Lecture 19 | Summary of PCA

1) [optional] Normalize the data (usually per-variable separately)

2) Centering the data: $A_c = A - \vec{\mu}$ ← column means

3) Compute covariance matrix Σ

$$\Sigma = \frac{1}{(N-1)} \cdot (A_c.T) @ (A_c)$$

Shape: (M, M)

4) Compute eigenVectors (p) and eigenValues using numpy:

```
e_vals, e_vecs = np.linalg.eig(Σ)
```

e_vecs: shape (M, M), p

e_vals: vector: len $= (M,)$

5) Sort e_vals high → low.

PC1 PC2 PC3

e.g. $[3.2, 1.1, 0.03] = e\_vals$ $M=3$

e_vecs

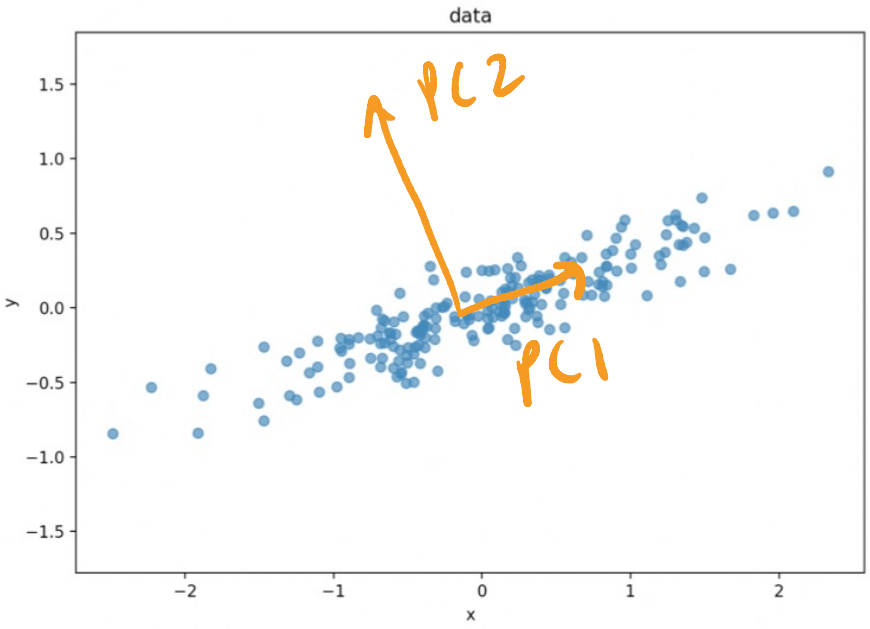
☆ Sort Columns of P in the same order that you sorted the eigenvalues.

$$A = \begin{bmatrix} x & y & z \\ | & | & | \\ | & | & | \end{bmatrix}$$

$$P = \begin{bmatrix} PC1 & PC2 & PC3 \\ | & | & | \\ | & | & | \end{bmatrix}$$

after sorting according to e_vals.

problem: if don't sort e_vecs like e_vals



don't want this!!

6a) Compute prop-var accounted for by each PC:

$$\text{total} = 3.2 + 1.1 + 0.03 = 4.33$$

$$\text{prop-var} = \left[\frac{3.2}{4.33}, \frac{1.1}{4.33}, \frac{0.03}{4.33} \right]$$

6b) Compute <u>cumulative</u> prop-var accounted by by 1st K eigenvalues/PCs.

$$\text{Cum-prop-var} = \left[\underbrace{\frac{3.2}{4.33}}_{PC1}, \underbrace{\frac{(3.2+1.1)}{4.33}}_{PC1+PC2}, \underbrace{\frac{(3.2+1.1+0.03)}{4.33}}_{PC1+PC2+PC3} \right]$$

always = 1 (100%)

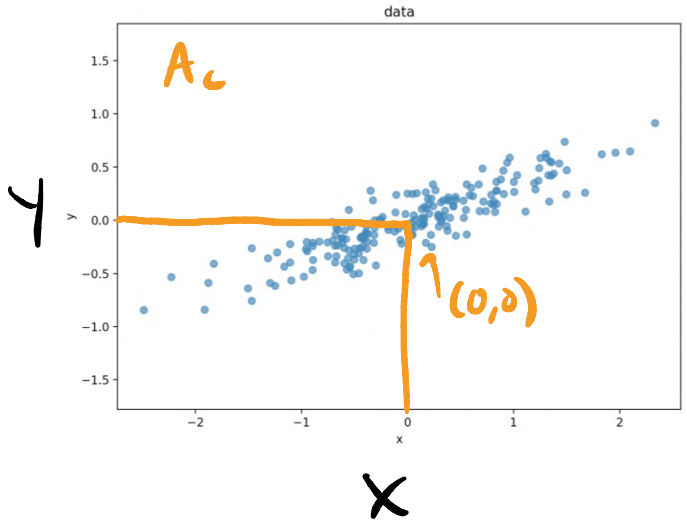
6c) Threshold/cut off point: Want to keep

