Linear Algebra Primer

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab

Another, very in-depth linear algebra review from CS229 is available here:
http://cs229.stanford.edu/section/cs229-linalg.pdf
And a video discussion of linear algebra from EE263 is here (lectures 3 and 4):
https://see.stanford.edu/Course/EE263

Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors

Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors

Vectors and matrices are just collections of ordered numbers that represent something: movements in space, scaling factors, pixel brightness, etc. We'll define some common uses and standard operations on them.

Vector

- A column vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$ where

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- A row vector $\mathbf{v}^T \in \mathbb{R}^{1\times n}$ where

$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

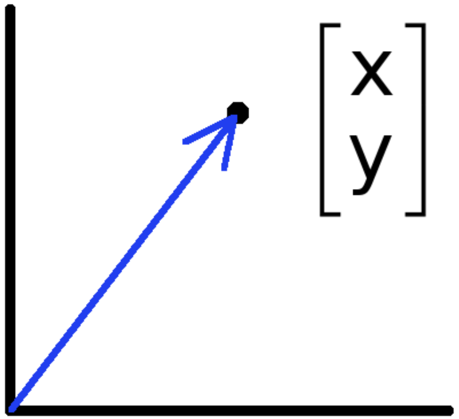
Vector

- We'll default to column vectors in this class

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

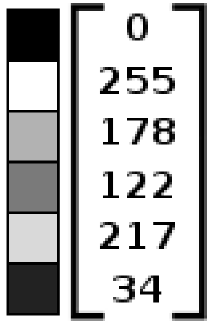
- You'll want to keep track of the orientation of your vectors when programming in python
- You can transpose a vector *V* in python by writing **V.t**. (But in class materials, we will **always** use V^T to indicate transpose, and we will use V' to mean "V prime")

Vectors have two main uses



- Vectors can represent an offset in 2D or 3D space.
- Points are just vectors from the origin.

- Data (pixels, gradients at an image keypoint, etc) can also be treated as a vector.



- Such vectors don't have a geometric interpretation, but calculations like "distance" can still have value.

Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an array of numbers with size by , i.e. m rows and n columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

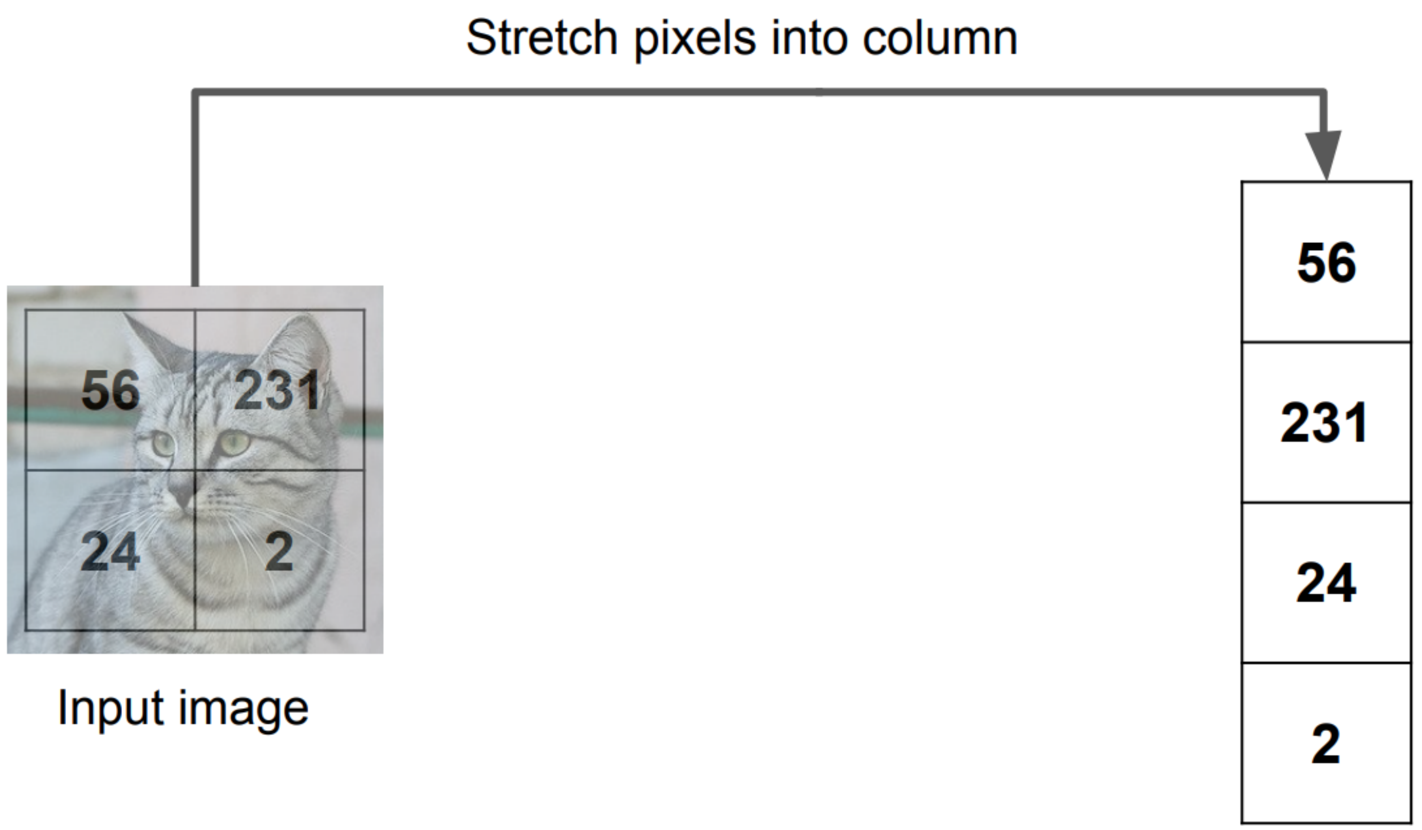
- If $m = n$, we say that $\mathbf{A}$ is square.

