

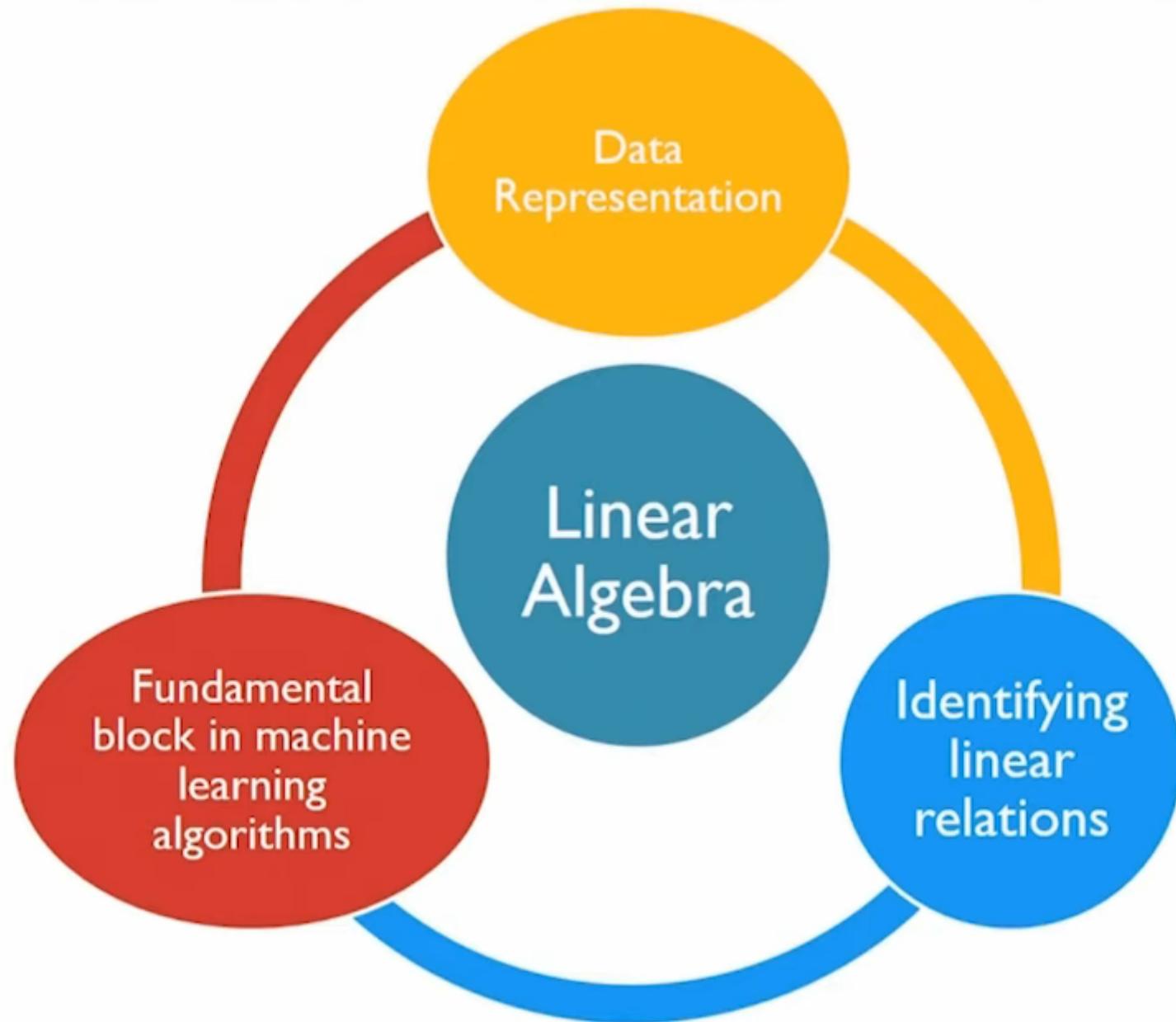
let's start with a question..

What linear algebra is useful for?

(When one talks about data science..)

1. **Data representation** becomes an important aspect.
 - Data is represented in a matrix form. *So, we talk about this representation and concepts.*
2. (*In the data science perspective..*) If the data contains several variables of interest, one might want to know **How many of these variables are really important?**, and **If there are any relationships, how can one uncover those?**
3. The ideas from linear algebra becomes fundamental concept in Machine Learning Algorithms.

these important questions are answered by the Linear Algebra.



Each of the above is addressed in further notes..

Let's start with the matrices.. and summarize the points which are relevant to the data science perspective **What is a matrix?**

- A matrix is a form of organizing data into rows and columns.

There are many ways of representing the data, matrix provides a convenient way..

- If you are an engineer, and you deal with multiple variables, how do you store them so that you can use later?
- Matrices can also be used to store the coefficients of the equations

Matrix

- Matrices can be used to represent samples

Data Representation: Examples

Data representation: Examples

- A real life example
 - Consider a reactor which needs to be controlled using multiple attributes from various sensors like Pressure (Pa), Temperature (K), Density (gm/m^3) etc.
 - Independently, the sensors have generated 1,000 data points
 - This complete set of information is contained in

	P	T	ρ
1	300	300	1000
:	:	:	:
1000	500	1000	5000

each row is a sample data.

A vector of attributes..

- Example 2:

$$X = [1,2,3]^T$$

$$Y = [2,4,6]^T$$

- X and Y are vectors pertaining to some attributes
- We define the A matrix using a column bind of X and Y thus representing data in a matrix format (the code for the same is attached)

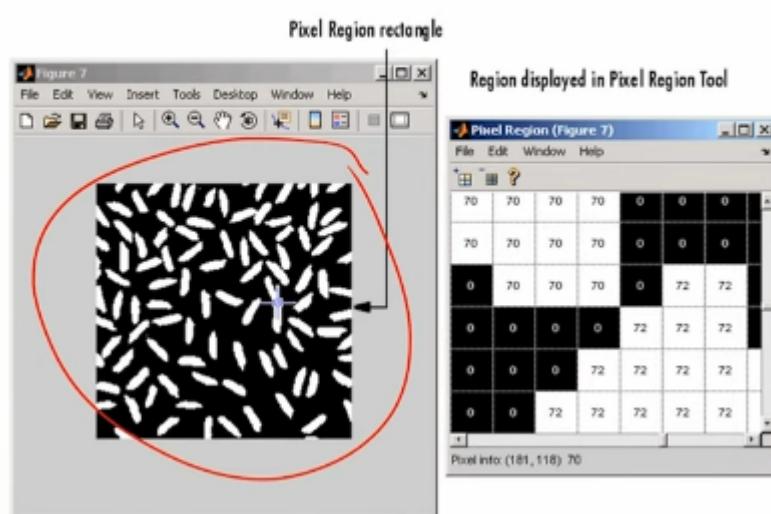
R Code
`x=c(1,2,3)
y=c(2,4,6)
A=cbind(x,y)
print(A)`

