PCA


Have patience. All things are difficult before they become easy
Wisdomword.in

Agenda:-

→ Variants of G.D

SGD

Mini Batch G.D

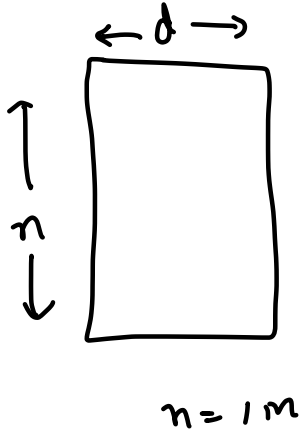
Batch G.D

→ Dim^n. Reduction

→ PCA

17 Gradient Descent

$$w = w - \eta \cdot \frac{\partial L}{\partial w}$$



$$\frac{\partial L}{\partial \bar{w}} = -\frac{1}{n} \sum_{i=1}^{n} y^i \cdot x^i + \frac{\lambda \bar{w}}{\|\bar{w}\|}$$

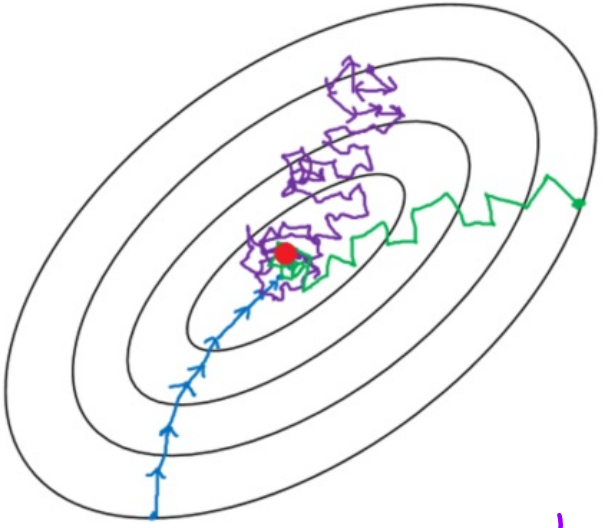
→ slow updates in 'w' → Batch Gradient Descent

Why? n = 1M

→ Calculate $\nabla_w L$ on subset of data Batchsize

$m = 500$ mini-Batch G.D

if $m = 1$ → Stochastic G.D



Batch gradient descent
Mini-batch gradient Descent
Stochastic gradient descent

epoch →

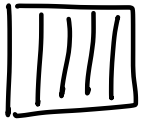
↓

"10 epoch"

Dim^n Reduction

features

#dim = #features



100 dim^n
(features)

Extract →

10 dim^n
(create new features)

→ max. info explained.
→ feature extraction
→ loss info. ↓

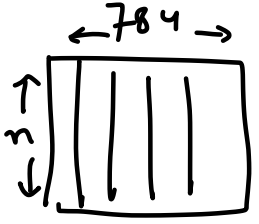
Why ?

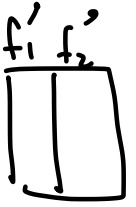
① Curse of Dim^n (d↑↑↑↑)

→ training ML Slow

② Viz

2D/3D →





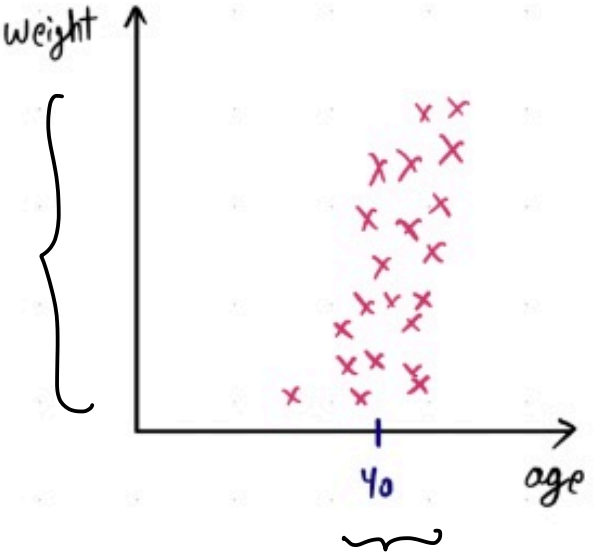
. scatter( )

