18AIE339T-MATRIX THEORY FOR ARTIFICIAL INTELLIGENCE

Dr. Antony Sophia N
Assistant Professor
Department of Computational Intelligence
SRM Institute of Science and Technology

Course Objective

Unit I – Linear System
Understand the basic concepts of linear algebra through computer science and Engineering applications

Unit II – Matrix Calculus
Learn the basic concepts of matrix calculus

Unit III – Matrix Analysis
Perform matrix analysis for various optimization algorithms

Unit IV – Matrix Solutions
Apply the concepts of vector spaces, linear transformations, matrices and inner product spaces in engineering

Unit V – Optimization
Solve problems in computer vision using optimization algorithms with single and multi-variables for large datasets

Reference Books

1. Xian-DaZhang, "A Matrix Algebra Approach to Artificial Intelligence" , Springer, 2021
2. Xian-DaZhang, "Matrix Analysis and Applications" ,Cambridge University Press, 2017
3. Charu C.Aggarwal, "Linear Algebra and Optimization for Machine Learning" , Springer, 2020.
4. Stephen Boyd,Lieven Vandenberghe, "Introduction to Applied Linear Algebra- Vectors, Matrices, and Least Squares" , Cambridge University Press, 2018
5. "LinearAlgebra", Kenneth Hoffman and RayKunze, Prentice Hall India,2013.
6. "LinearAlgebra", Cheney and Kincaid, Jones and Bartlett learning,2014

UNIT-II - Matrix Calculus

- Matrix Calculus
- Matrix Decomposition
- Operation and Properties of Matrix (Identity-Diagonal-Transpose-Symmetric-Trace-Norms)
- Operation and Properties of Matrix (Rank-Inverse-Orthogonal-Range-Determinant)
- Cramers Rule
- Eigenvalues and EigenVectors
- Cholesky Decomposition
- Qrdecomposition and LUdecomposition
- Eigen decomposition and Diagonalization
- Singular value Decomposition
- PCA
- Matrix Approximation

Matrix Calculus

- Matrix calculus is a specialized notation and mathematical method that simplifies the process of calculating derivatives for functions involving matrices and vectors.

- It collects these derivatives into organized structures, making it easier to handle complex multivariable calculus operations and solve problems in areas like optimization and differential equations.

- These collected derivatives are organized into matrices and vectors, which can be treated as unified entities during computations.

$$f(x) = \int_0^x f'(t)\, dt$$

Matrix Calculus (Contd.)

- Matrix calculus refers to a number of different notations that use matrices and vectors to collect the derivative of each component of the dependent variable with respect to each component of the independent variable.

Types of matrix derivative

Types	Scalar	Vector	Matrix
Scalar	$\frac{\partial y}{\partial x}$	$\frac{\partial \mathbf{y}}{\partial x}$	$\frac{\partial \mathbf{Y}}{\partial x}$
Vector	$\frac{\partial y}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	
Matrix	$\frac{\partial y}{\partial \mathbf{X}}$		

Differential Calculus

Differential Calculus

- The following rules are used for computing the derivatives of explicit functions
  - **Derivative of constants.** $\frac{d}{dx}c = 0.$
  - **Derivative of linear functions.** $\frac{d}{dx}(ax) = a.$
  - **Power rule.** $\frac{d}{dx}x^n = nx^{n-1}.$
  - **Derivative of exponentials.** $\frac{d}{dx}e^x = e^x.$
  - **Derivative of the logarithm.** $\frac{d}{dx}\log(x) = \frac{1}{x}.$
  - **Sum rule.** $\frac{d}{dx}\left(g(x) + h(x)\right) = \frac{dg}{dx}(x) + \frac{dh}{dx}(x).$
  - **Product rule.** $\frac{d}{dx}\left(g(x) \cdot h(x)\right) = g(x)\frac{dh}{dx}(x) + \frac{dg}{dx}(x)h(x).$
  - **Chain rule.** $\frac{d}{dx}g(h(x)) = \frac{dg}{dh}(h(x)) \cdot \frac{dh}{dx}(x).$

Scalar derivative rules

Rule	$f(x)$	Scalar derivative notation with respect to x	Example
Constant	c	0	$\frac{d}{dx}99 = 0$
Multiplication by constant	cf	$c\frac{df}{dx}$	$\frac{d}{dx}3x = 3$
Power Rule	x^n	nx^{n-1}	$\frac{d}{dx}x^3 = 3x^2$
Sum Rule	$f+g$	$\frac{df}{dx} + \frac{dg}{dx}$	$\frac{d}{dx}(x^2+3x) = 2x+3$
Difference Rule	$f-g$	$\frac{df}{dx} - \frac{dg}{dx}$	$\frac{d}{dx}(x^2-3x) = 2x-3$
Product Rule	fg	$f\frac{dg}{dx} + \frac{df}{dx}g$	$\frac{d}{dx}x^2x = x^2 + x2x = 3x^2$
Chain Rule	$f(g(x))$	$\frac{df(u)}{du}\frac{du}{dx}$, let $u = g(x)$	$\frac{d}{dx}ln(x^2) = \frac{1}{x^2}2x = \frac{2}{x}$

APPLICATIONS

AI Domain	Application	Use of Matrix Calculus
Gradient Descent and Optimization	Training machine learning models	Matrix calculus efficiently computes gradients, aiding in updating model parameters to minimize loss functions.
Backpropagation in Neural Networks	Training deep neural networks	Matrix calculus simplifies backpropagation by representing layer transformations and gradients as matrix operations.
Loss Functions and Regularization	Complex model formulation	Matrix calculus computes derivatives for loss functions and regularization terms, aiding in optimization.
Principal Component Analysis (PCA)	Dimensionality reduction	Matrix calculus is used in eigenvalue and eigenvector computations for PCA, helping understand data relationships.

APPLICATIONS

AI Domain	Application	Use of Matrix Calculus
Singular Value Decomposition (SVD)	Data compression, dimensionality reduction	Matrix calculus is used in the derivation of SVD and related operations.
Natural Language Processing (NLP)	Word embeddings, similarity computation	Matrix calculus is used to update embeddings, compute similarities, and optimize NLP models.
Reinforcement Learning	Policy gradients, Q-learning	Matrix calculus helps express gradients for optimizing policies and value functions in reinforcement learning.
Kernel Methods	SVMs, Gaussian Processes	Matrix calculus is essential for calculating kernel matrices and predictions in kernel methods.
Graph Neural Networks (GNNs)	Graph convolutions, gradients	Matrix calculus is used to propagate information, compute gradients, and optimize GNNs on graph data.
Matrix Factorization and Collaborative Filtering	Recommendation systems	Matrix calculus aids in optimizing factorizations for accurate user-item recommendations.

The Derivatives of Vector Functions

Let $\mathbf{x}$ and $\mathbf{y}$ be vectors of orders n and m respectively:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix},$$

where each component y_i may be a function of all the x_j, a fact represented by saying that $\mathbf{y}$ is a function of $\mathbf{x}$, or

$$\mathbf{y} = \mathbf{y}(\mathbf{x}).$$

If $n = 1$, $\mathbf{x}$ reduces to a scalar, which we call x. If $m = 1$, $\mathbf{y}$ reduces to a scalar, which we call y.

1.1 Derivative of Vector with Respect to Vector

The derivative of the vector **y** with respect to vector **x** is the $n \times m$ matrix

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \frac{\partial y_2}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

1.2 Derivative of a Scalar with Respect to Vector

If y is a scalar

$$\frac{\partial y}{\partial \mathbf{x}} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}.$$

It is also called the gradient of y with respect to a vector variable x, denoted by ∇y.

1.3 Derivative of Vector with Respect to Scalar

$$\frac{\partial \mathbf{y}}{\partial x} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \cdots & \frac{\partial y_m}{\partial x} \end{bmatrix}$$

Example 1

Given $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

and

$$y_1 = x_1^2 - x_2$$
$$y_2 = x_3^2 + 3x_2$$

the partial derivative matrix $\partial \mathbf{y} / \partial \mathbf{x}$ is computed as follows:

$$\boxed{\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} \\ \frac{\partial y_1}{\partial x_3} & \frac{\partial y_2}{\partial x_3} \end{bmatrix} = \begin{bmatrix} 2x_1 & 0 \\ -1 & 3 \\ 0 & 2x_3 \end{bmatrix}}$$

2 The Chain Rule for Vector Functions

Let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{bmatrix} \quad \text{and} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}$$

where z is a function of y, which is in turn a function of x, we can write

$$\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)^T = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} & \cdots & \frac{\partial z_1}{\partial x_n} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} & \cdots & \frac{\partial z_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial z_m}{\partial x_1} & \frac{\partial z_m}{\partial x_2} & \cdots & \frac{\partial z_m}{\partial x_n} \end{bmatrix}$$

Each entry of this matrix may be expanded as

$$\frac{\partial z_i}{\partial x_j} = \sum_{q=1}^{r} \frac{\partial z_i}{\partial y_q} \frac{\partial y_q}{\partial x_j} \qquad \begin{cases} i = 1, 2, \ldots, m \\ j = 1, 2, \ldots, n. \end{cases}$$

The Chain Rule for Vector Functions (Cont.)

