Computer Vision:
Maths review

Siheng Chen 陈思衡

Why we talk about maths?

Why we talk about maths?

Why we talk about maths?

Why we talk about maths?

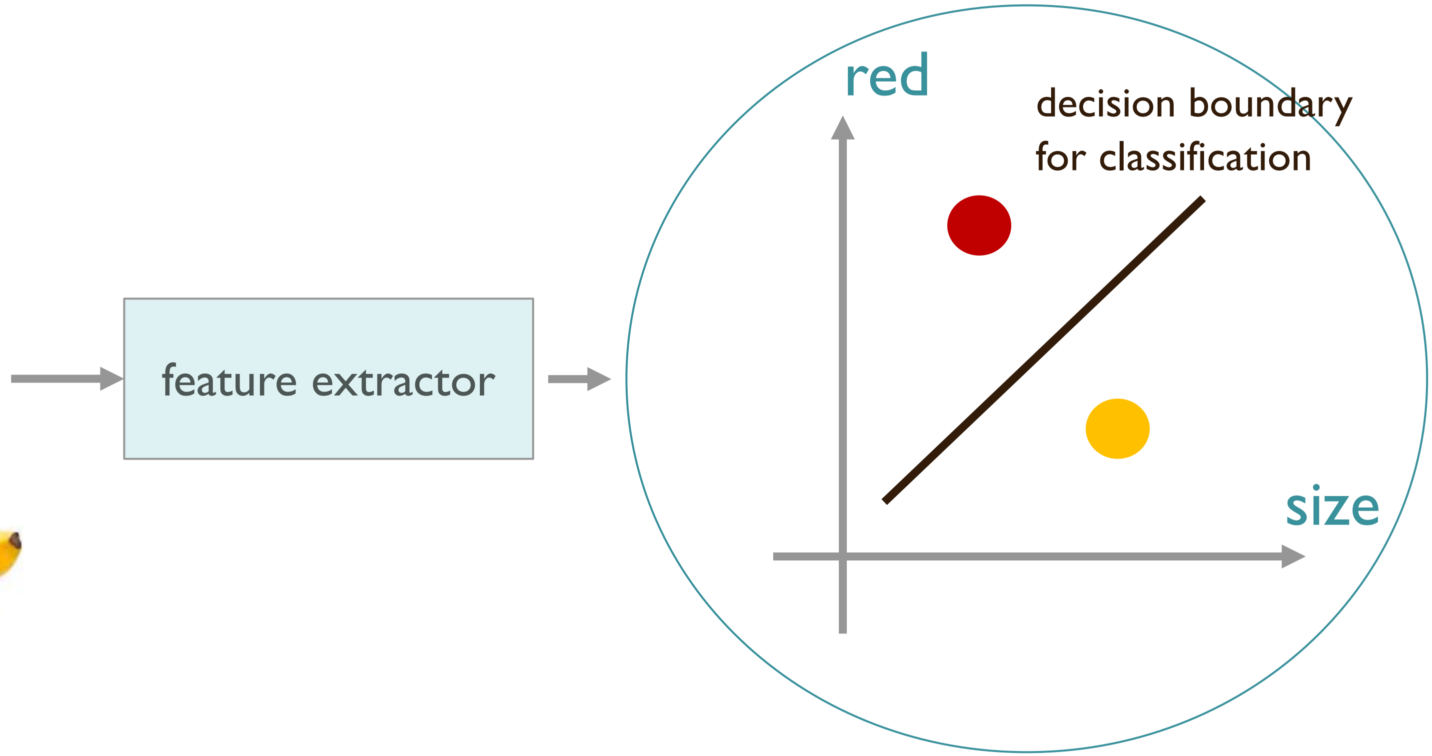


image space

feature space

vector space

Outline

Vector spaces

Derivatives

Vector spaces

DEFINITION 2.1 (VECTOR SPACE) A *vector space* over a field of scalars $\mathbb{C}$ (or $\mathbb{R}$) is a set of vectors, V, together with operations of vector addition and scalar multiplication. For any x, y, z in V and α, β in $\mathbb{C}$ (or $\mathbb{R}$), these operations must satisfy the following properties:

(i) *Commutativity:* $x + y = y + x$.

(ii) *Associativity:* $(x + y) + z = x + (y + z)$ and $(\alpha\beta)x = \alpha(\beta x)$.

(iii) *Distributivity:* $\alpha(x + y) = \alpha x + \alpha y$ and $(\alpha + \beta)x = \alpha x + \beta x$.

Furthermore, the following hold:

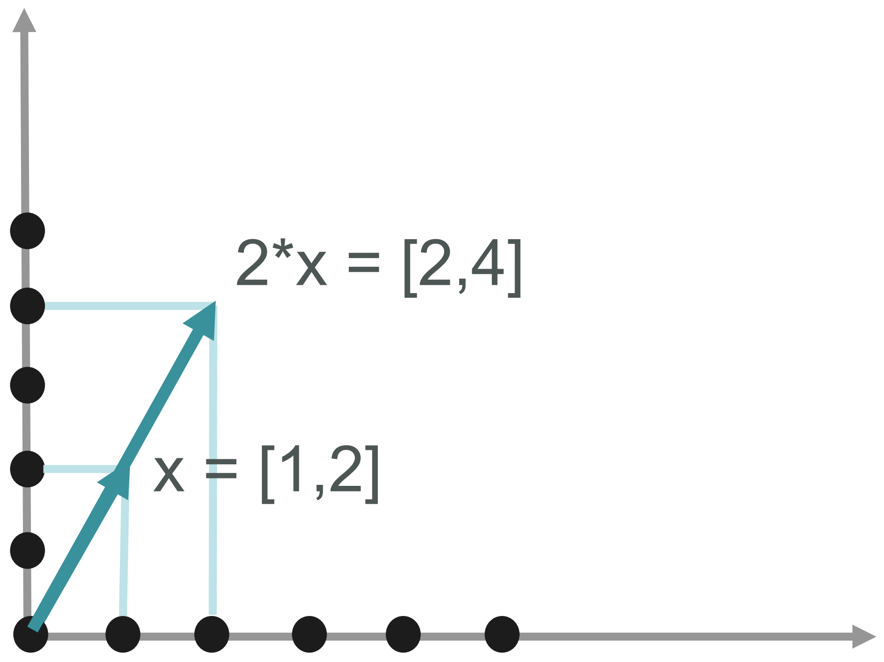
(iv) *Additive identity:* There exists an element $\mathbf{0}$ in V such that $x + \mathbf{0} = \mathbf{0} + x = x$ for every x in V.

(v) *Additive inverse:* For each x in V, there exists a unique element $-x$ in V such that $x + (-x) = (-x) + x = \mathbf{0}$.

(vi) *Multiplicative identity:* For every x in V, $1 \cdot x = x$.

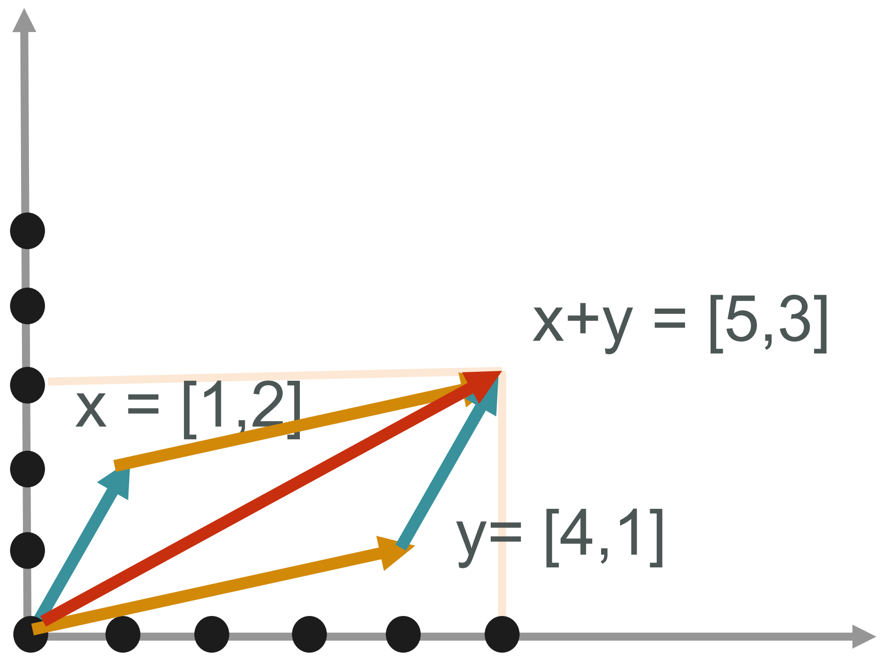
Vector spaces

Scale (vector, scalar -> vector)



