McGill
School of Computer Science

COMP 417 – Tutorial 3
October 4, 2019

Linear Algebra Tutorial

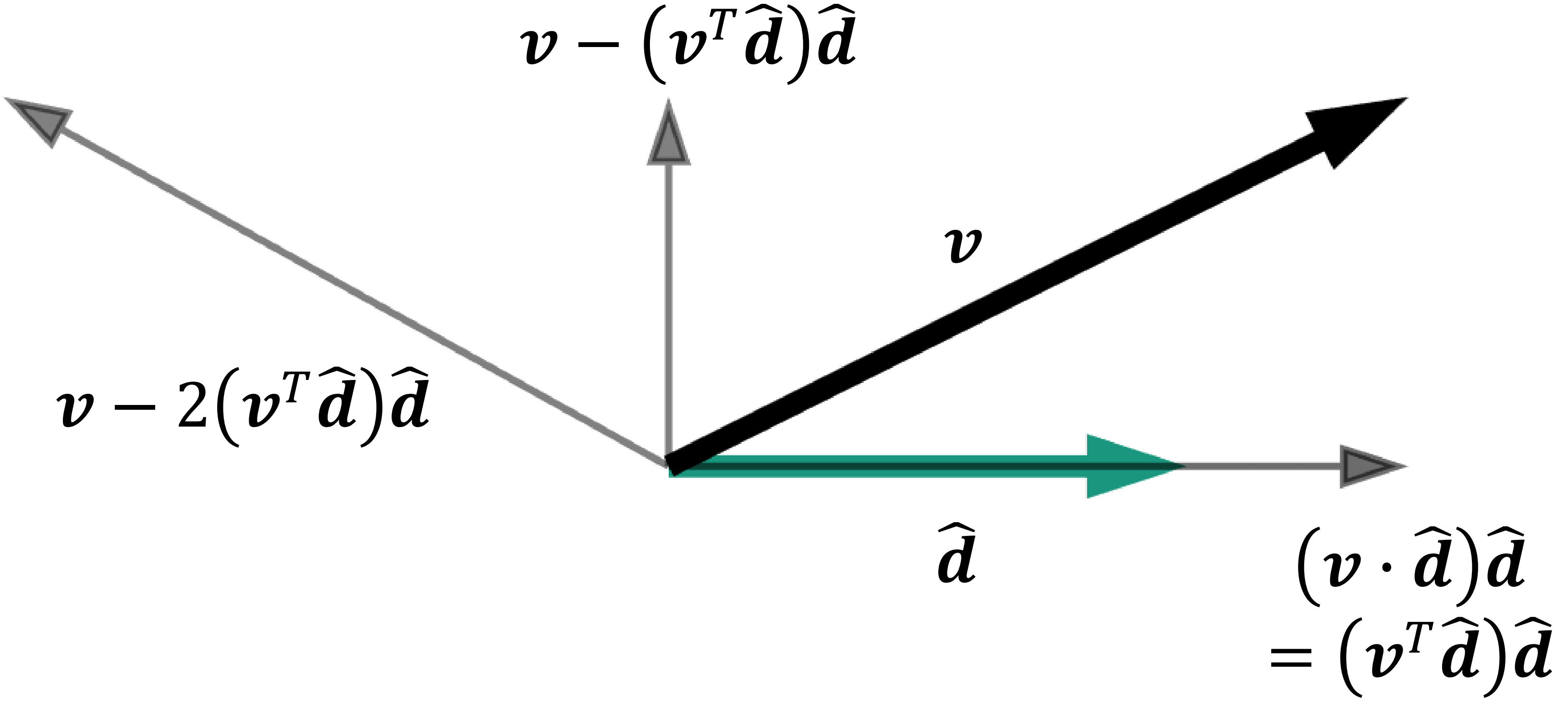
Outline

- Basic matrix and vector operations review
- Coordinate transforms
- Optimization: Least Squares, Total Least Squares, relation to decomposition methods

Basic Matrix and Vector Operations

Vector Directional Components

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



Orthogonal Vectors and Matrices

Orthogonal Vectors:

- Orthogonal vectors have no parallel component.