Then

$$\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)^T = \begin{bmatrix} \sum \frac{\partial z_1}{y_q}\frac{\partial y_q}{\partial x_1} & \sum \frac{\partial z_1}{\partial y_q}\frac{\partial y_q}{\partial x_2} & \cdots & \sum \frac{\partial z_2}{\partial y_q}\frac{\partial y_q}{\partial x_n} \\ \sum \frac{\partial z_2}{\partial y_q}\frac{\partial y_q}{\partial x_1} & \sum \frac{\partial z_2}{\partial y_q}\frac{\partial y_q}{\partial x_2} & \cdots & \sum \frac{\partial z_2}{\partial y_q}\frac{\partial y_q}{\partial x_n} \\ \vdots & & & \\ \sum \frac{\partial z_m}{\partial y_q}\frac{\partial y_q}{\partial x_1} & \sum \frac{\partial z_m}{\partial y_q}\frac{\partial y_q}{\partial x_2} & \cdots & \sum \frac{\partial z_m}{\partial y_q}\frac{\partial y_q}{\partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial z_1}{\partial y_1} & \frac{\partial z_1}{\partial y_2} & \cdots & \frac{\partial z_1}{\partial y_r} \\ \frac{\partial z_2}{\partial y_1} & \frac{\partial z_2}{\partial y_2} & \cdots & \frac{\partial z_2}{\partial y_r} \\ \vdots & & & \\ \frac{\partial z_m}{\partial y_1} & \frac{\partial z_m}{\partial y_2} & \cdots & \frac{\partial z_m}{\partial y_r} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial y_r}{\partial x_1} & \frac{\partial y_r}{\partial x_2} & \cdots & \frac{\partial y_r}{\partial x_n} \end{bmatrix}$$

$$= \left(\frac{\partial \mathbf{z}}{\partial \mathbf{y}}\right)^T \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right)^T = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\frac{\partial \mathbf{z}}{\partial \mathbf{y}}\right)^T.$$

On transposing both sides, we finally obtain $\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\frac{\partial \mathbf{z}}{\partial \mathbf{y}},$

This is the chain rule for vectors (different from the conventional chain rule of calculus, the chain of matrices builds toward the left)

Example 2

x, y are as in Example 1 and z is a function of y defined as

$$z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}, \textit{ and } \begin{cases} z_1 = y_1^{\,2} - 2y_2 \\ z_2 = y_2^2 - y_1 \\ z_3 = y_1^2 + y_2^2 \\ z_4 = 2y_1 + y_2 \end{cases}, \text{ we have}$$

$$\frac{\partial z}{\partial y} = \begin{pmatrix} \frac{\partial z_1}{\partial y_1} & \frac{\partial z_2}{\partial y_1} & \frac{\partial z_3}{\partial y_1} & \frac{\partial z_4}{\partial y_1} \\ \frac{\partial z_1}{\partial y_2} & \frac{\partial z_2}{\partial y_2} & \frac{\partial z_3}{\partial y_2} & \frac{\partial z_4}{\partial y_2} \end{pmatrix} = \begin{pmatrix} 2y_1 & -1 & 2y_1 & 2 \\ -2 & 2y_2 & 2y_2 & 1 \end{pmatrix}.$$

Therefore,

$$\frac{\partial z}{\partial x} = \frac{\partial y}{\partial x}\frac{\partial z}{\partial y} = \begin{pmatrix} 2x_1 & 0 \\ -1 & 3 \\ 0 & 2x_3 \end{pmatrix}\begin{pmatrix} 2y_1 & -1 & 2y_1 & 2 \\ -2 & 2y_2 & 2y_2 & 1 \end{pmatrix} = \begin{pmatrix} 4x_1y_1 & -2x_1 & 4x_1y_1 & 4x_1 \\ -2y_1 - 6 & 1 + 6y_2 & -2y_2 + 6y_2 & 1 \\ -4x_3 & 4x_3y_2 & 4x_3y_2 & 2x_3 \end{pmatrix}$$

Matrix Decomposition

Matrix Decomposition

- Matrix decomposition, also known as matrix factorization or matrix factorization techniques, is a fundamental mathematical tool used in various areas of artificial intelligence and machine learning.

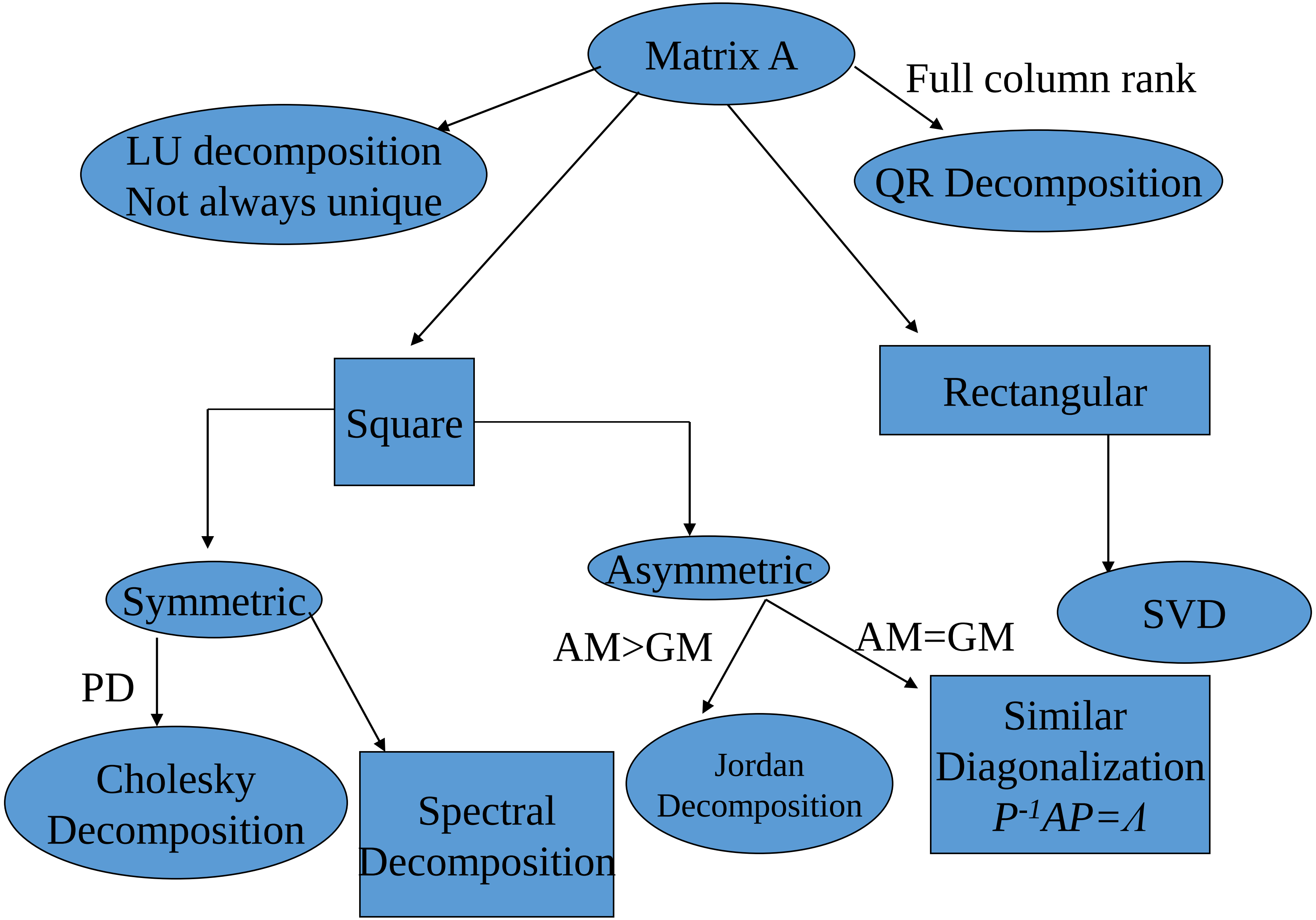
- It involves breaking down a matrix into multiple matrices whose combination reproduces the original matrix.

