rank of the matrix

$$\begin{array}{cc} \times & \times \\ \end{array}$$
$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \rightarrow R_K = 2$$

$$\begin{array}{cc} \times & \checkmark \\ \end{array}$$
$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \rightarrow R_K = 1$$

It's the number of independant rows or column in the matrix by independant means that it's not a linear combination of any other column

$$\begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array}$$
$$\begin{bmatrix} 1 & 1 & 2 & 4 & 2 \\ 2 & 1 & 3 & 5 & 4 \\ 1 & 1 & 2 & 4 & 2 \\ 0 & 1 & 1 & 3 & 0 \end{bmatrix}$$
$$\begin{array}{ccccc} \times & \times & \checkmark & \checkmark & \checkmark \end{array}$$

$R_K = 2$

- The Rank represent the maximum number of independant vectors that can span the colum of the matrix
- used mainly in dimensionality reduction

- in the last matrix the first two colums are independant as they can't be spaned from any other column
- The last 3 column are dependant
  - $a_3 = a_1 + a_2$
  - $a_4 = a_1 + 3a_2$
  - $a_5 = 2a_1 + 0a_2$

Scanned with CamScanner

we can write these combinations in a neat form "matrix maltiplication"

$$\begin{bmatrix} | & | & | & | & | \\ a_1 & a_2 & a_3 & a_4 & a_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 3 & 0 \end{bmatrix}$$

the original matrix

$A = A' \times R$

independant variables "ingredients" of the meal

"Recipes" that make the meal

why rank of the matrix is important ?

- number of elements in original matrix say "A" is $4 \times 5 = 20$

- by getting the Rank and A Prime matrix "A'" and R matrix we can represent the original ~~number~~ matrix using 18 element

$$\underset{20}{A} = \underset{\underbrace{8 \quad 10}_{18}}{A' \times R}$$

- without making decomposition to A we used 20

from 20 to 18 It's not a big deal but when scaling up to data science application the impact start to arise

A is $N \times P$
A' is $N \times K$] $N(K+P)$
R is $K \times P$

N number of rows
P " column
K rank of the matrix

we mainly use it in dimensionality reduction

$$\frac{K(N+P)}{NP} = \frac{K}{N} + \frac{K}{P}$$

if we have data with N instances = 1000 and feature P = 10

if we got the Rank of the data and find it equal K = 5

Then the compression = $\frac{5}{10} + \frac{5}{1000} \simeq 50\%$

after doing decomposition to the data we only have to save only 50% of the elements

Scanned with CamScanner

we can get the Rank of the matrix using reduced row echo in form

$$A = \begin{bmatrix} 1 & -1 & -2 \\ -1 & 2 & 3 \end{bmatrix} \begin{matrix} R \\ R_1+R_2 \end{matrix}$$

$$A = \begin{bmatrix} 1 & -1 & -2 \\ 0 & 1 & 1 \end{bmatrix} R_2+R_1$$

$$A = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

the first and the second columns are the A Prime of the Basis of the matrix

$$A = A' \times R$$

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

Scanned with CamScanner

in machine learning knowing the rank of the data:

1. understanding the dimensionality of the data so we can perform

- dimensionality reduction: since using high dimensional data "data with high dimensional data", understanding the rank helps in determining how many independant features are present
  - The rank give the maximum number of linearly independant colums, which indicates the effective dimensionality of the data

- techniques like PCA "principle component analysis" use the Rank to redule the data to its most significant features preserving as much variance as possible while reducing the number of dimensions

→ The higher the rank, the higher the unpredictability between the variables

→ The lower the Rank, the higher the relation between the variable

Scanned with CamScanner

eigen values and eigen vectors

The eigen vector is a vector which after applying the transformation stay at the same basis

→ after transformation the vector stay the same but multiplied by a scalar, this scalar is called eigen value

$$A\vec{X} = \lambda\vec{X}$$

$\vec{X}$ = eigen vector
λ = eigen value (scalar value)

- eigen vector must be not trivial zero vector

- matrix can have multiple eigen values (but no more than its number of rows and columns)