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{bmatrix}$$

Output
`> print(A)
x y
[1,] 1 2
[2,] 2 4
[3,] 3 6`

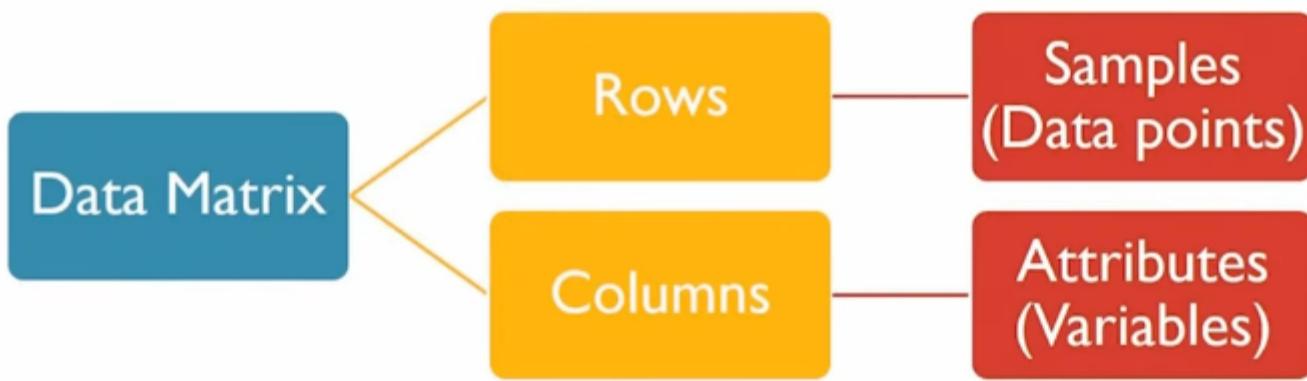
Storing images..

- The simplicity in representation will become apparent when the image below is considered



and answering, whether two images same or identify the sub-components in the image..... all of these can be performed through some form of matrix manipulation.

Data as matrix - Summary



now coming to the second aspect..

Identification of Independent attributes

- One might be interested in knowing, if all the variables that **are there are important?** (*in other words..*)
- In all these variables, **how many of these are independent?**, so that, if contains (say) 1000's of variables and there are 4-5 independent variables, then one can store these variables and calculate the remaining as a function of these variables.

These lead to the following questions..

- Now that we can represent the data into a matrix format, we ask the following questions
 - Are all the attributes in the data matrix relevant/ important?
 - Is there any method which can identify if some attributes are related to the other attributes?
 - If yes, how do we identify the linear relationship?
 - Can we use this to reduce the size of the data matrix?

If could find those independent variables, can drop the remaining. --- This is an important aspect in Machine Learning.

Identification of independent attributes: Example

- Consider the ideal reactor example with multiple (say, 4) attributes like Pressure, Temperature, Density, Viscosity, etc. with 500 samples.
- Thus we have a 500×4 matrix such that

$$A = [P \ T \ D \ \eta]$$
- P, T, D and η are vectors of 500 samples from the pressure, temperature, density and viscosity sensors.
- How does one identify the number of independent attributes?

Identification of independent attributes: Example

- Domain knowledge

$$D \sim f(P, T)$$

- Thus, in some sense **D** is a function of **P** and **T**

How can one identify the independent attributes? - Rank of matrix

- Let us assume that we have many more samples than attributes for now
- Is there any approach which can be used to identify the number of linear relationships between the attributes purely using data?
- This is addressed by the concept of the **rank** of the matrix.
- **Rank** of a matrix refers to the number of linearly independent rows or columns of the matrix
- The rank of a matrix can be found using the rank command: rank(A)

Rank of matrix:

Number of linearly independent rows or columns in a matrix. Denoted as `rank(A)`

See, the below matrix, its deliberately generated in such a way that, column-2 depends on column-1.

- Consider another example

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 0 \\ 3 & 6 & 0 \end{bmatrix}$$

- We observe that

- (Col. 2)=2 x (Col. 1)
- (Col. 3) is independent

- Thus, the rank of this matrix is 2

R Code
A=matrix(c(1,2,3,2,4,6,1,0,0),ncol=3,byrow=F)
library(pracma)
Rank(A)

Output
> Rank(A)
[1] 2

Rank: Advantages and summary

Rank

Work with reduced set of variables

Dependent attributes can be calculated if they are from the same data generation process

Independent of the size of the dataset if the data is from the same data generation process

ok, now we come to know the **no. of independent variables**, but **how about the relationship with the dependent variables...??** --this further motivates as ...

Identification of Linear relationships among attributes - via concept of **Null Space** and **Nullity**

NullSpace of a matrix

The null space of a matrix A consists of all vectors β , such that $A\beta = 0$ and $\beta \neq 0$. Then β is called the null space of the matrix.

What does $A\beta = 0$ mean? It's addressed in the next section formally. (intuitively explained in the below slide picture)..

Interestingly, **the size of the null space of a matrix provides us the no. of linear relations among the attributes.**

i.e., if a matrix having dimensions 5, then if we got a null space matrix's size as 2, then it implies that 5 - 2 = 3 are linearly independent. (The other two depend on these, so got nullified).

