



COMP 417 – Tutorial 3

October 4, 2019

Linear Algebra Tutorial



Outline

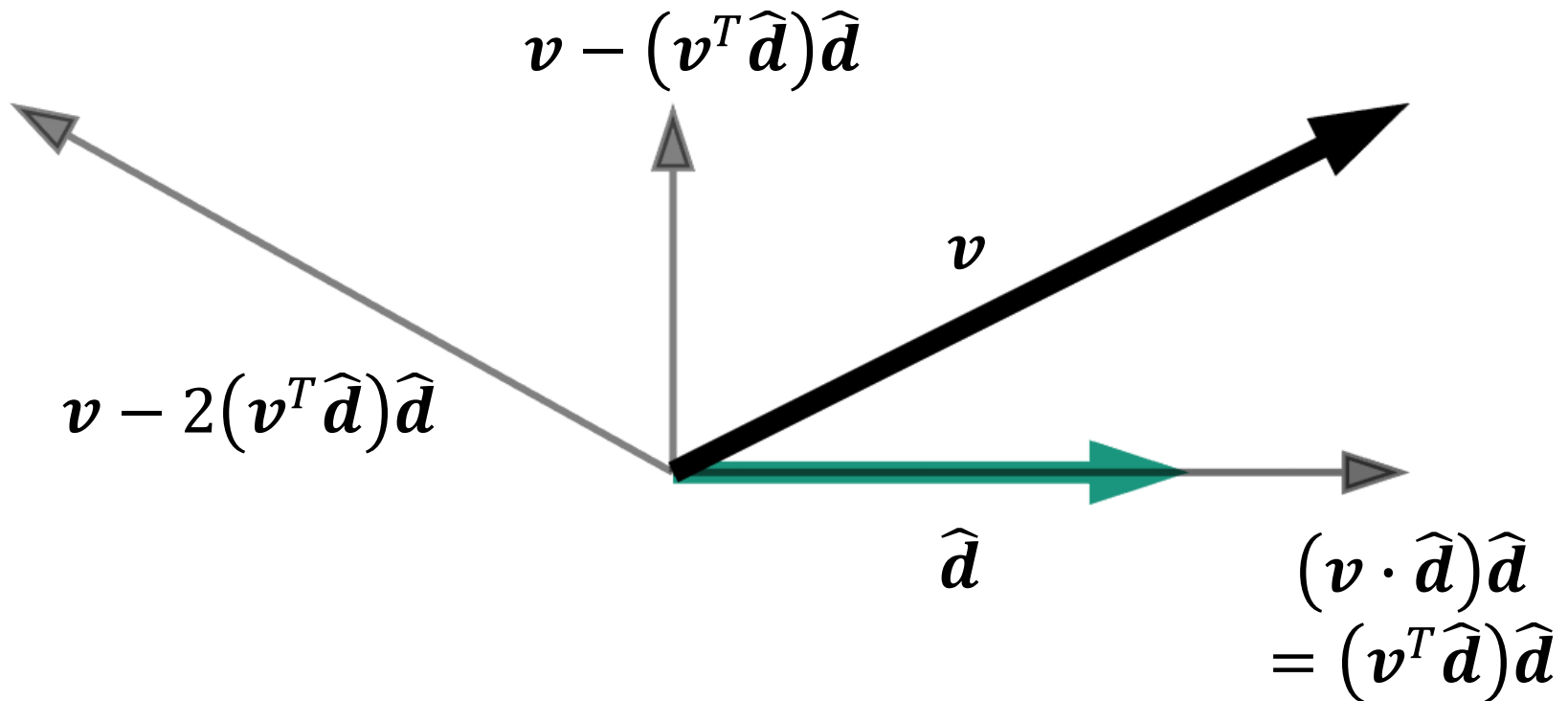
- Basic matrix and vector operations review
- Coordinate transforms
- Optimization: Least Squares, Total Least Squares, relation to SVD and eigendecomposition
- Optional ROS coordinate transform exercise



