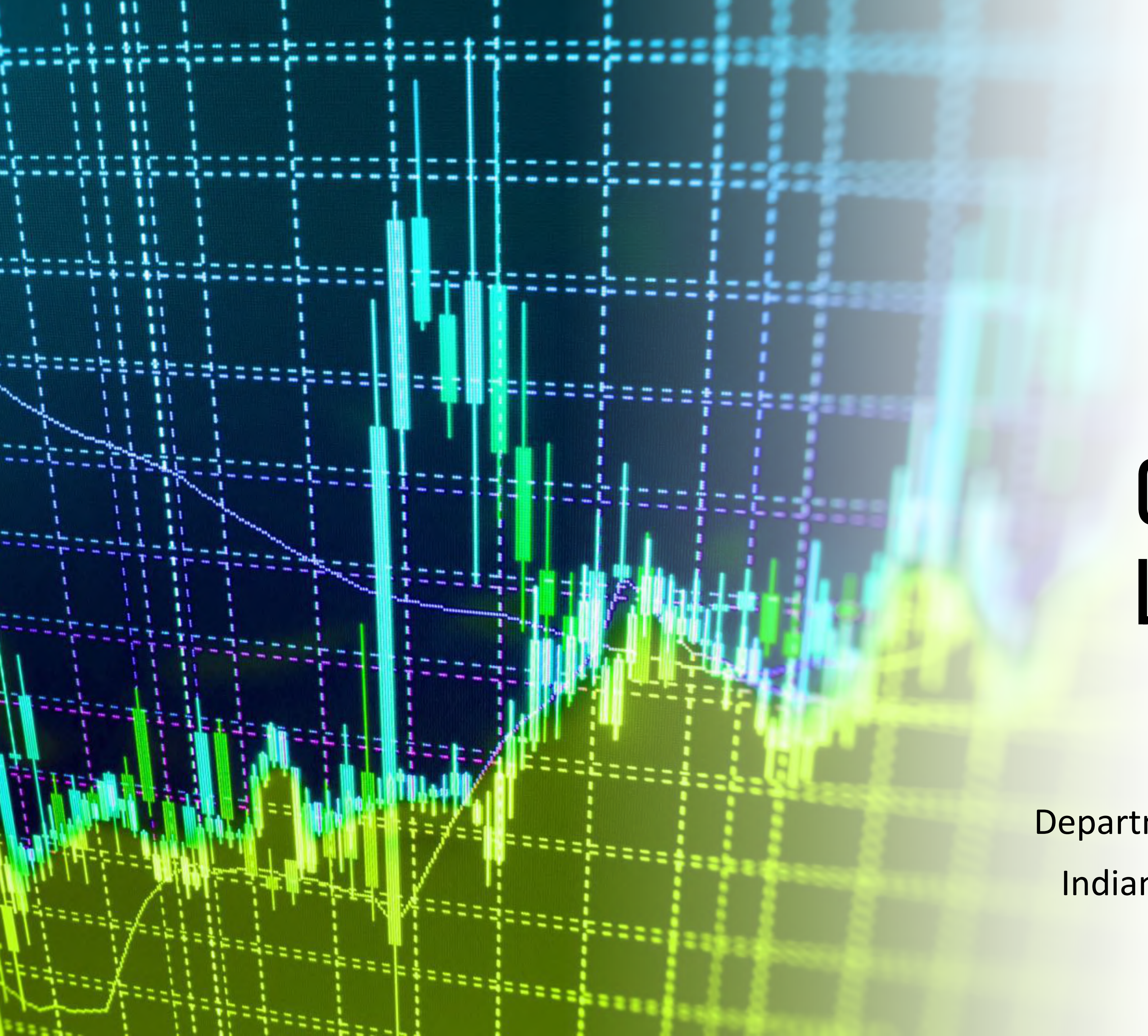
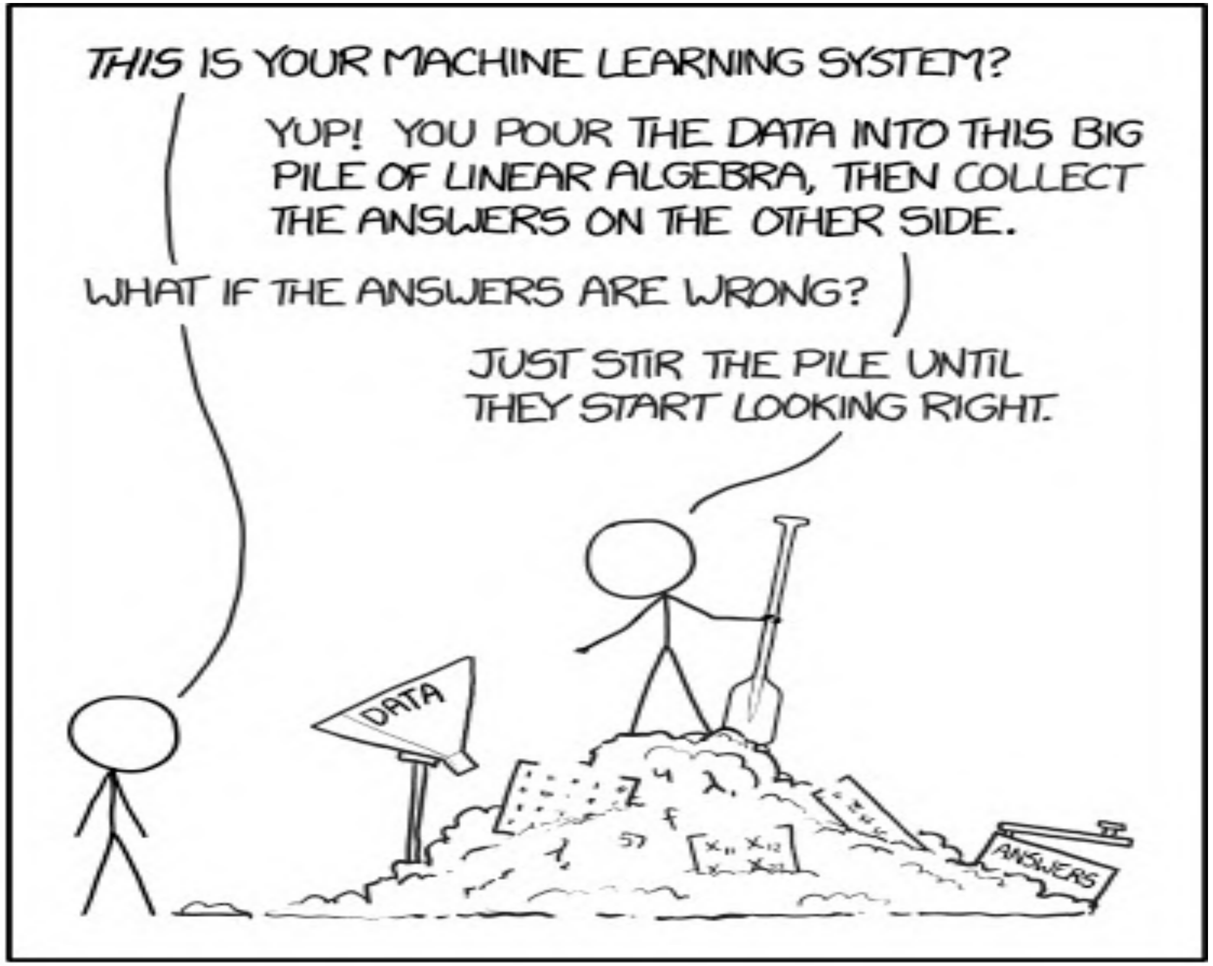


Computational Linear Algebra

Prof. Phani Motamarri
Department of Computational and Data Sciences
Indian Institute of Science, Bangalore-560012

Computational Linear Algebra in Data Science

Why Linear Algebra ?



https://imgs.xkcd.com/comics/machine_learning.png

Course Outline:

1. Properties of Vectors, Matrices ,Tensors and the operations between them.
2. Algorithms in Computational Linear Algebra that are frequently encountered in Data Science/ML

Learning Objectives:

1. Understand foundational aspects of vectors, Matrices, operations between them and their relevance to Data Science.
2. Be able to understand the theory behind the many of the ML algorithms from a linear algebra perspective.

Least Square Problems

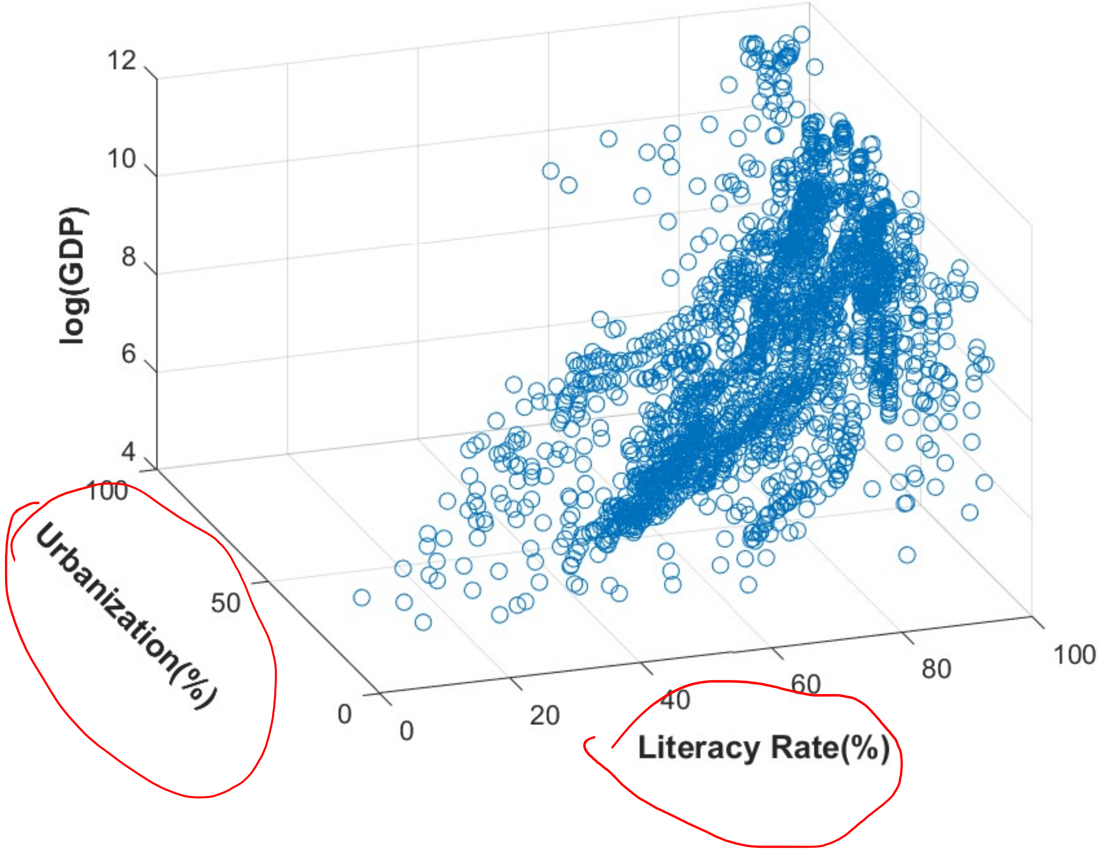
Least Square Problems

Motivation

$\cancel{y = w_0 + w_1 x}$

$y = w_0 + w_1 x^1 + w_2 x^2$

Recall.. The GDP problem.. We would like to introduce another parameter Literacy Rate



$$\varepsilon = \frac{1}{2}\sum_{i=1}^{i=N}\left(y_i - w^2 x_i^2 - w^1 x_i^1 - w^0\right)^2$$