Nullity of a matrix

Number of vectors in the null space of the given matrix -- (from understanding [Please correct if not]..it tells **no. of dependent variables in the matrix**)

- The null space of a matrix A consists of all vectors β such that $A\beta = 0$ and $\beta \neq 0$
- Nullity of a matrix is the number of vectors in the null space of the given matrix
- The size of the null space of a matrix provides us with the number of linear relations among the attributes
- And the null space vectors β are useful to identify these linear relationships

$$A_{3 \times 3} \beta_{3 \times 1} = \begin{matrix} (3 \times 1) \\ \downarrow 0 \end{matrix} \quad \text{S } \textcircled{2}$$



NullSpace: General Description

- Let us suppose

• $A = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$ is a data matrix and there is one vector in the null space of A , i.e., $\beta = [\beta_1 \dots \beta_n]^T$, then as per the definition, β satisfies all the equations given below

- $x_{11}\beta_1 + x_{12}\beta_2 + \cdots + x_{1n}\beta_n = 0$
• $x_{11}\beta_1 + x_{12}\beta_2 + \cdots + x_{1n}\beta_n = 0$
⋮
• $x_{m1}\beta_1 + x_{m2}\beta_2 + \cdots + x_{mn}\beta_n = 0$



those equations denote the $A\beta = 0$.. hope not got clear..!! (Referred from the dobut in above cell)

- and the generic form is written above all the equations in the red-ink. It can be interpreted as..

Where you can take any sample and substitute the values of variables in that x_1, x_2, \dots, x_n , and this is to be satisfied, its a true relationship.

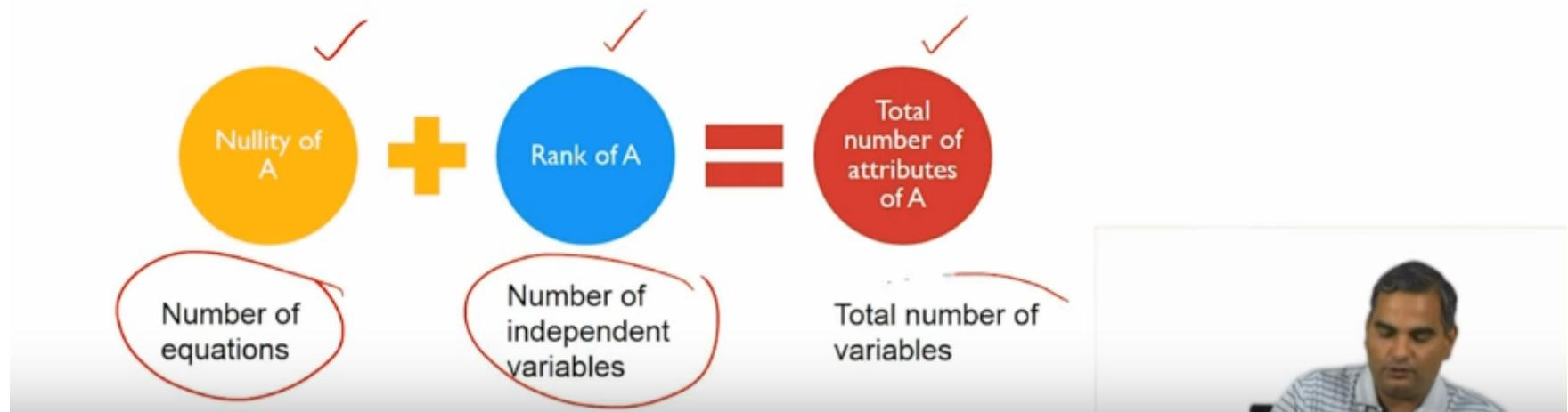
Null space: The idea

- Notice that if $A\beta = \mathbf{0}$, every row of A when multiplied by β goes to zero
- This implies that variable values in each sample (represented by a row) behave the same
- This helps in identifying the linear relationships in the attributes
- Every null space vector corresponds to one linear

It can be demonstrated as..

Rank nullity theorem

- Consider the data matrix A with the null space and nullity as defined before
- The rank-nullity theorem helps us to relate the nullity of the data matrix to the rank and the number of attributes in the data
- According to the rank-nullity theorem

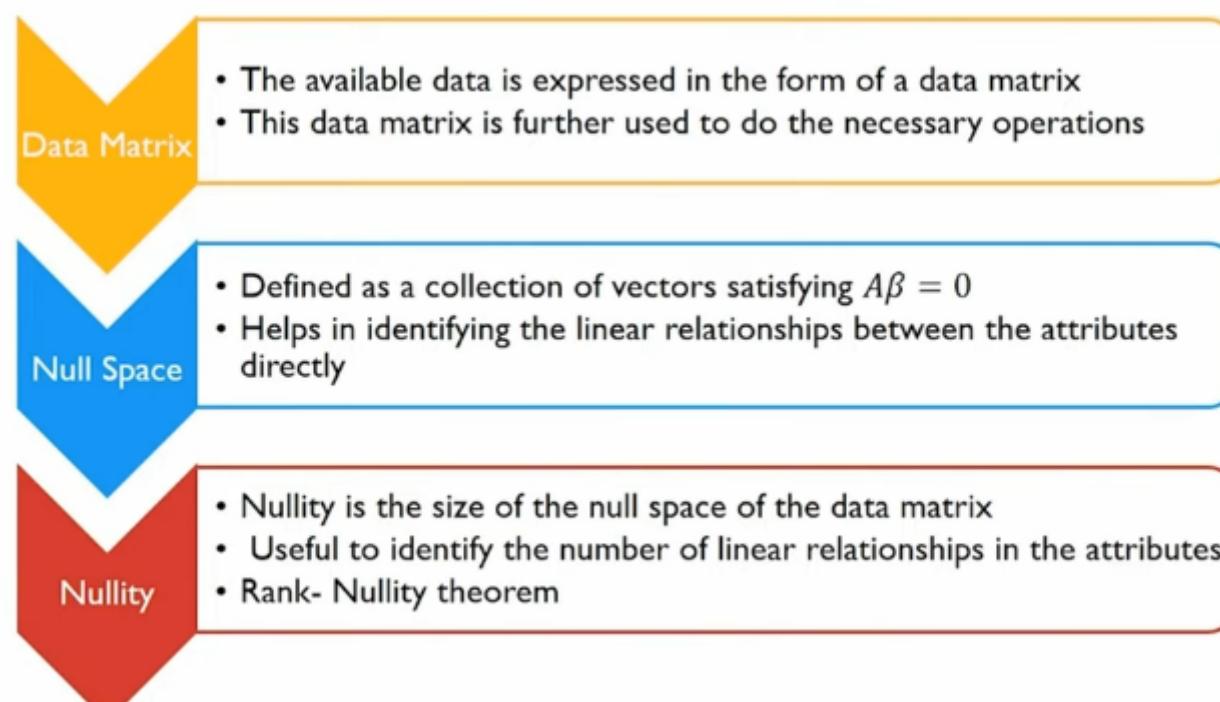


Why "Nullity of A" means "Number of equations"? Think..!!