- Matrix decomposition techniques are utilized for various purposes, including dimensionality reduction, data compression, feature extraction, and solving optimization problems.

Application : Matrix Decomposition

Matrix Decomposition Technique	Purpose	Application Areas
Singular Value Decomposition (SVD)	Dimensionality reduction, data compression, feature extraction	Image compression, collaborative filtering, recommendation systems
Principal Component Analysis (PCA)	Dimensionality reduction, feature extraction	Image analysis, data visualization, noise reduction
Non-Negative Matrix Factorization (NMF)	Feature extraction, pattern recognition	Topic modeling, text mining, image analysis
Eigenvalue Decomposition	Eigenvalue and eigenvector computation	Linear algebra, eigenvalue problems
LU Decomposition (Lower-Upper)	Solving linear systems, matrix inversion	Numerical analysis, solving linear equations
QR Decomposition	Least squares problems, eigenvalue problems	Optimization, regression analysis
Cholesky Decomposition	Symmetric positive definite matrix factorization	Solving linear systems, simulations

Decomposition in Diagram



Easy to solve system

Some linear system that can be easily solved

$$\begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

The solution:

$$\begin{bmatrix} b_1 / a_{11} \\ b_2 / a_{22} \\ \vdots \\ b_n / a_{nn} \end{bmatrix}$$

Easy to solve system (Cont.)

Lower triangular matrix:

$$\begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Solution: This system is solved using forward substitution

$$x_i = (b_i - \textstyle\sum_{j=1}^{i-1} a_{ij}.x_j)/a_{ii}$$

Forward Elimination

To eliminate x_1

$$\left.\begin{aligned} a_{ij} &\leftarrow a_{ij} - \left(\frac{a_{i1}}{a_{11}}\right) a_{1j} \quad (1 \le j \le n) \\ b_i &\leftarrow b_i - \left(\frac{a_{i1}}{a_{11}}\right) b_1 \end{aligned}\right\} 2 \le i \le n$$

To eliminate x_2

$$\left.\begin{aligned} a_{ij} &\leftarrow a_{ij} - \left(\frac{a_{i2}}{a_{22}}\right) a_{2j} \quad (2 \le j \le n) \\ b_i &\leftarrow b_i - \left(\frac{a_{i2}}{a_{22}}\right) b_2 \end{aligned}\right\} 3 \le i \le n$$

Forward Elimination (Contd.)

$$\text{To eliminate } x_k \quad \left.\begin{array}{l} a_{ij} \leftarrow a_{ij} - \left(\dfrac{a_{ik}}{a_{kk}}\right) a_{kj} \quad (k \le j \le n) \\[2ex] b_i \leftarrow b_i - \left(\dfrac{a_{ik}}{a_{kk}}\right) b_k \end{array}\right\} k+1 \le i \le n$$

Continue until x_{n-1} is eliminated.

Easy to solve system (Cont.)

Upper Triangular Matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & 0 & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Solution: This system is solved using Backward substitution

$$x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij}.x_j\right)/a_{ii}$$

Example : Forward Elimination

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ 26 \\ -19 \\ -34 \end{bmatrix}$$

Part 1 : Forward Eliminatio n

Step1 : Eliminate x_1 from equations $2, 3, 4$

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -27 \\ -18 \end{bmatrix}$$

Example : Forward Elimination (Contd.)

Step2 : Eliminate x_2 from equations 3, 4

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -21 \end{bmatrix}$$

Step3 : Eliminate x_3 from equation 4

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

Backward Substitution

$$x_n = \frac{b_n}{a_{n,n}}$$

$$x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}$$

$$x_{n-2} = \frac{b_{n-2} - a_{n-2,n}x_n - a_{n-2,n-1}x_{n-1}}{a_{n-2,n-2}}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{i,j}x_j}{a_{i,i}}$$

Example : Backward Substitution

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix}$$

Solve for x_4, then solve for x_3,... solve for x_1

$$x_4 = \frac{-3}{-3} = 1, \qquad x_3 = \frac{-9+5}{2} = -2$$

$$x_2 = \frac{-6-2(-2)-2(1)}{-4} = 1, \quad x_1 = \frac{16+2(1)-2(-2)-4(1)}{6} = 3$$

Operation and Properties of Matrix (Identity-Diagonal-Transpose-Symmetric-Trace-Norms)

Matrices

- *Matrix* is a rectangular array of real-valued scalars arranged in m horizontal rows and n vertical columns
  - Each element a_{ij} belongs to the i^{th} row and j^{th} column
  - The elements are denoted a_{ij} or $\mathbf{A}_{ij}$ or $[\mathbf{A}]_{ij}$ or $\mathbf{A}(\boldsymbol{i},\boldsymbol{j})$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- For the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the size (dimension) is $m \times n$ or (m, n)
  - Matrices are denoted by bold-font upper-case letters

Matrices

Matrices

- Addition or subtraction

$$(\mathbf{A} \pm \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \pm \mathbf{B}_{i,j}$$

$$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 1+5 \\ 1+7 & 0+5 & 0+0 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 8 & 5 & 0 \end{bmatrix}$$

- Scalar multiplication

$$(c\mathbf{A})_{i,j} = c \cdot \mathbf{A}_{i,j}$$

$$2 \cdot \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 8 & 2 \cdot -3 \\ 2 \cdot 4 & 2 \cdot -2 & 2 \cdot 5 \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 10 \end{bmatrix}$$

- Matrix multiplication

$$(\mathbf{AB})_{i,j} = \mathbf{A}_{i,1}\mathbf{B}_{1,j} + \mathbf{A}_{i,2}\mathbf{B}_{2,j} + \cdots + \mathbf{A}_{i,n}\mathbf{B}_{n,j}$$

  - Defined only if the number of columns of the left matrix is the same as the number of rows of the right matrix
  - Note that $\mathbf{AB} \neq \mathbf{BA}$

$$\begin{bmatrix} \underline{2} & \underline{3} & \underline{4} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \underline{1000} \\ 1 & \underline{100} \\ 0 & \underline{10} \end{bmatrix} = \begin{bmatrix} 3 & \underline{2340} \\ 0 & 1000 \end{bmatrix}$$

Matrices

Matrices

- *Transpose* of the matrix: $\mathbf{A}^T$ has the rows and columns exchanged

$$\left(\mathbf{A}^T\right)_{i,j} = \mathbf{A}_{j,i} \qquad \begin{bmatrix} 1 & 2 & 3 \\ 0 & -6 & 7 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 \\ 2 & -6 \\ 3 & 7 \end{bmatrix}$$

  - Some properties

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A} \qquad \mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$$

$$(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \boldsymbol{B}^T \qquad \mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$$

$$(\mathbf{A}^T)^T = \mathbf{A} \qquad (\mathbf{AB})^T = \boldsymbol{B}^T\mathbf{A}^T$$

- *Square matrix*: has the same number of rows and columns

- *Identity matrix* ($\mathbf{I}_n$): has ones on the main diagonal, and zeros elsewhere
  - E.g.: identity matrix of size 3×3 :

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrices

- ***Determinant*** of a matrix, denoted by $\det(\mathbf{A})$ or $|\mathbf{A}|$, is a real-valued scalar encoding certain properties of the matrix
  - E.g., for a matrix of size 2×2:

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

  - For larger-size matrices the determinant of a matrix id calculated as

$$\det(\mathbf{A}) = \sum_j a_{ij}(-1)^{i+j} det\left(\mathbf{A}_{(i,j)}\right)$$

  - In the above, $\mathbf{A}_{(i,j)}$ is a minor of the matrix obtained by removing the row and column associated with the indices i and j
- ***Trace*** of a matrix is the sum of all diagonal elements

$$\mathrm{Tr}(\mathbf{A}) = \sum_i a_{ii}$$

- A matrix for which $\mathbf{A} = \mathbf{A}^T$ is called a ***symmetric matrix***

Matrices

- Elementwise multiplication of two matrices **A** and **B** is called the *Hadamard product* or *elementwise product*
  - The math notation is $\odot$

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \dots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & \dots & a_{2n}b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \dots & a_{mn}b_{mn} \end{bmatrix}$$

