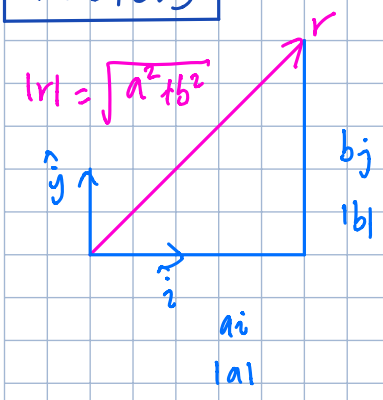
Mathematics for ML Specialisation

Course 1: Linear Algebra

vectors

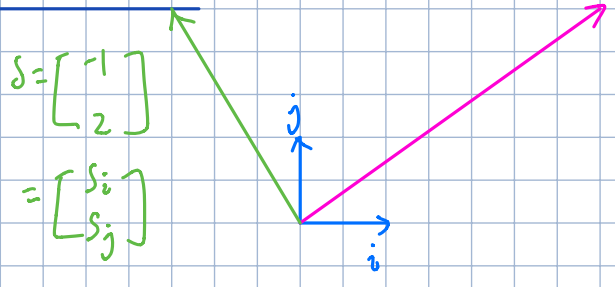
$\hat{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\hat{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



$r = a\hat{i} + b\hat{j} = \begin{bmatrix} a \\ b \end{bmatrix}$

$|r| = \sqrt{a^2 + b^2}$

dot product



$r \cdot s = r_i s_i + r_j s_j = 1$

commutative:

$r \cdot s = s \cdot r$

distributive over addition

$r \cdot (s + t) = r \cdot s + r \cdot t$

associative over scalar multiplication

$r \cdot (as) = a(r \cdot s) \quad a \in \mathbb{N}$

$r \cdot r = r_1 r_1 + r_2 + r_2$

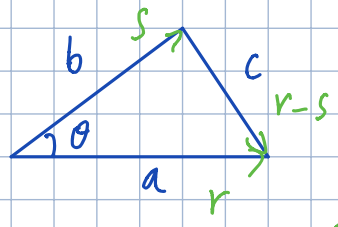
$= r_1^2 + r_2^2$

$= \left(\sqrt{r_1^2 + r_2^2}\right)^2$

$= |r|^2$

$\hookrightarrow |r| = \sqrt{r \cdot r}$

cosine rule



$c^2 = a^2 + b^2 - 2ab\cos\theta$

$|r - s|^2 = |r|^2 + |s|^2 - 2|r||s|\cos\theta$

$(r - s) \cdot (r - s) = r \cdot r - s \cdot r - s \cdot r - s \cdot -s$

$= |r|^2 - 2s \cdot r + |s|^2$

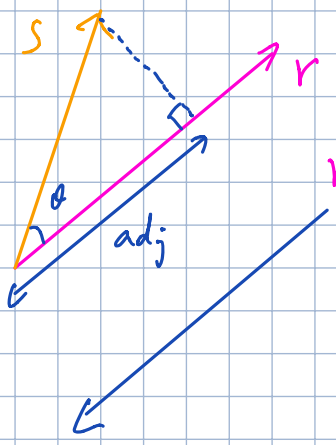
$$r \cdot s = |r||s|\cos\theta$$

i.e. $\cos 90^\circ = 0 \Rightarrow r \cdot s = 0$ ⊥

$\cos 0 = 1 \Rightarrow r \cdot s = |r||s|$ $\Rightarrow r$ $\Rightarrow s$

$\cos 180 = -1 \Rightarrow r \cdot s = -|r||s|$ $\Leftarrow r$ $\Rightarrow s$

Projection



$$\cos\theta = \frac{adj}{hyp} = \frac{adj}{|s|}$$

$$r \cdot s = |r||s|\cos\theta$$

$|s|\cos\theta$ = adj = |r| × projection (S onto R)

$$\frac{r \cdot s}{|r|} = |s|\cos\theta$$

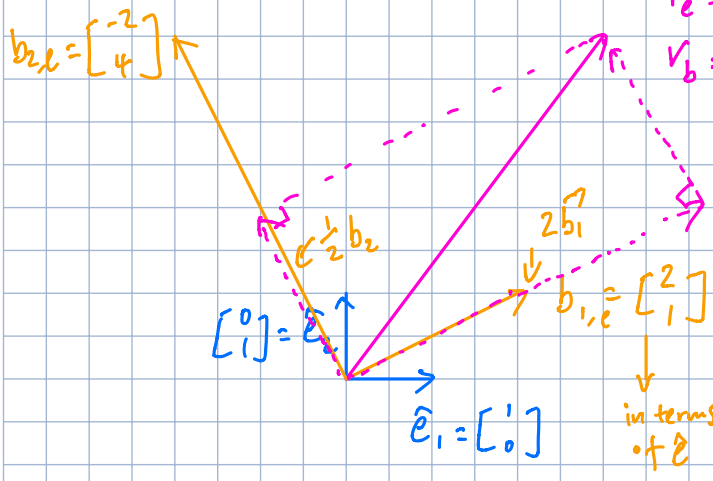
scalar projection → how much s goes along r

$$\frac{r \cdot s}{|r|}\frac{\vec{r}}{|r|}$$

vector projection → scalar projection encoded with direction along r

---

Changing basis



choice of $\{\hat{e}_1, \hat{e}_2\}$ is arbitary, we can get another basis $\{b_1, b_2\}$

i.e. $b_{1,e}$ and $b_{2,e}$

if we knows what $\{b_1, b_2\}$ in terms of $\{\hat{e}_1, \hat{e}_2\}$ ⇒ find out r_b

if $b_2 \perp b_1$ ⇒ can use dot product

else construct transformation matrix ⇒ act on the basis $e_i \to b_i$:

$$\frac{r_e \cdot b_1}{|b_1|^2}\vec{b}_1 = \frac{3\times 2 + 4\times 1}{2^2+1^2} = \boxed{2\vec{b}_1} = 2\begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}4\\2\end{bmatrix}$$

$$\frac{r_e \cdot b_2}{|b_2|^2}\vec{b}_2 = \boxed{\frac{1}{2}\vec{b}_2} = \frac{1}{2}\begin{bmatrix}-2\\4\end{bmatrix} = \begin{bmatrix}-1\\2\end{bmatrix}$$

$r_e = \begin{bmatrix}3\\4\end{bmatrix}$

$r_b = \begin{bmatrix}2\\1/2\end{bmatrix}$

Basis

1) linearly independent
2) span the space

the space is n-dimensional

Matrix

matrix transform space

$$\begin{pmatrix}2 & 3\\ 10 & 1\end{pmatrix}\begin{bmatrix}a\\b\end{bmatrix} = \begin{bmatrix}8\\13\end{bmatrix}$$

$$A \quad r = r'$$

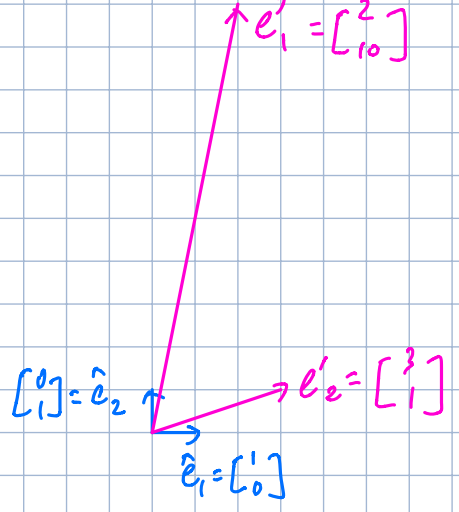
↳ $A(n\hat{e}_1 + m\hat{e}_2) = nA\hat{e}_1 + mA\hat{e}_2$

r ↓ $= ne_1' + me_2'$

i.e. A transforms the basis

↳ into a different 'grid system'.

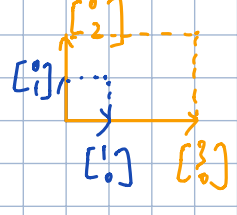
n, m unchanged



$\hat{e}_1 \to e_1'$
$\hat{e}_2 \to e_2'$
} transformed by A

Identity matrix: $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} = \begin{bmatrix}x\\y\end{bmatrix}$

scale matrix (leading diagonal): $\begin{bmatrix}3 & 0\\ 0 & 2\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} = \begin{bmatrix}3x\\2y\end{bmatrix}$



invert matrix $\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} = \begin{bmatrix}-x\\y\end{bmatrix}$

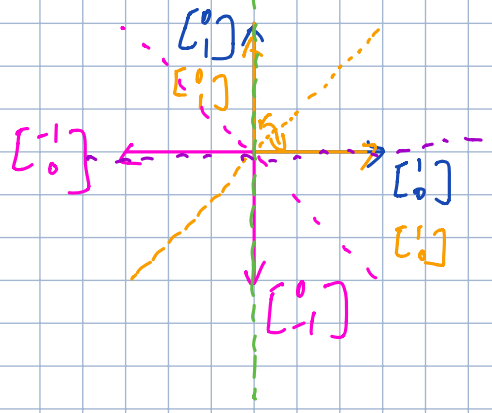
mirror

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

$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$

$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$

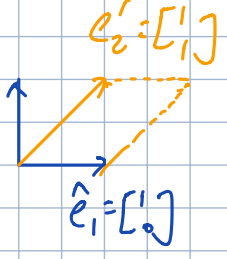
$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$



shear

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

$[\begin{smallmatrix}0\\1\end{smallmatrix}] = \hat{e}_2$



rotation

$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ 2D

$\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$ rotate about z-axis

↳ z is preserved

Composition of transformations

$A_3(A_2 A_1) = (A_3 A_2) A_1$

$A_3 A_2 A_1 \neq A_3 A_1 A_2$

Gaussian elimination

Given $A\vec{r} = \vec{s}$ e.g. $\begin{pmatrix} 2 & 3 \\ 10 & 1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 8 \\ 13 \end{pmatrix}$, solve $\begin{pmatrix} a \\ b \end{pmatrix}$

↳ $A^{-1}A\vec{r} = A^{-1}\vec{s}$

$r = A^{-1}s$ ⇒ find A^{-1} ⇒ get $r = \begin{pmatrix} a \\ b \end{pmatrix}$

more complicated (just Gaussian elimination, nothing to do with inverse)

$$\begin{pmatrix} 1 & 1 & 3 \\ 1 & 2 & 4 \\ 1 & 1 & 2 \end{pmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 15 \\ 21 \\ 13 \end{bmatrix}$$

↳ $\begin{pmatrix} 1 & 1 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & +1 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 15 \\ 6 \\ +2 \end{pmatrix} \rightarrow\rightarrow\rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 2 \end{pmatrix}$ *

↳ only solves a, b, c for a specific $\begin{bmatrix} 15 \\ 21 \\ 13 \end{bmatrix}$, i.e. specific $\vec{5}$

↳ for any $\vec{5}$, find A^{-1}

If $AB = I$ i.e. $B = A^{-1}$ ⇒ find B (or A^{-1}) i.e. all b_{ij}

↳ $\begin{pmatrix} 1 & 1 & 3 \\ 1 & 2 & 4 \\ 1 & 1 & 2 \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

↑ x-axis e.g.