Basic Matrix and Vector Operations

Vector Directional Components

- Given \mathbf{v} and directional unit vector $\hat{\mathbf{d}}$, can decompose \mathbf{v} :



Orthogonal Vectors and Matrices

Orthogonal Vectors:

- Orthogonal vectors have no parallel component.

$$\mathbf{v}^T \mathbf{u} = 0$$

Orthogonal Matrices:

- Square matrix whose columns and rows are orthogonal unit vectors.
- Off-diagonal entries correspond to multiplication of different vector pairs (0), while main diagonal is self-multiplication (1).

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}$$

Matrix Inverse

Definition:

$$AB = BA = I \leftrightarrow A = B^{-1} \text{ and } B = A^{-1}$$

- For non-square matrices only left or right inverse may exist

Many equivalent conditions to be invertible (Non-Singular). Some of most notable are:

- $\det(A) \neq 0$
- Full rank: $\text{rank}(A) = n$ (rows and columns linearly ind.)

For Orthogonal Matrices:

- By definition, is the transpose

$$U^T U = U U^T = I \rightarrow U^{-1} = U^T$$

Vector Norms

Norm Function: Assigns a positive length to a vector

- Length value depends on norm variant

P-norm general equation:

$$\|v\|_p = \left(\sum_{i=1}^n |v_i|^p \right)^{1/p}$$

L1 Norm:	Sum of absolute values
L2 Norm:	Euclidian Distance
Infinity Norm:	Magnitude of largest element

L2 Norm Squared: Also frequently used. Several equivalent representations:

$$\sum_i^n (u_i)^2 = \|\mathbf{u}\|_2^2 = \mathbf{u}^T \mathbf{u}$$

Gradient and Jacobian

Gradient: Partial derivative of a multivariable function w.r.t. a vector

$$\nabla_x f(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_n} f(\mathbf{x}) \end{bmatrix}$$

Jacobian: Partial derivative of a **vector** of functions w.r.t. a vector

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$



Coordinate Transforms

Homogenous Coordinates

- Point augmented with an additional coordinate to represent the point at different scales

Normalized (scaling = 1) Homogenous Coordinates

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Non-Normalized (scaling $\neq 1$) Homogenous Coordinates

$$p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow \tilde{p} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \quad p = \frac{\tilde{p}}{w}$$

Homogenous Coordinates

Applications – Point at Infinity

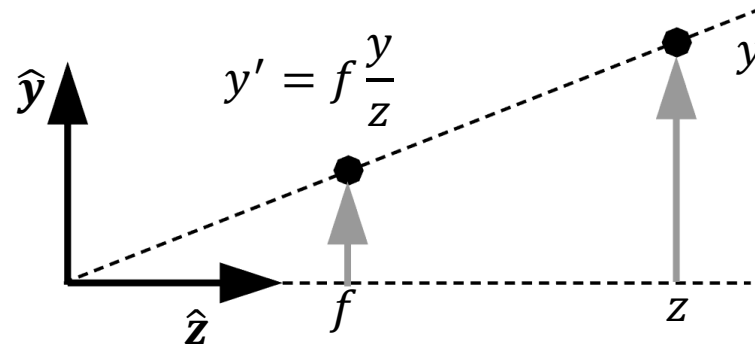
- Set $w = 0$
- Normative division causes resulting point to possess values at infinity.

$$\tilde{p} = \begin{bmatrix} x \\ y \\ z \\ w = 0 \end{bmatrix}$$
$$p = \frac{\tilde{p}}{w}$$

Homogenous Coordinates Applications

– Perspective Projection

- Conversion between non-normalized and normalized homogenous coordinates viewed as perspective projection of point at some depth to point on projection plane at length $f = 1$



- For arbitrary focal lengths, introduce scaling matrix:

$$\begin{bmatrix} f \frac{x}{z} \\ f \frac{y}{z} \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Homogenous Coordinates