Minimising the error

$$\frac{d\varepsilon}{dw^2} = \sum_{i=1}^{i=N}\left(y_i - w^2 x_i^2 - w^1 x_i^1 - w^0\right)x_i^2 = 0$$

$$\frac{d\varepsilon}{dw^1} = \sum_{i=1}^{i=N}\left(y_i - w^2 x_i^2 - w^1 x_i^1 - w^0\right)x_i^1 = 0$$

$$\frac{d\varepsilon}{dw^0} = \sum_{i=1}^{i=N}\left(y_i - w^2 x_i^2 - w^1 x_i^1 - w^0\right) = 0$$

Least Square Problems

Motivation

Consider other inputs such as company credit, energy use. Lets build the model for GDP

Let $\boldsymbol{x}_i$ be the input vector for a single data point.

$$\boldsymbol{x}_i = \left(\boxed{x_i^1}\ \boxed{x_i^2}\ \boxed{x_i^3}\ \boxed{x_i^4}\right)$$

URB%

LIT%

CREDIT

ERG

BUT…

$$y_i \approx w^1 x_i^1 + w^2 x_i^2 + w^3 x_i^3 + w^4 x_i^4 + w^0$$

$n \times (m+1)$

$y_i = (\qquad) + \varepsilon$

Least Square Problems

Equation

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & x_1^4 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^1 & x_m^2 & x_m^3 & x_m^4 & 1 \end{bmatrix} [w^1 \quad w^2 \quad w^3 \quad w^4 \quad w^0]^T + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_m \end{pmatrix}$$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\varepsilon}$$

Size of ε

2

Aim: To find the best set of coefficients to map the features and target.

This will be $\min ||\boldsymbol{\varepsilon}||_2$

$\sqrt{\varepsilon \cdot \varepsilon}$

$||\varepsilon||_2$

$$s = \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$$s = \boldsymbol{y}^T\boldsymbol{y} - 2(\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{X}\boldsymbol{w})$$

$||\varepsilon|| \geq 0$

$||\varepsilon|| = 0$

$\varepsilon = 0$

Taking derivative and setting to zero for minimization:

$$\frac{\partial s}{\partial \boldsymbol{w}} = -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{0}$$

$$\boxed{\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}}$$ → Normal Equation

Least Square Problems

Equation

$$s_1 = (\boldsymbol{Xw})^T \boldsymbol{y}$$

$$s_1 = \begin{bmatrix} \sum_{i=0}^{4} x_1^i w^i \\ \vdots \\ \sum_{i=0}^{4} x_m^i w^i \end{bmatrix}^T \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \sum_{j=1}^{m} y_j \sum_{i=0}^{4} x_j^i w^i$$

$$\frac{\partial s_1}{\partial \boldsymbol{w}} = \begin{pmatrix} \frac{\partial s_1}{\partial w^1} \\ \vdots \\ \frac{\partial s_1}{\partial w^0} \end{pmatrix} = \begin{bmatrix} \frac{\partial \sum_{j=1}^{m} \sum_{i=0}^{4} y_j x_j^i w^i}{\partial w^1} \\ \vdots \\ \frac{\partial \sum_{j=1}^{m} \sum_{i=0}^{4} y_j x_j^i w^i}{\partial w^0} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{m} y_j x_j^1 \\ \vdots \\ \sum_{j=1}^{m} y_j x_j^0 \end{bmatrix} = \boldsymbol{X}^T y$$

Least Square Problems

Equation

$$s_2 = (\boldsymbol{Xw})^T(\boldsymbol{Xw}) = \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{Xw}$$

$$s_2 = [w^1 \quad w^2 \quad w^3 \quad w^4 \quad w^0]\boldsymbol{X}^T\begin{pmatrix}\sum_{i=0}^4 x_1^i w^i \\ \vdots \\ \sum_{i=0}^4 x_m^i w^i\end{pmatrix} = [w^1 \quad w^2 \quad w^3 \quad w^4 \quad w^0]\begin{bmatrix}\sum_{j=1}^m x_j^1 \sum_{i=0}^4 x_j^i w^i \\ \vdots \\ \sum_{j=1}^m x_j^0 \sum_{i=0}^4 x_j^i w^i\end{bmatrix}$$

$$s_2 = w^1\left(\sum_{j=1}^m \sum_{i=0}^4 x_j^1 x_j^i w^i\right) + \ldots + w^4\left(\sum_{j=1}^m \sum_{i=0}^4 x_j^4 x_j^i w^i\right) + w^0\left(\sum_{j=1}^m \sum_{i=0}^4 x_j^0 x_j^i w^i\right)$$

$$\frac{\partial s_2}{\partial w^1} = \left\{\left(\sum_{j=1}^m \sum_{i=0}^4 x_j^1 x_j^i w^i\right) + x_j^1 x_j^1\right\} + \sum_{j=1}^m \{\ldots + w^4 x_j^4 x_j^1 + w^0 x_j^0 x_j^1\} = 2\sum_{j=1}^m \sum_{i=0}^4 x_j^1 x_j^i w^i$$

$$\frac{\partial s_2}{\partial \boldsymbol{w}} = \begin{pmatrix}\frac{\partial s_2}{\partial w^1} \\ \vdots \\ \frac{\partial s_2}{\partial w^0}\end{pmatrix} = 2\boldsymbol{X}^T\boldsymbol{X}w$$

Least Square Problems

Equation

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & x_1^4 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^1 & x_m^2 & x_m^3 & x_m^4 & 1 \end{bmatrix} [w^1 \quad w^2 \quad w^3 \quad w^4 \quad w^0]^T + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_m \end{pmatrix}$$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\varepsilon}$$

Aim: To find the best set of coefficients to map the features and target.

This will be $\min ||\boldsymbol{\varepsilon}||_2$

$$s = \boldsymbol{\varepsilon}^{\boldsymbol{T}}\boldsymbol{\varepsilon} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$$s = \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{2}(\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{y} + (\boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{X}\boldsymbol{w})$$

Taking derivative and setting to zero for minimization:

$$\boxed{\frac{\partial s}{\partial \boldsymbol{w}} = -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{0}}$$

$$\boxed{\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}} \longrightarrow \text{Normal Equation}$$

Least Square Problems

Solution Procedure

Rewriting the Normal equation

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$$

where,

- $\boldsymbol{X}$ is the input data $m \times (n+1)$ matrix with m data points and n features
- $\boldsymbol{y}$ is the output/dependent data
- $\boldsymbol{w}$ is the parameters that are used to fit the model

This equation is generally solved by appropriate matrix decompositions of $\boldsymbol{X}^T\boldsymbol{X}$

Matrix Decompositions

QR Factorization

Introduction

$A \in \mathbb{R}^{m\times n}$

→ m rows

→ n columns

$Q \in \mathbb{R}^{p\times p}$

$$\boldsymbol{A} = \boldsymbol{QR}$$

- $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a full rank matrix, $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ is an ~~orthogonal~~ matrix and $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is an Upper Triangular matrix

with orthogonal columns

$A_{m\times n} = Q_{m\times n} R_{n\times n}$

- <u>Applications:</u>
  - Popular technique to solve least squares problem.
  - Useful for solving eigenvalue problems.
  - Form successive orthonormal vectors $\boldsymbol{q}_j$ that span the successive columns spaces of $\boldsymbol{A}$. $\langle \boldsymbol{a}_1\, \boldsymbol{a}_2\, \boldsymbol{a}_3 \ldots \boldsymbol{a}_n\rangle = \langle \boldsymbol{q}_1\, \boldsymbol{q}_2\, \boldsymbol{q}_3 \ldots \boldsymbol{q}_n\rangle$.

QR Factorization

$q_1 = \frac{a_1}{\|a_1\|}$ $q_2 = \frac{v}{\|v\|}$

Introduction

Example:

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 0.894 & -0.4472 \\ 0.4472 & 0.8944 \end{pmatrix} \begin{pmatrix} 2.2361 & 1.7889 \\ 0 & 1.3416 \end{pmatrix}$$

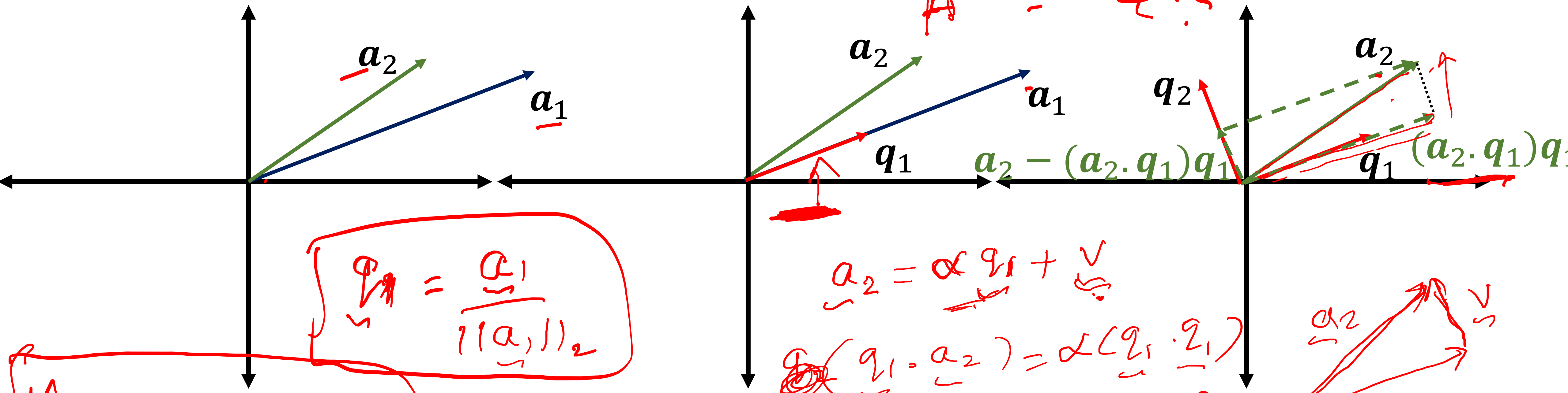
a_1 a_2

$$[a_1 \ \ a_2] = [q_1 \ \ q_2] \begin{bmatrix} \|a_1\| & a_2 \cdot q_1 \\ 0 & \|v\| \end{bmatrix}$$

$$A = QR$$

- Geometric Intuition:

$\boldsymbol{A} \in \mathbb{R}^{2\times 2}, \boldsymbol{A} = [\boldsymbol{a}_1 \quad \boldsymbol{a}_2]$



$$q_1 = \frac{a_1}{\|a_1\|_2}$$

$$a_2 = \alpha q_1 + v$$

$$(q_1 \cdot a_2) = \alpha (q_1 \cdot q_1) + \underbrace{(v \cdot q_1)}_{\to 0}$$

$$\alpha = q_1 \cdot a_2 = a_2 \cdot q_1$$

a_2, v, αq_1

$$\boxed{v = a_2 - (a_2 \cdot q_1) q_1}$$

$$\boxed{q_2 = \frac{v}{\|v\|}}$$

$\|v\|$

$$\alpha = \frac{q_1 \cdot a_2}{q_1 \cdot q_1}$$

QR Factorization

Method

$$(\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \boldsymbol{a}_3) = (\boldsymbol{q}_1 \quad \boldsymbol{q}_2 \quad \boldsymbol{q}_3)\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix}$$

$$\boldsymbol{A} = \boldsymbol{QR}$$

$$\boldsymbol{a}_1 = r_{11}\boldsymbol{q}_1$$

$$\boldsymbol{a}_2 = r_{12}\boldsymbol{q}_1 + r_{22}\boldsymbol{q}_2$$

$$\boldsymbol{a}_3 = r_{13}\boldsymbol{q}_1 + r_{23}\boldsymbol{q}_2 + r_{33}\boldsymbol{q}_3$$

$q_i^T a_j$

$m \times 1$ $\quad$ $m \times 1$

$O(m)$

m data pts & n features

$O(mn^2)$

The j^th orthonormal vector is found as below:

$$\boxed{\boldsymbol{q}_j = \frac{\boldsymbol{a}_j - \sum_{i=1}^{i=j-1} r_{ij}\boldsymbol{q}_j}{r_{jj}}}$$

$$r_{ij} = \boldsymbol{q}^T{}_i\boldsymbol{a}_j \ (i \neq j)$$

$$r_{jj} = ||\boldsymbol{a}_j - \sum_{i=1}^{i=j-1} r_{ij}\boldsymbol{q}_j||$$

$\boldsymbol{q}_j$ orthonormal $\langle \boldsymbol{q}_1\boldsymbol{q}_2\boldsymbol{q}_3 \ldots \boldsymbol{q}_{j-1}\rangle$

$O(mn^2)$

Computational Cost $\sim 2(num.rows) * (num.columns)^2$

QR Factorization

Solving Least Square Problem

$$X^T X w = X^T y$$
$$\to \hat{R}^T \hat{Q}^T \hat{Q} \hat{R} w = \hat{R}^T \hat{Q}^T y$$
$$\boxed{\hat{R} w = \hat{Q}^T y}$$

- <u>Algorithm:</u>
  1. Compute the QR factorization of $X = \hat{Q}\hat{R}$
  2. Compute the vector $\hat{Q}^T y$
  3. Solve the upper triangular system $\hat{R}w = \hat{Q}^T y$

- Total computation cost $\boxed{2mn^2 - {}^2/_3\, n^3}$ flops
- QR factorization is the costly step

$$\boxed{X^T X w = X^T y}$$
$$X = \hat{Q}\hat{R}$$
$$X^T = (\hat{Q}\hat{R})^T$$
$$= \hat{R}^T \hat{Q}^T$$
$$\boxed{R w = Q^T y}$$
$$\begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$
$$w_2 = \frac{y_2}{f}$$

Singular Value Decomposition

Introduction

AV

m data pts

n feature

A

$m > n$

$A = U\Sigma V^T$

Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be a matrix such that $m \geq n$

$$\boxed{\boldsymbol{A} = \boldsymbol{U\Sigma V}^T}$$

$\boldsymbol{U} \in \mathbb{R}^{m\times n}$ matrix with orthonormal columns, $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, a diagonal matrix, $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ orthogonal matrix

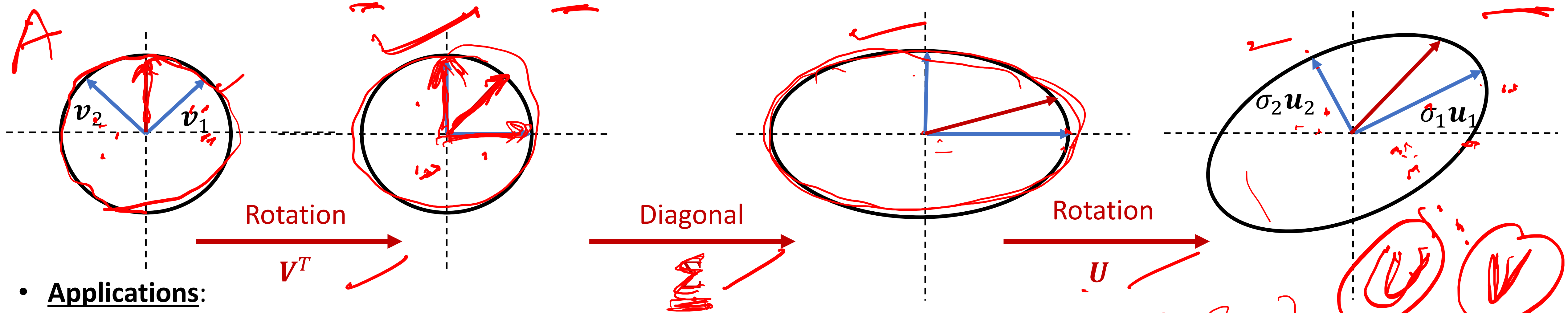
$$\boxed{\boldsymbol{A}\boldsymbol{v}_j = \sigma_j \boldsymbol{u}_j}$$

$A = U\Sigma \boxed{V^T}$

Geometric Intuition:

- Action of any matrix $\boldsymbol{A}$ on a unit circle $\boldsymbol{S}$: **Rotation ->Stretching** of unit circle to ellipse->**Rotation** of the ellipse

AS



- **Applications**:
  - **Searching-** Closest related images or documents
  - **Compression-** Reducing image size, **Image Recovery**
  - **Principle Component Analysis-** Finding the most representative features

$A v_j = \sigma_j u_j \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Singular Value Decomposition

$AV = 0$

Introduction

$$\begin{bmatrix} & \boldsymbol{A} & \end{bmatrix} [\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \cdots \quad \boldsymbol{v}_n] = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_m] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & & & & 0 \end{bmatrix}$$

- $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m\}$ are the m left singular unit vectors along the principle axis transformed d-ellipse
- $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$, right singular vectors of $\boldsymbol{A}$,
- $\sigma_1, \sigma_2, \sigma_3 \ldots \sigma_n$ are the length of semi principle axis of $\boldsymbol{AS}$ such that $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_n \geq 0$
- Existence & Uniqueness: Every matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ has a singular value decomposition. Furthermore, the singular values $\{\sigma_j\}$ are uniquely determined and, if $\boldsymbol{A}$ is square and σ_j are distinct, the left singular vectors $\{\boldsymbol{u}_j\}$ and right singular vectors $\{\boldsymbol{v}_j\}$ are uniquely determined.

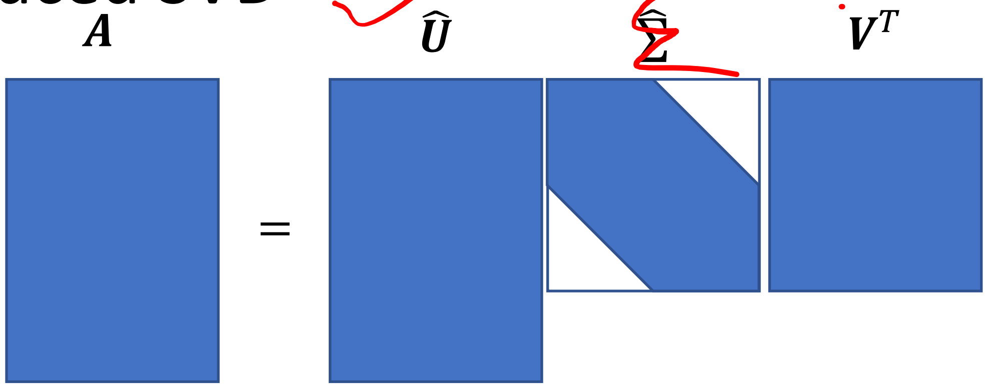
Singular Value Decomposition

$A v_j = \sigma_j u_j$

A S $y = Ax$ AS

Reduced SVD & Full SVD

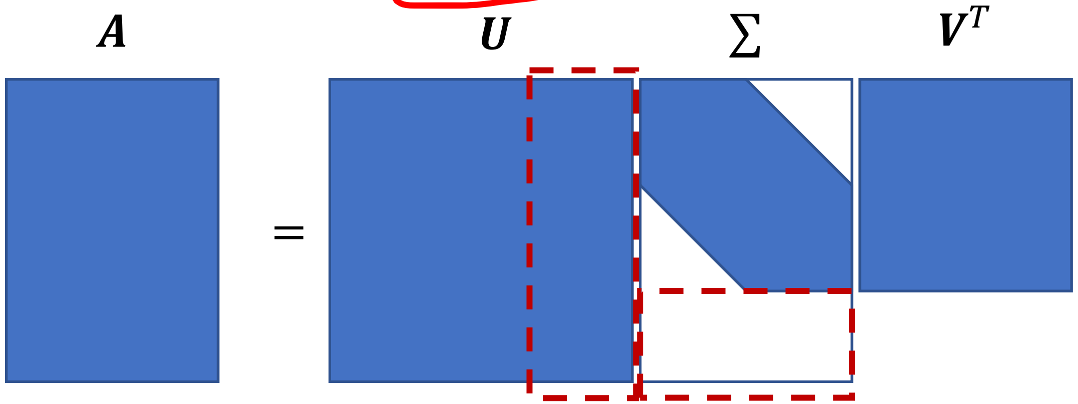
Reduced SVD



$$A = \hat{U}\hat{\Sigma}V^T$$

$\hat{\Sigma}$ is a $n \times n$ diagonal matrix, $\hat{\boldsymbol{U}}$ is an $m \times n$ matrix with orthonormal columns, $\boldsymbol{V}$ is $n \times n$ orthonormal matrix

Full SVD



$$A = U\Sigma V^T$$

$\boldsymbol{U}$ is formed by appending $(m - n)$ orthonormal columns to $\hat{\boldsymbol{U}}$. Σ is formed by appending $(m - n)$ rows of 0 to $\hat{\Sigma}$