↳ $\begin{pmatrix} & A & \end{pmatrix}\begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ Gaussian elimination →

↳ 2nd column of B →

↳ 3rd column of B →

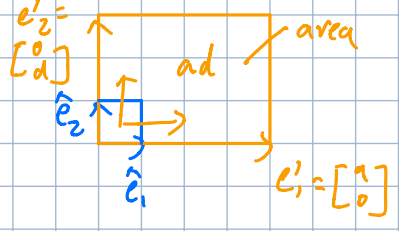
OR do all at once

↳ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} & B & \end{pmatrix} = \begin{pmatrix} 0 & -1 & 2 \\ -2 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix} = B$

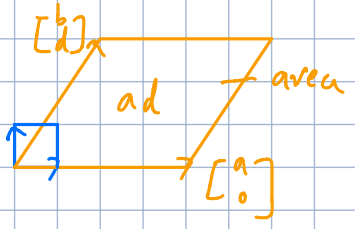
---

determinant

$\begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}$

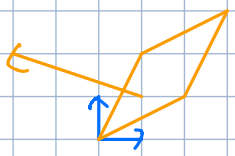


$$\begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$$



$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

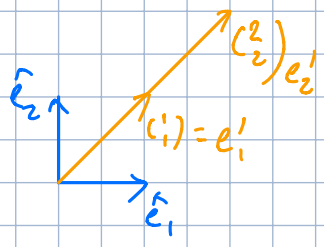
area $= ad - bc$



$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$A^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$



$\det(A) = 0$

↳ no inverse 2D → 1D ∵ basis is linearly dependent

⊗ cannot 'undo' this transformation

↳ ALWAYS check the basis on the transformation matrix is linearly independent, if not dimension is collapsed during transformation

---

Einstein Sum Convention

$$\begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nm} \end{pmatrix}_A \begin{pmatrix} b_{11} & & \\ & \ddots & \\ & & b_{nm} \end{pmatrix}_B = \begin{pmatrix} ab_{11} & & \\ & \ddots & \\ & & ab_{nm} \end{pmatrix}$$

$$ab_{ik} = \sum_j a_{ij} b_{jk} \overset{\text{Einstein Sum}}{=} a_{ij} b_{jk}$$

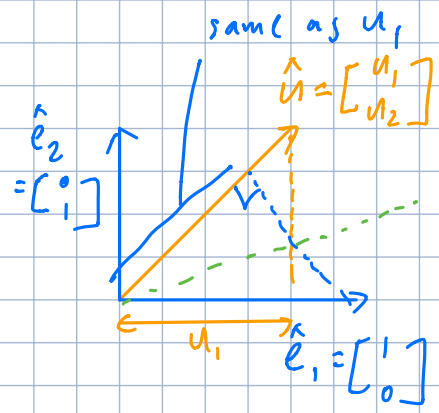
$AB = C$

$C_{ik} = a_{ij} b_{jk}$

$u \cdot v$

$\begin{pmatrix} u_i \end{pmatrix} \cdot \begin{pmatrix} v_i \end{pmatrix}$

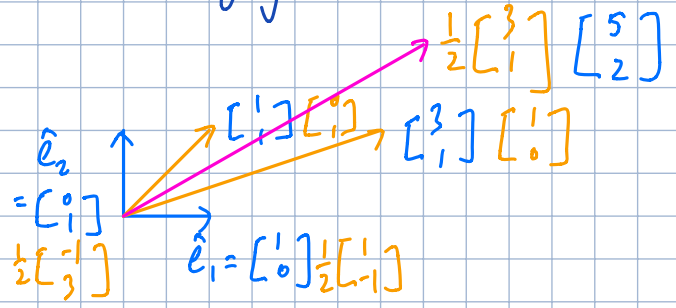
$\hookrightarrow u_i v_i$



dot product is symmetric

---

matrices changing basis



non-orthonormal

Bear's basis vectors is $\begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ in my frame

i.e. Bear's transformation matrix $\begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}$, i.e. $\begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \text{bear} \\ \text{rep} \end{bmatrix} = \begin{bmatrix} \text{me} \\ \text{rep} \end{bmatrix}$

$$\begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

Bear's basis in my coord.

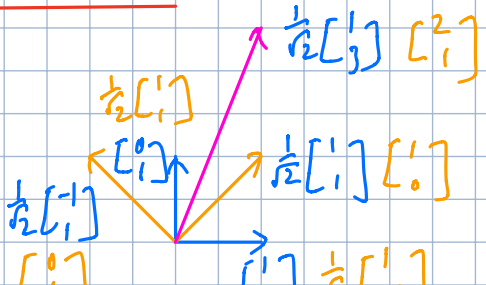
B

$$\Rightarrow B^{-1} \begin{bmatrix} \text{my} \\ \text{rep} \end{bmatrix} = \begin{bmatrix} \text{bear} \\ \text{rep} \end{bmatrix}$$

$$\frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

my basis in Bear's coord

orthonormal



Bear's transformation matrix

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\begin{bmatrix}2\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 3\end{bmatrix}$$

B

B^{-1}

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 3\end{bmatrix} = \begin{bmatrix}2\\ 1\end{bmatrix}$$

using Projection: if orthogonal

$$\left.\begin{aligned}\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 3\end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} &= \frac{1}{2}4 = 2\\ \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 3\end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{bmatrix}-1\\ 1\end{bmatrix} &= \frac{1}{2}2 = 1\end{aligned}\right\}\begin{bmatrix}2\\ 1\end{bmatrix}$$

(normalised vector)

---

say $B = \begin{pmatrix}3 & 1\\ 1 & 1\end{pmatrix}$ bear's basis

and a vector $\begin{bmatrix}x\\ y\end{bmatrix}$ defined in bear's basis, and we like to do a rotation → hard ∵ funny basis

In my $\{\hat{e}_1, \hat{e}_2\}$ basis, transformation $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix} = R$

① transform $\begin{bmatrix}x\\ y\end{bmatrix}$ into my basis

$$B\begin{bmatrix}x\\ y\end{bmatrix}$$

② apply R in my representation

$$RB\begin{bmatrix}x\\ y\end{bmatrix}$$

③ transform back to bear's system

$$B^{-1}RB\begin{bmatrix}x\\ y\end{bmatrix}$$

Orthogonal matrices

if $A_{n\times n}$ is defined as

$A^T_{ij} = A_{ji}$

$$\left[\left(a_1\right)\left(a_2\right) \cdots \left(a_i\right) \right] \Rightarrow A^T A = I$$

$\hookrightarrow A^T = A^{-1}$

$a_i \cdot a_j = \delta_{ij}$

i.e. a_i is orthonormal to each other

in ML, use orthonormal basis if possible

↳ means orthonormal transform matrix

↳ transformation is reversible

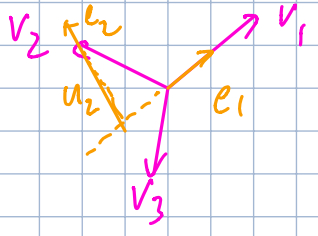
↳ inverse is easy to find

↳ projection just dot product

Gram-Schmidt

{linearly independent vectors that span the space} ⇒ {orthonormal basis set}

$V = \{v_1, v_2 \cdots v_n\}$



$v_1 \rightarrow e_1 = \frac{v_1}{|v_1|}$

$v_2 = (v_2 \cdot e_1) e_1 + u_2$

$u_2 = v_2 - (v_2 \cdot e_1) e_1 \Rightarrow \frac{u_2}{|u_2|} = e_2$

$v_3 - (v_3 \cdot e_1) e_1 - (v_3 \cdot e_2) e_2 = u_3 \Rightarrow \frac{u_3}{|u_3|} = e_3$

---

calculating eigenvector / eigen val

$A\vec{x} = \lambda\vec{x}$ A is nxn

x is n col vector

n dimension

$$\underbrace{(A-\lambda I)}_{\text{!! } 0}x = 0$$

↳ $\det(A-\lambda I)=0$

Simple example: 2×2 transformation matrix

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

↳ $\det\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}-\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right)=0 \Rightarrow \lambda^2-(a+d)\lambda+ad-bc=0$

if $A=\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \Rightarrow \det\begin{pmatrix} 1-\lambda & 0 \\ 0 & 2-\lambda \end{pmatrix}=(1-\lambda)(2-\lambda)=0$

$\lambda=1$ or 2

Substitude back $(A-\lambda I)x=0$

① $\lambda=1$ $\quad\left[\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}-\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right]x=0$

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=0$$

$$\begin{pmatrix} 0 \\ x_2 \end{pmatrix}=0$$

$\Rightarrow$ eigenvector $x=\begin{pmatrix} t \\ 0 \end{pmatrix}$

any vector along horizontal

② $\lambda=2$ $\quad\left[\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}-\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right]x=0$

$$\begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=0$$

$$\begin{pmatrix} -x_1 \\ 0 \end{pmatrix}=0$$

$$\begin{pmatrix} x_1 \\ 0 \end{pmatrix}=0$$

$\Rightarrow x=\begin{pmatrix} 0 \\ t \end{pmatrix}$

any vertical vector

e.g. rotation 90°

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \Rightarrow \det\left(\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right) = \det\begin{pmatrix} -\lambda & -1 \\ 1 & -\lambda \end{pmatrix}$$

$$= \lambda^2 + 1 = 0$$

↳ no real soln ⇒ no eigenvalue

---

changing to the eigenbasis

motivation:

Given transformation $T = \begin{pmatrix} 0.9 & 0.8 \\ -1 & 0.35 \end{pmatrix}$

and if $V_0 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$, we find $V_1 = TV_0$

↳ $V_2 = TV_1 = T^2V_0$

↳ $V_3 = TV_2 = T^3V_0$

⇒ tedious to find T^n

if T is a diagonal matrix: $T = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$

↳ $T^n = \begin{pmatrix} a^n & 0 & 0 \\ 0 & b^n & 0 \\ 0 & 0 & c^n \end{pmatrix}$

if T is NOT a diagonal matrix ⇒ change to a basis where T becomes diagonal ⇒ eigenbasis

↳ get T^n

↳ transform back

∴ each column of transformation matrix represents the new location of the transformed unit vector

∴ use eigenvectors as columns of the eigenbasis transformation matrix

transform matrix $C = \begin{pmatrix} x_1 & x_2 & x_3 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{pmatrix}$ x_i are eigenvectors (for 3D)

$D = \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix}$ λ_i are eigenvalue of T

After applying C to the vector v_0, applying T is just scaling $\Rightarrow$ T is diagonal in this frame, which is D, it contain eigenvalue of T

i.e $T \iff D$

our basis — eigen basis

This means that:

Apply T = ① convert to eigen basis
② apply the diagonalise matrix
③ convert back

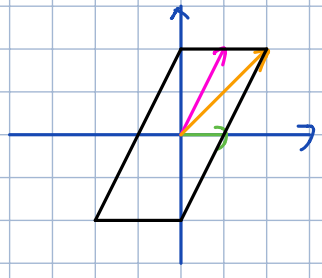
↳ $T = CDC^{-1} \iff D = C^{-1}TC$

↳ $T^2 = CDC^{-1}CDC^{-1}$
$= CD^2C^{-1}$
$T^n = CD^nC^{-1}$

---

eigen basis example: 2×2

$T = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$



consider the $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ vector (old orange vector)

$\begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$

<u>eigen vector</u> $v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $\lambda = 1$

$v = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\lambda = 2$

Now consider $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ vector

$T^2\begin{pmatrix} -1 \\ 1 \end{pmatrix}$: $\begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$ ↖ same

$$T^2 = \begin{pmatrix}1&1\\0&2\end{pmatrix}\begin{pmatrix}1&1\\0&2\end{pmatrix} = \begin{pmatrix}1&3\\0&4\end{pmatrix} \Rightarrow \begin{pmatrix}1&3\\0&4\end{pmatrix}\begin{pmatrix}-1\\1\end{pmatrix} = \begin{pmatrix}2\\4\end{pmatrix}$$

eigen basis way:

$$C = \begin{pmatrix}1&1\\0&1\end{pmatrix} \Rightarrow C^{-1} = \begin{pmatrix}1&-1\\0&1\end{pmatrix}$$

same

$$\Rightarrow T^2 = CD^2C^{-1} = \begin{pmatrix}1&1\\0&1\end{pmatrix}\begin{pmatrix}1&0\\0&2\end{pmatrix}^2\begin{pmatrix}1&-1\\0&1\end{pmatrix} = \begin{pmatrix}1&3\\0&4\end{pmatrix}$$

Course 2: Multivariate Calculus

Gradient

in 2D, gradient at $x = \lim_{\Delta x \to 0}\left(\frac{f(x+\Delta x)-f(x)}{\Delta x}\right) = f'(x) = \frac{df}{dx}$

e.g. $f(x) = 3x+2$

$$\hookrightarrow f'(x) = \lim_{\Delta x\to 0}\left(\frac{3(x+\Delta x)+2-3x-2}{\Delta x}\right)$$

$$= \lim_{\Delta x\to 0}\left(\frac{3x+3\Delta x+2-3x-2}{\Delta x}\right)$$

$$= \lim_{\Delta x\to 0}\left(\frac{3\Delta x}{\Delta x}\right)$$

$$= 3$$

Sum rule:

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

product rule:

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

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

chain rule:

$h = h(p) \qquad p = p(m)$

$$\frac{dh}{dm} = \frac{dh}{dp} \cdot \frac{dp}{dm}$$

Total derivative

$$\frac{df(x,y,z)}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt}$$

e.g. $f(x,y,z) = \sin(x)e^{yz^2}$ and $\begin{cases} x = t-1 \\ y = t^2 \\ z = 1/t \end{cases}$

Jacobian

Jacobian vector points uphill

if we have $f(x_1, x_2, x_3, \cdots)$

Jacobian is a vector where each entry is the partial derivative of f w.r.t. each one of the variable

$$J = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \cdots\right]$$ usually row vector

e.g. $f(x,y,z) = x^2y + 3z$

$$\left.\begin{aligned} \frac{\partial f}{\partial x} &= 2xy \\ \frac{\partial f}{\partial y} &= x^2 \\ \frac{\partial f}{\partial z} &= 3 \end{aligned}\right\} \quad J = [2xy, x^2, 3]$$

when (x,y,z) is supplied, $\vec{J}$ will point in the

direction of steepest slope of this function.

e.g $J(0,0,0) = [0, 0, 3]$ ↳ Steeper ⇒ bigger Jacobian

⊗ visualise in 2-D ⇒ $f(x,y)$ where f ⇒ is height

Jacobian Matrix

Jacobian matrix describes functions and the gradient of a multivariable systems

consider 2 functions:

$$u(x,y) = x - 2y$$
$$v(x,y) = 3y - 2x$$

} can make seperate row vector Jacobian

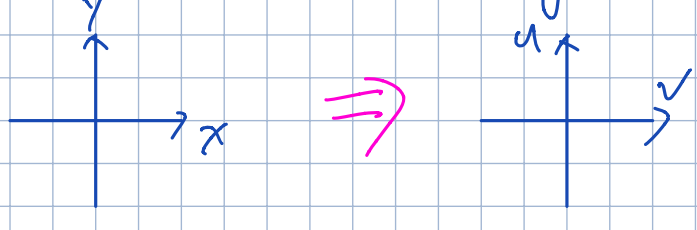
$$J_u = \left[\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right]$$
$$J_v = \left[\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}\right]$$

$$J = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix}$$

$$\frac{\partial u}{\partial x} = 1, \quad \frac{\partial u}{\partial y} = -2$$
$$\frac{\partial v}{\partial x} = -2, \quad \frac{\partial v}{\partial y} = 3$$

$$J = \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix}$$

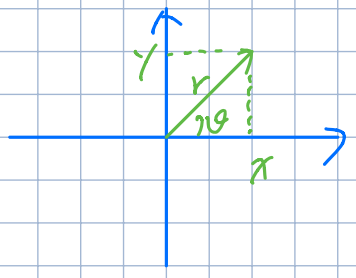
gradient is constant every where



this J transforms vector from xy space ⇒ uv space

$$J\begin{bmatrix}2\\3\end{bmatrix} = \begin{bmatrix}-4\\5\end{bmatrix}$$

e.g transforming cartesian → polar coordinate



$$x(r,\theta) = r\cos\theta$$
$$y(r,\theta) = r\sin\theta$$

↳ $$J = \begin{bmatrix}\frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta}\\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & -r\sin\theta\\ \sin\theta & r\cos\theta\end{bmatrix}$$

$$|J| = r(\cos^2\theta + \sin^2\theta) = r$$

↑ determinant

Hessian

Jacobian : collect all the 1st order derivatives of a function in to a vector.

Hessian : 2nd order derivatives into a matrix

$$H = \begin{bmatrix}\partial^2 f/\partial x_1^2 & \partial^2 f/\partial x_1 x_2 & \cdots & \partial^2 f/\partial x_1 x_n\\ \vdots & \ddots & & \\ \partial^2 f/\partial x_n x_1 & & & \partial^2 f/\partial x_n^2\end{bmatrix}$$

shorthand

$$\partial_{x_1 x_1} f \equiv \partial^2 f/\partial x_1^2$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial}{\partial x_1}\left(\frac{\partial f}{\partial x_2}\right)$$

↳ n×n square matrix

↳ n = no. of variable in function f

e.g. $f(x,y,z) = x^2yz$

$$J = [2xyz,\ x^2z,\ x^2y]$$

∂ ∂ ∂

$$H = \begin{bmatrix} 2yz & 2xz & 2xy \\ 2xz & 0 & x^2 \\ 2xy & x^2 & 0 \end{bmatrix}$$

symmetrical about diagonal

$\Rightarrow$ True if func is continuous

e.g. 2D example $f(x,y) = x^2 + y^2$

$$J = [2x, 2y]$$

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$|H| = 4$ { +ve: max or min
  ↳ look at $H[0,0]$
  ↳ +ve: min
  ↳ -ve: max
-ve: saddle

Multivariable chain rule

if we have

$$f(x_1, x_2, x_3, \cdots, x_n) = f(\vec{x})$$ ← n-dimensional vector

where $x_1(t)$ $x_2(t)$

we would like to find $\frac{df}{dt}$

recall total derivative

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt}$$

we need

$$\frac{\partial f}{\partial \vec{x}} = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix} \qquad \frac{\partial \vec{x}}{dt} = \begin{bmatrix} dx_1/dt \\ \vdots \\ dx_n/dt \end{bmatrix}$$

$$\frac{df}{dt} = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix} \cdot \begin{bmatrix} dx_1/dt \\ \vdots \\ dx_n/dt \end{bmatrix} = \frac{\partial f}{\partial x} \cdot \frac{d\vec{x}}{dt}$$

i.e. if $f(\vec{x}(t))$, then $\frac{df}{dt} = \frac{\partial f}{\partial \vec{x}} \cdot \frac{d\vec{x}}{dt}$

and $\frac{\partial f}{\partial \vec{x}}$ is just same as a Jacobian vector (but col vector instead of row vector)

$$\hookrightarrow \quad \frac{\partial f}{\partial \vec{x}} = (J_f)^T$$

$$\Rightarrow \frac{df}{dt} = J_f \frac{d\vec{x}}{dt}$$

(row/col doesn't matter here in operations)

e.g. more than 2 links, univariable (x)

$$f(x) = 5x$$
$$x(u) = 1-u$$
$$u(t) = t^2$$

method 1:

$$f(t) = 5(1-u) = 5(1-t^2) = 5-5t^2$$

$$\frac{df}{dt} = -10t$$

method 2:

$$\frac{df}{dt} = \frac{df}{dx}\frac{dx}{du}\frac{du}{dt} = (5)(-1)(2t) = -10t$$

multivariable, more than 2 links

$$f(\vec{x}(\vec{u}(t)))$$

$$f(\vec{x}) = f(x_1, x_2)$$

$$\vec{x}(\vec{u}) = \begin{bmatrix} x_1(u_1, u_2) \\ x_2(u_1, u_2) \end{bmatrix}$$

$$\vec{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

$t \hookrightarrow f$

↳ $$\frac{df}{dt} = \frac{\partial f}{\partial \vec{x}} \frac{\partial \vec{x}}{\partial \vec{u}} \frac{d\vec{u}}{dt}$$

$$= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right] \begin{bmatrix} \frac{\partial x_1}{\partial u_1} & \frac{\partial x_1}{\partial u_2} \\ \frac{\partial x_2}{\partial u_1} & \frac{\partial x_2}{\partial u_2} \end{bmatrix} \begin{bmatrix} \frac{du_1}{dt} \\ \frac{du_2}{dt} \end{bmatrix}$$