Applications – Translation Operation

- Can represent translation as a matrix multiplication with homogenous coordinates.

$$\mathbf{p}' = \mathbf{T}\mathbf{p}$$

$$\begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Matrix Transformations (1/2)

Matrix Transformations

- Apply matrix transformation F to transform a point:

$$p' = Fp \rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{F} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Translation:

- See previous slide

Scaling:

$$\begin{bmatrix} s_x x \\ s_y y \\ s_z z \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Matrix Transformations (2/2)

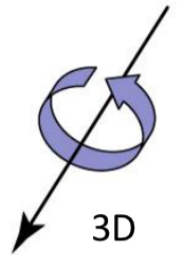
3d rotation about z / 2d rotation

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{2d} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3d rotation about x

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



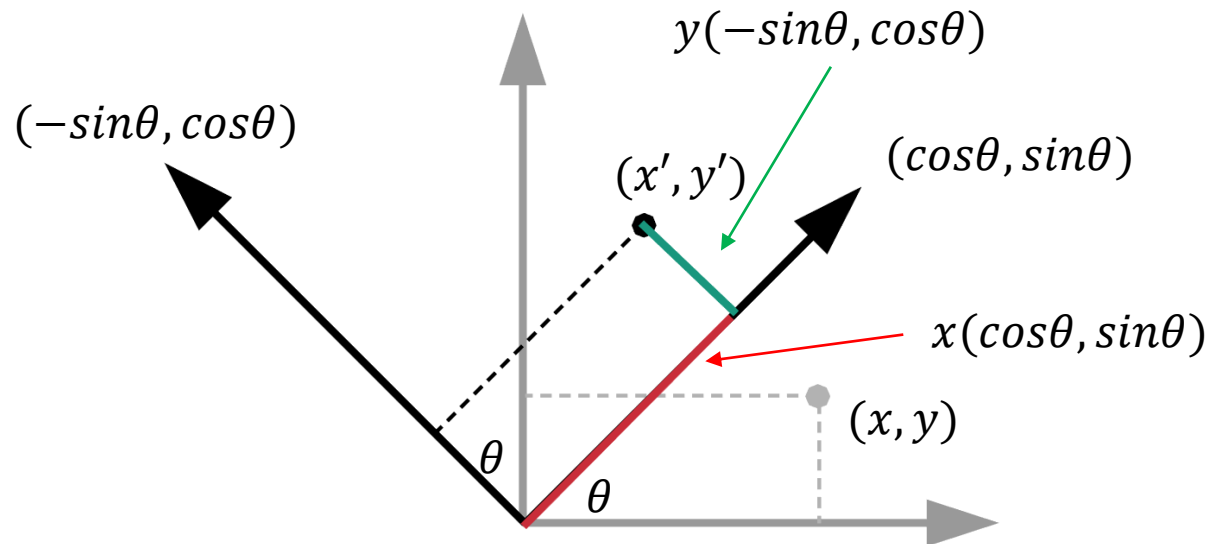
3d rotation about y

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation Matrix Equation Intuition

