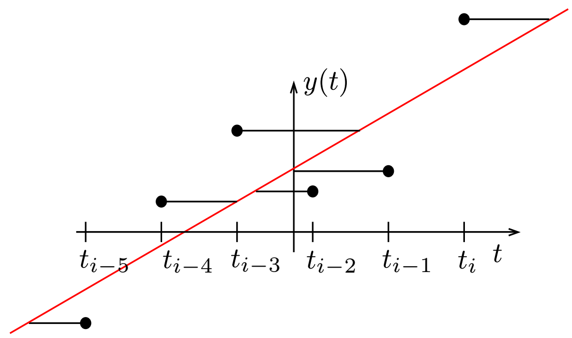
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$



Choose the $\mathbf{x}_{LS}$ that satisfies $(\mathbf{A})\mathbf{x} = \mathbf{b} + \mathbf{e}$ but requires the smallest $\|\mathbf{e}\|_2^2 = \sum_{i=1}^{m} e_i^2$.

$$\mathbf{Ax} + \mathbf{E} = \mathbf{b}$$

$$\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} + \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

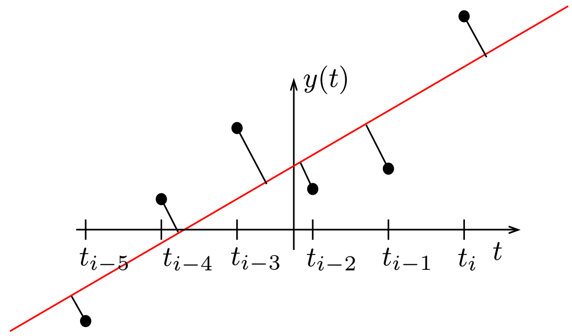


Choose the $\mathbf{x}_{DLS}$ that satisfies $(\mathbf{A} + \mathbf{E})\mathbf{x} = \mathbf{b}$ but requires the smallest $\|\mathbf{E}\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} e_{ij}^2$.

Total Least Squares

$$\mathbf{A}\mathbf{x} + \mathbf{E} = \mathbf{b} + \mathbf{e}$$

$$\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} + \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ e_{31} & e_{32} \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$



Choose the $\mathbf{x}_{TLS}$ that satisfies $(\mathbf{A} + \mathbf{E})\mathbf{x} = \mathbf{b}$ but requires the smallest $\|\mathbf{E}\|_F^2 + \|\mathbf{e}\|_2^2$.

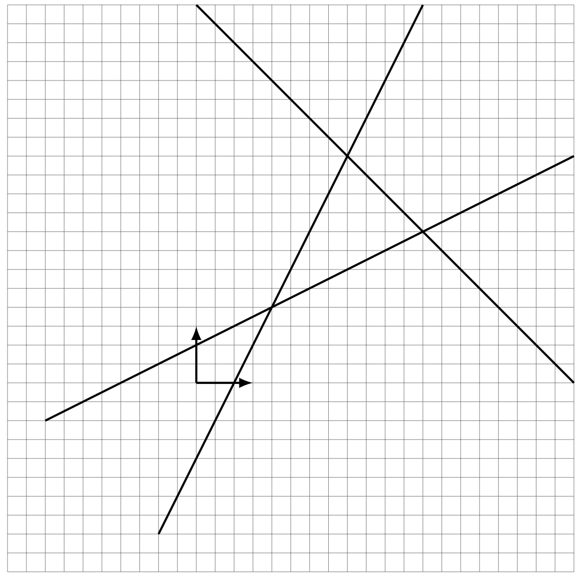
2.1 Least Square Problem

In practice it often happens that solution to a set of linear equations is not possible. The usual reason for this problem is too many equations. There are more equations than unknowns, unless readings are perfect, $\mathbf{b}$ is outside the column space of $\mathbf{A}$. This is typically due to noisy measurements. i.e.

$$\mathbf{A}\mathbf{x} = \mathbf{b} + \mathbf{e}$$

Lets consider an example

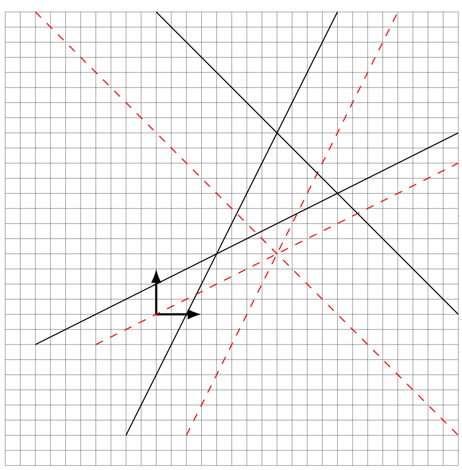
$$y = 2x - 2$$
$$y = 0.5x + 1$$
$$y = -x + 10$$

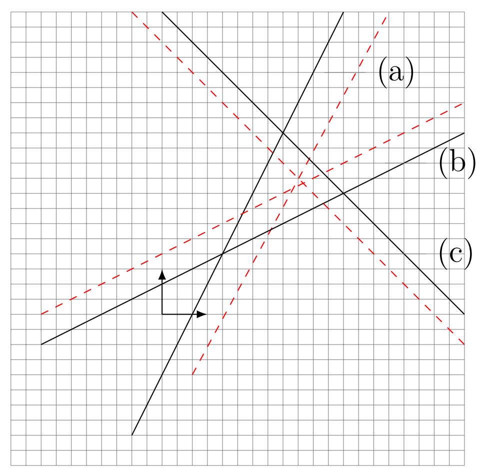


The plot clearly illustrates that there is no unique solution to this problem

$$y = 2x - 2 + e_1$$
$$y = 0.5x + 1 + e_2$$
$$y = -x - 10 + e_3$$

Add e_1, e_2 and e_3 such that there exist an (x, y) that satisfies all three equations. There are infinitely many possibilities but optimal solution is such that $\sum_{i=1}^{3} e_i^2$ is minimized.



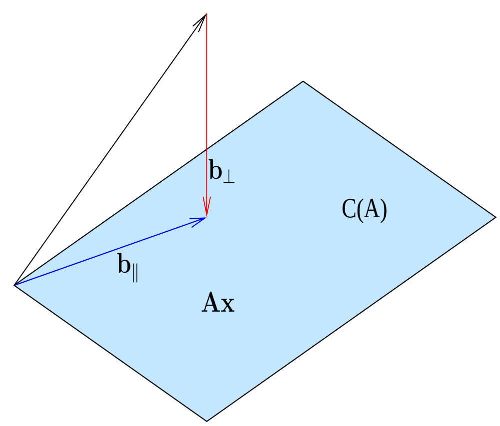


The cost function looks like

$$J = \sum_{i=1}^{3} e_i^2 = (y - 2x + 2)^2 + (y - 0.5x - 1)^2 + (y + x - 10)^2$$
$$\frac{\partial J}{\partial x} = 0 \Rightarrow -1.5y + 5.25x = 13.5$$
$$\frac{\partial J}{\partial y} = 0 \Rightarrow 3y - 1.5x = 9$$

Solving we get the LEAST-SQUARE solutions as $(x, y)_{\text{LS}} = (4, 5)$.

$$J(\mathbf{x}) = \|\mathbf{e}\|^2 = (\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x}) = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{A}^T\mathbf{b} - \mathbf{b}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{b}$$
$$\Rightarrow \frac{\partial J(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{b} + \mathbf{0} = \mathbf{0}$$
$$\Rightarrow \frac{\partial J(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{A}^T\mathbf{b} + \mathbf{0} = \mathbf{0}$$
$$\Rightarrow \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$
$$\Rightarrow \mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$$



$$\mathbf{A}\mathbf{x} = \mathbf{b} = \mathbf{b}_{\parallel} + \mathbf{b}_{\perp}$$

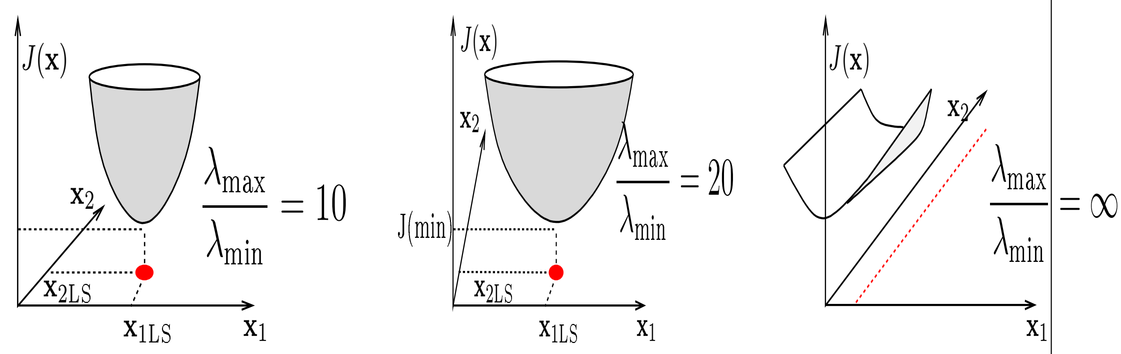
We further note that

$$\mathbf{A}^T\mathbf{b}_{\perp} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} b_{1\perp} \\ \vdots \\ b_{n\perp} \end{bmatrix} = \begin{bmatrix} < \mathbf{a}_1 \cdot b_{\perp} > \\ \vdots \\ < \mathbf{a}_n \cdot b_{\perp} > \end{bmatrix} = \mathbf{0}$$

Thus

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T(\mathbf{b}_{\|} + \mathbf{b}_{\perp}) = \mathbf{A}^T\mathbf{b}$$
$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

Properties of Quadratic Performance Surface

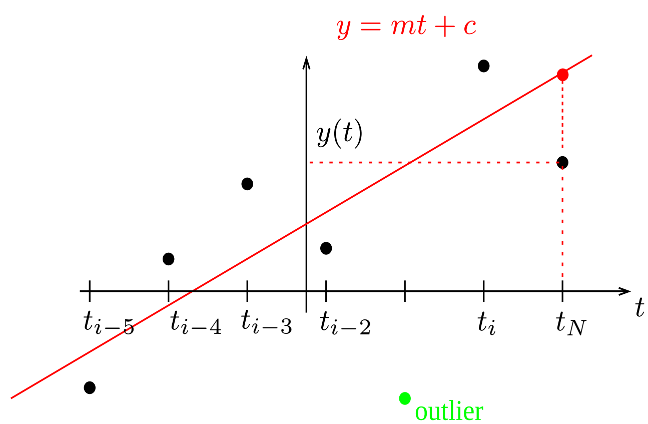


$$J(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{b}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{b}$$

Now consider the eigen-values $\{\lambda_i\}_{i=1}^n : \left(\mathbf{A}\mathbf{A}^T\right)\mathbf{v}_i = \lambda_i\mathbf{v}_i$ with $i = 1, 2, \cdots, n$.

2.2 Weighted and Constrained Least Squares

It is quite possible that some readings will be in error (called outliers).



$$e(t_i) = y(t_i) - y_i \qquad i = 1, 2, \ldots, N$$

Weighting each error term

$$J = \mathbf{e}^T \begin{bmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_n \end{bmatrix} \mathbf{e} = \sum_{i=1}^{n} w_i e^2(t_1)$$

Choice of weights could depend on the application, a typical choice could be $\sigma_i^2 = E\{e^2(t_i)\}$. Assuming errors are zero mean with variance $\sigma_i^2 = E\{e^2(t_i)\}$ so intuitively $J = \sum_{i=1}^{n} w_i e^2(t_i) = \sum_{i=1}^{N} \Big(\frac{1}{\sigma_i^2}\Big) e^2(t_i)$.

$$\mathbf{Ax} = \mathbf{b} - \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = J = \mathbf{e}^T \begin{bmatrix} \frac{1}{\sigma_1^2} & & & \\ & \frac{1}{\sigma_2^2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_n^2} \end{bmatrix} \mathbf{e} = \sum_{i=1}^{n} \Big(\frac{1}{\sigma_i^2}\Big) e^2(t_i)$$

this implies

$$\mathbf{x}_{\text{WLS}} = \arg\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_W^2 = \left(\mathbf{A}^T \mathbf{W} \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{W} \mathbf{b}$$

2.3 Constrained Least Squares Problem

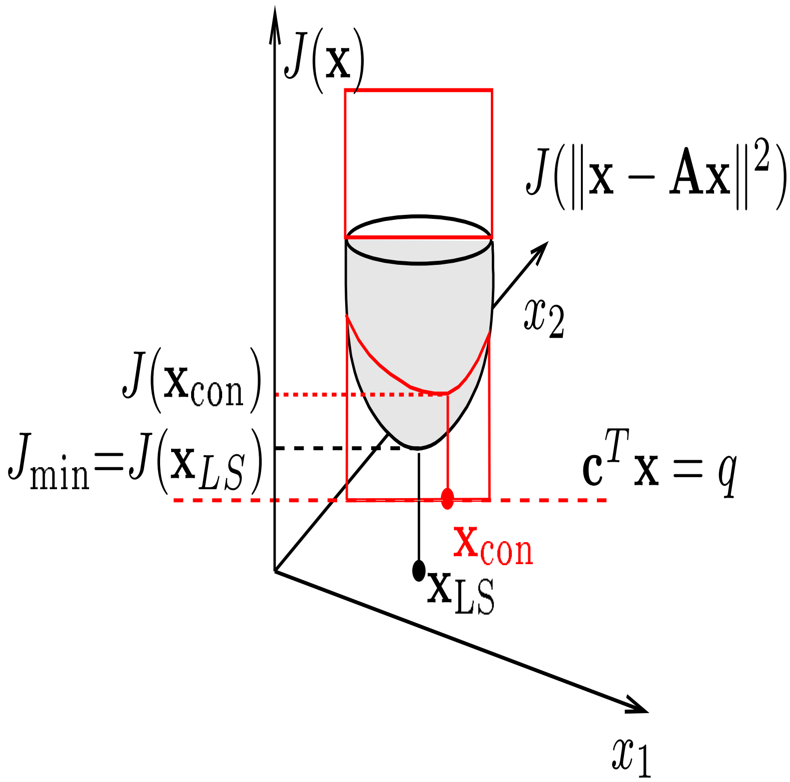
The cases often spring up in practice when solution to a set of equations must satisfy certain constraints for example the profit or revenue can not be negative, cut-off frequency of a filter should be less than a specified value. The constraints can be classified into two types generally:

- Linear Constraints.
- Quadratic Constraints.

$$\mathbf{x}_{\text{con}} = \arg\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|^2 \quad \text{subject to} \quad \mathbf{c}^T \mathbf{x} = q$$

$$\mathbf{x}_{\text{con}} = \arg\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|^2 \quad \text{subject to} \quad \mathbf{x}^T \mathbf{C} \mathbf{x} = q$$

Linear Constraints



The cost function

$$J(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 = \|\mathbf{A}(\mathbf{x} - \mathbf{x}_{\mathrm{LS}})\|^2 + \mathbf{b}^T\mathbf{P}_A^{\perp}\mathbf{b}$$

Doing a derivative in the presence of constraints

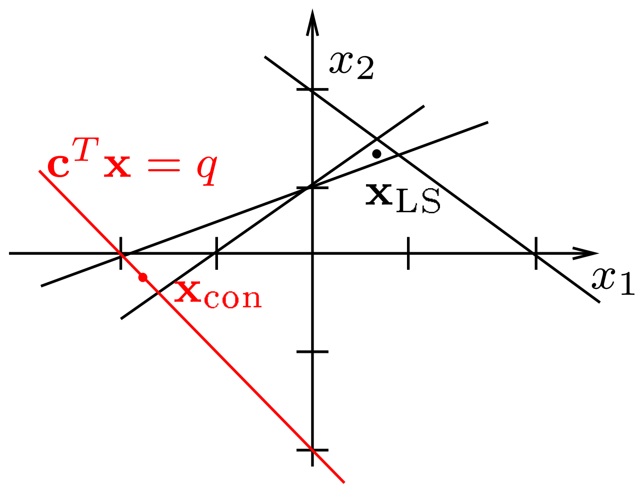
$$\begin{aligned}
J(\mathbf{x}) &= \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}_{\mathrm{LS}} + \mathbf{x}_{\mathrm{LS}}^T\mathbf{A}^T\mathbf{A}\mathbf{x}_{\mathrm{LS}} + \mathbf{b}^T\mathbf{P}_A^{\perp}\mathbf{b} + \lambda(\mathbf{c}^T\mathbf{x} - q) \\
\frac{\partial J}{\partial \mathbf{x}} &= 2\mathbf{A}^T\mathbf{A}\mathbf{x} - 2\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}_{\mathrm{LS}} + \lambda\mathbf{c} = \mathbf{0} \\
\mathbf{x} &= \mathbf{x}_{\mathrm{LS}} - 0.5\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\lambda\mathbf{c} \\
\frac{\partial J}{\partial \lambda} &= \mathbf{c}^T\mathbf{x} - q = \mathbf{0} \\
\lambda &= \frac{2\mathbf{c}^T\mathbf{x}_{\mathrm{LS}} - 2q}{\mathbf{c}^T\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{c}} \\
\mathbf{x}_{\mathrm{con}} &= \mathbf{x}_{\mathrm{LS}} - \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{c}\left[\frac{\mathbf{c}^T\mathbf{x}_{\mathrm{LS}} - q}{\mathbf{c}^T\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{c}}\right]
\end{aligned}$$

where $\mathbf{x}_{\mathrm{LS}} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$ and $\mathbf{P}_A^{\perp} = \mathbf{I} - \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$.

Example

Consider a linear system

$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ -0.5 & 1 \\ 1 & 1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, q = -2$. The system $\mathbf{Ax} = \mathbf{b}$ with 'fuzzy constraints' $\mathbf{c}^T \Rightarrow \begin{bmatrix} -1 & 1 \\ -0.5 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ -2 \end{bmatrix}$.



$$\mathbf{x}_{\text{wls}} = \arg\min_{\mathbf{x}} \|\tilde{\mathbf{b}} - \tilde{\mathbf{A}}\mathbf{x}\|_w^2 = \left(\tilde{\mathbf{A}}^T\mathbf{W}\tilde{\mathbf{A}}\right)^{-1} \tilde{\mathbf{A}}^T\mathbf{W}\tilde{\mathbf{b}}$$

where $\mathbf{W} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & w \end{bmatrix}$.

$$\mathbf{x}_{wls} = \arg\min_{\mathbf{x}} \|\tilde{b} - \tilde{\mathbf{A}}\mathbf{x}\|_w^2 = \left(\tilde{\mathbf{A}}^T\mathbf{W}\tilde{\mathbf{A}}\right)^{-1} \tilde{\mathbf{A}}^T\mathbf{W}\tilde{\mathbf{b}}, \mathbf{W} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & w \end{bmatrix}$$

$w = 0 \quad \mathbf{x}_{WLS} = [0.5358 \quad 1.4231]^T$

$w = 1 \quad \mathbf{x}_{WLS} = [-0.5358 \quad 0.5686]^T$

$w = 2 \quad \mathbf{x}_{WLS} = [-1.2222 \quad 0.0397]^T$

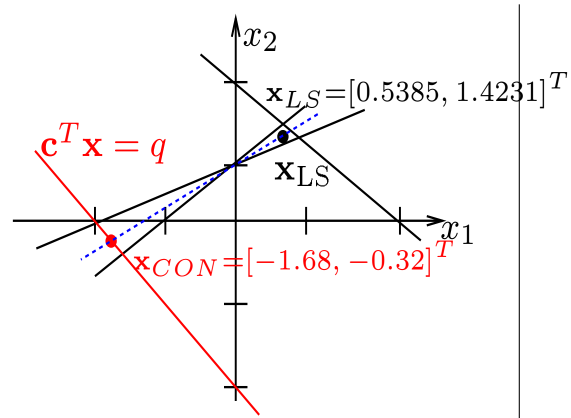
$w = 3 \quad \mathbf{x}_{WLS} = [-1.4502 \quad -0.1394]^T$

$w = 5 \quad \mathbf{x}_{WLS} = [-1.5914 \quad -0.2054]^T$

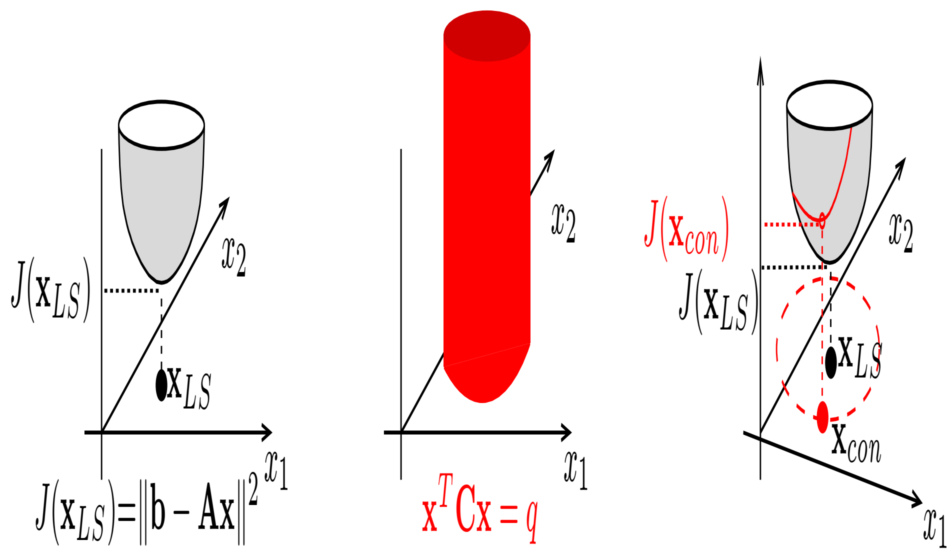
$w = 10 \quad \mathbf{x}_{WLS} = [-1.6572 \quad -0.3021]^T$