Diabetes prediction

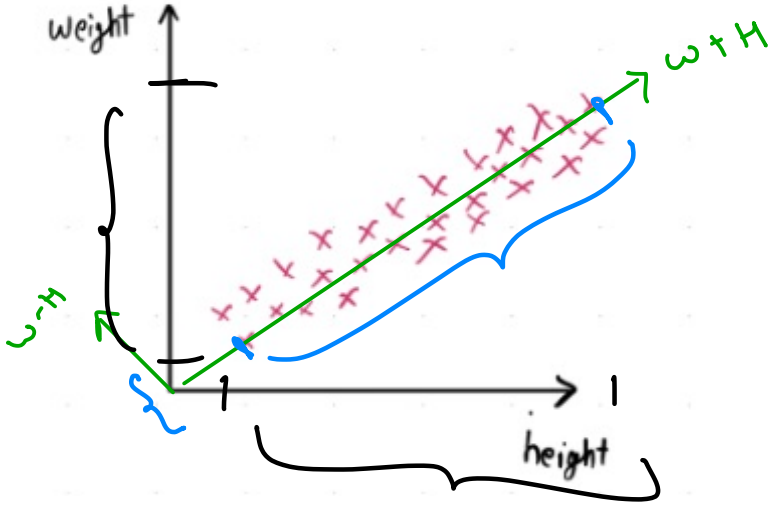
Weight	Age	Diabetic
⋮	⋮	Y
		N
		Y
		Y
		⋮

f_1 = weight

f_2 = Age



more spreadness = more info.



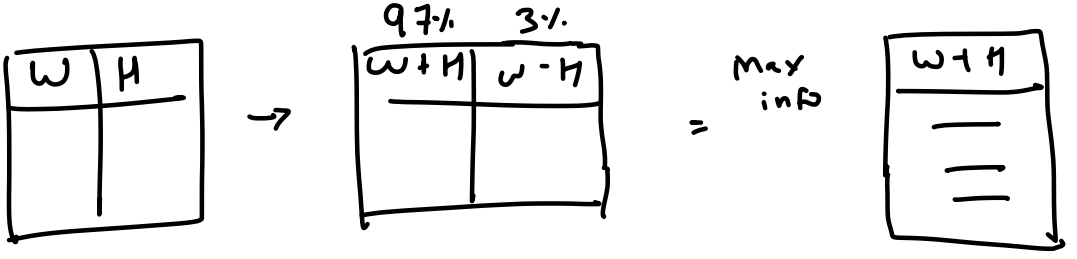
Which feature ?

