SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

18CSE481T - APPLIED MACHINE LEARNING

Unit-V

Unit V - Syllabus

- Biometric Face Recognition
- Face detection from the image and video
- Capturing and processing video from a webcam Resizing and Scaling
- Building a face detector using Haar cascades
- determine the location of a face in the video frames captured from the webcam
- Face detector on the grayscale image
- Building eye and nose detectors
- Face cascade classifier
- Visualize eye and nose detector
- Performing Principal Components Analysis
- PCA in face recognition systems
- Convert the dataset from a five dimensional set to a two-dimensional set

Biometric Face Recognition

- **Biometric Face Recognition** is the process and ability of a biometric machine to identify and recognise the face of an individual.
- This is either to grant access to a secured system or to find out the details of a person by matching the face with existing data in the machine's system.
- The biometric face maps and trace the nodes of a person's face geometrically and stores the data with the identity of a certain individual.
- In the initial step, all the data of group of people is fed into one system and can be accessed easily to run the facial recognition tests.
- This data is highly sealed to avoid leakage of this sensitive data of any individual. First, a photo has to be taken by a camera. This is to ensure that future scans will find some data to match with.

Biometric Face Recognition

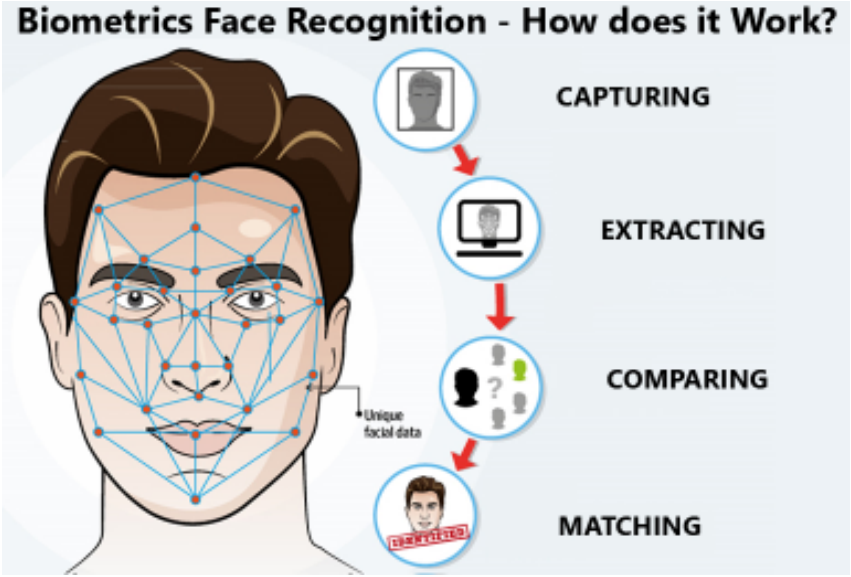
The device is automatic, once turned on and directed on the individual, it runs the authentication process and determines the results accordingly.

Mostly applied in government promises to keep away non-staff or terrorist and also to maintain law and order, banks when withdrawing or accessing the safe, or even in certain functions where it is for the invited only.

Many organizations use biometric attendance machine to monitor the attendance of employees to work. Employees have to initially submit their data physically through initial and complete scans of either their fingers for prints or their faces.

This system advantageous because it helps track the time an employee comes to work and the time they live

Biometric Face Recognition



BIOMETRICS

- A biometric is a unique, measurable characteristic of a human being that can be used to automatically recognize an individual or verify an individual's identity
- Biometrics can measure both physiological and behavioral characteristics.
- Physiological biometrics:-This biometrics is based on measurements and data derived from direct measurement of a part of the human body.
- Behavioral biometrics:-this biometrics is based on measurements and data derived from an action.

TYPES OF BIOMETRICS

PHYSIOLOGICAL

1.Facial Recognition

2.Finger-scan

3.Retina-scan

4.Iris-scan

BEHAVIORAL

1Voice-scan

2.Signature-scan

3.Keystroke-scan

Types of 2 Dimensional Facial Recognition

- **VERIFICATION** -The system compares the given individual with who they say they are and gives a yes or no decision.

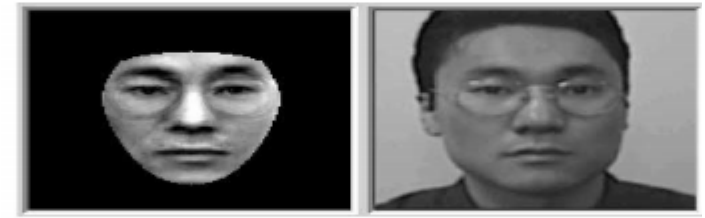
- **IDENTIFICATION** -The system compares the given individual to all the other individuals in the database and gives a ranked list of matches.

OPERATION

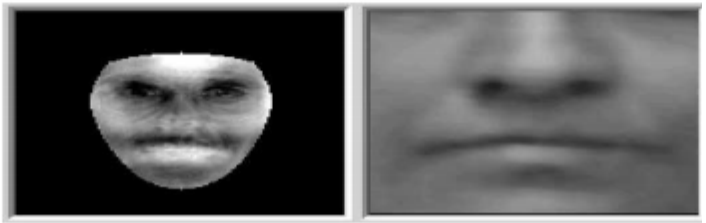
- Image acquisition
- Image processing
- Distinctive characteristic Location
- Template creation
- Template matching

Image acquisition

- Facial scan technology can acquire faces from almost any static camera or video system that generates images of sufficient quality and resolution
- High-quality enrollment is essential to eventual verification and identification enrollment images define the facial characteristics to be used in all future authentication events.



High-Quality Enrollment Image



Low Quality Enrollment Image

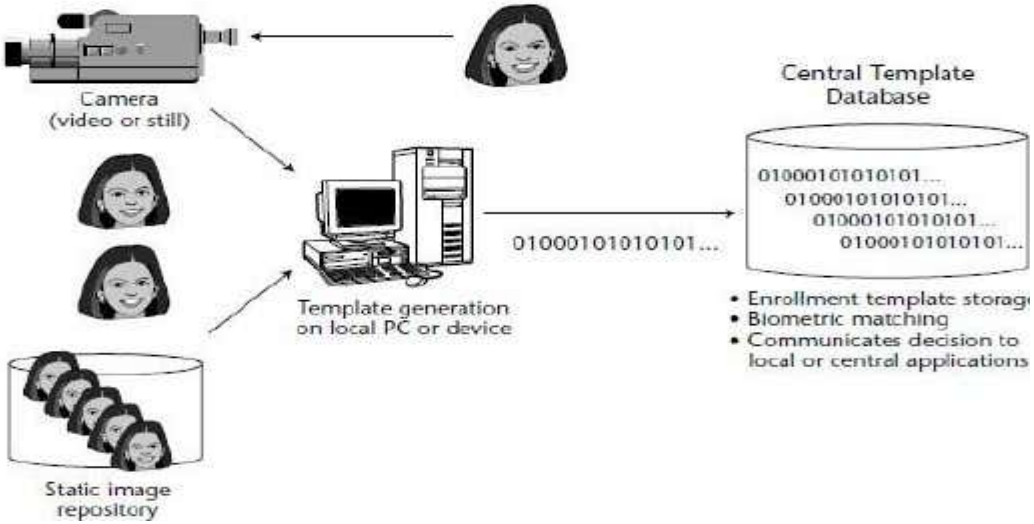
Image Processing

- Images are cropped such that the ovoid facial image remains, and color images are normally converted to black and white in order to facilitate initial comparisons based on grayscale characteristics.
- First the presence of faces or face in a scene must be detected.
- Once the face is detected, it must be localized and Normalization process may be required to bring the dimensions of the live facial sample in alignment with the one on the template.

Distinctive Characteristic Location

- All facial-scan systems attempt to match visible facial features in a fashion similar to the way people recognize one another
- The features most often utilized in facial scan systems are those least likely to change significantly overtime: upper ridges of the eye sockets, areas around the cheek bones, sides of the mouth, nose shape, and the position of major features relative to each other.
- Behavioral changes such as alteration of hairstyle, changes in makeup, growing or shaving facial hair, adding or removing eye glasses are behaviors that impact the ability of facial-scan systems to locate distinctive features, facial-scan systems are not yet developed to the point where they can over come such variables
-

TEMPLATE CREATION



TEMPLATE MATCHING

- It compares match templates against enrollment templates.
- A series of images is acquired and scored against the enrollment , so that a user attempting 1:1verification with in a facial –scan system may have 10 to 20 match attempts take place with in 1 to 2 seconds.
- Facial-scan is not as effective as finger-scan or iris scan in identifying a single individual from a large database, a number of potential matches are generally returned after large-scale facial-scan identification searches

HOW THIS SYSTEM WORK

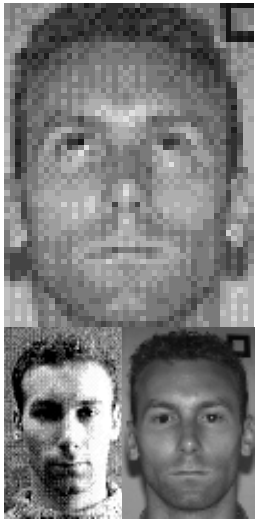
- Facial recognition software is based on the ability to first recognize faces, which is a technological feat in itself. If you look at the mirror, you can see that your face has certain distinguishable landmarks. These are the peaks and valleys that make up the different facial features.
- VISIONICS defines these landmarks as nodal points. There are about 80 nodal points on a human face.