Images

$$= \begin{bmatrix} 193 & 180 & 210 & 112 & 125 \\ 189 & 8 & 177 & 97 & 114 \\ 100 & 71 & 81 & 195 & 165 \\ 167 & 12 & 242 & 203 & 181 \\ 44 & 25 & 9 & 48 & 192 \end{bmatrix}$$

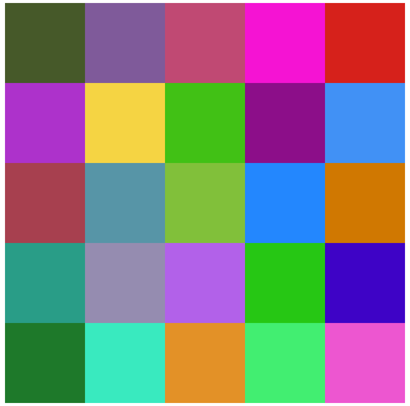
- Python represents an image as a matrix of pixel brightnesses
- Note that the upper left corner is [y,x] = (0,0)

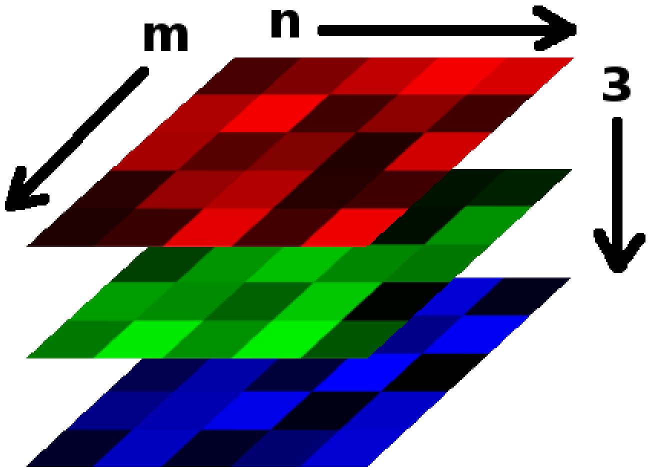
Images as both a **matrix** as well as a **vector**



