Machine Learning CS 4641

Linear Algebra Basics

Mahdi Roozbahani

(Covered by) Nimisha Roy

Lecturer, SCI, College of Computing, Georgia Tech

Director, Online Undergraduate Initiatives

Slides adapted from Max Roozbahani, Rod Borela, Le Song, Andres Mendez-Vazquez, and Chao Zhang

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition
- Matrix Calculus

Why linear algebra?

$X = \begin{bmatrix} \\ \\ \end{bmatrix}_{n \times d}$

dimensions

features

no. of datapoints

model training

⇓

Optimization

- Most data can be represented or stored in matrix-vector form
- Provides compact representation for sets of linear equations

$$\boxed{\begin{matrix} 4x_1 - 5x_2 = -13 \\ -2x_1 + 3x_2 = 9 \end{matrix}} \rightarrow \underbrace{\begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} -13 \\ 9 \end{bmatrix}}_{c_1, c_2} \rightarrow \mathbf{Ax} = \mathbf{b}$$

$c_1, c_2 \in \mathbb{R}$

2 2

- $\mathbf{A} \in \boxed{\mathbb{R}^{N \times D}}$ denotes a matrix with *N* rows and *D* columns, where elements belong to real numbers
- $\mathbf{x} \in \boxed{\mathbb{R}^{D}}$ denotes a vector with *D* real entries.

Linear algebra basics

- Transpose of a matrix results from flipping the rows and the columns. Given $\boxed{\mathbf{A} \in \mathbb{R}^{N \times D}}$, the transpose is $\boxed{\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{D \times N}}$

$$\mathbf{A} = \begin{bmatrix} 4 & -5 \\ -2 & 3 \end{bmatrix} \rightarrow \underline{\mathbf{A}^{\mathrm{T}}} = \begin{bmatrix} 4 & -2 \\ -5 & 3 \end{bmatrix}$$

- For each element of the matrix, the transpose can be written as $\mathrm{A}_{ij}^{T} = \mathrm{A}_{ji}$
- The following properties of the transposes are easily verified
  - $\left(\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{T}} = \mathbf{A}$
  - $(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$
  - $(\mathbf{A} + \mathbf{B})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}$

Linear algebra basics

- A square matrix $\boxed{\mathbf{A} \in \mathbb{R}^{D \times D}}$ is symmetric if $\boxed{\mathbf{A} = \mathbf{A}^{\mathrm{T}}}$ and it is skew-symmetric if $\boxed{\mathbf{A} = -\mathbf{A}^{\mathrm{T}}}$. Thus each matrix can be written as a sum of symmetric and anti-symmetric matrices:

$$\mathbf{A} = \underbrace{\frac{1}{2}(\mathbf{A} - \mathbf{A}^T)}_{H} + \underbrace{\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)}_{G}$$

$\boxed{A = A^T}$ ← A is symmetric

$\boxed{A = -A^T}$ ← A is skew-symmetric

$\boxed{H = -H^T}$ ← H is skew symmetric

$\boxed{G = G^T}$ ← G is symmetric

$H = A - A^T$

$H^T = (A - A^T)^T = A^T - A^{T^T} = A^T - A$

$= -(A - A^T)$

$= -H$

$G = A + A^T$

$G^T = (A + A^T)^T = A^T + A^{T^T} = A^T + A$

$= G$

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition

Norms

$\mathbb{R}^D \to \mathbb{R}$ $\quad \|\cdot\|$

Vector

$\|x\|$

- Norm of a vector $\|\mathbf{x}\|$ is informally a measure of the length of a vector
- More formally, a norm is any function $f: \mathbb{R}^D \to \mathbb{R}$ that satisfies:
    - For all $\mathbf{x} \in \mathbb{R}^D, f(\mathbf{x}) \geq 0$ (non-negativity)
    - $f(\mathbf{x}) = 0$ is and only if $x = 0$ (definiteness)
    - For $\mathbf{x} \in \mathbb{R}^D, t \in \mathbb{R}, f(t\mathbf{x}) = |t| f(\mathbf{x})$ (homogeneity)
    - For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D, f(\mathbf{x}+\mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (triangle inequality)

Norms

ℓ_p norms $\quad p = 1, 2, 3 \ldots$

$p=1 \rightarrow \ell_1$ norm $\rightarrow$ Manhattan distance

ℓ_2 norm $\rightarrow$ Euclidean distance

$x = [1, 3, -2]$

- Common norms used in machine learning are:
  - ℓ_2-norm: $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{D} x_i^2}$
  - ℓ_1-norm: $\|\mathbf{x}\|_1 = \sum_{i=1}^{D} |x_i|$
  - ℓ_∞-norm: $\|\mathbf{x}\|_\infty = \max_i |x_i|$

$\|x\|_2 = \sqrt{1^2 + 3^2 + (-2)^2} = \sqrt{14}$

$\|x\|_1 = |1| + |3| + |-2|$

$= 6$

$\|x\|_\infty = \max(|1|, |3|, |-2|)$

$= 3$

Georgia Tech

Norms

$\|x\| \sim \|x\|_2$

generally, we use $(\ell_2 \text{ norm})^2$

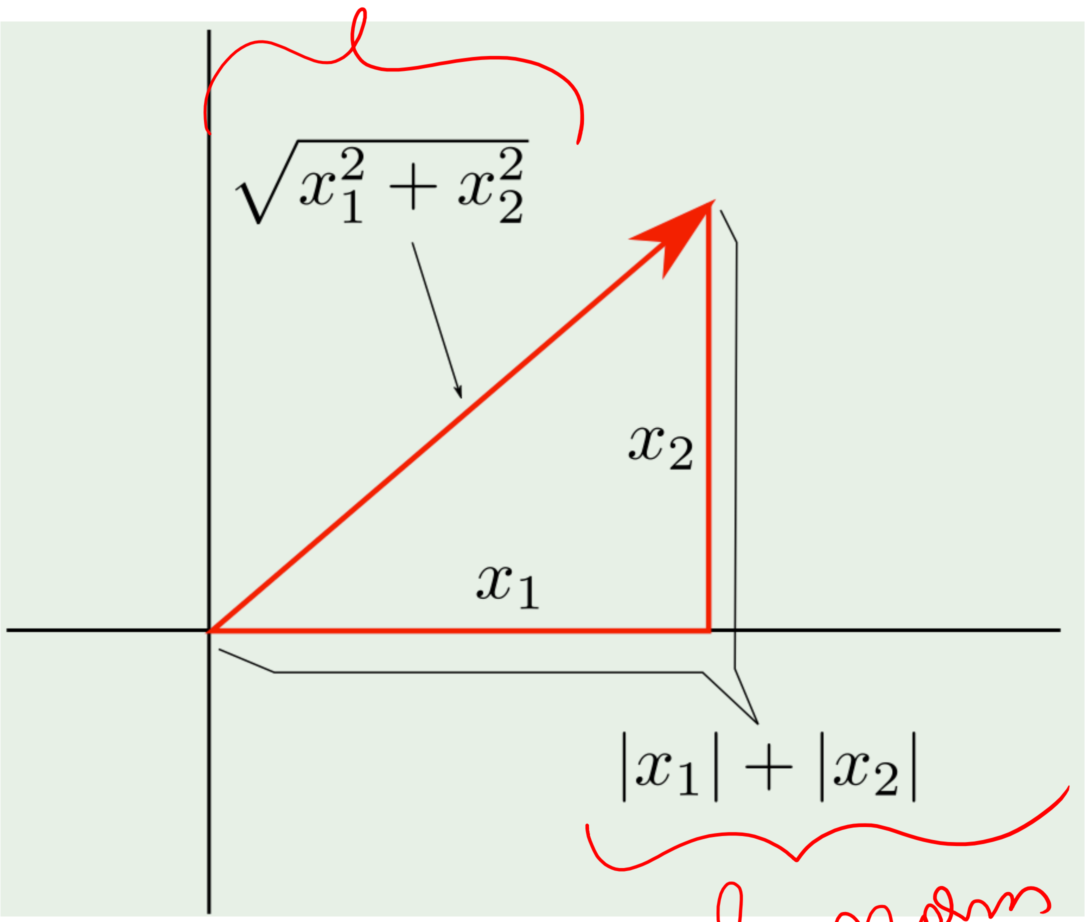
- All norms presented so far are examples of the family of ℓ_p norms, which are parametrized by a real number $p \geq 1$
  - $\boxed{\ell_p\text{-norm: } \|\mathbf{x}\|_p = \left(\sum_{i=1}^{D} |x_i|^p\right)^{\frac{1}{p}}}$
- Norms can be defined for matrices, such as the Frobenius norm
  - $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{D} \mathrm{A}_{\mathrm{ij}}^2} = \sqrt{\mathrm{tr}(\mathbf{A}^{\mathbf{T}}\mathbf{A})}$