Vector spaces

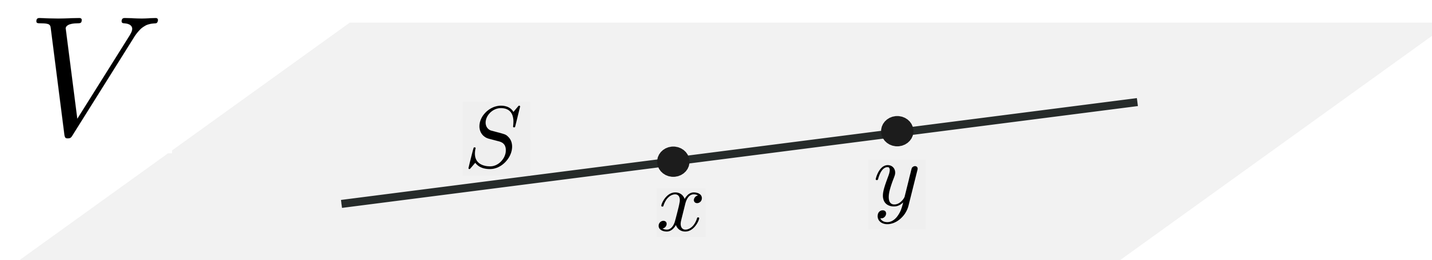
Add (vector, vector -> vector)



Vector spaces

Definition 2.2 (Subspace) A nonempty subset S of a vector space V is a *subspace* when it is closed under the operations of vector addition and scalar multiplication:

(i) For all x and y in S, $x + y$ is in S.
(ii) For all x in S and α in $\mathbb{C}$ (or $\mathbb{R}$), αx is in S.



Vector spaces

DEFINITION 2.7 (INNER PRODUCT) An *inner product* on a vector space V over $\mathbb{C}$ (or $\mathbb{R}$) is a complex-valued (or real-valued) function $\langle \cdot, \cdot \rangle$ defined on $V \times V$, with the following properties for any $x, y, z \in V$ and $\alpha \in \mathbb{C}$ (or $\mathbb{R}$):

(i) *Distributivity:* $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$.

(ii) *Linearity in the first argument:* $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$.

(iii) *Hermitian symmetry:* $\langle x, y \rangle^* = \langle y, x \rangle$.

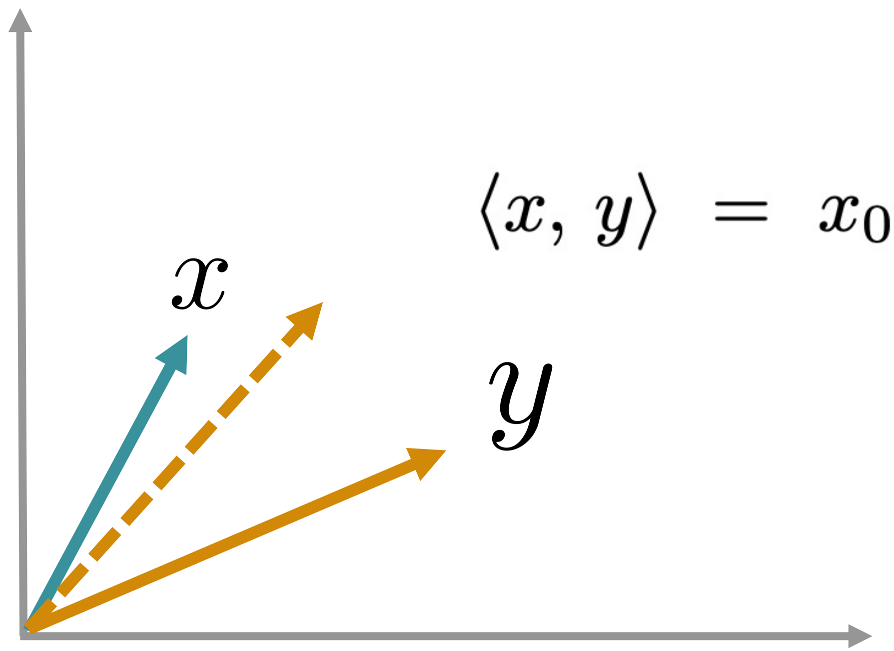
(iv) *Positive definiteness:* $\langle x, x \rangle \geq 0$, and $\langle x, x \rangle = 0$ if and only if $x = \mathbf{0}$.

"Similarity" $$\langle x, y \rangle = \sum_{n=0}^{N-1} x_n y_n^* = y^* x$$

Vector spaces

Inner product (vector, vector -> scalar)

$$\langle x,\ y\rangle \ = \ x_0 y_0 + x_1 y_1$$



Vector spaces

Definition 2.8 (Orthogonality)

(i) Vectors x and y are said to be *orthogonal* when $\langle x, y \rangle = 0$, written as $x \perp y$.

(ii) A set of vectors S is called *orthogonal* when $x \perp y$ for every x and y in S such that $x \neq y$.

(iii) A set of vectors S is called *orthonormal* when it is orthogonal and $\langle x, x \rangle = 1$ for every x in S.

(iv) A vector x is said to be *orthogonal* to a set of vectors S when $x \perp s$ for all $s \in S$, written as $x \perp S$.

(v) Two sets S_0 and S_1 are said to be *orthogonal* when every vector s_0 in S_0 is orthogonal to the set S_1, written as $S_0 \perp S_1$.

(vi) Given a subspace S of a vector space V, the *orthogonal complement* of S, denoted $S^\perp$, is the set $\{x \in V \mid x \perp S\}$.

Vector spaces

DEFINITION 2.9 (NORM) A *norm* on a vector space V over $\mathbb{C}$ (or $\mathbb{R}$) is a real-valued function $\|\cdot\|$ defined on V, with the following properties for any $x, y \in V$ and $\alpha \in \mathbb{C}$ (or $\mathbb{R}$):

(i) *Positive definiteness:* $\|x\| \geq 0$, and $\|x\| = 0$ if and only if $x = \mathbf{0}$.
(ii) *Positive scalability:* $\|\alpha x\| = |\alpha| \, \|x\|$.
(iii) *Triangle inequality:* $\|x + y\| \leq \|x\| + \|y\|$, with equality if and only if $y = \alpha x$.

"length"

standard norm

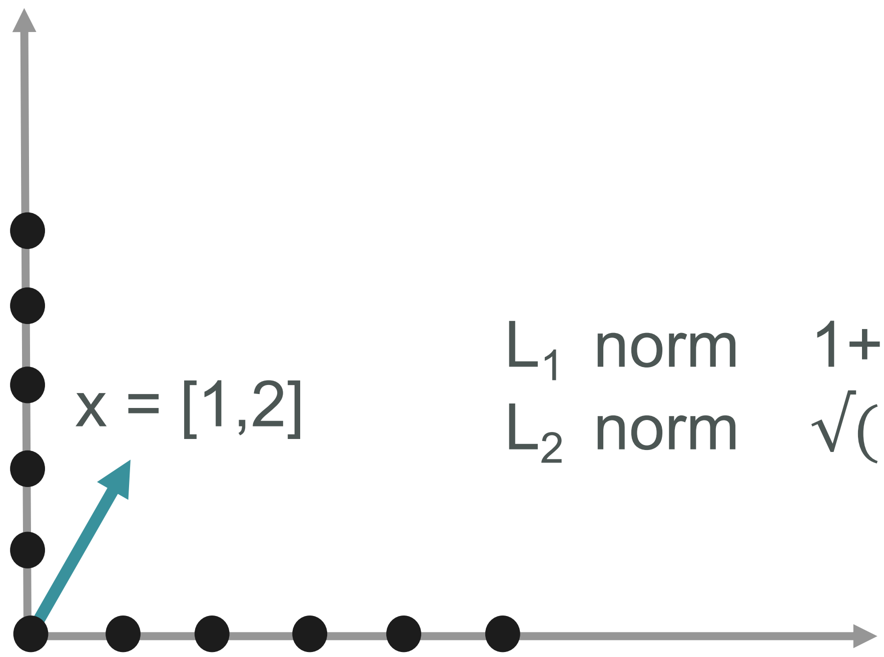
$$\|x\| = \sqrt{\langle x, x\rangle} = \left(\sum_{n=0}^{N-1} |x_n|^2\right)^{1/2}$$

p-norm

$$\|x\|_p = \left(\sum_{n=0}^{N-1} |x_n|^p\right)^{1/p}$$

Vector spaces

Norm (vector -> scalar)