Color Images

- Grayscale images have one number per pixel, and are stored as an m × n matrix.
- Color images have 3 numbers per pixel – red, green, and blue brightnesses (RGB)
- Stored as an m × n × 3 matrix





Basic Matrix Operations

- We will discuss:
  - Addition
  - Scaling
  - Dot product
  - Multiplication
  - Transpose
  - Inverse / pseudoinverse
  - Determinant / trace

Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

  - Can only add a matrix with matching dimensions, or a scalar.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

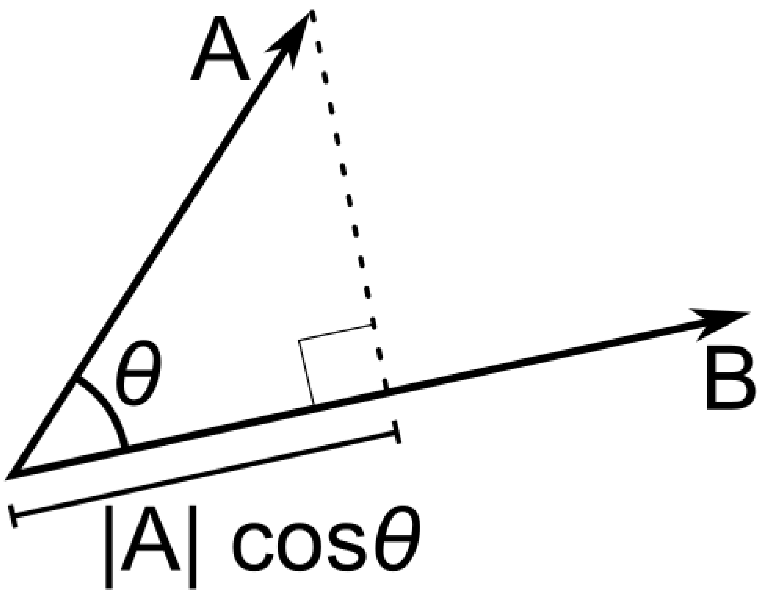
Matrix Operations

- Inner product (dot product) of vectors
  - Multiply corresponding entries of two vectors and add up the result
  - x·y is also |x||y|Cos(*the angle between x and y*)

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad \text{(scalar)}$$

Matrix Operations

- Inner product (dot product) of vectors
  - If B is a unit vector, then A·B gives the length of A which lies in the direction of B



Matrix Operations

- The product of two matrices

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

$$B \in \mathbb{R}^{n\times p}$$

$$C = AB \in \mathbb{R}^{m\times p}$$

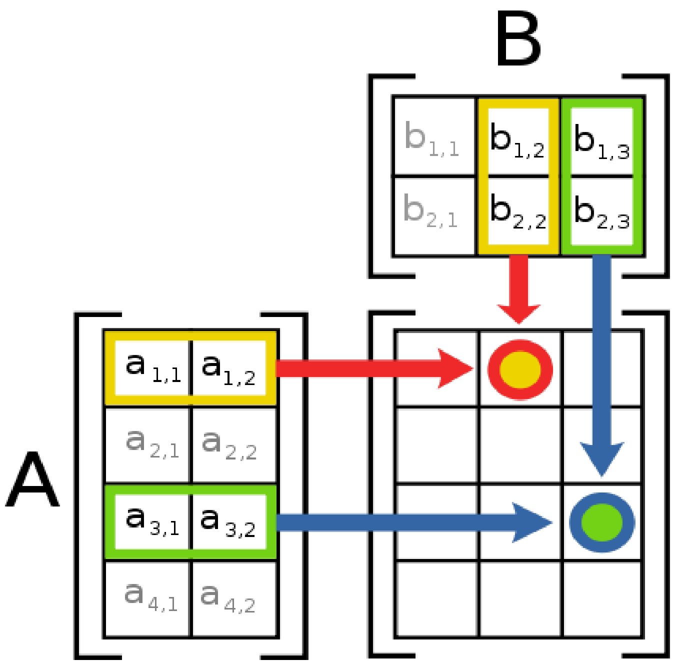
$$C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$$

$$C = AB = \begin{bmatrix} — & a_1^T & — \\ — & a_2^T & — \\ & \vdots & \\ — & a_m^T & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_p \\ | & | & & | \end{bmatrix} = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_p \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_p \end{bmatrix}.$$