- **Here are few nodal points that are measured by the software.**

1. Distance between the eyes
2. Width of the nose
3. Depth of the eye socket
4. Cheekbones
5. Jaw line
6. Chin

Limitations of 2D Face Recognition

System effectiveness is highly dependant on image capture conditions.



- **Lighting conditions.**
- Different lighting conditions for enrolment and query.
- Bright light causing image saturation.
- **Head orientation.**
- 2D feature distances appear to distort.
- **Image quality.**
- CCTV, Web-cams etc.
- **Facial expression.**
- Changes in feature location and shape.
- **Partial occlusion**
- Hats, scarves, glasses etc.

Result:

Face recognition is not as accurate as other biometrics.

Error rates that are too high for many applications in mind.

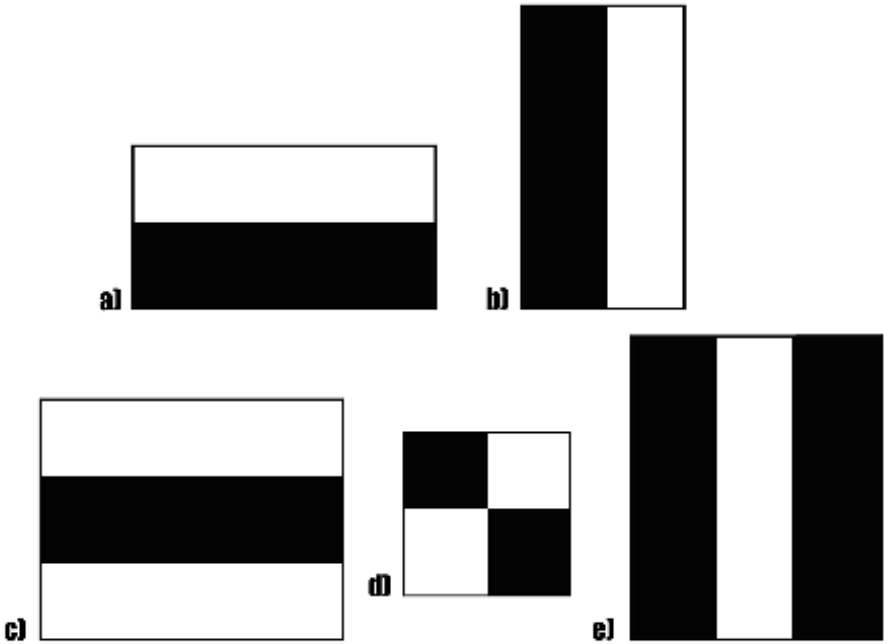
Haar cascade classifier

- Haar cascade classifier, is a machine learning object detection program that identifies objects in an image and video
- It was used in the first real-time face detector
- This method was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features
- Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.
  - **Positive images –** These images contain the images which we want our classifier to identify.
  - **Negative Images –** Images of everything else, which do not contain the object we want to detect.

Calculating Haar Features

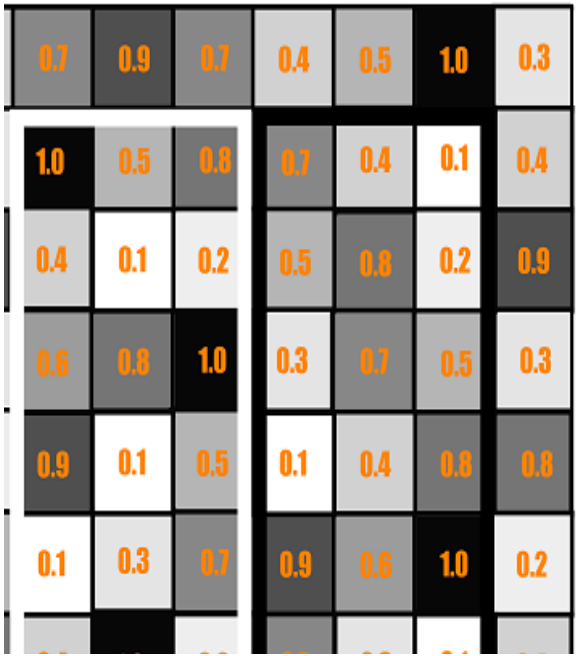
- The algorithm can be explained in four stages:
  - Calculating Haar Features
  - Creating Integral Images
  - Using Adaboost
  - Implementing Cascading Classifiers

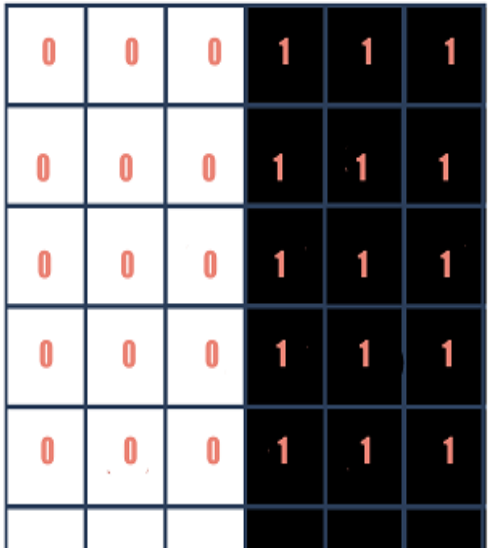
Haar Features



- These features on the image makes it easy to find out the edges or the lines in the image, or to pick areas where there is a sudden change in the intensities of the pixels.

Haar Features





=

SUM OF THE DARK PIXELS/NU
SUM OF THE LIGHT PIXELS/NUM

(0.7 + 0.4 + 0.1 + 0.5 + 0.8 +
0.1 + 0.4 + 0.8 + 0.9 + 0.6 + 1

-

(1.0 + 0.5 + 0.8 + 0.4 + 0.1 +
0.9 + 0.1 + 0.5 + 0.1 + 0.3 + 0

0.51-0.53 =

Calculating Haar Features

- In the figure The rectangle on the left is a sample representation of an image with pixel values 0.0 to 1.0.
- The rectangle at the center is a haar kernel which has all the light pixels on the left and all the dark pixels on the right.
- The haar calculation is done by finding out the difference of the average of the pixel values at the darker region and the average of the pixel values at the lighter region.
- If the difference is close to 1, then there is an edge detected by the haar feature.

Calculating Haar Features

- The darker areas in the haar feature are pixels with values 1, and the lighter areas are pixels with values 0.
- Each of these is responsible for finding out one particular feature in the image. Such as an edge, a line or any structure in the image where there is a sudden change of intensities.
- For ex. in the image above, the haar feature can detect a vertical edge with darker pixels at its right and lighter pixels at its left.
- **In the example above, there is no edge as the haar value is far from 1.**
- To detect an edge anywhere in the image, the haar feature needs to traverse the whole image.

Calculating Haar Features

Depending on the feature each one is looking for, these are broadly classified into three categories.

- The first set of ***two rectangle features*** are responsible for finding out the edges in a horizontal or in a vertical direction (as shown above).
- The second set of ***three rectangle features*** are responsible for finding out if there is a lighter region surrounded by darker regions on either side or vice-versa.
- The third set of ***four rectangle features*** are responsible for finding out change of pixel intensities across diagonals.

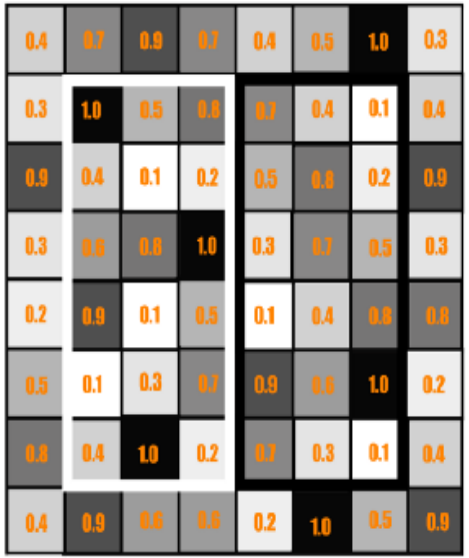
Integral Image

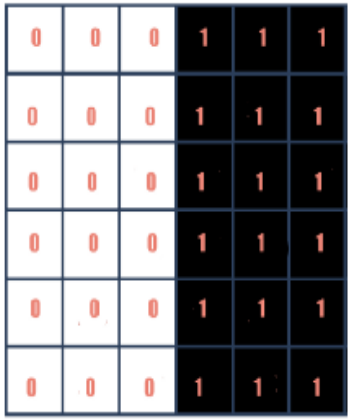
Integral Image

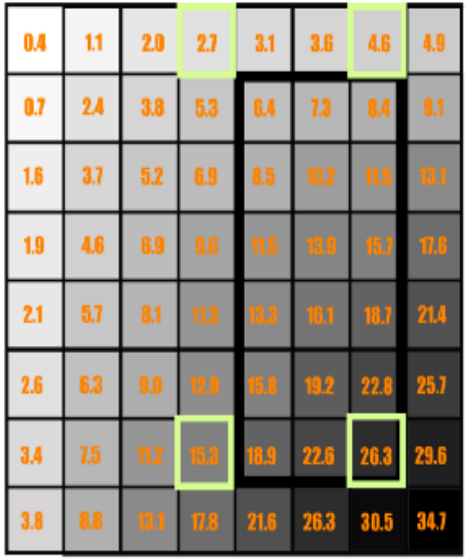
As we can see for a single rectangle on either side, it involves 18 pixel value additions (for a rectangle enclosing 18 pixels). Imagine doing this for the whole image with all sizes of the haar features. This would be a hectic operation even for a high performance machine.