<u>at least</u> X% variance: (90%)

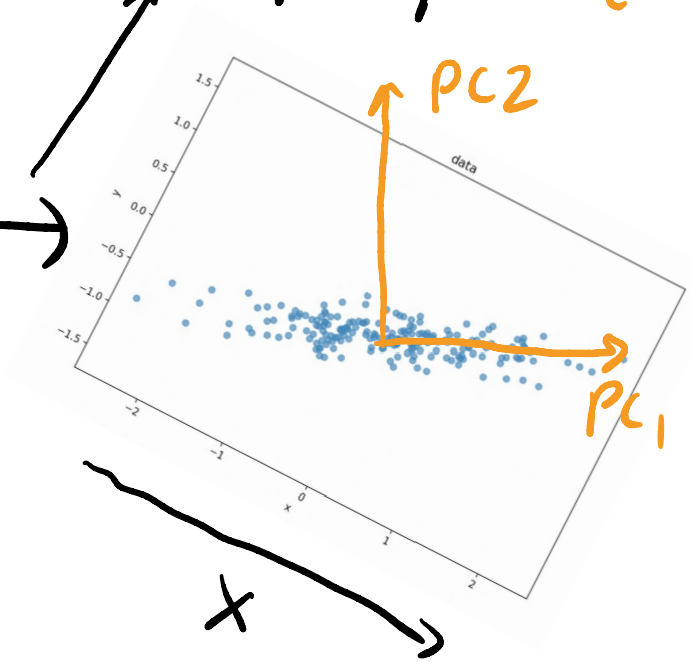
⇒ we ultimately keep K PCs

7) Project data from data space → PCA space

data space



PCA space: $\hat{A}_c$



$$\hat{A}_c = A_c \,@\, \hat{P}$$

(N, K) ← (N, M) ← (M, K)

$\hat{P}$: eigenvectors: (M, M)

P: PC1 PC2 PC3 PC4 — Cut off

Keep 2 PCs | drop PC3 + PC4

K | M

Keep k PCs: $\hat{P}$

$$\hat{P} = P[:, :k]$$

★ Reduced dimension

$M \rightarrow k$

⇒ Could be done

8) [optional] Reconstruction of data

PCA Space → data space

if we did not Normalize:

$$A_{reconstruct} = \underbrace{\hat{A}_c}_{(N,K)} @ \underbrace{\hat{P}.T}_{(K,M)\text{ transpose}} + \underbrace{\vec{\mu}}_{(1,M)}$$

(N,M) $\quad$ (N,M)

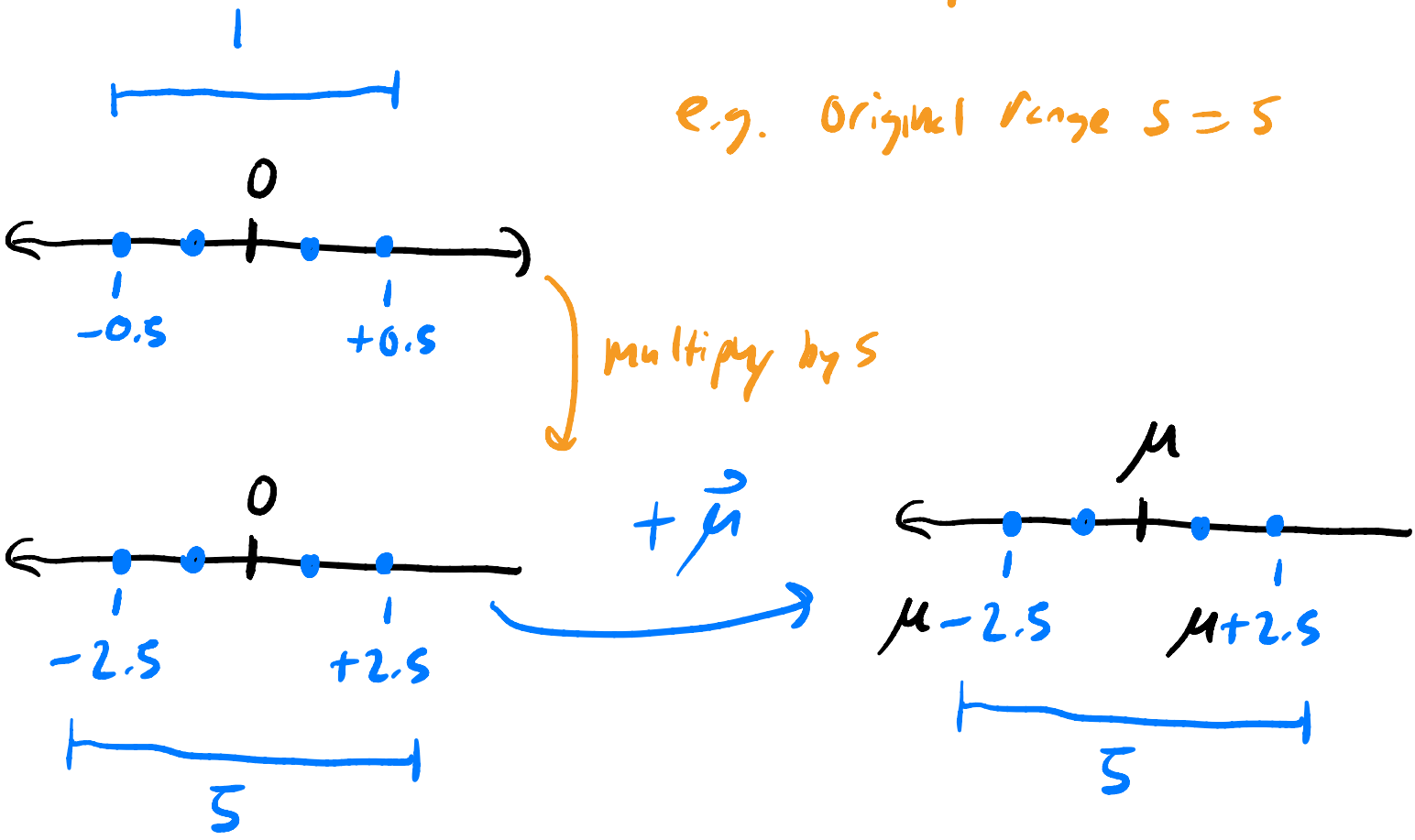
(N,M)