$\|A\|_F$

Vector norm examples

- ℓ_1-norm and ℓ_2-norm

ℓ_2 norm

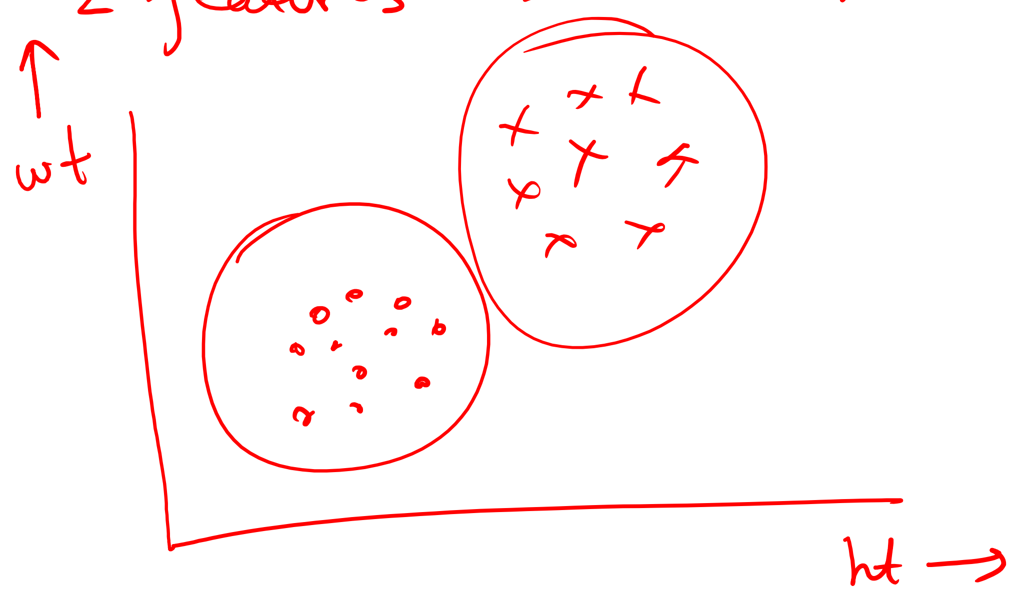


ℓ_1 norm

In context: how are norms useful in ML?

Unsupervised learning

X with 2 features → ht. & wt. .