L_1 norm $1+2 = 3$

L_2 norm $\sqrt{(1^2 + 2^2)}$

Vector spaces

Definition 2.10 (Metric, or distance) In a normed vector space, the *metric*, or *distance*, between vectors x and y is the norm of their difference:

$$d(x, y) = \|x - y\|.$$

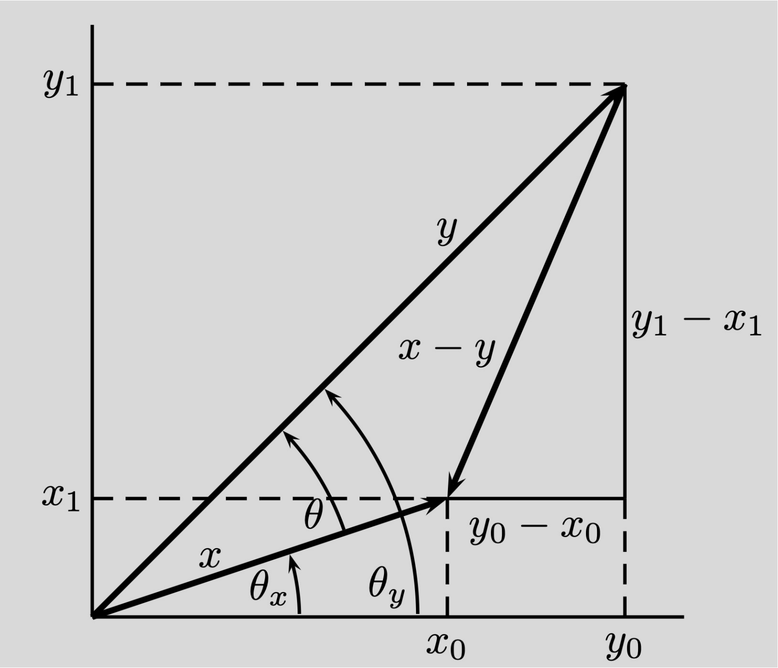
"length of difference"

Vector spaces: 2D example

Inner product

$$x = \begin{bmatrix} x_0 & x_1 \end{bmatrix}^\top \quad y = \begin{bmatrix} y_0 & y_1 \end{bmatrix}^\top$$

$$\begin{aligned}
\langle x,\, y\rangle &= x_0 y_0 + x_1 y_1 \\
&= (\|x\| \cos\theta_x)(\|y\| \cos\theta_y) + (\|x\| \sin\theta_x)(\|y\| \sin\theta_y) \\
&= \|x\|\,\|y\|(\cos\theta_x \cos\theta_y + \sin\theta_x \sin\theta_y) \\
&= \|x\|\,\|y\| \cos(\theta_x - \theta_y).
\end{aligned}$$



Norm

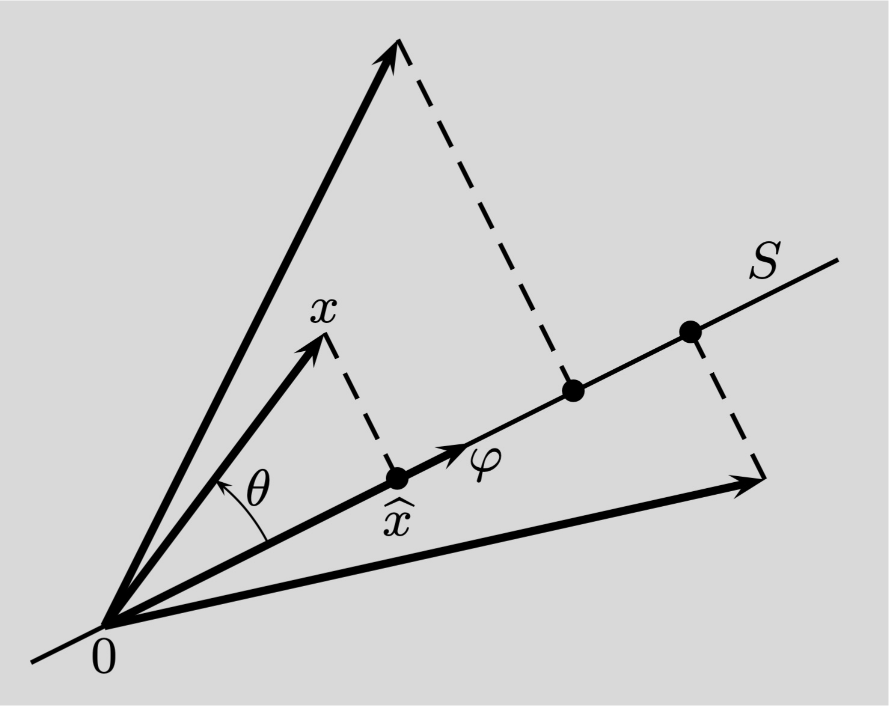
$$\|x\| = \sqrt{\langle x,\, x\rangle} = \sqrt{x_0^2 + x_1^2}.$$

Distance

$$d(x, y) = \|x - y\| = \sqrt{\langle x - y,\, x - y\rangle}$$

Vector spaces

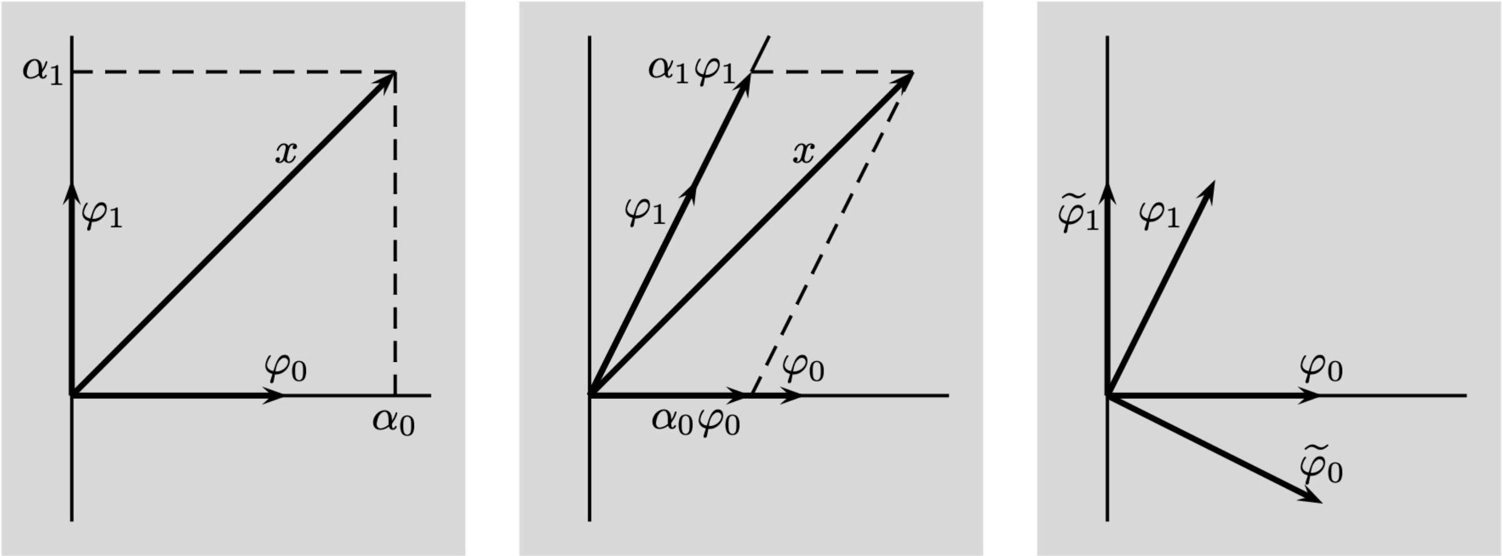
Projection



$$\widehat{x} \overset{(a)}{=} (\|x\| \cos\theta)\frac{\varphi}{\|\varphi\|} = (\|x\|\,\|\varphi\|\cos\theta)\frac{\varphi}{\|\varphi\|^2} \overset{(b)}{=} \frac{1}{\|\varphi\|^2}\langle x,\, \varphi\rangle\varphi$$

Vector spaces

Projection



(a) Expansion with an orthonormal basis.

(b) Expansion with a nonorthogonal basis.

(c) Basis $\{\varphi_0, \varphi_1\}$ and its dual $\{\widetilde{\varphi}_0, \widetilde{\varphi}_1\}$.