- Derived by doing vector addition along coordinate frame made by rotation

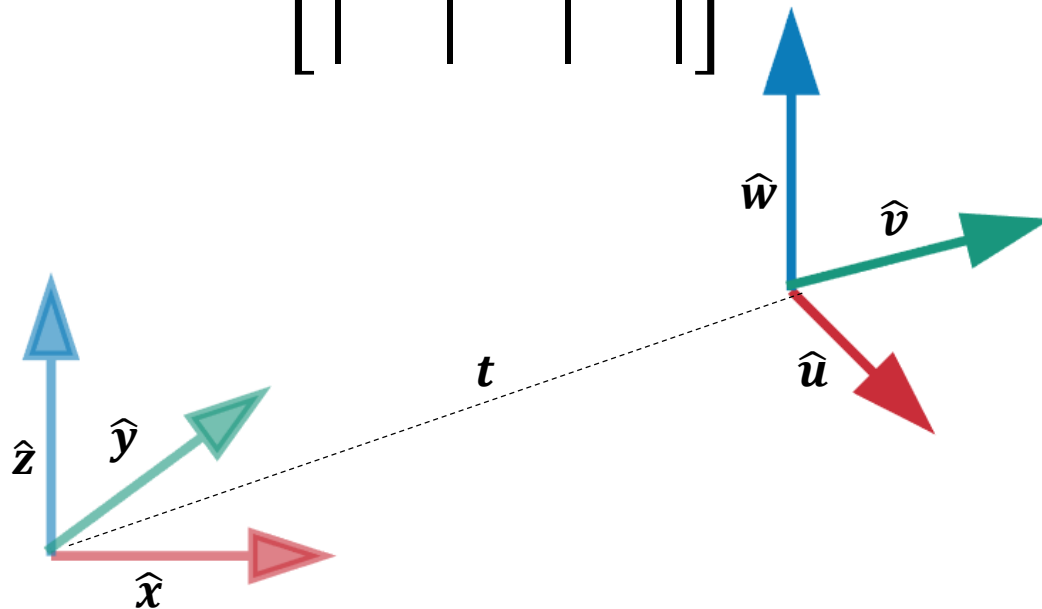
$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \textcolor{red}{\cos\theta} \\ \textcolor{red}{\sin\theta} \\ \textcolor{red}{0} \end{bmatrix} x + \begin{bmatrix} \textcolor{green}{-\sin\theta} \\ \textcolor{green}{\cos\theta} \\ \textcolor{green}{0} \end{bmatrix} y + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Coordinate Frames

- Described typically by set of orthogonal unit vectors \hat{u} , \hat{v} , \hat{w} and origin translation t .
- Grouped together in matrix F as column vectors

$$F = \begin{bmatrix} | & | & | & | \\ \hat{u} & \hat{v} & \hat{w} & t \\ | & | & | & | \end{bmatrix}$$



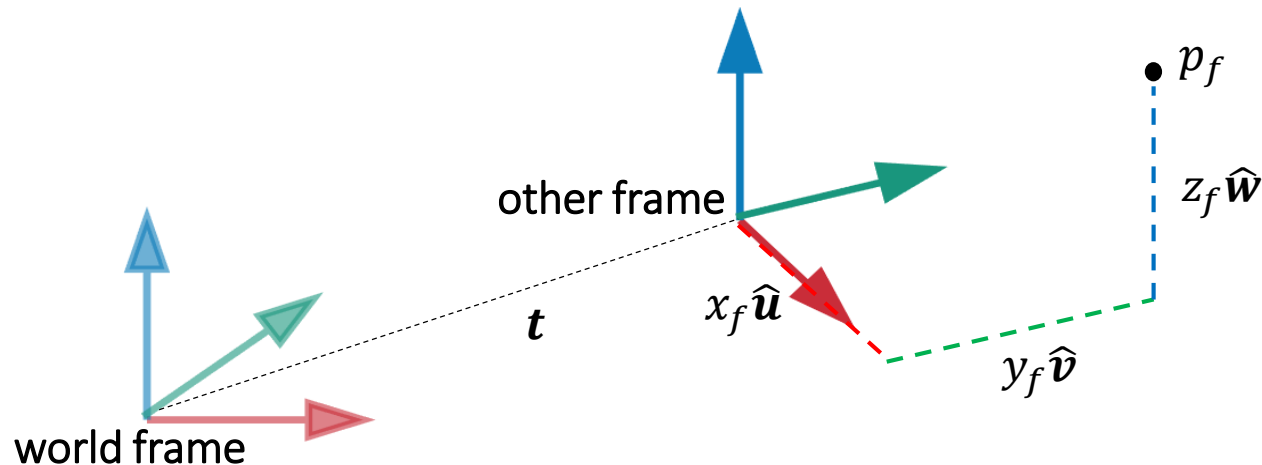
Coordinate Frame Transformations (1/2)

Goal: Represent point in different coordinate frames.