Jacobian vector of f | Jacobian matrix of x | derivative vector of u

$(1\times1) = (1\times2)\,(2\times2)\,(2\times1)$ ⇒ returns a scalar

---

A simple neural network

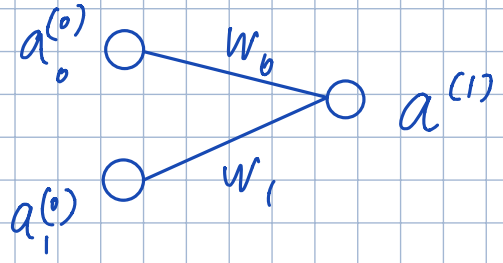
$$a^{(1)} = \sigma(W a^{(0)} + b)$$

a: activity

W: weight

b: bias

σ: activation function

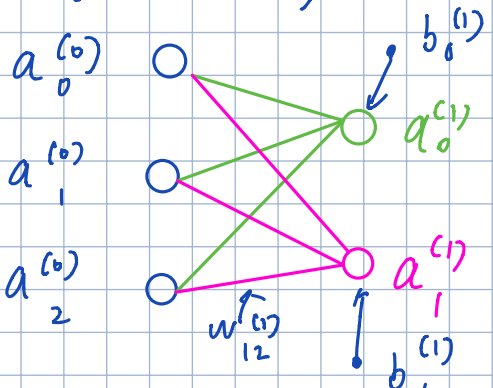


now $a^{(1)} = \sigma(W_0 a_0^{(0)} + W_1 a_1^{(0)} + b)$

for n input

$$a^{(1)} = \sigma\left[\left(\sum_{j=1}^{n} W_j a_j^{(1)}\right) + b\right] = \sigma(\vec{W} \cdot \vec{a}^{(0)} + b)$$

adding more output



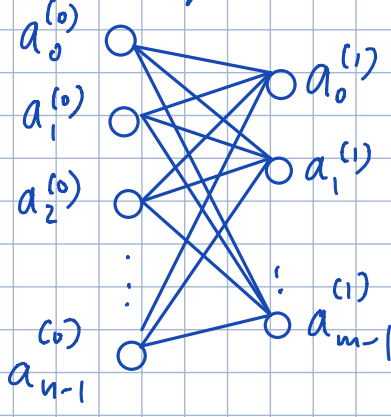
$$a_0^{(1)} = \sigma(\vec{w}_0 \cdot \vec{a}^{(0)} + b_0)$$

$$a_1^{(1)} = \sigma(\vec{w}_1 \cdot \vec{a}^{(0)} + b_1)$$

$$\vec{a}^{(1)} = \sigma(\vec{W}^{(1)} \cdot a^{(0)} + \vec{b}^{(1)})$$

$\vec{a}^{(1)}$: vector, $\vec{W}^{(1)}$: matrix, $\vec{b}^{(1)}$: vector

Single layer



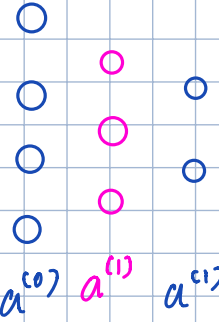
To describe this function

$$\vec{a}^{(1)} = \sigma(\vec{W}^{(1)} \cdot \vec{a}^{(0)} + \vec{b}^{(1)})$$

$$\begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \\ \vdots \\ a_{m-1}^{(1)} \end{bmatrix} = \sigma\left(\begin{bmatrix} w_{0,0}^{(1)} & w_{0,1}^{(1)} & \cdots & w_{0,n-1}^{(1)} \\ w_{1,0}^{(1)} & & & \\ \vdots & & \ddots & \\ w_{m-1,0}^{(1)} & & & w_{m-1,n-1}^{(1)} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ \vdots \\ a_{n-1}^{(0)} \end{bmatrix} + \begin{bmatrix} b_0^{(1)} \\ \vdots \\ b_{m-1}^{(1)} \end{bmatrix} \right)$$

$w_{ij}^{(1)}$ is the link between neuron j in previous layer and i in current layer

hidden layer

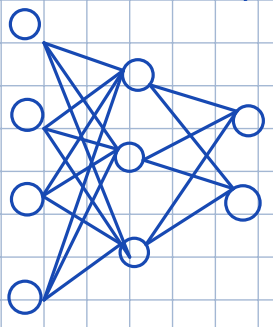


$$a^{(1)} = \sigma(\vec{W}^{(1)} \cdot \vec{a}^{(0)} + b^{(1)})$$

$$a^{(2)} = \sigma(\vec{W}^{(2)} \cdot a^{(1)} + b^{(2)})$$

i.e. $a^{(L)} = \sigma(\vec{W}^{(L)} \cdot \vec{a}^{(L-1)} + \vec{b}^{(L)})$

Back propergation



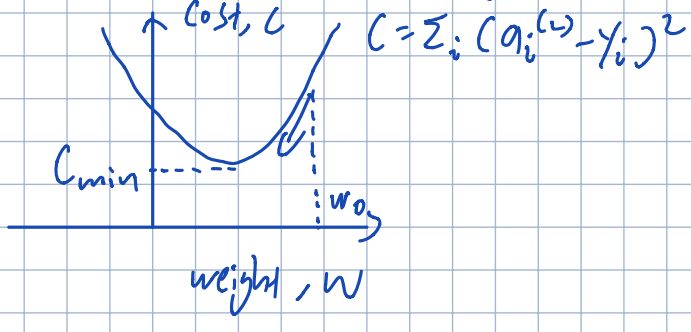
$$\vec{a}^{(L)} = \sigma(\vec{W}^{(L)} \cdot \vec{a}^{(L-1)} + \vec{b}^{(L)})$$

find 18 weights and 5 bias that cause our network to best match train inputs to their labels.
(start with random numbers)
then we define a cost function

$$C = \sum_i (a_i^{(L)} - y_i)^2$$

↑ desired output

e.g. of a particular weight



e.g. 2-node network

$a^{(0)}$ ——— $a^{(1)}$

$$a^{(1)} = \sigma(wa^{(0)} + b)$$

$$C = (a^{(1)} - y)^2$$

$$\begin{cases} \frac{\partial C}{\partial w} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial w} \\ \frac{\partial C}{\partial b} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial b} \end{cases}$$

for squishing activation function

usually ↗

$$z^{(1)} = wa^{(0)} + b \qquad C = (a^{(1)} - y)^2$$

$$a^{(1)} = \sigma(z^{(1)})$$

$$\begin{cases} \frac{\partial C}{\partial w} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial w} \\ \frac{\partial C}{\partial b} = \frac{\partial C}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial b} \end{cases}$$

For multiple neuron / layer

$$\vec{z}^{(L)} = \vec{W}^{(L)} \cdot \vec{a}^{(L-1)} + \vec{b}^{(L)}$$

$$\vec{a}^{(L)} = \sigma(\vec{z}^{(L)})$$

$$C_k = \sum_i (\vec{a}_i^{(L)} - y_i)^2$$

↑ for 1 training example (1 row) - k^{th} data sample

for multiple output (still 1 data sample)

↑ i.e. features

$$C_k = |\vec{a}^{(L)} - \vec{y}|^2$$

e.g. get derivative of cost w.r.t to the weight of final layer (assume, 3 layer, 4 input, 2 output), for k^{th} data

$$\frac{\partial C_k}{\partial \vec{W}^{(2)}} = \frac{\partial C_k}{\partial \vec{a}^{(2)}} \frac{\partial \vec{a}^{(2)}}{\partial \vec{z}^{(2)}} \frac{\partial \vec{z}^{(2)}}{\partial \vec{W}^{(2)}}$$

derivative of cost w.r.t weight of previous layer

$$\frac{\partial C_k}{\partial \vec{W}^{(1)}} = \frac{\partial C_k}{\partial \vec{a}^{(2)}} \frac{\partial \vec{a}^{(2)}}{\partial \vec{a}^{(1)}} \frac{\partial \vec{a}^{(1)}}{\partial \vec{z}^{(1)}} \frac{\partial \vec{z}^{(1)}}{\partial \vec{W}^{(1)}}$$

where $$\frac{\partial \vec{a}^{(2)}}{\partial \vec{a}^{(1)}} = \frac{\partial \vec{a}^{(2)}}{\partial \vec{z}^{(2)}} \frac{\partial \vec{z}^{(2)}}{\partial \vec{a}^{(1)}}$$

generalise

$$\frac{\partial C_k}{\partial \vec{W}^{(i)}} = \frac{\partial C_k}{\partial \vec{a}^{(N)}} \underbrace{\frac{\partial \vec{a}^{(N)}}{\partial \vec{a}^{(N-1)}} \frac{\partial \vec{a}^{(N-1)}}{\partial \vec{a}^{(N-2)}} \cdots \frac{\partial \vec{a}^{(i+1)}}{\partial \vec{a}^{(i)}}}_{\text{from layer N to layer i}} \frac{\partial \vec{a}^{(i)}}{\partial \vec{z}^{(i)}} \frac{\partial \vec{z}^{(i)}}{\partial W^{(i)}}$$

where $$\frac{\partial a^{(j)}}{\partial a^{(j-1)}} = \sigma'(\vec{z}^{(j)}) \vec{W}^{(j)}$$

$\because a^{(n)} = \sigma(z^{(n)})$ (scalar)

$a^{(n)} = \sigma(W^{(n)} a^{(n-1)} + b^{(n)})$

$z^{(n)} = w^{(n)} a^{(n-1)} + b^{(n)}$

Taylor Series (Power Series)

$g(x) = a + bx + cx^2 + dx^3 + \dots$

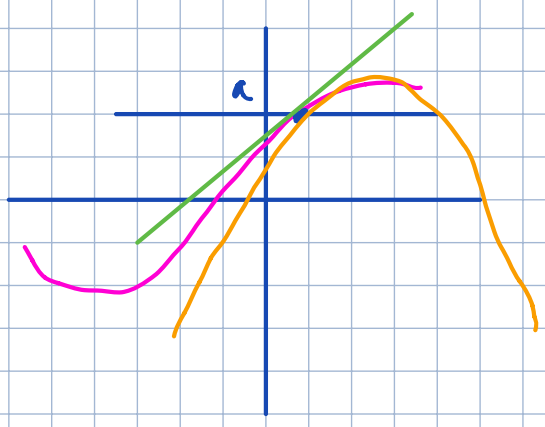
$g_0(x) = a$ 0^{th} order approximation

$g_1(x) = a + bx$ 1^{st} order approximation

$g_2(x) = a + bx + cx^2$ 2^{nd} order approximation

⋮

e.g.