What does Nullity of A tells? (from understanding, please correct if not) it tells the no of dependent variables. So, how can we get those back after dropped, via independent variables? -- via equation right..???. So, if had n no. of dependent variables, then we need same n no. of equations to get those right..??

NOTE: -- its purely based on understanding, by grace. Correct if found to be incorrect.

Summary



Null space: An Example

- Consider the matrix A with attributes $\{x_1, x_2\}$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Number of columns in A = 2

Rank of A = 2 ✓

Thus, nullity = 0 ✓

- This implies that the null space of the matrix A does not contain any vectors

- Thus we can claim that all the attributes are linearly independent

Console output

```
> A
[,1] [,2]
[1,] 1 2
[2,] 3 4
[3,] 5 6
```

Console output

```
> print(columns) ✓
[1] 2
> print(rank) ✓
[1] 2
> print(nullity) ✓
[1] 0
```

R Code

```
A=matrix(c(1,3,5,2,4,6),ncol=2,byrow=F)
columns=ncol(A)
library(pracma)
rank=Rank(A) ✓
nullity=columns-rank
```

Linear Algebra



Null space: Another example

- Now consider A with attributes $\{x_1, x_2, x_3\}$ such that

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 3 & 6 & 1 \end{bmatrix}$$

Number of columns in A = 3

Rank of A = 2

Thus, nullity = 1

R Code

```
A=matrix(c(1,2,3,2,4,6,0,0,1),ncol=3, byrow=F)
columns=ncol(A)
library(pracma)
rank=Rank(A)
nullity=columns-rank
```

Console output

```
> columns
[1] 3
> rank
[1] 2
> nullity
[1] 1
```



Null space: Further Example

$$A\beta = 0$$

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 0 \\ 3 & 6 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$A\beta = 0$$

$$C\beta$$

$$A^c \beta = 0$$

$$C \circ A \beta = 0$$

- Thus we obtain,

$$\begin{aligned} b_1 + 2b_2 &= 0 \\ b_3 &= 0 \end{aligned}$$

$$b_1 = -2b_2$$

- The null vector is $B = [b_1 \ b_2 \ b_3]^T = [-2b_2 \ b_2 \ 0]^T = k[-2 \ 1 \ 0]^T$

- We see that we obtain a direct linear relationship between the attributes of A using null space and rank-nullity theorem

$$-2n_1 + n_2 = 0$$

- The same concept can be extended for bigger data set

-- Need workout on this.

- R1's clear..
- The 2nd row's equation was dropped, as its the multiple of R1.
- R3's, how about 3 and 6..??

Overall Summary

Overall summary



- Represent data in a matrix form with rows and columns representing samples and attributes respectively
- Represent coefficients in several equations in a matrix form

- Number of independent variables or samples

- Identifies the number of linear relationships (if any)

- Null space vectors provide the linear relationships

End of Module-1

Module-2: Solving Linear Equations

-- on 24th August, 2021 ~ Tuesday_4

Recap

- We have established the importance and usefulness of matrix theory and linear algebra in data sciences
- Concepts covered previously
 - Data representation using matrices
 - Identifying linear relationships (if any) among attributes
- How do we establish these linear relationships?
 - Using null space
 - We will now focus on extracting solutions for matrix equations

Recap..

Solving Matrix Equations

Recollect..!! from the M-I of Diploma Math, Smt. Gomati Madam has taught the subject..

Preliminaries

- We consider the following set of equations

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

$$\mathbf{A}(m \times n); \mathbf{x}(n \times 1); \mathbf{b}(m \times 1)$$

- Generalized linear equations can be represented in the above format.
- m and n are the number of equations and variables respectively.
- \mathbf{b} is the general RHS commonly used

Categorization..

Categorization

$m = n$

- Number of equations and variables are the same
- Easiest case to solve

$m > n$

- More equations than variables
- Usually no solution

$m < n$

- Number of equations less than number of variables
- Usually multiple solutions

A short note on Ranks.. (as they are going to be used here..)

Full row and column rank: Concepts

- Consider a matrix data matrix A ($m \times n$)

Full Row Rank

- When all the rows of the matrix are linearly independent
- Data sampling does not present a linear relationship – samples are independent

Full Column Rank

- When all the columns of the matrix are linearly independent
- Attributes are linearly independent