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

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).

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

Matrix Inverse

Definition:

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

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

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

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

For Orthogonal Matrices:

- By definition, is the transpose

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

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:

$$\sum_{i}^{n} (u_i)^2 = ||\boldsymbol{u}||_2^2 = \boldsymbol{u}^T \boldsymbol{u}$$

Gradient and Jacobian

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

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

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

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

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

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

Homogenous Coordinates Applications – Point at Infinity

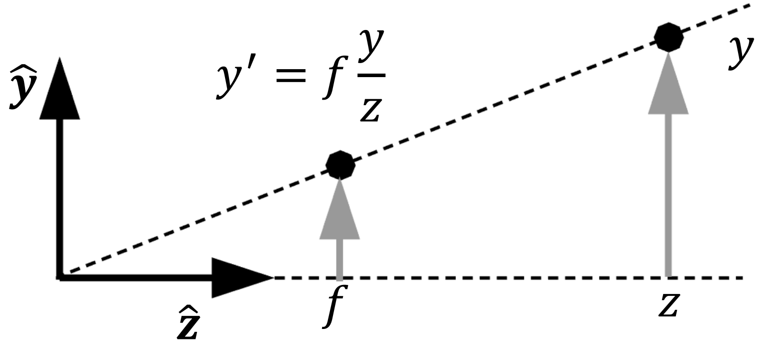
- Set $w = 0$
- Normative division causes resulting point to possess values at infinity.

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

$$p = \frac{\tilde{p}}{w}$$

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.

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

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

Matrix Transformations (1/2)

Matrix Transformations

- Apply matrix transformation $\boldsymbol{F}$ to transform a point:

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

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)