Singular Value Decomposition

Matrix Properties thru SVD

$A = U\Sigma V^T$

$A v_i = \sigma_i u_i$

- The rank of matrix $\boldsymbol{A}$ is r, number of non-zero singular values.
- $range(\boldsymbol{A}) = \langle \boldsymbol{u}_1\ \boldsymbol{u}_2 \ldots \boldsymbol{u}_r \rangle$ and $null(A) = \langle \boldsymbol{v}_{r+1}\ \boldsymbol{v}_{r+2} \ldots \boldsymbol{v}_n \rangle$
- $||\boldsymbol{A}||_2 = \sigma_1$ and $||\boldsymbol{A}||_F = \sqrt{\sigma_1^2 + \sigma_2^2 \ldots + \sigma_r^2}$
- Non zeros singular values of $\boldsymbol{A}$ are the positive square roots of non-zero eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ or $\boldsymbol{A}\boldsymbol{A}^T$

$||A||_F$

$A v_1 = \sigma_1 u_1$

$A v_2 = \sigma_2 u_2$

$A v_3 = \sigma_3 u_3$

$A v_4 = \sigma_4 u_4$

$\max_x \frac{||Ax||}{||x||}$

$\sigma_1 > 0$

$\sigma_2 > 0$

$\sigma_3 = 0$

$\sigma_4 = 0$

$A v_5 = 0$

$\sigma_5 = 0$

$\sigma_5 = 0, \sigma_4 = 0$

Singular Value Decomposition

Low Rank Approximation

$m \times n$

$n-1 \quad \sigma_n = \|A - B\|_F$

$Av_j = \sigma_j u_j \quad \|A - B\|_F$

$u_j \quad v_j$

$m\times 1 \quad n \times 1$

$u_j v_j^T$

$m \times n$

Left

Right

$$\boxed{A = U\Sigma V^T}$$

$Av_n = \quad Av_j = \sigma_j u_j$

$$A = [u_1 \quad u_2 \quad \cdots \quad u_n]\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n\end{bmatrix}[v_1 \quad v_2 \quad \cdots \quad v_n]$$

$A \sigma_4$

$$A = \sigma_1 u_1 v^T{}_1 + \sigma_2 u_2 v^T{}_2 + \cdots \sigma_r u_r v^T{}_r + \sigma_{r+1} u_{r+1} v^T{}_{r+1} + \cdots \sigma_n u_n v^T{}_n$$

u_{m-1}

$u_1 v_1^T \rightarrow$ Rank 1

$u_2 v_2^T \rightarrow$ Rank 1

$u_3 v_3^T \rightarrow$ Rank 1

$u_4 v_4^T \rightarrow$ Rank 1

$$A_\gamma = \sum_1^\gamma \sigma_j u_j v_j^T, 0 \le \gamma \le r$$

$$\|A - A_\gamma\| = \min(\|A - B\|),$$

$m \times n \quad 2$

$\gamma = 2$

$$A_\gamma = \sum_{j=1}^{2} \sigma_j u_j v_j^T$$

B is any matrix with rank γ

$A_\gamma = m \times n$

Singular Value Decomposition

Image Compression

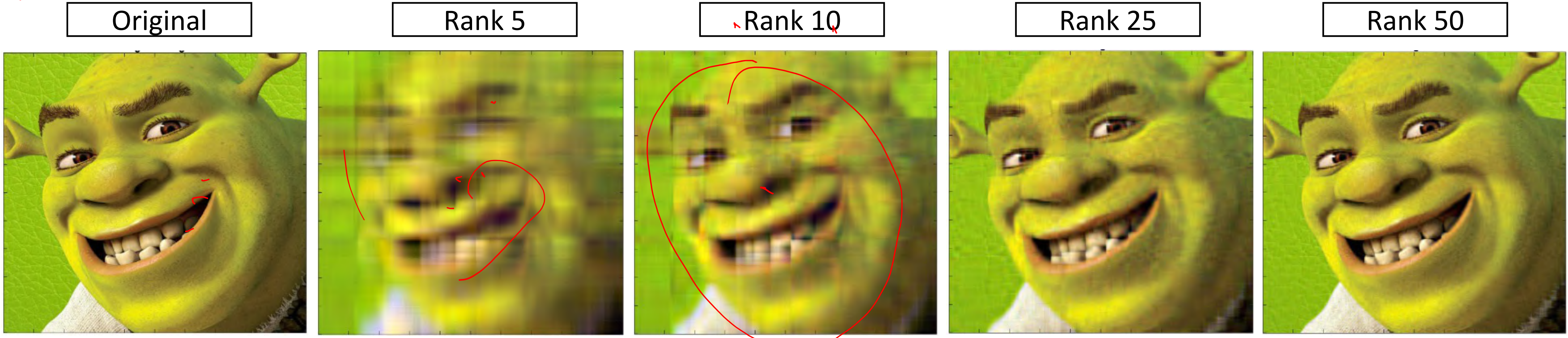
$A = U \Sigma V^T$

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}^T{}_1 + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}^T{}_2 + \cdots \sigma_r \boldsymbol{u}_r \boldsymbol{v}^T{}_r + \sigma_{r+1} \boldsymbol{u}_{r+1} \boldsymbol{v}^T{}_{r+1} + \cdots \sigma_n \boldsymbol{u}_n \boldsymbol{v}^T{}_n$$

$$A \approx \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}^T{}_1 + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}^T{}_2 + \cdots \sigma_r \boldsymbol{u}_r \boldsymbol{v}^T{}_r + \boldsymbol{0}$$

- **Total Saving:** Original A requires $\boldsymbol{m \times n}$ storage , reduced to $\boxed{\boldsymbol{r \times (m + n + 1)}}$

$m \times 1 \rightarrow u_1$ $m \times 1 \rightarrow v_1$ $(m+n+1)$



Singular Value Decomposition

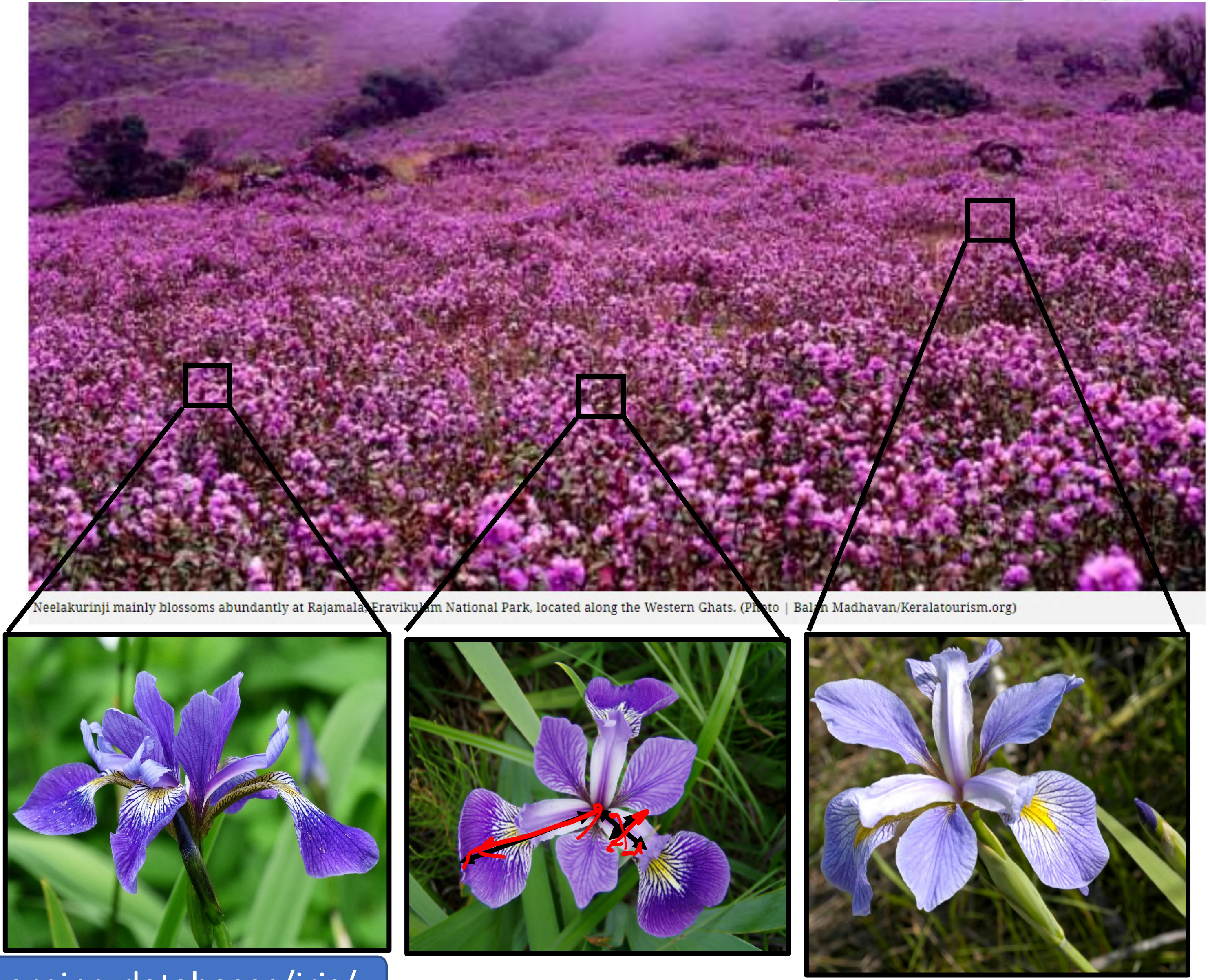
Fischer Iris Dataset

Imagine you are in a valley of Iris flowers.

You observe that there are some difference between the flowers shapes

- sepal length
- sepal width
- petal length
- petal width

How do you classify them?