a) Weight

b) Height

c) weight + weight

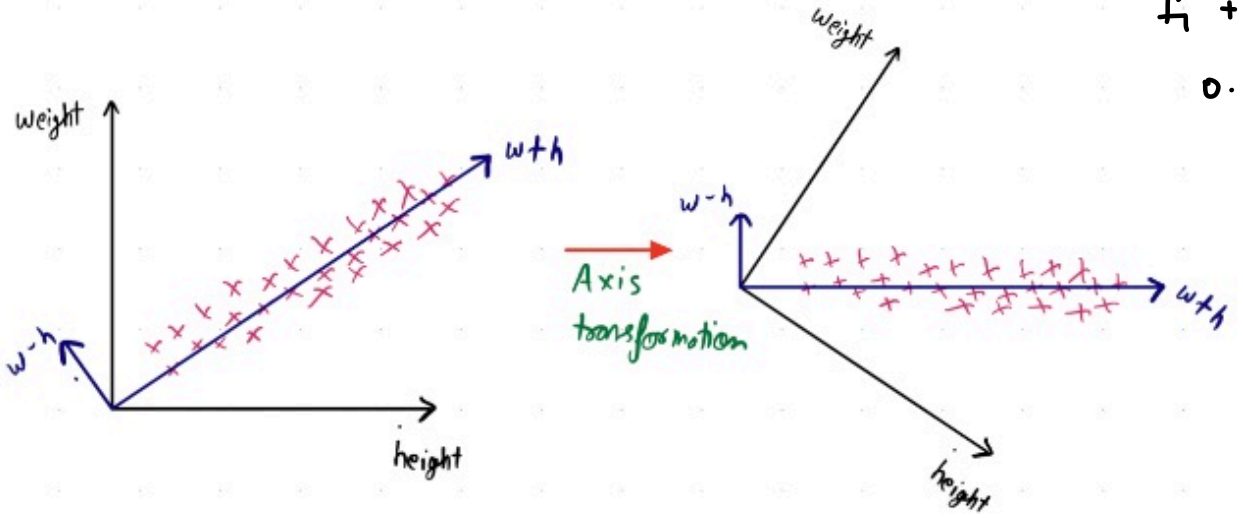
d) weight - height

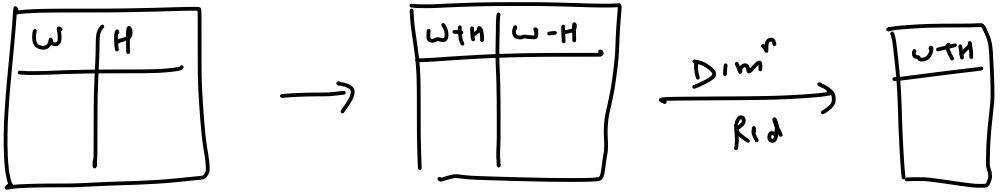


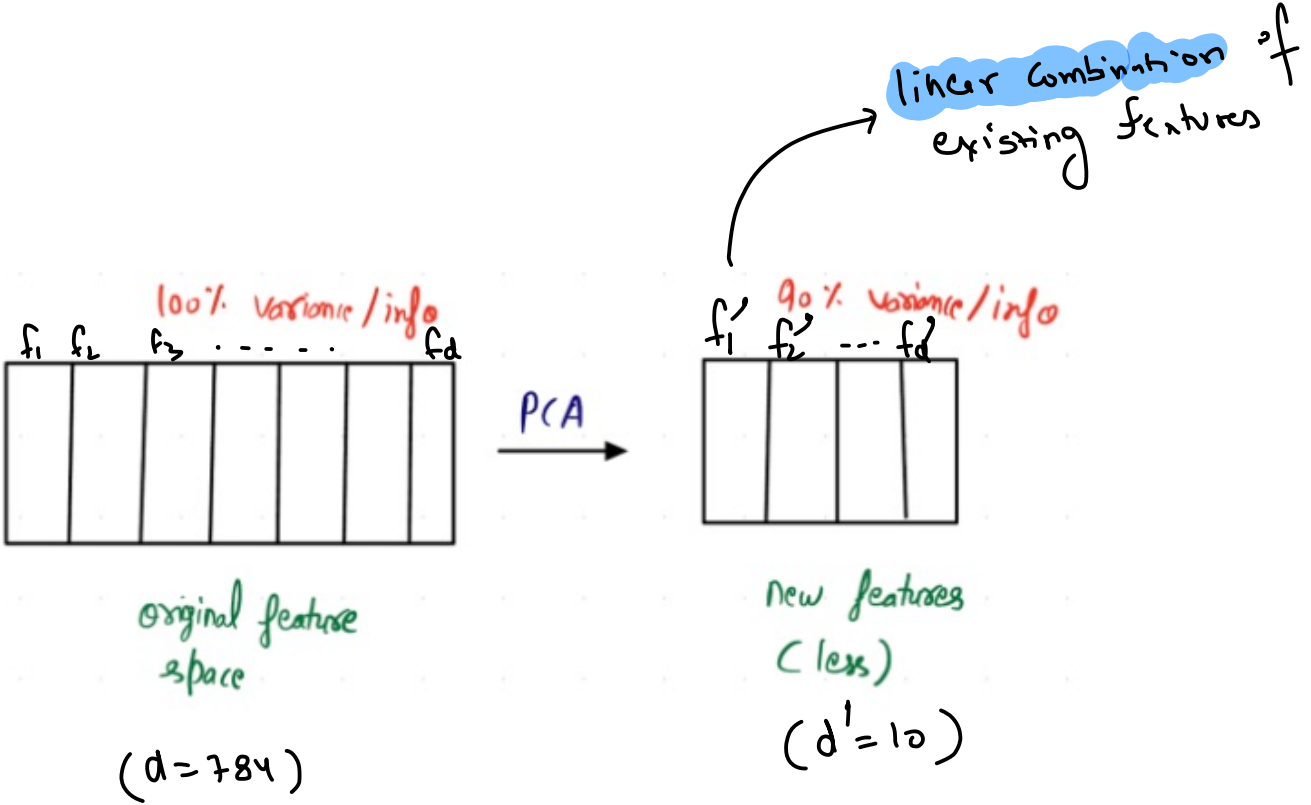
$f_1 \pm f_2$

$f_1 + 0.8 f_2$

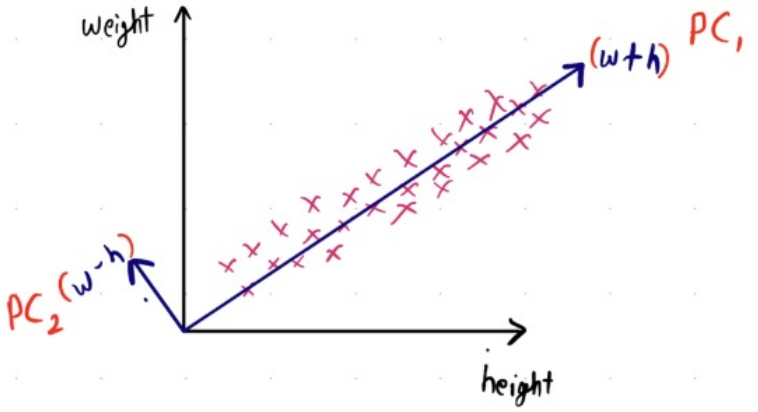
$0.5 f_1 - 4 f_2$