- An Integral Image is calculated from the Original Image in such a way that each pixel in this is the sum of all the pixels lying in its left and above in the Original Image.
- In the integral image the last pixel at the bottom right corner of the Integral Image will be the sum of all the pixels in the Original Image.

Integral Image is used here to calculate the haar value







SUM OF THE PIKELS IN DARK AREA/N

= (26.3 - 15.3 - 4.6 + 2.7)/18 = 9.

SUM OF THE PIXELS IN LIGHT AREA

= (15.3 - 3.4 - 2.7 + 0.4)/18 = 9

This is a case where there is a sudden change of pixel intensities moving vertically from the left towards the right in the image.

0.5	0.3	0.3	0.2	0.7	0.8	0.9	0.9
0.1	0.2	0.2	0.1	0.9	0.7	0.9	0.9
0.5	0.6	0.2	0.4	0.7	0.9	0.8	0.8
0.3	0.2	0.2	0.2	0.7	0.8	0.7	0.8
0.2	0.1	0.2	0.4	0.7	0.6	0.6	0.8
0.4	0.2	0.4	0.1	0.8	0.6	0.6	0.7
0.5	0.3	0.4	0.1	0.7	0.8	0.9	0.6
0.5	0.6	0.4	0.3	0.8	0.6	0.8	0.6

0	0	0	1	1	1		
0	0	0	1	1	1		
0	0	0	1	1	1		
0	0	0	1	1	1		
0	0	0	1	1	1		
0	0	0	1	1	1		

0.5	0.8	1.1	1.3	2.0	2.8	3.7	4.6
0.6	1.1	1.6	1.9	3.5	5.0	6.8	8.6
1.1	2.2	2.9	3.6	5.9	8.3	10.9	13.5
1.4	2.7	3.6	4.5	7.5	10.7	14.0	17.4
1.6	3.0	4.1	5.4	9.1	12.9	16.8	21.0
2.0	3.6	5.1	6.5	11.0	15.4	19.9	24.8
2.5	4.4	6.3	7.8	13.0	18.2	23.6	29.1
3.0	5.5	7.8	9.6	15.6	21.4	27.6	33.7

SUM OF THE PIXELS IN LIGHT AREA/NUMBER OF PIXELS

= (23.6 - 7.8 - 3.7 + 1.3)/18 = 13.4/18 = 0.74

Haar Value
0.74-0.28=0.54

0.5	0.8	1.1	1.3	2.0	2.8	3.7	4.6
0.6	1.1	1.6	1.9	3.5	5.0	6.8	8.6
1.1	2.2	2.9	3.6	5.9	8.3	10.9	13.5
1.4	2.7	3.6	4.5	7.5	10.7	14.0	17.4
1.6	3.0	4.1	5.4	9.1	12.9	16.8	21.0
2.0	3.6	5.1	6.5	11.0	15.4	19.9	24.8
2.5	4.4	6.3	7.8	13.0	18.2	23.6	29.1
3.0	5.5	7.8	9.6	15.6	21.4	27.6	33.7

SUM OF THE PIXELS IN LIGHT AREA/NUMBER OF PIXELS

= (7.8 - 2.5 - 1.3 + 0.5)/18 = 4.5/18 = 0.28

AdaBoost

We know that set of features which would capture certain facial structures like eyebrows or the bridge between both the eyes, or the lips etc.

But originally the feature set was not limited to this. The feature set had an approx. of 180,000 of them, which got reduced to 6000. We will discuss this in more detail below

A majority of these features will be irrelevant to the facial features. So need some a Feature Selection technique, to select a subset of features from the huge set which would not only select features performing better than the others, but also will eliminate the irrelevant ones.

AdaBoost

They used a ***Boosting Technique*** called ***AdaBoost***, in which each of these 180,000 features were applied to the images separately to create ***Weak Learners***.

These weak learners are designed in such a way that they would misclassify only a minimum number of images

"A weak learner produces a classifier which is only slightly more accurate than random classification."

With this technique, their final set of features got reduced to a total of 6000 of them

Attentional Cascade

The Attentional Cascade. The idea behind this is, not all the features need to run on each and every window. If a feature fails on a particular window, then we can say that the facial features are not present there. Hence, we can move to the next windows where there can be facial features present.

- Features are applied on the images in stages.
- The stages in the beginning contain simpler features, in comparison to the features in a later stage which are complex, complex enough to find the nitty gritty details on the face.
- If the initial stage won't detect anything on the window, then discard the window itself from the remaining process, and move on to the next window.
- This way, a lot of processing time will be saved, as the irrelevant windows will not be processed in the majority of the stages.

Attentional Cascade

sample 2 stage feature detection, where the haar features are applied on the image in a 4x4 window. The first stage has 2 simpler features, and the second stage has only 1 complex feature. The first stage is applied first on the 4x4 windows in the image, if it passes, then only the stage is applied.

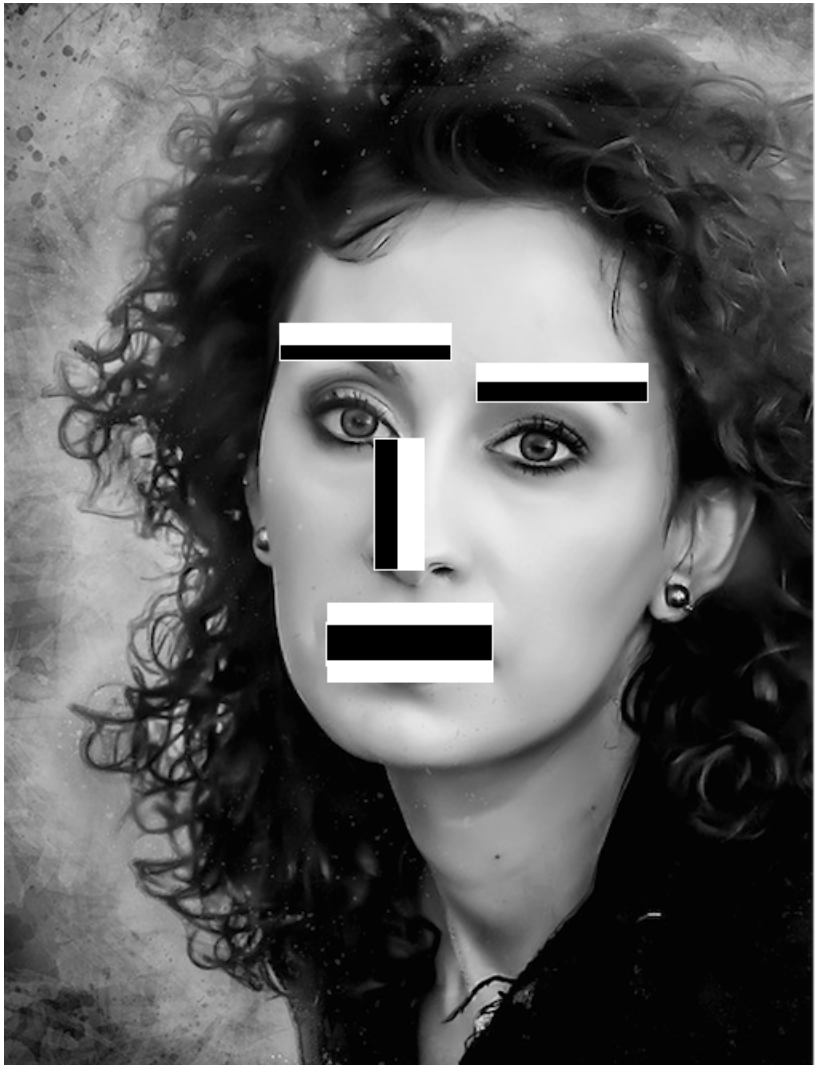
References

1. Rapid Object Detection using a Boosted Cascade of Simple Features
2. Cascade Classifier
3. Face Detection using Haar Cascades
4. Cascade Haar Explained

