*Principal Component Analysis

* Dimensionality Reduction

OBJECTIVES

1) Preserve in some sense, the original data
2) Smooth out the data for better generalization
3) Smaller dimension results in faster models.

} overfitting

$$X_{n \times d} \longrightarrow \underset{n \times k}{\hat{X}} \qquad k << d$$

reduce dimensions

n → no of data points

d → attributes

* easier model to deploy

* Covariance matrices

attributes of dataset
$$a_1 = [a_{11} \ldots a_{1n}]$$
$$a_2 = [a_{21} \ldots a_{2n}]$$

} covariance between these 2 vectors

$E(a_1)$, $E(a_2)$ → Calculate average of both

$$\text{Covariance}(a_1, a_2) = \left(\underset{1 \times n}{(a_1 - E[a_1])} \underset{n \times 1}{(a_2 - E[a_2])^T} \right) / n$$

↓ measures correlation

$$X_{n \times d} = \begin{array}{ccc} a_1 & a_2 & \ldots \\ \hline \end{array} \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

If $a_1 = a_2$

$a_1 = -a_2$

or

$a_2 = -a_1$

a_1	a_2
temp	temp
C°	F°

high correlation

hight covariance

Standardize

↓

Take Transpose & mutiply

↓

Get Correlation/Covariance

$$\underset{n \times d}{X} \xrightarrow{\text{standardization}} \left\{ \text{average of each attribute is zero} \right.$$

$a_1 - E[a_1]$
$a_2 - E[a_2]$
⋮ ⋮

} standardization

take transpose

$$\underset{n \times d}{X} \rightarrow \underbrace{\underset{d \times n}{X^T} \cdot \underset{n \times d}{X}} = \underset{d \times d}{C}$$

C_{ij} = i th row of X^T

$a_i \cdot a_j$

pairwise correlation

$a_i \cdot a_j$

Correlation/Covariance matrix

$X \rightarrow$

	A	B	C
1			
2			
3			

$X^T =$

	1	2	3
A			
B			
C			

$X \cdot X^T =$

	A	B	C
A			
B			
C			

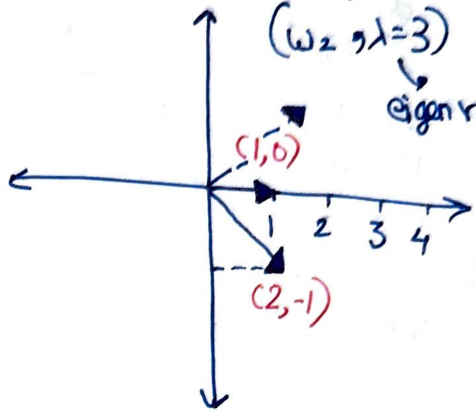
Eigen Values

* What is the action of a matrix to a vector?

$\left.\begin{array}{l} w_1 \\ w_2 \end{array}\right\}$ eigen vectors

$(w_1, \lambda = 1)$
$(w_2, \lambda = 3)$ → eigen values

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \underset{\text{Input}}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} = \underset{\text{Input}}{\begin{bmatrix} 2 \\ -1 \end{bmatrix}}$$



w_1

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\text{Input}} = \underbrace{\begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\text{Output}} . 1$$

no change

w_2

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \underset{\text{input}}{\begin{bmatrix} 1 \\ -1 \end{bmatrix}} = \underset{\text{output}}{\begin{bmatrix} 3 \\ -3 \end{bmatrix}} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} . 3$$

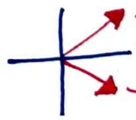
break into

Same direction
only length changes

$\frac{2 \times 2}{\text{matrix}}$ vectors → 2 eigen vectors
2 eigen values

* A property of eigenvalues

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0 \quad \Big\} \quad w_1^T \cdot w_2 = 0 \ *$$



} orthogonal

* $\lambda_1^2 + \lambda_2^2 = 1 + 9 = 10$

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

Square each entry

$= 2^2 + (-1)^2 + (1)^2 + 2^2$
$= 4 + 4 + 1 + 1$
$= 10$

* Generalization

A
$d \times d$

eigen vectors

$w_1, w_2, \ldots w_d$: not changing direction

$\lambda_1, \lambda_2 \ldots \ldots \lambda_d$: tell us how much vector stretch

eigen values

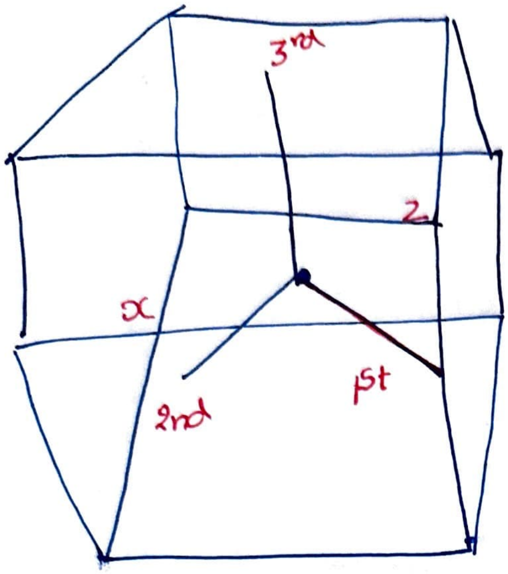
$$A \, w_i = \lambda_i \, w_i$$

$\omega_i^T \cdot \omega_j = 0$ } orthogonal for all i & j

$$\underbrace{\sum_{\substack{i=1..d \\ j=1..d}}^{d} A_{ij}^2}_{\text{Sum of all entries Square}} = \underbrace{\sum_{j=1}^{d} \lambda_j^2}_{\text{sum of squares of eigen values}}$$ } Frobenius norm of matrix

Dimensionality Reduction – the notions

Principal Component



vectors

$V1 = PC1$

$V2 = PC2$

$V3 = PC3$

X
$n \times 3$

$V1$ & $V2$ orthogonal

$V_1^T V_2 = 0$, $V_2^T V_3 = 0$, $V_1^T V_3 = 0$

mutually orthogonal vectors.