Vector spaces

Projection

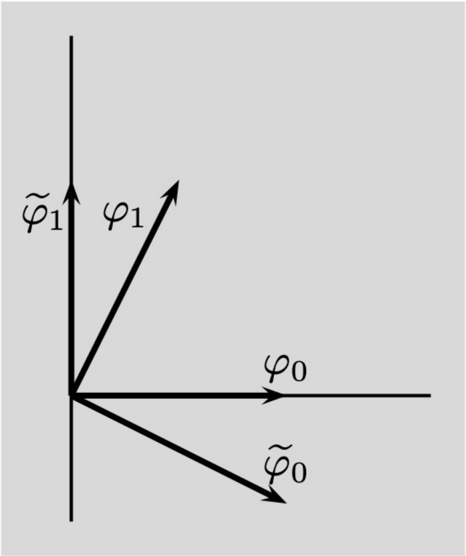
analysis

$$\alpha = \begin{bmatrix}\alpha_0 \\ \alpha_1\end{bmatrix} = \begin{bmatrix}\langle x, \widetilde{\varphi}_0\rangle \\ \langle x, \widetilde{\varphi}_1\rangle\end{bmatrix} = \underbrace{\begin{bmatrix}\widetilde{\varphi}_{00} & \widetilde{\varphi}_{01} \\ \widetilde{\varphi}_{10} & \widetilde{\varphi}_{11}\end{bmatrix}}_{\widetilde{\Phi}^\top} \begin{bmatrix}x_0 \\ x_1\end{bmatrix} = \widetilde{\Phi}^\top x.$$

synthesis

$$x = \begin{bmatrix}x_0 \\ x_1\end{bmatrix} = \alpha_0 \begin{bmatrix}\varphi_{00} \\ \varphi_{01}\end{bmatrix} + \alpha_1 \begin{bmatrix}\varphi_{10} \\ \varphi_{11}\end{bmatrix} = \underbrace{\begin{bmatrix}\varphi_{00} & \varphi_{10} \\ \varphi_{01} & \varphi_{11}\end{bmatrix}}_{\Phi} \begin{bmatrix}\alpha_0 \\ \alpha_1\end{bmatrix} = \begin{bmatrix}\varphi_0 & \varphi_1\end{bmatrix} \begin{bmatrix}\alpha_0 \\ \alpha_1\end{bmatrix}$$

$$= \Phi\alpha = \Phi\widetilde{\Phi}^\top x,$$



(c) Basis $\{\varphi_0, \varphi_1\}$ and its dual $\{\widetilde{\varphi}_0, \widetilde{\varphi}_1\}$.

Vector spaces

DEFINITION 2.34 (BASIS) The set of vectors $\Phi = \{\varphi_k\}_{k\in\mathcal{K}} \subset V$, where $\mathcal{K}$ is finite or countably infinite, is called a *basis* for a normed vector space V when

(i) it is *complete* in V, meaning that, for any $x \in V$, there is a sequence $\alpha \in \mathbb{C}^{\mathcal{K}}$ such that

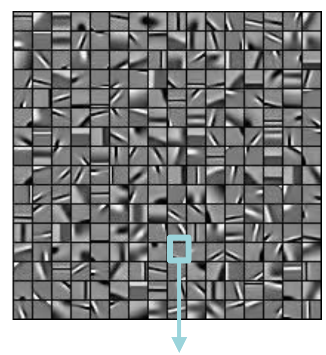
$$x = \sum_{k\in\mathcal{K}} \alpha_k \varphi_k; \tag{2.87}$$

and

(ii) for any $x \in V$, the sequence α that satisfies (2.87) is unique.

x

synthesis



$\varphi_k \quad \alpha_k$

Vector spaces

THEOREM 2.39 (ORTHONORMAL BASIS EXPANSIONS) Let $\Phi = \{\varphi_k\}_{k\in\mathcal{K}}$ be an orthonormal basis for a Hilbert space H. The unique expansion with respect to Φ of any x in H has expansion coefficients

analysis

$$\alpha_k = \langle x, \varphi_k \rangle \quad \text{for } k \in \mathcal{K}, \quad \text{or}, \tag{2.93a}$$

$$\alpha = \Phi^* x. \tag{2.93b}$$

Synthesis with these coefficients yields

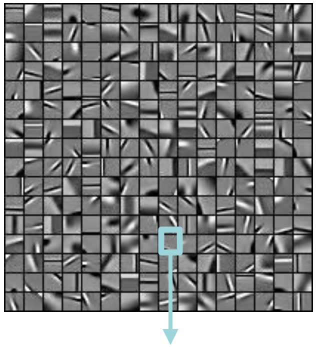
synthesis

$$x = \sum_{k\in\mathcal{K}} \langle x, \varphi_k \rangle \varphi_k \tag{2.94a}$$

$$= \Phi\alpha = \Phi\Phi^* x. \tag{2.94b}$$

analysis

synthesis



x

$\varphi_k \quad \alpha_k$

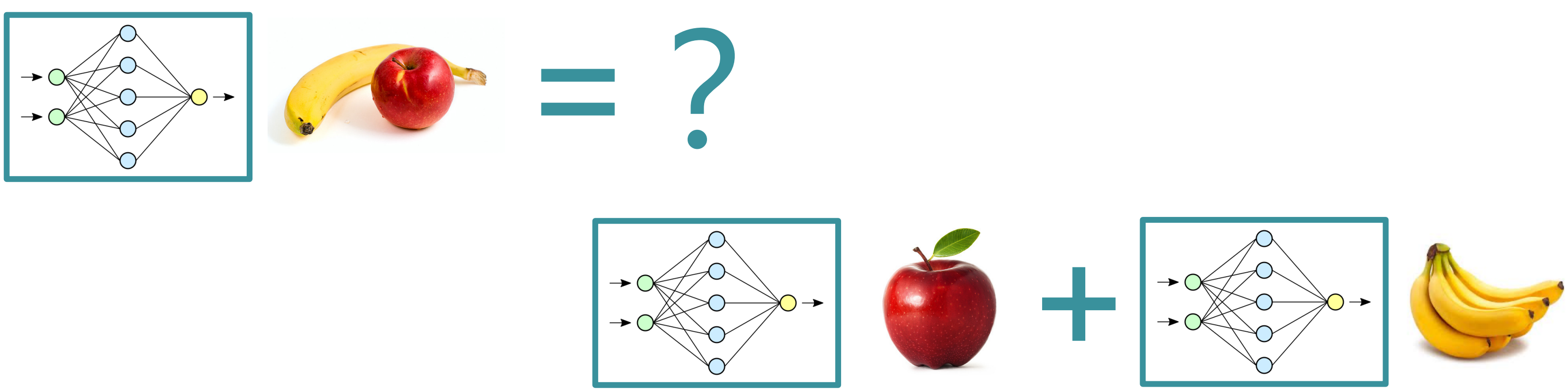
Vector spaces

Definition 2.17 (Linear operator) A function $A : H_0 \to H_1$ is called a *linear operator* from H_0 to H_1 when, for all x, y in H_0 and α in $\mathbb{C}$ (or $\mathbb{R}$), the following hold:

(i) *Additivity:* $A(x+y) = Ax + Ay$.

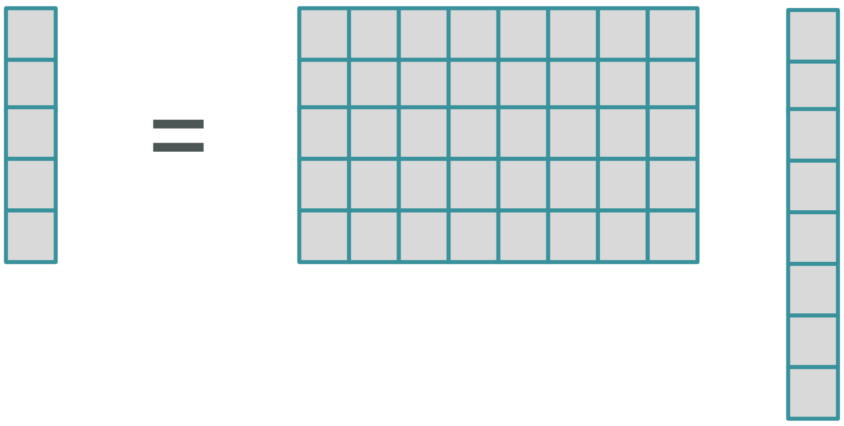
(ii) *Scalability:* $A(\alpha x) = \alpha(Ax)$.

When the domain H_0 and the codomain H_1 are the same, A is also called a linear operator on H_0.