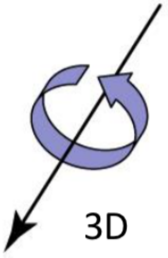
3d rotation about z / 2d rotation

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

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

3d rotation about x

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



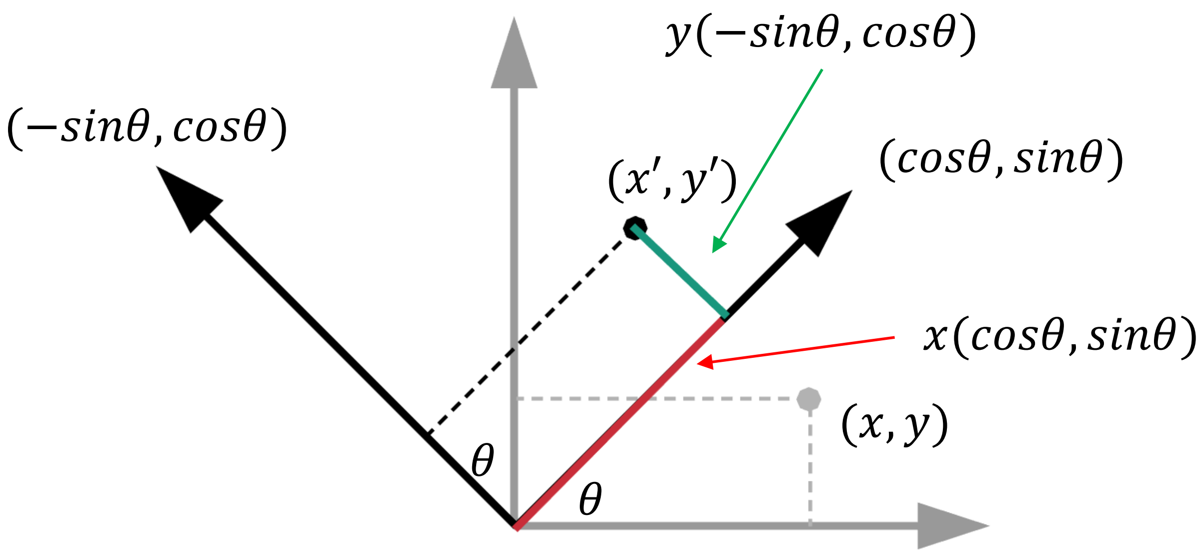
3d rotation about y

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

Rotation Matrix Equation Intuition

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

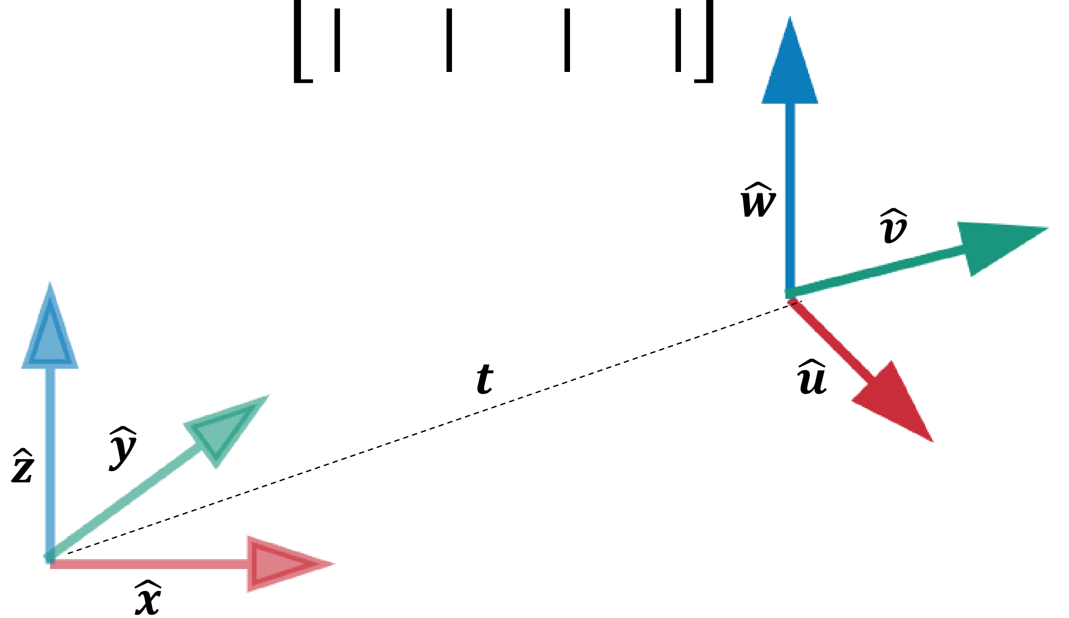
$$\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{cos\theta} \\ \boldsymbol{sin\theta} \\ \boldsymbol{0} \end{bmatrix} x + \begin{bmatrix} \boldsymbol{-sin\theta} \\ \boldsymbol{cos\theta}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



Coordinate Frames

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

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



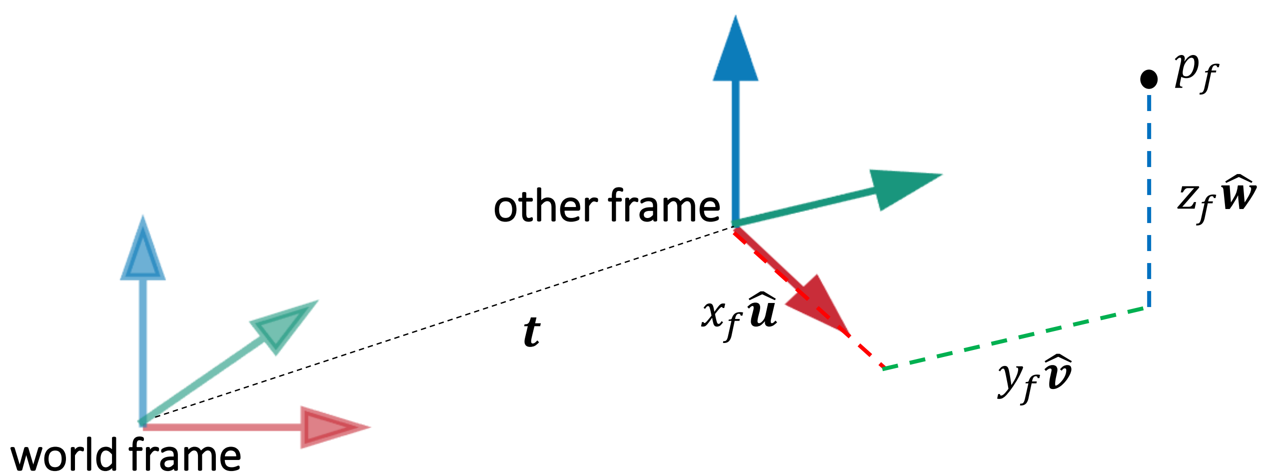
Coordinate Frame Transformations (1/2)

Goal: Represent point in different coordinate frames.