Row rank = Column rank

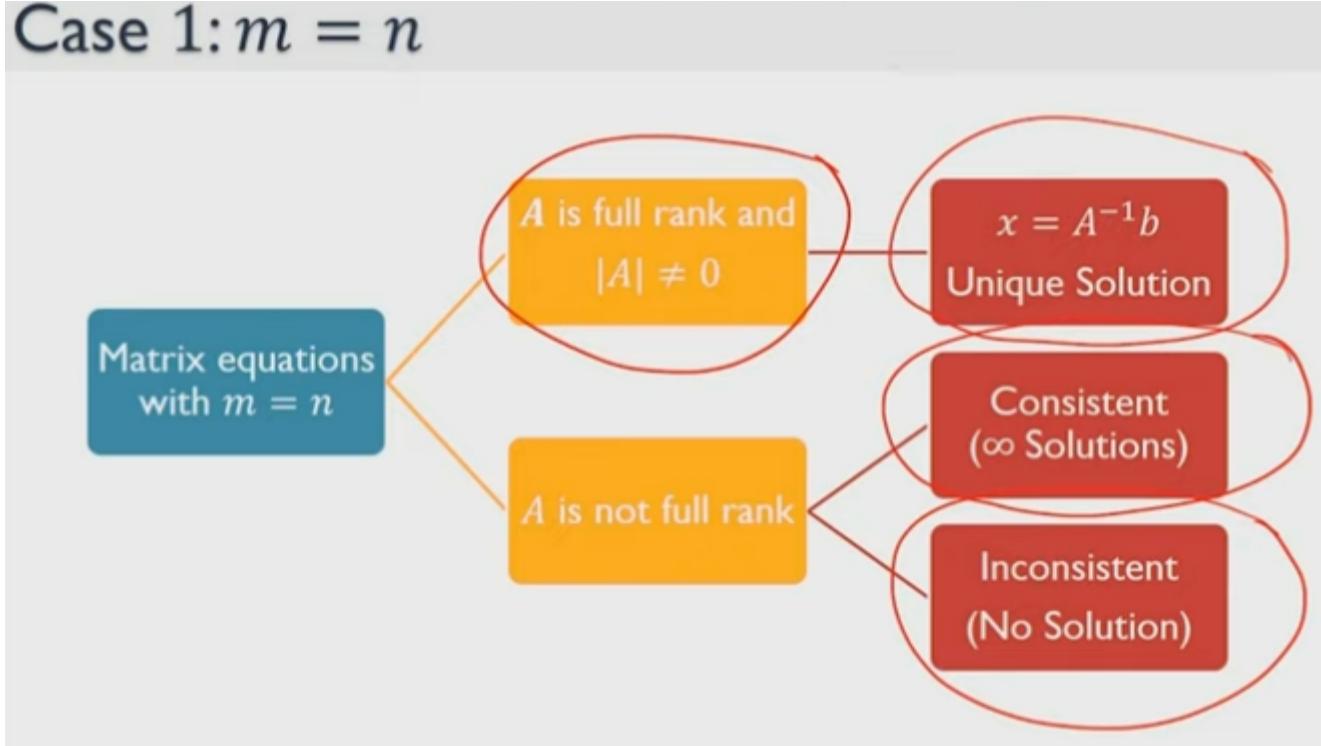
- Generally, if had a matrix $A_{m \times n}$, then... maximum rank value can be atmost m if $m < n$, else n if $n < m$ -- i.e., the smallest value in dimensions of the matrix, is the maximum value, a rank can be.
- Generally, whatever may be the size of the matrix, **Row Rank = column Rank** -- indicates that, if had certain no. of independent rows, then we have same no. of independent no. of cols.

P2N: rank tells the no. of independent cols (or rows)

Case-1: $m==n$

- If A is full rank (Now the max rank can be either m or n, as $m==n$), then
 - We have all the rows or cols independent of each other (*in other words*), we don't get any row or column such that [On LHS], its being linearly dependent on other. 00 then we have **Unique solution**, its $x = A^{-1}b$ -- How..?? as the equation is of the form $Ax = b$, and we need x , so by simply transposing A to R.H.S. ... we get that..
- The difficulty arises here.. think why the rank is less... on L.H.S. atleast one row or col, linearly depends on other. In this case, depending on what value we have on the R.H.S. we have 2 cases..
 - One is **Consistent** situation -- Will have many solutions..
 - Second is **Inconsistent** - no solution

Case 1: $m = n$



Example: With Full rank. -- demonstrating the *Unique solution*

Case 1: Example 1.1

$$A \mathbf{x} = \mathbf{b} \quad \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 10 \end{bmatrix}$$

R Code
 $A=\text{matrix}(c(1,2,3,4),\text{ncol}=2, \text{byrow}=F)$
 $b=c(7,10)$
 $x=\text{solve}(A)\%*\%b$

$$|A| \neq 0 \\ \text{rank}(A) = 2 = \text{no. of columns}$$

- This implies that A is full rank

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 7 \\ 10 \end{bmatrix} = \begin{bmatrix} -2 & 1.5 \\ 1 & -0.5 \end{bmatrix} \begin{bmatrix} 7 \\ 10 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- Thus, the solution for the given example is $(x_1, x_2) = (1, 2)$

Console output
 $> x$
 $[,1]$
 $[1,] 1$
 $[2,] 2$

--- P2N: No

other solution, can satisfy the equation-- as it is the unique solution.

Example: With $<$ full rank... -- demonstrating the *Consistent solution*

- Notice that, they are linearly dependent -- both column-wise and row-wise. -- its even said by *rank* and *determinant being=0*.
- We generally consider the LHS.

Case 1: Example 1.2

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

$$|A| = 0; \text{rank}(A) = 1; \text{nullity} = 1$$

- Checking consistency

$$\begin{bmatrix} x_1 + 2x_2 \\ 2x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

$\text{Row}(2) = 2\text{Row}(1)$

$$x_1 + 2x_2 = 5$$

$$\begin{cases} x_2 = 0, & x_1 = 5 \\ x_2 = 1, & x_1 = 3 \end{cases}$$

- The equations are consistent with only one linearly independent equation
- The solution set for (x_1, x_2) is infinite because we have only one linearly independent equation and 2 variables

- Also notice that, even the R.H.S. is also linearly dependent. It means that, we can calculate the one and get the other. **NOTICE** that, we get the freedom to choose x_1 or x_2 (satisfying the left-over equation ofcourse..).

What is meant by having infinite no. of solutions?

See that, We have infinite choice of choosing the x_1 or x_2 (one keeping fixed) and all those infinite choices are valid for the equation. Hence, we get the infinite no. of solutions.

Example: with $\text{rank } A < n$ -- demonstrating the *Inconsistent solutions*.

- Its same as above example, with a change that, RHS is not linearly dependent -- *Still the L.H.S. is linearly dependent*

Case 1: Example 1.3

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$$

$$|A| = 0$$

$$\text{rank}(A) = 1$$

$$\text{nullity} = 1$$

- Checking consistency

$$\begin{bmatrix} x_1 + 2x_2 \\ 2x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$$

Case-2: $m > n$

Case 2: $m > n$

- This is the case of not enough variables or attributes
- Since the number of equations is greater than the number of variables, in general, not all equations can be satisfied
- Hence it is sometimes termed as a no-solution case
- However, we can identify an appropriate solution by viewing this case from an optimization perspective

- Even though some times, its termed as no-solutions case -- from an optimization perspective can try..

- Instead of identifying a solution to $Ax - b = 0$, one can identify an x such that $(Ax - b)$ is minimized
- Notice that $(Ax - b)$ is a vector
- There will be as many error terms as the number of equations

means..

