MY LEARNING JOURNEY IN PROBABILITY & STOCHASTIC PROCESS

(SEMESTER 1)

24pbd022 | Mauli Patel | SXCA | BDA(2024-2026)

What I Learned – Linear Algebra & Linear Programming

Semester 1 | M.Sc. Big Data Analytics

24pbd022 Mauli Patel

St. Xavier's College, Ahmedabad

INTRODUCTION

- **Objective:** To understand the fundamental concepts of Linear Algebra and Linear Programming and their practical applications in data science and optimization.

- **Why it matters:** These mathematical disciplines are foundational for understanding algorithms in machine learning, data manipulation, and resource optimization.

- **What we'll cover:** We'll explore matrix operations, solving linear systems, and formulating/solving optimization problems, with practical demonstrations.

UNIT A: LINEAR ALGEBRA - CORE CONCEPTS

A. Linear Equations and Matrices:

- **Linear Equations:** Equations where variables are only multiplied by constants and added together (e.g., $2x+3y=7$).
- **Matrices:** Rectangular arrays of numbers. They are powerful tools for organizing and manipulating data, especially systems of linear equations.
- **Why matrices?** They provide a compact and efficient way to represent and solve large systems of equations, which are common in data problems.

UNIT A: LINEAR ALGEBRA - CORE CONCEPTS

B. Matrix Operations:

- **Addition/Subtraction:** Element-wise operations, require matrices of the same dimensions.
- **Scalar Multiplication:** Multiplying every element by a single number.
- **Matrix Multiplication (Dot Product):** A more complex operation, crucial for transformations and solving systems. The number of columns in the first matrix must equal the number of rows in the second.
  - *Example:* If we have a system of equations, we can represent it as Ax=b, where A is a matrix, x is a vector of variables, and b is a vector of constants.
- **Transpose:** Swapping rows and columns (AT).
  - *Practical Relevance:* Used extensively in statistical modeling (e.g., Ordinary Least Squares in regression).

UNIT A: LINEAR ALGEBRA - SOLVING SYSTEMS & ADVANCED CONCEPTS

- **C. Solving Systems of Linear Equations:**
- **Concept:** Finding values for variables that satisfy all equations in a system simultaneously.
- **Gauss-Jordan Method:** A systematic algorithm using elementary row operations to transform a matrix into reduced row echelon form, directly yielding the solution.
    - *Visualizing:* Think of it as systematically eliminating variables to isolate solutions.

UNIT A: LINEAR ALGEBRA - SOLVING SYSTEMS & ADVANCED CONCEPTS

D. Determinants and Matrix Inverses:

- **Determinant:** A scalar value computed from the elements of a square matrix. It tells us important properties, such as whether a matrix is invertible.
- **Matrix Inverse (A^{-1}):** A matrix that, when multiplied by the original matrix, yields the identity matrix ($AA^{-1}=I$).
  - **Importance:** Allows us to "divide" by a matrix to solve equations like $Ax=b \Longrightarrow x=A^{-1}b$.
  - **Condition:** Only square matrices can have inverses, and their determinant must be non-zero.

UNIT A: LINEAR ALGEBRA - SOLVING SYSTEMS & ADVANCED CONCEPTS

E. Eigenvalues and Eigenvectors:

- **Concept:** Special vectors that, when a linear transformation is applied, only change by a scalar factor (the eigenvalue), not direction.
- **Significance:** Crucial in dimensionality reduction techniques like Principal Component Analysis (PCA), where eigenvectors represent the principal components and eigenvalues represent the variance captured by each component.

UNIT A: LINEAR ALGEBRA - SOLVING SYSTEMS & ADVANCED CONCEPTS

F. Positive Semi-definite and Positive Definite Matrices:

- **Concept:** Special types of symmetric matrices important in optimization, quadratic forms, and ensuring convexity in problems.
- **Relevance:** Used in optimization to determine if a function has a minimum (positive definite) or to analyze the stability of systems.