$w = C_1 V_1 + C_2 V_2 + C_3 V_3$

any point w/vector can be in our dataset X represented as sum of these PC

Say w has less variance with V_3

So $C_1 V_1 + C_2 V_2 + C_3 V_3$

small (noise)

transformation

$X \rightsquigarrow \hat{X}$

$n \times 3$ $\quad$ $n \times 2$

every point w ⟶ point in new dataset

$[x_1, y_1, z_1]$ $\quad$ $[C_1, C_2]$

take every point w → transform and only keep 2 components (high variance components)

$\underset{n \times k}{\hat{X}} = \underset{n \times d}{X} \cdot \underset{d \times k}{W}$ } Find how to calculate W linear dimensionality reduction

C_1 & C_2 → capture the variance information most (kept)

C_3 → dropped after transformation

Remember

↳ correlation

gives correlation / variance between features

↓

cannot directly use it

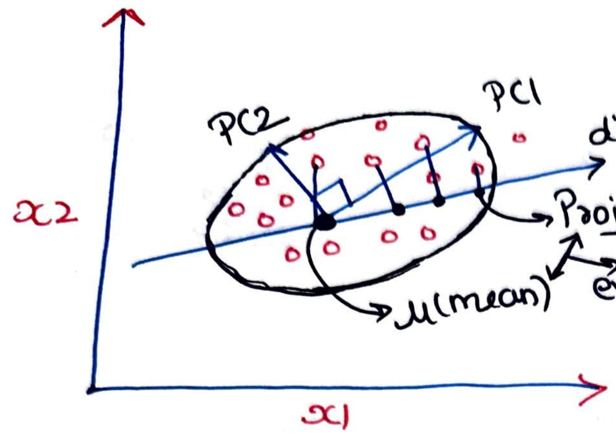
Convert/ decompose

we use eigen vector / values → multiply with data points

↓

find how much scaling the features need/do to the data.

* Dimensionality Reduction – Math Intuition :-



• projections $x_1 \dots x_n$

average

mean point

$$\frac{x_1 + \dots x_n}{n} = \mu$$

Distance of every red point from μ

we get

Variance $\underbrace{(x_2 - \mu)^2}_{\text{Scalar value}}$

How to compute PCA

$CW = \lambda W$

$(C - \lambda I)W = 0$ ← get W, λ both

$C - \lambda I = 0$

↓

Solve for λ → then substitute back

Look for direction that maximized variance

$\max_{d}$ Var d – PC1

Look for direction

How to compute PCA

$$\underset{n \times d}{X} \xrightarrow{\text{standardize}} \underset{\substack{d \times d \\ \text{correlation}}}{C} = X^T X$$

$$C\omega_i = \lambda_i \omega_i \quad \left| \begin{matrix} \omega_i, \lambda_1 \\ \vdots \quad \vdots \\ \omega_d, \lambda_d \end{matrix} \right]$$

$\lambda_1 > \lambda_2 > \lambda_3 \dots > \lambda_d$

λ_1^2 → variance of our data

$$\hat{X} = \underset{d \times k}{X \cdot W} = X \cdot \left(\omega_1 | \omega_2 | \dots | \omega_k \right)$$

What dimension to reduce to?

k-eigen vectors

$d \times k$

→ Sum of first k eigen values

$$C \qquad \text{Total Variance} = \|C\|_F = \sum C_{ij}^2 = \lambda_1^2 + \lambda_2^2 + \dots \lambda_d^2$$

Stop at λ_k^2

90% of Variance → Drop other eigen vector, values

Kernel PCA (non linear dimensionality reduction)

Covariance matrix

$$C\, w_i = \lambda_i \cdot w_i \Rightarrow X^T \cdot X \cdot w_i = \lambda_i w_i \xrightarrow{\text{multiply by } X}$$

C: $d \times d$

eigen values

$$\Rightarrow X \cdot X^T \cdot X w_i = \lambda \cdot X \cdot w_i \quad \} \; z_i = X w_i$$

$$\underbrace{X \cdot X^T}_{\substack{\text{matrix} \\ n \times n}} \cdot z_i = \lambda z_i$$

} not efficient

λ_1	z_1
λ_2	z_d
$\vdots$	$\vdots$

$$\hat{X} = |z_1|z_2|\cdots|z_k| \; : \text{identical } X \cdot w$$

Introducing the kernel

$$k = X \cdot X^T \quad \Big| \quad k_{ij} = \underbrace{x_i \cdot x_j}_{}$$

x_i: i^{th} row, x_j: j^{th} row

Similar to Correlation matrix

$$k_{i,j} = \underbrace{\langle x_i, x_i \rangle}_{\text{linear}} : \text{similarity}$$

$$k_{ij} = \exp(-\|x_i - x_j\|_2) \quad \text{RBF kernel}$$

] not explicitly high dimension } similarity.

non linear reduction → linear d . . . . reduction

→ Similarities measured in some space

kernels differentiates features based on similarity / by calculating distances in 2D, 3D . . spaces

* Kernel Trick