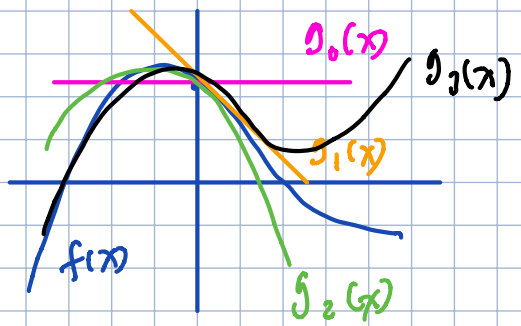
$g_0(x) = a$ 0^{th} order approx

$g_1(x) = a + bx$ 1^{st} order

$g_2(x) = a + bx + cx^2$ 2^{nd} order

If we know everything about the function at some point (i.e. its function value, 1^{st}, 2^{nd}... derivatives etc.) we can reconstruct the function everywhere else. Only for well behaved function. (i.e. continuous, can be differentiate as many times as we want)

e.g. choose the point $x = 0$



$g_0(x) = f(0)$

$g_1(x) = f(0) + f'(0)x$

$g_2(x) = f(0) + f'(0)x + \frac{1}{2}f''(0)x^2$

$g_3(x) = f(0) + f'(0)x + \frac{1}{2}f''(0)x^2 + \frac{1}{6}f^{(3)}(0)x^3$

$\Rightarrow g_4(x) = \frac{f^{(0)}(0)}{0!} + \frac{f^{(1)}(0)}{1!}x + \frac{f^{(2)}(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \frac{f^{(4)}(0)}{4!}x^4$

n^{th} term: $\frac{1}{n!} f^{(n)}(0) x^n$

$$\Rightarrow \boxed{g(x) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) x^n}$$

Maclaurin series
(∵ at $x=0$)
a special case

$$\boxed{g(x) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(p) (x-p)^n}$$

1-D Taylor series

---

at point p

$$g_1(x) = f(p) + f'(p)(x-p)$$

(∵ $(x-p) \times f'(p) =$ rise
run × gradient)

approximation near p

$$g_1(p+\Delta p) = f(p) + f'(p)(\Delta p)$$

substitute to x (same)

$$g_1(x+\Delta x) = f(x) + f'(x)(\Delta x)$$

Taylor series

$$\boxed{f(x+\Delta x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x)}{n!} \Delta x^n}$$

(focus on any 1 point $+\Delta x$)
where $\Delta x = x - p$

1st order approx. (linearisation)

$$f(x+\Delta x) = f(x) + f'(x)(\Delta x) + O(\Delta x^2)$$

Multi-dimensional Power Series

$$f(x+\Delta x, y+\Delta y)$$

$$= f(x,y) + (\partial_x f(x,y)\Delta x + \partial_y f(x,y)\Delta y) + \cdots$$

$$[\partial_x f(x,y), \partial_y f(x,y)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= J_f \Delta \vec{x}$$

nth — 1st order

$$+ \frac{1}{2}\left(\partial_{xx} f(x,y)\Delta x^2 + 2\partial_{xy} f(x,y)\Delta x \Delta y + \partial_{yy} f(x,y)\Delta y^2\right) + \dots$$

$$\frac{1}{2}\begin{bmatrix}\Delta x & \Delta y\end{bmatrix}\begin{bmatrix}\partial_{xx} f(x,y) & \partial_{xy} f(x,y) \\ \partial_{yx} f(x,y) & \partial_{yy} f(x,y)\end{bmatrix}\begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}$$

$$\frac{1}{2}\Delta x^T H_f \Delta x$$

i.e. $$\boxed{f(\vec{x}+\Delta\vec{x}) = f(\vec{x}) + J_f \Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T H_f \Delta\vec{x} + \dots}$$

for multi dimension Taylor Series

Newton-Raphson method

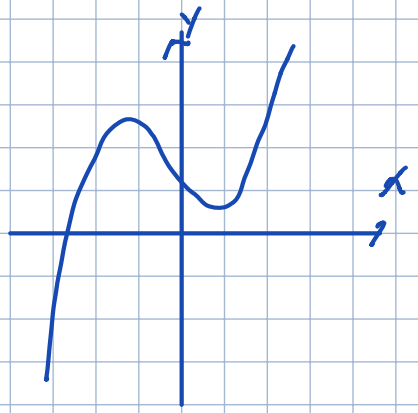
e.g. $y = x^3 - 2x + 2$

$$\frac{\partial y}{\partial x} = 3x^2 - 2$$

$$\boxed{x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}}$$

Newton-Raphson method

x_i ← prev guess



we want to find x when $y=0$

i	x_i	$y(x_i)$	$\frac{\partial y(x_i)}{\partial x}$
0	-2	-2	10
1	-1.8	-0.23	7.7
2	-1.77	-0.005	7.4
3	-1.769	-2.3E-6	

Recall we can ask what value of the function is at point $x_0 + \delta x$, a short distance away

$$f(x_0 + \delta x) = f(x_0) + f'(x_0)\delta x$$

if we assume the function goes to 0 somewhere, we can rearrange to find how far away, i.e. assume $f(x_0 + \delta x) = 0$, find δx:

$$\delta x = -\frac{f(x_0)}{f'(x_0)}$$

↳ repeat

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Optimisation: Gradient Descent

In Newton-Raphson method, we try to find the solution x, if $y=0$.
In Gradient Descent, we try to find the minimum (or max) value (we need to be 0, and imagine in a hyper-dimension contour plot)

Assume $f(x,y) = x^2 y$

how do we find the fastest / steepest way to get down the 3D graph?

$$\frac{\partial f}{\partial x} = 2xy \qquad \frac{\partial f}{\partial y} = x^2$$

put these 2 gradient in a vector

$$\nabla f = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

if we have a little vector $\vec{r} = \begin{bmatrix} dx \\ 0 \end{bmatrix}$

$\nabla f \cdot \vec{r}$ ⇒ move a bit along $\vec{r}$ direction (dx, here)

$$\begin{bmatrix} \frac{\partial f(a,b)}{\partial x} \\ \frac{\partial f(a,b)}{\partial y} \end{bmatrix}$$ ← evaluate at point (a,b)

If we want to know how much a function will change, when we move along some unit vector (in an opposite direction).

$$\nabla f \cdot \hat{r}$$

$$= \begin{bmatrix} \frac{df(a,b)}{dx} \\ \frac{df(a,b)}{dy} \end{bmatrix} \cdot \begin{bmatrix} c \\ d \end{bmatrix} = df$$

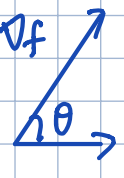
$\hat{r}$

directional gradient

$$\Rightarrow df = \frac{df}{dx} c + \frac{df}{dy} \cdot d$$

df: small change in f

what is the maximum value this directional gradient $\frac{df}{}$ can take?



when $\hat{r} \parallel \nabla f$

$$\nabla f \cdot \frac{\nabla f}{\|\nabla f\|} = \frac{\|\nabla f\|^2}{\|\nabla f\|} = \|\nabla f\|$$

$\hat{r}$

max value ↓ steepest gradient

size of ∇f

∇f points up the direction of the steepest descent

⊥ to the contour line

In Gradient descent method, we take small steps down the hill,

$$S_{n+1} = S_n - \gamma \nabla f$$

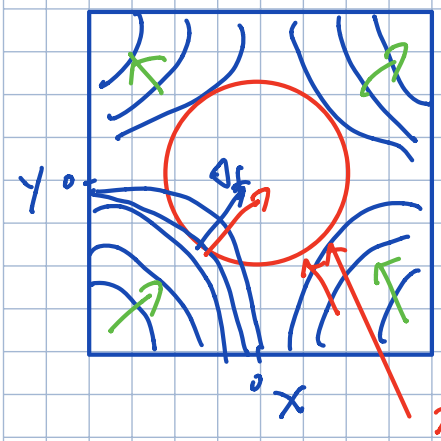
previous position

coefficient

evaluated at previous position

As we approach turning point (min or max), ∇f gets smaller.

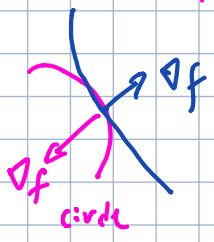
Constrained Optimisation: Lagrange Multipliers



Contour map (project to 2D)

maximise: $f(x,y) = x^2y$

Constraint: $g(x,y) = x^2+y^2 = a^2$



Solve $\nabla f = \lambda \nabla g$

where λ is Lagrange multiplier ↑ ∴ grad ⊥ contour ∴ grad time / grad constant

We wish to find min/max on the circle

max and min on the circle will be found where constraint (circle) // contour line

$$\nabla f = \nabla(x^2y) = \begin{bmatrix} 2xy \\ x^2 \end{bmatrix} = \lambda \nabla g = \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

① $2xy = \lambda 2x \Rightarrow y = \lambda$

② $x^2 = \lambda 2y \Rightarrow x^2 = 2y^2 \Rightarrow x = \pm\sqrt{2}y$

③ $x^2+y^2 = a^2 \Rightarrow 2y^2+y^2 = a \Rightarrow y = \pm a/\sqrt{3}$

Soln: $\frac{a}{\sqrt{3}}\begin{pmatrix}\sqrt{2}\\1\end{pmatrix}$, $\frac{a}{\sqrt{3}}\begin{pmatrix}\sqrt{2}\\-1\end{pmatrix}$, $\frac{a}{\sqrt{3}}\begin{pmatrix}-\sqrt{2}\\1\end{pmatrix}$, $\frac{a}{\sqrt{3}}\begin{pmatrix}-\sqrt{2}\\-1\end{pmatrix}$

$f(x,y) =$ $\frac{a^3}{3\sqrt{3}}2$ (max), $\frac{a^3}{3\sqrt{3}}(-2)$ (min), $\frac{a^3}{3\sqrt{3}}(2)$ (max), $\frac{a^3}{3\sqrt{3}}(-2)$ (min)

Simple linear regression

In a straight line data fitting, we can model the straight line Y as the i^{th} observation x_i and a vector a of the fitting parameters.

$$Y = Y(x_i; a_i) = mx_i + c \qquad a = \begin{bmatrix} m \\ c \end{bmatrix}$$

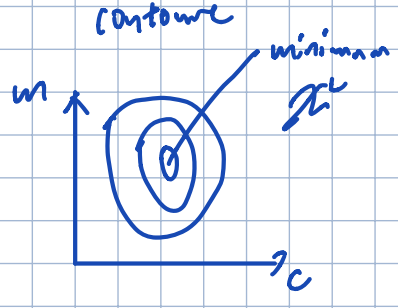
$\bar{x}, \bar{y}$ will be geometric center of mass of that dataset.

residue r = difference between the data items y_i and the predicted location of those on the line y, which will be $mx+c$

$$\Rightarrow r_i = y_i - mx_i - c$$

(expt data point)

$$\chi^2 \text{ (chi}^2\text{)} = \sum_i r_i^2 = \sum_i (y_i - mx_i - c)^2$$



Goal: minimise χ^2

Minimum is found when the gradient of $\chi^2 = 0$

i.e. $\nabla \chi^2 = \begin{bmatrix} \partial\chi^2/\partial m \\ \partial\chi^2/\partial c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ note the variable in this space is m and c

$$\frac{\partial \chi^2}{\partial m} = -2 \sum_i x_i (y_i - mx_i - c)$$

$$\frac{\partial \chi^2}{\partial c} = -2 \sum_i (y_i - mx_i - c)$$

$$\nabla \chi^2 = \begin{bmatrix} -2 \sum_i x_i (y_i - mx_i - c) \\ -2 \sum_i (y_i - mx_i - c) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

legend on m

$$c = \bar{y} - m\bar{x} \qquad m = \frac{\sum (x_i - \bar{x}) y_i}{\sum (x_i - \bar{x})^2}$$

solution

$\bar{x}$: mean

$$\sigma_c \simeq \sigma_m \sqrt{\bar{x}^2 + \frac{1}{n} \sum_i (x - \bar{x})^2}$$

$$\sigma_m^2 \simeq \frac{\chi^2}{\sum (x_i - \bar{x})^2 (n-2)}$$

General non-linear least squares

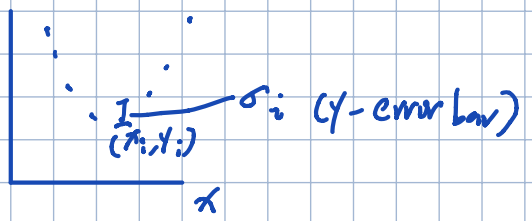
Say we have a function y, trying to fit (x, Y)

$$y(x; a_k) \qquad k = 1 \dots m$$

e.g. $(x - a_1)^2 + a_2$

a_k are parameters to evaluate for best fit

data points (Y_i, x_i, σ_i) $\quad i = 1 \dots n$



$$\chi^2 = \sum_{i=1}^{n} \frac{[Y_i - y(x_i; a_k)]^2}{\sigma^2}$$

($y(x_i; a_k)$: model)