Matrix-Vector Products

- Consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{x} \in \mathbb{R}^n$
- The matrix can be written in terms of its row vectors (e.g., $\mathbf{a}_1^T$ is the first row)

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix}$$

- The matrix-vector product is a column vector of length m, whose i^{th} element is the dot product $\mathbf{a}_i^T \mathbf{x}$

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{x} \\ \mathbf{a}_2^\top \mathbf{x} \\ \vdots \\ \mathbf{a}_m^\top \mathbf{x} \end{bmatrix}$$

- Note the size: $\mathbf{A}(m \times n) \cdot \mathbf{x}(n \times 1) = \mathbf{A}\mathbf{x}(m \times 1)$

Matrix-Matrix Products

Matrices

- To multiply two matrices $\mathbf{A} \in \mathbb{R}^{n\times k}$ and $\mathbf{B} \in \mathbb{R}^{k\times m}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{km} \end{bmatrix}$$

- We can consider the matrix-matrix product as dot-products of rows in **A** and columns in **B**

$$\mathbf{C} = \mathbf{A}\mathbf{B} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_m \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{b}_1 & \mathbf{a}_1^\top \mathbf{b}_2 & \cdots & \mathbf{a}_1^\top \mathbf{b}_m \\ \mathbf{a}_2^\top \mathbf{b}_1 & \mathbf{a}_2^\top \mathbf{b}_2 & \cdots & \mathbf{a}_2^\top \mathbf{b}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_n^\top \mathbf{b}_1 & \mathbf{a}_n^\top \mathbf{b}_2 & \cdots & \mathbf{a}_n^\top \mathbf{b}_m \end{bmatrix}$$

- Size: $\mathbf{A}(n \times k) \cdot \mathbf{B}(k \times m) = \mathbf{C}(n \times m)$

Linear Depedence

Matrices

- For the following matrix $\mathbf{B} = \begin{bmatrix} 2 & -1 \\ 4 & -2 \end{bmatrix}$

- Notice that for the two columns $\mathbf{b}_1 = [2, 4]^T$ and $\mathbf{b}_2 = [-1, -2]^T$, we can write $\mathbf{b}_1 = -2 \cdot \mathbf{b}_2$
  - This means that the two columns are linearly dependent
- The weighted sum $a_1\mathbf{b}_1 + a_2\mathbf{b}_2$ is referred to as a linear combination of the vectors $\mathbf{b}_1$ and $\mathbf{b}_2$
  - In this case, a linear combination of the two vectors exist for which $\mathbf{b}_1 + 2 \cdot \mathbf{b}_2 = \mathbf{0}$
- A collection of vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ are ***linearly dependent*** if there exist coefficients $a_1, a_2, \dots, a_k$ not all equal to zero, so that

$$\sum_{i=1}^{k} a_i \mathbf{v_i} = 0$$

- If there is no linear dependence, the vectors are ***linearly independent***

Matrix Norms

- ***Frobenius norm*** – calculates the square-root of the summed squares of the elements of matrix **X**
  - This norm is similar to Euclidean norm of a vector

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2}$$

- ***Spectral norm*** – is the largest singular value of matrix **X**
  - Denoted $\|\mathbf{X}\|_2$
  - The singular values of **X** are $\sigma_1, \sigma_2, \ldots, \sigma_m$

$$\|\mathbf{X}\|_2 = \sigma_{max}(\mathbf{X})$$

- ***$L_{2,1}$ norm*** – is the sum of the Euclidean norms of the columns of matrix **X**

$$\|\mathbf{X}\|_{2,1} = \sum_{j=1}^{n}\sqrt{\sum_{i=1}^{m} x_{ij}^2}$$

- ***Max norm*** – is the largest element of matrix **X**

$$\|\mathbf{X}\|_{\max} = \max_{i,j}(x_{ij})$$

Operation and Properties of Matrix (Rank-Inverse-Orthogonal-Range-Determinant)

Matrix Rank

- For an $n \times m$ matrix, the *rank* of the matrix is the largest number of linearly independent columns
- The matrix $\mathbf{B}$ from the previous example has $rank(\mathbf{B}) = 1$, since the two columns are linearly dependent

$$\mathbf{B} = \begin{bmatrix} 2 & -1 \\ 4 & -2 \end{bmatrix}$$

- The matrix $\mathbf{C}$ below has $rank(\mathbf{C}) = 2$, since it has two linearly independent columns
  - I.e., $\mathbf{c}_4 = -1 \cdot \mathbf{c}_1$, $\mathbf{c}_5 = -1 \cdot \mathbf{c}_3$, $\mathbf{c}_2 = 3 \cdot \mathbf{c}_1 + 3 \cdot \mathbf{c}_3$

$$\mathbf{C} = \begin{bmatrix} 1 & 3 & 0 & -1 & 0 \\ -1 & 0 & 1 & 1 & -1 \\ 0 & -3 & 1 & 0 & -1 \\ 2 & 3 & -1 & -2 & 1 \end{bmatrix}$$

Inverse of a Matrix

- For a square $n \times n$ matrix $\mathbf{A}$ with rank n, $\mathbf{A}^{-1}$ is its ***inverse matrix*** if their product is an identity matrix $\mathbf{I}$

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

- Properties of inverse matrices

$$\left(\mathbf{A}^{-1}\right)^{-1} = \mathbf{A}$$

$$\left(\mathbf{AB}\right)^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

- If $\det(A) = 0$ (i.e., $\mathrm{rank}(A) < n$), then the inverse does not exist
  - A matrix that is not invertible is called a singular matrix
- Note that finding an inverse of a large matrix is computationally expensive
  - In addition, it can lead to numerical instability
- If the inverse of a matrix is equal to its transpose, the matrix is said to be orthogonal matrix

$$\mathbf{A}^{-1} = \mathbf{A}^{T}$$

Pseudo-Inverse of a Matrix

Matrices

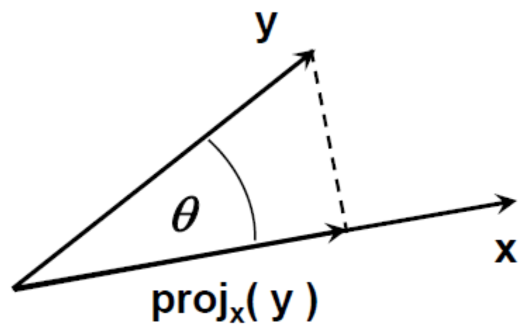
- *Pseudo-inverse* of a matrix
  - Also known as Moore-Penrose pseudo-inverse
- For matrices that are not square, the inverse does not exist
  - Therefore, a pseudo-inverse is used
- If $m > n$, then the pseudo-inverse is $\mathbf{A}^{\dagger} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{T}$ and $\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{I}$
- If $m < n$, then the pseudo-inverse is $\mathbf{A}^{\dagger} = \mathbf{A}^{T}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)^{-1}$ and $\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{I}$
  - E.g., for a matrix with dimension $\mathbf{X}_{2\times3}$, a pseudo-inverse can be found of size $\mathbf{X}^{\dagger}_{3\times2}$, so that $\mathbf{X}_{2\times3}\mathbf{X}^{\dagger}_{3\times2} = \mathbf{I}_{2\times2}$

Vector Projection

Vectors

- *Orthogonal projection* of a vector $\mathbf{y}$ onto vector $\mathbf{x}$
  - The projection can take place in any space of dimensionality ≥ 2
  - The unit vector in the direction of $\mathbf{x}$ is $\frac{\mathbf{x}}{\|\mathbf{x}\|}$
    - A unit vector has norm equal to 1
  - The length of the projection of $\mathbf{y}$ onto $\mathbf{x}$ is $\|\mathbf{y}\| \cdot cos(\theta)$
  - The orthogonal project is the vector $\mathbf{proj_x}(\mathbf{y})$

$$\mathbf{proj_x}(\mathbf{y}) = \frac{\mathbf{x} \cdot \|\mathbf{y}\| \cdot cos(\theta)}{\|\mathbf{x}\|}$$



Cramer’s Rule

Cramer's Rule

- Cramer's Rule is a method for solving a system of linear equations using determinants.

- It's applicable when the number of equations is equal to the number of variables (i.e., the system is square) and the determinant of the coefficient matrix is non-zero.

- Cramer's Rule provides a way to find the individual values of the variables by expressing them as ratios of determinants.