Scanned with CamScanner

Solving eigen value and eigen vector

$$AX = \lambda X \rightarrow (A - \lambda I)X = 0$$

This can't be the zero vector

Some constrains
if this matrix were invertable :

$$(A-\lambda I)^{-1}(A-\lambda I)\vec{X} = (A-\lambda I)^{-1} 0$$

$$\vec{X} = 0$$

The $(A-\lambda I)$ must be non invertable thus the determant must be zero

$\rightarrow |A-\lambda I| = 0$ characteristic polynomial

very important notice the eigen vector is any multiple of its basic vector

$\vec{X}$, $C\vec{X}$, $S\vec{X}$ are eigen vectors

Scanned with CamScanner

Principle Component analysis

PCA is the bedrock dimensionality reduction technique for probability and statistics and it's very popular in machine learning

- It's done by projecting each data point onto only the first few principal components to obtain lower dimensional data while preserving as much of the data variation as possible

- Principal components are eigenvectors of the data covariance matrix

Steps for dimensionality reduction using PCA

1. Standardize the data
2. Compute the covariance matrix
3. calculate the eigen vector and eigen values
4. Sort eigenvalues in descending order and compute the principle components
5. Reduce the dimensions of the data set

Scanned with CamScanner

Data Standardization

if the features of the dataset have a different range of values say height from .5 m to 2 m and weight 10 to 120 we should standardize the data in order to have similar range of values and the model can perform well on these features

we perform it using Z-Score or Standard Score

$$Z = \frac{x - \bar{x}}{\sigma}$$

where x = data value
$\bar{x}$ = data mean
σ = standard deviation of data

- It gives us an idea of how far from the mean a data point is

- after standardization the mean = zero and the standard deviation = 1

Scanned with CamScanner

Calculate the covariance matrix

- covariance is always measured between two variables of features
- It measures only the directional relationship between two variables not the strength of the relationship between them

formula of covariance

$$Cov(x_1, x_2) = \frac{\sum_{i=1}^{n} [(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)]}{n-1}$$

n = number of samples

Computing the covariance matrix between 3 features (x_1, x_2, x_3)

$$C = \begin{bmatrix} Cov(x_1, x_1) & Cov(x_1, x_2) & Cov(x_1, x_3) \\ Cov(x_2, x_1) & Cov(x_2, x_2) & Cov(x_2, x_3) \\ Cov(x_3, x_1) & Cov(x_3, x_2) & Cov(x_3, x_3 \end{bmatrix}$$

$Cov(x_1, x_1)$ is the variance of x_1

$$= \frac{\sum (x_1 - \bar{x}_1)^2}{n-1}$$

Scanned with CamScanner

Calculate the eigen values and eigen vectors

- required in the determination of the principle that are required to determine the principle component from the covariance matrix

- Eigenvectors help in finding the new transformation where there is maximum variance

$$Cv = \lambda v$$

- Eigenvector, v are those vectors which keep the direction same when multiplied by a matrix say C

- Eigen values, λ are the scalars of the respective eigen vectors

From the covariance matrix C, first find eigen values

$$Cv = \lambda v$$
$$|C - \lambda I| = 0 \quad \text{we get } \lambda_1, \lambda_2$$

- and then get the eigen vectors v_1, v_2

Scanned with CamScanner

Principle Components in PCA

They are the new uncorrelated which are constructed and extracted from the given features

if we have a data with 5 feature (dim-5)

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5$

from this 5 principle components can be expected and it tries to put the maximum possible informations in the first component and maximum remaining information in the second so we can make use of the first few components only as the most of the informations are compressed on them

previously we computed the eigen values and eigen vectors of the data

$\lambda_1 = 1.284 \qquad \lambda_2 = .049$

$V_1 = \begin{bmatrix} .6779 \\ .7352 \end{bmatrix} \qquad V_2 = \begin{bmatrix} .7352 \\ -.6779 \end{bmatrix}$

to get the principle components we must sort the eigen values in descending order

Scanned with CamScanner

Feature vector = $\begin{bmatrix} .6779 & .7352 \\ .7352 & -.6779 \end{bmatrix}$

- Eigen vector corresponding to highest eigen value will be the first PC in the direction of the max variance

- Reduce the dimension of the data using PCA

Covariance matrix

$$C = \begin{bmatrix} .617 & .615 \\ .615 & .717 \end{bmatrix}$$

$\lambda_1 = 1.284 \qquad \lambda_2 = .049$

Feature vector v =

$$\begin{bmatrix} .6779 & .7352 \\ .7352 & -.6779 \end{bmatrix}$$

Example

x_1	x_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	.9

% of Total variation $= \dfrac{1.284}{1.284 + .049} = 96.3\%$ | $\dfrac{.049}{1.284 + .049} = 3.7\%$