↖ sort-of removing uncertainty weightage from χ^2 (if no uncertainty, can $\sigma \Rightarrow 1$)

minimum χ^2 when

$$\nabla \chi^2 = 0$$

most of the time we cannot solve analytically

→ solve by steepest descent

by updating the vector of fitting parameters a

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \end{bmatrix}$$

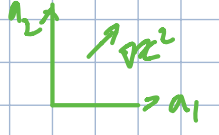
$$a_{next} = a_{curr} - \text{constant} \nabla \chi^2 = a_{curr} + \text{constant} \sum_{i=1}^{n} \frac{[Y_i - y(x_i; a_k)]}{\sigma^2} \frac{dy}{da_k}$$

$$\nabla \chi^2 = \begin{bmatrix} \partial \chi^2 / a_1 \\ \partial \chi^2 / a_2 \\ \vdots \end{bmatrix}$$

constant in front

learning rate

evaluated @ a_{curr}



until $\nabla \chi^2 = 0$ (or converge to some value)

we need $\frac{d\chi^2}{da_k}$ for each k

$$\frac{d\chi^2}{da_k} = \sum_{i=1}^{n} -2 \frac{[Y_i - y(x_i; a_k)]}{\sigma_i^2} \frac{dy}{da_k}$$

e.g. $y(x; a_k) = (x - a_1)^2 + a_2$ (i.e. $y(x; a_1, a_2)$)

(data, parameter)

$\frac{dy}{\quad}$

$\frac{dy}{da_1} = -2(x - a_1)$

$\frac{dy}{da_2} = 1$

Course 3: PCA

Mean of dataset

Say we have a dataset $D = \{x_1, \cdots, x_N\}$

$E[D] = \frac{1}{N} \sum_{n=1}^{N} x_n$

↑ Expectation value

e.g. $D' = \{1, 2, 4, 6, 6\}$

$E[D'] = 1+2+4+6+6 / 5 = 3.8$

Variance of dataset

In 1D, we can look at the average squared distance of a data point from the mean value of this dataset.

$D_1 = \{1, 2, 4, 5\}$ $E[D_1] = 3$

$D_2 = \{-1, 3, 7\}$ $E[D_1] = 3$

D_1: $\frac{(1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2}{4} = 10/4$

D_2: $\frac{(-1-3)^2 + (3-3)^2 + (7-3)^2}{3} = 32/3$

⇒ average squared distance

$X: \{x_1 \dots x_N\}$ → $\mu = E[X]$

$$Var[X] = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

↳ $\sqrt{Var[X]}$ = Standard deviation $[X]$

↙ same units as X

Covariance

$$cov[x, y] = E[(x - \mu_x)(y - \mu_y)]$$

$\mu_x = E[x]$

$\mu_y = E[y]$

↳ for 2D dataset, we can get

$Var[x]$, $Var[y]$, $cov[x,y]$, $cov[y,x]$ } →

$$\begin{bmatrix} Var[x] & cov[x,y] \\ cov[y,x] & Var[y] \end{bmatrix}$$

Covariance matrix

if $cov[x,y] > 0$, on avg y increase if we increase x

$= 0$, x and y has nothing to do with each other

↳ always a symmetric positive definite matrix

if $D = \{x_1, \dots x_N\}$ ← D dimension, $x_i \in \mathbb{R}^D$

→ Covariance matrix $D \times D$

$$Var[D] = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T$$

Linear transformation: effect on means

e.g. $D = \{-1, 2, 3\}$

$E[D] = 4/3$

$D' = \{1, 4, 5\} = D + 2$

$E[D'] = 10/3 = E[D] + 2$

* Data shift by a constant

$E[D + a] = a + E[D]$, a is constant

$D'' = \{-2, 4, 6\} = D \times 2$

$E[D''] = 8/3 = E[D] \times 2$

* Data scaled by a factor

$E[\alpha D] = \alpha E[D]$

$\Rightarrow E[\alpha D + a] = \alpha E[D] + a$

α: scale factor; a: shift

Linear Transformation: effect on (co) variance

* Variance remains the same if data is shifted by a constant

$Var[D] = Var[D + a]$

a: offset on every element

$$Var[\alpha D] = \frac{1}{N}\sum_{i=1}^{N}(\alpha x_i - \alpha\mu)(\alpha x_i - \alpha\mu)^T$$

$$= \frac{1}{N}\sum_{i=1}^{N}\alpha^2(x_i - \mu)(x_i - \mu)^T$$

$$= \alpha^2 Var[D]$$

$\Rightarrow Var[\alpha D + a] = \alpha^2 Var[D]$

α: scale factor; a: shift

$$Var[aD] = a^2 Var[D]$$

High Dimension

$D = \{x_1, \ldots, x_N\}$ $x_i \in \mathbb{R}^p$ ← p dimension for each element

variance of this high dimension dataset is given by covariance matrix

If $Ax_i + b$ on every data

- A: linear transform matrix
- b: offset vector

$$Var[AD + b] = A\,Var[D]\,A^T$$

- $Var[AD+b]$: covariance matrix
- $Var[D]$: $p \times p$ covariance matrix

Dot product

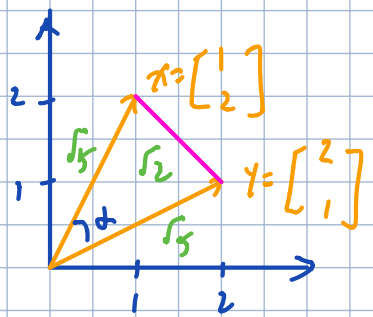
If $\vec{x}$ and $\vec{y}$ are 2 vectors, then the dot product is defined:

$$x^T y = \sum_{i=1}^{N} x_i y_i \qquad x, y \in \mathbb{R}^N \quad (N \text{ dimensional vectors})$$

length of $\vec{x}$ is then defined as square root of the dot product of $\vec{x}$ with itself:

$$\|x\| = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{N} x_i^2}$$

e.g.



$$\|x\| = \sqrt{1^2 + 2^2} = \sqrt{5}$$
$$\|y\| = \sqrt{2^2 + 1^2} = \sqrt{5}$$

distance between 2 vectors: length of the difference vector

$$d(x,y) = \|x - y\| = \sqrt{(x-y)^T (x-y)}$$

so $x - y = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \Rightarrow d(x,y) = \sqrt{(-1)^2 + 1^2} = \sqrt{2}$

$$\cos\alpha = \frac{x^T y}{\|x\| \|y\|}$$

e.g. $\cos\alpha = \frac{(1\ 2)\begin{pmatrix} 2 \\ 1 \end{pmatrix}}{\sqrt{5} \cdot \sqrt{5}} = 4/5$

$\alpha \approx 0.64$ rad

Inner Product

definition

$\vec{x}, \vec{y} \in V$ (vector space V)

symmetric, positive definite, bilinear mapping

$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$

Bilinear:

$x, y, z \in V, \quad \lambda \in \mathbb{R}$

$\langle \lambda x + z, y \rangle = \lambda \langle x, y \rangle + \langle z, y \rangle$

$\langle x, \lambda y + z \rangle = \lambda \langle x, y \rangle + \langle x, z \rangle$

Positive definite

$\langle x, x \rangle \geq 0 \quad \langle x, x \rangle = 0$ iff $\vec{x} = 0$

Symmetry

$\langle x, y \rangle = \langle y, x \rangle$

e.g. in $\mathbb{R}^2$

$\langle x, y \rangle = x^T I y$ ← dot product

If we define $\langle x, y \rangle = x^T A y$, $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

↳ $2x_1y_1 + x_2y_1 + x_1y_2 + 2x_2y_2$

Any A that is symmetric, positive definite matrix defines a valid inner product.

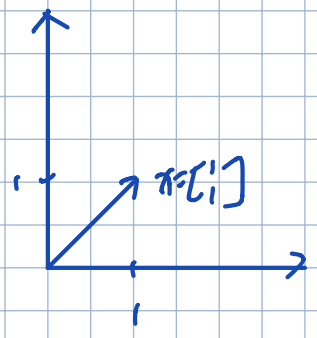
Inner product: lengths

$$\|x\| = \sqrt{\langle x, x \rangle}$$

depending on the choice of inner product
↳ geometry of vector space can be different

e.g.

$$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



standard dot product

$\langle x, y \rangle = x^T y \Rightarrow \|x\| = \sqrt{2}$

$$\langle x, y \rangle = x^T \begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix} y$$

$$= x_1y_1 - \tfrac{1}{2}(x_1y_2 + x_2y_1) + x_2y_2$$

$$\Rightarrow \|x\| = \sqrt{1 - 1 + 1} = 1$$

$\|x\|^2 = \langle x, x \rangle = 1$ ← same

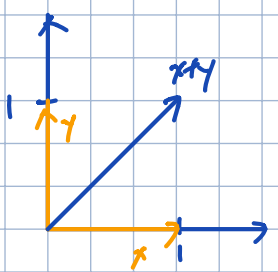
$\Rightarrow \|x\| = 1$

* stretching

$$\|\lambda x\| = |\lambda| \, \|x\|$$

* triangle inequality

$$\|x + y\| \leq \|x\| + \|y\|$$



$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Standard dot : $\|x\| = 1 = \|y\|$

$\|x+y\| = \sqrt{2}$

Cauchy – Schwarz inequality

$|\langle x, y\rangle| \leq \|x\| \|y\|$

Inner product. distance between vectors

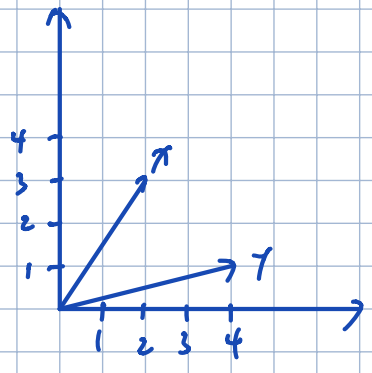
$$d(x,y) = \|x-y\| = \sqrt{\langle x-y, x-y\rangle}$$

if we use a dot product definition, then the distance is called the Euclidean distance.