Haar Features used in Face Detection

Haar features are sequence of rescaled square shape functions proposed by Alfred Haar in 1909. They are ***similar to convolution*** *in the Convolution Neural Networks course*

(1) (2) (3) (4)



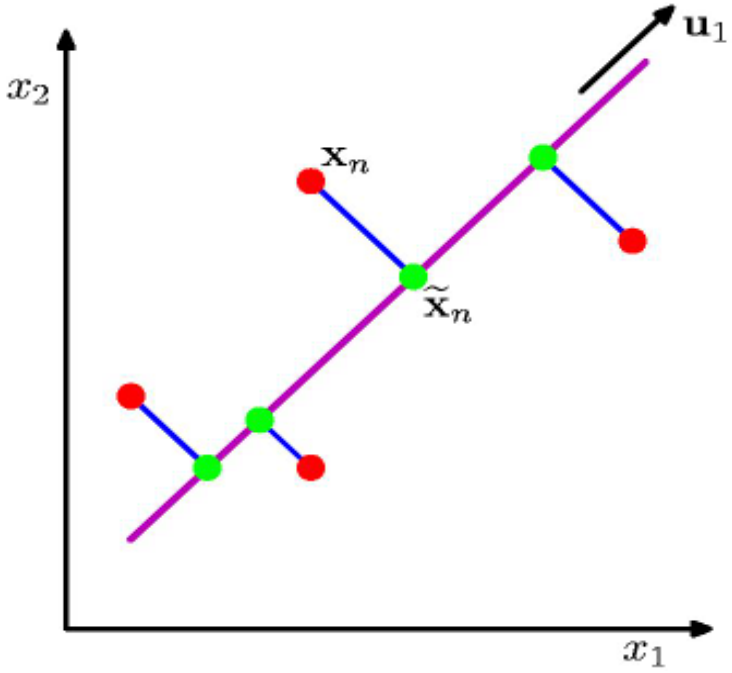
Principal Components Analysis

- Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.
- Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

Principle Component Analysis

Orthogonal projection of data onto lower-dimension linear space that...

- maximizes variance of projected data (purple line)
- minimizes mean squared distance between
  - data point and
  - projections (sum of blue lines)



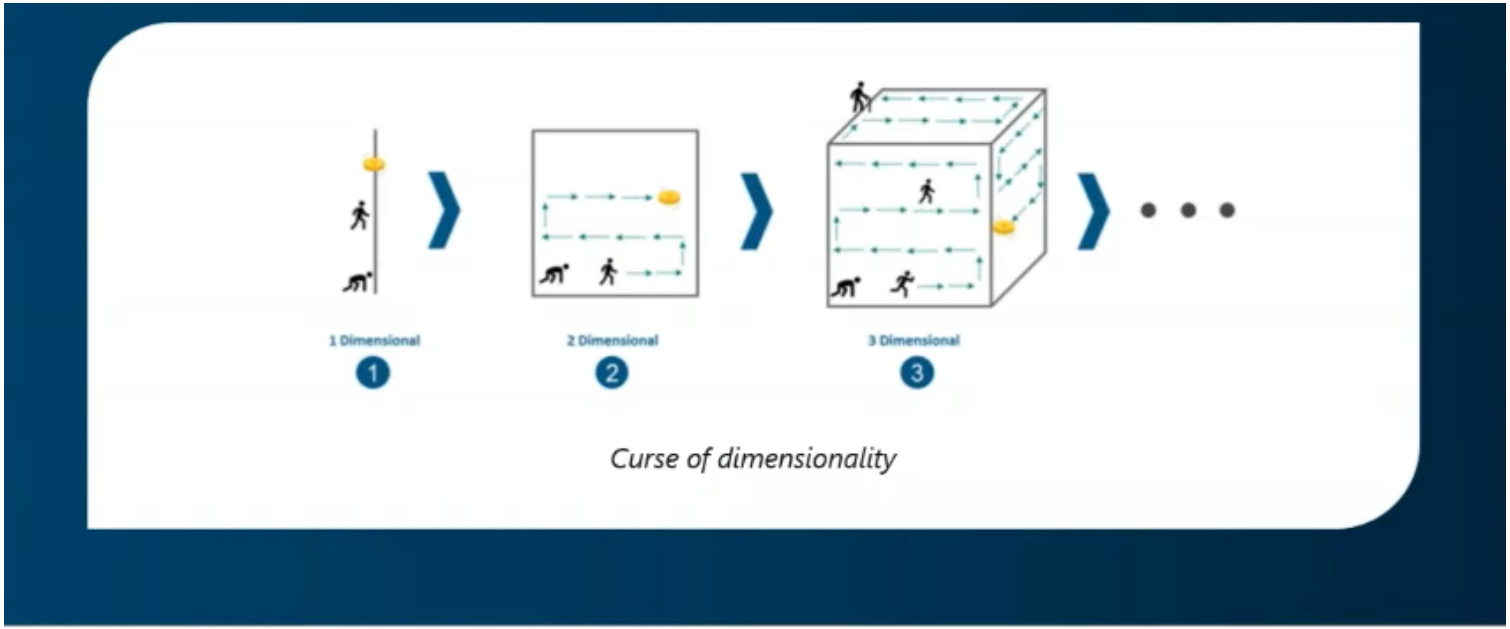
The Principal Components

- **Vectors** originating from the center of mass
- Principal component #1 points in the direction of the **largest variance**.
- Each subsequent principal component...
  - is **orthogonal** to the previous ones, and
  - points in the directions of the **largest variance of the residual subspace**

PCA - Steps

1. Standardize the range of continuous initial variables
2. Compute the covariance matrix to identify correlations
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Create a feature vector to decide which principal components to keep
5. Recast the data along the principal components axes

Principal Component Analysis

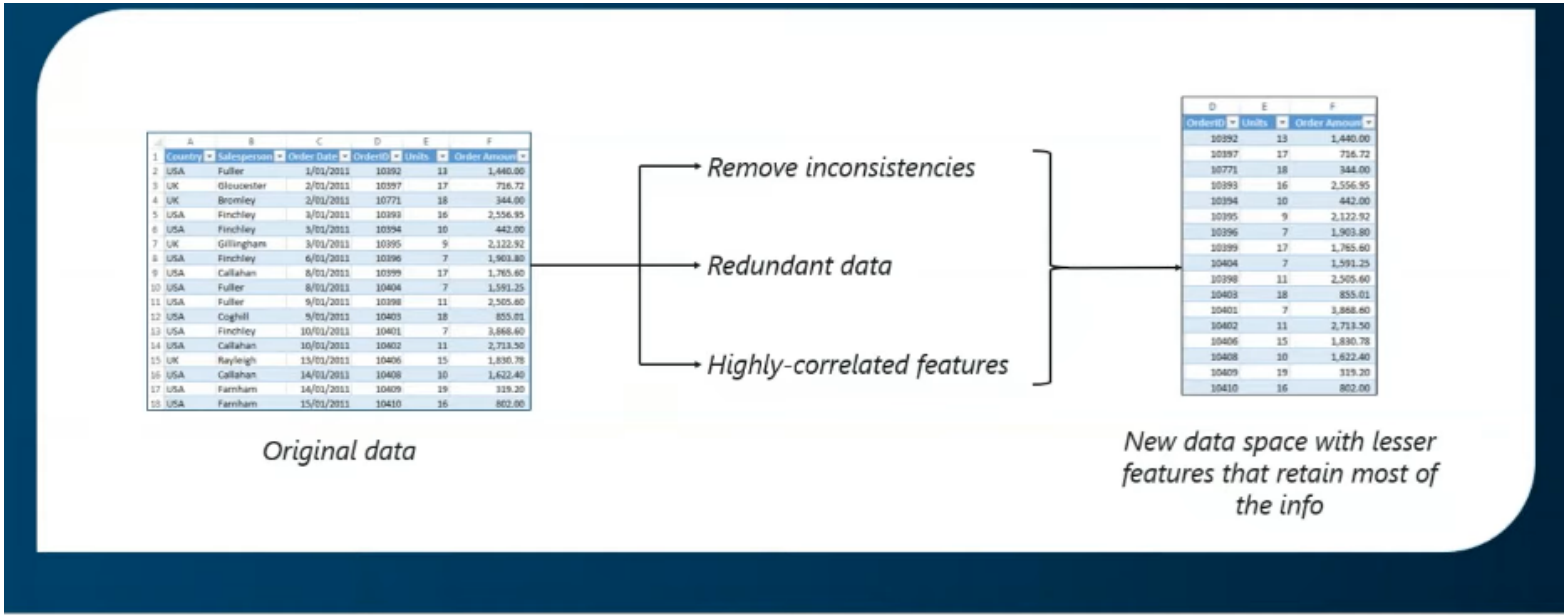


Curse of dimensionality

Need For PCA

High dimension data is extremely complex to process due to inconsistencies in the features which increase the computation time and make data processing and EDA more convoluted.