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

Example: Write p_f in world coordinates:

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



Coordinate Frame Transformations (2/2)

Summary:

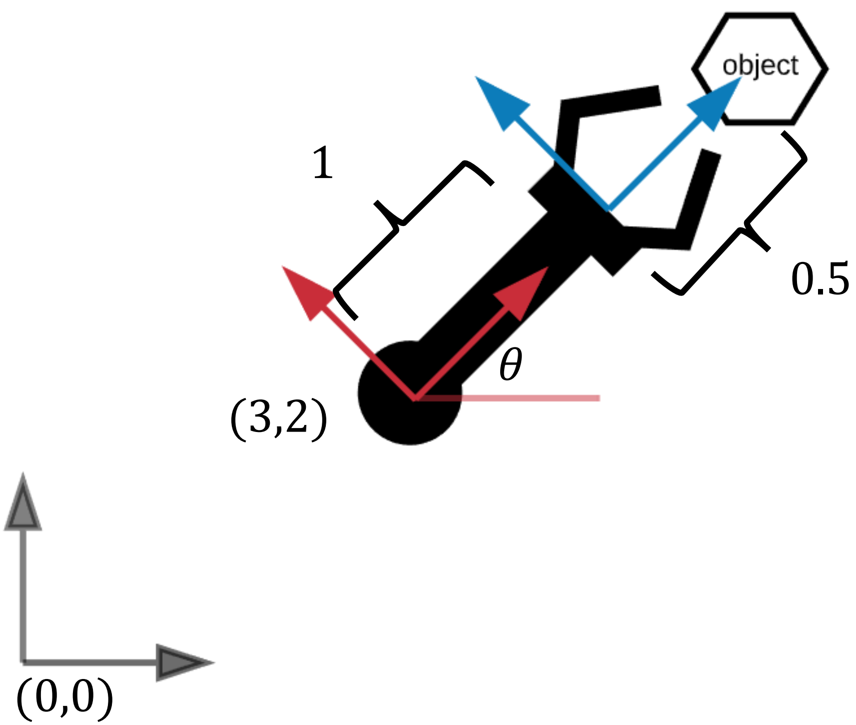
Coordinate Frame to World:	$p_w = F p_f$
World to Coordinate Frame:	$p_f = F^{-1} p_w$

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:

$$\boldsymbol{F} = \boldsymbol{T_1 R T_2} = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\frac{\pi}{4} & -sin\frac{\pi}{4} & 0 \\ sin\frac{\pi}{4} & cos\frac{\pi}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\frac{\pi}{4} & -sin\frac{\pi}{4} & 3 + \cos\frac{\pi}{4} \\ sin\frac{\pi}{4} & cos\frac{\pi}{4} & 2 + sin\frac{\pi}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

Arm Frame to World Coordinates:

- Note that point represented in homogenous coordinates

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

Matrix Transformations Applied to Multiple Points

- Multiple points can be stacked column-wise:

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

Optimization and Decomposition Methods

Least Squares Optimization

Motivation:

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

Least Squares Optimization:

$$Loss\ (L) = \sum_{i}^{M} \left(y^{(i)} - f_{\boldsymbol{w}}(\boldsymbol{x}^{(\boldsymbol{i})})\right)^2$$