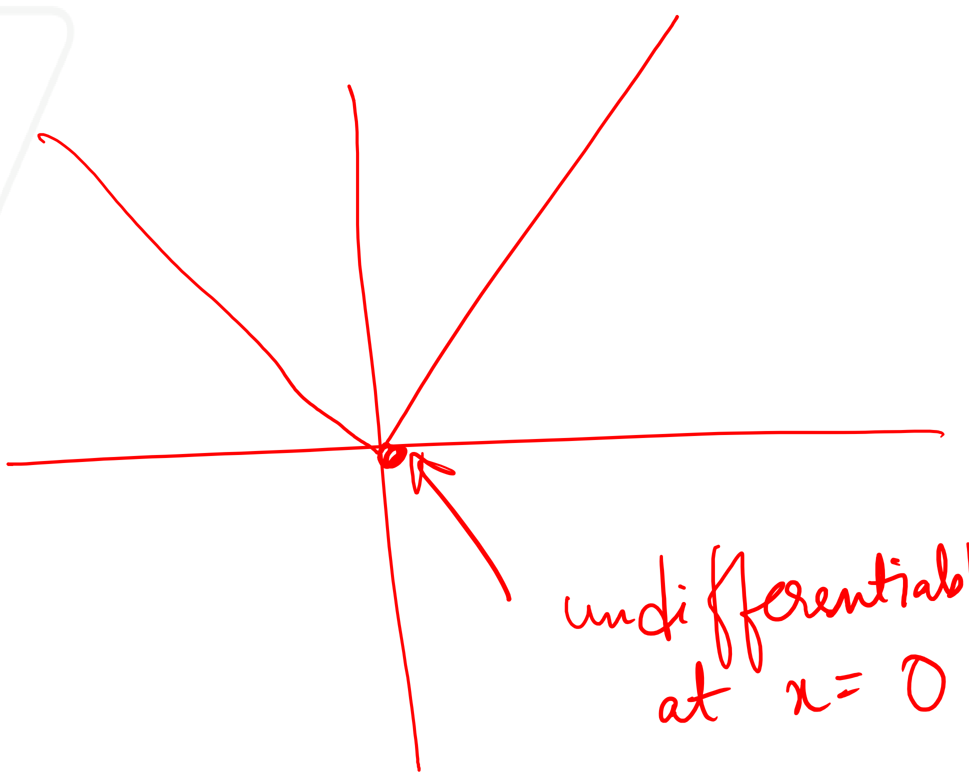


In context: choosing an appropriate norm

Manhattan vs. Euclidean

ℓ_1

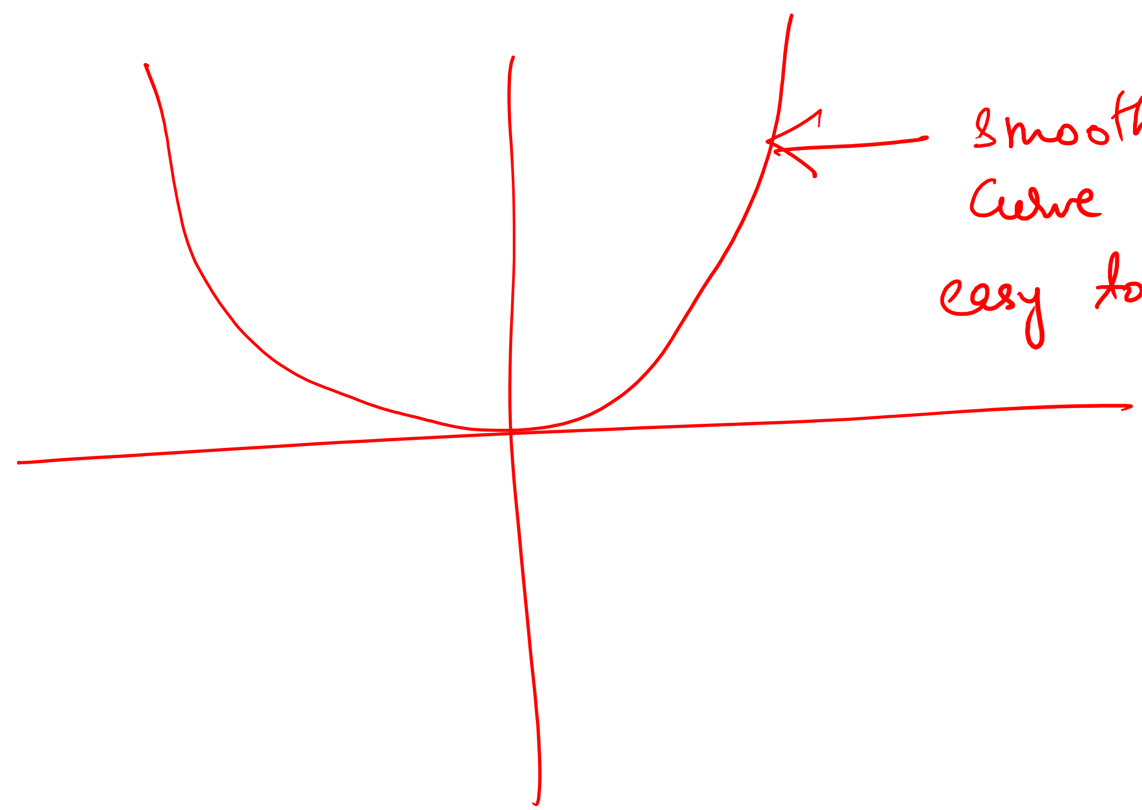
$y = |x|$



undifferentiable at $x = 0$

ℓ_2

$y = x^2$



smooth curve & very easy to differentiate

Special Matrices

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Do not Like List

1. For loops
2. Inverse

- The identity matrix, denoted by $I \in \mathbb{R}^{d \times d}$ is a square matrix with ones on the diagonal and zeros everywhere else

- A diagonal matrix is a matrix where all non-diagonal 'ELEMENTS' are 0. This is typically denoted as $D = diag\,(d_1, d_2, \ldots, d_d)$

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} = \begin{bmatrix} a^2 & 0 & 0 \\ 0 & b^2 & 0 \\ 0 & 0 & c^2 \end{bmatrix}$$

Why is diagonal matrix helpful?

- $\boxed{A^T A}$= elementwise squaring or all diagonal elements
- A^{-1} =1/A (elementwise)

$$A^{-1} = \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix}$$

Special Matrices

$x \cdot y = 0$

$$U = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \end{array}$$

- Two vectors x, y ∈ $\mathbb{R}^d$ are orthogonal if $x.y$ = 0. A square matrix U ∈ $\mathbb{R}^{d \times d}$ is **Orthonormal** if all its columns are orthogonal to each other and are normalized

$c_1 \cdot c_2 = 0$

$c_1 \cdot c_3 = 0$

$c_2 \cdot c_3 = 0$

$\|c_1\|_2 = 1$

$\|c_2\|_2 = \|c_3\|_2 = 1$

unitary ≈ orthonormal

- It follows from orthogonality and normality that
  - $\boxed{U^T U = I = U U^T}$
  - $\|Ux\|_2 = \|x\|_2$

Is the inverse of a unitary matrix equal to its transpose?

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

Orthonormal Matrices do not STRETCH OR SHRINK a vector. It only ROTATES it.

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition

Multiplications

Matrix multiplication is a bunch of dot products

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ d_1 & e_1 & f_1 \\ g_1 & h_1 & i_1 \end{bmatrix} \quad B = \begin{bmatrix} a_2 & b_2 & c_2 \\ d_2 & e_2 & f_2 \\ g_2 & h_2 & i_2 \end{bmatrix}$$

$A*B$

$A*B_{(1,1)} = a_1a_2 + b_1d_2 + c_1g_2$

- The product of two matrices $\mathbf{A} \in \mathbb{R}^{N\times D}$ and $\mathbf{B} \in \mathbb{R}^{D\times P}$ is given by $\mathbf{C} \in \mathbb{R}^{N\times P}$, where $C_{ij} = \sum_{k=1}^{D} A_{ik}B_{kj}$

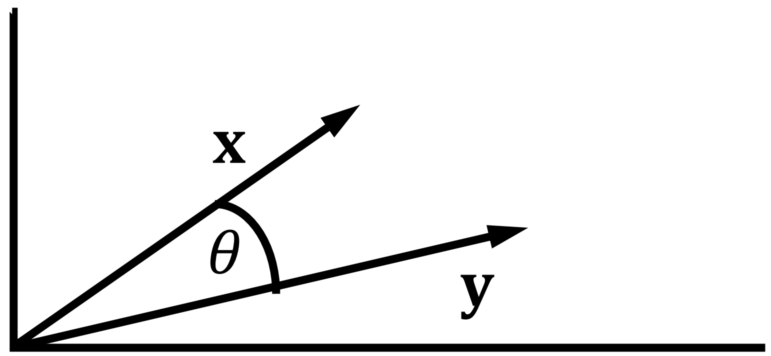
- Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$, the term $\mathbf{x} \cdot \mathbf{y}$ (or $\mathbf{x}^{\mathrm{T}}\mathbf{y}$) is called the inner product or dot product of the vectors, and is a real number given by $\sum_{k=1}^{D} x_i y_i$. For example,

$$\underset{1\times 3}{\mathbf{x}^{\mathrm{T}}}\,\underset{3\times 1}{\mathbf{y}} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \sum_{i=1}^{3} x_i y_i$$

Multiplications

project x onto y $= x \cdot \frac{y}{\|y\|_2}$

- The dot product also has a geometrical interpretation, for vectors in $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ with an angle θ between them:

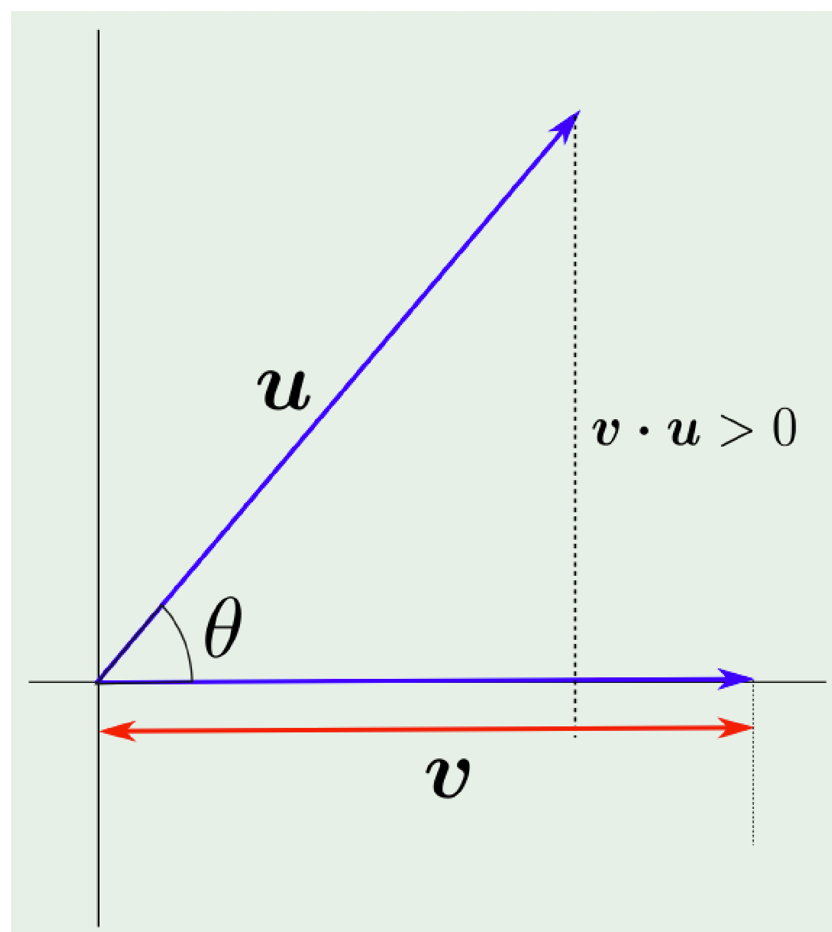


$$\mathbf{x} \cdot \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos\theta$$