- Multiply by F or its inverse to convert between frames.

Example: Write p_f in world coordinates:

$$p_w = F p_f \rightarrow \begin{bmatrix} | \\ \hat{\mathbf{u}} \\ | \end{bmatrix} x_f + \begin{bmatrix} | \\ \hat{\mathbf{v}} \\ | \end{bmatrix} y_f + \begin{bmatrix} | \\ \hat{\mathbf{w}} \\ | \end{bmatrix} z_f + \begin{bmatrix} | \\ \mathbf{t} \\ | \end{bmatrix}$$



Coordinate Frame Transformations (2/2)

Summary:

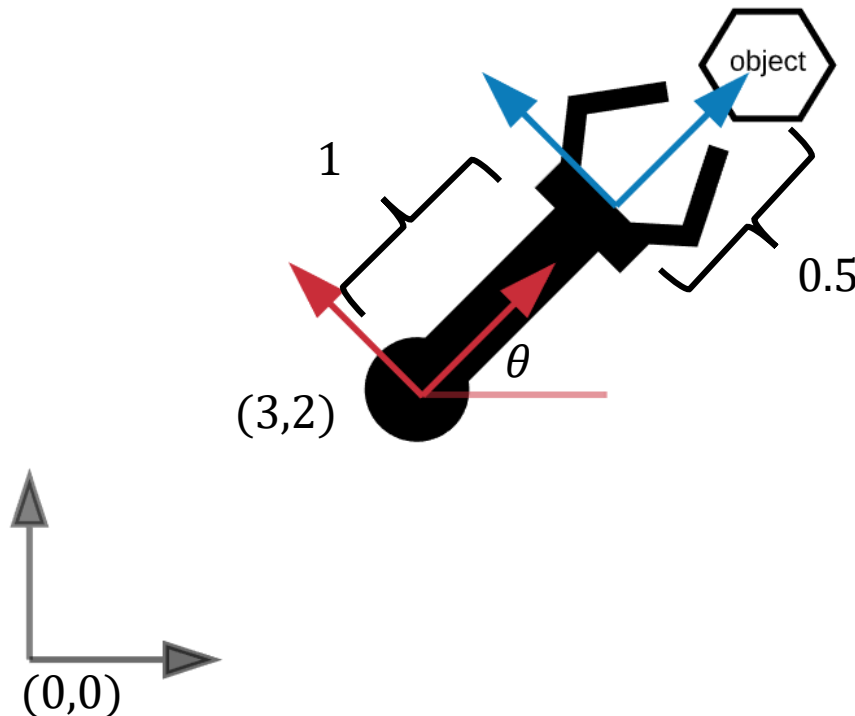
Coordinate Frame to World:	$p_w = F p_f$
World to Coordinate Frame:	$p_f = F^{-1} p_w$

Relation to Matrix Transformations:

- Example: Rotation matrix can be viewed as rotated coordinate frame with 0 origin translation.
 - Any orthogonal coordinate frame is considered a rotation.
- Example: Translation can be viewed as coordinate frame with no rotation but offset origin.

Coordinate Transform Example (1/2)

A robot arm with a single rotating joint is placed at (3,2) in a 2d plane. The joint is rotated to $\pi/4$. An object is detected (0.5, 0) in front of the rotated arm. The arm has a length of 1. What is the position of the object in world coordinates?