$$\boldsymbol{w}^* = argmin_{\boldsymbol{w}}\ L$$

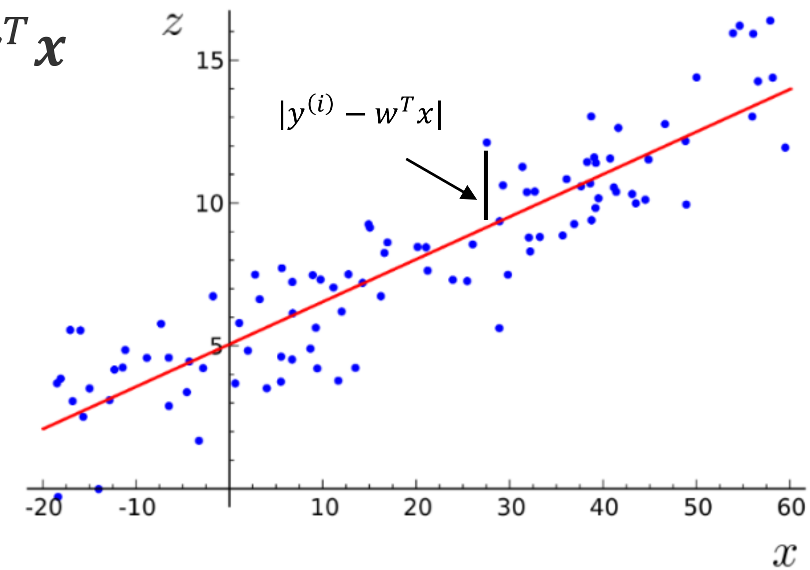
Linear Regression

Motivation:

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

$$f_w(x) = (1)w_0 + x_1 w_1 + \cdots x_n w_n = \boldsymbol{w}^T \boldsymbol{x}$$

$$argmin_w \sum_{i}^{M} \left(y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(\boldsymbol{i})}\right)^2$$



Linear Regression Matrix Format

- Can represent linear regression in purely matrix form since $\sum_i^M (u_i)^2 = \left|\left|\boldsymbol{u}\right|\right|^2 = \boldsymbol{u}^T\boldsymbol{u}$ where $\boldsymbol{u} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}$

$$argmin_{\boldsymbol{w}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$$\begin{bmatrix} \text{X} \\ \text{(MxN)} \end{bmatrix} \begin{bmatrix} \text{w} \\ \text{(Nx1)} \end{bmatrix} = \begin{bmatrix} \hat{y} \\ \text{(Mx1)} \end{bmatrix}$$

Linear Regression Solution

Closed-Form Solution:

- Solve minimization by taking gradient and setting to 0

$$\nabla_{\mathbf{w}}[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})] = 0$$

$$\rightarrow \nabla_{\boldsymbol{w}}[\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}] = 0$$

$$\rightarrow -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = 0$$

$$\rightarrow \boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Problem: Matrix inverse expensive or may be unstable

Incremental Numerical Solution: Gradient Descent

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

Eigenvalues and Eigenvectors

- Square matrix $\boldsymbol{A}$ has eigenvector $\boldsymbol{v}$ and eigenvalue λ under condition:

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

Interpretation:

- Describes case where transformation $\boldsymbol{A}$ on $\boldsymbol{v}$ equivalent to applying scaling factor λ

Eigendecomposition

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

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

$\boldsymbol{\Lambda}$: Diagonal matrix of eigenvalues

$\boldsymbol{V}$: Matrix of column vectors corresponding to eigenvectors

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

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

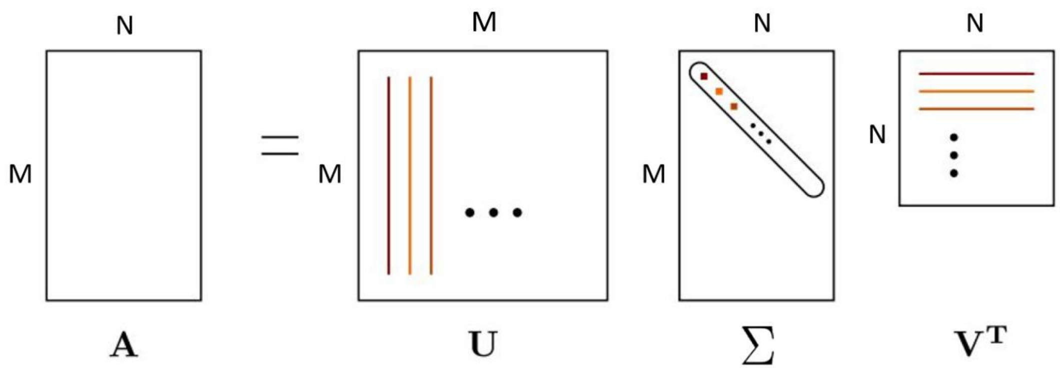
Interpretation as Coordinate Transform Operation:

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

Singular Value Decomposition (SVD)

- Decomposes MxN matrix $\boldsymbol{A}$ into:

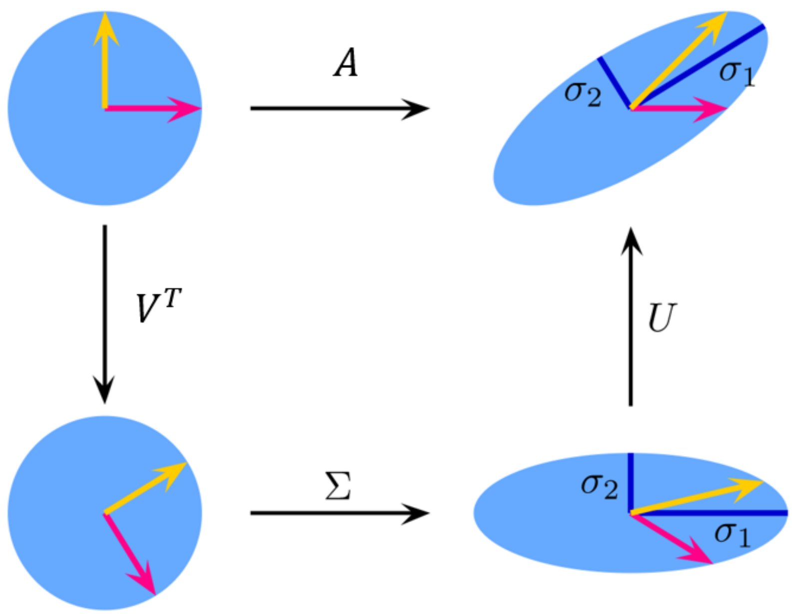
$$\boxed{\boldsymbol{A = U\Sigma V^T}}$$



$\boldsymbol{\Sigma}$	**Singular values**. MxN diagonal matrix. Positive square root of eigenvalues of A^TA or AA^T $(\sqrt{\lambda_i})$
$\boldsymbol{V}$	**Right-singular vectors**. NxN orthogonal matrix Eigenvectors of A^TA. Proof: $A^TA = (V\Sigma^TU^T)(U\Sigma V^T) = V\Lambda V^T$
$\boldsymbol{U}$	**Left-singular vectors**. MxM orthogonal matrix Eigenvectors of AA^T. Proof: $AA^T = (U\Sigma V^T)(V\Sigma^TU^T) = \mathrm{U}\Lambda U^T$

SVD Interpretation

1. Rotation to new coordinate system defined by $\boldsymbol{V}$
2. Scaling by $\boldsymbol{\Sigma}$
3. Final opposing rotation defined by $\boldsymbol{U}$



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

Application: Total Least Squares

Problem Statement:

$$argmin_{\boldsymbol{w}} \left|\left|\boldsymbol{Xw}\right|\right|^2 \text{ with constraint } \left|\left|\boldsymbol{w}\right|\right| = 1$$

Solution:

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

Intuition:

$$\left|\left|\boldsymbol{Xw}_{min}\right|\right|^2 = \boldsymbol{w}_{min}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_{min} = some\ min\ value = \lambda$$

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

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

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

Application: Pseudo-Inverse and Least Squares Linear Regression

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

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{T}} \rightarrow \boldsymbol{A}^{+} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}$$

$$\text{(since } \mathbf{A}^{+}\mathrm{A} = \left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}}\mathrm{A} = (\boldsymbol{V}\boldsymbol{\Sigma}^{-\mathbf{1}}\boldsymbol{U}^{\boldsymbol{T}})(U\Sigma V^{T}) = I\text{)}$$

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

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

(where $\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\boldsymbol{T}}$ is computed from SVD of X)

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