$w = 100 \quad \mathbf{x}_{WLS} = [-1.6798 \quad -0.3198]^T$

$w = 1000 \quad \mathbf{x}_{WLS} = [-1.6800 \quad -0.3200]^T$



Quadratic Constraints



$$J_1(\mathbf{x}) = \mathbf{x}^T\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{x} - \lambda\underbrace{\left(\mathbf{x}^T\mathbf{C}\mathbf{x} - q\right)}_{\text{Constraint}}$$

$$\Rightarrow \frac{\partial J_1(\mathbf{x})}{\partial \mathbf{x}} = 2\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{x} - 2\lambda\mathbf{C}\mathbf{x} = \mathbf{0}$$

$$\Rightarrow \left(\mathbf{A}^T\mathbf{A}\right)\mathbf{x}_i = \lambda_i\mathbf{C}\mathbf{x}_i \qquad i = 1, 2, \cdots, n$$

So by choosing $\mathbf{x}_i$ and corresponding λ_i we are trying to minimize

$$J(\mathbf{x}_i) = \mathbf{x}_i^T\underbrace{\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{x}_i}_{\lambda_i\mathbf{C}\mathbf{x}_i} = \lambda_i\mathbf{x}_i^T\mathbf{C}\mathbf{x}_i = \lambda_i q \quad \left(\text{since } \mathbf{x}^T\mathbf{C}\mathbf{x} = q\right) \tag{2.1}$$

$$\mathbf{x}_{\text{con}} = \mathbf{x}_{\min} \quad \text{corresponding to} \quad \lambda_{\min} = \min_i \{\lambda_i\}_{i=1}^n \Rightarrow J(\mathbf{x}_{\text{con}}) = q\lambda_{\min}$$

Deterministic Least Squares

For $\mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{e}$ choose $\hat{\mathbf{x}}$ to minimize $J(\mathbf{x}) = \|\mathbf{e}\|^2 = \sum_{k=1}^{n} |e_k|^2$

Stochastic Least Mean Squares

For $\mathbf{y} = \mathbf{A}\mathbf{x}$ choose $\hat{\mathbf{x}}$ to minimize

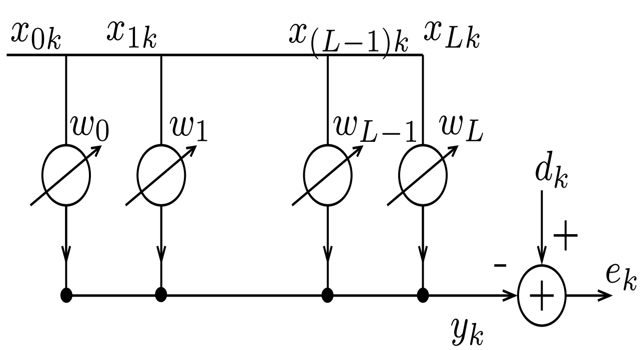
$$E\Big\{(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\Big\}$$

Quadratic Objective Functions

Deterministic Least Squares

$$\begin{aligned} e_k &= d_k - y_k \\ y_k &= \mathbf{x}_k^T \mathbf{w} = \mathbf{W}^T \mathbf{x}_k \\ |e_k|^2 &= d_k^2 + \mathbf{w}^T \mathbf{x}_k \mathbf{x}_k^T \mathbf{w} - 2 d_k \mathbf{x}_k^T \mathbf{w} \end{aligned}$$

Stochastic Least Mean Squares



$$\begin{aligned} J(\mathbf{w}) = E\Big\{|e_k|^2\Big\} &= E\Big\{d_k^2\Big\} + \mathbf{w}^T \underbrace{E\Big\{\mathbf{x}_k^T \mathbf{x}_k\Big\}}_{\mathbf{R}_{xx}} \mathbf{w} - 2 \underbrace{E\Big\{d_k \mathbf{x}_k^T\Big\}}_{\mathbf{P}_{\mathbf{dx}}} \\ &= E\Big\{d_k^2\Big\} + \mathbf{w}^T \mathbf{R}_{xx} \mathbf{w} - 2\mathbf{P}_{dx}^T \mathbf{w} \end{aligned}$$

Where

$$\mathbf{R}_{xx} = E\{\mathbf{x}_k\mathbf{x}_k^T\} = E\left\{\begin{bmatrix} x_{0k}^2 & x_{0k}x_{1k} & \cdots & x_{0k}x_{Lk} \\ x_{1k}x_{0k} & x_{1k}^2 & \cdots & x_{1k}x_{Lk} \\ \vdots & \vdots & \cdots & \\ x_{Lk}x_{0k} & x_{Lk}x_{1k} & \cdots & x_{Lk}^2 \end{bmatrix}\right\}$$

$$\mathbf{P}_{dx} = E\{d_k\mathbf{x}_k\} = E\left\{\begin{bmatrix} d_kx_{0k} \\ d_kx_{1k} \\ \vdots \\ d_kx_{Lk} \end{bmatrix}\right\}$$

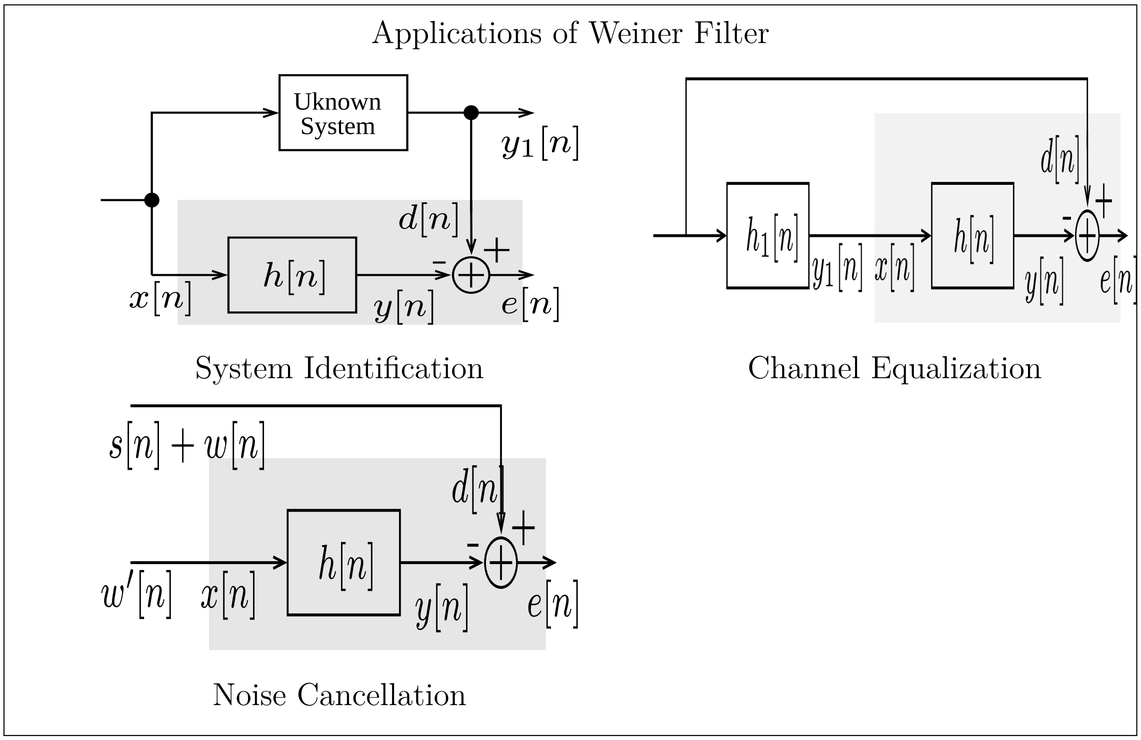
where $E\{x_{ik}x_{jk}\} = \dfrac{1}{N}\sum_{k=1}^{N} x_{ik}x_{jk}$ and $E\{d_kx_{jk}\} = \dfrac{1}{N}\sum_{k=1}^{N} d_kx_{jk}$.

$$J(\mathbf{w}) = E\{d_k^2\} + \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} - 2\mathbf{P}_{dx}^T\mathbf{w}$$

The optimal coefficients can be obtained through derivation of the cost function

$$\frac{\partial J}{\partial \mathbf{w}} = \begin{bmatrix} \dfrac{\partial J}{\partial w_0} \\ \dfrac{\partial J}{\partial w_1} \\ \vdots \\ \dfrac{\partial J}{\partial w_L} \end{bmatrix} = 2\mathbf{R}_{xx}\mathbf{w} - 2\mathbf{P}_{dx} = \mathbf{0}$$

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{P}_{dx}$$

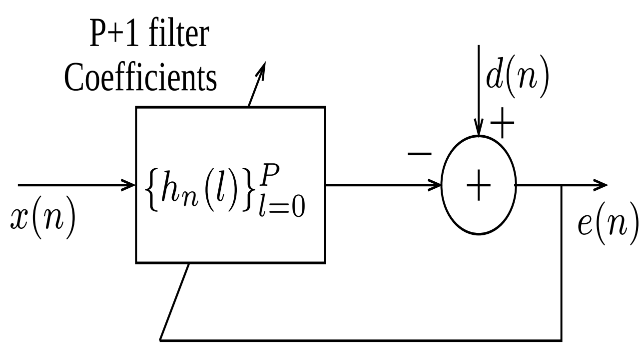


The problems with Least Mean Square (Wiener Solution)

- The estimates of $\mathbf{R}_{xx}$ and $\mathbf{P}_{dx}$.
- The statistical properties may change from time to time.
- Recalculation is computationally complex.

The most straight forward solution to this problem is the Least Mean Square Algorithm.

2.4 Least Mean Square Algorithm



- for $n = 0, 1, 2, \cdots$ evaluate $\Big\{h_n(l)\Big\}_{l=0}^{P}$. Compute

$$
\begin{aligned}
y(n) &= \sum_{l=0}^{P} h_n(l)x(n-l), \quad n = 0, 1, 2, \cdots \\
e(n) &= d(n) - y(n)
\end{aligned}
$$

- Choose $\Big\{h_n(l)\Big\}_{l=0}^{P}$ to minimize $J(n) = E\Big\{e^2(n)\Big\}$.

We defined

$$
\begin{aligned}
\mathbf{x}(n) &= \Big[x(n) \quad x(n-1) \quad \cdots \quad x(n-P)\Big]^T \\
\mathbf{h}(n) &= \Big[h_n(0) \quad h_n(1) \quad \cdots \quad h_n(P)\Big]^T \\
y(n) &= \mathbf{h}^T(n)\mathbf{x}(n) \\
J(n) &= E\Bigg\{e^2(n)\Bigg\}
\end{aligned}
$$

The stochastic least mean square solution of this problem would be

$$
\begin{aligned}
J(n) = E\Big\{e^2(n)\Big\} &= E\Bigg\{\Big(d(n) - \mathbf{h}^T\mathbf{x}\Big)^T\Big(d(n) - \mathbf{h}^T\mathbf{x}\Big)\Bigg\} \\
&= \sigma^2 - 2\mathbf{h}^T(n)\underbrace{E\Big\{d(n)x(n)\Big\}}_{p(n)} + \mathbf{h}^T(n)\underbrace{E\Big\{\mathbf{x}(n)\mathbf{x}^T(n)\Big\}}_{\mathbf{R}_{xx}(n)}\mathbf{h}(n) \\
&= \sigma_d^2 - 2\mathbf{h}^T(n)\mathbf{p}(n) + \mathbf{h}^T(n)\mathbf{R}_{xx}(n)\mathbf{h}(n)
\end{aligned}
$$

setting the derivative of this equation to ‘zero’ yields.

$$
\begin{aligned}
\frac{\partial J(n)}{\partial \mathbf{h}} &= 2\mathbf{R}_{xx}(n)\mathbf{h}(n) - 2\mathbf{p}(n) = 0 \quad \text{Wiener-Hopf Eqns} \\
\Rightarrow \mathbf{h}_{\text{opt}}(n) &= \mathbf{R}_{xx}^{-1}(n)\mathbf{p}(n)
\end{aligned}
$$

Computation of exact correlation matrix is hard to calculate and in some cases it may not be possible therefore

$$
\hat{\mathbf{h}}_{\text{opt}}(n) = \hat{\mathbf{R}}_{xx}^{-1}(n)\hat{\mathbf{p}}(n)
$$

The steepest descent avoids matrix inverses

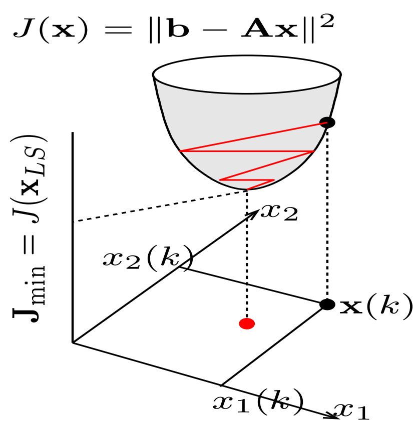
$$
\hat{\mathbf{h}}_{\text{opt}}(n+1) = \hat{\mathbf{h}}_{\text{opt}}(n) - \mu\frac{\partial J(n)}{\partial \mathbf{h}} = \hat{\mathbf{h}}_{\text{opt}}(n) - 2\mu\Big[\mathbf{R}_{xx}(n)\hat{\mathbf{h}}_{\text{opt}}(n) - \mathbf{p}(n)\Big]
$$

This expression still requires estimates of $\mathbf{R}_{xx}(n)$ and still computationally complex

$$\hat{\mathbf{h}}_{\text{opt}}(n+1) = \hat{\mathbf{h}}_{\text{opt}}(n) - \mu \frac{\partial J(n)}{\partial \mathbf{h}} = \hat{\mathbf{h}}_{\text{opt}}(n) - 2\mu \Big[\hat{\mathbf{R}}_{xx}(n) \hat{\mathbf{h}}_{\text{opt}}(n) - \hat{\mathbf{p}}(n) \Big]$$

The solution to this problem is LMS-stochastic gradient approach

Iterative Minimization / Gradient Descent



$$\mathbf{x}(k+1) = \mathbf{x}(k) - \mu \frac{\partial J(b)}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}(k)}$$

where

$$\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{A}^T \mathbf{A}\mathbf{x} - 2\mathbf{A}^T \mathbf{b} = 2\mathbf{A}^T \Big(\underbrace{\mathbf{A}\mathbf{x} - \mathbf{b}}_{\mathbf{e}} \Big)$$

For k=1 to N

$$\mathbf{e}(k) = \mathbf{b} - \mathbf{A}\mathbf{x}(k)$$

$$\mathbf{x}(k+1) = \mathbf{x}(k) + 2\mu \mathbf{A}^T \mathbf{e}(k)$$

next k

$$J(\mathbf{n}) = E\Big\{ e^2(n) \Big\} = \frac{\partial J(n)}{\partial \mathbf{h}} = 2E\Big\{ e(n) \frac{\partial e(n)}{\partial \mathbf{h}} \Big\}$$

$$= \frac{\partial J(n)}{\partial \mathbf{h}} \simeq 2e(n) \frac{\partial e(n)}{\partial \mathbf{h}} = \frac{\partial \hat{J}(n)}{\partial \mathbf{h}}$$

But $e(n) = d(n) - \mathbf{x}^T \mathbf{h} \Rightarrow \dfrac{\partial e(n)}{\partial} = -\mathbf{x}(n)$ and so

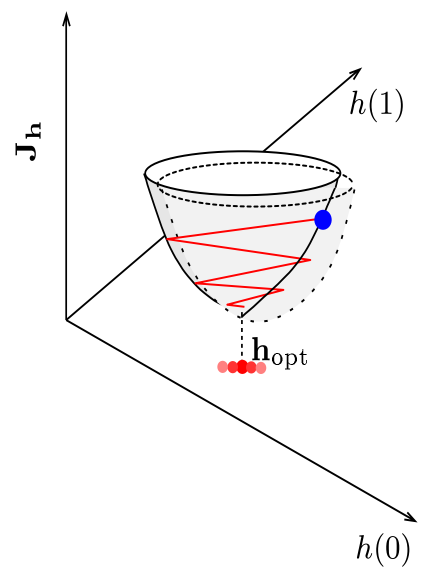
$\hat{\mathbf{h}}_{\text{opt}}(n+1) = \hat{\mathbf{h}}_{\text{opt}}(n) - \mu \dfrac{\partial \hat{J}(n)}{\partial \mathbf{h}} = \hat{\mathbf{h}}_{\text{opt}}(n) + 2\mu e(n) \mathbf{x}(n).$

LMS algorithm

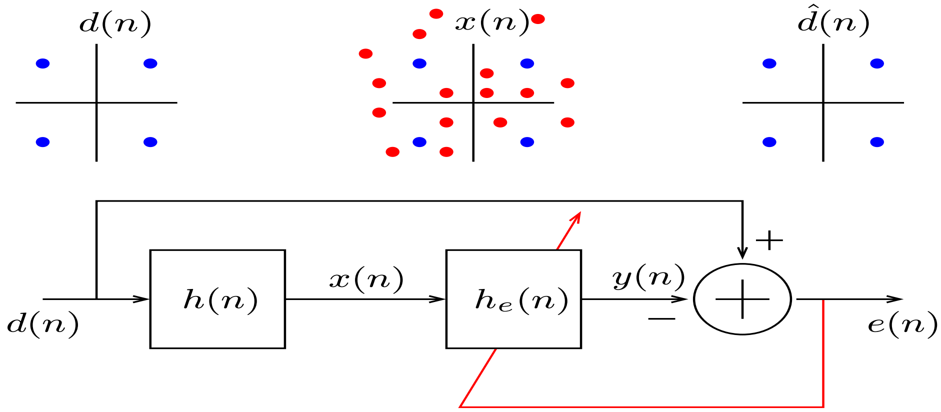
Given $\mathbf{x}[n] = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(P) \end{bmatrix}^T$
$\mathbf{h}[n] = \begin{bmatrix} h_n(0) & h_n(1) & \cdots & h_n(P) \end{bmatrix}^T$
The LMS algorithm comprises of

$$y[n] = \mathbf{h}^T[n]\mathbf{x}[n]$$
$$e[n] = d[n] - y[n]$$
$$\mathbf{h}[n+1] = \mathbf{h}[n] + 2\mu e[n]\mathbf{x}[n]$$

Although LMS is very simple to implement and fairly reliable however the convergence of algorithm primarily depends on step size coefficients μ.



For large values of μ algorithm converges rapidly whoever oscillates around optimal point never gets there. For small values of μ algorithm is slow to converge. The choice of inital guess also is important.



Implementation Complexity of LMS algorithm increases linearly with the number of coefficients to be calculated. Which makes it an ideal choice for low complexity solutions.
Some variants of LMS algorithm are

$$\mathbf{h}(n+1) = \mathbf{h}(n) + 2\mu e(n)\mathbf{x}(n) \qquad \text{Linear}$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + 2\mu e(n)\mathrm{sgn}\Big[\mathbf{x}(n)\Big] \qquad \text{Clipped}$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + 2\mu \mathrm{sgn}\Big[e(n)\Big]\mathbf{x}(n) \qquad \text{Pilot}$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + 2\mu \mathrm{sgn}\Big[e(n)\Big]\mathrm{sgn}\Big[\mathbf{x}(n)\Big] \qquad \text{Zero Forcing}$$

Increase hardware efficiency at the expense of performance

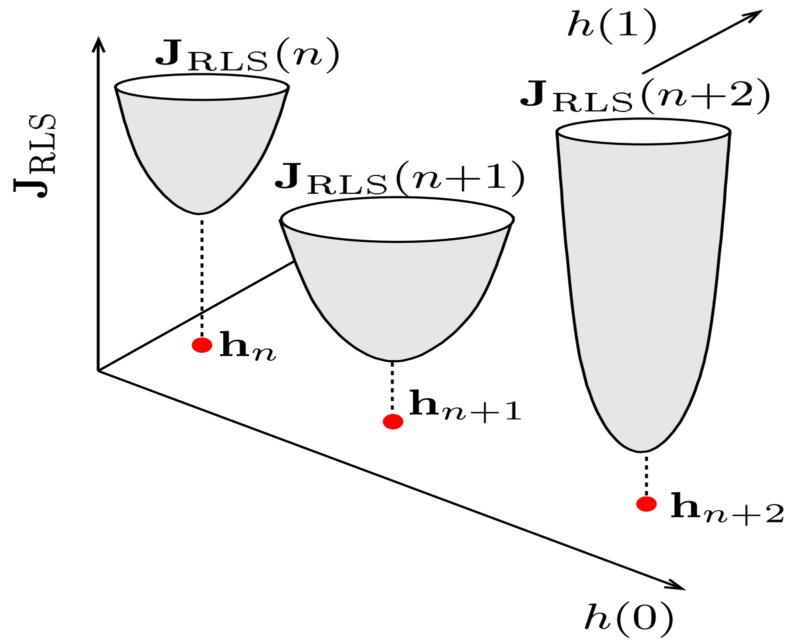
2.5 Recursive Least Squares

Now we consider the time-varying scenario

$$\mathbf{A}(n)\mathbf{x} = \mathbf{b}(n)$$

The least square solution to this problem would be

$$\mathbf{x}_{\mathrm{LS}}(n) = \left(\mathbf{A}^T(n)\mathbf{A}(n)\right)^{-1}\mathbf{A}^T(n)\mathbf{b}(n)$$



Repeated inversion of matrix is not feasible solution. Since $\mathbf{x}_{\mathrm{LS}}(n)$ may be related to $\mathbf{x}_{\mathrm{LS}}(n-1)$ depending on how $\mathbf{A}(n)$ and $\mathbf{b}(n)$ are related to $\mathbf{A}(n-1)$ and $\mathbf{b}(n-1)$. The principle of RLS algorithm to update the filter coefficients is outline as follows For each $n = 0, 1, 2, \cdots, \infty$

i. Calculate $y_n[k] = \sum_{l=0}^{P} h_n[l]x[k-l]$ with $k = 0, 1, 2, \cdots, n$.

ii. Calculate $e_n[k] = d[k] - y_n[k]$ with $k = 0, 1, 2, \cdots, n$.

iii. Choose: $\Big\{h_n(l)\Big\}_{l=0}^{P}$ to minimize $J_{\mathrm{RLS}} = \sum_{k=0}^{n} \lambda^{n-k} e_n^2(k)$ with forgetting factor $0 \le \lambda \le 1$.