Coordinate Transform Example (2/2)

Robot Arm End-Effector Coordinate frame:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} & 0 \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} & 3 + \cos \frac{\pi}{4} \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} & 2 + \sin \frac{\pi}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

Arm Frame to World Coordinates:

- Note that point represented in homogenous coordinates

$$\mathbf{p}_w = \mathbf{F} \begin{bmatrix} 0.5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} & 3 + \cos \frac{\pi}{4} \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} & 2 + \sin \frac{\pi}{4} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4.06 \\ 3.06 \\ 1 \end{bmatrix}$$

Matrix Transformations Applied to Multiple Points

- Multiple points can be stacked column-wise:

$$\begin{bmatrix} | & | & & | \\ p_1' & p_2' & \dots & p_m' \\ | & | & & | \end{bmatrix} = \mathbf{F} \begin{bmatrix} | & | & & | \\ p_1 & p_2 & \dots & p_m \\ | & | & & | \end{bmatrix}$$



Optimization and Decomposition Methods

Least Squares Optimization

Motivation:

- Given dataset of M tuples $(\mathbf{x}^{(i)}, y^{(i)})$, fit function approximation $\hat{y} = f_{\mathbf{w}}(\mathbf{x})$.
- Function approximation parameterized by weights \mathbf{w} .

Least Squares Optimization:

$$Loss(L) = \sum_i^M (y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)}))^2$$

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} L$$

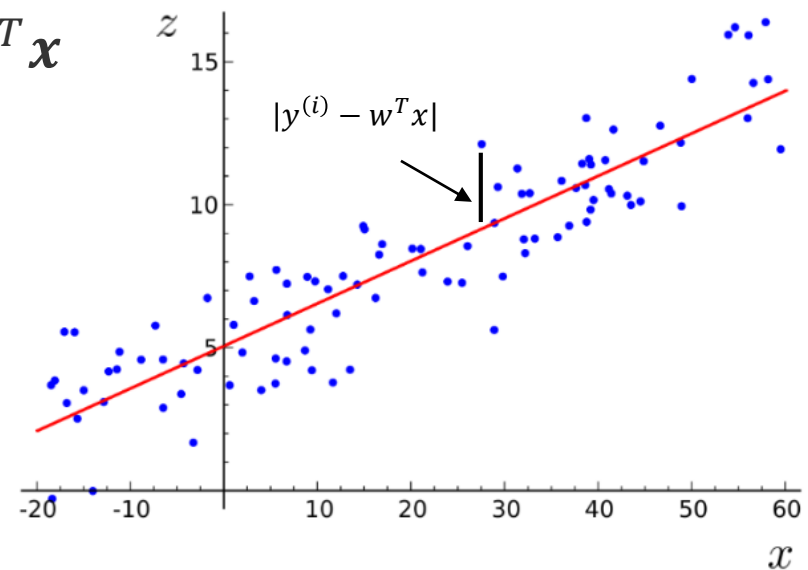
Linear Regression

Motivation:

- A specific instance of Least Squares Optimization
- Function approximator written as linear combination of weights:

$$f_w(x) = (1)w_0 + x_1w_1 + \cdots x_nw_n = \mathbf{w}^T \mathbf{x}$$

$$\operatorname{argmin}_w \sum_i^M (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2$$



Linear Regression Matrix Format

- Can represent linear regression in purely matrix form since $\sum_i^M (u_i)^2 = ||\mathbf{u}||^2 = \mathbf{u}^T \mathbf{u}$

$$\operatorname{argmin}_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\begin{bmatrix} \mathbf{X} \\ (M \times N) \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ (N \times 1) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}} \\ (M \times 1) \end{bmatrix}$$

Linear Regression Solution

Closed-Form Solution:

- Solve minimization by taking gradient and setting to 0

$$\begin{aligned}\nabla_{\mathbf{w}}[(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})] &= 0 \\ \rightarrow \nabla_{\mathbf{w}}[\mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}] &= 0 \\ \rightarrow -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w} &= 0 \\ \rightarrow \mathbf{w} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\end{aligned}$$

Problem: Matrix inverse expensive or may be unstable

Incremental Numerical Solution: Gradient Descent

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \alpha \nabla_{\mathbf{w}} L$$