e.g. $x = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$y = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$

$x-y = \begin{bmatrix} 2-4 \\ 3-1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$



define inner product

① Dot product

$\|x-y\| = \sqrt{(-2)^2 + 2^2} = \sqrt{8}$

② $\langle x, y\rangle = x^T \begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix} y$

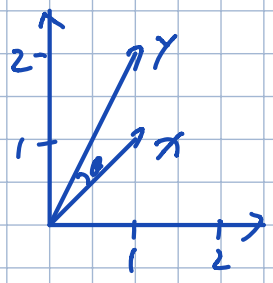
$\|x-y\| = \sqrt{12}$

Inner product: angles / orthogonality

$$\cos w = \frac{\langle x, y\rangle}{\|x\| \|y\|}$$

w : angle between 2 vectors

e.g. $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$



① dot product as inner product

$$\cos w = \frac{x^T y}{\sqrt{x^T x}\sqrt{y^T y}}$$

$$= 3/\sqrt{10}$$

$w \approx 0.32$ rad $\approx 18°$

e.g. $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

dot product as inner product

$\cos w = 0 \Rightarrow w = \pi/2$ rad $= 90°$ ← orthogonal

2 non-zero vectors are orthogonal iff their inner product $= 0$

if we choose another inner product

$$\langle x, y \rangle = x^T \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} y \Rightarrow \langle x, y \rangle = -1$$

↑ not orthogonal

$$\cos w = \frac{-1}{\sqrt{3}\sqrt{3}} = 1.9 \text{ rad}$$

$\|x\| = \sqrt{\langle x, x \rangle} \Rightarrow \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$= 3$

$\Rightarrow \sqrt{3}$

$\|y\| = \begin{bmatrix} -1, & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

$= 2 + 1 = 3 \Rightarrow \sqrt{3}$

Inner product of functions

u, v are functions

$$\langle u, v \rangle = \int_a^b u(x) v(x)\, dx$$

e.g. $u(x) = \sin(x)$

$v(x) = \cos(x)$

$f(x) = u(x) v(x) \Rightarrow$ odd function

$\int_{-\pi}^{\pi} f(x) = 0$ i.e. $u(x)$ and $v(x)$ are orthogonal

Inner product of random variable

$\Rightarrow \mathrm{Var}[X+Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$

(random variable)

if we define

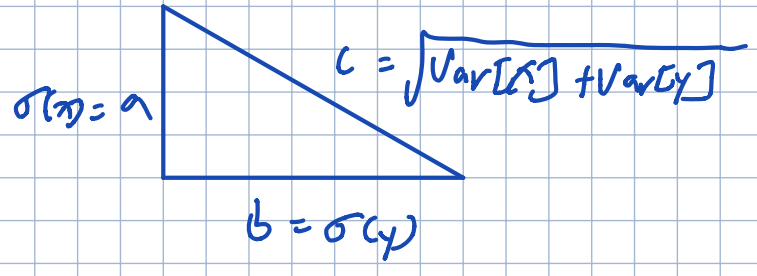
$\langle X, Y \rangle = \mathrm{cov}[X, Y]$

$\mathrm{cov}[\lambda X + Y, Z] = \lambda\, \mathrm{cov}[X, Z] + \mathrm{cov}[Y, Z]$

$\|X\| = \sqrt{\mathrm{cov}[X, X]} = \sqrt{\mathrm{Var}[X]} = \sigma(X)$

angle between 2 random variable

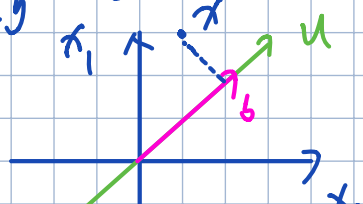
$$\cos\theta = \frac{\langle X, Y \rangle}{\|X\| \|Y\|} = \frac{\mathrm{cov}[X, Y]}{\sqrt{\mathrm{Var}[X]\,\mathrm{Var}[Y]}}$$



---

orthogonal projections onto 1D subspace

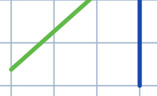
e.g.



$x = \lambda r_1 + \beta r_2$

2D space $\mathbb{R}^2$ basis vectors of $\mathbb{R}^2$: r_1, r_2

1D space U : basis vector b



2D space

↑   ↗ 1D space.

$b = x\,u \qquad u \in U$

orthogonal projection of x onto U:

$\pi_U(x)$

2 important properties

① ∵ $\pi_U(x)$ is in U, ⇒ there exist a λ in $\mathbb{R}$ such that

$\pi_U(x) = \lambda b$

↳ b: basis vector that spans U

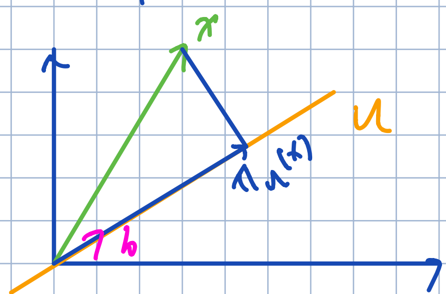
↳ λ: coordinate of the projection w.r.t. to the basis b of the subspace U.

② The difference vector of x and its projection onto U is orthogonal to U.

↳ i.e. it is orthogonal to the basis vector that spans U

$\langle b, \pi_U(x) - x \rangle = 0$

Generally hold for any x in $\mathbb{R}^D$ and 1-D subspace U



e.g. we want to find $\pi_U(x)$

① $\pi_U(x) = \lambda b$

② $\langle b, \pi_U(x) - x \rangle = 0$

$\langle b, \pi_U(x) - x \rangle = 0$

$\Rightarrow \langle b, \pi_U(x) \rangle - \langle b, x \rangle = 0$

$\Rightarrow \langle b, \lambda b \rangle - \langle b, x \rangle = 0$

$\Rightarrow \lambda \langle b, b \rangle - \langle b, x \rangle = 0$

($\langle b, b \rangle$ ↓ $\|b\|^2$)

$\Rightarrow \lambda = \dfrac{\langle b, x \rangle}{\|b\|^2}$

$$\Rightarrow \pi_u(x) = \lambda b = \frac{\langle b, x \rangle}{\|b\|^2} b$$

If we use dot product as the inner product,

$$\pi_u(x) = \frac{b^T x}{\|b\|^2} b = \frac{b b^T}{\|b\|^2} x$$

($b^T x$: scalar)

$\left(\frac{x^T b}{\|b\|^2} b\right)$

$\frac{b b^T}{\|b\|^2}$: projection matrix from any point in 2D → 1D ← square and symmetric

if $\|b\| = 1$, $\Rightarrow \lambda = b^T x$

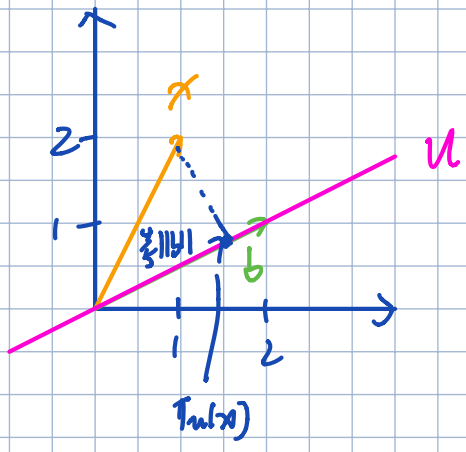
$\pi_u(x) = b b^T x$

the projection $\pi_u(x)$ is still a vector in $\mathbb{R}^D$

but we no longer need D coordinates to represent it, only need 1 (but loss some info)

---

e.g. $b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

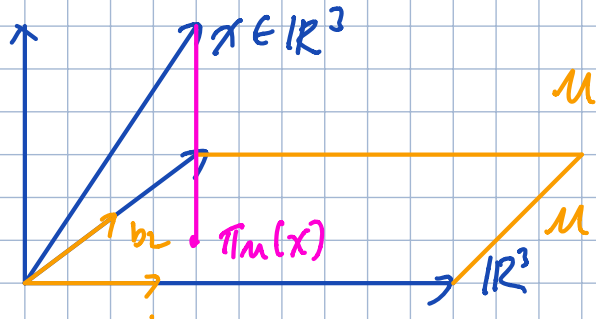


$$\pi_u(x) = \frac{x^T b}{\|b\|} b$$

$$= \frac{2+2}{5} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$= \frac{4}{5} \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Projection into higher-dimensional subspaces



$u = [b_1, b_2]$

orthogonal projection onto subspace u

b_1

① $\pi_u(x) = \lambda_1 b_1 + \lambda_2 b_2$ → generalize to → $\pi_u(x) = \sum_{i=1}^{M} \lambda_i b_i$

② $\langle x - \pi_u(x), b_1 \rangle = 0$
$\langle x - \pi_u(x), b_2 \rangle = 0$
→ D-dim proj to M-dim → $\langle \pi_u(x) - x, b_i \rangle = 0 \quad i = 1, \dots, M$

we need to find

$$\lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_M \end{bmatrix}_{M \times 1} \qquad B = [b_1 | \dots | b_M]_{D \times M}$$

$\pi_u(x) = B\lambda$

Assume we use dot product as inner product

$\langle \pi_u(x) - x, b_i \rangle = \langle B\lambda - x, b_i \rangle = 0$

$\Rightarrow \langle B\lambda, b_i \rangle - \langle x, b_i \rangle = 0, \quad i = 1 \dots M$

$\Rightarrow \lambda^T B^T b_i - x^T b_i = 0 \quad i = 1 \dots M$

(summarise)

$\Rightarrow \lambda^T B^T B - x^T B = 0$

$\Rightarrow \lambda^T = x^T B (B^T B)^{-1}$ ($B^T B$ through eqv)

$\Rightarrow \lambda = (B^T B)^{-1} B^T x$

↳ $\pi_u(x) = B\lambda = \underbrace{B (B^T B)^{-1} B^T}_{\text{proj matrix}} x$

if B is orthonormal: $B^T B = I$

↳ $\pi_u(x) = B B^T x$

---

e.g. projection onto a 2D subspace

$x = \begin{bmatrix} z \\ \vdots \end{bmatrix}$

$b_1 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$ $b_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

$M = [b_1, b_2]$

$\pi_M(x) = B\lambda$

$B = [b_1 | b_2]$

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

$\lambda = (B^T B)^{-1} B^T x$

$B^T x = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$

$B^T B = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$

$B^T B \lambda = B^T x$

$\Rightarrow \lambda = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$

$\Rightarrow \pi_M(x) = -1 b_1 + 3 b_2 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$