Kicking up the cost function bit further

$$\frac{\partial J_{\mathrm{RLS}}(n)}{\partial h_n(r)} = \sum_{k=0}^{n} \lambda^{n-k} e_n^2(k) = \sum_{k=0}^{n} \lambda^{n-k} \Big(\underbrace{d(k) - y_n(k)}_{e_n(k)}\Big)^2$$

$$\begin{aligned}
\frac{\partial J_{\mathrm{RLS}}(n)}{\partial h_n(r)} &= \sum_{k=0}^{n} \lambda^{n-k} 2e_n(k) \frac{\partial e_n(k)}{\partial h_n(r)} = \sum_{k=0}^{n} \lambda^{n-k} 2e_n(k)(-x(k-r)) = 0 \\
&= -\sum_{k=0}^{n} \lambda^{n-k} \Big[d(k) - \sum_{l=0}^{P} h_n(l)x(k-l)\Big] x(k-r) = 0 \\
&= \sum_{l=0}^{P} h_n(l) \Big[\sum_{k=0}^{n} \lambda^{n-k} x(k-l)x(k-r)\Big] - \sum_{k=0}^{n} \lambda^{n-k} d(k)x(k-r) = 0 \quad r = 0, 1, \cdots, P
\end{aligned}$$

The above expression can be expressed in matrix form as

$$\begin{bmatrix} \sum_{k=0}^{n} \lambda^{n-k} x^2(k) & \sum_{k=0}^{n} \lambda^{n-k} x(k-1)x(k) & \cdots & \sum_{k=0}^{n} \lambda^{n-k} x(k-P)x(k) \\ \sum_{k=0}^{n} \lambda^{n-k} x(k)x(k-1) & \sum_{k=0}^{n} \lambda^{n-k} x^2(k-1) & \cdots & \sum_{k=0}^{n} \lambda^{n-k} x(k-P)x(k-1) \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{k=0}^{n} \lambda^{n-k} x(k)x(k-P) & \sum_{k=0}^{n} \lambda^{n-k} x(k-1)x(k-P) & \cdots & \sum_{k=0}^{n} \lambda^{n-k} x^2(k-P) \end{bmatrix} \begin{bmatrix} h_n(0) \\ h_n(1) \\ \vdots \\ h_n(P) \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{n} \lambda^{n-k} d(k)x(k) \\ \sum_{k=0}^{n} \lambda^{n-k} d(k)x(k-1) \\ \vdots \\ \sum_{k=0}^{n} \lambda^{n-k} d(k)x(k-P) \end{bmatrix}$$

In short hand notation this adds up to

$$\mathbf{R}_{xx}^{\lambda}(n)\mathbf{h}_n = \mathbf{r}_{xd}^{\lambda}(n)$$
$$\mathbf{h}_n = \left(\mathbf{R}_{xx}^{\lambda}(n)\right)^{-1} \mathbf{r}_{xd}^{\lambda}(n) \qquad n = 0, 1, \cdots, \infty$$

so it is possible to calculate $\mathbf{h}_n$ from $\mathbf{h}_{n-1}$ without having to calculate $\left(\mathbf{R}_{xx}^{\lambda}(n)\right)^{-1}$.

$$\mathbf{h}_n = \left(\mathbf{R}_{xx}^{\lambda}(n)\right)^{-1} \mathbf{r}_{xd}^{\lambda}(n)$$

Define $\mathbf{x}(k) = \Big[x(k), x(k-1), \cdots, x(k-P)\Big]^T$ then

$$\begin{aligned} \mathbf{R}_{xx}^{\lambda}(n) &= \sum_{k=0}^{n} \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^T(k) \\ \mathbf{r}_{xd}^{\lambda}(n) &= \sum_{k=0}^{n} \lambda^{n-k} \mathbf{d}(k)\mathbf{x}(k) \\ \Rightarrow \mathbf{R}_{xx}^{\lambda}(n) &= \lambda \mathbf{R}_{xx}^{\lambda}(n-1) + \mathbf{x}(k)\mathbf{x}^T(k) \\ \mathbf{r}_{xd}^{\lambda}(n) &= \mathbf{r}_{xd}^{\lambda}(n-1) + \mathbf{d}(k)\mathbf{x}(k) \end{aligned}$$

So how do we evaluate $\left(\mathbf{R}_{xx}^{\lambda}(n)\right)^{-1}$ from $\left(\mathbf{R}_{xx}^{\lambda}(n-1)\right)^{-1}$?

Woodbury's identity

$$\left(\mathbf{A} + \mathbf{u}\mathbf{u}^T\right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{u}^T\mathbf{A}^{-1}}{1 + \mathbf{u}^T\mathbf{A}^{-1}\mathbf{u}}$$

using this identity to our problem we have

$$\left(\mathbf{R}_{xx}^{\lambda}(n)\right)^{-1} = \lambda^{-1}\left(\mathbf{R}_{xx}^{\lambda}(n-1)\right)^{-1}$$
$$-\underbrace{\left[1+\lambda^{-1}\mathbf{x}^T(n)\left(\mathbf{R}_{xx}^{\lambda}(n-1)\right)^{-1}\mathbf{x}(n)\right]^{-1}}_{\text{scalar}}\lambda^{-2}\left(\mathbf{R}_{xx}^{\lambda}(n-1)\right)^{-1}\mathbf{x}(n)\mathbf{x}^T(n)\left(\mathbf{R}_{xx}^{\lambda}(n-1)\right)^{-1}$$

RLS Summary

Initialize:

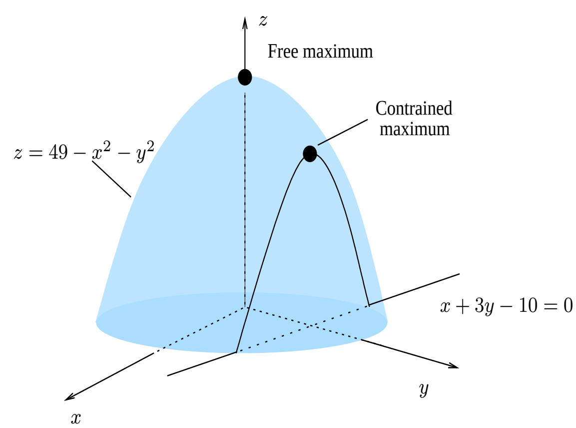
$$\mathbf{h}_{-1}^T = \mathbf{0}$$
$$\mathbf{P}(-1) = \delta^{-1}\mathbf{I}_{P+1\times P+1} \quad \text{where} \quad \mathbf{P}(n) = \mathbf{R}_{xx}^{-1}(n)$$

Computation: For $n = 0, 1, 2, \cdots$

$$\mathbf{x}(n) = \Big[x(n), x(n-1), \cdots, x(n-P)\Big]^T$$
$$\mathbf{h}_n = \Big[h_n(0), h_n(1), \cdots, h_n(P)\Big]^T$$
$$\epsilon(n) = d(n) - \mathbf{h}_{n-1}^T\mathbf{x}(n)$$
$$\mathbf{g}(n) = \mathbf{P}(n-1)\mathbf{x}(n)\Big\{\lambda + \mathbf{x}^T\mathbf{P}(n-1)\mathbf{x}(n)\Big\}^{-1}$$
$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)$$
$$\mathbf{h}_n = \mathbf{h}_{n-1} + \epsilon(n)(n)$$

2.6 Langrange Multipliers Method

This technique was developed by *Joseph-Louis Lagrange* a French mathematician to solve min-max problems in geometry.



For any general multi-variate function $f(\mathbf{x})$ we know that the maxima and the minima of a general differentiable function occurs where the derivative of the function in all directions is zero.

$$\nabla f(x) = 0 \tag{2.2}$$

where ∇ is the gradient operator. $\nabla f(x)$ is the direction in which rate of increase is maximum. There may be a problem that $\nabla f(x)$ is zero and we may still not be at global maximum or minimum this happens due to a possible local minima or maxima.

The method of Lagrange multiplier can restrict the search of solution in the feasible set of values of $\mathbf{x}$. The problem is typically formulated as

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} f(\mathbf{x}) \\ \text{subject to} \quad g_i(\mathbf{x}) &= 0 \qquad\qquad \forall i = 1, \cdots, m \end{aligned}$$

In English, find solution that minimizes $f(\mathbf{x})$, as long as all equalities $g_i(\mathbf{x}) = 0$ hold.

The Lagrange Multiplier method works by putting the cost as well as the constraints in a single minimization problem, but multiply each constraint by λ_i.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \quad L(\mathbf{x}, \lambda) = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$$

The optimal value can be found through

$$\nabla L(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) + \sum_i \lambda_i \nabla g_i(\mathbf{x}) = 0$$

and

$$\frac{\partial}{\partial \lambda_i} L(\mathbf{x}, \lambda) = g_i(\mathbf{x}) = 0$$

For the case if we have n variables and m constraints we have $n + m$ equations and same number of unknowns.

The constraints are not merely limited to equalities on the contrary they can be inequalities. The Karush-Kuhn-Tucker (KKT) conditions extend the method of Lagrange multipliers to allow inequalities and KKT conditions are necessary for optimality.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x})$$
$$\text{subject to} \quad g_i(\mathbf{x}) = 0 \qquad \forall i = 1, \cdots, m$$
$$\text{subject to} \quad h_i(\mathbf{x}) \leq 0 \qquad \forall i = 1, \cdots, n$$

In english, find the solution that minimizes $f(\mathbf{x})$ as long as all equalities $g_i(\mathbf{x})$ and all the inequalities $h_i(\mathbf{x}) \leq 0$ hold.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \quad L(\mathbf{x}, \lambda, \mu) = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{i=1}^{n} \mu_i h_i(\mathbf{x})$$

where $L(\mathbf{x}, \lambda, \mu)$ is the Lagrangian and depends on λ, μ which are the vectors of multipliers.

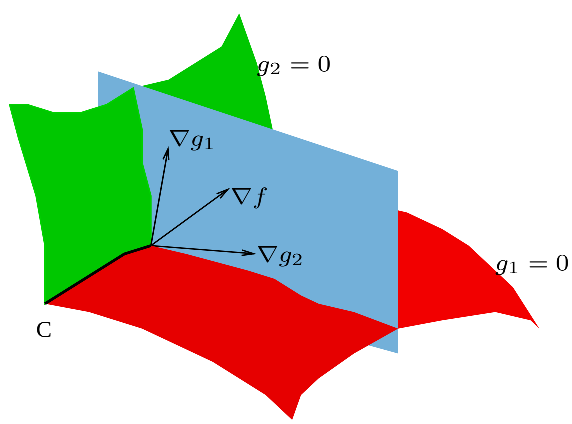
One need to be mindful of the fact that in some cases minimums and maximums won't exist even through the method will seem to imply they do. Every solution should be examined.

Suppose that $f(x, y, z)$ and $g(x, y, z)$ are differentiable. To find the local maximum and minimum value of $f(\cdot)$ subject to constraint $g(x, y, z) = 0$. Find the values of (x, y, z) and λ that simultaneously satisfy the equations

$$\nabla f = \lambda \nabla g \qquad \text{and} \qquad g(x, y, z) = 0 \tag{2.3}$$

In case of multiple constraints for example $g_1(x, y, z)$ and $g_2(x, y, z)$ when $g_1(\cdot)$ and $g_2(\cdot)$ are both differentiable with ∇g_1 not parallel to ∇g_2 then

$$\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 \qquad \text{and} \qquad g_1(x, y, z) = 0 \qquad g_2(x, y, z) = 0 \tag{2.4}$$



Now (2.4) can be interpreted as follows the surface $g_1 = 0$ and $g_2 = 0$ usually intersect a smooth curve C and along this curve we seek points where $f()$ has local maximum and minimum values relative to other values on the curve. These are the points where ∇f is normal to C and ∇g_1 and ∇g_2 are also normal to C at these points because C lies in the surfaces $g_1(x,y,z) = 0$ and $g_2(x,y,z) = 0$ which is a requirement in (2.4)

Example

Find the dimension of the box with largest volume is total surface area is 64cm^2. We want to find the greatest volume so the function that we want to optimize is given by

$$f(x,y,z) = xyz$$

Next we know that the surface area of the box must be a constant 64. So this is the constraint. The surface area of a box is simply the sum of the area of each of the sides so the constraint is given by,

$$2xy + 2xz + 2yz = 64 \qquad \Rightarrow \quad xy + xz + yz = 32$$

The equation for $g(x,y,z)$ is thus

$$g(x,y,z) = xy + xz + yz$$

Here are the four equations that we need to solve

$$\begin{aligned} yz &= \lambda(y+z) & f_x &= \lambda g_x & &\text{(a)} \\ xz &= \lambda(x+z) & f_y &= \lambda g_y & &\text{(b)} \\ xy &= \lambda(x+y) & f_z &= \lambda g_z & &\text{(c)} \\ xy + xz + yz &= 32 & g(x,y,z) &= 32 & &(\dagger) \end{aligned}$$

There are many ways to solve this system. Multiplying (a),(b) and (c) by x,y and z respectively we have

$$\begin{aligned} xyz &= \lambda x(y+z) \quad \text{(d)} \\ xyz &= \lambda y(x+z) \quad \text{(e)} \\ xyz &= \lambda z(x+y) \quad \text{(f)} \end{aligned}$$

setting (d) and (e) equal gives

$$\begin{aligned} \lambda x(y+z) &= \lambda y(x+z) \\ \lambda(xy+xz) - \lambda(xy+yz) &= 0 \\ \lambda(xz - yz) &= 0 \\ \Rightarrow \qquad \lambda &= 0 \qquad \text{or} \qquad xz = yz \end{aligned}$$

We have two possibilities. The first $\lambda = 0$ is not possible since if this was true (a) would reduce to

$$yz = 0 \qquad \Rightarrow \quad y = 0 \quad \text{or} \quad z = 0$$

Since we are talking about dimension neither of these is possible so we can safely discount $\lambda = 0$. This leaves the second possibility

$$xz = yz \qquad (*)$$

since we know that $z \neq 0$ (as we are talking about the dimensions of a box) we can cancel the z from both sides to have

$$x = y$$

Likewise lets set (e) and (f) equal

$$\begin{aligned}
\lambda y(x + z) &= \lambda z(x + y) \\
\lambda(xy + yz) - \lambda(xz + yz) &= 0 \\
\lambda(xy - xz) &= 0 \\
&\Rightarrow \quad \lambda = 0 \quad \text{or} \quad xy = zx
\end{aligned}$$

We know that $\lambda = 0$ is not possible so this leaves

$$xy = zx \qquad \Rightarrow \quad y = z \qquad (**)$$

Plugging (*) and (**) in (†) we have

$$y^2 + y^2 + y^2 = 3y^2 = 32 \qquad \text{y}{=}\pm\sqrt{\frac{32}{3}} = \pm 3.266$$

Since y must be a positive number therefore the only solution that makes physical sense is

$$x = y = z = 3.266$$

so the box actually is a cube.

Example

Minimize the l_2-norm of the variable subjects to constraints

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \quad L(\mathbf{x}, \lambda) = \arg\min_{\mathbf{x}} \|\mathbf{x}\|^2 + \lambda^T(\mathbf{y} - \mathbf{A}\mathbf{x})$$

the number of Lagrange multipliers is equal to the number of elements in $\mathbf{y}$ and the gradiant with respect to $\mathbf{x}$ is

$$2\mathbf{x} - \mathbf{A}^T\lambda = 0$$

As $\mathbf{x}$ depends on λ. From gradient of the constraint function with respect to λ we get

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

Pre-multiply the first equation by $\mathbf{A}$ to get

$$\begin{aligned}
2\mathbf{A}\mathbf{x} &= \mathbf{A}\mathbf{A}^T\lambda \\
2\mathbf{y} &= \mathbf{A}\mathbf{A}^T\lambda \\
\lambda &= 2(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y} \\
\mathbf{x} &= \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y}
\end{aligned}$$

Example

The plane $x+y+z=1$ cuts the cylinder $x^2+y^2=1$ in an ellipse . Find the points on the ellipse that lie closest to and farthest from origin.
Solution: we find the extreme value of $f(x,y,x) = x^2+y^2+z^2$ the square of the distance from (x,y,z) to origin subject to constraints

$$g_1(x,y,z) = x^2+y^2-1 = 0 \tag{2.5}$$
$$g_2(x,y,z) = x+y+z-1 = 0 \tag{2.6}$$

according to (2.4) the gradient of $f(\cdot)$ and $g(\cdot)$

$$\begin{aligned}
2x\boldsymbol{i} + 2y\boldsymbol{j} + 2z\boldsymbol{k} &= \lambda_1(2x\boldsymbol{i} + 2y\boldsymbol{j}) + \lambda_2(\boldsymbol{i} + \boldsymbol{j} + \boldsymbol{k}) \\
2x\boldsymbol{i} + 2y\boldsymbol{j} + 2z\boldsymbol{k} &= (\lambda_1 2x + \lambda_2)\boldsymbol{i} + (2\lambda_1 y + \lambda_2)\boldsymbol{j} + \lambda_2\boldsymbol{k}
\end{aligned}$$

which implies simply that

$$\begin{aligned}
2x &= 2\lambda_1 x + \lambda_2 \\
2y &= 2\lambda_1 y + \lambda_2 \\
2z &= \lambda_2
\end{aligned}$$

$$
\begin{aligned}
2x = 2\lambda_1 x + 2z &\quad\Rightarrow\quad (1-\lambda_1)x = z \\
2y = 2\lambda_1 x + 2z &\quad\Rightarrow\quad (1-\lambda_1)y = z
\end{aligned}
$$

These equations are satisfied simultaneously if either $\lambda = 1$ and $z = 0$ or $\lambda \neq 1$ and $x = y = z/(1-\lambda)$.

If $z = 0$, then solving (2.5) and (2.6) simultaneously to find the corresponding points in the ellipse gives two points $(1,0,0)$ and $(0,1,0)$ which makes sense as in figure.

If $x = y$ then (2.5) and (2.6) give

$$
\begin{aligned}
x^2 + x^2 - 1 = 0 &\quad\Rightarrow\quad x + x + z - 1 = 0 \\
2x^2 = 1 &\quad\Rightarrow\quad z = 1 - 2x \\
x = \pm\frac{1}{\sqrt{2}} &\quad\Rightarrow\quad z = 1 \mp \sqrt{2}
\end{aligned}
$$

The corresponding points on the ellipse are

$$
P_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 1-\sqrt{2}\right) \quad \text{and} \quad P_2 = \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 1+\sqrt{2}\right)
$$

But here while P_1 and P_2 both give local maxima of f on the ellipse, P_2 is farther from the origin than P_1.

Example

Find the maximum and minimum of $f(x,y) = 5x - 3y$ subject to constraint $x^2 + y^2 = 136$.

Using the definition

$$\begin{aligned}
\frac{\partial f(x,y)}{\partial x} &= \frac{\partial g(x,y)}{\partial x} & 5 &= 2\lambda x & (1)\\
\frac{\partial f(x,y)}{\partial y} &= \frac{\partial g(x,y)}{\partial y} & -3 &= 2\lambda y & (2)\\
g(x,y) &= c & x^2+y^2 &= 136 & (3)
\end{aligned}$$

setting $\lambda = 0$ won't satisfy the first two equations. So assuming that $\lambda \neq 0$ we can solve (1) and (2) to find

$$x = \frac{5}{2\lambda} \quad y = -\frac{3}{2\lambda}$$

Plugging these into constraint equation

$$\frac{25}{4\lambda^2} + \frac{9}{4\lambda^2} = \frac{17}{2\lambda^2} = 136$$

which implies $\lambda^2 = \frac{1}{16} \to \lambda = \pm\frac{1}{4}$. Now that we have λ we can find a few points which will be potential maxima and minima

$$\begin{aligned}
\text{if } \lambda = \frac{1}{4} \text{ we get} \quad & x = -10 \quad y = 6\\
\text{if } \lambda = -\frac{1}{4} \text{ we get} \quad & x = 10 \quad y = -6
\end{aligned}$$

Maxima and Minima can be determined by plugging in the values

$$\begin{aligned}
f(-10,6) &= -68 \quad \text{Minimum } (-10,6)\\
f(-10,6) &= 68 \quad \text{Maximum } (10,-6)
\end{aligned}$$

Thus far we have considered cases with constraints with equality now will consider cases which include inqualities

Example

Find the maximum and minimum values of $f(x,y,z) = xyz$ subject to constraints $x+y+z = 1$. Assume that $x,y,z \geq 0$.

Our constraint is a sum of three positive or zero numbers and it must be 1. Therefore the solution will fall in the range $0 \leq x,y,z \leq 1$. So according to extreme value theorem the maximum and minimum value must exist.

$$\frac{\partial f(x,y,z)}{\partial x} = \frac{\partial g(x,y,z)}{\partial x} \qquad yz = \lambda \qquad (1)$$
$$\frac{\partial f(x,y,z)}{\partial y} = \frac{\partial g(x,y,z)}{\partial y} \qquad xz = \lambda \qquad (2)$$
$$\frac{\partial f(x,y,z)}{\partial y} = \frac{\partial g(x,y,z)}{\partial y} \qquad xy = \lambda \qquad (3)$$
$$g(x,y,z) = c \qquad x+y+z = 1 \quad (3)$$

we notice that (1),(2) and (3) are equal to λ, by equating (1) and (2) we fine

$$yz = xz \quad \Rightarrow \quad z(y-x) = 0 \quad \Rightarrow \quad z = 0 \qquad \text{or} \qquad y = x$$

two possibilities i.e. either $z = 0$ or $y = x$. Starting with $z = 0$. In this case we can see from (1) and (2) that we must have $\lambda = 0$. From (3) we see that this means $xy = 0$. Which in turn means either $x = 0$ or $y = 0$. Thus the possible options look like

$$z = 0, \qquad x = 0 \qquad \Rightarrow \quad \text{y=1}$$
$$z = 0, \qquad y = 0 \qquad \Rightarrow \quad \text{x=1}$$

So we have two possible solutions $(0, 1, 0)$ and (1,0,0).
Now lets consider the other possibility $x = y$. We have two possible cases to look at in this case.