Eigenvalues and Eigenvectors

- Square matrix A has eigenvector v and eigenvalue λ under condition:

$$\boxed{Av = \lambda v} \text{ where } v \neq \vec{0}$$

Interpretation:

- Describes case where transformation A on v equivalent to applying scaling factor λ

Eigendecomposition

- Previous equation described single eigenvalue/vector pair
- Square, diagonalizable A has N eigenvectors:

$$AV = V\Lambda \rightarrow \boxed{A = V\Lambda V^{-1}}$$

Λ : Diagonal matrix of eigenvalues

V : Matrix of column vectors corresponding to eigenvectors

- For **symmetric** matrix, ensured orthogonal eigenvectors:

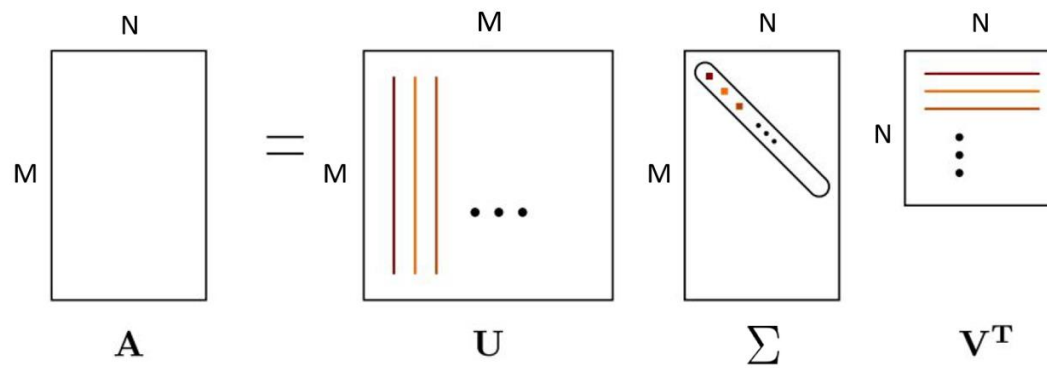
$$A = V\Lambda V^{-1} \rightarrow \boxed{A = V\Lambda V^T}$$

Interpretation as Coordinate Transform Operation:

- Transformation to new coordinate basis (V^{-1}) where application of A acts as simple scaling (Λ) followed by final inverse transform (V).

Singular Value Decomposition (SVD)

- Decomposes $M \times N$ matrix A into:

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


The diagram illustrates the SVD decomposition $A = U \Sigma V^T$. Matrix A (size $M \times N$) is shown as a rectangle. It is equal to the product of matrix U (size $M \times M$), matrix Σ (size $M \times N$), and matrix V^T (size $N \times N$). Matrix U is shown with vertical orange lines. Matrix Σ is shown with a diagonal of orange dots. Matrix V^T is shown with horizontal orange lines.

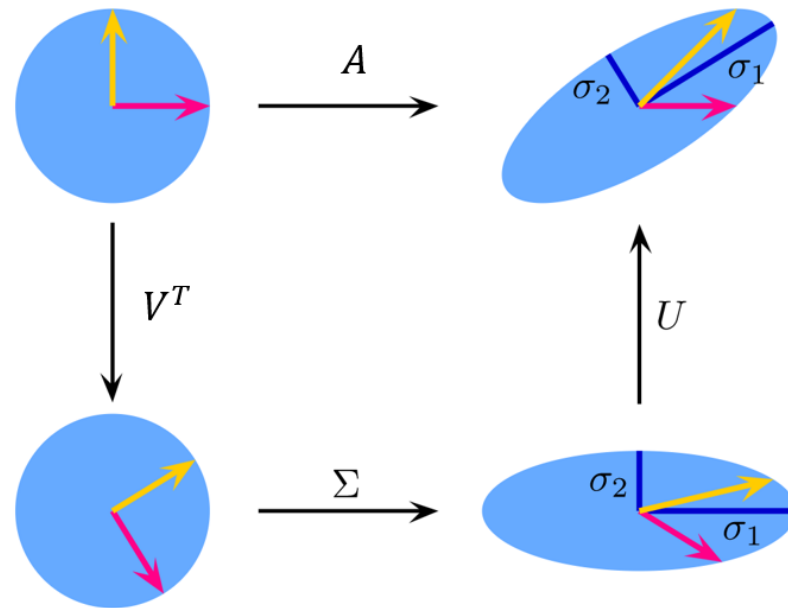
Σ **Singular values.** $M \times N$ diagonal matrix.
Square root of eigenvalues of $A^T A$ or AA^T ($\sqrt{\lambda_i}$)