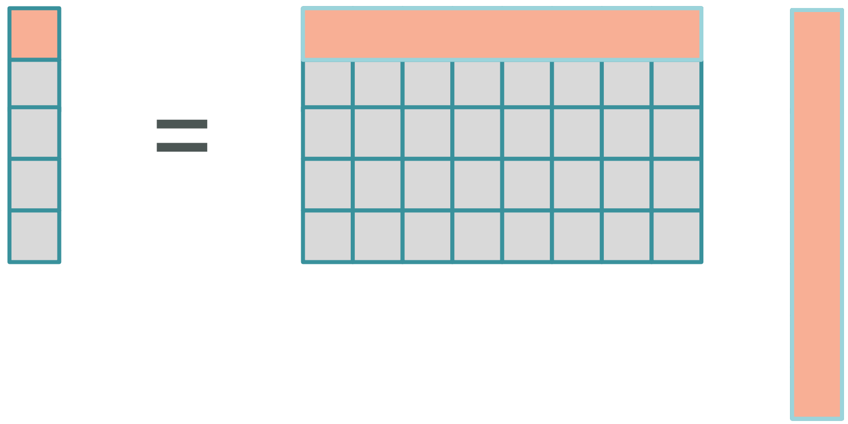
Vector spaces

Matrix-vector product



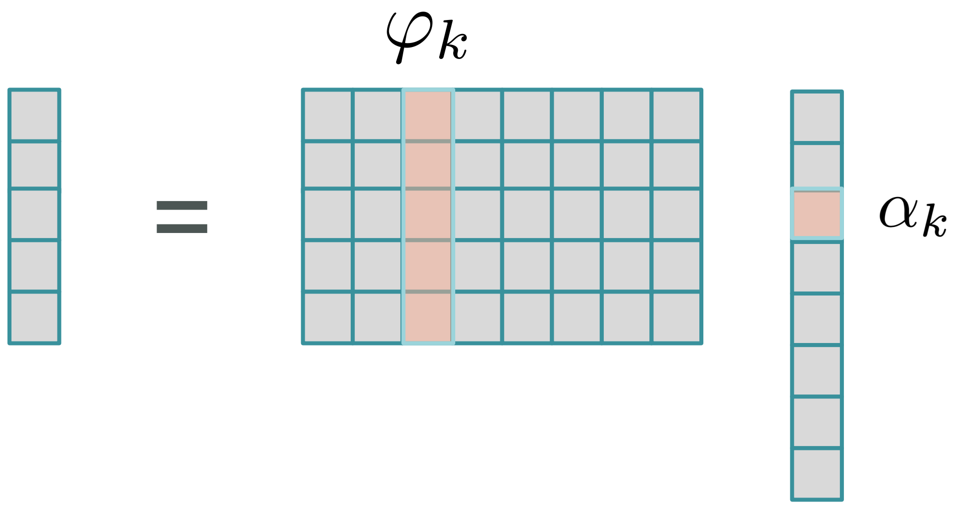
Vector spaces

Matrix-vector product



Vector spaces

Matrix-vector product



$$x = \sum_{i \in \mathcal{K}} \alpha_i \varphi_i$$

Vector spaces

Subspaces about a matrix

- Column space

  $$\mathcal{R}(A) \;=\; \mathrm{span}(\{a_0,\, a_1,\, \ldots,\, a_{N-1}\}) \;=\; \{y \in \mathbb{R}^M \mid y = Ax \text{ for some } x \in \mathbb{R}^N\}$$

- Row space

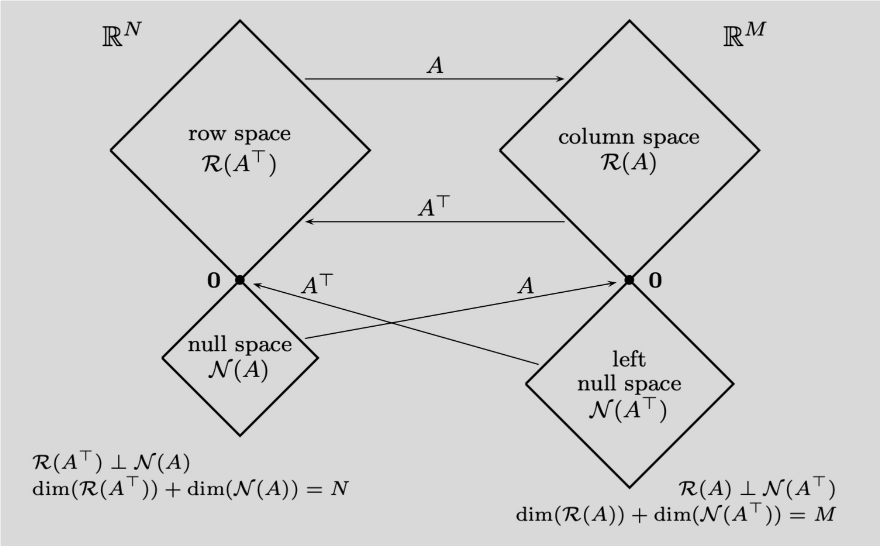
  $$\mathcal{R}(A^\top) \;=\; \mathrm{span}(\{b_0^\top,\, b_1^\top,\, \ldots,\, b_{M-1}^\top\}) \;=\; \{x \in \mathbb{R}^N \mid x = A^\top y \text{ for some } y \in \mathbb{R}^M\}$$

- Null space

  $$\mathcal{N}(A) \;=\; \{x \in \mathbb{R}^N \mid Ax = \mathbf{0}\}$$

- Left null space

  $$\mathcal{N}(A^\top) \;=\; \{y \in \mathbb{R}^M \mid A^\top y = \mathbf{0}\}$$



Vector spaces

Subspaces about a matrix

Space	**Symbol**	**Definition**	**Dimension**
Column space (range)	$\mathcal{R}(A)$	$\{y \in \mathbb{C}^M \mid y = Ax \text{ for some } x \in \mathbb{C}^N\}$	$\operatorname{rank} A$
Left null space	$\mathcal{N}(A^*)$	$\{y \in \mathbb{C}^M \mid A^*y = 0\}$	$M - \operatorname{rank} A$
	$\dim(\mathcal{R}(A)) + \dim(\mathcal{N}(A^*)) = M$		
Row space	$\mathcal{R}(A^*)$	$\{x \in \mathbb{C}^N \mid x = A^*y \text{ for some } y \in \mathbb{C}^M\}$	$\operatorname{rank} A$
Null space (kernel)	$\mathcal{N}(A)$	$\{x \in \mathbb{C}^N \mid Ax = 0\}$	$N - \operatorname{rank} A$
	$\dim(\mathcal{R}(A^*)) + \dim(\mathcal{N}(A)) = N$		

Vector spaces

DEFINITION 2.18 (OPERATOR NORM AND BOUNDED LINEAR OPERATOR) The *operator norm* of A, denoted by $\|A\|$, is defined as

$$\|A\| = \sup_{\|x\|=1} \|Ax\|. \tag{2.45}$$

A linear operator is called *bounded* when its operator norm is finite.

"strength of operation"

Vector spaces

Matrix norm

- Frobenius norm $$\|A\|_{\mathrm{F}} = \sqrt{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|A_{i,j}|^2} = \sqrt{\mathrm{tr}(AA^*)}$$
- Operator norm $$\|A\|_{p,q} = \sup_{\|x\|_p=1} \|Ax\|_q$$
- Some examples

$$
\begin{aligned}
\|A\|_1 &= \|A\|_{1,1} = \max_{0\le j\le N-1} \textstyle\sum_{i=0}^{M-1}|A_{i,j}|,\\
\|A\|_{1,2} &= \max_{0\le j\le N-1} \left(\textstyle\sum_{i=0}^{M-1}|A_{i,j}|^2\right)^{1/2},\\
\|A\|_{1,\infty} &= \max_{0\le i\le M-1,\, 0\le j\le N-1} |A_{i,j}|,\\
\|A\|_2 &= \|A\|_{2,2} = \sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^*A)},\\
\|A\|_{2,\infty} &= \max_{0\le i\le M-1} \left(\textstyle\sum_{j=0}^{N-1}|A_{i,j}|^2\right)^{1/2},\\
\|A\|_\infty &= \|A\|_{\infty,\infty} = \max_{0\le i\le M-1} \textstyle\sum_{j=0}^{N-1}|A_{i,j}|.
\end{aligned}
$$