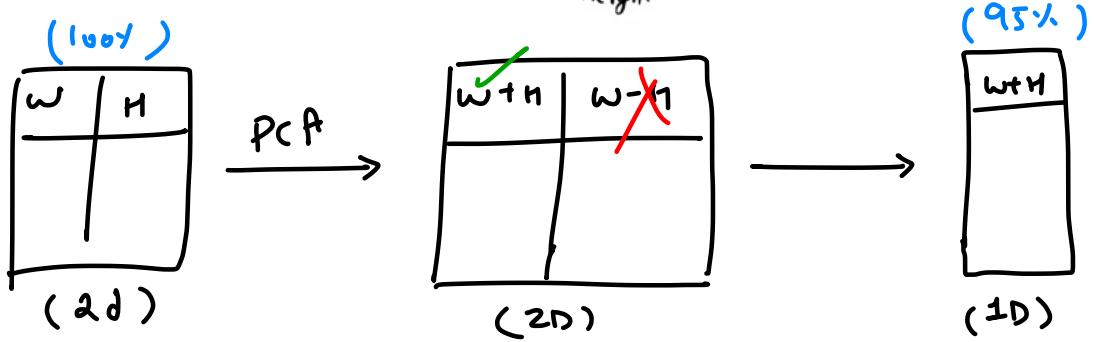



linear combination of existing features
100% variance / info
f1 f2 f3 . . . . . fd
original feature space
(d = 784)
PCA
90% variance / info
f1' f2' ... fd'
New features (less)
(d' = 10)

Principal Component Analysis [PCA]



