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

SOLVING MATRIX EQUATIONS

Preliminaries

- We consider the following set of equations

$$\boldsymbol{Ax} = \boldsymbol{b}$$

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

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

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

We look into these cases independently

Full row and column rank: Concepts

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

Full Row Rank

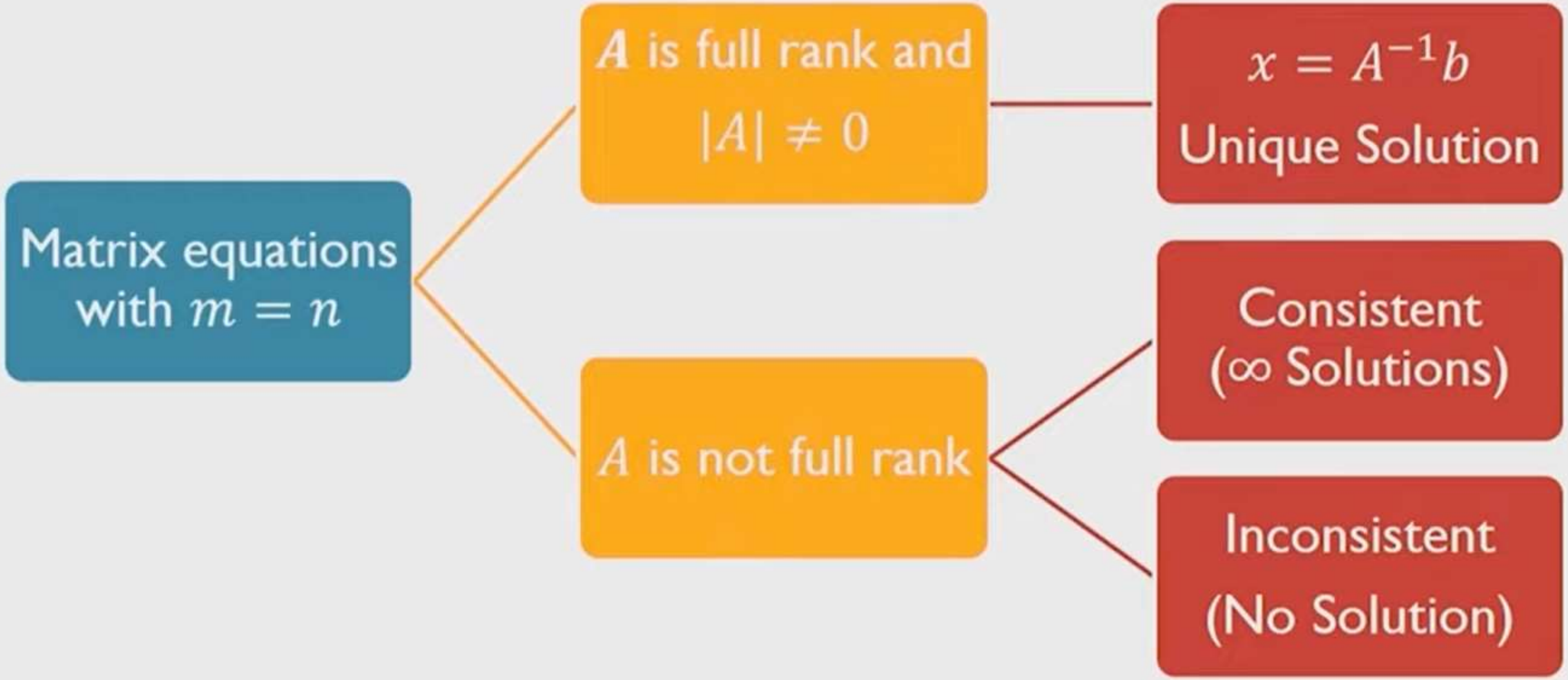
- When all the rows of the matrix are linearly independent
- Data sampling does not present a linear relationship – samples are independent

Full Column Rank

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

Row rank = Column rank

Case 1: $m = n$


Matrix equations with $m = n$
A is full rank and $|A| \neq 0$
$x = A^{-1}b$ Unique Solution
A is not full rank
Consistent (∞ Solutions)
Inconsistent (No Solution)

Case 1: Example 1.1

$Ax = b$

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

R Code

```
A=matrix(c(1,2,3,4),ncol=2, byrow=F)
b=c(7,10)
x=solve(A)%*%b
```

$$|\boldsymbol{A}| \neq 0$$

$$rank(\boldsymbol{A}) = 2 = no. of\ columns$$

- 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
```

Case 1: Example 1.2

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

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

- Checking consistency

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

$$Row(2) = 2Row(1)$$

- 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

Case 1: Example 1.3

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

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

- Checking consistency

$$\begin{bmatrix} x_1 + 2x_2 \\ 2x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$$
$$2Row(1) = 2x_1 + 4x_2 = 10 \neq 9$$

- Thus the equations are inconsistent
- One cannot find a solution to (x_1, x_2)

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

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

Case 2: An optimization perspective

- This optimization problem is

$$\min[(Ax-b)^T(Ax-b)]$$
$$= \min[(b^T - x^TA^T)\,(Ax-b)]$$
$$= min[(x^TA^TAx - 2b^TAx + b^Tb) = f(x)]$$

- 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

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

Case 2: An optimization perspective

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

$$2(A^T A)x - 2A^T b = 0$$
$$(A^T A)x = A^T b$$

- Assuming that all the columns are linearly independent

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