Matrix Operations

- Multiplication

- The product AB is:



- Each entry in the result is (that row of A) dot product with (that column of B)
- Many uses, which will be covered later

Matrix Operations

- Multiplication example:

$$A \times B$$

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

$$\begin{bmatrix} 0 & 2 \\ 4 & 6 \end{bmatrix} \quad \begin{bmatrix} 10 & 14 \\ 34 & 54 \end{bmatrix}$$

$$0 \cdot 3 + 2 \cdot 7 = 14$$

– Each entry of the matrix product is made by taking the dot product of the corresponding row in the left matrix, with the corresponding column in the right one.

Matrix Operations

- The product of two matrices

Matrix multiplication is associative: $(AB)C = A(BC)$.

Matrix multiplication is distributive: $A(B + C) = AB + AC$.

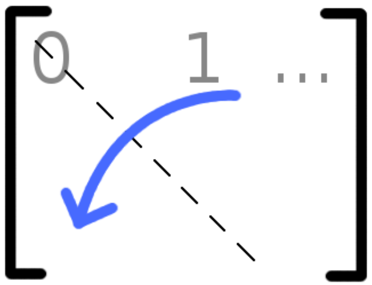
Matrix multiplication is, in general, *not* commutative; that is, it can be the case that $AB \neq BA$. (For example, if $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times q}$, the matrix product BA does not even exist if m and q are not equal!)

Matrix Operations

- Powers
  - By convention, we can refer to the matrix product AA as A^2, and AAA as A^3, etc.
  - Obviously only square matrices can be multiplied that way

Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1



$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

Matrix Operations

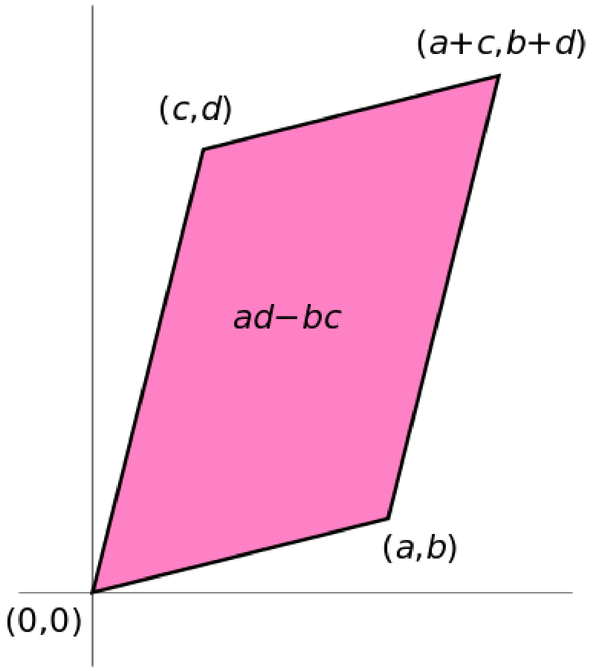
- **Determinant**
  - $\det(\mathbf{A})$ returns a scalar
  - Represents area (or volume) of the parallelogram described by the vectors in the rows of the matrix
  - For $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\det(\mathbf{A}) = ad - bc$
  - Properties:

$$\det(\mathbf{AB}) = \det(\mathbf{BA})$$

$$\det(\mathbf{A}^{-1}) = \frac{1}{\det(\mathbf{A})}$$

$$\det(\mathbf{A}^T) = \det(\mathbf{A})$$

$$\det(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A} \text{ is singular}$$



Matrix Operations

- **Trace**

$$\mathrm{tr}(\mathbf{A}) = \text{sum of diagonal elements}$$

$$\mathbf{tr}(\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}) = 1 + 7 = 8$$

  – Invariant to a lot of transformations, so it's used sometimes in proofs. (Rarely in this class though.)