Keep N samples,
return to M data vars
(original ones)

if we normalized:

original data range in each var

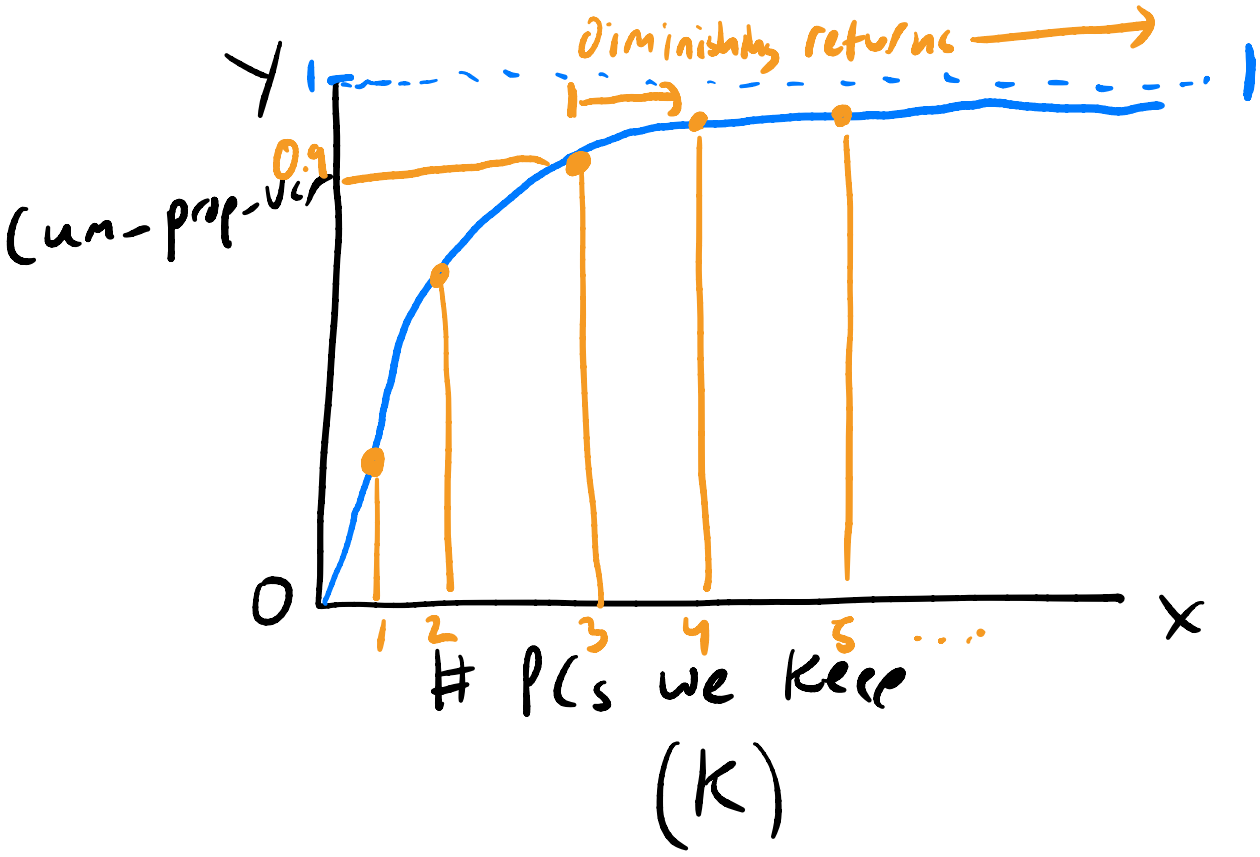
$$A_{reconstruct} = \vec{s} \cdot (\hat{A}_c @ \hat{P}.T) + \vec{\mu}$$