$\|x\|_2 \cos\theta$

Inner product properties

- The inner product is a measure of correlation between two vectors scaled by the norms of the vectors
- Here correlation term is used in the loose sense of directional alignment – not statistical sense



$>0 \quad >0$

$$u \cdot v = \|u\|_2 \cdot \|v\|_2 \cdot \cos\theta$$

>0

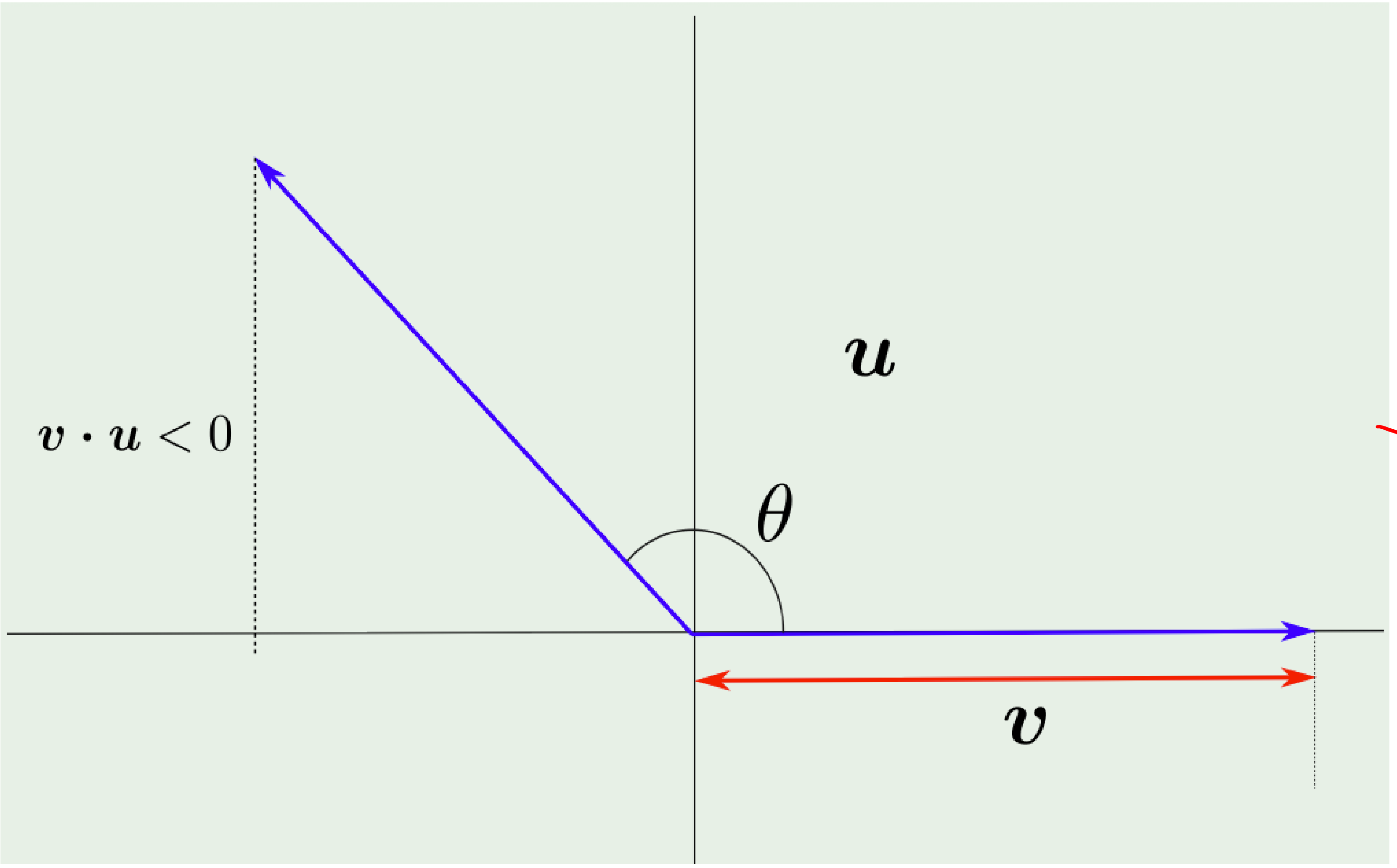
$\theta < 90$

$\cos\theta > 0$

$u \cdot v > 0$

Inner product properties

- The inner product is a measure of correlation between two vectors, scaled by the norms of the vectors



$\theta > 90^\circ$

$\cos\theta < 0$

>0

$u \cdot v = \|u\|_2 \cdot \|v\|_2 \cdot \cos\theta$

>0

$-ve$

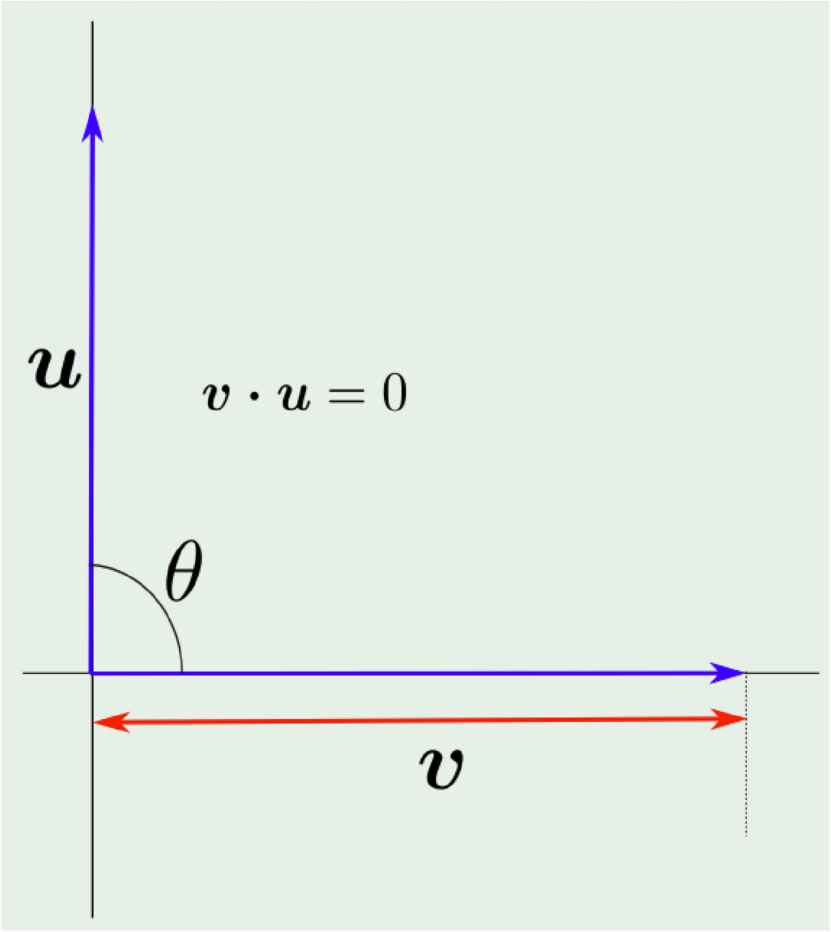
Hence,
$u \cdot v < 0$

Inner product properties

Dot product is a linear operation.