Vector spaces

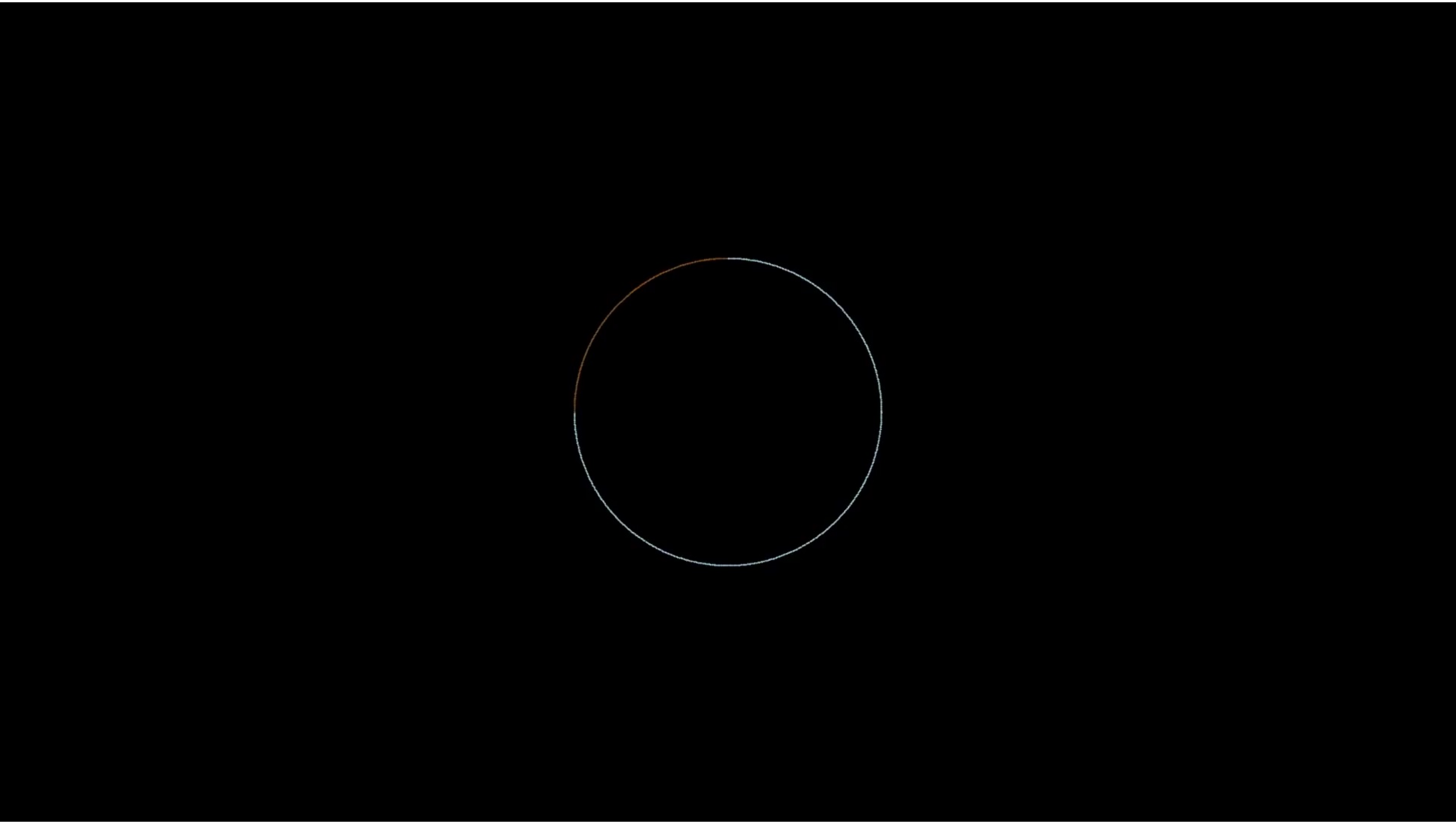
Eigen-decomposition

$$Av = \lambda v$$

$$A = V \Lambda V^{-1}$$

$$Ax = A \left(\sum_{k=0}^{N-1} \alpha_k v_k \right) \overset{(a)}{=} \sum_{k=0}^{N-1} \alpha_k (A v_k) \overset{(b)}{=} \sum_{k=0}^{N-1} (\alpha_k \lambda_k) v_k$$

Vector spaces



https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

Vector spaces

Singular value decomposition

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

$$AA^* = (U\Sigma V^*)(V\Sigma^* U^*) = U\Sigma^2 U^*$$
$$A^*A = (V\Sigma^* U^*)(U\Sigma V^*) = V\Sigma^2 V^*$$

$$\sigma^2(A) = \lambda(AA^*) = \lambda(A^*A)$$

Vector spaces: Implementation

Sum across axes

Let **A** be a matrix of shape (N, 2):

```
A = np.random.randn(N, 2)
```

$$\boldsymbol{A} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

Vector spaces: Implementation

Broadcasting

You Probably Saw Matrix Addition

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

What is this? FYI: e is a scalar

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

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Vector spaces: Implementation

Broadcasting

Given: Matrix **P** of shape (N, 2) vector **v** of shape (2, 1)
Want: Difference matrix **D** of shape (N, 2)

$$\boldsymbol{P} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} a \\ b \end{bmatrix} \quad \boldsymbol{D} = \begin{bmatrix} x_1 - a & y_1 - b \\ \vdots & \vdots \\ x_N - a & y_N - b \end{bmatrix}$$

Vector spaces: Implementation

Broadcasting

```
x = np.ones(10, 20)
y = np.ones(20)
z = x + y
print(z.shape)
(10,20)
```

```
x = np.ones(10, 20)
y = np.ones(10)
z = x + y
print(z.shape)
ERROR
```

```
x = np.ones(10, 20)
y = np.ones(10, 1)
z = x + y
print(z.shape)
(10,20)
```

```
x = np.ones(1, 20)
y = np.ones(10, 1)
z = x + y
print(z.shape)
(10,20)
```

Vector spaces: Implementation

Broadcasting

The same broadcasting rules apply to tensors with any number of dimensions!

```
x = np.ones(30)
y = np.ones(20, 1)
z = np.ones(10, 1, 1)
w = x + y + z
print(w.shape)

(10, 20, 30)
```

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Vector spaces: Implementation

Vectorization example

- Suppose I have two sets of (D-dimensional) vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$ and I want to compute all pairwise distances $d_{i,j} = ||x_i - y_j||$
- Identity: $\|\boldsymbol{x} - \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^T\boldsymbol{y}$
- Or: $\|\boldsymbol{x} - \boldsymbol{y}\| = (\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^T\boldsymbol{y})^{1/2}$

Vector spaces: Implementation

Vectorization example

$$\boldsymbol{X} = \begin{bmatrix} - & \boldsymbol{x}_1 & - \\ & \vdots & \\ - & \boldsymbol{x}_N & - \end{bmatrix} \quad \boldsymbol{Y} = \begin{bmatrix} - & \boldsymbol{y}_1 & - \\ & \vdots & \\ - & \boldsymbol{y}_M & - \end{bmatrix} \quad \boldsymbol{Y}^T = \begin{bmatrix} | & & | \\ \boldsymbol{y}_1 & \cdots & \boldsymbol{y}_M \\ | & & | \end{bmatrix}$$

N×D M×D

Compute a Nx1 vector of norms (can also do Mx1)

$$\boldsymbol{\Sigma}(\boldsymbol{X}^2, \boldsymbol{1}) = \begin{bmatrix} \|\boldsymbol{x}_1\|^2 \\ \vdots \\ \|\boldsymbol{x}_N\|^2 \end{bmatrix}$$

Compute a NxM matrix of dot products

$$\left(\boldsymbol{X}\boldsymbol{Y}^T\right)_{ij} = \boldsymbol{x}_i^T \boldsymbol{y}_j$$

Vector spaces: Implementation

Vectorization example

$$\mathbf{D} = \left(\Sigma(\boldsymbol{X}^{\mathbf{2}}, 1) + \Sigma(\boldsymbol{Y}^{\mathbf{2}}, 1)^{\boldsymbol{T}} - 2\boldsymbol{X}\boldsymbol{Y}^{\boldsymbol{T}}\right)^{1/2}$$

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Vector spaces: Implementation

Vectorization example

$$\mathbf{D} = \left(\Sigma(\boldsymbol{X}^2, 1) + \Sigma(\boldsymbol{Y}^2, 1)^T - 2\boldsymbol{X}\boldsymbol{Y}^T\right)^{1/2}$$

$$\mathbf{D}_{ij} = \|\boldsymbol{x}_i\|^2 + \|\boldsymbol{y}_j\|^2 + 2\boldsymbol{x}^T\boldsymbol{y}$$