The first case $x = y = 0$ in this case we can see from the constraint that we must have $z = 1$ and so we now have the third solution (0,0,1).

The second case $x = y \neq 0$. setting (2) and (3) equal

$$xz = xy \quad \Rightarrow \quad x(z-y) = 0 \quad \Rightarrow \quad x = 0 \qquad \text{or} \qquad z = y$$

Since we have already assumed that $x \neq 0$ and so the only possibility is that $z = y$ which also means that $x = y = z$.
Using this constraint gives

$$3x = 1 \qquad \Rightarrow \quad x = \frac{1}{3}$$

Which implies $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$. These four solutions have been obtained by setting (1) and (2). To find all solutions (1) must be set equal to (3) and similarly (2) must be set equal to (3).

$$yz = xy \quad \Rightarrow \quad y(z-x) = 0 \quad \Rightarrow \quad y = 0 \qquad \text{or} \qquad z = x$$
$$xz = xy \quad \Rightarrow \quad x(z-y) = 0 \quad \Rightarrow \quad x = 0 \qquad \text{or} \qquad z = y$$

Not complete here! Lets check which solutions are maximum and minimum here

$$\begin{array}{llll} f(1,0,0)=0 & f(0,1,0)=0 & f(0,0,1)=0 & \text{All minimum} \\ f(\frac{1}{3},\frac{1}{3},\frac{1}{3}) & & & \text{Maximum} \end{array}$$

Now lets consider a problem dealing with multiple constraints

Example

Find the maximum and minimum values of $f(x, y, z) = 4y-2z$ subject to constraints $2x - y - z = 2$ and $x^2 + y^2 = 1$.

From the second constraint it is visible that $-1 \leq x, y \leq 1$. With this in mind there must also be set of limits on z in order to make sure that the first term constraints is met.

The definition of Lagrange multiplier for multiple constraints

$$\nabla f(\mathbf{x}) - \lambda_i \sum_i g_i(\mathbf{x}) = 0$$

$$\begin{array}{lll} \frac{\partial f(x,y,z)}{\partial x} = 0 & \frac{\partial g_1(x,y,z)}{\partial x} = 2 & \frac{\partial g_2(x,y,z)}{\partial x} = 2x \\ \frac{\partial f(x,y,z)}{\partial y} = 4 & \frac{\partial g_1(x,y,z)}{\partial y} = -1 & \frac{\partial g_2(x,y,z)}{\partial y} = 2y \\ \frac{\partial f(x,y,z)}{\partial z} = -2 & \frac{\partial g_1(x,y,z)}{\partial z} = -1 & \frac{\partial g_2(x,y,z)}{\partial y} = 0 \end{array}$$

The system of equations to be solved

$$0 = 2\lambda_1 + 2\lambda_2 x \qquad f_x = \lambda_1 g_{1x} + \lambda_2 g_{2x} \tag{1}$$

$$4 = -\lambda_1 + 2\lambda_2 y \qquad f_y = \lambda_1 g_{1x} + \lambda_2 g_{2x} \tag{2}$$

$$-2 = -\lambda_1 \qquad f_z = \lambda_1 g_{1z} + \lambda_2 g_{2z} \tag{3}$$

$$2x - y - z = 2 \tag{4}$$

$$x^2 + y^2 = 1 \tag{5}$$

We start by noticing that from (5) we get $\lambda_1 = 2$ plugging this (1) and (2) and solving for x and y respectively gives

$$0 = 4 + 2\lambda_2 x \quad \Rightarrow \quad x = -\frac{2}{\lambda_2}$$

$$4 = -2 + 2\lambda_2 y \quad \Rightarrow \quad y = \frac{2}{\lambda_2}$$

Plugging these result in (5)

$$\frac{4}{\mu^2} + \frac{9}{\lambda_2^2} = \frac{12}{\lambda_2^2} \qquad \Rightarrow \quad \lambda_2 = \pm\sqrt{13}$$

♣ Putting the value of $\lambda_2 = +\sqrt{13}$ in above to determine values of x and y

$$x = -\frac{2}{\sqrt{13}} \quad y = \frac{2}{\sqrt{13}}$$

Plugging these results in (4) yields

$$-\frac{4}{\sqrt{13}} - \frac{3}{\sqrt{13}} - z = 2 \quad \Rightarrow \quad z = -2 - \frac{7}{\sqrt{13}}$$

♣ Now we assume $\lambda_2 = -\sqrt{13}$ then

$$x = \frac{2}{\sqrt{13}} \quad y = -\frac{3}{\sqrt{13}}$$

Plugging these results in (4) yields

$$\frac{4}{\sqrt{13}} + \frac{3}{\sqrt{13}} - z = 2 \quad \Rightarrow \quad z = -2 + \frac{7}{\sqrt{13}}$$

Now that we have the two solutions applying them to the function to find which is the minimum and maximum.

$$f\left(-\frac{2}{\sqrt{13}}, \frac{3}{\sqrt{13}}, -2 - \frac{7}{\sqrt{13}}\right) = 4 + \frac{26}{\sqrt{13}} = 11.211 \quad \text{Maximum}$$
$$f\left(\frac{2}{\sqrt{13}}, -\frac{3}{\sqrt{13}}, -2 + \frac{7}{\sqrt{13}}\right) = 4 - \frac{26}{\sqrt{13}} = -3.211 \quad \text{Minimum}$$

Example Find the maximum and minimum of $f(x, y) = 4x^2 + 10y^2$ on a disk $x^2 + y^2 \leq 4$.

Sol: Using the definition:

$$\langle f_x, f_y, f_z \rangle = \lambda \langle g_x, g_y, g_z \rangle = \langle \lambda g_x, \lambda g_y, \lambda g_z \rangle$$

$$\frac{\partial f}{\partial x} = 8x \qquad \frac{\partial g}{\partial x} = 2\lambda x$$
$$\frac{\partial f}{\partial y} = 20y \qquad \frac{\partial g}{\partial y} = 2\lambda y$$
$$\frac{\partial f}{\partial \lambda} = 0 \qquad \frac{\partial g}{\partial \lambda} = x^2 + y^2 - 4$$

from the above three equations we have

$$
\begin{aligned}
8x - 2\lambda x &= 0 \qquad & x = 0 \quad \text{or} \quad \lambda = 4 \\
x(4-\lambda) &= 0 & \\
x &= 0 & y = \pm 2 \\
\lambda &= 4 & \\
20y - 2\lambda y &= 0 & 20y = 8y \to y = 0 \\
x &= \pm 2 &
\end{aligned}
$$

therefore the possible set of solutions is

$$(0,2),(0,-2),(2,0) \quad \text{and} \quad (-2,0)$$

Example Find the minimum value of function $f(x,y,z) = x^2 + 2y^2 + z^2$ subject to constraints

$$
\begin{aligned}
x + 2y + 3z &= 1 \\
x - 2y + z &= 5
\end{aligned}
$$

The Lagrangian function is of the form

$$L(x,y,z,\lambda,\mu) = x^2 + 2y^2 + z^2 + \lambda(x + 2y + 3z - 1) + \mu(x - 2y + z - 5)$$

$$
\begin{aligned}
\frac{\partial L}{\partial x} &= 2x + \lambda + \mu = 0 & x &= -\frac{1}{2}(\lambda + \mu) \\
\frac{\partial L}{\partial y} &= 4y + 2\lambda - 2\mu = 0 & y &= -\frac{1}{4}(2\lambda - 2\mu) \\
\frac{\partial L}{\partial z} &= 2z + 3\lambda + \mu = 0 & z &= -\frac{1}{2}(3\lambda + \mu) \\
\frac{\partial L}{\partial \lambda} &= x + 2y + 3z - 1 = 0 & -6\lambda - \mu &= 1 \\
\frac{\partial L}{\partial \mu} &= x - 2y + z - 5 & -\lambda - 2\mu &= 5
\end{aligned}
$$

which leads to solution set

$$
\begin{aligned}
\lambda = \frac{3}{11} & \qquad \qquad \mu = -\frac{29}{11} \\
x = \frac{13}{11} \quad y = -\frac{16}{11} & \quad z = \frac{10}{11}
\end{aligned}
$$

Example: A company produces steel boxs at three plants in amount x, y and z respectively, producing an annual revenue of $f(x,y,z) = 8xyz^2 - 200(x+y+z)$. The company is to produce 100 units annual. How should the production be distributed to maximize revenue.

Solution: The Lagrangian function is of the form:

$$L(x, y, z, \lambda) = 8xyz^2 - 200(x + y + z) + \lambda(x + y + z - 100)$$

$\nabla L(x, y, z, \lambda)$ is determined as

①	$\dfrac{\partial L}{\partial x} = 8yz^2 - 200 + \lambda = 0$	from ① and ②	$8yz^2 = 8xz^2$
②	$\dfrac{\partial L}{\partial y} = 8xz^2 - 200 + \lambda = 0$	from ② and ③	$8xz^2 = 16xyz$
③	$\dfrac{\partial L}{\partial z} = 16xyz - 200 + \lambda = 0$		
④	$\dfrac{\partial L}{\partial \lambda} = x + y + z - 100 = 0$	The optimal sol. is	$x = 25, y = 25$ and $z = 50$

Supplementary Problems

1. Find minimum of the function $f(x, y) = x^2 + y^2 - 2x + 8y$ subject to constraint $x + 2y = 7$.

2. Find maximum of the function $f(x, y) = 9x^2 + 36xy - 4y^2 - 18x - 8y$ subject to constraint $3x + 4y = 32$.

3. Suppose the temperature at point (x,y) on a metal plate is $T(x, y) = 4x^2 - 4xy + y^2$. An ant walking on the plate traverses a circle of 5 centered around origin. What is the highest and lowest temperature encountered by the ant.

4. Golf ball manufacturer, Pro-T, has developed a profit model that depends on the number x of golf balls sold per month (measured in thousands), and the number of hours per month of advertising y, according to the function

$$z = f(x, y) = 48x + 96y - x^2 - 2xy - 9y^2$$

where z is measured in thousands of dollars. The budgetary constraint function relating the cost of the production of thousands golf balls and advertising units is given by $20x + 4y = 216$. Find the values of x and y that maximize profit, and find the maximum profit.

Chapter 3

Linear Programming

Outline

The objective of this chapter is to familiarize the students with the fundamental concepts of mechanics which will form basis of pivotal concepts of Robotics. The topics included here are

Ⓐ Problem Formulation
Ⓑ Linear Programming Problem
Ⓒ The Simplex Algorithm
Ⓓ The Interior Point Method
Ⓔ Duality in Linear Programming.
Ⓕ Matlab implementation

Optimization is an old problem, but the credit of developing a formal algorithm goes to George Dantzig, who published his algorithm commonly known as 'Simplex Method' and the field is known as linear programming.The linear programming is a small subset of problems within the class of general nonlinear optimization problems. These problems are discussed in later chapters

Linear Programming

It is the process of minimizing a linear objective function subject to finite number of linear equality and inequality constraints. The word programming is historical and predates computer programming.
Examples applications:
♣ Airline crew scheduling

♣ Manufacturing and production planning

♣ Telecommunication Network design

3.1 Several Examples

Activity Analysis and Product Mix

A lumber mills saws both Finish-grade and Construct-grade boards from the logs that it receives. Suppose that it takes 2 hours to rough-saw 1000 boardfeet of finish-grade boards and 5 hours to plane each 1000 broad-foot of the construction-grade boards. Suppose also that it takes 2 hours to rough-saw each 1000 board feet of construction-grade boards but it takes only 3 hours to plane each 1000 board feet of these boards. The saw is available for 8 hours/day and the plane is available for 15 hours/day. If the profit on each 1000 board feet of finish-grade boards is 120 $ and the profit on each 100 board feet of the construction-grade boards is 100 $, how many board feet of each type of the lumber should be sawed to maximize the profit.

Mathematical Model: let x and y denote the amount of finish grade and construction grade lumber respectively to be sawed per day. Let units of x and y be thousand board feet. The number of hours required daily for the saw is

$$2x + 2y$$

since saw is available only for 8 hours a day thus x and y must satisfy the quantity

$$2x + 2y \leq 8$$

Similarly the number of hours required for the plane is

$$5x + 3y$$

So x and y must satisfy

$$5x + 3y \leq 15$$

ofcourse, we must have

$$x \geq 0 \quad \text{and } y \geq 0$$

The profit is $ to be maximized is given by

$$z = 120x + 100y$$

Thus our model is
Find the value of x and y that will

$$\begin{aligned} &\text{maximize } z = 120x + 100y \\ \text{Subject to restrictions} \quad & 2x + 2y \leq 8 \\ & 5x + 3y \leq 15 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

Graphical Visualization and Matlab Demo

3.1.1 Feasible Set

Each linear ineqaulity divides the n-dimensional space into two half-spaces one where the inequality is satisfied, and one where it is not. Feasible set is solutions to a family of linear inequalities.
The linear cost function defines a family of parallel hyperplanes (line in 2D, plane in 3D and so on) want to find one of minimum cost $\rightarrow$ must occur at corner of the feasible set.
The feasible set of standard LP:

- Intersection of a set of half-spaces, called a poly-hydron.
- Its bounded and non-empty its a polytope.

There are three cases:

- Feasible set is empty.
- Cost function is unbounded on feasible set.
- Cost function is maximum on feasible set.

For the two cases are very uncommon for real problems in economics and engineering.

Diet Problem

A nutritionist is planning a menu consisting of two main foods **A** and **B**. Each once of **A** contains 2 units of Fat, 1 unit of Carbohydrates and 4 units of proteins. Each ounce of **B** contains 3 units of Fat, 3 units of Carbohydrates and 3 units of Proteins. The nutritionist wants a meal to provide atleast 18 units of Fat, atleast 12 units of Carbohydrates and atleast 24 units of Proteins. If an ounce of **A** costs 20 and an ounce of **B** costs 25 ; how many ounces of each food should be served to minimize cost of the meal yet satisfy the requirements.

Mathematical Model

let x and y denote the number of ounces of food **A** and **B** that are served. Then x and y have to satisfy the inequality such that

$$2x + 3y \geq 18$$

similarly to meet the nutritionist requirements for carbohydrates and proteins, we must have x and y satisfy

$$x + 3y \geq 12$$
$$4x + 3y \geq 24$$

of course, we also require

$$x \geq 0 \quad \text{and } y \geq 0$$

The cost to be minimized is given by

$$z = 20x + 25y$$

Thus our model is :
Find the values of x and y that will

$$\begin{aligned} &\text{minimize } z = 20x + 25y \\ \text{Subject to restrictions} \qquad &2x + 3y \geq 18 \\ &x + 3y \geq 12 \\ &4x + 3y \geq 24 \\ &x \geq 0 \\ &y \geq 0 \end{aligned}$$

Transportation Problem

A manufacturer of plastic has two plants in Multan and Sukkur respectively. There are three distributing warehouses in Karachi, Lahore and Faisalabad. The Sukkur plant can provide 120 tons of supply per week, whereas Multan unit can supply 140 tons of material per week.
The karachi warehouse needs 100 tons to meet demand, the Lahore warehouse needs 60 tons while Faisalabad needs 80 tons weekly. The following table gives the shipping costs per ton of the product

	To		
From	Karachi	Lahore	Faisalabad
Sukkur	5	7	9
Multan	6	7	10

How many tons of plastic should be shipped from each plant to each warehouse to minimize total shipping cost while meeting demand?
Mathematical Model let P_1 and P_2 denote plants at Sukkur and Multan, respectively. Let W_1, W_2 and W_3 denote the warehouses in Karachi, Lahore and Faisalabad respectively. Let

$$x_{ij} = \text{number of tons shipped from } P_i \text{ to } W_j$$
$$c_{ij} = \text{cost of shipping 1 ton from } P_i \text{ to } W_j$$

for $i = 1, 2$ and $j = 1, 2, 3$. The total amount of plastic to sent from P_1 is

$$x_{11} + x_{12} + x_{13}$$

Since P_1 can supply only 120 tons, we must have

$$x_{11} + x_{12} + x_{13} \leq 120$$

Similarly, since P_2 can supply only 140 tons, we must have

$$x_{21} + x_{22} + x_{23} \leq 140$$

The amount of plastic received at W_1 is

$$x_{11} + x_{21}$$

The demand at W_1 is 100 tons, we must have

$$x_{11} + x_{21} \geq 100$$

similarly, the demand at W_2 and W_3 are 60 and 80 tons respectively,

$$x_{12} + x_{22} \geq 60$$
$$x_{13} + x_{23} \geq 80$$

Of course

$$x_{ij} \geq 0 \quad \text{for } i = 1, 2 \quad \text{and} \quad j = 1, 2, 3$$

The total transportation cost which we want to minimize is:

$$\text{minimize } z := c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{33}$$

Thus our mathematical model is:
Find the values of x_{11},x_{13},x_{13},x_{21},x_{22},x_{23} that will

$$\begin{array}{lll} \text{minimize} & z = \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}x_{ij} & \\ \text{Subject to restrictions} & \sum_{j=1}^{3} x_{ij} \leq s_i & i = 1, 2 \\ & \sum_{j=1}^{3} x_{ij} \geq d_j & j = 1, 2, 3 \\ & x_{ij} \geq 0 & \end{array}$$

where available supply $s_1 = 120$ and $s_2 = 140$ and the required demand are $d_1 = 100$, $d_2 = 60$ and $d_3 = 80$.

Example:
A store sells two types of toys, A and B. The store owner pays 8$ and 14$ for each one unit of toy A and B respectively. One unit of toys A yields a profit of 2$ while a unit of toys B yields a profit of 3$. The store owner estimates that no more than 2000 toys will be sold every month and he does not plan to invest more than 20,000$ in inventory of these toys. How many units of each type of toys should be stocked in order to maximize his monthly total profit profit?

x and y is number of toys A & B	$x \geq 0, y \geq 0$
a total of 2000 toys is to be sold	$x + y \leq 2000$
one unit of toy A yields profit of 2$	
	Profit=$2x + 3y$
one unit of toy B yields profit of 3$	
Cost of A and B is 8$ & 14$	
	$8x + 14y \leq 20000$
total budget is 20,000$	

Example:

A company produces two types of tables, T1 and T2. It takes 2 hours to produce the parts of one unit of T1, 1 hour to assemble and 2 hours to polish.It takes 4 hours to produce the parts of one unit of T2, 2.5 hour to assemble and 1.5 hours to polish. Per month, 7000 hours are available for producing the parts, 4000 hours for assembling the parts and 5500 hours for polishing the tables. The profit per unit of T1 is 90$ and per unit of T2 is 110$. How many of each type of tables should be produced in order to maximize the total monthly profit?

x,y is the number of T1 and T2 tables	$x \geq 0, \quad y \geq 0$
Profit from sale of T1 and T2 is 90$ and 110$	$P(x, y) = 90x + 110y$
T1 Takes 2 / 1 / 2 hrs to produce, assemble and polish	$2x + 4y \leq 7000$
T2 Takes 4 / 2.5 / 1.5 hrs to produce assemble and polish	$x + 2.5y \leq 4000$
total 7000/ 4000 / 5500 hrs to produce, assemble and polish	$2x + 1.5y \leq 5500$

A farmer plans to mix two types of food to make a mix of low cost feed for the animals in his farm. A bag of food A costs 10$ and contains 40 units of proteins, 20 units of minerals and 10 units of vitamins. A bag of food B costs 12$ and contains 30 units of proteins, 20 units of minerals and 30 units of vitamins. How many bags of food A and B should the consumed by the animals each day in order to meet the minimum daily requirements of 150 units of proteins, 90 units of minerals and 60 units of vitamins at a minimum cost?

x,y is number of Bags of food A& B	$x \geq 0 \quad y \geq 0$
Profit from sale of T1 and T2 is 90$ and 110$	$P(x, y) = 90x + 110y$
Food A has 40/ 20 / 10 units of Proteins, Minerals & vitamins	$40x + 4y \leq 7000$
Food B has 30/ 20 /30 units of Proteins, Minerals & vitamins	$x + 2.5y \leq 4000$
Minimum diet of 150, 90 and 60 units of p,m and v is required	$2x + 1.5y \leq 5500$

3.2 General Linear Programming Problem

From the above examples a general linear programming problem can be stated as follows:
Find $x_1, x_2, \cdots, x_n$ that will

$$\begin{array}{ll} \text{Min or Max} & z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{Subject to Restrictions} & \end{array}$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq (\geq)(=) b_1 \\ a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq (\geq)(=) b_2 \\ \vdots \quad + \quad \vdots + \cdots + \vdots &\leq (\geq)(=) \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq (\geq)(=) b_m \end{aligned} \tag{3.1}$$

The General Linear Programming Problem

where in each expression one and only one of the symbols $\geq$, $\leq$ and $=$ occurs. The linear function in (equation) is called the objective function. The equalities and/or inequalities in (equation) are called constraints. Note that the left-hand sides of all the inequalities or equalities are linear function of variables $x_1, x_2, \cdots, x_n$ just as the objective function is. A problem in which not all of the constraints or the objective functions are linear function of the variables is a non-linear programming problems. A linear Programming problem in **standard form** if it is in the following form:

find values of $x_1, x_2, \cdots x_n$ that will

$$\begin{array}{ll} \text{maximize} & z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{s.t.} & \end{array}$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ \vdots \; + \; \vdots \; + \; \cdots \; + \; \vdots &\leq \; \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \end{aligned} \tag{3.2}$$