- The inner product is a measure of correlation between two vectors scaled by the norms of the vectors

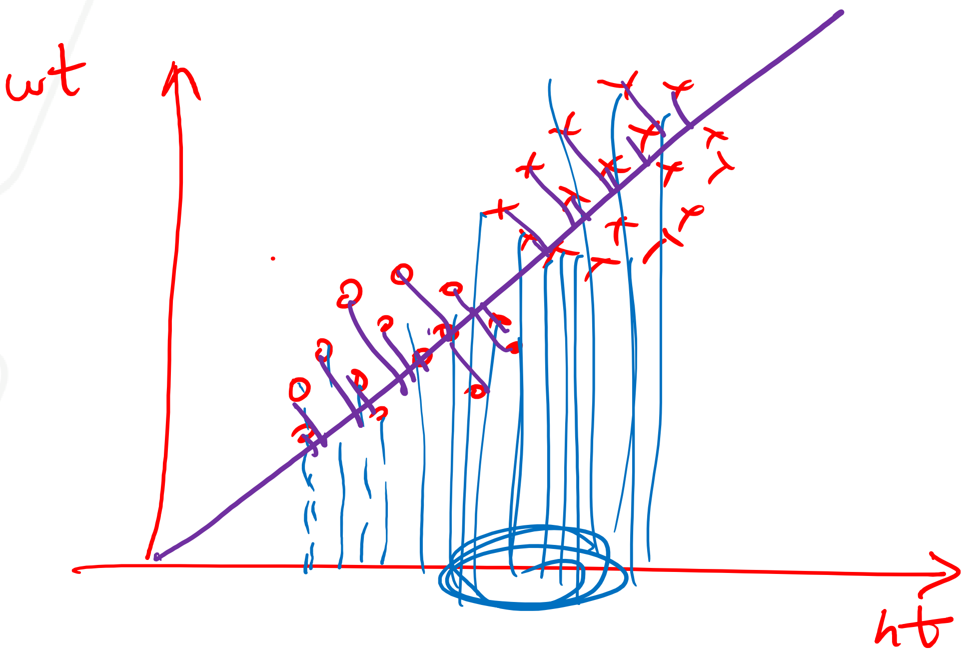
Linearly independent vectors

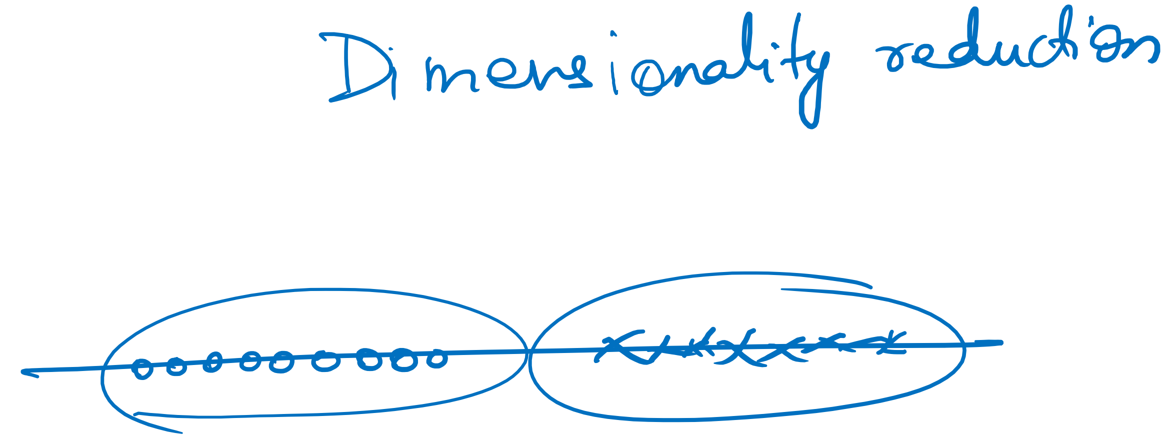


$u \cdot v = 0$

$\cos 90 = 0$

In context: how is the inner product useful in ML?
Projecting data onto a new direction





This will be helpful in dimensionality reduction, classification and feature engineering

Multiplications

- Given two vectors $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^D$ the term $\mathbf{x}\mathbf{y}^\mathrm{T} \in \mathbb{R}^{N\times D}$ is called the outer product of the vectors and is a matrix given by $\left(x_i y_j\right)^\mathrm{T} = x_i y_j$.
- For example, $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{y} \in \mathbb{R}^2$

$$\mathbf{x}\mathbf{y}^\mathrm{T} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} [y1 \quad y2] = \begin{bmatrix} x_1y_1 & x_1y_2 \\ x_2y_1 & x_2y_2 \\ x_3y_1 & x_3y_2 \end{bmatrix}$$

3x1 1x2

3x2

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Matrix Decomposition

$A^{-1}Aw = A^{-1}b$

$w = A^{-1}b$ —— X^TX

Linear Independence and Matrix Rank

- A set of vectors $\{x_1, \ x_2, \ldots, x_d\} \subset \mathbb{R}^d$ are said to be ***(linearly) independent*** if no vector can be represented as a linear combination of the remaining vectors. That is if

$$x_d = \sum_{i=1}^{d-1} \alpha_i x_i$$

$$A = \begin{array}{c} \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \\ \begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \end{bmatrix} \end{array}$$

for some scalar values $\alpha_1, \alpha_2, \ldots \in \mathbb{R}$ then we say that the vectors are linearly ***dependent***; otherwise the vectors are linearly independent

$$A = \begin{array}{c} \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \quad \text{rank}(A) = 3$$

A is a linearly independent matrix

full rank

$$B = \begin{array}{c} \begin{array}{ccc} c_1 & c_2 & c_3 \end{array} \\ \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 10 \end{bmatrix} \end{array}$$

$c_1 = \alpha_1 c_2 + \alpha_2 c_3$

$c_2 = 2c_1 + 0 \cdot c_3$

c_3 ✗

$\text{rank}(B) = 2$

In ML, we want as many linearly independent columns as possible

Georgia Tech

tall matrix $C = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \\ 5 & 8 \end{bmatrix}$ column rank = 2, row rank = 2

wide matrix $D = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 2 & 4 & 6 & 7 \end{bmatrix}$ row rank = 2 = column rank

Linear Independence and Matrix Rank

- The **column rank** of a matrix $A \in \mathbb{R}^{n \times d}$ is the size of the largest subset of columns of A that constitute a linearly independent set. **Row rank** of a matrix is defined similarly for rows of a matrix.

It can be easily shown that the row and column ranks are equivalent, therefore we shall refer only to the **rank** of a matrix.

In general, for a full rank rectangular matrix, rank is the min of number of rows and number of columns.