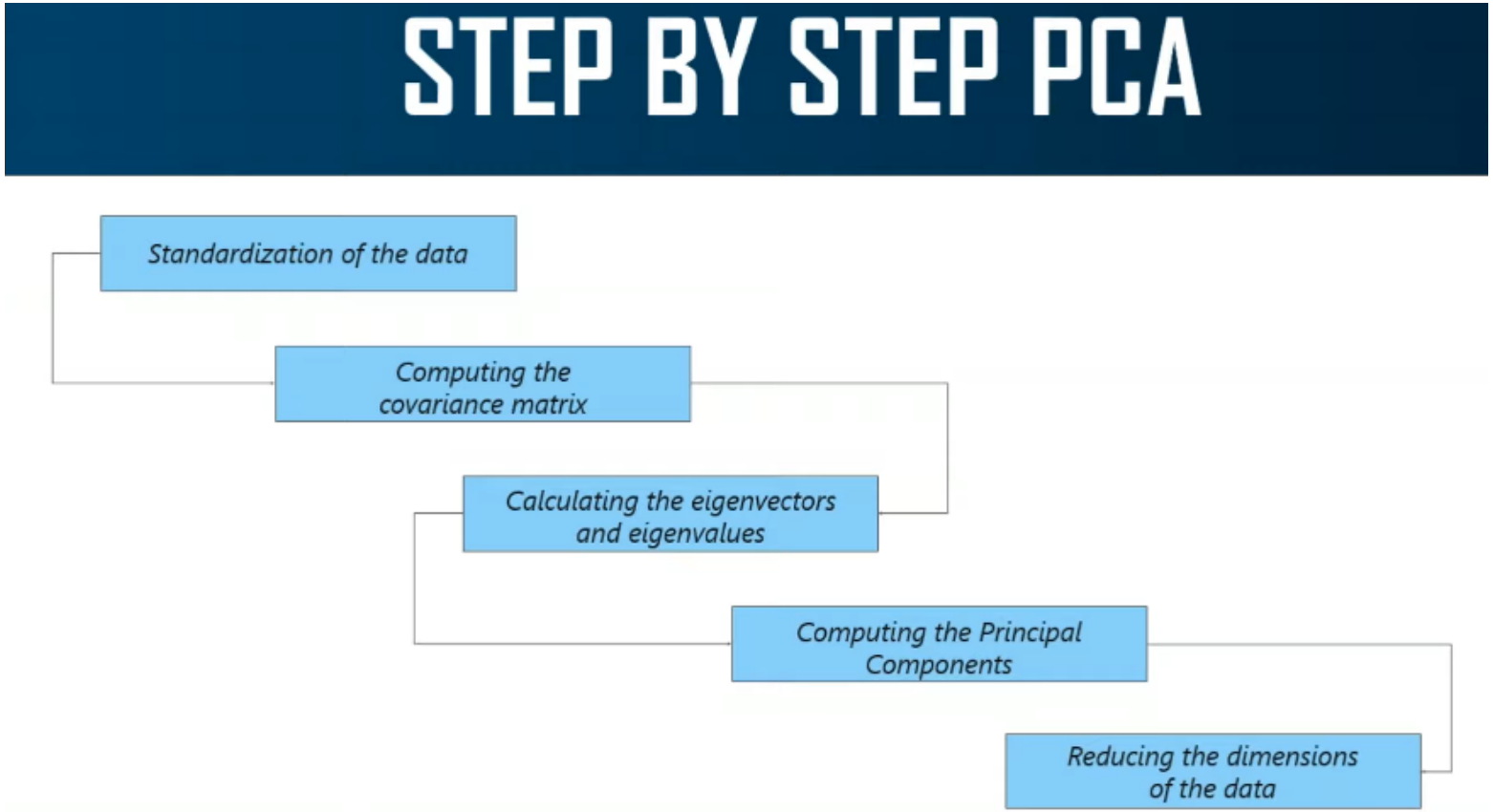
Principal Component Analysis



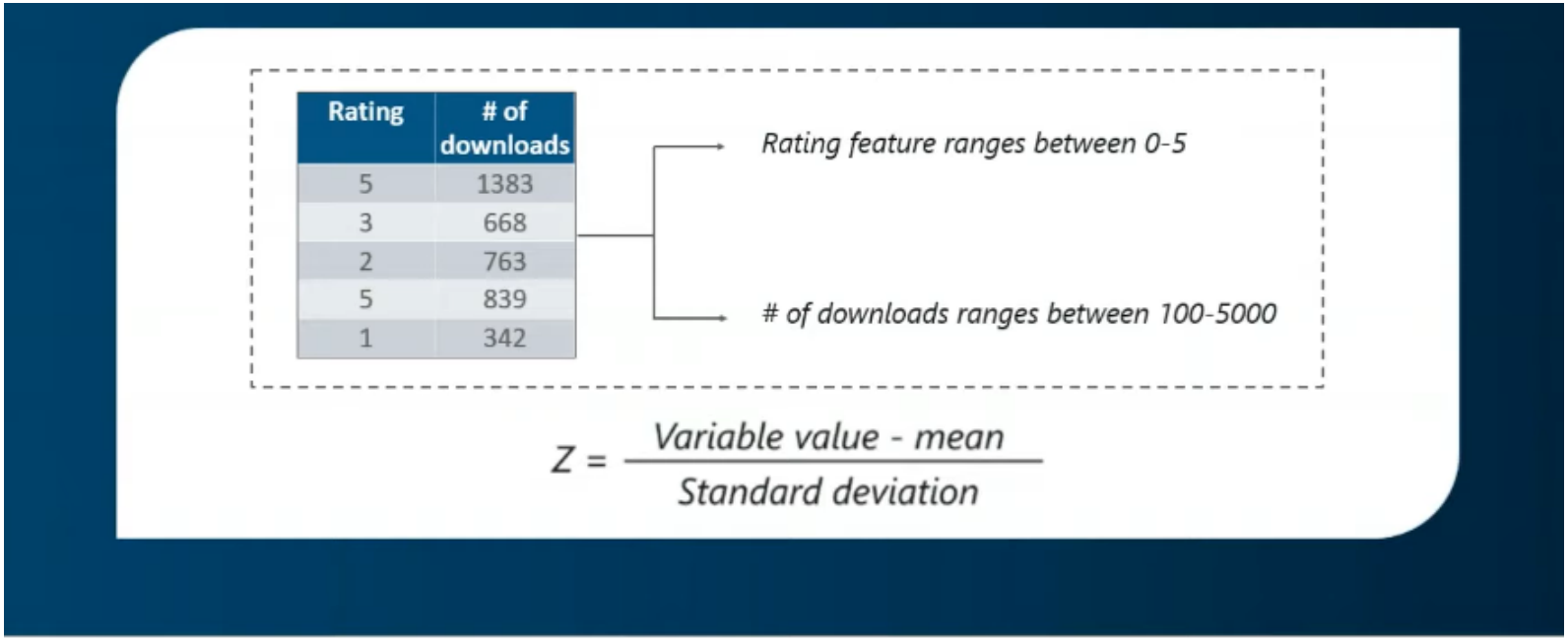
What Is Principal Component Analysis (PCA)?

Principal components analysis (PCA) is a dimensionality reduction technique that enables you to identify correlations and patterns in a data set so that it can be transformed into a data set of significantly lower dimension without loss of any important information.

Steps in PCA


STEP BY STEP PCA
Standardization of the data
Computing the covariance matrix
Calculating the eigenvectors and eigenvalues
Computing the Principal Components
Reducing the dimensions of the data

Step 1 in PCA



Step 1: Standardization of the data

Standardization is all about scaling your data in such a way that all the variables and their values lie within a similar range.

Step 1 in PCA

STEP 1 : Get some Data and Plot.