Neelakurinji mainly blossoms abundantly at Rajamala, Eravikulam National Park, located along the Western Ghats. (Photo | Balan Madhavan/Keralatourism.org)

DataSource: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/

Singular Value Decomposition

SL, SW, PL, PW

Fischer Iris Dataset

150 rows; X (4 columns), y; 50 / 50 / 50

X				y
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor
5.5	2.3	4	1.3	Iris-versicolor
6.5	2.8	4.6	1.5	Iris-versicolor
5.7	2.8	4.5	1.3	Iris-versicolor
6.3	3.3	4.7	1.6	Iris-versicolor
4.9	2.4	3.3	1	Iris-versicolor
6.7	2.5	5.8	1.8	Iris-virginica
7.2	3.6	6.1	2.5	Iris-virginica
6.5	3.2	5.1	2	Iris-virginica
6.4	2.7	5.3	1.9	Iris-virginica
6.8	3	5.5	2.1	Iris-virginica
5.7	2.5	5	2	Iris-virginica
5.8	2.8	5.1	2.4	Iris-virginica

Neelakurinji mainly blossoms abundantly at Rajamala Eravikulam National Park, located along the Western Ghats. (Photo | Balan Madhavan/Keralatourism.org)

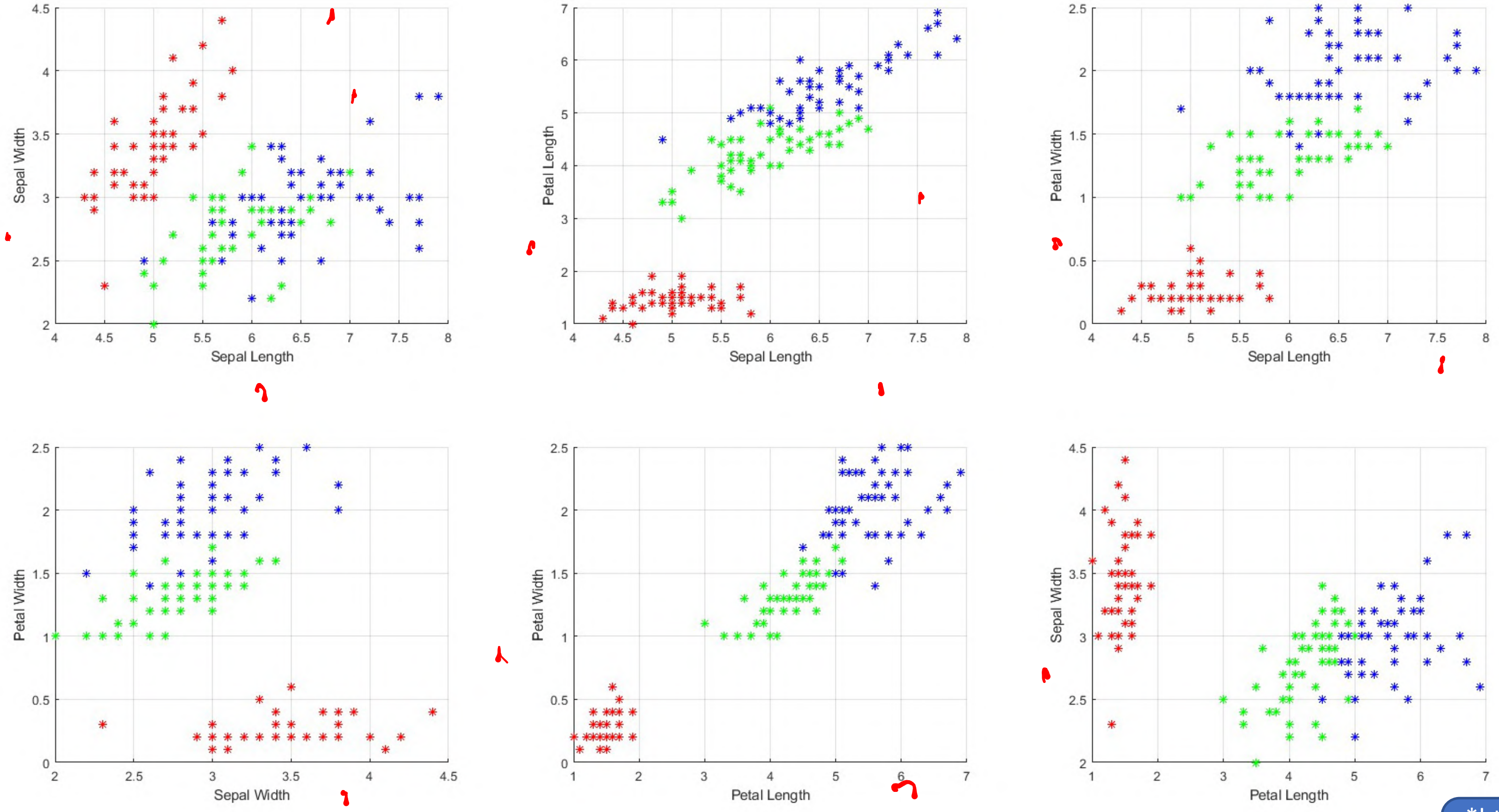
DataSource: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/

Singular Value Decomposition

6

PCA

- Data is of 4 dimensions: Sepal Length, Sepal Width, Petal Length, Petal Width . Aim is to reduce to 2 dimensions, using PCA



A good read: https://arxiv.org/pdf/1404.1100.pdf

*https://en.wikipedia.org/wiki/File:Iris_versicolor_3.jpg
**https://en.wikipedia.org/wiki/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg
***https://en.wikipedia.org/wiki/File:Iris_virginica.jpg

Singular Value Decomposition

X

150×4

$U = [u_1 \; u_2 \; u_3 \; u_4]$

$m \times n$

$\sigma_1 \; \sigma_2 \; \sigma_3 \; \sigma_4$

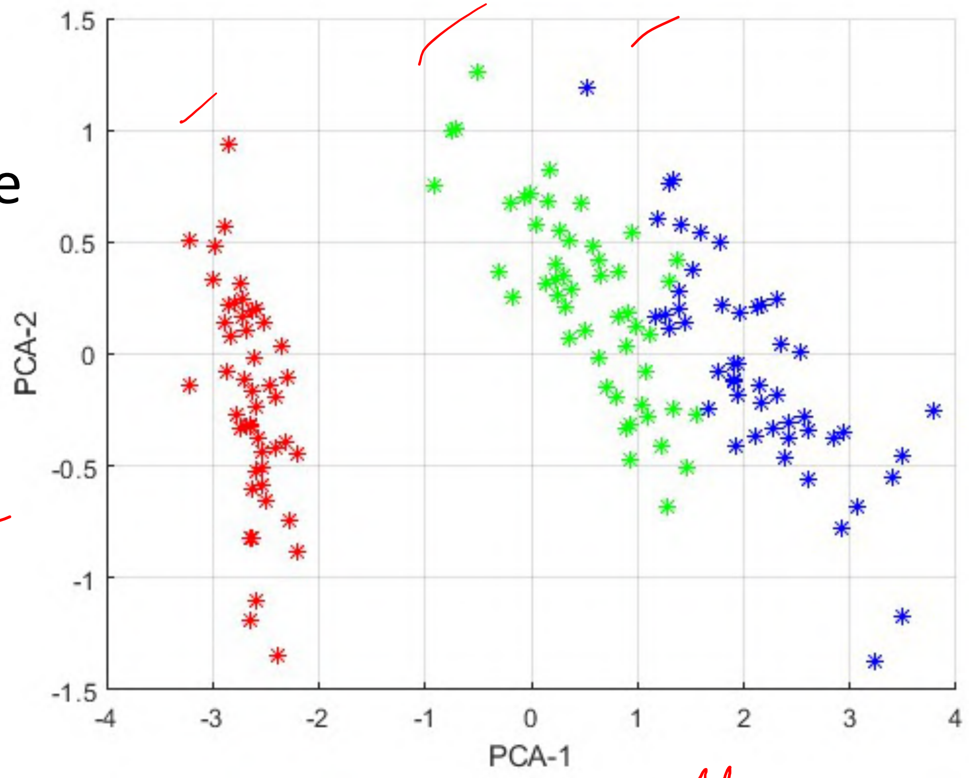
$v_1 \; v_2 \; v_3 \; v_4$

PCA

- <u>Algorithm:</u>
  - Create $\boldsymbol{X}$ formed by using the m-measurements of n-variables
  - Subtract mean for each measurement type.
  - Find the SVD of $\boldsymbol{X}$
  - To find the reduced dimensional representation of the given data we find dominant left singular vectors
    - Since $\widehat{\boldsymbol{U}}$ is orthonormal set of range of $\boldsymbol{X}$, using the first 2 singular vectors we can get the 2d reduced data.

$X =$

u_2



u_1

DataSource: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/

Singular Value Decomposition

Solving Least Square Problem

$X = U\Sigma V^T$

$\Sigma a = U^T y$

$a = \Sigma^{-1} U^T y$

$X^T X w = X^T y$

$$X^T X w = X^T y$$

$$V\hat{\Sigma}^T \hat{U}^T \hat{U} \hat{\Sigma} V^T w = V\hat{\Sigma}^T \hat{U}^T y$$

$V^T w = a$

$w = Va$

- Algorithm:
  1. Compute the reduced SVD of $X = \hat{U}\hat{\Sigma}V^T$
  2. Compute the vector $\hat{U}^T y$
  3. Solve the diagonal system $\hat{\Sigma} a = \hat{U}^T y$
  4. Get $w = Va$
- Total computation cost $\boxed{2mn^2 + 11n^3}$ flops
- The more stable algorithm*

$\Sigma V^T w = U^T y$

Cholesky Decomposition

Introduction

- <u>Symmetric Positive Definite Matrices:</u> Let $\boldsymbol{A} \in \mathbb{R}^{nxn}$ be a symmetric positive matrix then:
  - $a_{ij} = a_{ji}$, this gives $\boldsymbol{A} = \boldsymbol{A}^T$
  - $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > \boldsymbol{0}\ \forall \boldsymbol{x} \in \mathbb{R}^n$

$$\boxed{\boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R},} \qquad \boldsymbol{r}_{jj} > \boldsymbol{0}$$

$\boldsymbol{R}$ is an upper triangle matrix with diagonal terms positive

$$\boxed{A = R^T R}$$

$A \in \mathbb{R}^{n \times n}$

$A = A^T$ ✓

$\boxed{x^T A x > 0}$ ✓ $\forall x \in \mathbb{R}^n$

$x^T x$

Cholesky Decomposition

Solving Least Square Problem

$$\begin{bmatrix} 1 & 0 & 0 \\ 5 & 2 & 0 \\ 6 & 7 & 3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ \\ \\ \end{bmatrix}$$

$$\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{y}$$

$$\boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{R}^T \boldsymbol{R},\ \boldsymbol{R}^T \boldsymbol{R} \boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{y}$$

$$\boldsymbol{R} \boldsymbol{w} = \boldsymbol{a}$$

$QR \rightarrow 2mn^2 - \frac{2}{3}n^3$

$SVD \rightarrow 2mn^2 + 11n^3$

$\text{Cholesky} \rightarrow mn^2 + \frac{1}{3}n^3$

$\kappa = \frac{\sigma_{max}}{\sigma_{min}}$

- Algorithm:
  1. Compute the Cholesky factorization of $\boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{R}^T \boldsymbol{R}$
  2. Solve Lower triangle system $\boldsymbol{R}^T \boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{y}$ (a)
  3. Solve Upper triangle system $\boxed{\boldsymbol{R} \boldsymbol{w} = \boldsymbol{a}}$
- Total computation cost $mn^2 + {}^1/_3\, n^3$ flops

Eigenvalue Decomposition

$A \in \mathbb{R}^{n \times n}$

Introduction

Let $\boldsymbol{x} \in \mathbb{R}^n$, $\boxed{\boldsymbol{A} \in \mathbb{R}^{n\times n}}$ *and* $\lambda \in \mathbb{C}$, if

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

$$\boxed{Ax' = \lambda x}$$

Then λ is the eigenvalue and x is the eigen vector.

- The action of $\boldsymbol{A}$ on special set of vectors of $\mathbb{R}^n$, does not rotate but scales the vector .
- The special subspace is called the eigenspace of $\boldsymbol{A}$, the set of all eigenvalues is called the spectrum of $\boldsymbol{A}$.

Least squares – Comparison of Matrix Decompositions

- **Cholesky Decomposition Approach**: Computationally cheaper of all to solve least Squares Problem if feature matrix **X** is full rank matrix. There can be instabilities i.e accumulation of round of errors when **X** has high condition number (Ratio of max singular value to min singular value).