Georgia Tech

Matrix Rank: Examples

What are the ranks for the following matrices? How about an identity matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$$

$\text{rank}(A) = 1$

Rank Deficiency means that the number of linearly independent vectors in our set is smaller than the smallest dimension

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

$\text{rank}(B) = 3$

$c_1 - 1 = c_2$

$c_1 + (-1)c_4 = c_2$

full rank matrix

Matrix Inverse

- The inverse of a square matrix $A \in \mathbb{R}^{d \times d}$ is denoted A^{-1} and is the unique matrix such that $A^{-1}A = I = AA^{-1}$

- For some square matrices A^{-1} may not exist, and we say that A is **singular or non-invertible.** In order for A to have an inverse, A must be **full rank.**

- For non-square matrices the inverse, denoted by A^{+} ,is given by $A^{+} = (A^{\top}A)^{-1}A^{\top}$ called the **pseudo inverse**

$\left(A^{T}_{d \times n} A_{n \times d}\right)^{-1} \cdot A^{T}$

In context: why is matrix rank important in ML?
Dataset preprocessing

If full rank $\longrightarrow$ V. Good

If not full rank $\longrightarrow$ Either combine dependent features or just get rid of it

$\underbrace{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}}$

No redundancy

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition

Matrix Trace

- The trace of a matrix $A \in \mathbb{R}^{d \times d}$, denoted as $\boldsymbol{tr(A)}$, is the sum of the diagonal elements in the matrix

$$tr(A) = \sum_{i=1}^{d} A_{ii}$$

- The trace has the following properties
  - For $A \in \mathbb{R}^{d \times d}$, $tr(A) = trA^{\top}$
  - For $A, B \in \mathbb{R}^{d \times d}$, $tr(A + B) = tr(A) + tr(B)$
  - For $A \in \mathbb{R}^{d \times d}$, $t \in \mathbb{R}$, $tr(tA) = t \cdot tr(A)$
  - For A, B, C such that ABC is a square matrix $tr(ABC) = tr(BCA) = tr(CAB)$

- The trace of a matrix helps us easily compute norms and eigenvalues of matrices as we will see later

Matrix Determinant

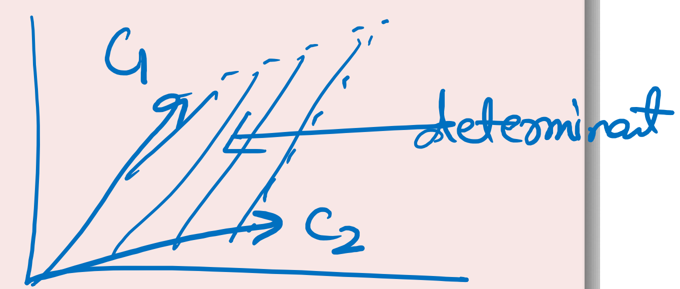
$$\begin{bmatrix} \cancel{a} & \cancel{b} & \cancel{c} \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$a(ei - fh) - b(di - fg) + c(dh - eg)$$

Definition (Determinant)

The determinant of a square matrix A, denoted by $|A|$, is defined as

$$\det(A) = \sum_{j=1}^{d} (-1)^{i+j} a_{ij} M_{ij}$$



where M_{ij} is determinant of matrix A without the row i and column j.

For a 2×2 matrix $A = \begin{array}{c} \begin{array}{cc} c_1 & c_2 \end{array} \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array}$

$$|A| = ad - bc$$

Linearly dependent columns → matrix is not full rank → determinant 0 → non-invertible matrix

Georgia Tech

Properties of Matrix Determinant

Basic Properties

- $|A| = \left|A^T\right|$
- $|AB| = |A|\,|B|$
- $|A| = 0$ if and only if A is not invertible
- If A is invertible, then $\left|A^{-1}\right| = \frac{1}{|A|}$.

In context: how is the determinant useful in ML?
Matrix inversion

$A w = b$ $\quad w = A^{-1} b \quad$ A is not invertible

linear regression

Testing out if det(A) = 0 before trying to invert a matrix saves on computation

In that case, we can try out numerical methods

Gradient Descent

Recap

- Special Matrices

diagonal → $X^T X$ → square the diagonal terms

diagonal → X^{-1} → reciprocal of diagonal entries

Orthonormal → $X^{-1} = X^T$

Orthonormal → $\|Xx\|_2 = \|x\|_2$ → only allow rotation

- Norms

$\mathbb{R}^d \rightarrow \mathbb{R}$

"length"

$l_p \rightarrow l_2$ norm

To compute distances between datapoints in clustering

- Matrix Multiplications → is a bunch of dot product

gives us information about directional alignment / projection

$x \cdot y > 0$

$x \cdot y < 0$

$x \cdot y = 0$ → LINEARLY INDEPENDENT

- Matrix Inversion

$Aw = b$

$w = A^{-1} b$

2 indications of non-invertible matrix

Rank deficient → X not invertible

determinant is 0 → X not invertible

- Trace and Determinant

→ Area of formed by vectors

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition

Eigenvalues and Eigenvectors

- Given a square matrix $A \in \mathbb{R}^{d \times d}$ we say that $\lambda \in \mathbb{C}$ is an eigenvalue of A and $x \in \mathbb{C}^d$ is an eigenvector if

eigenvector eigenvalue

$$\boxed{Ax = \lambda x,} \qquad x \neq 0$$

dxd

- Intuitively this means that upon multiplying the matrix A with a vector x, we get the same vector, but scaled by a parameter λ

stretch or shrink ONLY

- So, eigenvectors are the special vectors whose direction is preserved under transformation (no rotation)

Matrix Eigen Decomposition

- All the eigenvectors can be written together as $\boxed{AX = X\Lambda}$ where the columns of X are the eigenvectors of A, and Λ is a diagonal matrix whose elements are eigenvalues of A

$A_{d\times d}$

$$\underbrace{\begin{bmatrix} A \end{bmatrix}}_{d\times d} \underbrace{\begin{bmatrix} e_1 & e_2 & \cdots & e_d \end{bmatrix}}_{X \quad d\times d} = \underbrace{\begin{bmatrix} X \end{bmatrix}}_{d\times d} \underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 & & \\ 0 & \lambda_2 & 0 & & \\ 0 & & \lambda_3 & 0 & \\ & & & \ddots & \\ & & & & \lambda_d \end{bmatrix}}_{\Lambda \quad d\times d}$$

$$\boxed{AX = X\Lambda}$$

$$AXX^{-1} = X\Lambda X^{-1}$$

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

Matrix Eigen Decomposition

- If the eigenvectors of A are invertible, then $\boxed{A = X\Lambda X^{-1}}$
- There are several properties of eigenvalues and eigenvectors
  - $Tr(A) = \sum_{i=1}^{d} \lambda_i$
  - $|A| = \prod_{i=1}^{d} \lambda_i$
  - If A is non-singular then $1/\lambda_i$ are the eigenvalues of A^{-1}
  - The eigenvalues of a diagonal matrix are the diagonal elements of the matrix itself!