Linear Equations and Determinants

- The solutions of linear equations can sometimes be expressed using determinants.

  - To illustrate, let's solve the following pair of linear equations for the variable *x*.

$$\begin{cases} ax + by = r \\ cx + dy = s \end{cases}$$

Linear Equations and Determinants

- To eliminate the variable y, we multiply the first equation by d and the second by b, and subtract.

$$\begin{array}{r} adx + bdy = rd \\ bcx + bdy = bs \\ \hline adx - bcx = rd - bs \end{array}$$

Linear Equations and Determinants

- Factoring the left-hand side, we get:

$$(ad - bc)x = rd - bs$$

  - Assuming that $ad - bc \neq 0$, we can now solve this equation for x:

$$x = \frac{rd - bs}{ad - bc}$$

  - Similarly, we find:

$$y = \frac{as - cr}{ad - bc}$$

Cramer's Rule for Systems in Two Variables

- The linear system
  $$\begin{cases} ax + by = r \\ cx + dy = s \end{cases}$$
  has the solution
  $$x = \frac{\begin{vmatrix} r & b \\ s & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} \qquad y = \frac{\begin{vmatrix} a & r \\ c & s \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}}$$
  provided
  $$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \neq 0$$

Cramer's Rule

- Using the notation

$$D = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad D_x = \begin{bmatrix} r & b \\ s & d \end{bmatrix} \quad D_y = \begin{bmatrix} a & r \\ c & s \end{bmatrix}$$

the solution of the system can be written as:

$$x = \frac{|D_x|}{|D|} \quad \text{and} \quad y = \frac{|D_y|}{|D|}$$

Example —Cramer's Rule for a System with Two Variables

- Use Cramer's Rule to solve the system.

$$\begin{cases} 2x + 6y = -1 \\ \quad x + 8y = \ \ 2 \end{cases}$$

Example (Contd.)

- For this system, we have:

$$|D| = \begin{vmatrix} 2 & 6 \\ 1 & 8 \end{vmatrix} = 2 \cdot 8 - 6 \cdot 1 = 10$$

$$|D_x| = \begin{vmatrix} -1 & 6 \\ 2 & 8 \end{vmatrix} = (-1)8 - 6 \cdot 2 = -20$$

$$|D_y| = \begin{vmatrix} 2 & -1 \\ 1 & 2 \end{vmatrix} = 2 \cdot 2 - (-1)1 = 5$$

Example (Contd.)

- The solution is:

$$x = \frac{|D_x|}{|D|} = \frac{-20}{10} = -2$$

$$y = \frac{|D_y|}{|D|} = \frac{5}{10} = \frac{1}{2}$$

Cramer's Rule

- Cramer's Rule can be extended to apply to any system of *n* linear equations in *n* variables in which the determinant of the coefficient matrix is not zero.
- As we saw in the preceding section, any such system can be written in matrix form as:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Example —Cramer's Rule for a System of Three Variables

- Use Cramer's Rule to solve the system.

$$\begin{cases} 2x - 3y + 4z = 1 \\ x \qquad\quad + 6z = 0 \\ 3x - 2y \qquad\quad = 5 \end{cases}$$

- First, we evaluate the determinants that appear in Cramer's Rule.

Example 2 : Cramer's Rule for a System of Three Variables

$\lvert D\rvert = \begin{vmatrix} 2 & -3 & 4 \\ 1 & 0 & 6 \\ 3 & -2 & 0 \end{vmatrix} = -38$	$\lvert D_x\rvert = \begin{vmatrix} 1 & -3 & 4 \\ 0 & 0 & 6 \\ 5 & -2 & 0 \end{vmatrix} = -78$
$\lvert D_y\rvert = \begin{vmatrix} 2 & 1 & 4 \\ 1 & 0 & 6 \\ 3 & 5 & 0 \end{vmatrix} = -22$	$\lvert D_z\rvert = \begin{vmatrix} 2 & -3 & 1 \\ 1 & 0 & 0 \\ 3 & -2 & 5 \end{vmatrix} = 13$

- Note that D is the coefficient matrix and that D_x, D_y, and D_z are obtained by replacing the first, second, and third columns of D by the constant terms.

Example—Cramer's Rule for a System of Three Variables

- Now, we use Cramer's Rule to get the solution:

$$x = \frac{|D_x|}{|D|} = \frac{-78}{-38} = \frac{39}{19}$$

$$y = \frac{|D_y|}{|D|} = \frac{-22}{-38} = \frac{11}{19}$$

$$z = \frac{|D_z|}{|D|} = \frac{13}{-38} = -\frac{13}{38}$$

Limitations of Cramer's Rule

- However, in systems with more than three equations, evaluating the various determinants involved is usually a long and tedious task.

- This is unless you are using a graphing calculator.

- Moreover, the rule doesn't apply if $|D| = 0$ or if D is not a square matrix.

- So, Cramer's Rule is a useful alternative to Gaussian elimination—but only in some situations.

Definition of Eigenvalues and Eigenvectors

DEFINITION 1. Definition of Eigenvalues and Eigenvectors

If there exists a non-zero solution vector $\vec{x}$ for an arbitrary $n \times n$ matrix A, then the number λ can be called an eigenvalue of matrix A, where

$$A\vec{x} = \lambda\vec{x} \tag{2}$$

The solution vector $\vec{x}$ is the corresponding eigenvector for eigenvalue λ.

In this case, equation (2) can be rewritten by the properties of matrices as follows.

$$(A - \lambda I)\vec{x} = 0 \tag{3}$$

Here, I is the identity matrix.

There are two conditions for equation (3) to hold, which are when the expression inside the parentheses becomes 0, and when $\vec{x} = 0$. If only the first condition is met, we can find an appropriate λ and a non-zero $\vec{x}$. However, if the second condition is satisfied, we obtain a 'trivial solution' with λ and $\vec{x} = 0$.

Therefore, in order to avoid obtaining the 'trivial solution' of $\vec{x} = 0$ from the expression inside the parentheses of equation (3), the necessary and sufficient condition for having nontrivial solutions is

Therefore, by the properties of matrices, we have $(A - \lambda I)\vec{x} = 0$. Moreover, in order for $\vec{x}$ to have nontrivial solutions, the following condition must be satisfied.

$$det(A - \lambda I) = 0 \tag{7}$$

Hence,

$$det(A - \lambda I) = det\left(\begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}\right) = 0 \tag{8}$$

$$\Rightarrow (2-\lambda)^2 - 1 \tag{9}$$

$$= (4 - 4\lambda + \lambda^2) - 1 \tag{10}$$

$$= \lambda^2 - 4\lambda + 3 = 0 \tag{11}$$

Therefore, $\lambda_1 = 1$ and $\lambda_2 = 3$.

In other words, the eigenvalues of the linear transformation A are 1 and 3. This means that when a vector that does not change in direction but only scales in size is transformed, its size will be multiplied by 1 and 3, respectively. Now, let's find the eigenvectors.

Again, Equation (2) must satisfy both eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 3$. Therefore, for the case of $\lambda_1 = 1$,

$$A\vec{x} = \lambda_1\vec{x} \tag{12}$$

$$\Rightarrow \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{13}$$

must be satisfied, which leads to the following system of linear equations:

$$2x_1 + x_2 = x_1 \tag{14}$$

$$x_1 + 2x_2 = x_2 \tag{15}$$

Therefore, the eigenvector for $\lambda_1 = 1$ is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{16}$$

Similarly, using the same method, the eigenvector for $\lambda_2 = 3$ is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{17}$$

Geometrically, this means that the vector $\vec{x} = [1, 1]$ remains unchanged in direction but scales by a factor of 3 under the linear transformation A. Similarly, the vector $\vec{x} = [1, -1]$ remains unchanged in direction but scales by a factor of 1 under the linear transformation A.