Principal Components →

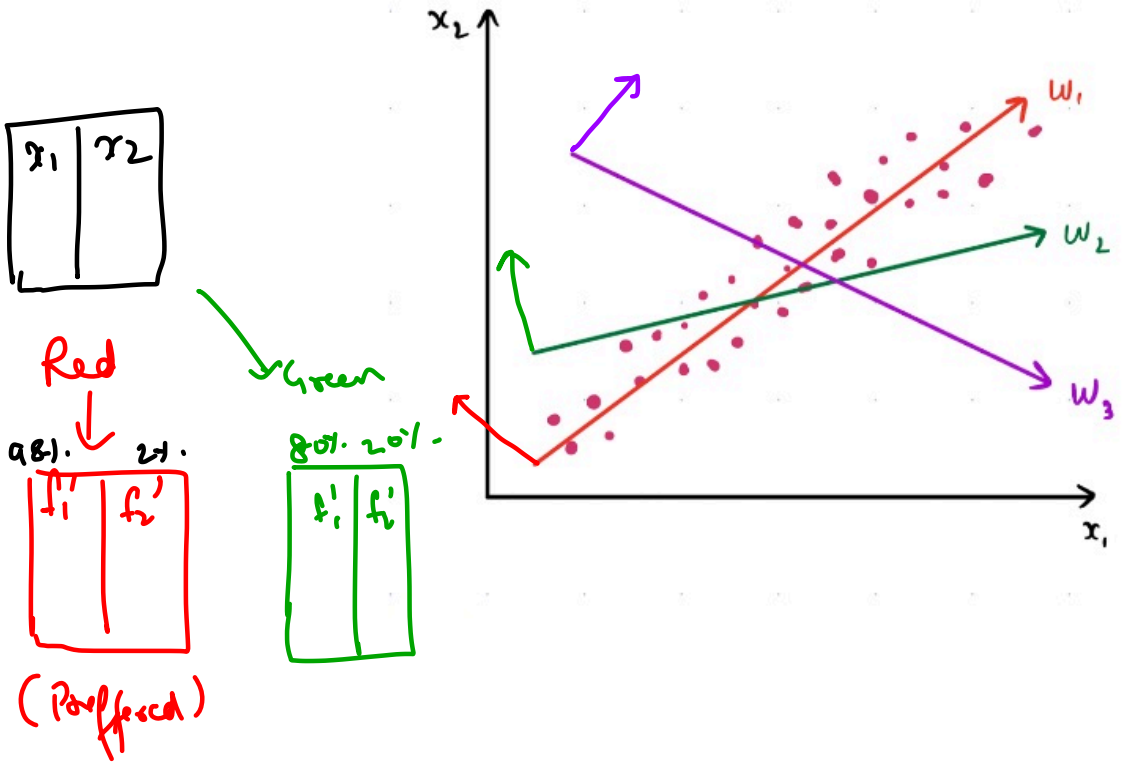


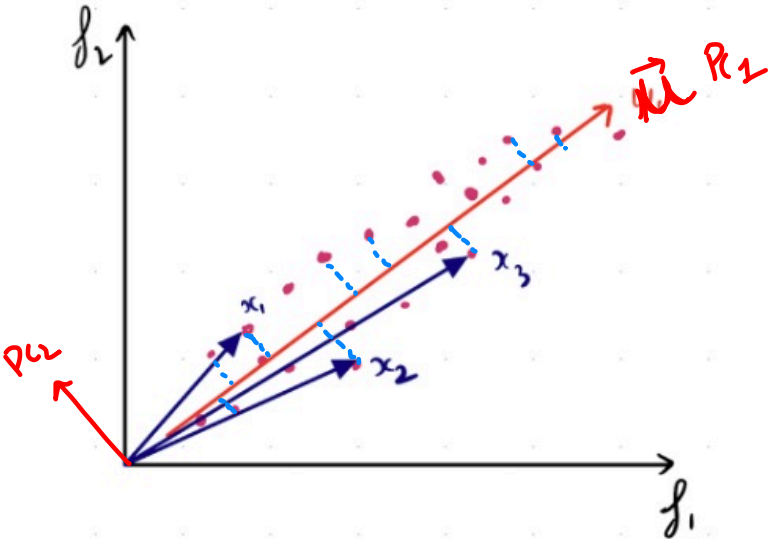
Goal: "95% info/variance should be preserved"