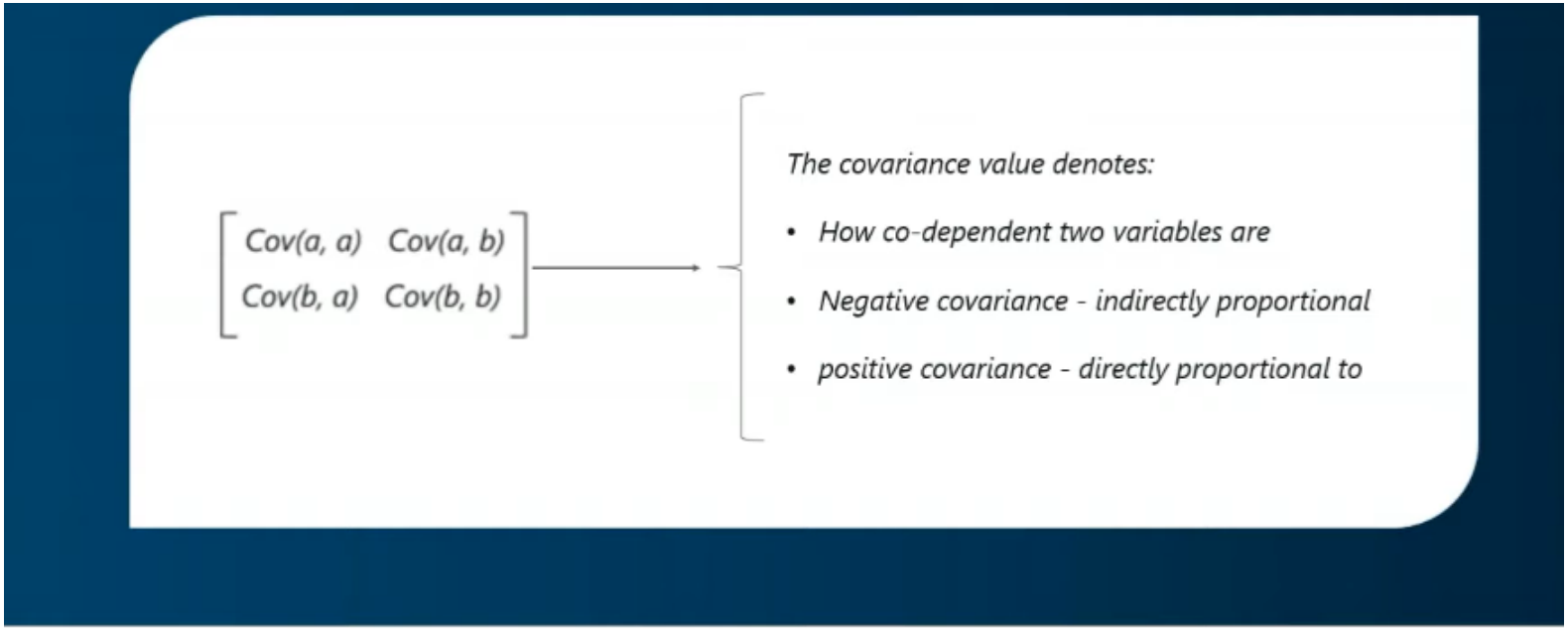
Find mean(X), and mean(Y)

Mean (X) =1.81

Mean (Y) =1.91

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1	1.1
1.5	1.6
1.1	0.9

Step 2 in PCA



Step 2: Computing the covariance matrix

A covariance matrix expresses the correlation between the different variables in the data set. It is essential to identify heavily dependent variables because they contain biased and redundant information which reduces the overall performance of the model.

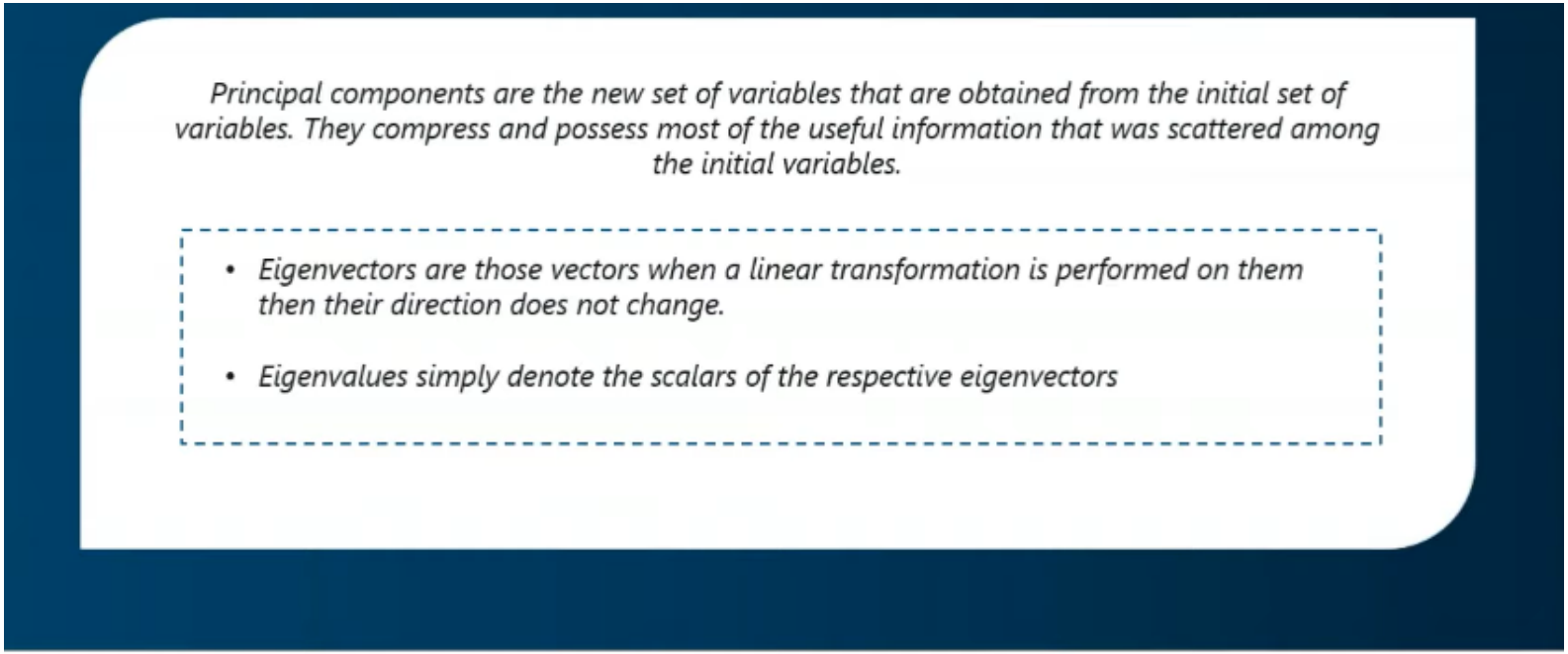
Step 2 in PCA

STEP 2: Subtract the mean to make the Data pass through the origin.

It is called as DATA ADJUST

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1	1.1
1.5	1.6
1.1	0.9

Step 3 in PCA



Step 3: Calculating the Eigenvectors and Eigenvalues

Eigenvectors and eigenvalues are the mathematical constructs that must be computed from the covariance matrix in order to determine the principal components of the data set.

Step 3 in PCA

STEP 3 : Calculate the Covariance Matrix

	X	Y
X	COV(X,X)	COV(X,Y)
Y	COV(Y,X)	COV(Y,Y)

- Shows how two variables vary together

$$\text{COV}(X,Y) = \sum_{i=1}^{n} \frac{(X - \bar{X})(Y - \bar{Y})}{n-1}$$

- If the Value is +ve, Both variables increase together

$$\text{COV} \begin{pmatrix} 0.616 & 0.6154 \\ 0.615 & 0.7165 \end{pmatrix}$$

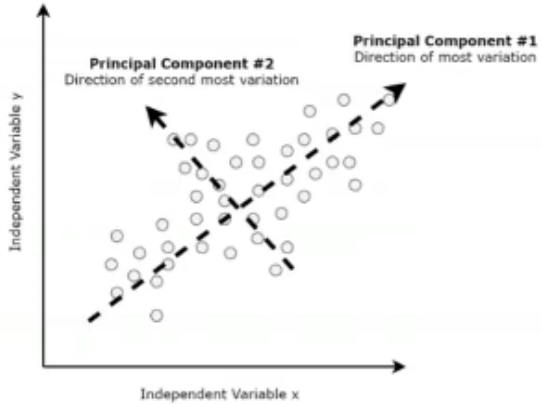
$$C_{mat} = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

$$I1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad C(I1) = \begin{pmatrix} cov(1,1) & cov(2,5) & cov(3,9) \\ cov(4,1) & cov(5,5) & cov(6,9) \\ cov(7,1) & cov(8,5) & cov(9,9) \end{pmatrix}$$

- Since Non Diagonal elements are +ve , X and Y Increase together

Step 4 in PCA

- *PC1 is the most significant and stores the maximum possible information.*
- *PC2 is the second most significant PC and stores remaining maximum info and so on*

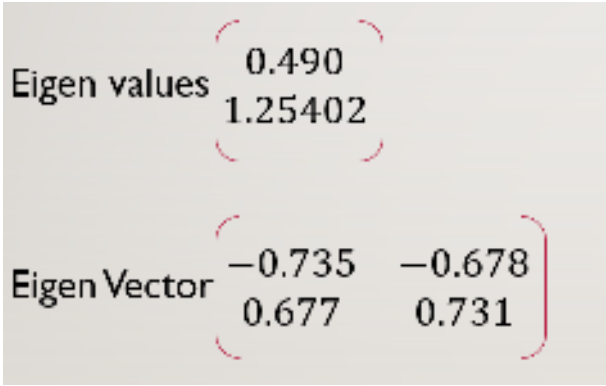


Step 4: Computing the Principal Components

Once we have computed the Eigenvectors and eigenvalues, all we have to do is order them in the descending order, where the eigenvector with the highest eigenvalue is the most significant and thus forms the first principal component.