Eigen Decomposition

Eigen Decomposition

- ***Eigen decomposition*** is decomposing a matrix into a set of eigenvalues and eigenvectors
- ***Eigenvalues*** of a square matrix $\mathbf{A}$ are scalars λ and ***eigenvectors*** are non-zero vectors $\mathbf{v}$ that satisfy

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

- Eigenvalues are found by solving the following equation

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

- If a matrix $\mathbf{A}$ has n linearly independent eigenvectors $\{\mathbf{v}^1, \dots, \mathbf{v}^n\}$ with corresponding eigenvalues $\{\lambda_1, \dots, \lambda_n\}$, the eigen decomposition of $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

  - Columns of the matrix $\mathbf{V}$ are the eigenvectors, i.e., $\mathbf{V} = [\mathbf{v}^1, \dots, \mathbf{v}^n]$
  - $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues, i.e., $\mathbf{\Lambda} = [\lambda_1, \dots, \lambda_n]$
- To find the inverse of the matrix A, we can use $\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^{-1}$
  - This involves simply finding the inverse $\mathbf{\Lambda}^{-1}$ of a diagonal matrix

Eigen Decomposition

Eigen Decomposition

- Decomposing a matrix into eigenvalues and eigenvectors allows to analyze certain properties of the matrix
  - If all eigenvalues are positive, the matrix is positive definite
  - If all eigenvalues are positive or zero-valued, the matrix is positive semidefinite
  - If all eigenvalues are negative or zero-values, the matrix is negative semidefinite
    - Positive semidefinite matrices are interesting because they guarantee that $\forall \mathbf{x}, \mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$
- Eigen decomposition can also simplify many linear-algebraic computations
  - The determinant of A can be calculated as

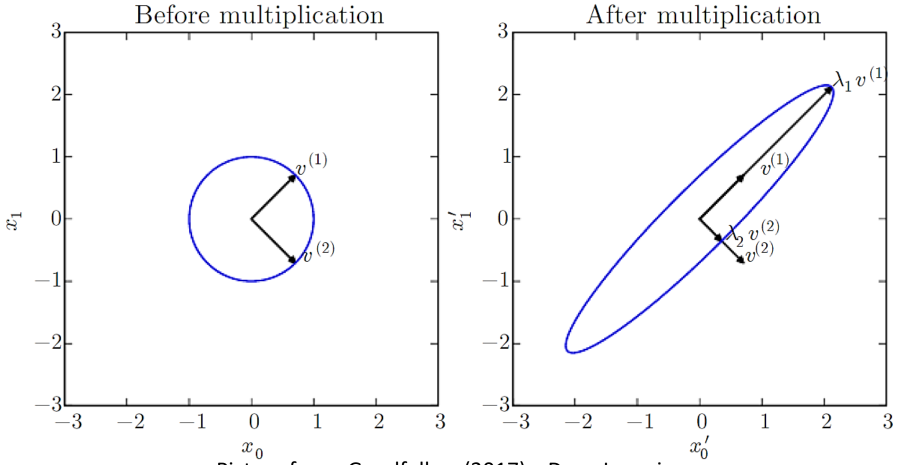
$$\det(\mathbf{A}) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$$

  - If any of the eigenvalues are zero, the matrix is singular (it does not have an inverse)
- However, not every matrix can be decomposed into eigenvalues and eigenvectors
  - Also, in some cases the decomposition may involve complex numbers
  - Still, every real symmetric matrix is guaranteed to have an eigen decomposition according to $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{V}$ is an orthogonal matrix

Eigen Decomposition

Eigen Decomposition

- Geometric interpretation of the eigenvalues and eigenvectors is that they allow to stretch the space in specific directions
  - Left figure: the two eigenvectors $\mathbf{v}^1$ and $\mathbf{v}^2$ are shown for a matrix, where the two vectors are unit vectors (i.e., they have a length of 1)
  - Right figure: the vectors $\mathbf{v}^1$ and $\mathbf{v}^2$ are multiplied with the eigenvalues λ_1 and λ_2
    - We can see how the space is scaled in the direction of the larger eigenvalue λ_1
- E.g., this is used for dimensionality reduction with PCA (principal component analysis) where the eigenvectors corresponding to the largest eigenvalues are used for extracting the most important data dimensions



Picture from: Goodfellow (2017) – Deep Learning

LU Decomposition

Method

For most non-singular matrix $[A]$ that one could conduct Naïve Gauss Elimination forward elimination steps, one can always write it as

$$[A] = [L][U]$$

where

$[L]$ = lower triangular matrix

$[U]$ = upper triangular matrix

How does LU Decomposition work?

If solving a set of linear equations	$[A][X] = [C]$
If $[A] = [L][U]$ then	$[L][U][X] = [C]$
Multiply by	$[L]^{-1}$
Which gives	$[L]^{-1}[L][U][X] = [L]^{-1}[C]$
Remember $[L]^{-1}[L] = [I]$ which leads to	$[I][U][X] = [L]^{-1}[C]$
Now, if $[I][U] = [U]$ then	$[U][X] = [L]^{-1}[C]$
Now, let	$[L]^{-1}[C]=[Z]$
Which ends with	$[L][Z] = [C] \quad (1)$
and	$[U][X] = [Z] \quad (2)$

LU Decomposition

How can this be used?

Given $[A][X] = [C]$

1. Decompose $[A]$ into $[L]$ and $[U]$
2. Solve $[L][Z] = [C]$ for $[Z]$
3. Solve $[U][X] = [Z]$ for $[X]$

Is LU Decomposition better than Gaussian Elimination?

Solve $[A][X] = [B]$

T = clock cycle time and nxn = size of the matrix

Forward Elimination

$$CT|_{FE} = T\left(\frac{8n^3}{3} + 8n^2 - \frac{32n}{3}\right)$$

Back Substitution

$$CT|_{BS} = T\left(4n^2 + 12n\right)$$

Decomposition to LU

$$CT|_{DE} = T\left(\frac{8n^3}{3} + 4n^2 - \frac{20n}{3}\right)$$

Forward Substitution

$$CT|_{FS} = T\left(4n^2 - 4n\right)$$

Back Substitution

$$CT|_{BS} = T\left(4n^2 + 12n\right)$$

Is LU Decomposition better than Gaussian Elimination?

To solve $[A][X] = [B]$

Time taken by methods

Gaussian Elimination	LU Decomposition
$T\left(\frac{8n^3}{3}+12n^2+\frac{4n}{3}\right)$	$T\left(\frac{8n^3}{3}+12n^2+\frac{4n}{3}\right)$

T = clock cycle time and nxn = size of the matrix

So both methods are equally efficient.

To find inverse of [A]

Time taken by Gaussian Elimination

$$= n\left(CT|_{FE} + CT|_{BS}\right)$$

$$= T\left(\frac{8n^4}{3} + 12n^3 + \frac{4n^2}{3}\right)$$

Time taken by LU Decomposition

$$= CT|_{DE} + n \times CT|_{FS} + n \times CT|_{BS}$$

$$= T\left(\frac{32n^3}{3} + 12n^2 - \frac{20n}{3}\right)$$

To find inverse of [A]

Time taken by Gaussian Elimination

$$T\left(\frac{8n^4}{3}+12n^3+\frac{4n^2}{3}\right)$$

Time taken by LU Decomposition

$$T\left(\frac{32n^3}{3}+12n^2-\frac{20n}{3}\right)$$

Table 1 Comparing computational times of finding inverse of a matrix using LU decomposition and Gaussian elimination.

n	10	100	1000	10000
$CT\|_{\text{inverse GE}} / CT\|_{\text{inverse LU}}$	3.288	25.84	250.8	2501

For large n, $CT|_{\text{inverse GE}} / CT|_{\text{inverse LU}} \approx n/4$

Method: [A] Decomposes to [L] and [U]

$$[A] = [L][U] = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

[*U*] is the same as the coefficient matrix at the end of the forward elimination step.