3rd important property diagonal matrix

Matrix Eigen Decomposition

- For a symmetric matrix $\mathbf{A}$ it can be shown that eigenvalues are real and the eigenvectors are orthonormal. Thus it can be represented as $X\boldsymbol{\Lambda}\boldsymbol{X}^{\mathrm{T}}$

$A \leftarrow$ square

also symmetric

$X \leftarrow$ bunch of eigenvectors

ORTHONORMAL

$X^{-1} = X^{T}$

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

square

symmetric

orthonormal

diagonal

Eigenvalues and Eigenvectors

Geometrically, we are transforming the matrix A (if symmetric) from its original orthonormal basis/coordinates to a new set of orthonormal basis x with magnitude as **λ**

- If A is symmetric, eigenvectors are orthogonal. So, eigenvectors form an orthonormal basis
- Eigenvectors define the directions along which the transformation acts independently
- In the eigenvector basis, there is no rotation or shear, only scaling by the eigenvalues
- Each eigenvalue λi controls the amount of stretching or compression along eigenvector xi

Computing Eigenvalues and Eigenvectors

- We can rewrite the original equation in the following manner

$|A-\lambda I| \neq 0$

$(A-\lambda I)^{-1}$

$|A-\lambda I| = 0$

$$Ax = \lambda x, \qquad x \neq 0$$
$$\Rightarrow \boxed{(A-\lambda I)\,x = 0,} \qquad x \neq 0$$

$(A-\lambda I)^{-1}(A-\lambda I)\,x = 0$

$I \cdot x = 0 \Rightarrow x = 0$

- This is only possible if $(A-\lambda I)$ is singular, that is $|(A-\lambda I)| = 0$.

- Thus, eigenvalues and eigenvectors can be computed.
  - Compute the determinant of $A - \lambda I$.
    - This results in a polynomial of degree d.
  - Find the roots of the polynomial by equating it to zero.
    - The d roots are the d eigenvalues of A. They make $A - \lambda I$ singular.
  - For each eigenvalue λ, solve $(A-\lambda I)\,x$ to find an eigenvector x

Eigenvalue Example

$|A - \lambda I| = 0$

$\left| \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$

$= \left| \begin{bmatrix} 1-\lambda & 2 \\ 3 & -4-\lambda \end{bmatrix} \right| = 0$

$= (1-\lambda)(-4-\lambda) - 6 = 0$

$= -4 - \lambda + 4\lambda + \lambda^2 - 6 = 0$

Matrix $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}_{2\times 2}$

1. Compute the determinant of $\mathbf{A} - \lambda\mathbf{I}$

$$\mathbf{A} - \lambda\mathbf{I} = \begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & 2 \\ 3 & -4-\lambda \end{bmatrix}$$

$$|\mathbf{A} - \lambda\mathbf{I}| = (1-\lambda)(-4-\lambda) - 6$$

2. Find the roots of the polynomial equating it to zero

$$|\mathbf{A} - \lambda\mathbf{I}| = 0 \rightarrow (1-\lambda)(-4-\lambda) - 6 = 0 \rightarrow \begin{cases} \lambda_1 = -5 \\ \lambda_2 = 2 \end{cases}$$

Eigenvalue Example

eigenvectors $\rightarrow$ we only care about direction

$Ax = \lambda x$

$A cx = c Ax = c \lambda x$

$= \lambda (cx)$

$A(cx) = \lambda (cx)$

λ is an eigenvalue for x & cx

3. For each eigenvalue λ solve $(\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = \mathbf{0}$ to find eigenvector $\mathbf{x}$

$$\begin{bmatrix} 1-\lambda & 2 \\ 3 & -4-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \begin{cases} (1-\lambda)x_1 + 2x_2 = 0 \\ 3x_1 - (4+\lambda)x_2 = 0 \end{cases}$$

Eigenvector for $\lambda_1 = -5$

$$\boxed{\begin{cases} 6x_1 + 2x_2 = 0 \\ 3x_1 + x_2 = 0 \end{cases}} \rightarrow \mathbf{x}_1 = \boxed{\begin{bmatrix} 1 \\ -3 \end{bmatrix}} \text{ or } \boxed{\begin{bmatrix} 0.3162 \\ -0.9487 \end{bmatrix}}$$

Eigenvector for $\lambda_2 = 2$

$$\begin{cases} -x_1 + 2x_2 = 0 \\ 3x_1 - 6x_2 = 0 \end{cases} \rightarrow \boxed{\mathbf{x}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 0.8944 \\ 0.4472 \end{bmatrix}}$$

Georgia Tech

Can a matrix have the same eigenvalues?

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$(1, 1, 1)$

If two vectors are linearly independent, does it mean they are orthogonal to each other?

if orthogonal

$\downarrow$

linearly independent

$$\begin{bmatrix} 1 & 5 \\ 2 & 7 \\ 3 & 9 \end{bmatrix}$$

Vice Versa is correct

In context: how is eigenvalues and eigenvectors helpful in ML?

- Eigenvectors are “special directions” where a matrix doesn’t rotate or shear — it only stretches or shrinks.
- The stretch amount is the eigenvalue.

- Many ML tools boil down to understanding **data transformations** (via matrices). Eigenvectors/eigenvalues tell us the **essential structure** of those transformations:

- Eigenvectors reveal the fundamental directions along which the transformation acts independently.
- Eigenvalues quantify the strength of that action (how much stretching or shrinking occurs).
- This decomposition simplifies complex transformations into basic “rotate–stretch–rotate back” operations, making it easier to analyze patterns, reduce dimensionality, or understand stability and sensitivity in models.

Georgia Tech

Outline

- Linear Algebra Basics
- Norms
- Multiplications
- Matrix Inversion
- Trace and Determinant
- Eigen Values and Eigen Vectors
- Singular Value Decomposition

Eigenvalue decomposition

$$A = X \Lambda X^T$$

Square & symmetric

Singular Value Decomposition

$$\bar{X} = \begin{bmatrix} a_1 - \mu_1 \\ a_2 - \mu_1 \\ a_3 - \mu_1 \\ \vdots \\ a_n - \mu_1 \end{bmatrix}_{n \times d} \qquad \mu_1 = \frac{a_1 + \cdots + a_n}{n}$$

c_1

center

$$\bar{X}_{n\times d}$$

n: datapoints
d: dimensions
X is a centered matrix

orthonormal

$$\bar{X} = \boxed{U\Sigma V^T}$$

$$U_{n\times n} \rightarrow unitary\ matrix \rightarrow U \times U^T = I$$

$$\Sigma_{n\times d} \rightarrow diagonal\ matrix$$

$$X = [\ ]_{100{,}000 \times 100}$$

$$np.svd(X_{n\times d}) = \underset{n\times n}{U}\ \underset{n\times d}{\Sigma}\ \underset{d\times d}{V}$$

$$V_{d\times d} \rightarrow unitary\ matrix \rightarrow V \times V^T = I$$

$$X = \begin{bmatrix} u_{1\times 1} & \cdots & \cdots & \cdots & u_{1\times n} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ u_{1\times 1} & \cdots & \cdots & \cdots & u_{n\times n} \end{bmatrix} \times \begin{bmatrix} \sum_{1\times 1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sum_{d\times d} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} v_{1\times 1} & \cdots & \cdots & \cdots & v_{1\times d} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ v_{d\times 1} & \cdots & \cdots & \cdots & v_{d\times d} \end{bmatrix}$$

$n \times d$ $\qquad U \qquad \boxed{n\times n} \qquad \Sigma \qquad n\times d \qquad V^T \qquad d\times d$

$$d < n$$

Covariance matrix:

$(ab)^T = b^T a^T$

$\frac{\bar{X}^T \bar{X}}{n}$

DO NOT LIKE LIST
① FOR LOOP
② INVERSE
③ MATRIX MULP.