- **SVD Approach**: Computationally expensive but a stable algorithm when **X** is close to rank deficient or rank deficient matrix.

- **QR Algorithm**: When number of features are small, same computational cost as SVD approach. This is cheaper when m is close to n but not as stable as SVD approach when **X** is close to rank deficient.

Eigenvalue Decomposition

Introduction

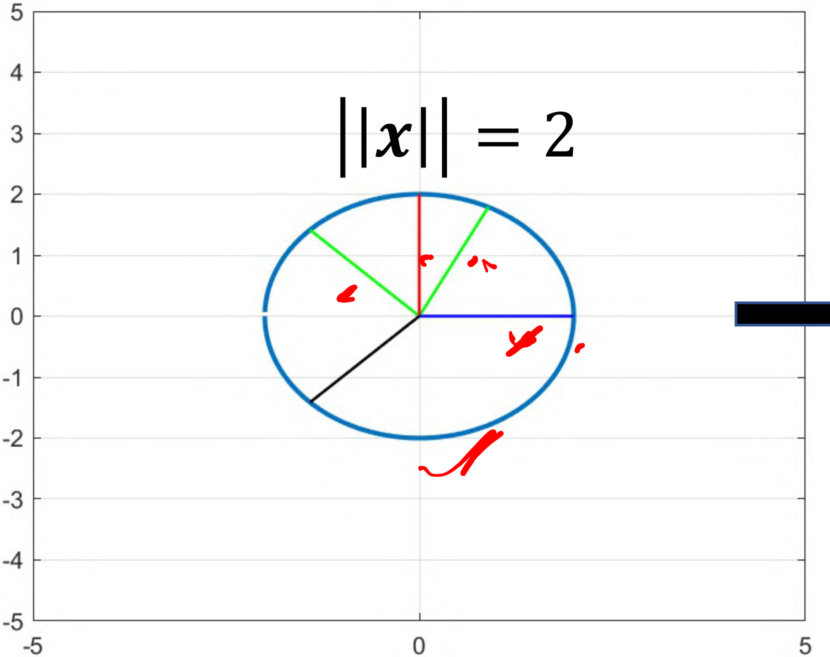
Let $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{A} \in \mathbb{R}^{n\times n} and\ \lambda \in \mathbb{C}$, if

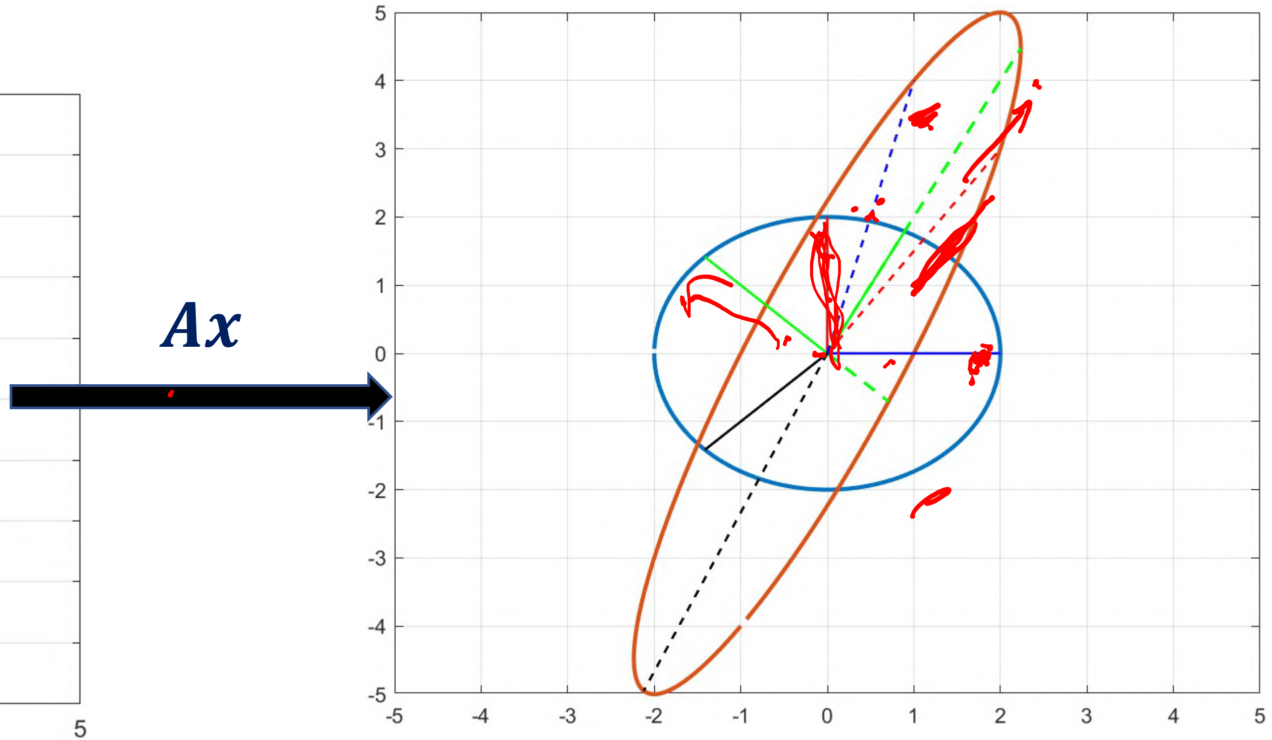
$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$$

Then λ is the eigenvalue and $\boldsymbol{x}$ is the eigen vector.

- The action of $\boldsymbol{A}$ on special set of vectors of $\mathbb{R}^n$, does not rotate but scales the vector .

<u>Geometric Intuition:</u>





Eigenvalue Decomposition

Applications

1. PageRank Algorithm: Important webpages are based on the number of webpages that are linked to it.
2. Dimensionality Reduction
3. Quantum Mechanics Schrodinger Equation:
4. Stability Analysis of Systems:
5. Checking positive definite of Random Matrix(Finance):

Eigenvalue Decomposition

Calculating Eigenvalue

- Characteristic Polynomial: The characteristic polynomial of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the n degree polynomial defined as

$$\boxed{p_{\boldsymbol{A}}(z) = \det(\boldsymbol{A} - z\boldsymbol{I})}$$

$A\underset{\sim}{x} = z\underset{\sim}{x}$

$\boxed{(A - zI)\underset{\sim}{x} = 0}$

$(A - zI)\underset{\sim}{x} = 0$

- The eigenvalues of $\boldsymbol{A}$ are the roots of the characteristic polynomial. $p_{\boldsymbol{A}}(\lambda) = 0$

- Example: $\boldsymbol{A} = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$, $\boxed{p_{\boldsymbol{A}}(z) = z^2 - 4z - 5}$

$\det\left[\begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix} - z\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right] = 0$

$$p_{\boldsymbol{A}}(\lambda) = 0$$
$$\lambda = 5, -1$$

$$\boxed{\boldsymbol{A}\boldsymbol{x}_1 = 5\boldsymbol{x}_1}$$
$$\begin{pmatrix} x_1 + 2y_1 \\ 4x_1 + 3y_1 \end{pmatrix} = \begin{pmatrix} 5x_1 \\ 5y_1 \end{pmatrix}$$
$$\boxed{\boldsymbol{x}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}}$$

$$\boxed{\boldsymbol{A}\boldsymbol{x}_2 = -1\boldsymbol{x}_2}$$
$$\begin{pmatrix} x_2 + 2y_2 \\ 4x_2 + 3y_2 \end{pmatrix} = \begin{pmatrix} -x_2 \\ -y_2 \end{pmatrix}$$
$$\boxed{\boldsymbol{x}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}}$$

Eigenvalue Decomposition

$A = A^T$

$(x^T x)^T = x^T x$

$\boxed{A = -A^T}$

Properties

Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, then

- If $\boldsymbol{A}$ is symmetric then the eigenvalues are real and the eigenvectors are orthogonal. (Eg: Graph Laplacian)
- If $\boldsymbol{A}$ is skew-symmetric then the eigenvalues are complex
- If $\boldsymbol{A}$ is an orthogonal matrix then magnitude of eigenvalues of $\boldsymbol{A}$ is 1

$A^T A = I$ $\quad$ $\boxed{|\lambda| = 1}$

Eigenvalue Decomposition

Special Properties

- Trace of $\boldsymbol{A}$: is the sum of eigenvalues of matrix $\boldsymbol{A}$

$$tr(\boldsymbol{A}) = \sum \lambda_i$$

- Determinant of $\boldsymbol{A}$: is the product of eigenvalues of matrix $\boldsymbol{A}$

$$\det(\boldsymbol{A}) = \prod \lambda_i$$

- Singular matrices have at least one eigenvalue 0. The determinant is also 0
- Normal Matrices: Matrices that commute with its transpose

$$\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{A}^T\mathbf{A}$$

  - **Normal Matrices have orthogonal eigenvectors**

$Tr(A)$

$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ $Tr(A)$ $a+e+i$

$AB \neq BA$

$\boxed{AA^T = A^T A}$

$QQ^T = Q^TQ = I$

$A^2 = A^2$ $A = A^T$ $QQ^T = I$

Eigenvalue Decomposition

Special Properties

Let $\boldsymbol{x} \in \mathbb{R}^n$ *be the eigenvector*, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *and* $\lambda \in \mathbb{C}$, be the eigenvalue,then

$$\boldsymbol{Ax} = \lambda\boldsymbol{x}$$
$$\boldsymbol{A^k x} = \boldsymbol{\lambda^k x}$$
$$\boldsymbol{AAx} = \boldsymbol{A}\lambda\boldsymbol{x}$$
$$\boldsymbol{A}^2\boldsymbol{x} = \lambda^2\boldsymbol{x}$$

$$Ax = \lambda x$$
$$A(Ax) = \lambda(Ax)$$
$$= \lambda^2 x$$
$$A^2 x = \lambda^2 x$$
$$A^k x = \lambda^k x$$

Eigenvalue Decomposition

Similar Matrices

- Two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are similar if there exists a non-singular matrix $\boldsymbol{T}$ such that: $\boldsymbol{B} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}$
- If $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$, $\boldsymbol{T}^{-1}\boldsymbol{x}$ is an eigenvector of $\boldsymbol{B}$
- Eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ are same
- Proof:

$$\begin{aligned} p_{\boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}}(z) &= \det(z\boldsymbol{I} - \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}) \\ &= \det(\boldsymbol{T}^{-1}(z\boldsymbol{I} - \boldsymbol{A})\boldsymbol{T}) \\ &= \det(\boldsymbol{T}^{-1})\det((z\boldsymbol{I} - \boldsymbol{A}))\det(\boldsymbol{T}) \\ &= p_{\boldsymbol{A}}(z) \end{aligned}$$

- Used for computing the eigenvalues of matrices

$B = T^{-1} A T$

$A \rightarrow \lambda$

$B \rightarrow \lambda$

Eigenvalue Decomposition

$\boxed{A^T A = A A^T}$

Diagonalization/ Decomposition for Normal Matrix

- Eigenvalue Decomposition: Normal matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ can be decomposed to the form:

$A = X \Lambda X^{-1}$ $\quad X^{-1} = X^T$

$$\boxed{\boldsymbol{A} = \mathbf{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T}$$
$$\boldsymbol{AQ} = \boldsymbol{Q\Lambda}$$

$A = Q \Lambda Q^T$

- Where $\boldsymbol{Q}$ is the eigenspace of $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ is the diagonal matrix with the eigenvalues along the diagonal.