784 → 1 [20%]

784 → 32 [92%] ✗

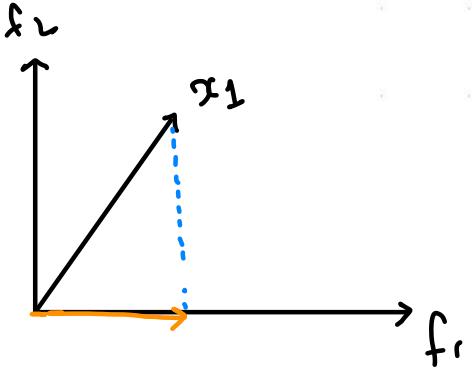
784 → 45 [95.8%] ✓


x_1 x_2
Red
Green
80%. 20%.
f_1' f_2'
f_1' f_2'
(Perfect)
x_2
x_1
w_1
w_2
w_3



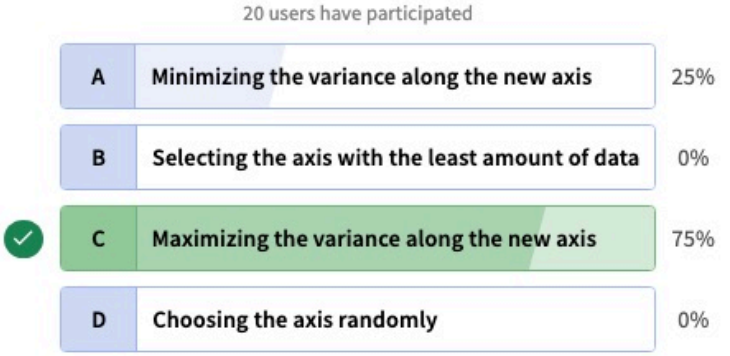
Projection of Points to PC_1

find Variance of Projected Points

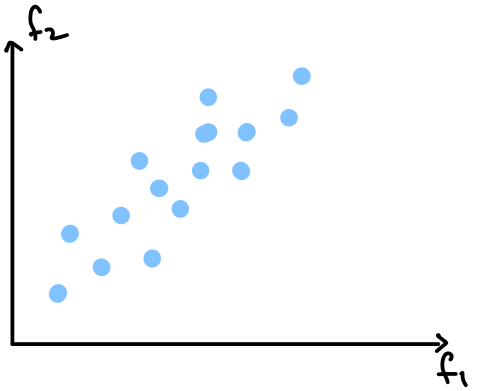


$$\text{Proj}_{f_1} x_1 = \frac{\vec{x}_1 \cdot \vec{f_1}}{\|f_1\|}$$

In Principal Component Analysis (PCA), when rotating the axis to create new axis, which criterion is typically used to select the axis?