$C = \frac{\bar{X}^T \bar{X}}{N}$

$$\boxed{C_{d\times d} = \frac{\underset{10000\times 100}{\bar{X}^T}\ \underset{100\times 10000}{\bar{X}}}{n}} = C_{100k \times 100k}$$

$n \leftarrow$ no. of datapoints

$\bar{X} = U \Sigma V^T$

$C = \frac{(U\Sigma V^T)^T (U \Sigma V^T)}{N} = \frac{V \Sigma^T \boxed{U^T U} \Sigma V^T}{N} = \frac{V \Sigma^T \Sigma V^T}{N} = \frac{V \Sigma^2 V^T}{N}$

$$\bar{X} = U\Sigma V^T$$

$$C = \frac{\bar{X}^T \bar{X}}{n}$$

$$C = \frac{V\Sigma^T U^T U \Sigma V^T}{n} = \frac{V\Sigma^2 V^T}{n}$$

diagonal

$AX = X\Lambda$

square

Orthonormal

diagonal

$$\boxed{C V = V \frac{\Sigma^2}{N}}$$

square

$C = \frac{V \Sigma^2 V^T}{N}$

$CV = \frac{V \Sigma^2 V^T V}{N} = \frac{V \Sigma^2}{n}$

$$C = \frac{V\Sigma^2V^T}{n} = V\frac{\Sigma^2}{n}V^T$$

$$\begin{bmatrix} \Sigma_1 & & & \\ & \Sigma_2 & & \\ & & \ddots & \\ & & & \Sigma_d \\ 0 & 0 & 0 & 0 \end{bmatrix}_{n \times d}$$

$$CV = V\frac{\Sigma^2}{n}V^TV = V\frac{\Sigma^2}{n}$$

$$CV = V\Lambda$$

Remember: $$AX = X\Lambda$$

$\lambda_i = \frac{\Sigma_i^2}{n}$ ➔ The eigenvalues of covariance matrix

λ_i: Eigenvalue of C or covariance matrix

Σ_i: Singular value of X matrix

So, we can **directly** calculate eigenvalue and eigenvectors of a covariance matrix by having the singular value decomposition of matrix X

So, we can **directly** calculate eigenvalue and eigenvectors of a covariance matrix by having the singular value decomposition of matrix X

Why is this helpful?

Because SVD **always exists** for **any** matrix (rectangular, non-symmetric) and is **more numerically stable**. That's why in ML (e.g., PCA) we usually prefer **SVD of X** over eigen-decomposition of $\boldsymbol{X^T X}$.

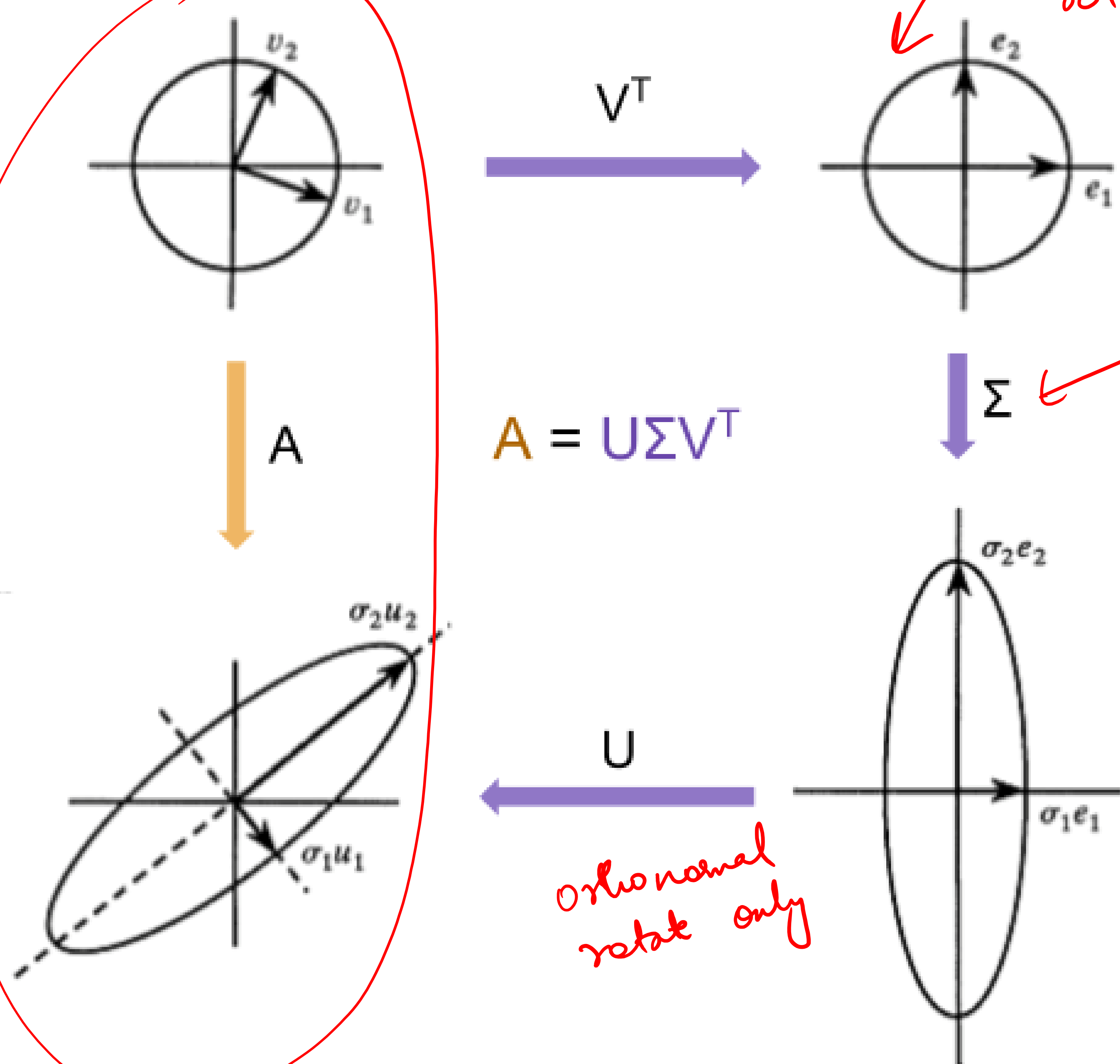
Georgia Tech

Geometric Meaning of SVD

$Xx = U\Sigma V^T x$

SVD tells us that *any* linear transformation can be decomposed into:
a rotation → a scaling → another rotation.

rotating only



diagonal stretching only

orthonormal rotate only

Image Credit: Kevin Binz

In context: why is SVD and covariance matrix useful in ML?

SVD (X=$U\Sigma V^T$)

•Finds the **most informative directions** in the data (columns of V); σ_i tell how strong each is.
•Gives the **best low-rank approximation** → dimensionality reduction, compression, **denoising** (Low values of σ_i means often represents noise)

Covariance (C= $\frac{X^T X}{n}$ with centered X)

•Summarizes **spread** (variances) and **relationships** (correlations) between features.
•Drives **PCA**: eigenvectors = principal components; eigenvalues = **explained variance**.
•Basis for **whitening/standardization**, which helps many models.

→ Covariance tells us what varies; **SVD** shows how to **use** it—rotate to those directions and keep the big values.

Georgia Tech

Quick Knowledge Check

What property defines an orthonormal matrix Q?

A. $Q^T Q = I$ |B. $Q = Q^T$ |C. Entries are 0 or 1 | D. Determinant always positive

$Q^T = Q^{-1}$

When is a square matrix not invertible?

A. Columns linearly dependent | B. Matrix symmetric | C. Determinant nonzero | D. Columns orthogonal

What happens when a matrix acts on its eigenvector?

A. Direction unchanged | B. Vector rotates | C. Stretch or Shrink only| D. Direction becomes random

Why is SVD useful in machine learning?

A. Finds key directions | B. Removes all noise | C. Makes matrix invertible | D. Avoids multiplication

$C = X^T X$