Linear Programming Problem in Standard Form

A linear program is in **conical form** if it is in the following form:

find values of $x_1, x_2, \cdots x_n$ that will

$$\begin{aligned} \text{maximize} \quad & z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\ \text{s.t.} \quad & \\ & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ & \vdots + \vdots + \cdots + \vdots = \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{aligned} \tag{3.3}$$

Linear Programming Problem in Canonical Form

3.3 Some More Examples

The following problems are neither in canonical nor standard form, why!?

$$\begin{aligned} \text{(a)} \quad \min \quad & z = 3x + y \\ \text{s.t.} \quad & \\ & 2x + y \leq 4 \\ & 3x - 2y \leq 6 \\ & x \geq 0 \ \ y \geq 0 \end{aligned}$$

$$\begin{aligned} \text{(b)} \quad \max \quad & z = 2x_1 + 3x_2 + 4x_3 \\ \text{s.t.} \quad & \\ & 3x_1 + 2x_2 - 3x_3 \leq 4 \\ & 2x_1 + 3x_2 + 2x_3 \leq 6 \\ & 3x_1 - x_2 + 2x_3 \geq -8 \\ & x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \end{aligned}$$

$$\begin{aligned} \text{(c)} \quad \max \quad & z = 3x + 2y + 3v - 2w \\ \text{s.t.} \quad & \\ & 2x + 6y + 2v - 4w = 7 \\ & 3x + 2y - 5v + w = 8 \\ & 3x + 2y - 5v + w \leq 4 \\ & x \geq 0, \quad y \geq 0, \quad v \geq 0, \quad w \geq 0 \end{aligned}$$

$$\begin{aligned}
\text{(d)}\quad \min \quad & z = 2x + 5y + u + v + 4w \\
\text{s.t.} \quad & \\
& 3x + 2y - 4u = 4 \\
& 4x + 5y + 3u + 2v = 7 \\
& 6x + 7y + 2v + 5w \le 4 \\
& x \ge 0, \quad y \ge 0, \\
& u \ge 0, \quad v \ge 0, \\
& w \ge 0,
\end{aligned}$$

$$\begin{aligned}
\text{(e)}\quad \min \quad & z = 2x + 5y \\
\text{s.t.} \quad & \\
& 3x + 2y \le 6 \\
& 2x + 9y \le 8 \\
& x \ge 0
\end{aligned}$$

$$\begin{aligned}
\text{(f)}\quad \min \quad & z = 2x_1 + 3x_2 + x_3 \\
\text{s.t.} \quad & \\
& 2x_1 + x_2 - x_3 = 4 \\
& 3x_1 + 2x_2 + x_3 = 8 \\
& x_1 - x_2 = 6 \\
& x_1 \ge 0, \quad x_2 \ge 0
\end{aligned}$$

Minimization Problem as Maximization Problem

Every minimization problem can be viewed as a maximization problem and conversely. This can be seen from the observation that

$$\min \quad \sum_{i=1}^{n} c_i x_i = \max\Bigg(-\sum_{i=1}^{n} c_i x_i \Bigg)$$

That is to minimize the objective functions we could maximize its negative and then change the sign of the answer. The solution **(a)** would be to change the sign of cost function.

$$\begin{aligned}
\text{(a)}\quad \min \quad & -z = -3x - y \\
\text{s.t.} \quad & \\
& 2x + y \le 4 \\
& 3x - 2y \le 6 \\
& x \ge 0 \;\; y \ge 0
\end{aligned}$$

Reversing an Inequality

If we multiply the inequality

$$k_1x_1 + k_2x_2 + \cdots + k_nx_n \geq b$$

by **-1**, we obtain the inequality

$$-k_1x_1 - k_2x_2 - \cdots - k_nx_n \leq -b$$

so in the part **(b)** of optimization problem can be converted into standard form by multiplying the 3rd constraint inequality with a **-1**.

$$
\begin{array}{lll}
\text{(b)} & \max & z = 2x_1 + 3x_2 + 4x_3 \\
 & \text{s.t.} & \\
 & & \ \ 3x_1 + 2x_2 - 3x_3 \leq 4 \\
 & & \ \ 2x_1 + 3x_2 + 2x_3 \leq 6 \\
 & & -3x_1 + x_2 - 2x_3 \leq 8 \\
 & & x_1 \geq 0, \ \ x_2 \geq 0, \\
 & & x_3 \geq 0.
\end{array}
$$

Changing Equality into Inequality

Observe that we can write the equation $x = 6$ into a pair of inequalities $x \leq 6$ and $-x \leq -6$. In general case the equation

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \tag{3.4}$$

can be written as a pair of inequalities

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad \sum_{j=1}^{n} -a_{ij}x_j \leq -b_i$$

Thus **(c)** can be formulated as an standard problem by rewriting constraints (1) and (2) as an inequality:

$$
\begin{aligned}
\text{(c)} \quad \max \quad & z = 3x + 2y + 3v - 2w \\
\text{s.t.} \quad & \\
& 2x + 6y + 2v - 4w \leq 7 \\
& -2x - 6y - 2v + 4w \leq -7 \\
& 3x + 2y - 5v + w \leq 8 \\
& -3x - 2y + 5v - w \leq -8 \\
& 3x + 2y - 5v + w \leq 4 \\
& x \geq 0, \quad y \geq 0, \\
& v \geq 0, \quad w \geq 0.
\end{aligned}
$$

Unconstrained Variables

Suppose that x_j is not constrained to be a non-negative. We replace x_j with two new variables x_j^+ and x_j^-, letting

$$x_j = x_j^+ - x_j^-$$

where $x_j^+ \geq 0$ and $x_j^- \geq 0$. That is any number is the difference of two non-negative numbers. Therefore problems **(e)** and **(f)** can be converted into standard form as

$$
\begin{aligned}
\text{(e)} \quad \min \quad & z = 2x + 5y^+ - 5y^- \\
\text{s.t.} \quad & \\
& 3x + 2y^+ - 2y^- \leq 6 \\
& 2x + 9y^+ - 9y^- \leq 8 \\
& x \geq 0, \qquad y^+ \geq 0, \qquad y^- \geq 0,
\end{aligned}
$$

$$
\begin{aligned}
\text{(f)} \quad \min \quad & z = -2x_1 - 3x_2 - x_3^+ + x_3^- \\
\text{s.t.} \quad & \\
& 2x_1 + x_2 - x_3^+ + x_3^- = 4 \\
& 3x_1 + 2x_2 + x_3^+ - x_3^- = 8 \\
& x_1 - x_2 = 6 \\
& x_1 \geq 0, \quad x_2 \geq 0 \\
& x_3^+ \geq 0, \quad x_3^- \geq 0
\end{aligned}
$$

3.3.1 Matrix Notation

It is convenient to write a linear programming problem in matrix notation. Consider the standard linear programming problem

$$\begin{aligned}
\max \quad & z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.} \quad & \\
& a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \quad \le b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \quad \le b_2 \\
& \ \vdots \ + \ \vdots \ + \ \cdots \ + \ \vdots \quad \le \ \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \quad \le b_m \\
& x_j \ge 0 \qquad j = 1, 2, \cdots, n
\end{aligned}$$

letting

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad \text{and,} \qquad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

We can write our linear programming problem as:
find vector $\mathbf{x} \in \mathbb{R}^n$ that will

$$\begin{aligned}
\max \quad & z = \mathbf{c}^T\mathbf{x} \\
\text{s.t} \quad & \mathbf{Ax} \le \mathbf{b} \\
& \mathbf{x} \ge 0
\end{aligned}$$

where inequality implies that every entry of the vector satisfies the condition.
Example The lumber problem:
Find a vector $\mathbf{x} \in \mathbb{R}^2$ that will

$$\begin{aligned}
\max \quad & z = \begin{bmatrix} 120 & 100 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
\text{s.t} \quad & \begin{bmatrix} 2 & 2 \\ 5 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} 8 \\ 15 \end{bmatrix} \\
& \begin{bmatrix} x \\ y \end{bmatrix} \ge 0
\end{aligned}$$

Definition: A vector $\mathbf{x} \in \mathbb{R}^n$ satisfying the constraints of a linear programming problem is called **feasible solution** to the problem. A feasible solution that maximizes the objective function of a linear programming problem is called **optimal solution**.

Changing an Inequality to an Equality

Consider a constraint

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \tag{3.5}$$

it is possible to convert (3.5) into an equation by introducing a new variable u_i as writing

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + u_i = b_i \tag{3.6}$$

The variable u_i is non-negative and is called **slack variable** because it takes up the slack between left and right side of the equation (3.5).
Converting the linear programming problem from the standard form as defined in (3.2) to a problem in canonical form (3.3) by introducing a slack variable in each of the constraints. Note each constraint will get a different variable. In the $i-$th constraint inequality

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \tag{3.7}$$

we introduce a slack variable x_{n+i} and write

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + x_{n+i} = b_i \tag{3.8}$$

because of the direction of the inequality we know that $x_{i+1} \leq 0$; Therefore the canonical form of the problem is

$$\begin{array}{ll}
\max & z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.} & \\
& \begin{array}{lllll}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{n+1} & & & = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & + x_{n+2} & & = b_2 \\
\vdots + \vdots + \cdots + \vdots & & & = \vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & & + x_{n+m} & = b_m
\end{array} \\
& x_1 \geq 0, x_2 \geq 0, \cdots, x_n \geq 0, x_{n+1} \geq 0, \cdots, x_{n+m} \geq 0
\end{array} \tag{3.9}$$

the new problem has m equations and $n+m$ unknowns in addition to non-negativity restrictions on variable $x_1, x_2, \cdots, x_n, x_{n+1}, \cdots, x_{n+m}$.
If $\mathbf{y} = [y_1, \cdots, y_n]^T$ is a feasible solution to the problem given by (3.2) then we

define y_{n+i} for $i = 1, 2, \cdots, m$ by

$$y_{n+i} = b_i - a_{i1}y_1 - a_{i2}y_2 - a_{i3}y_3 - \cdots - a_{in}y_n \tag{3.10}$$

That is y_{n+i} is the difference between the right side of the $i-$th constraint in (3.2) and the value of the left side of this constraint at feasible solution $\mathbf{y}$. Since each constraint in (3.2) is of the $\leq$ form. We conclude that

$$y_{n+i} \leq 0, \qquad i = 1, 2, \cdots, m \tag{3.11}$$

This $[y_1, y_2, \cdots, y_n, y_{n+1}, \cdots, y_{n+m}]^T$ satisfies (3.6) and (3.9) and the vector $\hat{\mathbf{y}} = [y_1, y_2, \cdots, y_n, y_{n+1}, \cdots, y_{n+m}]$ is feasible solution to the linear programming problem in canonical form given by (3.9). Then clearly $y_1 \leq 0$, $y_2 \leq 0 \cdots$ $y_n \leq 0$; Since $y_{n+i} \geq 0$, $i = 1, 2, \cdots, m$ we see that

$$a_{i1}y_1 + a_{i2}y_2 + \cdots + a_{in}y_n \leq b_i, \qquad i = 1, 2, \cdots, m \tag{3.12}$$

Here $\mathbf{y} = [y_1, \cdots, y_n]^T$ is a feasible solution to the linear programming problem in a standard form given by (3.2). The discussion above shows that a feasible solution to a standard linear programming problem yields solution to a canonical linear programming problem by adjoining the values of the slack variables. conversely, a feasible solution to the corresponding standard linear programming problem by truncating the slack variables.

Now coming back to the mill example and introducing the slack variables

$$\begin{array}{rl} \text{maximize} & z = 120x + 100y \\ \text{s.t} & 2x + 2y \quad + u \qquad = 8 \\ & 5x + 3y \qquad\quad + v = 15 \\ x \geq 0 & y \geq 0 \quad u \geq 0 \quad v \geq 0 \end{array}$$

In terms of model the slack variables u and v is the difference between the total amount of the time that the saw is available, **8 hours** and the amount of time is it actually used $2x + 2y$ hours. Similarly, the slack variable v is the difference between the total amount of time that the plane is available **15 hours** and the amount of time it is actually used $5x + 3y$ hours.

Assuming $x = 2$, $y = 1$ is a feasible solution to the problem in standard form. For this feasible solution we have

$$u = 8 - 2 \cdot 2 - 2 \cdot 1 = 2$$
$$v = 15 - 5 \cdot 2 - 3 \cdot 1 = 2$$

Thus $x = 2$, $y = 1$, $u = 2$ and $v = 2$ is a feasible solution to the new form of the problem.
Now consider the following set of values ($x = 1$, $y = 1$, $u = 4$ and $v = 7$) these values lead to the new solution since

$$2 \cdot 1 + 2 \cdot 1 + 4 = 8$$
$$5 \cdot 1 + 3 \cdot 1 + 7 = 15$$

Consequently $x = 1$ and $y = 1$ is a feasible solution to the given problem. As we have already evaluated the solution we know that if $x = \frac{3}{2}, y = \frac{5}{2}$, Therefore in canonical form the solution to this problem

$$u = 2 \cdot \frac{3}{2} + 2 \cdot \frac{5}{2} - 8 = 0$$
$$v = 5 \cdot \frac{3}{2} + 3 \cdot \frac{5}{2} - 15 = 0$$

This is an optimal solution to the problem in canonical form is

$$x = \frac{3}{2} \quad y = \frac{5}{2} \quad u = 0 \quad v = 0$$

The linear program problem given in (3.9) can be written in matrix form as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & & \ddots & \ddots & 0 \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \qquad \text{and} \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$
\begin{aligned}
\max \quad & z = \mathbf{c}^T\mathbf{x} \\
\text{s.t} \quad & \mathbf{Ax} = \mathbf{b} \\
& \mathbf{x} \geq 0
\end{aligned}
$$

Note: This problem is in canonical form.

Example: Convert the following LPP into canoncial form.

$$
\begin{aligned}
\max \quad & 80x + 70y \\
\text{s.t.} \quad & \\
& 6x + 3y \leq 96 \\
& x + y \leq 18 \\
& 2x + 6y \leq 72 \\
& x \geq 0 \quad y \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & 6x_1 + 14x_2 + 13x_3 \\
\text{s.t.} \quad & \\
& 0.5x_1 + 2x_2 + x_3 \leq 24 \\
& x_1 + 2x_2 + 4x_3 \leq 60 \\
& x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0
\end{aligned}
$$

Exercise: Students are encourage to sketch and then use linprog tool to evaulate the following LPPs

$$
\begin{aligned}
\max \quad & z = x + 2y \\
\text{s.t.} \quad & \\
& 3x + y \leq 6 \\
& 3x + 4y \leq 12 \\
& x \geq 0 \quad y \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & z = 5x - 3y \\
\text{s.t.} \quad & \\
& x + 2y \leq 4 \\
& x + 3y \geq 6 \\
& x \geq 0 \quad y \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & z = 3x + y \\
\text{s.t.} \quad & \\
& -3x + y \geq 6 \\
& 3x + 5y \leq 15 \\
& x \geq 0 \quad y \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & z = 2x + 3y \\
\text{s.t.} \quad & \\
& 3x + y \leq 6 \\
& x + y \leq 4 \\
& x \geq 0 \quad y \geq 0
\end{aligned}
$$

3.4 Duality

1. Every linear program has a Dual.
2. If the original is a maximization, the dual is a minimization problem and vice-versa
3. Solution of one problems leads to the solution of other.

Primal Problem: Maximize $\mathbf{c}^T\mathbf{x}$ subject to $\mathbf{Ax} \leq \mathbf{b}$, $\mathbf{x} \geq 0$.

Dual Problem: Minimize $\mathbf{y}^T\mathbf{b}$ subject to $\mathbf{A}^T\mathbf{y} \geq \mathbf{c}$, $\mathbf{y} \geq 0$.

If one has optimal solution so does the other and their values are the same.

In this section we illustrate how to associate a minimization problem with linear programming problem in standard form. There are some very interesting interpretations of the associated problem that we discuss.

Generally a problem in standard form can be thought of as manufacturing problem, one in which scarce resources are allocated in a way that maximize profit. The associated minimization problem is the one that seeks to minimize cost.

Consider the pair of linear programming problem

$$\begin{aligned} \max \quad & z = 5x + 2y \\ \text{s.t} \quad & x + 3y \leq 12 \\ & 3x - 4y \leq 9 \\ & 7x + 8y \leq 20 \\ & x \geq 0 \quad y \geq 0 \end{aligned}$$

The corresponding minimization problem would be

$$\begin{aligned} \max \quad & z = 12a + 9b + 20c \\ \text{s.t} \quad & a + 3b + 7c \geq 5 \\ & 3a - 4b + 8c \geq 2 \\ & a \geq 0 \quad b \geq 0 \quad c \geq 0 \end{aligned}$$

$$\begin{aligned} \max \quad & z = \mathbf{c}^T\mathbf{x} & \qquad \min \quad & w = \mathbf{b}^T y \\ \text{s.t} \quad & \mathbf{Ax} \leq b & \qquad \text{s.t} \quad & \mathbf{A}^T y \geq c \\ & \mathbf{x} \geq 0 & & \mathbf{y} \geq 0 \end{aligned} \tag{3.13}$$

where $\mathbf{A}$ is $m \times n$ matrix $\mathbf{c}$ and $\mathbf{x}$ are $n \times 1$ column vectors and $\mathbf{b}$ and $\mathbf{y}$ are $m \times 1$ column vectors.
These problems are called dual problems. The formal is called primal and later is call dual problem. Theorem: given a primal problem, the dual of its dual problem is again the primal problem.

Primal Problem	Dual Problem
Maximization	Minimization
Coefficients of the objective function	Right-hand side of constraints
Coefficients of $i-$th constraint is an inequality	Coefficients of $i-$th variable, one in each constraint
$i-$th constraint is an inequality $\leq$	$i-$th variable is ≥ 0
$i-$th constraint is an inequality $=$	$i-$th variable is unrestricted
$j-$th variable is unrestricted	$j-$th constraint is an equality
$j-$th variable is ≥ 0	$j-$th constraint is an inequality $\geq$
Number of variables	Number of constraints

Relation between Primal and Dual Problem

Any linear programming problem can have three possible outcomes

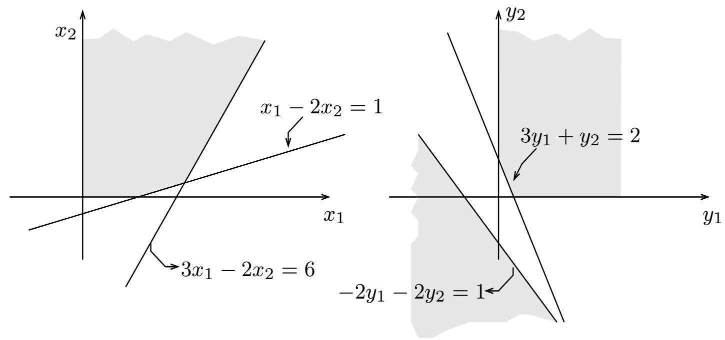
1. No feasible solution exists.
2. There is a finite optimal solution
3. Feasible solution does not exist but objective function is unbounded.

Since the dual problem is a linear programming problem attempting to solve it also leads to these three possible outcomes. Consequently in considering relationship between solutions to the primal and dual problems there are nine alternative pairs of solutions. We now present theorems that show which of these alternatives actually can occur.