The first principle component (PC1) is the first column corresponding to the highest eigen value $\begin{bmatrix} .6779 \\ .7352 \end{bmatrix}$ and have 97% of the variation of the data

So we can reduce the dimension from 2-D t one D using the PC1

Transforming and reducing the dimension of the original dataset

You transform the data using Z-Score and then mutiply the standard data by the PC1

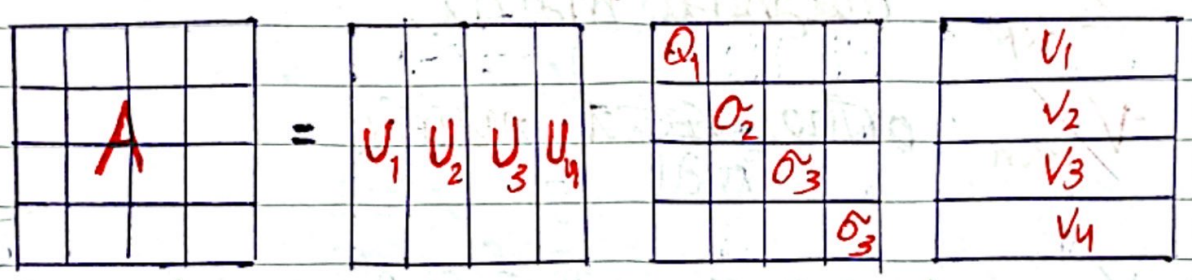
Transformed data = Z * PC1
hence you get one dimensional dataset

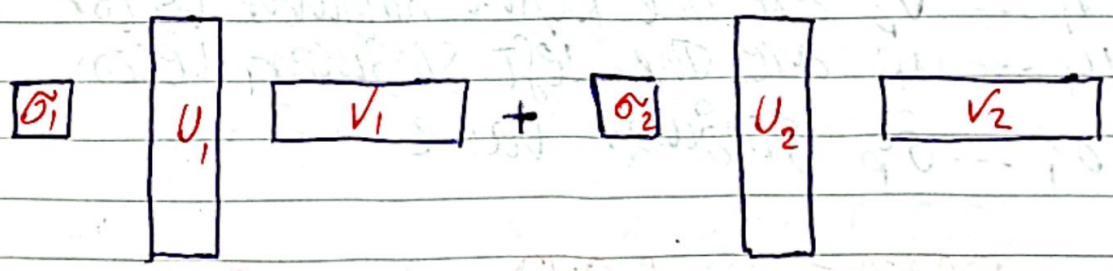
Scanned with CamScanner

Singular value decomposition

Approximation by rank one matrices:

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





and so on

σ_1 and σ_2 ... are the singular number

- Assume that σ_1 and σ_2 are large enough compared to the other value thus we can ignore the other value
- So we can get a very closer ~~value~~ matrix to $\boxed{A}$ with less rank and dimesionality

The eigen value decomposition works only for diagonalizable matrices and square matrices but SVD works for any shape

$U_{m \times p}$ orthogonal $\boxed{U^T U = I_p}$

$\Sigma_{p \times p}$ diagonal matrix

$V^T_{p \times n}$ orthogonal matrix

$V_1 \dots V_p$ are the right singular vector
$U_1 \dots U_p$ are the left singular vector
$\sigma_1 \dots \sigma_p$ singular value

Application of SVD

1. Dimensionality reduction: used in PCA to reduce the dimensionality of the matrix while preserving its variance

2. matrix approximation: like I mentioned if the top singular value are large compared to other values you can approximate this matrix using the first singular values

Scanned with CamScanner

3- image Compression

- $U_{m \times m}$ matrix orthogonal whose columns equal the unit eigenvectors of AA^T
- $V_{n \times n}$ matrix whose colums are equal to the unit eigen vector of A^TA
- Σ $m \times n$ matrix with singular values on the main diagonal and other value of zero

and each value is the square root of the positive eigenvalue of A^TA or AA^T

We can get u using this formula

$$A\vec{v}_1 = \sigma_1 \vec{u}_1 \longrightarrow \frac{1}{\sigma_1} A\vec{v}_1 = u_1$$

$$\frac{1}{\sigma_2} A\vec{v}_2 = \vec{u}_2$$