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

- When had

perfect solutions for all the three equations (i., the values for which x_1, x_2, x_3 satisfies).

$$a_{11}x_1 + a_{12}x_2 - b_1 = 0$$

Then can write as

- But, whenever we don't have the perfect solution, let's call it error. And we allowed it, so we get 3 (or as many depending on equations)

collectively by minimizing $\sum_{i=1}^3 e_i^2$ (e_i)

$$(Ax - b)^T (Ax - b)$$

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

different errors right.. It's represented as

- in slide represented as

How do we minimize the errors collectively?

- One might say that **sum up all those**, but that won't be a good idea as, e_1 could be a vey big error in +ve direction, and e_2 could be a -ve direction which may sum up to 0, and e_3 --- that's not a good solution at all.
- One way to do is, **Collectively minimize the error**

- One could minimize all the errors collectively by minimizing $\sum_{i=1}^m e_i^2$

i.e., Instead of going $e_1 + e_2 + e_3$, can go as $e_1^2 + e_2^2 + e_3^2$ --- As long as these (e_1, e_2, e_3) are away from 0, the contribution to error is high. (as squaring right..!!) -- that ensures that, you do not go far away from 0. -- this is the **Least Square Solution**.

- Can also try as $|e_1| + |e_2| + |e_3|$ (as $|$ always gives +ve) -- that's also possible, but in general we consider this method.

$$(e_1, e_2, e_3)^T \cdot (e_1, e_2, e_3) = e_1^2 + e_2^2 + e_3^2$$

- Notice that, one can write this $(e_1^2 + e_2^2 + e_3^2)$ also as $e^T e$ which is equivalent to

Case 2: An optimization perspective

- Instead of identifying a solution to $Ax - b = 0$, one can identify an x such that $(Ax - b)$ is minimized
- Notice that $(Ax - b)$ is a vector
- There will be as many error terms as the number of equations
- Denote $(Ax - b) = e (mx1)$; there are m errors $e_i, i = 1:m$
- One could minimize all the errors collectively by minimizing $\sum_{i=1}^m e_i^2$
- This is the same as minimizing $(Ax - b)^T (Ax - b)$

$$(e_1, e_2, e_3)^T \cdot (e_1, e_2, e_3) = e_1^2 + e_2^2 + e_3^2$$

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2(e_2)$$

$$a_{31}x_1 + a_{32}x_2 = b_3(e_3)$$

To put it together..



- This optimization problem is

$$\begin{aligned} & \min[(Ax - b)^T (Ax - b)] \\ &= \min[(b^T - x^T A^T)(Ax - b)] \\ &= \min[(x^T A^T Ax - 2b^T Ax + b^T b)] = f(x) \end{aligned}$$

- We observe that the optimization problem is a function of x
- Solving the optimization problem will result in a solution for x
- The solution to this optimization problem is obtained by differentiating $f(x)$ with respect to x and setting the differential to zero

$$\nabla f(x) = 0$$

-- The top-centered

final equation is obtained after some algebraic manipulation -- that's what sir, said about it. You please work out on it.

- Differentiating $f(x)$ and setting the differential to zero results in

$$\begin{aligned} & 2(A^T A)x - 2A^T b = 0 \\ & (A^T A)x = A^T b \end{aligned}$$

- Assuming that all the columns are linearly independent

$$x = (A^T A)^{-1} A^T b$$

-- the final solution x , might not satisfy

for all the values of equations, it ensures that, error is collectively minimized.

Concludes the lecture, in the next lecture..

- An ex. to illustrate how these are satisfied or not satisfied. Next case-3 where $m > n$.
- Next, how can these be combined into one elegant solution.

Module-3

--- on 2nd September, 2021 ~ Thursday (Ekadashi)

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

• $m = 3, n = 2$

- You can observe that the first 2 equations are inconsistent. i.e., $1x_1 = 1 \dots \#1$ and $2x_2 = -0.5 \dots \#2$, think here.. by what value can you satisfy these two combinely? -- that isn't possible, hence its inconsistent.
- Coming to the $3x_1 + x_2 = 5 \dots \#3$, by fixing x_1 , can get x_2 .

On the whole, this is not solvable.

Then, how do we solve these..?? --- via **Optimization concept** : $x = (A^T A)^{-1} A^T b$ as discussed in previous lecture (Refer to the previous

Case 2: Example - I

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

• $m = 3, n = 2$

- Using the optimization concept,

$$x = (A^T A)^{-1} A^T b$$
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

(lecture notes)

Case 2: Example continued

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.2 & -0.6 \\ -0.6 & 2.8 \end{bmatrix} \begin{bmatrix} 15 \\ 5 \end{bmatrix}$$
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

- Thus, the solution for the given example is $(x_1, x_2) = (0, 5)$
- Substituting in the equation shows

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \neq \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

Workout part...

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

$$A \cdot x = b$$

Optimization concept $x = (A^T \cdot A)^{-1} A^T \cdot b$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \\ 5 \end{bmatrix}$$

$$\frac{1}{1 \times 1 + 2 \times 2 + 3 \times 3} = \frac{1}{14}$$

$$\frac{1}{1 \times 1 + 2 \times (0.5) + 3 \times 1} = \frac{1}{5}$$

$$\begin{bmatrix} 14 & 3 \\ 3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 15 \\ 5 \end{bmatrix}$$

$$A^{-1} = \frac{\text{adj} A}{|A|}$$

$$\text{adj} A \text{ for } 14_{(1,1)} = 1 \cdot (-1)$$

$$3_{(1,2)} = 3 \cdot (-1)$$

$$3_{(2,1)} = 3 \cdot (1)$$

$$1_{(2,2)} = 14 \cdot (1)$$

$$|A| = 14 \cdot 1 - 9$$

$$= 5$$

$$\frac{\text{adj} A}{|A|} = \frac{\begin{bmatrix} 1 & -3 \\ -3 & 14 \end{bmatrix}}{5} \Rightarrow \begin{bmatrix} 1/5 & -3/5 \\ -3/5 & 14/5 \end{bmatrix} \Rightarrow$$

$$1/5 \quad -3/5 \quad -0.6 \quad 15 \quad | \quad \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Case 2: Example

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

- $m = 3, n = 2$
- Using the optimization concept,

$$x = (A^T \cdot A)^{-1} A^T \cdot b$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