		Primal		
		Finite Optimal	Unbounded	infeasible
Dual	Finite Optimal	possible	impossible	impossible
Dual	Unbounded	impossible	impossible	possible
Dual	Infeasible	impossible	possible	possible

Example-1

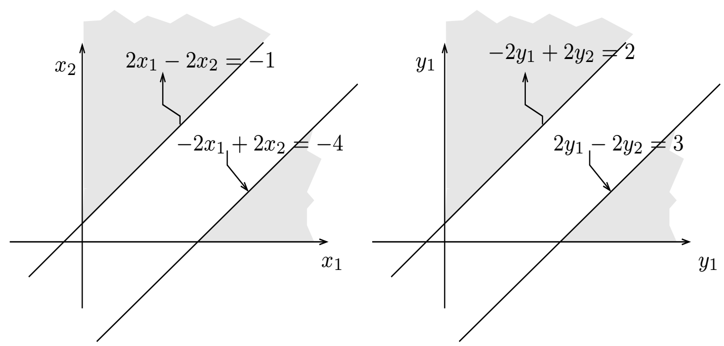
$$\begin{aligned}
&\max \quad z = 2x_1 + x_2 && \min \quad z' = 6w_1 + w_2 \\
&\text{s.t.} && \text{s.t.} \\
&\quad 3x_1 - 2x_2 \leq 6 && \quad 3w_1 + w_2 \geq 2 \\
&\quad x_1 - 2x_2 \leq 6 && \quad -2w_1 - 2w_2 \geq 1 \\
&\quad x_1 \geq 0, x_2 \geq 0 && \quad w_1 \geq 0, w_2 \geq 0
\end{aligned}$$



Example-2

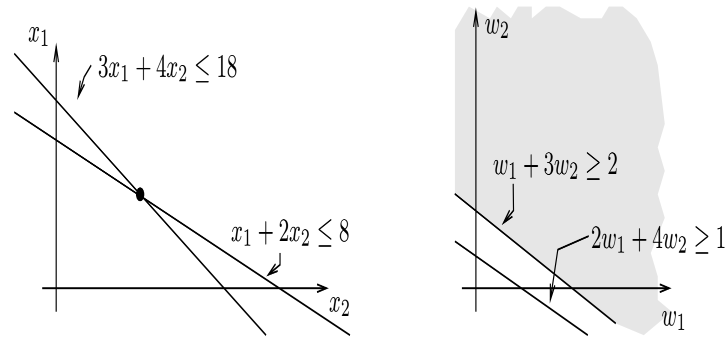
$$\begin{aligned} \max \quad & z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & \\ & 3x_1 - 2x_2 \leq -1 \\ & -2x_1 + 2x_2 \leq -4 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & z' = -w_1 - 4w_2 \\ \text{s.t.} \quad & \\ & 2w_1 - 2w_2 \geq 3 \\ & -2w_1 + 2w_2 \geq 2 \\ & w_1 \geq 0, w_2 \geq 0 \end{aligned}$$



Example-3

$$\begin{array}{llll} \max & z = 2x_1 + x_2 & \min & z' = 8w_1 + 18w_2 \\ \text{s.t.} & & \text{s.t.} & \\ & x_1 + 2x_2 \le 1 & & w_1 + 3w_2 \ge 2 \\ & 3x_1 + 4x_2 \le 18 & & 2w_1 + 4w_2 \ge 1 \\ & x_1 \ge 0, x_2 \ge 0 & & w_1 \ge 0, w_2 \ge 0 \end{array}$$



3.4.1 Weak Duality Theorem

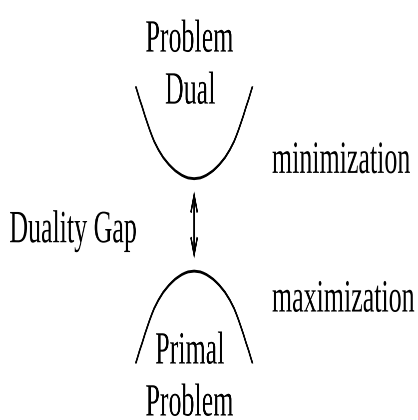
if x_0 is a feasible solution to the primal problem

$$\begin{array}{llll} \max & z = \mathbf{c}^T\mathbf{x} & \min & w = \mathbf{b}^T y \\ \text{s.t} & \mathbf{A}\mathbf{x} \le b & \text{s.t} & \mathbf{A}^T y \ge c \\ & \mathbf{x} \ge 0 & & \mathbf{y} \ge 0 \end{array} \tag{3.14}$$

then

$$\mathbf{c}^T\mathbf{x}_0 \le b^T\mathbf{y}_0 \tag{3.15}$$

i.e. the value of objective function of the dual problem is always greater than or equal to the value of the objective function of the primal problem



Proof:

$$\mathbf{A}\mathbf{x}_0 \leq \mathbf{b}$$

it follows from above that

$$\mathbf{y}_0^T \mathbf{A}\mathbf{x}_0 \leq \mathbf{y}_0^T \mathbf{b} = \mathbf{b}^T \mathbf{y}_0$$

since $\mathbf{y}_0 \geq 0$. The equality in above expression comes from the fact that $\mathbf{y}_0 \mathbf{b}$ is a scalar.
Since $\mathbf{y}_0$ is a feasible solution we have

$$\mathbf{A}^T \mathbf{y}_0 \geq \mathbf{c}$$

taking its transpose yields

$$\mathbf{y}_0 \mathbf{A} \geq \mathbf{c}^T$$

Multiply by $\mathbf{x}_0$ (which in non-negative) and does not change in the inequality. We get

$$\mathbf{y}_0^T \mathbf{A}\mathbf{x}_0 \geq \mathbf{c}^T \mathbf{x}_0$$

combining the inequalities above gives the desired results.

3.4.2 Strong Duality Theorem

a. if either the primal or dual problem has a feasible solution with a finite optimal objective value, then the other problem has feasible solution with same objective value.

b. if primal and dual equations as defined in (3.13) have feasible solutions then

- if the primal problem has an optimal solution- say $\mathbf{x}_0$
- if the dual problem has an optimal solution - say $\mathbf{y}_0$ then
- $\mathbf{c}^T \mathbf{x}_0 = \mathbf{b}^T \mathbf{y}_0$

Proof is not pursued here.

3.5 Complementary Slackness Theorem

For an pair of optimal solutions to primal and duals problems we have:

a for $i = 1, 2, \cdots, m$ the product of the $i-$th slack variable for the primal problem and the $i-$th dual variable is *zero*. That is $x_{n+i} \cdot w_i = 0$ for $i = 1, \cdots, m$.

b for $j = 1, 2, \cdots, n$, the product of $j-$th slack variable for the dual problem and the $j-$th variable for the primal problem is zero.

Another way to state the theorem is that if $i-$th slack variable of the primal problem is not zero, then $i-$th dual variable must be zero. Likewise, if $j-$th slack variable of the dual problem is not zero, then the $j-$th primal variable must be zero. Note that it is possible for both the slack variable and its corresponding dual variable to be zero.
Consider we have a linear optimization problem is standard form

$$\begin{array}{llll} & \text{Primal} & & \text{Dual} \\ \max & z = \sum_{j=1}^{n} c_j x_j & \min & w = \sum_{i=1}^{n} y_i b_i \\ \text{s.t.} & & \text{s.t.} & \\ & z = \sum_{j=1}^{n} a_{ij} x_j \leq b_i & & w = \sum_{i=1}^{n} a_{ij} y_i \geq c_j \\ & x_j \geq 0 \quad j = 1, \cdots, m & & y_i \geq 0 \quad i = 1, \cdots, n \end{array}$$

we add slack variables to convert the system into canonic form

$$\begin{array}{llll} \max & z = \sum_{j=1}^{n} c_j x_j & \min & w = \sum_{i=1}^{n} y_i b_i \\ \text{s.t.} & & \text{s.t.} & \\ & z = \sum_{j=1}^{n} a_{ij} x_j + s_i = b_i & & w = \sum_{i=1}^{n} a_{ij} y_i - e_j = c_j \\ & i = 1, \cdots, m & & j = 1, \cdots, n \\ & x_j \geq 0 \quad j = 1, \cdots, n & & y_i \geq 0 \quad i = 1, \cdots, n \\ & s_i \geq 0 \quad i = 1, \cdots, m & & e_j \geq 0 \quad j = 1, \cdots, m \end{array} \tag{3.16}$$

The complementary slackness theorem states that:
let $\mathbf{x} = [x_1, \cdots, x_n]$ be a feasible solution to a primal
let $\mathbf{y} = [y_1, \cdots, y_m]$ be a feasible solution to a dual
Then $\mathbf{x}$ is a primal and y is dual optimal solution if

$$\begin{array}{ll} s_i y_i = 0 & i = 1, \cdots, m \\ e_j x_j = 0 & j = 1, \cdots, n \end{array} \tag{3.17}$$

Primal		Dual	
max	$z = 60x_1 + 30x_2 + 20x_3$	min	$w = 48y_1 + 20y_2 + 8y_3$
s.t.		s.t.	
	$8x_1 + 6x_2 + x_3 \leq 48$		$8y_1 + 4y_2 + 2y_3 \geq 60$
	$4x_1 + 2x_2 + 1.5x_3 \leq 20$		$6y_1 + 2y_2 + 1.5y_3 \geq 30$
	$2x_1 + 1.5x_2 + 0.5x_3 \leq 8$		$8y_1 + 4y_2 + 2y_3 \geq 20$

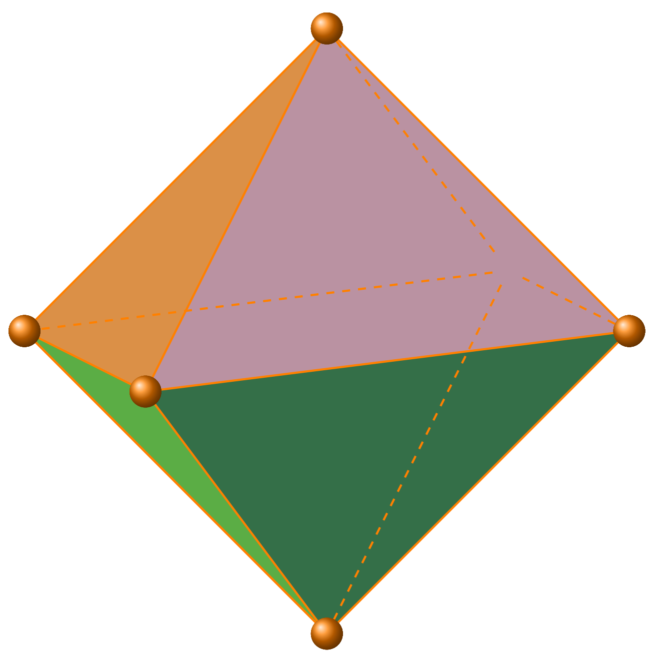
(3.18)

Optimal Primal Solution	Optimal Dual Solution
$x_1 = 2, x_2 = 0, x_3 = 8$	$y_1 = 0, y_2 = 10, y_3 = 10$
$s_1 = 24, s_2 = 0, s_3 = 0$	$e_1 = 0, e_2 = 5, e_3 = 0$
$z^* = 280$	$w^* = 280$

3.6 Simplex Algorithm

Simplex algorithm is one of the most important things to be invented discovered in 20th century. This has allowed for systematic optimization of linear optimization problems. The algorithm has evolved tremendously since its conception in the late 1940's.

The main idea of simplex method is that it finds the optimal problem by traversing through the corner points of the feasibility region as illustrated in the figure below:



The main draw back of this algorithm is its computational complexity is polynomial order i.e. C_n^k, where n is the number of variables while C and k are some +ve constants. And this algorithm may eventually not converge to optimality under certain conditions. This algorithm is elegant and suitable for computer implementation however its equally solvable for simple optimization problems.

The simplex Method
Standard Form

To solve a linear programming problem in standard form use the following steps

a. Convert each inequality in the set of constraints to an equation by adding slack variables.

b. Create initial simplex tableau.

c. Locate the most negative entry in at bottom row. The column from this entry is called **entry** column. If tie occurs, any of the tied entries can be used to determine the entering column.

d. From the ratio of the entries in the b-column with their corresponding positive enteries in the entering column. The **Departing** row corresponds to the smallest non-negative ratio b_i/a_{ij}, if all entries in the entering column are 0 or negative, then there is no maximum solution. For ties choose either entry. The entry in the departing row and entering column is called the **pivot**.

e. Use elementary row operations so that pivot is **1**, and all other entries in the entering column are 0. The process is called pivoting.

f. If all entries in the bottom row are zero or positive, this is the final tableau. If not, go back to step e.

Example-1:lets consider the Lumber mill optimization problem

$$\begin{aligned}
\max \quad & z = 120x + 100y \\
\text{s.t.} \quad & \\
& 2x + 2y \leq 8 \\
& 5x + 3y \leq 15 \\
& x \geq 0 \quad y \geq 0
\end{aligned}$$

must be converted into canonical form

$$\begin{aligned}
& z - 120x - 100y = 0 \\
& 2x + 2y + s_1 \qquad = 8 \\
& 5x + 3y \qquad + s_2 = 15
\end{aligned}$$

①	x	y	s_1	s_2	z	b
s_1	2	2	1	0	0	8
s_2	⑤	3	0	1	0	15
	-120 ↑	-100	0	0	1	0

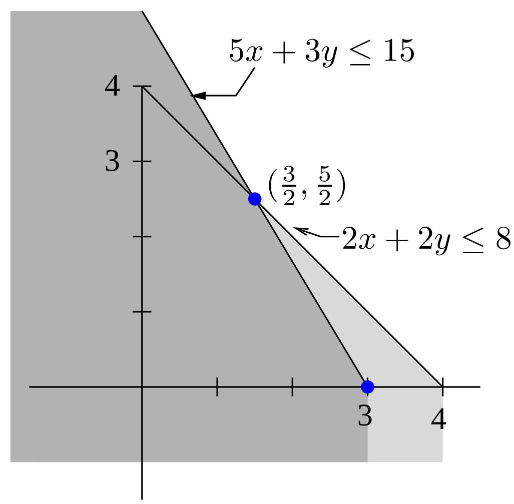
identify the most negative entry in the last row i.e. *entering row* and calculate the ratio $\frac{b_i}{a_{ij}}$. The row which provides the lowest ratio is considered and corresponding row element is called the *pivot element*.

Now we need to set the pivot element $\rightarrow 1$ and the other elements of pivot column $\rightarrow 0$.

$$R_1 = R_1 - \frac{2}{5} \times R_2$$
$$R_2 = \frac{1}{5} \times R_2$$
$$R_3 = R_3 + 24 * R_2$$

Application of these row operations on the tableau ① would yield into

②	x	y	s_1	s_2	z	b
s_1	0	$\frac{4}{5}$	1	-$\frac{2}{5}$	0	2
x	1	$\frac{3}{5}$	0	$\frac{1}{5}$	0	3
	0	-28	0	24	1	360



marking the entering column and leaving row into the simplex tableau above

②	x	y	s_1	s_2	z	b
s_1	0	$\textcircled{\frac{4}{5}}$	1	$-\frac{2}{5}$	0	2
x	1	$\frac{3}{5}$	0	$\frac{1}{5}$	0	3
	0	-28 ↑	0	24	1	360

$$R_1 = \frac{5}{4} \times R_1$$
$$R_2 = R_2 - \frac{3}{4} \times R_1$$
$$R_3 = R_3 + 35 * R_1$$

Application of these row operations on the tableau ② would yield into

③	x	y	s_1	s_2	z	b
y	0	1	$\frac{5}{4}$	-$\frac{1}{2}$	0	$\frac{5}{2}$
x	1	0	-$\frac{3}{4}$	$\frac{1}{2}$	0	$\frac{3}{2}$
	1	0	35	10	1	430

Since all the elements of in bottom row are ≥ 0 the optimal point has been reached.
Example-2:lets consider another optimization problem

$$\begin{aligned} \max \quad & z = 2x + 3y \\ \text{s.t.} \quad & \\ & x + 3y \leq 9 \\ & 2x + 3y \leq 12 \\ & x \geq 0 \quad y \geq 0 \end{aligned}$$

must be converted into canonical form

$$\begin{aligned} z - 2x - 3y &= 0 \\ x + 3y + s_1 \qquad &= 9 \\ 2x + 3y \qquad + s_2 &= 12 \end{aligned}$$

①	x	y	s_1	s_2	z	b
s_1	1	③	1	0	0	9
s_2	2	3	0	1	0	12
	-2	-3 ↑	0	0	1	0

Now we need to set the pivot element → 1 and the other elements of pivot column → 0.

$$R_1 = \frac{1}{3} R_1$$
$$R_2 = R_2 - R_1$$
$$R_3 = R_3 + R_1$$

Application of these row operations on the tableau ① would yield into

②	x	y	s_1	s_2	z	b
y	$\frac{1}{3}$	1	$\frac{1}{3}$	0	0	3
s_2	1	0	-1	1	0	3
	-1 ↑	0	1	0	1	9

marking the entering column and leaving row into the simplex tableau above

$$
\begin{aligned}
R_1 &= R_1 - \frac{1}{3} \times R_2 \\
R_2 &= R_2 \\
R_3 &= R_3 + R_2
\end{aligned}
$$

Application of these row operations on the tableau ② would yield into

③	x	y	s_1	s_2	z	b
y	0	1	$\frac{2}{3}$	-$\frac{1}{3}$	0	2
x	1	0	-1	1	0	3
	0	0	0	1	1	12

Example-3: Lets consider the following optimization problem

$$
\begin{aligned}
\max \quad & f = 2x + 4y + 3z \\
\text{s.t.} \quad & \\
& x + y + z \leq 12 \\
& x + 3y + 3z \leq 24 \\
& 3x + 6y + 4z \leq 90 \\
& x \geq 0 \quad y \geq 0 \quad z \geq 0
\end{aligned}
$$

must be converted into canonical form

$$
\begin{aligned}
f - 2x - 4y - 3z &= 0 \\
x + y + z + s_1 \qquad\qquad &= 12 \\
x + 3y + 3z + \quad +s_2+ \quad &= 24 \\
3x + 6y + 4z + \qquad +s_3 &= 90
\end{aligned}
$$

	x	y	z	s_1	s_2	s_3	f	b
s_1	1	1	1	1	0	0	0	12
s_2	1	3	3	0	1	0	0	24
s_3	3	6	4	0	0	1	0	90
	-2	-4 ↑	-3	0	0	0	1	0

$$R_1 =$$
$$R_2 =$$
$$R_3 =$$

	x	y	z	s_1	s_2	s_3	z	b
s_1	$\frac{2}{3}$	0	0	1	$-\frac{1}{3}$	0	0	4
s_2	$\frac{1}{3}$	1	1	0	$\frac{1}{3}$	0	0	8
s_3	1	0	-2	0	-2	1	0	42
	$-\frac{2}{3}$	0	1	0	$\frac{4}{3}$	0	1	32

$$R_1 =$$
$$R_2 =$$
$$R_3 =$$

	x	y	z	s_1	s_2	s_3	z	b
s_1	1	0	0	$\frac{3}{2}$	$-\frac{1}{2}$	0	0	6
s_2	0	1	1	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	6
s_3	0	0	-2	$-\frac{3}{2}$	$-\frac{3}{2}$	1	0	36
	0	0	1	1	1	0	1	36

3.6.1 Variants of Simplex Algorithm

Simplex method opened up an new field of research aiming to to optimize the implementation and avoiding degeneracy. Only a few methods are presented here for brevity.

3.6.2 The big-M Method

One way to guarantee that the new optimal solution is optimal for the original LP, is to modify the objective function, so that the artificial variable will take value zero in the new optimal solution. In other words, a very large penalty is added to the objective function if the slack variables take positive value.
Consider the following LP:

$$\begin{aligned} \max \quad & z = 2x + 3y \\ \text{s.t.} \quad & \\ & x + 3y \leq 9 \\ & 2x + 3y \leq 12 \\ & x \geq 0 \quad y \geq 0 \end{aligned}$$

$$\begin{aligned} z - 2x - 3y &= 0 \\ x + 3y + s_1 \qquad + Ma_1 &= 9 \\ 2x + 3y \quad + s_2 \qquad &= 12 \end{aligned}$$

Here M is a symbolic *big* positive number. It is so big that even if a_1 is slightly big than 0, the penalty -Ma_1 will be severe. In this case, it is reasonable that the optimal solution to this new LP will take value 0 for the artificial variable a_1, and hence an optimal solution for the original LP.

3.6.3 Two Phase Method

The two-phase method and big-M method are equivalent. In practice, however, most computer codes utilizes the two-phased method. The reasons are that the inclusion of the big number M may cause round-off error and other computational difficulties. The two-phase method, on the other hand, does not involve the big number M and hence all the problems are avoided. The two-phase method, as it is called, divides the process into two phases.
Phase 1: The goal is to find a BFS for the original LP. Indeed, we will ignore the original objective for a while, and instead try to minimize the sum of all artificial variable. At the end of phase 1, a **basic feasible solution** (BFS) is obtained if the minimal value of this LP is zero.
Phase 2: Drop all the artificial variables, change the objective function back to the original one. Use just the regular simplex algorithm, with the starting BFS obtained in Phase 1.