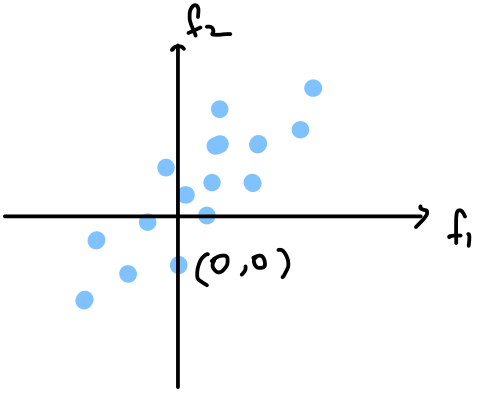


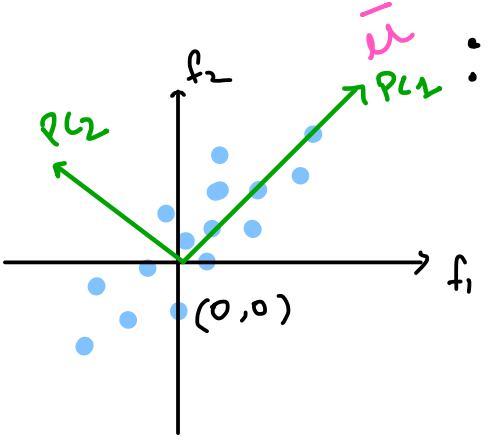
Maths



feature scaling
mean centering

$X - \mu$





$: w^T x + w_0 = 0$

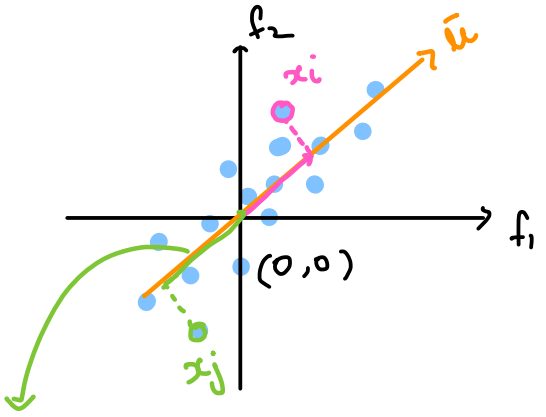
$w_0 = 0$

$w^T x = 0$

$\mu^T x = 0$

find $\bar{\mu}^*$!

$\mu^* = \begin{bmatrix} 2.5 \\ 3.1 \end{bmatrix}$



$$\text{Proj}_{\vec{u}} x_i = \frac{x \cdot u}{\|u\|}$$

Calculate Projection of all Points on $\vec{u}$

$$\text{Proj}_{u} x_j = \frac{x_j \cdot u}{\|u\|}$$

$$\underset{\vec{u}}{\text{argmax}} \; \frac{1}{n} \sum_{i=1}^{n} \frac{x_i \cdot u}{\|u\|}$$

→ avg. Projection length on u

$$\underset{\bar{u}}{\text{argmax}} \left[\frac{1}{n} \sum_{i=1}^{n} \frac{(x_i \cdot u)^2}{\|u\|} \right]$$

→ differentiable
→ -ve & +ve proj. length

Constraint : $\|u\| = 1$

Constraint opt. Problem

↓

Convert into unconstraint opt.

$$\underset{\bar{u}, \lambda}{\text{argmax}} \; \frac{1}{n} \sum_{i=1}^{n} (x_i \cdot \bar{u})^2 + \lambda \left(\|u\|^2 - 1 \right)$$

↓ Solve ? " G.D "

$A^2 = A^T A = \|A\|^2$

$$\frac{(X \cdot u)^2}{n} + \lambda(\|u\|^2 - 1)$$

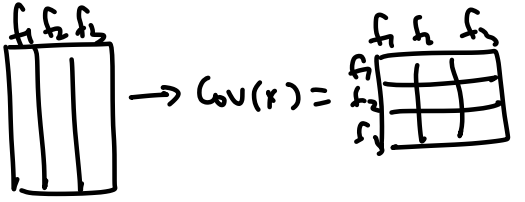
$$\frac{(X \cdot u)^T \cdot (X \cdot u)}{n} + \lambda(\|u\|^2 - 1)$$

$(AB)^T = B^T A^T$

$$\underset{u,\lambda}{\operatorname{argmax}} \; u^T \frac{X^T \cdot X}{n} u + \lambda(u^T u - 1)$$

$\frac{X^T \cdot X}{n}$ → cov. matrix(X) = V

np.cov(X)



$$\underset{u,\lambda}{\operatorname{argmax}} \left[u^T V u + \lambda u^T u - \lambda \right]$$