Elbow plot: useful if have large # variables (e.g $\mu = 1000$)

Plot Cumulative prop var (y axis)

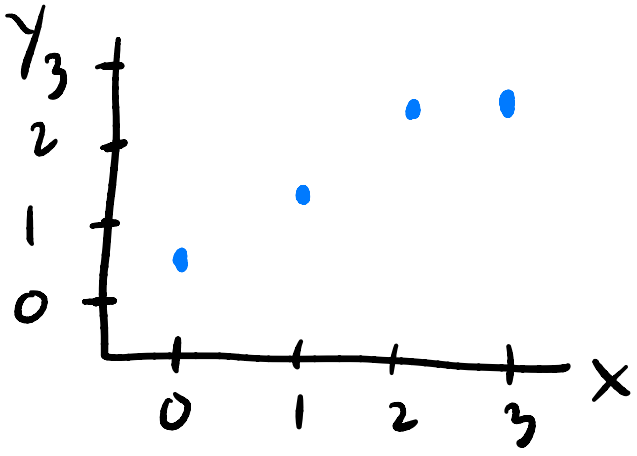
PC # on X axis



⇒ 90% is a good cut off point: keep PC1, PC2, PC3, drop PC 4+

Example:

X	Y
0	0.5
1	1.1
2	2.3
3	2.5



1) Normalize → Skip.

2) Center data : $A_c = A - \vec{\mu}$

$\vec{\mu} = (\mu_x, \mu_y)$ $\quad \mu_x = \frac{0+1+2+3}{4} = 1.5$

$\mu_y = \frac{0.5+1.1+2.3+2.5}{4} = 1.6$

X	Y
0	0.5
1	1.1
2	2.3
3	2.5

A

$-[1.5, 1.6]$ ← $\vec{\mu}$

X_c	Y_c
-1.5	-1.1
-0.5	-0.5
0.5	0.7
1.5	0.9

A_c

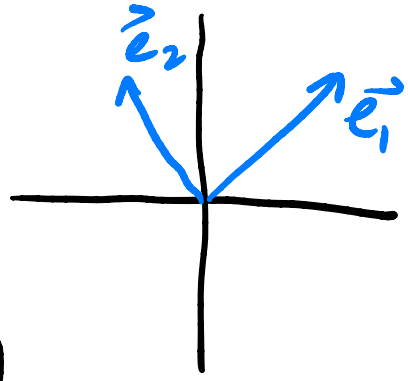
3) Compute $\Sigma = \frac{1}{(N-1)} \cdot (A_c \cdot T) @ (A_c)$

$$\frac{1}{3}\begin{bmatrix} -1.5 & -0.5 & 0.5 & 1.5 \\ -1.1 & -0.5 & 0.7 & 0.9 \end{bmatrix} @ \begin{bmatrix} -1.5 & -1.1 \\ -0.5 & -0.5 \\ 0.5 & 0.7 \\ 1.5 & 0.9 \end{bmatrix}$$

$$= \begin{bmatrix} 1.67 & 1.2 \\ 1.2 & 0.92 \end{bmatrix} \Sigma$$

4) e_vals, e_vecs of Σ

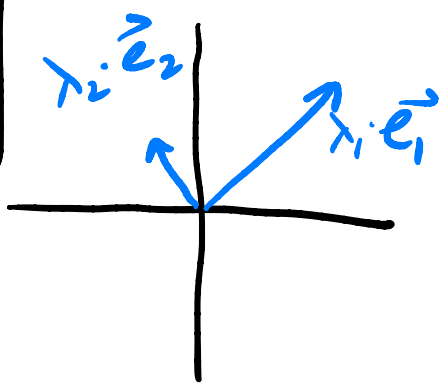
eigen values: $\underbrace{2.55}_{\lambda_1}, \underbrace{0.04}_{\lambda_2}$



eigen vectors (P):

$$\begin{bmatrix} 0.81 & -0.59 \\ 0.51 & 0.81 \end{bmatrix}$$

$\vec{e}_1$ $\vec{e}_2$



5) Sort e_vecs / e_vals high → low according to e_vals