3.7 Karush-Kuhn-Tucker Conditions

For a general optimization problem

$$\begin{aligned} \text{optimize} \quad & f(x) \qquad x \in \Re^n \\ & g_i(x) \geq b_i \qquad i = 1, \cdots, n \\ \text{s.t.} h_j(\mathbf{x}) &= b_j \qquad j = 1, \cdots, m \end{aligned}$$

The necessary condition for optimization

1. Primal Feasibility

$$g_i(x^*) - b_i \quad \text{for} \qquad i = 1, \cdots, n \quad \text{feasible}$$

$$h_j(x^*) - b_j \quad \text{for} \qquad j = 1, \cdots, m \quad \text{feasible}$$

2. Stationarity

$$\max \quad \nabla f(x^*) = \sum_{i=1}^{n} \mu_i \nabla (g_i(x^*) - b_i) + \sum_{i=1}^{m} \lambda_j \nabla (h_j(x^*) - b_j)$$

3. Complimentary Slackness

$$\mu_i g_i(x^*) = 0 \quad \text{where} \mu_i \geq 0$$

Example-1 Consider the following optimization

$$\begin{aligned} & f(x) = x \\ & s.t. \\ & \quad y \geq (1-x)^3 \\ & \quad y \geq 0 \end{aligned}$$

According to Karush-Khun-Tucker conditions, any feasible solution must satisfy the following condition

$$\nabla f(\mathbf{x}^*) - \sum_{j=1}^{k} \lambda_j \nabla h_j(\mathbf{x}^*) - \sum_{i=1}^{m} \mu_i \nabla g_i(\mathbf{x}^*) = 0$$

$$\mu_i^* g_i(\mathbf{x}^*) = 0$$

$$\mu_i^* \geq 0$$

please recall that

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\nabla f(\cdot)} - \underbrace{\begin{bmatrix} -3(x-1)^2 \\ 1 \end{bmatrix}}_{\nabla g_1(\cdot)} - \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{\nabla g_2(\cdot)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We need to plot the graph to illustrate the nature of the graph and also we need to justify why the value is 'zero'.
Since no value of μ_1 and μ_2 exist such that

$$\nabla f(\mathbf{x}^*) - \sum_{i=1}^{m} \mu_i \nabla g_i(\mathbf{x}^*) = 0$$

Example-2 Consider the following optimization

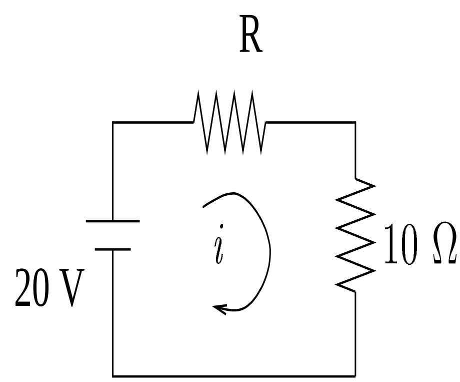
$$\begin{aligned} \max \quad & -(x-2)^2 - 2(y-1)^2 \\ \text{s.t.} \quad & \\ & x + 4y \leq 3 \\ & -x + y \leq 0 \end{aligned}$$

$$\underbrace{\begin{bmatrix} -2(x-2) \\ -4(y-1) \end{bmatrix}}_{\nabla f(\cdot)} \qquad \underbrace{\begin{bmatrix} 1 \\ 4 \end{bmatrix}}_{\nabla g_1(\cdot)} \qquad \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\nabla f(\cdot)}$$

there are several possibilities and one of them may be feasible and optimal

$$\mu_1 = \mu_2 = 0 \rightarrow x = 2, y = 1$$
$$\mu_1 = 0, x - y = 0 \rightarrow x = \frac{4}{3}, \mu_2 = -\frac{4}{3}$$
$$3 - x - 4y = 0, \mu_2 = 0 \rightarrow x = \frac{5}{3}, y = \frac{1}{3}, \mu_1 = \frac{2}{3}$$
$$3 - x - 4y = 0, x - y = 0 \rightarrow x = \frac{3}{5}, y = \frac{3}{5}, \mu_1 = \frac{22}{25}, \mu_2 = -\frac{48}{25}$$

Example



$$i = \frac{20}{10 + R} \qquad\qquad p = i^2 R$$

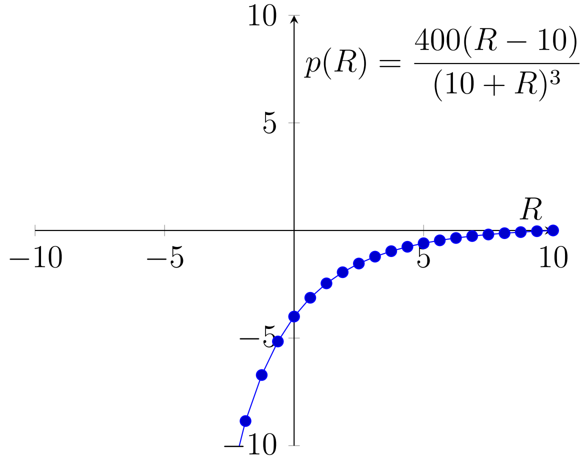
The optimization problem is defined as

$$\begin{aligned} \min \quad & -\frac{400R}{(10+R)^2} \\ \text{s.t.} \quad & \\ & -R \geq 0 \end{aligned}$$

The KKT condition for a feasible solution

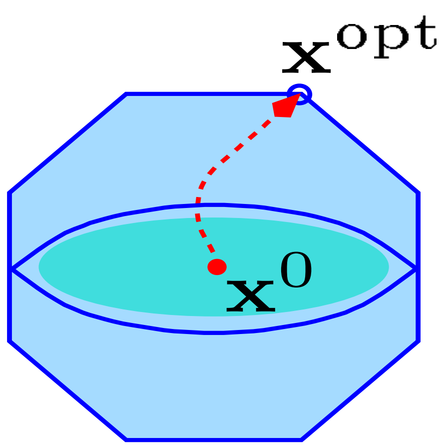
$$\nabla f(\cdot) - \mu \nabla g(\cdot) = 0$$
$$\frac{400(R - 10)}{(10 + R)^3} - \mu = 0$$

The graphical illustration of the graph is as follows



3.8 Interior Point Method

The simplex method (SM) has some convergence problems (i.e. it does not always converge to the optimal solution) additionally the complexity of SM is polynomial time. In 1984 Karamaker proposed an algorithm which unlike SM (which treads the edges of the Feasibility region) works through the interior of feasibility region to find the optimal point. The path following method is not described by Newton's method and Barrier function. The method is called interior point method because this algorithm is initialized with a reference point $\mathbf{x}_0$ in the interior of feasibility region. The concept is illustrated in figure below:



The idea is based on Newton's Method on finding minima and maxima of function. For a smooth nonlinear function solve:

$$f(\mathbf{x}) = 0$$

Taylor's theorem (linearization)

$$f(\mathbf{x}^0 + d_{\mathbf{x}}) \approx f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0 + d_{\mathbf{x}})$$

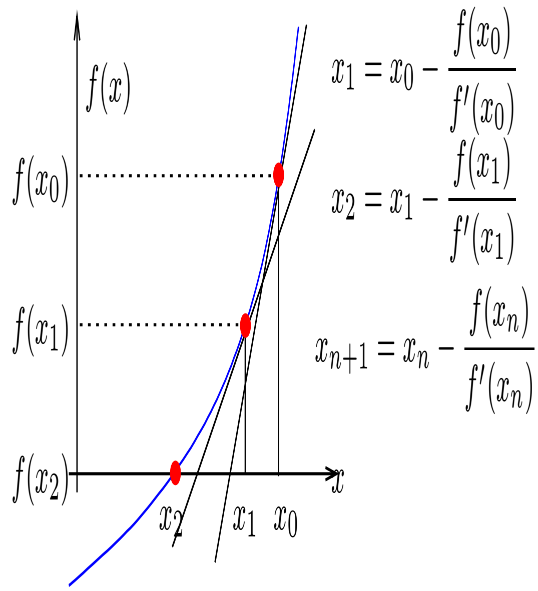
If $\mathbf{x}^0$ is initial guess, computer $d_{\mathbf{x}}$ such that $f(\mathbf{x} + d_{\mathbf{x}}) = 0$

$$f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0) d_{\mathbf{x}} = 0 \Rightarrow d_{\mathbf{x}} = -(\nabla f(\mathbf{x}^0))^{-1} f(\mathbf{x}^0)$$

$d_{\mathbf{x}}$ defines the search direction , new point $\mathbf{x}^+$ i.e.

$$\mathbf{x}^+ = \mathbf{x}^0 + \alpha d_{\mathbf{x}}$$

where $0 \leq \alpha \leq 1$ is the step size

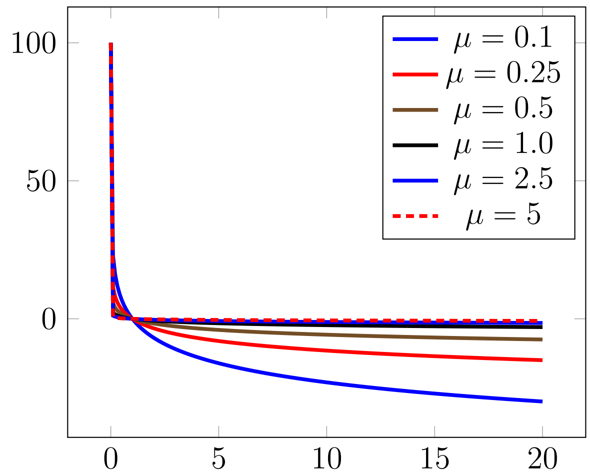


The problem can be reformulated from the standard to canonical form

min $\quad f(x) \quad x \in \Re^n$	min $\quad f(x) \quad x \in \Re^n$
s.t.	s.t.
$h(\mathbf{x})$	$h(\mathbf{x}) - \mathbf{b} - \mathbf{w}$
$\mathbf{w} \geq 0$	$h(\mathbf{x}) = 0$
	$\mathbf{x} \geq 0$
	$\mathbf{w} \geq 0$

An elegant way to get rid of limitations on variables $\mathbf{x}$ is to include a penalty in the cost function. This technique is typically known as the Barrier function

Primal		Dual	
min	$f(x) \quad x \in \Re^n$	min	$f(x) \quad x \in \Re^n$
s.t.		s.t.	
	$h(\mathbf{x}) = 0$		$h(\mathbf{x}) - \mu \sum_{i=1}^{n} \ln(x_i)$
	$\mathbf{x} \geq 0$		$h(\mathbf{x}) = 0$
			$c(\mathbf{x}) = 0$

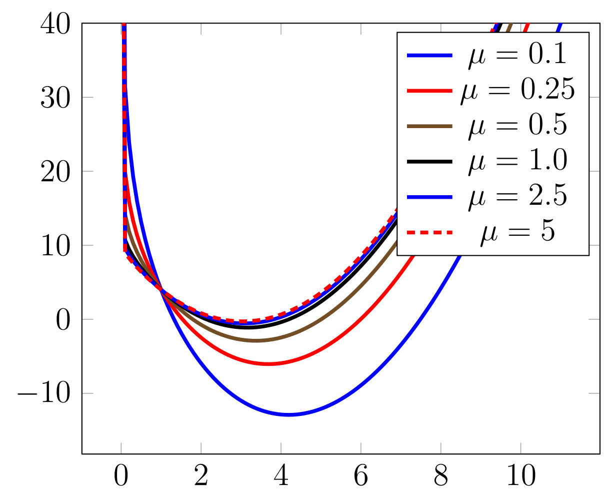


Example: Consider the optimization problem in standard format

$$\begin{aligned} \min \quad & (x-3)^2 \\ \text{s.t.} \quad & \\ & x \leq 0 \end{aligned}$$

which is

$$\min \quad (x-3)^2 - \mu \ln(x)$$



$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \\ & g_i(\mathbf{x}) \geq 0 \qquad i = 1, \cdots, m \\ & \mathbf{x} \leq 0 \end{aligned}$$

Translating the problem into canonical form

$$\begin{aligned}\min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \\ & g_i(\mathbf{x}) - \mathbf{s} = 0 \qquad i = 1, \cdots, m \\ & \mathbf{s} \leq 0\end{aligned}$$

The logarithmic barrier function is now introduced:

$$\begin{aligned}\min \quad & f(\mathbf{x}) - \mu \sum_{i=1}^{m} \log(s_i) \\ \text{s.t.} \quad & \\ & h(\mathbf{x}) - \mathbf{s} = 0 \qquad i = 1, \cdots, m \\ & \mathbf{s} \leq 0\end{aligned}$$

Now incorporate the equality constraint into the objective function using Lagrange multiplier

$$\min \qquad f(\mathbf{x}) - \mu \sum_{i=1}^{m} \log(s_i) - \mathbf{y}^T(g(\mathbf{x}) - \mathbf{s})$$

checking the stationarity of KKT conditions

$$\begin{aligned}\nabla f(\mathbf{x}) - \nabla g(\mathbf{x})^T \mathbf{y} &= \mathbf{0} \\ -\mu \mathbf{W}^- 1\mathbf{e} + \mathbf{y} &= \mathbf{0} \\ g(\mathbf{x}) - \mathbf{s} &= \mathbf{0}\end{aligned}$$

rearranging these equations we have

$$\begin{aligned}\nabla f(\mathbf{x}) - \nabla g(\mathbf{x})^T \mathbf{y} &= \mathbf{0} \\ \mathbf{WYe} &= \mu \mathbf{e} \\ g(\mathbf{x}) - \mathbf{s} &= \mathbf{0}\end{aligned}$$

Utilize newton's method to determine the search directions Δx, Δs and Δy,

$$\begin{bmatrix}\mathbf{G}(x,y) & \mathbf{0} & -\mathbf{A}(x)^T \\ \mathbf{0} & \mathbf{Y} & \mathbf{W} \\ \mathbf{A}(x) & -\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\Delta x \\ \Delta s \\ \Delta y\end{bmatrix} = \begin{bmatrix}-\Delta f(\mathbf{x}) + A(\mathbf{x})^T \mathbf{y} \\ \mu \mathbf{e} - \mathbf{WYe} \\ -g(\mathbf{x}) + \mathbf{s}\end{bmatrix}$$

where $\mathbf{G}(x,y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 g_i(\mathbf{x})$ and $\mathbf{A}(x) = \nabla g(x)$.

Using the set of equations

$$\begin{bmatrix} -\mathbf{G}(x,y) & \mathbf{A}^T(x) \\ \mathbf{A}(x) & \mathbf{W}\mathbf{Y}^{-1} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{bmatrix} = \begin{bmatrix} \Delta f(\mathbf{x}) - A^T(\mathbf{x})\mathbf{y} \\ -g(\mathbf{x}) + \mu\mathbf{Y}^{-1}\mathbf{e} \end{bmatrix}$$

From here, perform iterations:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \Delta\mathbf{x}^k$$
$$\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha^k \Delta\mathbf{s}^k$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha^k \Delta\mathbf{y}^k$$

The interior point method approximates the constraints of a linear programming model as a set of boundaries surrounding a region. These approximations are used when the problem has constraints that are discontinuous or otherwise troublesome, but can me modified so that a linear solver can handle them. Once the problem is formulated in the correct way, Newton's method is used to iteratively approach more and more optimal solutions within the feasible region. Two practical algorithms exist in IP method: barrier and primal-dual. Primal-dual method is a more promising way to solve larger problems with more efficiency and accuracy. As shown in the figure above, the number of iterations needed for the primal-dual method to solve a problem increases logarithmically with the number of variables, and standard error only increases rapidly when a very large number of dimensions exist.

Chapter 4

NonLinear Programming Techniques

Outline

Now that we have familiarized ourselves with simplest (Linear) programming techniques which are although important but very limited in their capabilities. In this chapter we study some of the wider class of Programming techniques which cater to a wider class of linear programming techniques.

- Ⓐ Quadratic Programming
- Ⓑ Cone Programming
- Ⓒ Integer Programming
- Ⓓ Second Order Cone Programming.
- Ⓔ Semi-Definite Programming .
- Ⓕ Matlab Implementations.
- Ⓖ .
- Ⓗ.

4.1 Integer Programming

Thus far we have been concerned with finding optimal values of parameters which are continuous in nature. Now we consider a special type of optimization problems where optimal solution take on only integer values. This kind of optimization is particularly useful if we have to decide which stocks to buy, the optimal route to destination, the optimal warehouse to get/send goods to.

$$\begin{aligned} \min \quad & z : c^T x \\ \text{s.t} \quad & Ax \leq b \\ & 0 \leq x_j \leq 1 \qquad x_j \in \mathbb{Z} \quad j = 1, \cdots, n \end{aligned}$$

In its general form the Integer programming problem can have a mix of possible outcomes i.e. some of the parameters can take on real values while other parameters could only be integer values. While in another case the problem could be a purely binary problem.

Example: Depot Location

Example: a company has selected m possible sites for distribution of its products in a certain area. There are n customers in the area and the transport cost of supplying the whole of customer j's requirements over the given planning period from potential cite i is c_{ij}. Should site i be developed it will cost f_i to construct depot there. Which sites should be selected to minimize the total construction and transportation cost?

To solve this problem we introduce m variables $y_1, \cdots, y_m$ which can only take values 0 or 1 and correspond to a particular site being not developed or developed respectively. We next define x_{ij} to be the fraction of customer j's requirements supplied from depot i in a given solution. The problem can be expressed as

$$\begin{aligned}
\min \quad & z : \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i \\
\text{s.t} \quad & \sum_{i=1}^{m} x_{ij} = 1; x_{ij} \leq y_i \quad i = 1, \cdots, m \quad j = 1, \cdots, n \\
& x_{ij} \geq 0, 0 \leq y_i \leq 1, \quad y_i \in \mathbb{Z} \quad i = 1, \cdots, m \quad j = 1, \cdots, n
\end{aligned}$$

Note that $y_i = 0$ then $f_i y_i = 0$ and there is no contribution to the total cost. Also $x_{ij} \leq y_i$ implies $x_{ij} = 0$ for $j = 1, \cdots n$ and so no good are distributed from site i (i.e. no depot at site i).

On the other hand, if $y_i = 1$ then $f_i y_i = f_i$ which the cost of constructing depot i. Also $x_{jj} \leq y_i$ becomes $x_{ij} \leq 1$ which holds anyway from constraints.

this need to be explained better

4.2 Quadratic Programming

The problem is defined as

$$\begin{aligned}
\min \quad & x^T P x + q^T x \\
\text{s.t} \quad & Gx \succeq h \\
& Ax = b \\
& x \geq 0
\end{aligned}$$

where $Gx \succeq h$ implies every element-wise inequality.

Example:

$$\begin{aligned}
\min \quad & \frac{1}{2}x^2 + 3x + 4y \\
\text{s.t} \quad & x + 3y \geq 15 \\
& 2x + 5y \leq 100 \\
& 3x + 4y \leq 80 \\
& x, y \geq 0
\end{aligned}$$

This problem can be expressed as

$$\min_{x,y} \frac{1}{2} \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

This problem can be fed into matlab 'Quadprog' for solution for example.

a. Write the following polynomial as $x^T Ax$.

b. Formulate the following optimization problem

$$\begin{aligned}
\min \quad & 2x_1^2 + x_2^2 - 2x_1x_2 - 5x_1 - 2x_2 \\
\text{s.t} \quad & 3x_1 + 2x_2 \leq 20 \\
& -5x_1 + 3x_2 \leq 4 \\
& x_1, x_2 \geq 0
\end{aligned}$$

Notice that the constraints are only linear yet a more generalized form 'Quadratic Constraints Quadratic Programming' considers more general quadratic constraints (more on this later).

4.3 Geometric Programming

Geometric Programming was invented by Duffin, Petterson and Zener in 1967. This type of optimization problem finds application in geometrical problems, problems which can be accurately approximated by power laws. A geometric program is classified into two types namely monomials and posynomials which are defined as follows

$$f(x) = Cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$$

where $C \geq 0$, $a_i \in \mathbb{R}$, a_i can take -ve values aswell. The posynomials can be defined as

$$f(x) = \sum_{k=1}^{K} C_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$$

Quick Examples:

$$\begin{array}{ll} 2x_1^{-\pi} x_2^{0.5} + 3x_1 x_3^{100} & \text{is a posynomial in } \mathbf{x} \\ x_1 - x_2 & \text{is not a posynomial} \\ x_1/x_2 & \text{is a monomial thus also a posynomial.} \end{array}$$

The optimization problem can be framed as

$$\begin{array}{llll} & \min & f_0(x) & \\ \text{GP} & \text{s.t} & f_i(x) \leq 1 & i = 1, \cdots, n \\ & & h_j(x) = 0 & j = 1, \cdots, m \\ & & x \in \mathbb{R} & \\ & & x \geq 0 & \end{array}$$

where $f_i(x)$ is posynomial and $h_j(x)$ is a monomial.

Quick example: Consider a general maximization problem

$$\begin{array}{ll} \max & \dfrac{x^2}{yz} \\ \text{s.t} & 1 < x < 5 \\ & 2 < y < 4 \\ & y^2 + 2xy + 5\dfrac{z^2}{x} + x < \sqrt{y} \\ & \dfrac{x}{z} = y^2 \\ & x, y, z \in \mathbb{R}; x, y, z > 0 \end{array}$$

The problem can be equivalently represented (in standard form) as

$$\begin{array}{ll} \min & x^{-2}yz \\ \text{s.t} & x^{-1} \leq 1 \\ & \dfrac{1}{5}x \leq 1 \\ & 2y^{-1} \leq 1 \\ & y^{\frac{3}{2}} + 2x^{\frac{1}{2}} + 5z^2x^{-1}y^{-\frac{1}{2}} + xy^{-\frac{1}{2}} \leq 1 \\ & xz^{-1}y^{-2} \leq 1 \end{array}$$

4.4 Cone Programming

In the case of linear programming (Primal problem) in standard form we had

$$\begin{aligned} \min \quad & z : c^T x \\ \text{s.t} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Now, with conic programming we replace the condition by requiring that $x \in K$. The primal form of conic program:

$$\begin{aligned} \inf \quad & z : c^T x \\ \text{s.t} \quad & Ax = b \\ & x \in K \end{aligned}$$

K is a set such that if $x, y \in K$ then $\alpha x + \beta y \in K$ for $\alpha, \beta \geq 0$.
Some commonly used cones using in conic programming

1. The non-negative orthant, $K = \{x \in R^n : x \geq 0\}$.
2. The second order cone $K_{\text{soc}} = \{x \in R^2 : x_n^2 \geq \sum_{i=1}^{n-1} x_i^2, x_n \geq 0\}$ which is also called Lorentz cone or the ice cream cone.
3. The positive semi-definite cone $K = \{X \in R^N : X^T = X, v^T X v \geq 0 \forall v \geq 0 \forall v \in R^n\}$. Note that X is a symmetric matrix and our conic program becomes

$$\begin{aligned} \inf \quad & z : \sum c_{ij} x_{ij} \\ \text{s.t} \quad & \sum_{ij} a_{ij}(k) x_{ij} = b_k \quad k = 1, \ldots, m \\ & x_{ij} \in K \end{aligned}$$

4.5 Semi-definite Programming

Semi-definite programming is another significant milestone in the general theory of optimization the term was coined in 1990's and has been an active area of research since then.

Semi-definite programming as a generalization of linear programming enables us to specify in addition to linear constraints a set of 'semi-definite' constraints, special form of nonlinear constraints. Starting with the case of linear programming we know that a linear optimization problem is framed as

$$
\text{LP} \quad \begin{array}{ll} \min & \mathbf{c} \cdot \mathbf{x} \\ \text{s.t} & \mathbf{a}_i \cdot \mathbf{x} = b_i \qquad i = 1, \cdots, m \\ & x \in \Re^n_+ \\ & x_1, x_2 \geq 0 \end{array}
$$

$$
\text{LD} \quad \begin{array}{ll} \max & \sum_{i=1}^{m} y_i b_i \\ \text{s.t} & \sum_{i=1}^{m} y_i a_i + s = c \\ & x \in \Re^n_+ \\ & x_1, x_2 \geq 0 \end{array}
$$

Duality gap: $\mathbf{c} \cdot \mathbf{x} - \sum_{i=1}^{m} y_i b_i = (c \cdot x - \sum_{i=1}^{m} y_i a_i) \cdot x = s \cdot x \geq 0.$

Definition: if X is an $n \times n$ matrix then X is positive semi-definite matrix if

$$v^T \mathbf{X} v \geq 0 \qquad \forall \qquad v \in \Re^n$$

and positive definite matrix if $v^T \mathbf{X} v > 0$

$\mathcal{S}^n$: set of $n \times n$ symmetric matrices.
$\mathcal{S}^n_+$: set of positive semi-definite $n \times n$ symmetric matrices. $X \succcurlyeq 0$.
$\mathcal{S}^n_{++}$: set of positive definite $n \times n$ symmetric matrices. $X \succ 0$.

$$X \succcurlyeq Y \Rightarrow X - Y \succcurlyeq 0$$

For a symmetric matrix A, the following statements are equivalent

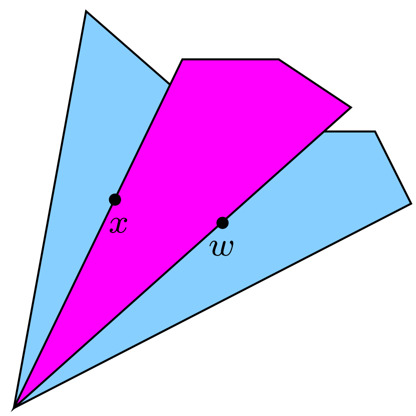
- $A \succcurlyeq 0$

- All eigenvalues of A are non-negative.