[*L*] is obtained using the *multipliers* that were used in the forward elimination process

Finding the [*U*] matrix

Using the Forward Elimination Procedure of Gauss Elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix}$$

Step 1: $\dfrac{64}{25} = 2.56; \quad Row2 - Row1(2.56) = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 144 & 12 & 1 \end{bmatrix}$

$$\frac{144}{25} = 5.76; \quad Row3 - Row1(5.76) = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix}$$

Finding the [U] Matrix

Matrix after Step 1: $\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix}$

Step 2: $\dfrac{-16.8}{-4.8} = 3.5; \quad Row3 - Row2(3.5) = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$

$$[U] = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

Finding the [L] matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix}$$

Using the multipliers used during the Forward Elimination Procedure

From the first step of forward elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix}$$

$$\ell_{21} = \frac{a_{21}}{a_{11}} = \frac{64}{25} = 2.56$$

$$\ell_{31} = \frac{a_{31}}{a_{11}} = \frac{144}{25} = 5.76$$

Finding the [L] Matrix

From the second step of forward elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix} \qquad \ell_{32} = \frac{a_{32}}{a_{22}} = \frac{-16.8}{-4.8} = 3.5$$

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix}$$

Does [L][U] = [A]?

$$[L][U]=\begin{bmatrix}1 & 0 & 0\\ 2.56 & 1 & 0\\ 5.76 & 3.5 & 1\end{bmatrix}\begin{bmatrix}25 & 5 & 1\\ 0 & -4.8 & -1.56\\ 0 & 0 & 0.7\end{bmatrix}= \ ?$$

Using LU Decomposition to solve SLEs

Solve the following set of linear equations using LU Decomposition

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix}$$

Using the procedure for finding the [*L*] and [*U*] matrices

$$[A] = [L][U] = \begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix} \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

Example

Set $[L][Z] = [C]$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix}$$

Solve for $[Z]$

$$z_1 = 10$$
$$2.56 z_1 + z_2 = 177.2$$
$$5.76 z_1 + 3.5 z_2 + z_3 = 279.2$$

Example

Complete the forward substitution to solve for $[Z]$

$$
\begin{aligned}
z_1 &= 106.8 \\
z_2 &= 177.2 - 2.56 z_1 \\
&= 177.2 - 2.56(106.8) \\
&= -96.2 \\
z_3 &= 279.2 - 5.76 z_1 - 3.5 z_2 \\
&= 279.2 - 5.76(106.8) - 3.5(-96.21) \\
&= 0.735
\end{aligned}
$$

$$[Z] = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ -96.21 \\ 0.735 \end{bmatrix}$$

Example

Set $[U][X] = [Z]$

$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ -96.21 \\ 0.735 \end{bmatrix}$$

Solve for $[X]$

The 3 equations become

$$25a_1 + 5a_2 + a_3 = 106.8$$

$$-4.8a_2 - 1.56a_3 = -96.21$$

$$0.7a_3 = 0.735$$

Example

From the 3rd equation

$$0.7a_3 = 0.735$$

$$a_3 = \frac{0.735}{0.7}$$

$$a_3 = 1.050$$

Substituting in a_3 and using the second equation

$$-4.8a_2 - 1.56a_3 = -96.21$$

$$a_2 = \frac{-96.21 + 1.56a_3}{-4.8}$$

$$a_2 = \frac{-96.21 + 1.56(1.050)}{-4.8}$$

$$a_2 = 19.70$$

Example

Substituting in a_3 and a_2 using the first equation

$$25a_1 + 5a_2 + a_3 = 106.8$$

$$a_1 = \frac{106.8 - 5a_2 - a_3}{25}$$

$$= \frac{106.8 - 5(19.70) - 1.050}{25}$$

$$= 0.2900$$

Hence the Solution Vector is:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0.2900 \\ 19.70 \\ 1.050 \end{bmatrix}$$

LU Decomposition

In the **LU decomposition** post, it was introduced that LU decomposition is a matrix decomposition method obtained using the basic row operations used in performing Gaussian elimination.

However, even if we assume that matrix A is decomposed into the product of a lower triangular matrix and an upper triangular matrix, we can still obtain the result of LU decomposition as it is.

Let's consider that the matrix A of arbitrary size 3×3 can be decomposed into the following form.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{11}l_{21} & u_{12}l_{21} + u_{22} & u_{13}l_{21} + u_{23} \\ u_{11}l_{31} & u_{12}l_{31} + u_{22}l_{32} & u_{13}l_{31} + u_{23}l_{32} + u_{33} \end{bmatrix} \tag{2}$$

Based solely on these results, it can be seen that u_{11}, u_{12}, and u_{13} are the same as a_{11}, a_{12}, and a_{13}, respectively, and the internal values of the next row can be obtained from the calculated u_{11}, u_{12}, and u_{13}. In this way, the elements of L and U can be sequentially obtained.

LU factorization of symmetric matrices?

For symmetric matrices[1], LU factorization can also be considered in the following way.

If A is a symmetric matrix, **could** it be factored as follows?

$$A = LL^T = L^T L \tag{3}$$

Because a symmetric matrix satisfies $A = A^T$, we could write $(LL^T)^T = LL^T$, and since L^T is an upper triangular matrix, we might obtain a result similar to the LU decomposition.

However, just because A is a symmetric matrix, it doesn't always mean that A can be factored as $A = LL^T = L^T L$. Let's think about what characteristics any L must have.

Consider the product of the matrix L and an arbitrary vector x, Lx. The L2-norm value of this Lx vector is always greater than or equal to 0.

And the L2-norm can also be calculated by inner product, which can be written as follows:

$$|Lx|^2 = (Lx)^T (Lx) \tag{4}$$

By the properties of the transpose operator, we can rearrange this as follows:

$$x^T L^T L x \tag{5}$$

LU factorization of symmetric matrices?

And by grouping the parentheses around $L^T L$, we have:

$$x^T (L^T L) x \tag{6}$$

And if we let $(L^T L)$ be some matrix A,

$$x^T A x \tag{7}$$

and since this calculation comes from a method of calculating the L2-norm of an arbitrary vector Lx, we have

$$x^T A x \geq 0 \tag{8}$$

We call a matrix that satisfies the property $x^T A x \geq 0$ a semi-positive definite matrix. In other words, the matrix that satisfies $A = LL^T = L^T L$ must be semi-positive definite.

Assuming that matrix A is a square matrix, symmetric, and semi-positive definite, it can be factorized as $A = LL^T = L^T L$ using Cholesky factorization method.

Calculation of Cholesky factorization

Cholesky factorization can be computed in a similar context as in the previous method of LU decomposition.

Assuming that matrix A is symmetric and semi-positive definite, it can be factorized as follows.

$$A = LL^T = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} L_{11} & L_{21} & L_{31} \\ 0 & L_{22} & L_{32} \\ 0 & 0 & L_{33} \end{bmatrix} \tag{9}$$

When calculating the matrix product, the following result can be obtained:

$$\Rightarrow \begin{bmatrix} L_{11}^2 & & (\text{symmetric}) \\ \\ L_{21}L_{11} & L_{21}^2 + L_{22}^2 & \\ \\ L_{31}L_{11} & L_{31}L_{21} + L_{32}L_{22} & L_{31}^2 + L_{32}^2 + L_{33}^2 \end{bmatrix} \tag{10}$$

Calculation of Cholesky factorization

By comparing the elements of matrix A to the above calculation result one-to-one, it can be seen that they can be organized as follows:

$$L = \begin{bmatrix} \sqrt{a_{11}} & 0 & 0 \\ a_{21}/L_{11} & \sqrt{a_{22} - L_{21}^2} & 0 \\ a_{31}/L_{11} & (a_{32} - L_{31}L_{21})/L_{22} & \sqrt{a_{33} - L_{31}^2 - L_{32}^2} \end{bmatrix} \tag{11}$$

It is also possible to consider generalizing this pattern as follows.

$$L_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} L_{jk}^2} \tag{12}$$

$$L_{ij} = \frac{1}{L_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk} \right) \quad \text{for } i > j \tag{13}$$

QR Decomposition

QR decomposition is a process of decomposing a matrix using orthonormal basis vectors found by the Gram-Schmidt process.

Let the orthonormal basis vectors obtained through the Gram-Schmidt process be denoted by $q_1, \cdots, q_n$, and let the matrix that collects them be denoted by Q. Then the following holds:

$$A = QR \tag{13}$$

$$\begin{bmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} a_1 \cdot q_1 & a_2 \cdot q_1 & \cdots & a_n \cdot q_1 \\ a_1 \cdot q_2 & a_2 \cdot q_2 & \cdots & a_n \cdot q_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1 \cdot q_n & a_2 \cdot q_n & \cdots & a_n \cdot q_n \end{bmatrix} \tag{14}$$

If we consider $a_1 \cdot q_2$, the value is 0 since q_2 has removed all components of a_1 and q_1.

For the same reason, if $i < j$, then $a_i \cdot q_j = 0$. This is because q_j has removed all components of a_i where $i < j$.

Therefore, the following equation holds:

$$= \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} a_1 \cdot q_1 & a_2 \cdot q_1 & \cdots & a_n \cdot q_1 \\ 0 & a_2 \cdot q_2 & \cdots & a_n \cdot q_2 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \cdot q_n \end{bmatrix} \tag{15}$$

This is called QR decomposition.

Example Problem

It can be difficult to understand the explanation of QR decomposition in words alone, so let's practice the Gram-Schmidt normalization process and QR decomposition through the example below.

Problem: Perform QR decomposition of the matrix A below.