UNIT B: LINEAR PROGRAMMING - FUNDAMENTALS

A. Problem Definition:

- **Linear Programming (LP):** A mathematical method for determining a way to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships.
- **Components:**
    - **Objective Function:** The function to be maximized or minimized (e.g., profit, cost).
    - **Decision Variables:** The quantities we need to determine (e.g., number of units to produce).
    - **Constraints:** Linear inequalities or equalities that limit the values of the decision variables (e.g., resource availability, production capacity).

UNIT B: LINEAR PROGRAMMING - FUNDAMENTALS

B. Convex Sets, Corner Points, and Feasibility:

- **Feasible Region:** The set of all possible combinations of decision variables that satisfy all the constraints. In LP, this region is always a convex polygon (or polyhedron in higher dimensions).
- **Convex Set:** A set where for any two points in the set, the line segment connecting them is entirely within the set.
- **Corner Points (Extreme Points):** The vertices of the feasible region. A fundamental theorem of LP states that the optimal solution (if one exists) always occurs at one of these corner points.

UNIT B: LINEAR PROGRAMMING - SIMPLEX METHOD

Solutions that correspond to the corner points of the feasible region. These are the candidates for the optimal solution.

UNIT B: LINEAR PROGRAMMING - SIMPLEX METHOD

D. Simplex Method:

- **Concept:** An iterative algorithm for solving linear programming problems. It systematically moves from one basic feasible solution (corner point) to an adjacent one, always improving the objective function value, until the optimal solution is found.
- **How it works (Simplified):** It explores the vertices of the feasible region in an efficient manner, avoiding checking every single point.
- **Importance:** The most widely used algorithm for solving large-scale linear programming problems in practice.

PRACTICAL APPLICATIONS AND PYTHON IMPLEMENTATION

A. Python as Tools for Linear Algebra & Linear Programming:

- **Linear Algebra (R: base, Matrix package; Python: NumPy):**
    - Performing matrix operations (addition, multiplication, transpose, inverse).
    - Solving systems of linear equations (e.g., solve() in R, np.linalg.solve() in Python).
    - Calculating eigenvalues and eigenvectors (eigen() in R, np.linalg.eig() in Python).
    - ***Project 1 Connection:*** We demonstrated implementing Simple Linear Regression from scratch using NumPy, directly applying matrix multiplication, transpose, and inverse to find regression coefficients.
- **Linear Programming (R: lpSolve, Rglpk; Python: scipy.optimize.linprog):**
    - Formulating LP problems programmatically.
    - Solving LP problems to find optimal solutions for decision variables and objective function.
    - ***Project 2 Connection:*** We visualized the feasible region of a small LP problem and used scipy.optimize.linprog to find the optimal solution, showcasing both graphical and computational approaches.

PRACTICAL PROJECTS

- Simple Linear Regression from Scratch:
- Graphical and Solver-based Linear Programming

IMPORTANCE IN BIG DATA ANALYTICS

Linear Algebra: Essential for:
• Data representation (dataframes as matrices).
• Dimensionality reduction (PCA, SVD).
• Machine learning algorithms (regression, neural networks, support vector machines).
• Graph analytics.

Linear Programming: Essential for:
• Resource allocation and scheduling.
• Supply chain optimization.
• Portfolio optimization.
• Production planning.
• Any scenario requiring optimal decision-making under constraints.

CONCLUSION

Summary of Key Learnings:

- We've established a strong understanding of **Linear Algebra**, covering matrix operations, solving systems, and the significance of eigenvalues/eigenvectors. This provides the mathematical backbone for data manipulation and many machine learning algorithms.
- We've delved into **Linear Programming**, learning how to formulate optimization problems, understand feasible regions, and apply the Simplex method to find optimal solutions.

Thank You

- Questions?
- Let’s connect: LinkedIn | GitHub | Email