Step 4 in PCA

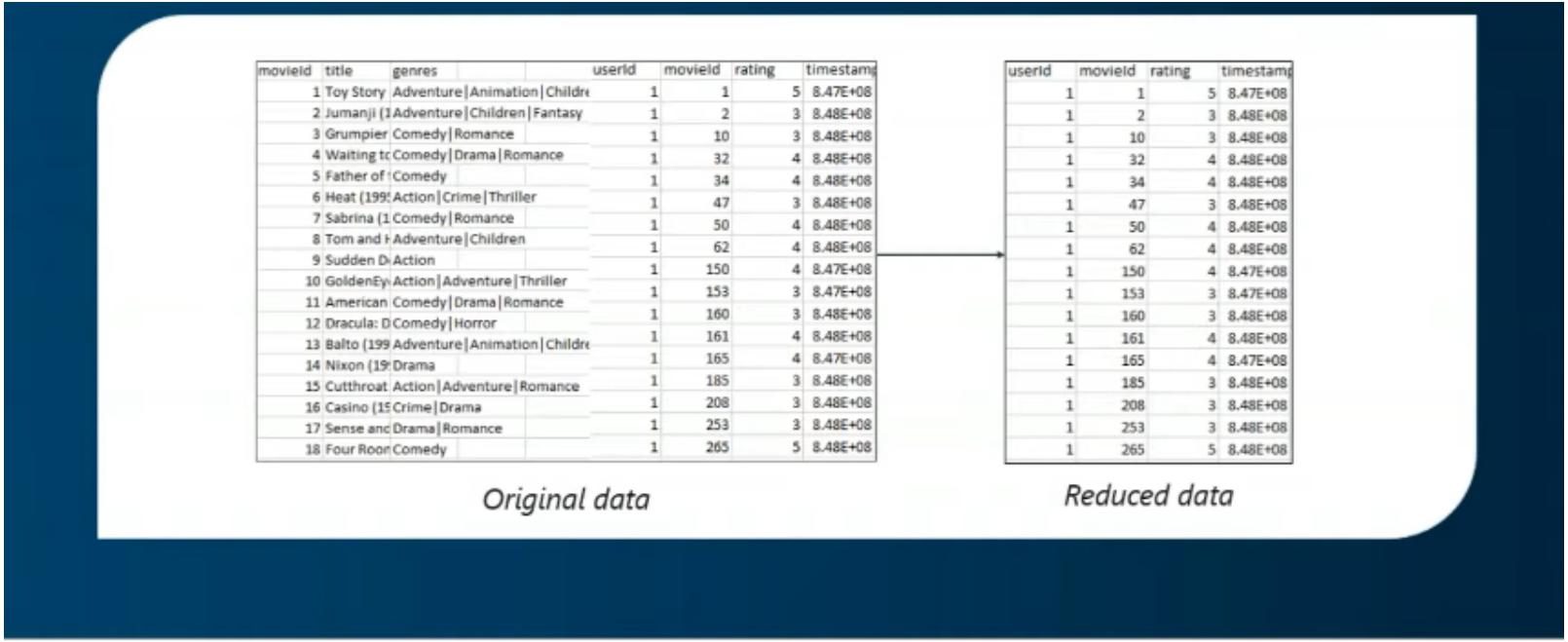
STEP 4 : Calculate Eigen values and Eigen values for Co-variance matrix



The most important eigen vector would have the director in which the variables strongly correlate

https://www.youtube.com/watch?v=ldsV0RaC9jM – Watch this video to understand calculating Eigen value & Eigen Vector.

Step 5 in PCA



movieId	title	genres	userId	movieId	rating	timestamp
1	Toy Story	Adventure|Animation|Childre	1	1	5	8.47E+08
2	Jumanji (1	Adventure|Children|Fantasy	1	2	3	8.48E+08
3	Grumpier	Comedy|Romance	1	10	3	8.48E+08
4	Waiting to	Comedy|Drama|Romance	1	32	4	8.48E+08
5	Father of	Comedy	1	34	4	8.48E+08
6	Heat (199	Action|Crime|Thriller	1	47	3	8.48E+08
7	Sabrina (1	Comedy|Romance	1	50	4	8.48E+08
8	Tom and H	Adventure|Children	1	62	4	8.48E+08
9	Sudden De	Action	1	150	4	8.47E+08
10	GoldenEye	Action|Adventure|Thriller	1	153	3	8.47E+08
11	American	Comedy|Drama|Romance	1	160	3	8.48E+08
12	Dracula: D	Comedy|Horror	1	161	4	8.48E+08
13	Balto (199	Adventure|Animation|Childre	1	165	4	8.47E+08
14	Nixon (19	Drama	1	185	3	8.48E+08
15	Cutthroat	Action|Adventure|Romance	1	208	3	8.48E+08
16	Casino (19	Crime|Drama	1	253	3	8.48E+08
17	Sense and	Drama|Romance	1	265	5	8.48E+08
18	Four Roon	Comedy				

Original data

userId	movieId	rating	timestamp
1	1	5	8.47E+08
1	2	3	8.48E+08
1	10	3	8.48E+08
1	32	4	8.48E+08
1	34	4	8.48E+08
1	47	3	8.48E+08
1	50	4	8.48E+08
1	62	4	8.48E+08
1	150	4	8.47E+08
1	153	3	8.47E+08
1	160	3	8.48E+08
1	161	4	8.48E+08
1	165	4	8.47E+08
1	185	3	8.48E+08
1	208	3	8.48E+08
1	253	3	8.48E+08
1	265	5	8.48E+08

Reduced data

Step 5: Reducing the dimensions of the data set

The last step in performing PCA is to re-arrange the original data with the final principal components which represent the maximum and the most significant information of the data set.

Step 5 in PCA

STEP 5 : The eigen vectors with highest eigen value will be the PCA.

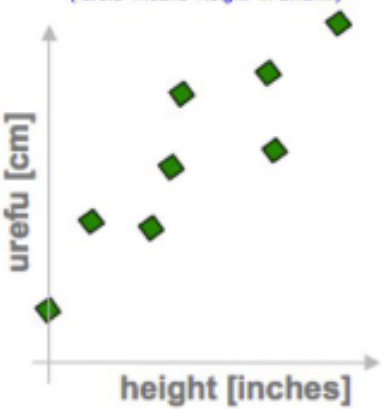
N- dimensions of Data (Features) => n Eigen Vectors

Choose P eigen vectors where p < n

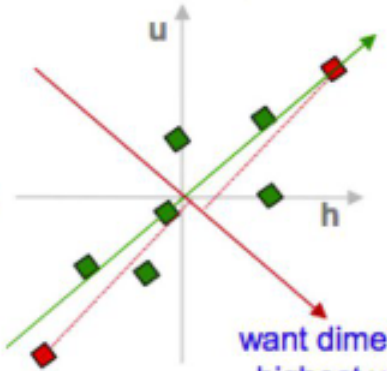
PCA in a nutshell

1. correlated hi-d data

("urefu" means "height" in Swahili)



2. center the points



want dimension of highest variance

3. compute covariance matrix

$$\begin{array}{c} \\ h \\ u \end{array}\begin{array}{c} \begin{array}{cc} h & u \end{array} \\ \begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \end{array} \quad \text{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$$

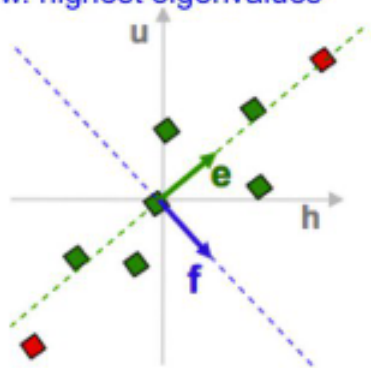
4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

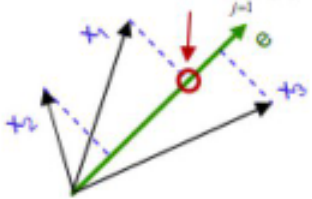
`eig(cov(data))`

5. pick m<d eigenvectors w. highest eigenvalues

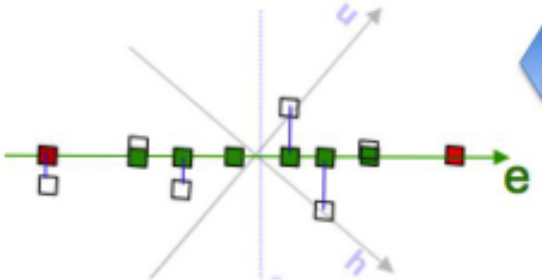


6. project data points to those eigenvectors

$$x_e' = x^T e = \sum_{j=1}^{d} x_j e_j$$



7. uncorrelated low-d data



- Steps for PCA algorithm

1. **Getting the dataset**

   Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

2. **Representing data into a structure**

   Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.

3. **Standardizing the data**

   In this step, we will standardize our dataset. Such as in a particula the features with high variance are more important compared to the features with lower variance.

   If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.