Numpy code:

```
XNorm = np.sum(X**2,axis=1,keepdims=True)
YNorm = np.sum(Y**2,axis=1,keepdims=True)
D = (XNorm+YNorm.T-2*np.dot(X,Y.T))**0.5
```

N×1

N×1

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Vector spaces: Implementation

Vectorization example

$$\mathbf{D} = \left(\Sigma(\boldsymbol{X}^2, 1) + \Sigma(\boldsymbol{Y}^2, 1)^{T} - 2\boldsymbol{X}\boldsymbol{Y}^{T}\right)^{1/2}$$

$$\mathbf{D}_{ij} = \|\boldsymbol{x}_i\|^2 + \|\boldsymbol{y}_j\|^2 + 2\boldsymbol{x}^T\boldsymbol{y}$$

Numpy code:

```
XNorm = np.sum(X**2,axis=1,keepdims=True)
YNorm = np.sum(Y**2,axis=1,keepdims=True)   M×1
D = (XNorm+YNorm.T-2*np.dot(X,Y.T))**0.5
       N×1      M×1
```

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Vector spaces: Implementation

Vectorization example

$$\mathbf{D} = \left(\Sigma(\boldsymbol{X}^2, 1) + \Sigma(\boldsymbol{Y}^2, 1)^{\boldsymbol{T}} - 2\boldsymbol{X}\boldsymbol{Y}^{\boldsymbol{T}}\right)^{1/2}$$

$$\mathbf{D}_{ij} = \|\boldsymbol{x}_i\|^2 + \left\|\boldsymbol{y}_j\right\|^2 + 2\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y}$$

Numpy code:

```
XNorm = np.sum(X**2,axis=1,keepdims=True)
YNorm = np.sum(Y**2,axis=1,keepdims=True)
D = (XNorm+YNorm.T-2*np.dot(X,Y.T))**0.5
```

N×M

N×1 1×M N×M

Vector spaces: Implementation

Vectorization example

Computing pairwise distances between 300 and 400 128-dimensional vectors

1. for x in X, for y in Y, using native python: 9s
2. for x in X, for y in Y, using numpy to compute distance: 0.8s
3. vectorized: 0.0045s (~2000x faster than 1, 175x faster than 2)

Expressing things in primitives that are optimized is usually faster

Even more important with special hardware like GPUs or TPUs!

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Outline

Vector spaces

Derivatives

Derivatives

Rate at which a function f(x) changes at a point as well as the **direction** that increases the function

Given quadratic function f(x)

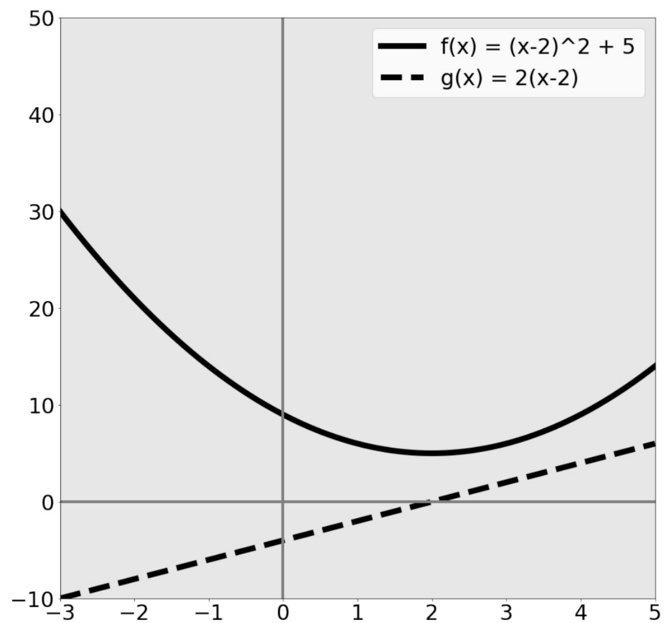
$$f(x, y) = (x - 2)^2 + 5$$

$f(x)$ is function

$$g(x) = f'(x)$$

aka

$$g(x) = \frac{d}{dx} f(x)$$



Derivatives

Rate at which a function f(x) changes at a point as well as the **direction** that increases the function

Given quadratic function f(x)

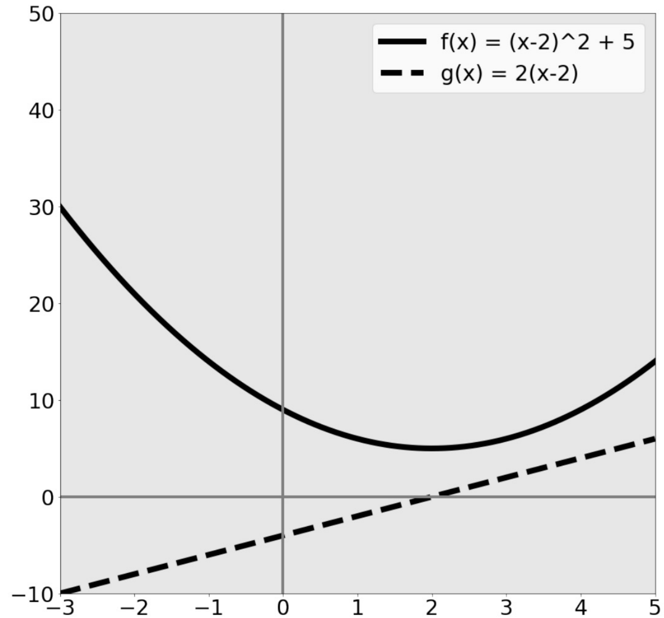
$$f(x, y) = (x-2)^2 + 5$$

What's special about x=2?

$f(x)$ minim. at 2
$g(x) = 0$ at 2

a = minimum of f → $g(a) = 0$

Reverse is ***not true***



https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Derivatives

Rate at which a function f(x) changes at a point as well as the **direction** that increases the function

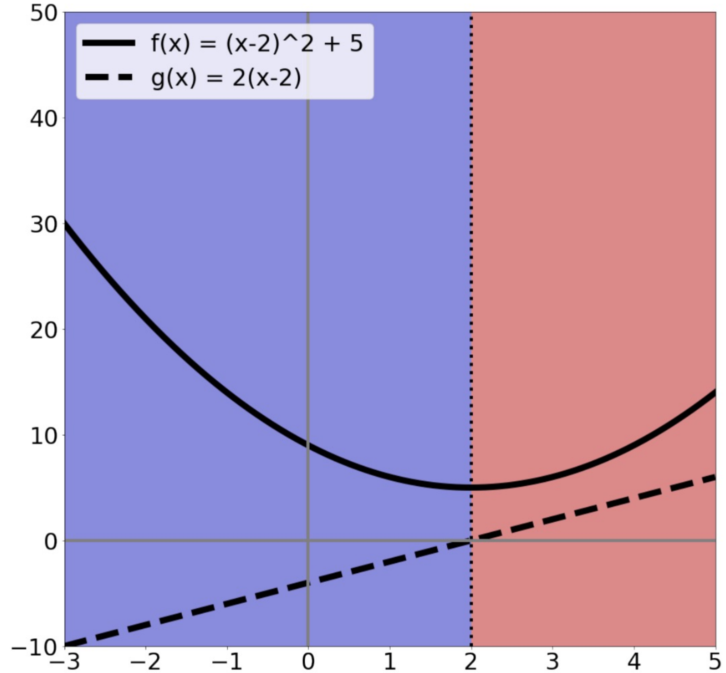
Rates of change

$$f(x,y) = (x-2)^2 + 5$$

Suppose I want to increase f(x) by changing x:

Blue area: move left
Red area: move right

Derivative tells you direction of ascent and rate



https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Derivatives

Basics

- Given a scalar-valued function $f: \mathbb{R} \to \mathbb{R}$, the **derivative** $f'(x) \in \mathbb{R}$ of f at the point $x \in \mathbb{R}$ is the rate at which the function changes at that point