  – Properties:

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$$
$$\mathrm{tr}(\mathbf{A} + \mathbf{B}) = \mathrm{tr}(\mathbf{A}) + \mathrm{tr}(\mathbf{B})$$

Vectors

- **Norm** $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}.$

- More formally, a norm is any function $f : \mathbb{R}^n \to \mathbb{R}$ that satisfies 4 properties:

- **Non-negativity:** For all $x \in \mathbb{R}^n,\ f(x) \geq 0$
- **Definiteness**: f(x) = 0 if and only if x = 0.
- **Homogeneity**: For all $x \in \mathbb{R}^n,\ t \in \mathbb{R},\ f(tx) = |t|\, f(x)$
- **Triangle inequality**: For all $x, y \in \mathbb{R}^n,\ f(x+y) \leq f(x) + f(y)$

Matrix Operations

- **Vector Norms**

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \|x\|_\infty = \max_i |x_i|.$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}. \qquad \|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- **Matrix norms:** Norms can also be defined for matrices, such as

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2} = \sqrt{\operatorname{tr}(A^T A)}.$$

Special Matrices

- Identity matrix $\mathbf{I}$
  - Square matrix, 1's along diagonal, 0's elsewhere
  - $\mathbf{I}$ · [another matrix] = [that matrix]

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

- Diagonal matrix
  - Square matrix with numbers along diagonal, 0's elsewhere
  - A diagonal · [another matrix] scales the rows of that matrix

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

Special Matrices

- Symmetric matrix

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

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

- Skew-symmetric matrix

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

$$\begin{bmatrix} 0 & -2 & -5 \\ 2 & 0 & -7 \\ 5 & 7 & 0 \end{bmatrix}$$

Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors

The inverse of a transformation matrix reverses its effect

Inverse

- Given a matrix **A**, its inverse $\mathbf{A}^{-1}$ is a matrix such that $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
- E.g. $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$
- Inverse does not always exist. If $\mathbf{A}^{-1}$ exists, **A** is *invertible* or *non-singular*. Otherwise, it's *singular*.
- Useful identities, for matrices that are invertible:

$$
\begin{aligned}
(\mathbf{A}^{-1})^{-1} &= \mathbf{A} \\
(\mathbf{A}\mathbf{B})^{-1} &= \mathbf{B}^{-1}\mathbf{A}^{-1} \\
\mathbf{A}^{-T} \triangleq (\mathbf{A}^{T})^{-1} &= (\mathbf{A}^{-1})^{T}
\end{aligned}
$$

Matrix Operations

- Pseudoinverse
  - Fortunately, there are workarounds to solve AX=B in these situations. And python can do them!
  - Instead of taking an inverse, directly ask python to solve for X in AX=B, by typing **np.linalg.solve(A, B)**
  - Python will try several appropriate numerical methods (including the pseudoinverse if the inverse doesn't exist)
  - Python will return the value of X which solves the equation
    - If there is no exact solution, it will return the closest one
    - If there are many solutions, it will return the smallest one

Matrix Operations

- Python example:

$$AX = B$$

$$A = \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

```
>> import numpy as np
>> x = np.linalg.solve(A,B)
x =
     1.0000
    -0.5000
```

Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors

The rank of a transformation matrix tells you how many dimensions it transforms a vector to.