4. **Calculating the Covariance of Z**

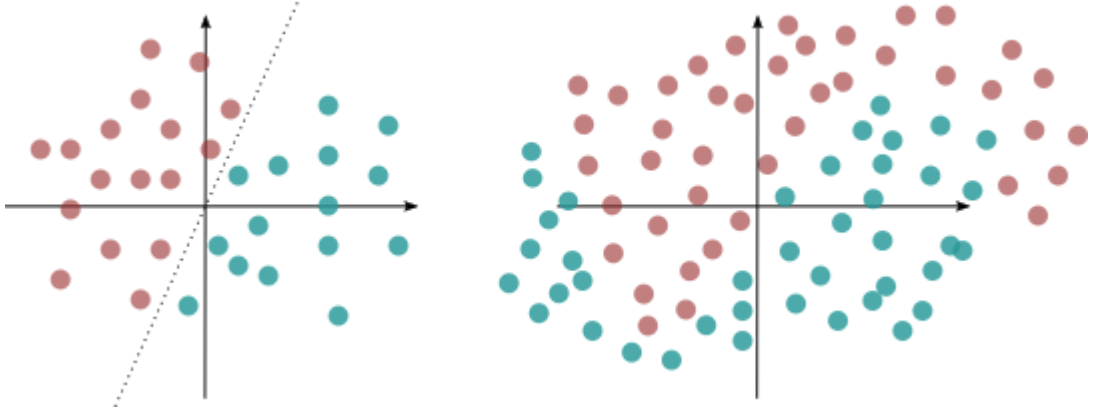
   To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.

5. **Calculating the Eigen Values and Eigen Vectors**
   Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
6. **Sorting the Eigen Vectors**
   In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.

7. **Calculating the new features Or Principal Components**
   Here we will calculate the new features. To do this, we will multip
   matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.
8. **Remove less or unimportant features from the new dataset.**
   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

Kernal Principal Components Analysis

- In general PCA transformations are linear transformations.
- The process of matrix decomposition into eigenvectors is a linear transformation. In other words, each principal component is a linear combination of the original wavelengths.



- On the left hand side the two classes are linearly separable. On the right hand side the classification boundary is more complicated, something that may look like a higher order polynomial, a non-linear function at any rate

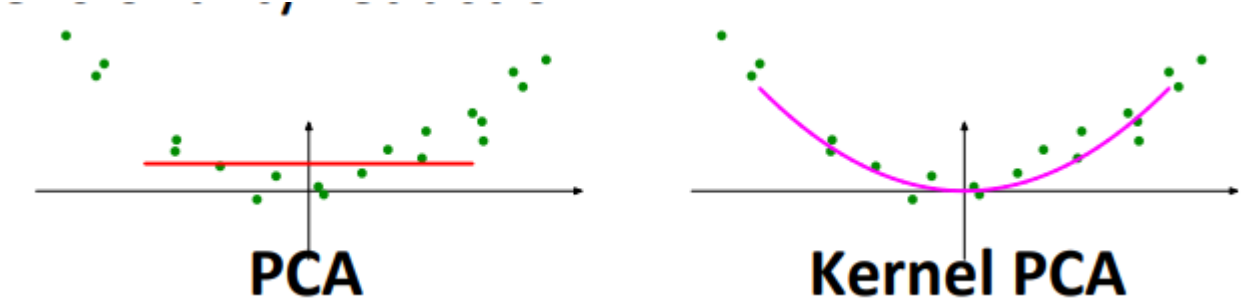
Intuition behind KPCA

- The idea of KPCA relies on the intuition that many datasets, which are not linearly separable in their space, can be made linearly separable by projecting them into a higher dimensional space. The added dimensions are just simple arithmetic operations performed on the original data dimensions.
- So we project our dataset into a higher dimensional feature space, and because they become linearly separable, then we can apply PCA on this new dataset.
- Preforming this linear dimensionality reduction in that space will be equivalent to a non-linear dimensionality reduction in the original space

Intuition behind KPCA

Example:

- consider this dataset where the brown and blue points are not linearly separable in two dimensions.
- We define the linear mapping
- T(x_1,x_2)=(z_1,z_2,z_3)=(x_1,x_2,x_1^2+x_2^2)
  This mapping function will project the data from a lower dimensional (2D) to a higher dimensional (3D) space as shown in the figure
- But doing PCA in high dimensional space will need a lot of computations, so in order to solve this problem we use kernel methods.



Steps of KPCA:

1. First we will choose a kernel functions k(x_i, x_j) and let T be any transformation to a higher dimension.

2. And like PCA, we will find the covariance matrix of our data. But here, we will use kernel function to calculate this matrix. So will compute kernel matrix, which is the matrix that results from applying kernel function to all pairs of data.

K=T(X)T(X)^T

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_d) \\ k(x_2, x_1) & k(x2, x_2) & \cdots & k(x_2, x_d) \\ \vdots & \vdots & & \vdots \\ k(x_d, x_1) & k(x_d, x_2) & \cdots & k(x_d, x_d) \end{bmatrix}$$

Steps of KPCA:

3. Center our kerenl matrix (this equivalent to substract the mean of the transformed data and dividing by standard deviations) :

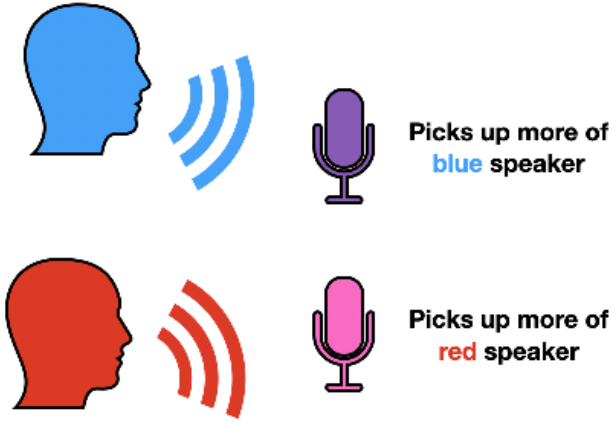
$$K\_new = K - 2(I)K + (I)K(I)$$

where I is a matrix that its all elements are equal to i/d.

4. Then, we will find eigenvectors and eigenvalues of this matrix.
5. Sort our eigenvectors based on their corrsponding eigenvalues in a decreasing order.
6. We will choose what number of dimensions that we want our reduced dataset to be, let's call it m. Then we will choose our first m eigenvectors and concatenate them in one matrix.
7. Finally, Calculate the product of that matrix with your data. The result will be your new reduced dataset.
8. There are various kernel methods like linear, polynomial, and gaussian

Independent Component Analysis

Independent Component Analysis (ICA) is a machine learning technique to separate independent sources from a mixed signal. Unlike principal component analysis which focuses on maximizing the variance of the data points, the independent component analysis focuses on independence, i.e. independent components.



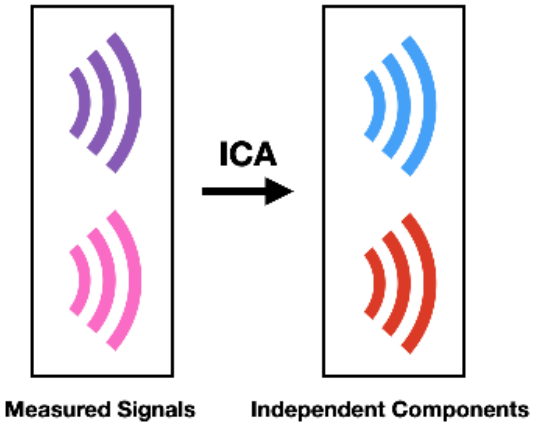
Example: Cocktail Party Problem

ICA - Example

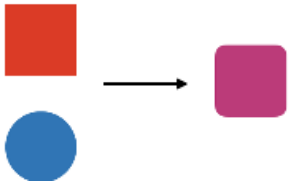
Example problem used to describe ICA is the "Cocktail Party Problem".
This problem imagine two people having a conversation at a cocktail party
The objective is to *separate the two voices to obtain isolated recordings of each speaker*
Solution:

- Two microphones placed near both party-goers (like the purple and pink microphones above).
- Both voices are heard by *both* microphones at different volumes based on the distance between the person and microphone.
- Then apply **Independent Component Analysis (ICA)** which **transforms a set of vectors into a maximally independent set**.
- The **number of inputs and outputs are the same**, and since the outputs are mutually independent there is no obvious way to drop components like in **Principal Component Analysis (PCA)**.

ICA - Example



PCA Vs ICA

Principal Component Analysis	Independent Component Analysis
It reduces the dimensions to avoid the problem of overfitting.	It decomposes the mixed signal into its independent sources' signals.
It deals with the Principal Components.	It deals with the Independent Components.
It focuses on maximizing the variance.	It doesn't focus on the issue of variance among the data points.
It focuses on the mutual orthogonality property of the principal components.	It doesn't focus on the mutual orthogonality of the components.
It doesn't focus on the mutual independence of the components.	It focuses on the mutual independence of the components.
Compress Information 	Separates Information