Derivatives

Partial derivatives

- Pretend other variables are constant, take a derivative. That's it.
- Make our function a function of two variables

$$f(x) = (x-2)^2 + 5$$

$$\frac{\partial}{\partial x} f(x) = 2(x-2) * 1 = 2(x-2)$$

$$f_2(x,y) = (x-2)^2 + 5 + \boxed{(y+1)^2}$$

Pretend it's constant → derivative = 0

$$\frac{\partial}{\partial x} f_2(x) = 2(x-2)$$

https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

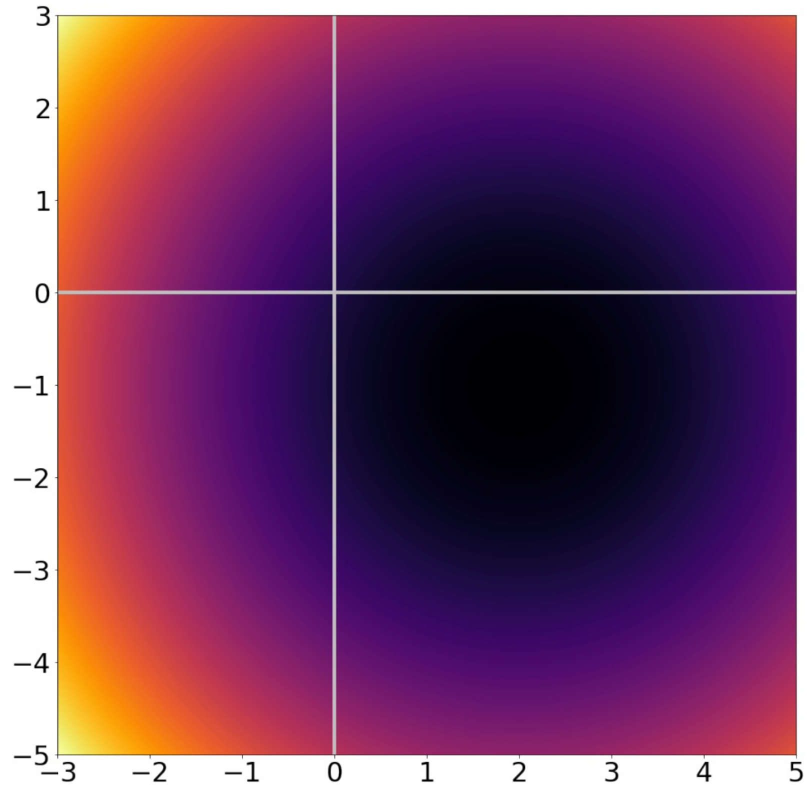
Derivatives

Partial derivatives

Zooming Out

$$f_2(x, y) = (x-2)^2 + 5 + (y+1)^2$$

Dark = f(x,y) low
Bright = f(x,y) high



https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Derivatives

Partial derivatives

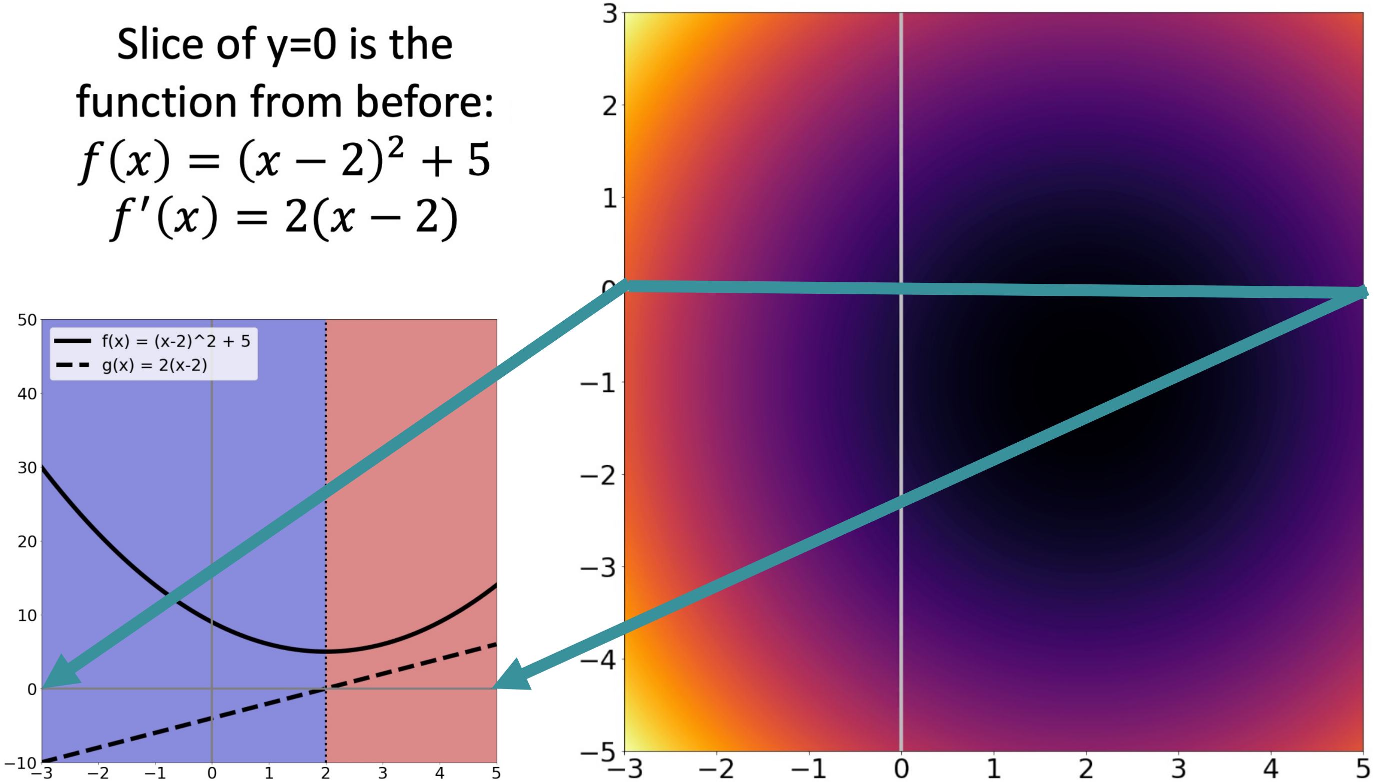
Taking a slice of

$$f_2(x, y) = (x-2)^2 + 5 + (y+1)^2$$

Slice of y=0 is the function from before:

$$f(x) = (x-2)^2 + 5$$

$$f'(x) = 2(x-2)$$



Derivatives

Partial derivatives

Zooming Out

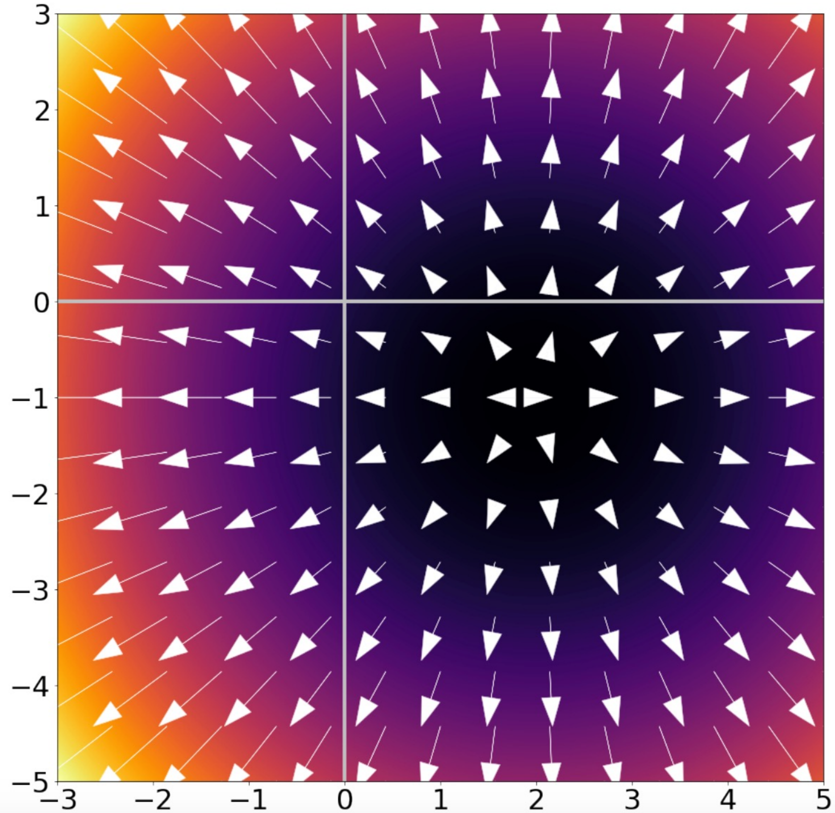
$$f_2(x, y) = (x-2)^2 + 5 + (y+1)^2$$

Gradient/Jacobian: Making a vector of

$$\nabla_f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

gives rate and direction of change.

Arrows point OUT of minimum / basin.



https://web.eecs.umich.edu/~justincj/slides/eecs442/WI2021/442_WI2021_lecture04.pdf

Next lecture: Image filtering

The first assignment is due on this Friday

extremely simple! 10 mins work,
But 4 points !

Thank you very much!

sihengc@sjtu.edu.cn