V **Right-singular vectors.** $N \times N$ orthogonal matrix
Eigenvectors of $A^T A$. Proof: $A^T A = (V \Sigma^T U^T)(U \Sigma V^T) = V \Lambda V^T$

U **Left-singular vectors.** $M \times M$ orthogonal matrix
Eigenvectors of AA^T . Proof: $AA^T = (U \Sigma V^T)(V \Sigma^T U^T) = U \Lambda U^T$

SVD Interpretation

1. Rotation to new coordinate system defined by V
2. Scaling by Σ
3. Final opposing rotation defined by U



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

Application: Total Least Squares

Problem Statement:

$$\operatorname{argmin}_{\mathbf{w}} ||\mathbf{X}\mathbf{w}||^2 \text{ with constraint } ||\mathbf{w}|| = 1$$

Solution:

- Eigenvector \mathbf{w} of $\mathbf{X}^T \mathbf{X}$ corresponding to smallest eigenvalue. Solve by taking SVD of \mathbf{X} and taking eigenvector in \mathbf{V} .

Reasoning:

$$||\mathbf{X}\mathbf{w}_{min}||^2 = \mathbf{w}_{min}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{min} = \text{some min value} = \lambda$$

- Same result obtained by starting with eigenvector/value relation:

$$\mathbf{X}^T \mathbf{X} \mathbf{w}_{min} = \lambda \mathbf{w}_{min} \rightarrow \mathbf{w}_{min}^T \mathbf{X}^T \mathbf{X} \mathbf{w}_{min} = \lambda$$

- Therefore eigenvector \mathbf{w}_{min} satisfies minimization.

Application: Pseudo-Inverse

- If SVD decomposition is known, computation of pseudo-inverse is trivial

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

$((V\Sigma^+ U^T)(U\Sigma V^T) = I$ and Σ^+ is reciprocal transpose of Σ)

- Above pseudo-inverse can be used in place of left pseudo-inverse in Least Squares linear regression equation:

$$w = (X^T X)^{-1} X^T y \rightarrow V\Sigma^+ U^T y$$

(where $X = A$)

- Recall in case where inverse exists, pseudo-inverse is same

SVD and Eigendecomposition Equivalency

Requirements on matrix A :

- Symmetric ($A^T = A$)
- Positive semi-definite $\mathbf{z}^T A \mathbf{z} \geq 0$ for any \mathbf{z}

Total Least Squares Example: $A = X^T X$

- Symmetric: $(X^T X)^T = X^T X$
- Positive semi-definite: $\mathbf{z}^T X^T X \mathbf{z} = (X \mathbf{z})^T (X \mathbf{z}) = ||X \mathbf{z}||^2 \geq 0$

Eigendecomposition	SVD
$X^T X = A = V \Lambda V^T$	$\begin{aligned} X^T X &= V \Sigma^T U^T U \Sigma V^T \\ &= V \Sigma^T \Sigma V^T \\ &= V \Lambda V^T \end{aligned}$



Optional ROS Coordinate Frame Exercise

- There is a short exercise involving coordinate transforms and ROS found here:

https://github.com/comp417-fall2019-tutorials/linear_algebra_tutorial

- See the **README.md** file for instructions