PCA: high level idea

$X = \{x_1, \dots x_N\}$, where $x_i \in \mathbb{R}^D$

↑ data set

we want to find low level representations that is as similar as poss. ble

① $x_n = \sum_{i=1}^{D} \beta_{in} b_i$

(β_{in}: coefficient; b_i: orthonormal)

basis of $\mathbb{R}^D$

assume we use the dot product

② ↳ $\beta_{in} = x_n^T b_i$ ⇒ i.e. β_{in} to be the orthogonal projection of x_n onto the 1D subspace spanned by the i^{th} basis vector

③ $B = (b_1, \dots b_M)$ i.e. a matrix that consists of these orthonormal basis vectors

then the projection of x onto the subspace

↳ $\tilde{x} = BB^T x$ coordinates of $\tilde{x}$ w.r.t. the basis vectors collected in matrix B ⇒ Code

↓

orthogonal projection of x onto the subspace spanned by the M basis vectors

Key idea in PCA is to find a lower dimension representation $\tilde{x}_n$ of x_n that can be expressed using fewer basis vectors (from $D \to M$ dimension)

✠ we assume the data is centered, i.e. $\mathbb{E}[x] = 0$

↑

dataset has mean = 0

✠ assume, $b_1 \dots b_D$ are orthonormal basis of $\mathbb{R}^D$

In general we can write

$$\tilde{x}_n = \sum_{i=1}^{M} \beta_{in} b_i + \sum_{i=M+1}^{D} \beta_{in} b_i \quad \in \mathbb{R}^D$$

$\tilde{x}_n$: 1 datapoint in X dataset

$\sum_{i=1}^{M} \beta_{in} b_i$: principal subspace

$\sum_{i=M+1}^{D} \beta_{in} b_i$: orthogonal complement ← ignore in PCA

$b_1, \dots b_M$ span the principal subspace

$\tilde{x}$ is still a D-dimensional vector, it lives in an M-dimensional subspace of $\mathbb{R}^D$ and only M coordinates $b_1 \dots b_M$ are necessary to represent it.

Setting. Given X dataset find β_{in} and b_i such that avg sq reconstruct error is minimized.

$$J = \frac{1}{N}\sum_{n=1}^{N} \| x_n - \tilde{x}_n \|^2$$

approach:

compute partial derivatives of J w.r.t. the parameters β_{in}, b_i (i.e. $\frac{\partial J}{\partial \beta_{in}} = 0$ $\frac{\partial J}{\partial b_i} = 0$)

notice that these 2 parameters only enter this loss function through $\tilde{x}_n$. $\Rightarrow$ we need chain rule for that

$$\frac{\partial J}{\partial \{\beta_{in}, b_i\}} = \boxed{\frac{\partial J}{\partial \tilde{x}_n}} \frac{\partial \tilde{x}_n}{\partial \{\beta_{in}, b_i\}}$$

$$\frac{\partial J}{\partial \tilde{x}_n} = -\frac{2}{N}(x_n - \tilde{x}_n)^T$$

Determine coordinates of the projected data

assumption: centered data: $E[x] = 0$, $b_1, \dots b_M$ orthonormal basis $\leftarrow$ M subspace.

$$\frac{\partial J}{\partial \beta_{in}} = \frac{\partial J}{\partial \tilde{x}_n} \cdot \boxed{\frac{\partial \tilde{x}_n}{\partial \beta_{in}}}$$

$$\frac{\partial \tilde{x}_n}{\partial \beta_{in}} = b_i \quad , i = 1, \dots, M$$

$$\frac{\partial J}{\partial \beta_{in}} = -\frac{2}{N}(x_n - \tilde{x}_n)^T b_i$$

$$= -\frac{2}{N}\left(x_n - \sum_{j=1}^{M} \beta_{jn} b_j\right)^T b_i$$

$$= -\frac{2}{N}\left(x_n^T b_i - \beta_{in}\right) \quad (i \text{ orthonormal})$$

$$\tilde{x}_n = \sum_{j=1}^{M} \beta_{jn} b_j \quad (A)$$

$$J = \frac{1}{N}\sum_{n=1}^{N} \| x_n - \tilde{x}_n \|^2 \quad (B)$$

$$\frac{\partial J}{\partial \tilde{x}_n} = -\frac{2}{N}(x_n - \tilde{x}_n) \quad (C)$$

$N\left(x_n b_i - \beta_{in} b_i b_i\right)$ (orthonormal basis)

$b_j^T b_i = \delta_{ij}$

$= -\frac{2}{N}\left(x_n^T b_i - \beta_{in}\right)$

set to 0 $\Rightarrow$

↳ iff $\beta_{in} = x_n^T b_i \quad i = 1 \ldots M$ (D)

i.e. optimal coord of $\tilde{x}_n$ w.r.t our basis, are the orthogonal projections of the coord. of the original data points onto i^{th} basis $(b_{1 \ldots M})$ vector that spans the principal subspace

reduced

---

rephrase on the loss function

$$\tilde{x}_n = \sum_{j=1}^{M} \beta_{jn} b_j \quad \text{by A}$$

$$= \sum_{j=1}^{M} \underbrace{(x_n^T b_j)}_{\text{scalar (or) dot prod}} b_j \quad \text{by D}$$

$$= \sum_{j=1}^{M} b_j (b_j^T x_n) = \underbrace{\left(\sum_{j=1}^{M} b_j b_j^T\right)}_{\text{projection matrix}} x_n$$

Similarly

$$x_n = \left(\sum_{j=1}^{M} b_j b_j^T\right) x_n \to \hat{x}_n$$

$$+ \left(\sum_{j=M+1}^{D} b_j b_j^T\right) x_n \to \text{missing}$$

i.e. the difference vector x_n and $\tilde{x}_n$ is

$$x_n - \tilde{x}_n = \left(\sum_{j=M+1}^{D} b_j b_j^T\right) x_n$$

$$= \sum_{j=M+1}^{D} \left(x_n^T b_j\right) b_j$$

$$= \sum_{j=M+1}^{D} (b_j^T x_n) b_j \quad (E)$$

From B, we get

$$J = \frac{1}{N} \sum_{n=1}^{N} \| x_n - \tilde{x}_n \|^2$$

$$= \frac{1}{N} \sum_{n=1}^{N} \| \sum_{j=M+1}^{D} (b_j^T x_n) b_j \|^2 \quad \text{by } E$$

$\because$ orthonormal basis $b_i b_j$

$$= \frac{1}{N} \sum_{n=1}^{N} \sum_{j=M+1}^{D} (b_j^T x_n)^2$$

$$= \frac{1}{N} \sum_{n} \sum_{j} b_j^T x_n x_n^T b_j \quad \text{(same)}$$

rearrange

$$= \sum_{j=M+1}^{D} b_j^T \left(\frac{1}{N} \sum_{n=1}^{N} x_n x_n^T \right) b_j$$

S: data covariance matrix S

$$= \sum_{j=M+1}^{D} b_j^T S b_j \quad (F)$$

$$= \text{trace}\left(\left(\sum_{j=M+1}^{D} b_j b_j^T \right) S \right)$$

projection matrix

↙ takes S to the orthogonal complement of the principal subspace

⇒ formulate J as the variance of the data onto the subspace we ignore.

Determine the basis vectors that span the principal subspace

$$J = \sum_{j=M+1}^{D} b_j^T S b_j$$

S: data covariance matrix

$J = \sum_{j=M+1}$ ...

minimize this ($\partial J / \partial b_j$) need us to find the orthonormal basis that spans the space we ignore ⇒ after that, we take orthogonal complement as the basis of the principal subspace

e.g. for 2D space → 1D

b_1, b_2

b_1 spans principal subspace

b_2 orthogonal complement

$$b_i^T b_j = \delta_{ij}$$

$$J = b_2^T S b_2 \quad , \quad b_2^T b_2 = 1$$

$$L = b_2^T S b_2 + \lambda(1 - b_2^T b_2)$$

L: Lagrangian; λ: Lagrange multiplier

$$\frac{\partial L}{\partial \lambda} = 1 - b_2^T b_2 = 0 \iff b_2^T b_2 = 1$$

$$\frac{\partial L}{\partial b_2} = 2 b_2^T S - 2\lambda b_2^T = 0$$

$$\iff S b_2 = \lambda b_2$$

eigenvalue problem

So $J = b_2^T \underline{S b_2} = \underline{b_2^T b_2} \lambda = \lambda$ ← should smallest eigenvalue of S

i.e. need to choose b_2 as the corresponding eigenvector span the subspace we ignore

b_1 which spans the principal subspace is the eigenvector with larger eigenvalue of S

General case:

If we wants to find the M dimensional principal subspace of a D- dimensional data set and we solve for the basis vectors b_j

$$b_j \;, j = M+1, \dots D$$

$$S b_j = \lambda_j b_j, \quad j = M+1, \dots, D$$

loss function is the sum of the corresponding eigenvalue

$$J = \sum_{j=M+1}^{D} \lambda_j$$

$\Downarrow$

principal subspace is spanned by the eigenvectors belong to the M largest eigenvalues of the data covariance matrix

↳ eigenvectors are orthogonal to each other (sym)

eigenvector belong to the largest eigenvalue points towards the direction of data with largest variance.

↳ the variance in that direction is given by the eigenvalue.

Steps of PCA

centering data is not necessary but easier for computation
$\Rightarrow S$ will be large if not centered.

also divide every dimension of the centered data by the corresponding standard deviation $\Rightarrow$ data $\rightarrow$ unit free

→ variance in every dimension = 1
(no change in correlation)

① Subtract the mean of data (centered)

② divide by standard deviation (for each dimension)

③ get S → eigen value
→ eigen vector

④ we can project any data point onto the principal subspace

standardize data: $x_*^{(d)} \leftarrow \frac{x_*^{(d)} - \mu^{(d)}}{\sigma^{(d)}}$ d: dimension x^* a data point

projection $\tilde{x}_* = \pi_U(x_*) = BB^T x_*$

B: matrix that contains the eigenvectors that belongs to the largest eigenvalue as columns

$B^T x_*$: coord of the projections w.r.t the basis of the principal subspace.

PCA in high dimension

In D dimension, data covariance matrix is $D \times D$ matrix

If D is big, compute eigen vector/value is expensive.

If we have data set

$\{x_1, \dots x_N\} \in \mathbb{R}^D$

$S = \frac{1}{N} X^T X$ (standardized)

$X = \begin{bmatrix} x_1^T \\ x_N^T \end{bmatrix}$ $\mathbb{R}^{N \times D}$

if $N \ll D$ rank $(S) = N$

$S b_i = \lambda_i b_i$

$\underbrace{\frac{1}{N} X^T X}_{S} b_i = \lambda_i b_i$

$\underbrace{\frac{1}{N} X X^T}_{N \times N \text{ matrix}} \underbrace{X b_i}_{c_i} = \lambda_i \underbrace{X b_i}_{c_i}$

i.e. from $D \times D \rightarrow N \times N$

↳ from here, recover eigenvectors of S

$\underbrace{\frac{1}{N} X^T X}_{S} X^T c_i = \lambda_i X^T c_i$