$A^T A = A A^T$ $\quad A = Q \Lambda Q^T$

$$\begin{bmatrix} & A & \end{bmatrix}[\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_n] = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_n]\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

- Example: $\mathbf{A} = \begin{pmatrix} 10 & 2 \\ 1 & 11 \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} 1 & -2 \\ 1 & 1 \end{pmatrix}, \boldsymbol{\Lambda} = \begin{pmatrix} 9 & 0 \\ 0 & 12 \end{pmatrix}$

Eigenvalue Decomposition

$\boxed{A = X\Lambda X^{-1}}$ $\Rightarrow Q \Lambda Q^T$

$Ax = \lambda x$

$O(mn^2)$

EVD vs SVD

Eigenvalue Decomposition	Singular Value Decomposition
Only certain square matrices can be decomposed as: $\boxed{A = \mathbf{X\Lambda X^{-1}}}$ $X \Lambda X^T$	Matrices can be decomposed as: $\boxed{A = U\Sigma V^T}$
(Applicable only for square matrices)	(Applicable for all square and rectangular matrices) $x^T x \Rightarrow x^T y$
The elements of the diagonal matrix are can be real or complex	~~The elements of the diagonal matrix~~ are positive and real
The non diagonal matrices($\mathbf{X}$) are the inverse of each other	The non diagonal matrices need not be the inverse of each other

- The non zero singular values of A are the positive square roots of eigenvalues of $\boldsymbol{A^T A}$, $\boldsymbol{AA^T}$
- If $\boldsymbol{A = A^T}$ the singular values of A are the absolute value of eigenvalues of $\boldsymbol{A}$

Eigenvalue Decomposition

Page Rank as EVP

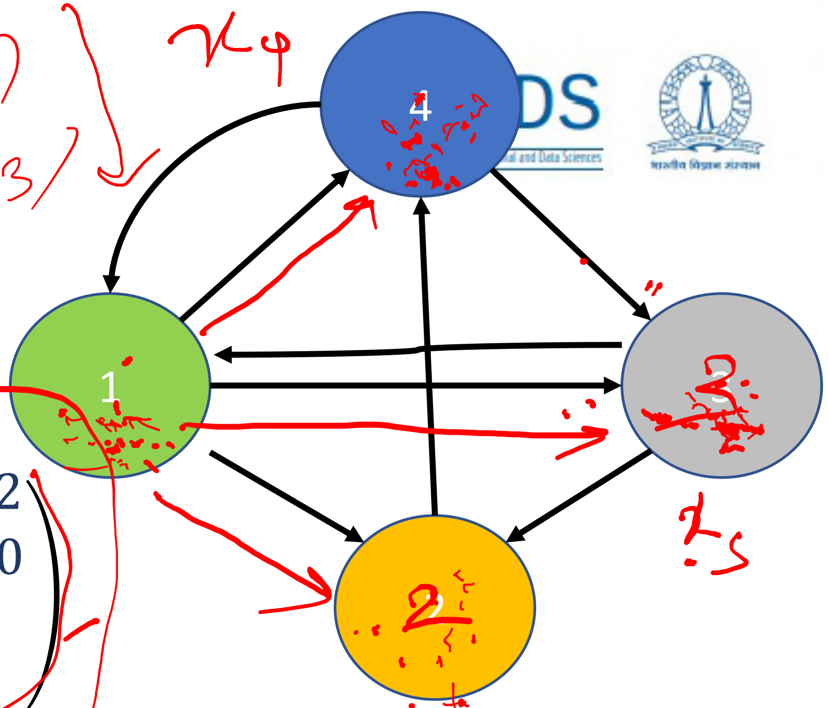
SVD $O(N^3)$ $O(mn^2)$ $O(n^3)$

- **Setting up as eigenvalue problem:**

$$x_1 = 0.5x_3 + 0.5x_4$$
$$x_2 = 0.33x_1 + 0.5x_3$$
$$x_3 = 0.33x_1 + 0.5x_4$$
$$x_4 = 0.33x_1 + x_2$$
$$\boxed{\boldsymbol{Mx} = \boldsymbol{x}}$$

$A = M$

$$\boldsymbol{M} = \begin{pmatrix} 0 & 0 & 1/2 & 1/2 \\ 1/3 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1 & 0 & 0 \end{pmatrix}$$



$\boxed{Mx = \lambda x}$

- **Power Iteration:**
  - $\frac{\boldsymbol{x}}{||\boldsymbol{x}||}, \frac{\boldsymbol{Ax}}{||\boldsymbol{Ax}||}, \frac{\boldsymbol{A^2x}}{||\boldsymbol{A^2x}||}, \frac{\boldsymbol{A^3x}}{||\boldsymbol{A^3x}||}, \ldots$ converges to the eigenvector with largest eigenvalue

$$\boldsymbol{A} = \mathbf{X\Lambda X^{-1}}$$
$$\boldsymbol{A^k} = \mathbf{X\Lambda^k X^{-1}}$$

$Ax =$

$\frac{AAAx}{||x||}$

- **Page Rank via power iteration:**

Let $r^{(0)} = [\frac{1}{N}, \frac{1}{N} \ldots \frac{1}{N}]'$ *here N=4*

The series is …

$$\begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}, \begin{pmatrix} 0.489 \\ 0.4082 \\ 0.4082 \\ 0.6532 \end{pmatrix}, \begin{pmatrix} 0.5352 \\ 0.3705 \\ 0.4940 \\ 0.5764 \end{pmatrix}, \begin{pmatrix} 0.5389 \\ 0.4283 \\ 0.4698 \\ 0.5527 \end{pmatrix}, \ldots, \begin{pmatrix} 0.5249 \\ 0.4082 \\ 0.4666 \\ 0.5832 \end{pmatrix}$$

x_4

Eigenvalue Decomposition

$A\,(x_1 \cdot x_2 \cdots x_n)$

Simultaneous Iteration

- <u>Recall power iteration:</u> Where we saw that a series of $\frac{x}{||x||_2}, \frac{Ax}{||Ax||_2}, \frac{A^2x}{||A^2x||_2}, \frac{A^3x}{||A^3x||_2}, \ldots$ converges to the eigenvector with largest eigenvalue.
- <u>In simultaneous iteration:</u> we apply power iteration to several vectors at once .

  Define:

$$V^{(0)} = \left[v_1^{(0)} \quad \ldots \quad v_n^{(0)}\right],$$

$$after\ applying\ succeivily\ on\ A\ we\ get$$

$$V^{(k)} = \left[v_1^{(k)} \quad \ldots \quad v_n^{(k)}\right] = AV^{(0)}$$

  Instead of using $V^{(0)}$ and $V^{(k)}$, we can use $Q^{(0)}$ and $Q^{(k)}$ by QR factorization at each step. Since columns of $Q^{(k)}$ are linearly independent, this iteration behaves better(improved condition number)

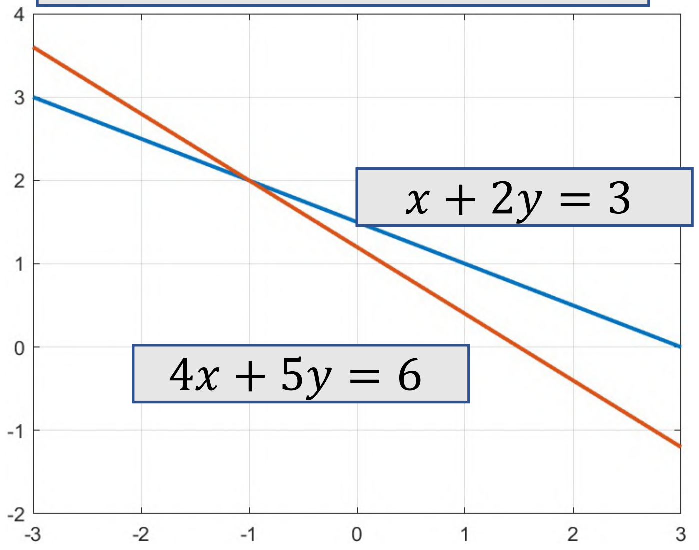
- <u>Algorithm:</u>
  1. $V^{(0)} = \widehat{Q^{(0)}}\widehat{R}$
  2. Iterate until convergence…
     1. $Z = A\widehat{Q^{(i-1)}}$
     2. $Q^{(i-1)}\widehat{R^{(i-1)}}$=Z

LU Decomposition for System of Equations

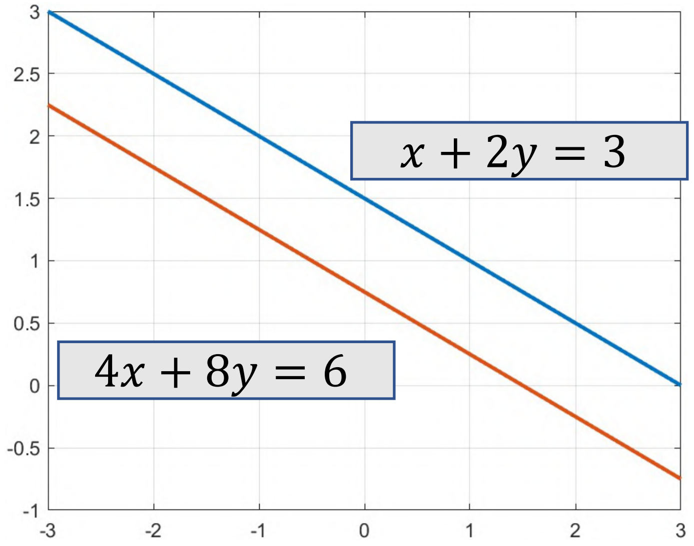
Introduction

- Linear Equations lead to geometry of planes.
- The solution of these system of equations is the point of intersection of all the planes
- Using this method for higher dimensions is difficult to visualise(e.g. 9D plane) , hence we use a **matrix** based approach