Linear independence

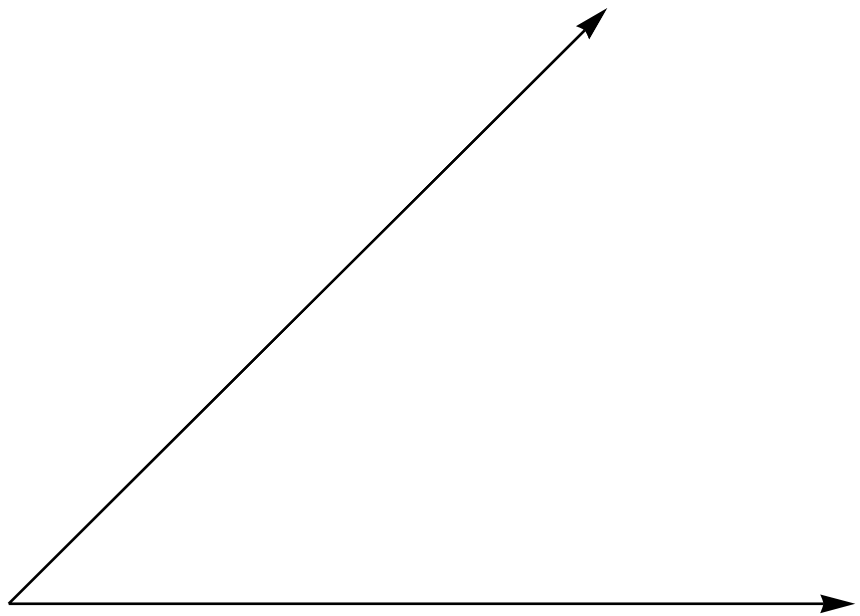
- Suppose we have a set of vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$
- If we can express $\mathbf{v}_1$ as a linear combination of the other vectors $\mathbf{v}_2 \ldots \mathbf{v}_n$, then $\mathbf{v}_1$ is linearly *dependent* on the other vectors.
  – The direction $\mathbf{v}_1$ can be expressed as a combination of the directions $\mathbf{v}_2 \ldots \mathbf{v}_n$. (E.g. $\mathbf{v}_1 = .7\,\mathbf{v}_2 - .7\,\mathbf{v}_4$)

Linear independence

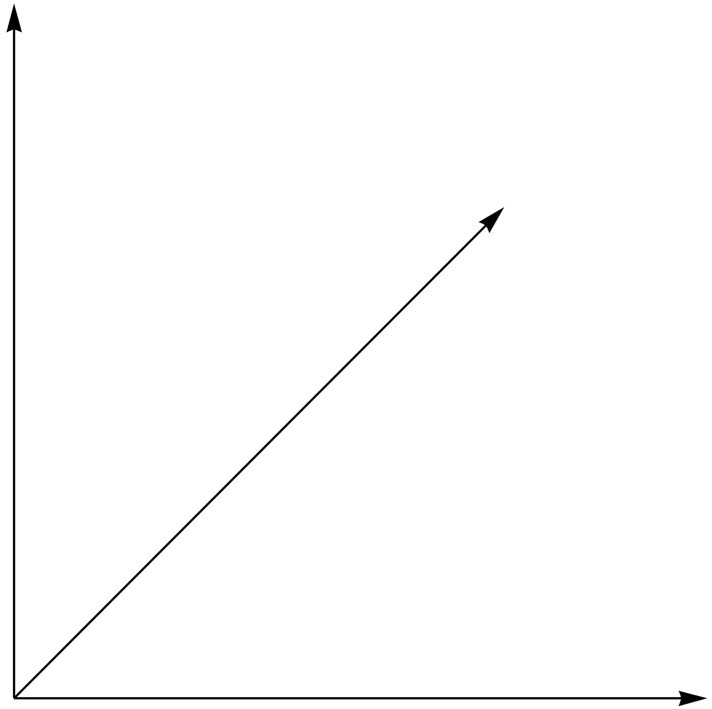
- Suppose we have a set of vectors $\mathbf{v_1, \ldots, v_n}$
- If we can express $\mathbf{v}_1$ as a linear combination of the other vectors $\mathbf{v}_2 \ldots \mathbf{v}_n$, then $\mathbf{v}_1$ is linearly *dependent* on the other vectors.
  - The direction $\mathbf{v}_1$ can be expressed as a combination of the directions $\mathbf{v}_2 \ldots \mathbf{v}_n$. (E.g. $\mathbf{v}_1 = .7\ \mathbf{v}_2 - .7\ \mathbf{v}_4$)
- If no vector is linearly dependent on the rest of the set, the set is linearly *independent*.
  - Common case: a set of vectors $\mathbf{v_1, \ldots, v_n}$ is always linearly independent if each vector is perpendicular to every other vector (and non-zero)

Linear independence

Linearly independent set



Not linearly independent



Matrix rank

- Column/row rank

$$\text{col-rank}(\mathbf{A}) = \text{ the maximum number of linearly independent column vectors of } \mathbf{A}$$
$$\text{row-rank}(\mathbf{A}) = \text{ the maximum number of linearly independent row vectors of } \mathbf{A}$$

  – Column rank always equals row rank

- Matrix rank

$$\text{rank}(\mathbf{A}) \triangleq \text{col-rank}(\mathbf{A}) = \text{row-rank}(\mathbf{A})$$

Matrix rank

- For transformation matrices, the rank tells you the dimensions of the output
- E.g. if rank of **A** is 1, then the transformation

p'=Ap

maps points onto a line.