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{16}$$

Let a_1, a_2, and a_3 be the column vectors of matrix A, as follows:

$$a_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, a_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, a_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \tag{17}$$

For convenience, we will denote each column vector of A as $a_1 = (1, 1, 0)$, etc.

To perform QR decomposition, let's apply the Gram-Schmidt process to the three vectors.

Let's denote the vectors obtained by orthogonalization but not normalization as u_1, u_2, etc., and the normalized orthonormal vectors as e_1, e_2, etc.

Example (Contd.)

First, consider a_1:

$$a_1 = (1, 1, 0) \tag{18}$$

According to the Gram-Schmidt process, we can use the first vector as it is.

$$u_1 = a_1 = (1, 1, 0) \tag{19}$$

Now let's calculate u_2. u_2 is the vector obtained by subtracting the component of a_2 in the direction of u_1 from a_2.

$$u_2 = a_2 - \mathrm{proj}_{u_1} a_2 \tag{20}$$

$$= (1, 0, 1) - \left(\frac{u_1 \cdot a_2}{u_1 \cdot u_1} \right) u_1 \tag{21}$$

$$= (1, 0, 1) - \frac{1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1}{1^2 + 1^2 + 0^2} (1, 1, 0) \tag{22}$$

$$= \left(\frac{1}{2}, -\frac{1}{2}, 1 \right) \tag{23}$$

Let's also calculate u_3. u_3 is the vector obtained by subtracting the component of a_3 in the direction of u_1 and u_2 from a_3.

$$u_3 = a_3 - \mathrm{proj}_{u_1} a_3 - \mathrm{proj}_{u_2} a_3 \tag{24}$$

$$= (0, 1, 1) - \left(\frac{u_1 \cdot a_3}{u_1 \cdot u_1} \right) u_1 - \left(\frac{u_1 \cdot a_3}{u_2 \cdot u_2} \right) u_2 \tag{25}$$

Example (Contd.)

$$= (0,1,1) - \left(\frac{0+1+0}{1^2+1^2+0^2}\right)(1,1,0) - \left(\frac{(-1/2+1)}{1/4+1/4+1}\right)\left(\frac{1}{2},-\frac{1}{2},1\right) \tag{26}$$

$$(0,1,1) - \frac{1}{2}(1,1,0) - \frac{1}{3}\left(\frac{1}{2},-\frac{1}{2},1\right) \tag{27}$$

$$= \left(-\frac{2}{3},\frac{2}{3},\frac{2}{3}\right) \tag{28}$$

To summarize, the vectors u_1, u_2, u_3 are as follows:

$$u_1 = (1,1,0), u_2 = \left(\frac{1}{2},-\frac{1}{2},1\right), u_3 = \left(-\frac{2}{3},\frac{2}{3},\frac{2}{3}\right) \tag{29}$$

Normalizing the above three vectors, we get:

$$e_1 = \frac{u_1}{\|u_1\|} = \frac{1}{\sqrt{2}}(1,1,0) = \left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}},0\right) \tag{30}$$

$$e_2 = \frac{u_2}{\|u_2\|} = \sqrt{\frac{2}{3}}\left(\frac{1}{2},-\frac{1}{2},1\right) = \left(\frac{1}{\sqrt{6}},-\frac{1}{\sqrt{6}},\frac{2}{\sqrt{6}}\right) \tag{31}$$

$$e_3 = \frac{u_3}{\|u_3\|} = \frac{1}{\sqrt{3\cdot(2/3)^2}}\left(-\frac{2}{3},\frac{2}{3},\frac{2}{3}\right) = \left(-\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right) \tag{32}$$

Therefore, we can perform QR decomposition as follows, considering e_1, e_2, and e_3 as corresponding to q_1, q_2, and q_3 in $A = QR$.

Example (Contd.)

$$A = QR = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 0 & 2/\sqrt{6} & 1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 2/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 3/\sqrt{6} & 1/\sqrt{6} \\ 0 & 0 & 2/\sqrt{3} \end{bmatrix} \tag{33}$$

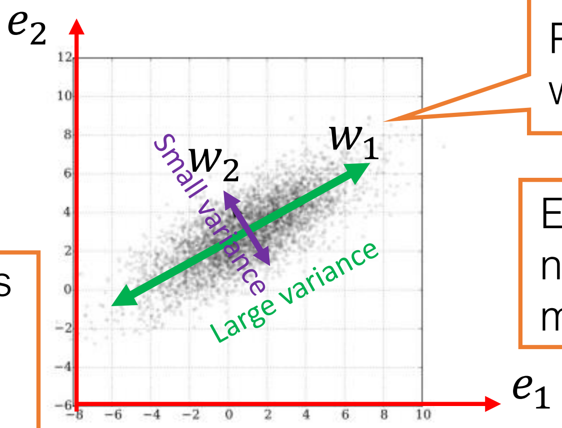
Here, Q is an orthonormal matrix and R is an upper triangular matrix.

Principal Component Analysis (PCA)

- A classic linear dimension reduction method
- Can be seen as
  - Learning directions (co-ordinate axes) that capture maximum variance in data

e_1, e_2: Standard co-ordinate axis ($\boldsymbol{x} = [x_1, x_2]$)

w_1, w_2: New co-ordinate axis ($\boldsymbol{z} = [z_1, z_2]$)



PCA is essentially doing a change of axes in which we are representing the data

Each input will still have 2 co-ordinates, in the new co-ordinate system, equal to the distances measured from the new origin

To reduce dimension, can only keep the co-ordinates of those directions that have largest variances (e.g., in this example, if we want to reduce to one-dim, we can keep the co-ordinate z_1 of each point along w_1 and throw away z_2). We won't lose much information

  - Learning projection directions that result in smallest reconstruction error

$$\underset{\boldsymbol{W},\boldsymbol{Z}}{\operatorname{argmin}} \sum_{n=1}^{N} ||\boldsymbol{x}_n - \boldsymbol{W}\boldsymbol{z}_n||^2 = \underset{\boldsymbol{W},\boldsymbol{Z}}{\operatorname{argmin}} ||\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}||^2$$

Subject to orthonormality constraints: $\boldsymbol{w}_i^\top \boldsymbol{w}_j = 0$ for $i \neq j$ and $||\boldsymbol{w}_i||^2 = 1$

- PCA also assumes that the projection directions are orthonormal

Principal Component Analysis: the algorithm

- Center the data (subtract the mean $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n$ from each data point)
- Compute the $D \times D$ covariance matrix $\mathbf{S}$ using the centered data matrix $\mathbf{X}$ as

$$\mathbf{S} = \frac{1}{N}\mathbf{X}^\top \mathbf{X}$$

(Assuming $\mathbf{X}$ is arranged as $N \times D$)

- Do an eigendecomposition of the covariance matrix $\mathbf{S}$ (many methods exist)
- Take top $K < D$ leading eigvectors $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K\}$ with eigvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$
- The K-dimensional projection/embedding of each

$$\mathbf{z}_n \approx \mathbf{W}_K^\top \boldsymbol{x}_n$$

$\mathbf{W}_\mathrm{K} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K]$ is the "projection matrix" of size $D \times K$

Note: Can decide how many eigvecs to use based on how much variance we want to campure (recall that each λ_k gives the variance in the k^{th} direction (and their sum is the total variance)

Singular Value Decomposition

Singular Value Decomposition

- ***Singular value decomposition*** (SVD) provides another way to factorize a matrix, into singular vectors and singular values
  - SVD is more generally applicable than eigen decomposition
  - Every real matrix has an SVD, but the same is not true of the eigen decomposition
    - E.g., if a matrix is not square, the eigen decomposition is not defined, and we must use SVD
- SVD of an $m \times n$ matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^{\boldsymbol{T}}$$

  - $\mathbf{U}$ is an $m \times m$ matrix, $\mathbf{D}$ is an $m \times n$ matrix, and $\mathbf{V}$ is an $n \times n$ matrix
  - The elements along the diagonal of $\mathbf{D}$ are known as the singular values of A
  - The columns of $\mathbf{U}$ are known as the left-singular vectors
  - The columns of $\mathbf{V}$ are known as the right-singular vectors
- For a non-square matrix $\mathbf{A}$, the squares of the singular values σ_i are the eigenvalues λ_i of $\mathbf{A}^{\boldsymbol{T}}\mathbf{A}$, i.e., $\sigma_i^2 = \lambda_i$ for $i = 1,2, \ldots, n$
- Applications of SVD include computing the pseudo-inverse of non-square matrices, matrix approximation, determining the matrix rank

THANK YOU