Scalar derivative			Vector derivative		
$f(x)$	$\rightarrow$	$\frac{df}{dx}$	$f(\mathbf{x})$	$\rightarrow$	$\frac{df}{d\mathbf{x}}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{B}$	$\rightarrow$	$\mathbf{B}$
bx	$\rightarrow$	b	$\mathbf{x}^T\mathbf{b}$	$\rightarrow$	$\mathbf{b}$
x^2	$\rightarrow$	$2x$	$\mathbf{x}^T\mathbf{x}$	$\rightarrow$	$2\mathbf{x}$
bx^2	$\rightarrow$	$2bx$	$\mathbf{x}^T\mathbf{B}\mathbf{x}$	$\rightarrow$	$2\mathbf{B}\mathbf{x}$

$$\frac{\partial L}{\partial u} = 0 \qquad \frac{\partial L}{\partial \lambda} = 0$$

$$2Vu + 2u\lambda = 0$$

$$2Vu = -2u\lambda$$

$$Vu = -\lambda u \Rightarrow$$

$$u^T u - 1 = 0$$

$$u^T u = 1$$

eigenvector, scalar - eigenvalue

$$Vu = \lambda u$$

$$V = np.cov(x) = \frac{X^T X}{n}$$

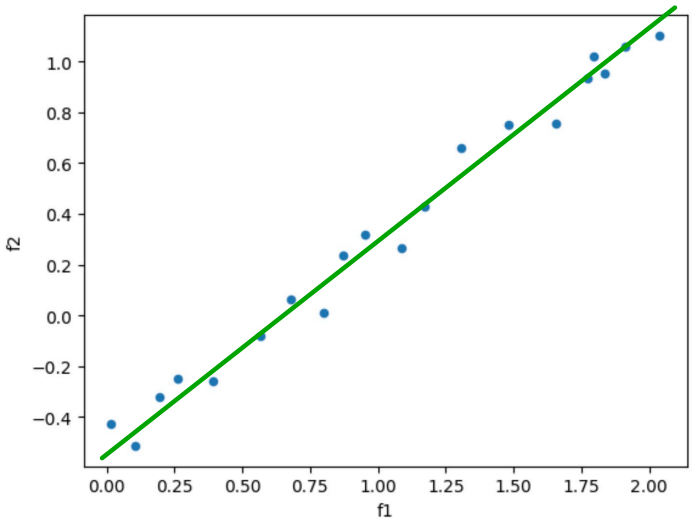
↓

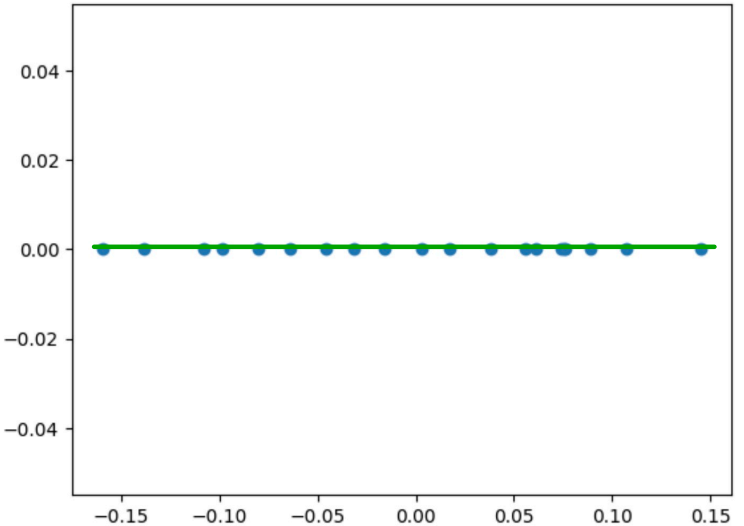
"Eigen decomposition"

https://online.stat.psu.edu/statprogram/reviews/matrix-algebra/eigendecomposition

```
eigvec, eigval = np.linalg.eig(V)
```

Higher Eigenvalue ⇒ Better eigenvector / Principal Component


f2
f1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


0.04
0.02
0.00
−0.02
−0.04
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15

$$p(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(\frac{-(x-\mu)^2}{\sigma^2}\right)}$$