- Here's a matrix with rank 1:

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+y \\ 2x+2y \end{bmatrix}$$

All points get mapped to the line y=2x

Matrix rank

- If an m x m matrix is rank m, we say it's "full rank"
  - Maps an m x 1 vector uniquely to another m x 1 vector
  - An inverse matrix can be found
- If rank < m, we say it's "singular"
  - At least one dimension is getting collapsed. No way to look at the result and tell what the input was
  - Inverse does not exist
- Inverse also doesn't exist for non-square matrices

Outline

- Vectors and matrices
  - Basic Matrix Operations
  - Determinants, norms, trace
  - Special Matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors(SVD)

Eigenvector and Eigenvalue

- An eigenvector **x** of a linear transformation *A* is a non-zero vector that, when *A* is applied to it, does not change direction.

$$Ax = \lambda x, \quad x \neq 0.$$

Eigenvector and Eigenvalue

- An eigenvector **x** of a linear transformation *A* is a non-zero vector that, when *A* is applied to it, does not change direction.
- Applying *A* to the eigenvector only scales the eigenvector by the scalar value λ, called an eigenvalue.

$$Ax = \lambda x, \quad x \neq 0.$$

Eigenvector and Eigenvalue

- We want to find all the eigenvalues of A:

$$Ax = \lambda x, \quad x \neq 0.$$

- Which can we written as:

$$Ax = (\lambda I)x \quad x \neq 0.$$

- Therefore:

$$(\lambda I - A)x = 0, \quad x \neq 0.$$

Eigenvector and Eigenvalue

- We can solve for eigenvalues by solving:

$$(\lambda I - A)x = 0, \quad x \neq 0.$$

- Since we are looking for non-zero **x**, we can instead solve the above equation as:

$$|(\lambda I - A)| = 0.$$

Properties

- The trace of a A is equal to the sum of its eigenvalues:

$$\mathrm{tr}A = \sum_{i=1}^{n} \lambda_i.$$

- The determinant of A is equal to the product of its eigenvalues

$$|A| = \prod_{i=1}^{n} \lambda_i.$$

- The rank of A is equal to the number of non-zero eigenvalues of A.
- The eigenvalues of a diagonal matrix D = diag(d1, . . . dn) are just the diagonal entries d1, . . . dn

Diagonalization

- An $n \times n$ matrix A is diagonalizable if it has n linearly independent eigenvectors.
- Most square matrices are diagonalizable:
  – Matrices with n distinct eigenvalues are diagonalizable

Lemma: Eigenvectors associated with distinct eigenvalues are linearly independent.

Diagonalization

- Eigenvalue equation:

$$AV = VD$$

$$A = VDV^{-1}$$

  - Where D is a diagonal matrix of the eigenvalues

$$\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

Diagonalization

- Eigenvalue equation:

$$AV = VD$$
$$A = VDV^{-1}$$

- Assuming all λ_i's are unique:

$$A = VDV^T$$

  - Remember that the inverse of an orthogonal matrix is just its transpose and the eigenvectors are orthogonal

Symmetric matrices

- Properties:
  - Symmetric matrices are always diagonalizable
  - For a symmetric matrix A, all the eigenvalues are real.
  - The eigenvectors of A are orthonormal.

$$A = VDV^T$$

Some applications of Eigenvalues

- PageRank
- Schrodinger's equation
- SVD
- We are going to use it to compress images in future classes

What we have learned

- Vectors and matrices
  - Basic Matrix Operations
  - Special Matrices
- Matrix inverse
- Matrix rank
- Eigenvalues and Eigenvectors