Case 2: Example continued

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.2 & -0.6 \\ -0.6 & 2.8 \end{bmatrix} \begin{bmatrix} 20 \\ 5 \end{bmatrix}$$
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- Thus, the solution for the given example is $(x_1, x_2) = (1, 2)$
- Substituting in the equation shows

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}$$

R Code

```
A=matrix(c(1,2,3,0,0,1),ncol=2, byrow=F)
b=matrix(c(1,2,5),ncol=2, byrow=F)
x=inv(t(A)%%A)%%t(A)%%b
x
```

Console output

```
> x=inv(t(A)%%A)%%t(A)%%b
> x
[1,] 1
[2,] 2
>
```



Observe that # equations > # variables

In [5]:

```
# Equations.  
A = matrix(c(1, 2, 3, 0, 0, 1), ncol=2, byrow=F)  
b = matrix(c(1, 2, 5), ncol=1, byrow=F)  
A  
b
```

```
1 0  
2 0  
3 1  
  
1  
2  
5
```

In [10]:

```
install.packages("pracma")
```

```
Updating HTML index of packages in '.Library'  
Making 'packages.html' ... done
```

In [11]:

```
# Solving the equations, via least square method..  
x = inv(t(A) %*% A) %*% t(A) %*% b  
x
```

```
Error in inv(t(A) %*% A): could not find function "inv"  
Traceback:
```

case -3 : Where no. equations(m) < no. of variables(n)

---- i.e., rows(m) < cols(n)

- This case addresses the problem of more attributes or variables than equations
- Since the number of attributes is greater than the number of equations, one can obtain multiple solutions for the attributes
- This is termed as an infinite-solution case

How to think this or understand this..??

Consider 2 equations ($m=2$) with 3 variables ($n=3$), this can be thought as...

* **You could choose any value for x_3 , plug in the 2 equations,, take them to R.H.S.*(as now they became constants)* and solve for rest x_1 \$ and x_2**
* Here **one can choose infinite range of values for x_3**, Hence *Inifinite solutions*..
-----making a clear sense right.. think for a while..$$

- How does one choose a single solution from the set of infinite possible solutions?

Now...if had infinite solutions....

There is no way to distinguish between these infinite possible solutions, we need to have a metric -- with which we can pickup a solution.

Case 3: $m < n$

- This case addresses the problem of more attributes or variables than equations
- Since the number of attributes is greater than the number of equations, one can obtain multiple solutions for the attributes
- This is termed as an infinite-solution case
- How does one choose a single solution from the set of infinite possible solutions?

- Pose the following optimization problem

$$\min \left(\frac{1}{2} x^T x \right) \text{ s.t. } Ax = b$$

An optimization perspective.. (like the previous case...)

- Notice the constraint: $Ax = b$ (**s.t.**: subject-to). On an whole tells that: **Need to minimize $\frac{1}{2}x^T x$ subject to the constraint $Ax = b$.**
 - In other words.... Whatever solution we get for x , has to satisfy the $Ax = b$. This is not the problem, as we have infinite no. of solutions. and what the $\min \left(\frac{1}{2}x^T x \right)$ doing is... **How do I pick that one solution which minimizes the $x^T x$**

1/2 is to make sure that, solution comes out in a nice form.

Engineering Perspective: Say you have many parameters for manufacturing, and you need only of the smallest cost/size...

Unlike previous one, where there is no constraint, so its called **Unconstrained Optimization**, here as there exists a constraint, its **Constrained Optimization**.

Case 3: An optimization perspective

- Pose the following optimization problem

$$\min \left(\frac{1}{2}x^T x \right) \text{s.t. } Ax = b$$

- Define a Lagrangian function $f(x, \lambda)$

$$\min \left[f(x, \lambda) = \frac{1}{2}x^T x + \lambda^T (Ax - b) \right]$$

- Differentiating the Lagrangian with respect to x , and setting to zero

$$x + A^T \lambda = 0$$



Case 3: An optimization perspective

$$x = -A^T \lambda$$

Pre-multiplying by A
 $Ax = b = -AA^T \lambda$

Thus we obtain $\lambda = -(AA^T)^{-1}b$ assuming that all the rows are linearly independent

$$x = -A^T \lambda = A^T (AA^T)^{-1} b$$



Case 3: Example

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- $m = 2, n = 3$
- Using the optimization concept,

$$x = A^T (AA^T)^{-1} b$$

$$x = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Example..

Case 3: Example

$$x = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 14 & 3 \\ 3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$
$$x = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} -0.2 \\ 1.6 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -0.2 \\ -0.4 \\ 1 \end{bmatrix}$$

- The solution for the given example is $(x_1, x_2, x_3) = (-0.2, -0.4, 1)$



R Code
A=matrix(c(1,0,2,0,3,1),ncol=3)
b=c(2,1)
library(MASS)
x = t(A) %*% inv(A %*% t(A)) %*% b
x

Console output
A=matrix(c(1,0,2,0,3,1),ncol=3, byrow=F)
b=c(2,1)
x = t(A) %*% inv(A %*% t(A)) %*% b
x



Check...

Case 3: Example

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- The solution for the given example is $(x_1, x_2, x_3) = (-0.2, -0.4, 1)$
- Verify this is a solution that satisfies the original equation
- This also turns out to be minimum norm solution

-- Minimum Norm Solution

Workout...

Case 3

Example

$$m=2 \quad \begin{matrix} n=3 \\ \downarrow \end{matrix} \quad \left[\begin{array}{ccc} 1 & 2 & 3 \\ 0 & 1 & 1 \end{array} \right] \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

VIA optimization concept...

$$x = A^T (A \cdot A^T)^{-1} b$$

$m < n$

equations < # variables

by substitution method

$$x_1 + 2x_2 + 3x_3 = 2 \quad \text{--- } \#1$$

$$x_3 = 1$$

$$\text{#2) in #1.} \quad x_1 + 2x_2 + \frac{3(1)}{-1} = 2 - 3$$

$$x_1 + 2x_2 = -1$$

In [15]:

```
A = matrix(c(0, 1, 0, 2, 0, 3, 1), ncol=3, byrow=F)
b = c(2, 1)
x = t(A) %*% inv(A %*% t(A)) %*% b
```

Error in inv(A %*% t(A)): could not find function "inv"
Traceback:

Till now, we came across, different methods for different case right.., Is there any generalized method, with which one can solve all the cases (whether square [$m=n$] or rectangle [$m \neq n$: $m < n$ or $m > n$]) --- Yes.. its **Moore-Penrose pseudo-inverse matrix**

Generalization

- The described cases cover all the scenarios one might encounter while solving linear equations
- Is there any form in which the results obtained for cases 1, 2 and 3 can be generalized ?
- The concept we used to generalize the solutions is called as Moore-Penrose pseudo-inverse of a matrix
- The pseudo inverse is used as follows

$$Ax = b$$

$$x = A^{-1}b$$

The solution becomes

$$x = A^+b$$

- Singular Value Decomposition can be used to calculate the pseudo inverse or the generalized inverse (A^+)

-- here $x = A^{-1}b$ is the solution

which we were used to do in previous cases, now this way...