- $A = C^T C$, where $C \in \mathbb{R}^{m \times n}$

Semi-definite Cone:
K is a closed convex cone if

$$x, w \in K \Rightarrow \alpha x + \beta w \in k, \qquad \alpha, \beta \geq 0$$
$$K \text{ is a closed set.}$$



Remark 1: $S_+^n = \{X \in S^n | X \succcurlyeq 0\}$ is a closed convex cone in $\Re^{n^2}$ of dimension $n \times (n+1)/2$. Proof: Suppose that $X, W \in S_+^n$ $\forall \alpha, \beta \geq 0$, $\forall v \in \Re^n$

$$v^T(\alpha X + \beta W)v = \alpha v^T X v + \beta v^T W v \geq 0$$

where $\alpha \cdot X + \beta \cdot W \in S_+^n$.
Properties of Symmetric Martix:
$X \in S^n \Rightarrow X = QDQ^T$ where Q is orthonormal i.e. $Q^T = Q^{-1}$, D is a diagonal matrix. The columns of Q form a set of n orthogonal eigenvectors of X, whose eignenvalues are the corresponding entries of D.

Facts of Symmetric Matrices

- If $X \in S^n$, then $X = QDQ^T$ for some orthonormal matrix Q and some diagonal matrix D. (recall that Q is orthonormal means $Q^{-1} = Q^T$).

- if $X = QDQ^T$ as above, then the columns of Q form a set of n orthogonal eigenvectors of X whose eigen values correspond to the entries of diagonal matrix D.

- $X \succcurlyeq 0$ if and only if $X = QDQ^T$ where eigenvalues (i.e. diagonals of D) are all non-negative.
- $X \succ 0$ if and only if $X = QDQ^T$ where eigenvalues (i.e. diagonals of D) are all positive.
- M is a symmetric, then $\det(M) = \Pi_{j=1}^{n} \lambda_j$.
- $X \succcurlyeq 0$ then $X_{ii} \geq 0$, $i = 1, \cdots, n$.
- $X \succcurlyeq 0$ and if $X_{ii} = 0$, then $X_{ij} = X_{ji}0$ for all $j = 1, \cdots, n$.
- Consider matrix M defined as

$$M = \begin{bmatrix} P & v \\ v^T & d \end{bmatrix}$$

  where $P \succ 0$, v is a vector , d is a scalar then $M \succcurlyeq 0$ if and only if $d - v^T P^{-1} v \geq 0$.
- for a given column vector a, the matrix $X = aa^T$ is a symmetric positive semi-definite i.e. $X = aa^T \succcurlyeq 0$

Linear function of X
if $C(X)$ is a linear function of X, then $C(X)$ can be written as $C * X$, where

$$C * X := \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \tag{4.1}$$

if X is a symmetric matrix, wlog matrix C is also symmetric

Semi-definite Program:

$$\begin{array}{lll} & \min & \mathbf{C} * \mathbf{X} \\ & \text{s.t} & \mathbf{A}_i * \mathbf{X} = b_i \qquad i = 1, \cdots, m \\ \text{SDP} & & X \succcurlyeq 0 \end{array}$$

$$A_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{bmatrix} \quad A_1 = \begin{bmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{bmatrix}, \; b_1\text{=}11 \text{ and } b_2 = 19$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{bmatrix}$$

$$C * X = x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33}$$

$$\begin{array}{cl} \min & x_{11} + 4x_{12} + 6x_{13} + 9x_{22} + 0x_{23} + 7x_{33} \\ \text{s.t} & x_{11} + 2x_{13} + 3x_{13} + 2x_{21} + 9x_{22} + 0x_{23} + 7x_{33} = 11 \\ \text{s.t} & 0x_{11} + 4x_{12} + 16x_{13} + 6x_{22} + 0x_{23} + 4x_{33} = 19 \\ & \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{12} & x_{22} & x_{23} \\ x_{13} & x_{23} & x_{33} \end{bmatrix} \succcurlyeq 0 \end{array}$$

SDP looks remarkably similar to the linear program. However, the standard LP constraints that x must lie in the non-negative orthant is replaced by the constraint that the variable X must lie in the cone of positive semi-definite matrices. Just as $x \geq 0$ states that each of the n components of x must be non-negative, it may be helpful to think that $X \succcurlyeq 0$ as stating that each of the n eigenvalues of X must be non-negative. It is easy to see that a linear program LP is a special instance of SDP.

Semi-definite Programming Duality

The dual problem of SDP is defined as:

$$\begin{array}{ccl} & \max & \sum_{i=1}^{m} y_i b_i \\ \text{SDD} & \text{s.t} & \sum_{i=1}^{m} y_i A_i + S = C \\ & & S \succcurlyeq 0 \end{array}$$

One convenient way of thinking about this problem is as follows. Given multipliers $y_1, \cdots, y_m$, the objective is to maximize the linear function $\sum_{i=1}^{m} y_i b_i$. The constraints of SDD state that the matrix S defined as

$$S = C - \sum_{i=1}^{m} y_i A_i$$

must be postive semi-definite. That is

$$C - \sum_{i=1}^{m} y_i A_i \succcurlyeq 0$$

Construction of the dual of the problem presented earlier

$$\begin{array}{lll} & \max & 11y_1 + 19y_2 \\ & \text{s.t} & y_1 \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{bmatrix} + y_2 \begin{bmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{bmatrix} + S = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{bmatrix} \\ \text{SDP} & & S \succcurlyeq 0 \end{array}$$

which can be written as

$$\begin{array}{lll} & \max & 11y_1 + 19y_2 \\ & \text{s.t} & y_1 \begin{bmatrix} 1 - y_1 - 0y_2 & 2 - 0y_1 - 2y_2 & 3 - 1y_1 - 8y_2 \\ 2 - 0y_1 - 2y_2 & 9 - 3y_1 - 6y_2 & 0 - 7y_1 - 0y_2 \\ 3 - y_1 - 8y_2 & 0 - 7y_1 - 0y_2 & 7 - 5y_1 - 4y_2 \end{bmatrix} \\ \text{SDP} & & S \succcurlyeq 0 \end{array}$$

According to the authors it is often easier to see and work with a semi-definite program when it is presented in dual SDD, since the variables are the m multipliers $y_1, \cdots, y_m$.

As in linear programming, we can switch from one format of SDP to any other format with great ease, and there is no loss of generality in assuming particular specific format for the primal and the dual.

SDP for Convex Quadratically Constrained Quadratic Programming:

A convex quadratically constrained quadratic program is a problem of the form

$$\begin{array}{lll} & \min & x^T Q_0 x + q_0^T x + c_0 \\ \text{QCQP} & \text{s.t} & x^T Q_i x + q_i^T x + c_i \leq 0 \qquad i = 1, \cdots, m \\ & & Q_0, Q_i \succcurlyeq 0 \qquad i = 1, \cdots, m \end{array}$$

If we can factor each Q_i in $M_i^T M_i$ for some matrix M_i. Then the optimization problem can be reformulated as

$$
\begin{array}{lll}
 & \min\limits_{\mathrm{x},\theta} & \theta \\
\text{QCQP} & \text{s.t} & \begin{bmatrix} I & M_0 x \\ x^T M_0^T & -c_0 - q_0^T x + \theta \end{bmatrix} \succcurlyeq 0 \\
\text{QCQP} & \text{s.t} & \begin{bmatrix} I & M_i x \\ x^T M_i^T & -c_i - q_i^T x + \theta \end{bmatrix} \succcurlyeq 0 \qquad i = 1, \cdots, m \\
 & & Q_0, Q_i \succcurlyeq 0 \qquad i = 1, \cdots, m
\end{array}
$$

SDP for Second order Cone Programming

A second order cone optimization problem (SOCP) is an optimization problem of the form:

$$
\begin{array}{lll}
 & \min & c^T x \\
\text{SOCP} & \text{s.t} & Ax = b \\
 & & \|Q_i x + d_i\| \leq (g_i^T x + h_i) \qquad i = 1, \cdots, K
\end{array}
$$

where $\|v\|$ is the standard Euclidean norm i.e. $\|v\|^2 := \sqrt{v^T v}$. The norm constraints in SOCP are called the 'second order cone constraints'. The constraints are convex. Since

$$
\|Qx + d\| \leq (g^T x + h) \iff \begin{bmatrix} (g^T x + h)I & (Qx + d) \\ (Qx + d)^T & g^T x + h \end{bmatrix} \succcurlyeq 0 \tag{4.2}
$$

Chapter 5

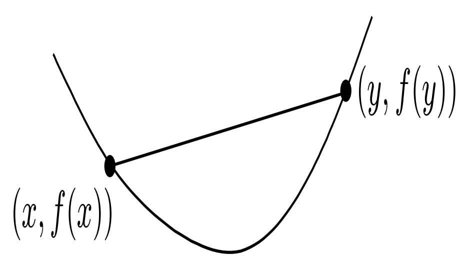
Introduction to Convex Optimization

5.1 Convex function

A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if **dom** f is a convex set and if for all x and $y \in$ **dom** f and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1 - \theta f(y)) \tag{5.1}$$

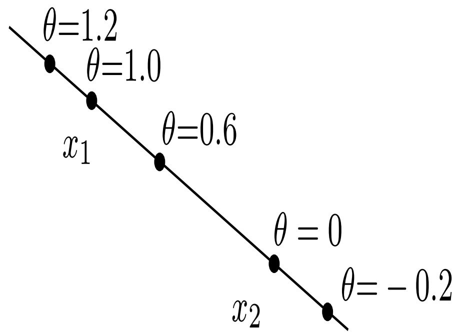
geometrically, this inequality means that the line segment between $(x, f(x))$ and $(y, f(y))$, which is a chord from x and y lies above the graph of f. A function f is called strictly convex if strict inequality holds in (5.1) whenever $x \neq y$ and $0 \leq \theta \leq 1$. We say f is concave if $-f$ is convex, and strictly concave, and strictly concave if $-f$ is strictly convex.



For an affine function we always have equality in (5.1), so all affine (therefore also linear) functions are both convex and concave. Conversely, any function that is convex and concave is also affine.

5.1.1 Affine Sets

A set $C \subseteq \mathbb{R}^n$ is affine if the line through any two distinct points in C lies in C, i.e. if for any $x_1, x_2 \in C$ and $\theta \in \mathbb{R}$, we have $\theta x_1 + (1-\theta)x_2 \in C$. In other words C contains the linear combination of any points in C, provided the coefficients in the linear combination sum to one.



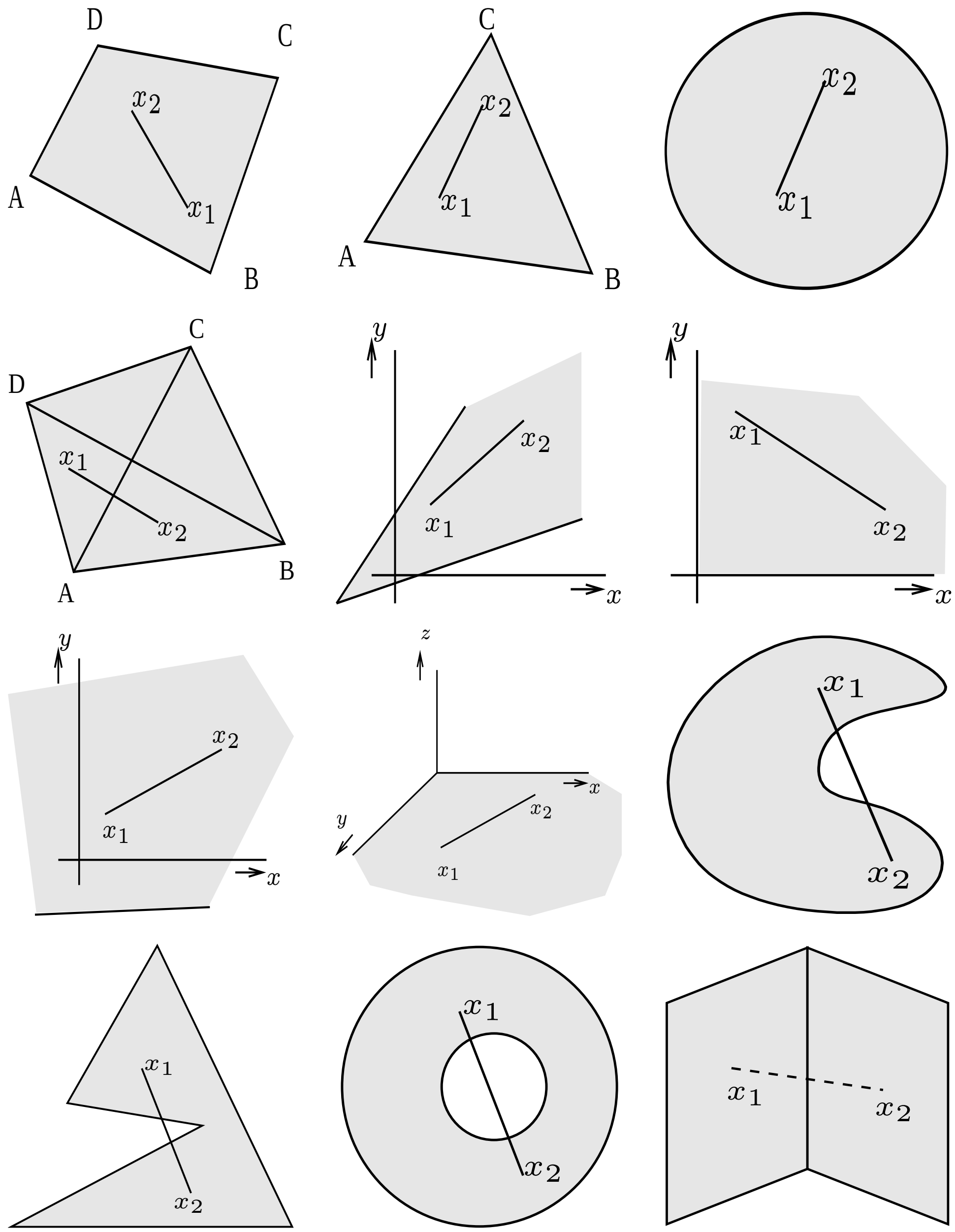
How to Check if a function is convex?

A twice-differentiable function of many variables is concave if and only if second derivative is non-positive everywhere. similarly
A twice-differentiable function of many variables is convex if and only if second derivative is non-negative everywhere.

For $f(\cdot)$ is a twice differentiable function of n-variables. The Hessian of $f(\cdot)$ at a vector x is

$$H(x) = \begin{bmatrix} f''_{11}(x) & f''_{12}(x) & \cdots & f''_{1n}(x) \\ f''_{21}(x) & f''_{22}(x) & \cdots & f''_{2n}(x) \\ \vdots & \vdots & \cdots & \vdots \\ f''_{n1}(x) & f''_{n2}(x) & \cdots & f''_{nn}(x) \end{bmatrix} \tag{5.2}$$

- $f(x)$ is concave if and only if $H(x)$ is negative semidefinite for all $x \in \mathcal{S}$.
- if $H(x)$ is negative definite for all $x \in \mathcal{S}$ then $f(\cdot)$ is strictly concave.
- $f(x)$ is convex if an only if $H(x)$ is positive semidefinite for all $x\ in \mathcal{S}$.
- if $H(x)$ is positive definite for all $x \in \mathcal{S}$ then f is strictly convex.



Why Convex functions are so important

All linear and affine functions are convex (and concave). Quadratic functions can also we convex (or concave). Some more operations which yield in convex functions are

- Exponential $e^{(ax)}$ is convex on $\mathbb{R}$ for any $a \in \mathbb{R}$.
- Powers x^a is convex on $\mathbb{R}_{++}$ when $a \geq 1$ or $a \leq 0$ and concave $0 \leq a \leq 1$.
- power of absolute value $|x|^p$ for $p \geq 1$ is convex on $\mathbb{R}$.
- logarithm $\log x$ is concave on R_{++}.

5.1.2 Epigraph

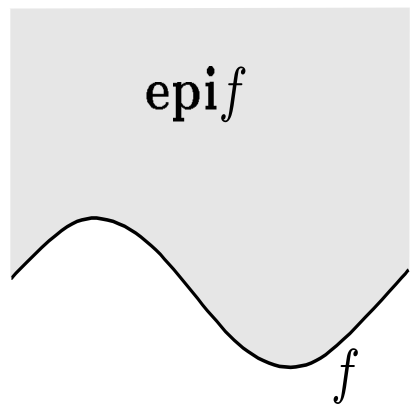
The graph of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as

$$\{(x, f(x))|x \in \textbf{dom } f\}$$

which is a subset of $\mathbb{R}^{n+1}$. The epigraph of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined as:

$$\textbf{epi } f = \{(x, t)|x \in \textbf{dom } f, \qquad f(x) \leq t\},$$

which is a subset of $\mathbb{R}^{n+1}$. 'Epi' means 'above' so epigraph means above the graph.



The link between convex sets and convex functions is via the epigraph: A convex function is convex only if epigraph is a convex set. A function is concave if any only if its *hypograph*, defined as

$$\textbf{hypo } f = \{(x, t)|t \leq f(x)\},$$

is a convex set.

5.2 Operations that preserve convexity

Non-negative weighted sums

Evidently if f is a convex function and $\alpha \geq 0$ then function αf is convex. If f_1 and f_2 are both convex functions then so is their sum $f_1 + f_2$. Combining non-negative scaling and addition, we see that the set of convex functions is itself a convex cone;

A non-negative weighted sum of convex functions

$$f = w_1 f_1 + \cdots + w_n f_n$$

is convex, similarly a non-negative weighted sum of concave functions is concave. These properties extend to infinite sums and integrals as well. For example if $f(x, y)$ is convex in x for each $y \in A$ and $w(y) \geq 0$ for each $y \in A$, then the function g defined as

$$g(x) = \int_A w(y) f(x, y) dy$$

is convex in x if such integral exists.

Composition with affine mapping

suppose $f : \mathbb{R}^n \to R$, $A \in \mathbb{R}^{n \times m}$, and $b \in \mathbb{R}^n$. Define $g : \mathbb{R}^m \to R$ by

$$g(x) = f(Ax + b)$$

with $\mathbf{dom}\ g = \{x | Ax + b \in \mathbf{dom}\ f\}$. Then if f is convex, so is g, if f is concave, so is g.

Composition

Consider $h : \mathbb{R}^k \to R$ and $g : \mathbb{R}^n \to \mathbb{R}^k$ that guarantee convexity or concavity of their composition $f = h \circ g : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$f(x) = h(g(x)), \qquad \mathbf{dom}\ f = \{x \in \mathbf{dom}\ g | g(x) \in \mathbf{dom}\ h\}$$

this implies:
f is convex if h is convex and non-decreasing, and g is convex,
f is convex if h is convex and non-increasing, and g is concave,
f is concave if h is concave and non-decreasing, and g is concave,
f is concave if h is concave and non-increasing, and g is convex,
Some further implications:
if g is convex then $\exp(g(x))$ is convex
if g is concave and positive, then $\log(g(x))$ is concave
if g is concave and positive, then $1/g(x)$ is convex
if g is convex and non-negative and $p \geq 1$, then $g(x)^p$ is convex.
if g is convex then $-\log(-g(x))$ is convex on $x | g(x) < 0$.

Perspective

if $f:\mathbb{R}^n \to R$, then the perspective of f is the function $g:\mathbb{R}^{n+1} \to R$ defined by

$$g(x,t) = tf(x/t)$$

with domain

$$\textbf{dom} g = \{(x,t)|x/t \in \textbf{dom} f, t > 0\}$$

The perspective operation preserves convexity: if f is a convex function, then so is its perspective function g. Similarly if f is concave , then so is g.

Chapter 6

Optimization Algorithms

6.1 Particle Swarm Optimization

6.2 Genetic Algorithm

Non-negative weighted sums

Evidently if f is a convex function and $\alpha \geq 0$ then function αf is convex. If f_1 and f_2 are both convex functions then so is their sum $f_1 + f_2$. Combining non-negative scaling and addition, we see that the set of convex functions is itself a convex cone; A non-negative weighted sum of convex functions

$$f = w_1 f_1 + \cdots + w_n f_n$$

is convex, similarly a non-negative weighted sum of concave functions is concave. These properties extend to infinite sums and integrals as well. For example if $f(x, y)$ is convex in x for each $y \in A$ and $w(y) \geq 0$ for each $y \in A$, then the function g defined as

$$g(x) = \int_A w(y) f(x, y) dy$$

is convex in x if such integral exists.

Composition with affine mapping

6.3 Ant Colony Optimization

if $f : \mathbb{R}^n \to R$, then the perspective of f is the function $g : \mathbb{R}^{n+1} \to R$ defined by

$$g(x, t) = tf(x/t)$$

with domain

$$\mathbf{dom} g = \{(x, t)|x/t \in \mathbf{dom} f, t > 0\}$$

The perspective operation preserves convexity: if f is a convex function, then so is its perspective function g. Similarly if f is concave , then so is g.