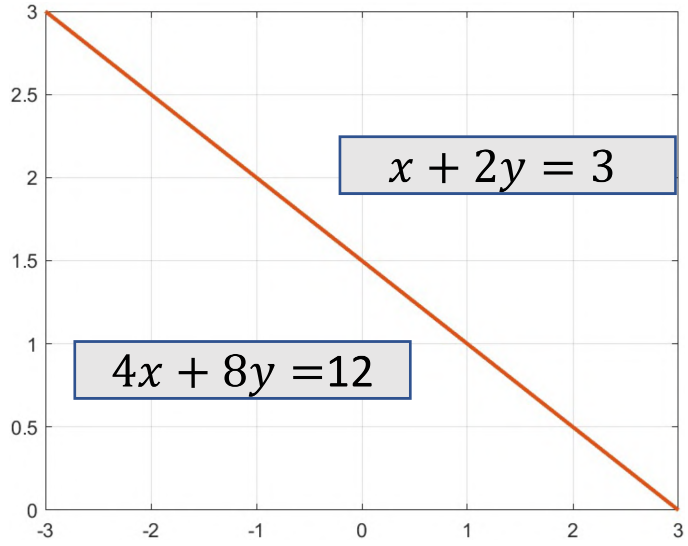
One Solution



Parallel: No solution



Whole line of solution



LU Decomposition -- Gaussian Elimination

Introduction

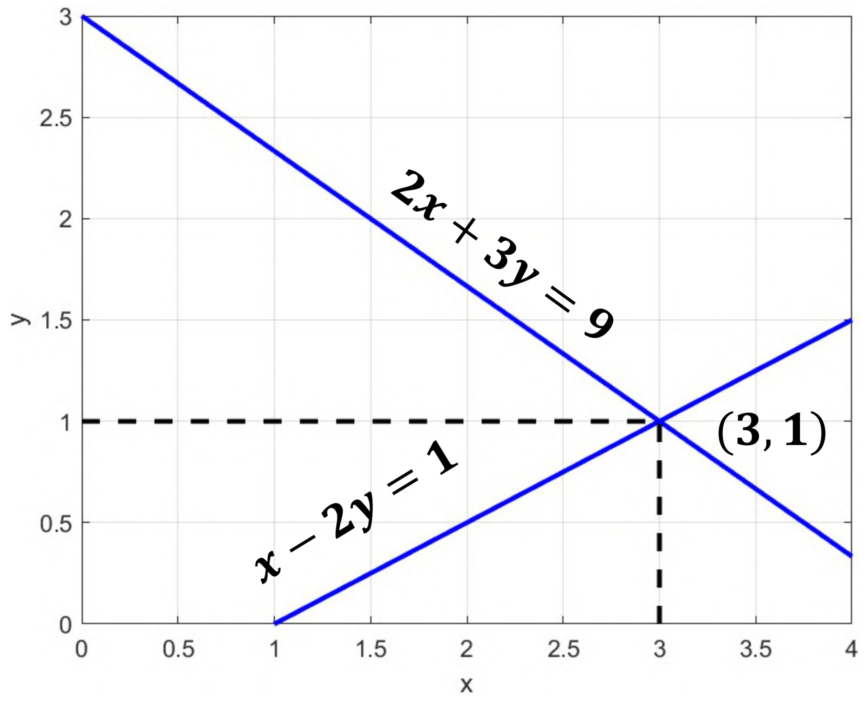
$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$$
$$\boldsymbol{A}, \boldsymbol{L}, \boldsymbol{U} \in \mathbb{R}^{mxm}$$

Gaussian Elimination transforms a full linear system into an upper triangular matrix $\boldsymbol{U}$ by applying simple linear transformations to the left $\boldsymbol{L}$

$$\boxed{\boldsymbol{L}_{m-1}\boldsymbol{L}_{m-2} \ldots \boldsymbol{L}_2\, \boldsymbol{L}_1}\boldsymbol{A} = \boldsymbol{U}$$

$$\boxed{\boldsymbol{L}^{-1}}$$

$$\begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \end{bmatrix}$$



LU Decomposition -- Gaussian Elimination

Procedure

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{L_1} \begin{bmatrix} x & x & x & x \\ 0 & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ 0 & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ 0 & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \end{bmatrix} \xrightarrow{L_2} \begin{bmatrix} x & x & x & x \\ 0 & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ 0 & 0 & \boldsymbol{x} & \boldsymbol{x} \\ 0 & 0 & \boldsymbol{x} & \boldsymbol{x} \end{bmatrix} \xrightarrow{L_3} \begin{bmatrix} x & x & x & x \\ 0 & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ 0 & 0 & \boldsymbol{x} & \boldsymbol{x} \\ 0 & 0 & \boldsymbol{0} & \boldsymbol{x} \end{bmatrix}$$

$$\boldsymbol{x}_k = \begin{pmatrix} x_{1k} \\ \cdots \\ x_{k,k} \\ x_{k+1,k} \\ \cdots \\ x_{mk} \end{pmatrix} \rightarrow \boldsymbol{L}_k \boldsymbol{x}_k = \begin{pmatrix} x_{1k} \\ \cdots \\ x_{k,k} \\ 0 \\ \cdots \\ 0 \end{pmatrix}, l_{jk} = \frac{x_{jk}}{x_{kk}}; j > k$$

LU Decomposition -- Gaussian Elimination

Example

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \boldsymbol{b} = \begin{pmatrix} 5 \\ -2 \\ 9 \end{pmatrix}$$

Operation on $\boldsymbol{A}$…
$row2$ -2*row1
row3 + row1

$$\boldsymbol{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ \boxed{-2} & 1 & 0 \\ \boxed{1} & 0 & 1 \end{bmatrix}; \boldsymbol{L}_1\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 8 & 3 \end{bmatrix}$$

Operation on $\boldsymbol{L_1A}$…
row3 + row2

$$\boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \boxed{1} & 1 \end{bmatrix}; \boldsymbol{L}_2\boldsymbol{A} = \boxed{\boldsymbol{U} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix}}$$

$$\boxed{\boldsymbol{L} = \boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \boxed{2} & 1 & 0 \\ \boxed{-1} & \boxed{-1} & 1 \end{bmatrix}}$$

Notice the relation of $\boldsymbol{L}, \boldsymbol{L_1}, \boldsymbol{L_2}$

LU Decomposition -- Gaussian Elimination

Example

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \boldsymbol{b} = \begin{pmatrix} 5 \\ -2 \\ 9 \end{pmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boldsymbol{Ly} = \boldsymbol{b}$ (Forward Substitution)

$$\boldsymbol{y} = \begin{pmatrix} 5 \\ -12 \\ 2 \end{pmatrix}$$

$\boldsymbol{Ux} = \boldsymbol{y}$ (Backward Substitution)

$$\boldsymbol{x} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

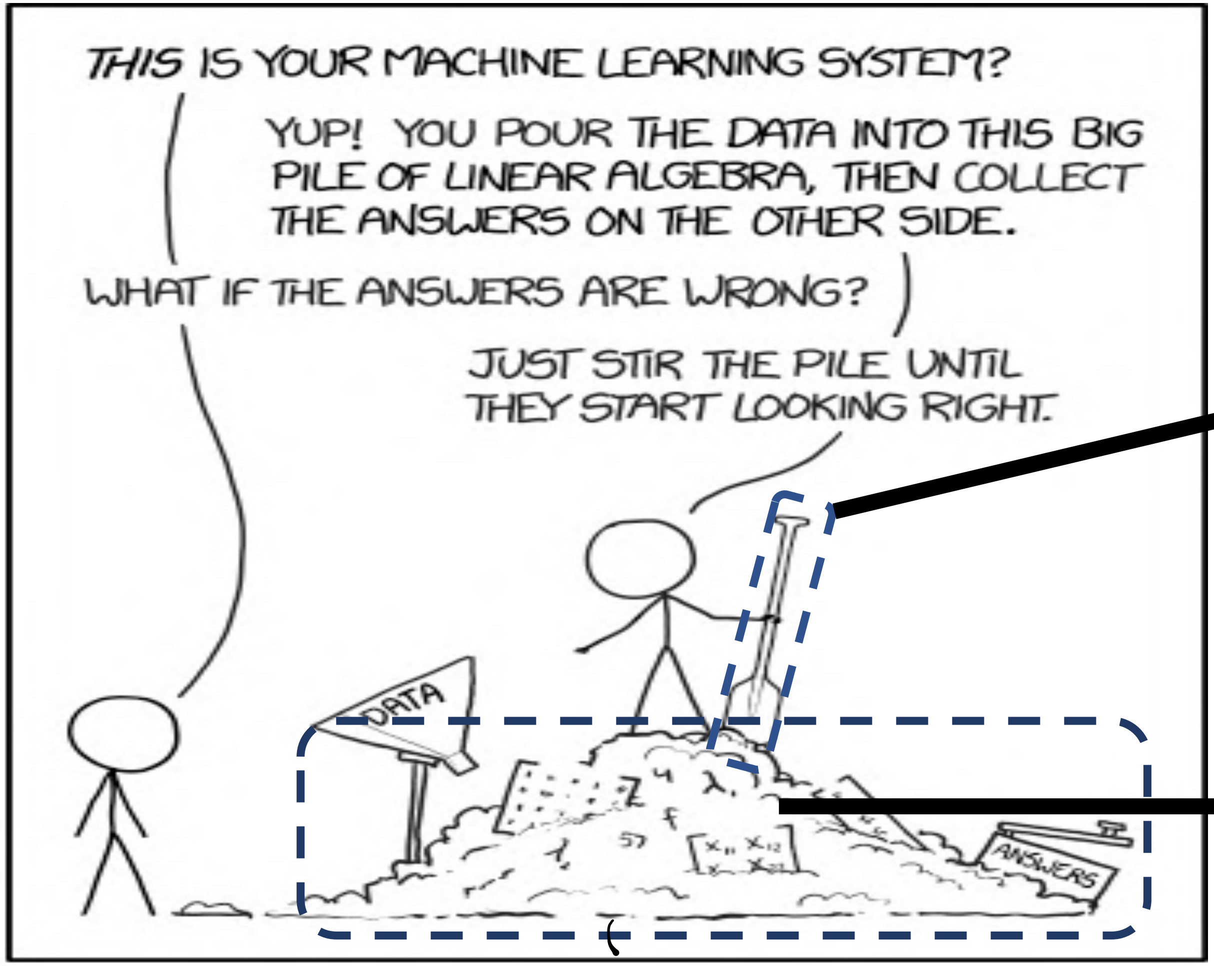
Gaussian Elimination

Procedure

- <u>Algorithm:</u>
  - $\boldsymbol{U} = \boldsymbol{A}, \boldsymbol{L} = \boldsymbol{I}$
  - for k=1:m-1
    - for j= k+1:m
      - $l_{jk} = {}^{\boldsymbol{u_{jk}}}/_{\boldsymbol{u_{kk}}}$
      - $\boldsymbol{u_{j,k:m}} = \boldsymbol{u_{j,k:m}} - l_{jk}\boldsymbol{u}_{k,k:m}$

- <u>Operation Count:</u> $\sim \frac{2}{3}m^3$
- Pure Gaussian Elimination can lead to 0 in the diagonal element. Row exchanges can be done by using a left transformation permutation matrix $\boldsymbol{P}$ at every step.

Concluding Remarks…



https://imgs.xkcd.com/comics/machine_learning.png

Matrix Decompositions
Tools to stir the data pile

Vectors, Matrices, Tensors and their operations
Store and Manipulate data