- There are many ways to compute pseudo-inverse, SVD-Singular Value Decomposition is one of the method.

In [20]:

```
A = matrix(c(1, 2, 3, 0, 0, 1), ncol=2, byrow=F)
b = matrix(c(1, 2, 5), ncol=1, byrow=F)
A
b
```

```
1 0
2 0
3 1
1
2
5
```

In [21]:

```
# Calculate the pseudo-inverse of the matrix.. which is of more variables than equations ... m<n case..
library(MASS)
x = ginv(A) %*% b # Here 'g' in "ginv" stands for ""Generalization"""
x
```

In [27]:

```
A = matrix( c(1, 0, 2, 0, 3, 1), ncol=3, byrow=F)
b = c(2, 1)
A
b
```

```
1 2 3
0 0 1
```

```
1.2
2.1
```

In [28]:

```
# Calculate the pseudo-inverse of the matrix.. which is of more equations than variables ...m>n case..
library(MASS)
x = ginv(A) %*% b
x
```

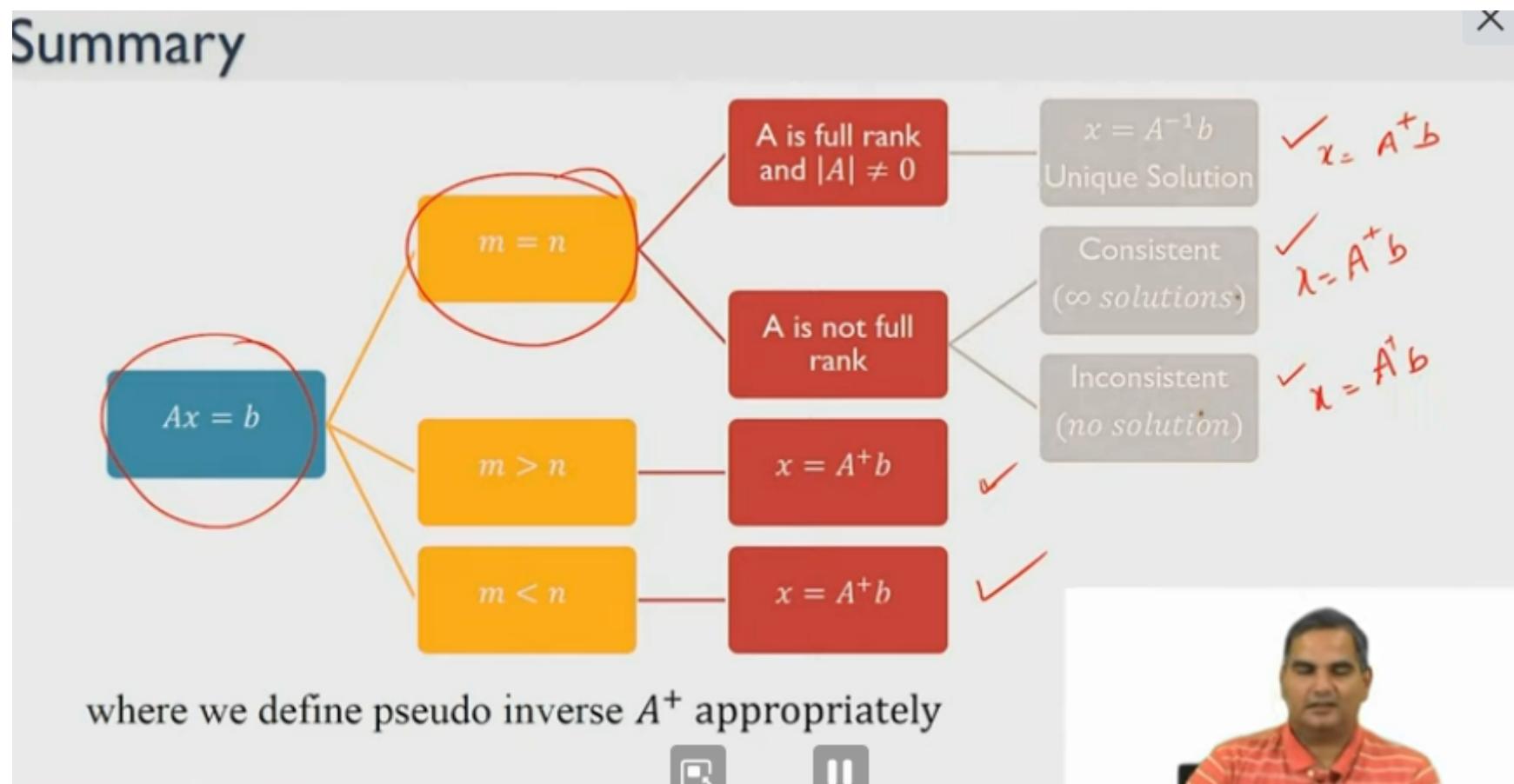
```
-0.2
-0.4
1.0
```

The values of x can be calculated via these... **How to interpret those....?**

That's what done in the lecture..

- For the case of $m > n$, its the solution: **Least Square Solution** (which minimizes the error collectively) i.e., minimizes $e_1^2 + e_2^2 + e_3^2$
- For the case of $m < n$, --- Minimum norm solution, this is the solution that is closest to the origin when had infinite no. of solutions.

Summary



So, without worrying about the square or rectangular system and not worrying about, whether the columns are dependent or independent, we can use generalized solution as one unify concept to find the solution to all the cases.

Coming up next...

- Geometrical View of these..