Principle Component Analysis (PCA)

M T W T F S S DATE:

- Our discussions of unsupervised learning have focused on clustering techniques, which seek to "discover" labels on features data that has no historical labels.
- We will now shift toward unsupervised algorithms that focus on dimension reduction.

Motivation behind Dimension Reduction:

- Imagine a dataset with 30+ features how would you understant the key features?
- Visualization and Data Analysis have limitation when the number of features dimension increases.

- Dimension Reduction Outcome:

- Understant which features describe the most variance in the dataset.
- Aid human understanding of large features sets, especially through visualization.

M T W T F S DATE:_____

- **Important Note:**
  - Dimensionality Reduction algorithm such as PCA do not simply choose a subset of the existing features.
  - They create new dimensional components that are combination of proportions of the existing features.

- **Section Overview**
  - Theory and Intuition of PCA.
  - Manually create PCA Algorithm
  - Utilize scikit learn to perform PCA.
  - PCA Exercise project overview
  - PCA Exercise Solution.

M T W T F S DATE:_____

PCA - Model 1

Theory and Intuition

part-One

- 1901 - Karl Pearson publishes" on lines and planes of closest fit to systems of points in Space" based on the pricipal axis theorm in the field of geometry - Pearson was the protege of Francis Galton and the Pearson correlation coefficient is named after it.

- 1933 - American mathematician and economist Harold Hotelling independently develops and names principal Component Analysis in the publication. "Analysis of a complex statistical variables into principal components"-

- Hotelling's paper perfectly describe the purpose of PCA:
  - Analyzing a complex set of variable into principle components-

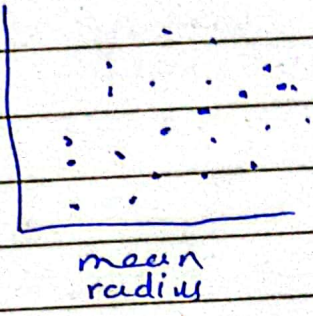
- let review the motivation behind and basic idea behind PCA

M T W T F S DATE:

• Principal Components Analysis Outcomes:

- Reduce the number of data-1
  - Show which features explain the most variance in the data.

- Dimension Reduction



- Helps visualize and understand complex dataset.
- Can also act as a simpler data set for training data for machine learning algorithms.
  - Reduce dimensions then train ML Algorithm on smaller dataset.
- Help reduce N features to a desired smaller set of components through a transformation.

- It does not simply select subset of features

• Variance Explained

- We have often seen that certain feature are more important or have more explanatory power than other features.
- For example, size of house is probably much more important than the color of a house when explaining the price of house.

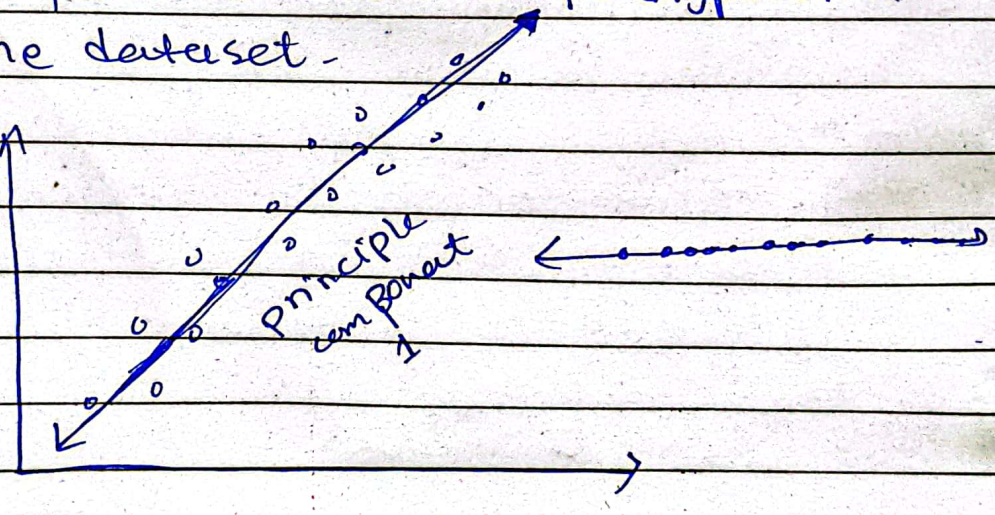
M T W T F S DATE:

The idea of more important features is easy to understand when we can directly correlates feature to a know label. But what about unlabelled data?

- What measurement we can use to determine feature "importance"?

~~Variance~~

Measure the proportion to which each features accounts for dispersion in the dataset.



- principle components is a linear combination of original features account for, the more influence it has over the principle components.
- Here we went from 2 features down to 1 principal component.
- This single principal component can "explain" some percentage of the original data, for example 90% of the variance explained by

M T W T F S DATE:_____

principal component.

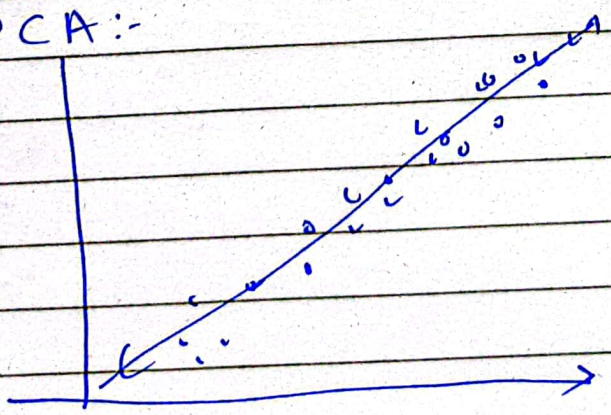
- 100% variance in the data is explain by all ~~the~~ original features.
- We trade off some of the explained variance for less dimensions.
- This can be significant saving for data set ~~by~~ with many dimensions, but only a few strong feature.

Theory and Intuition part-two

- principle Component Analysis operates by creating a new set of dimensions (the principal components) that are normalized linear combinations of the original features.

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \cdots + \phi_{p1} X_p$$

PCA:-

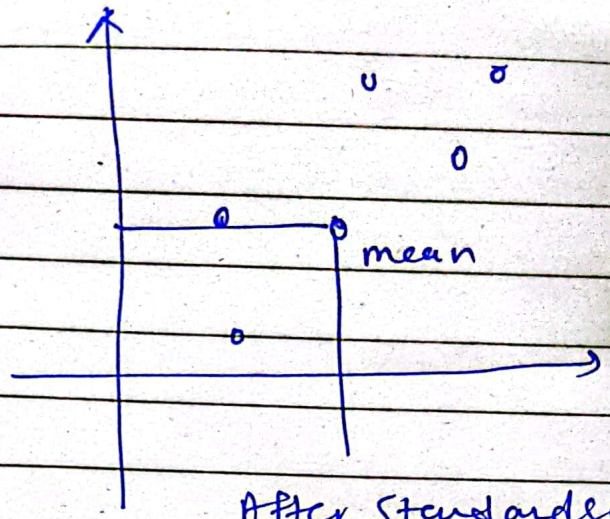


$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2$$

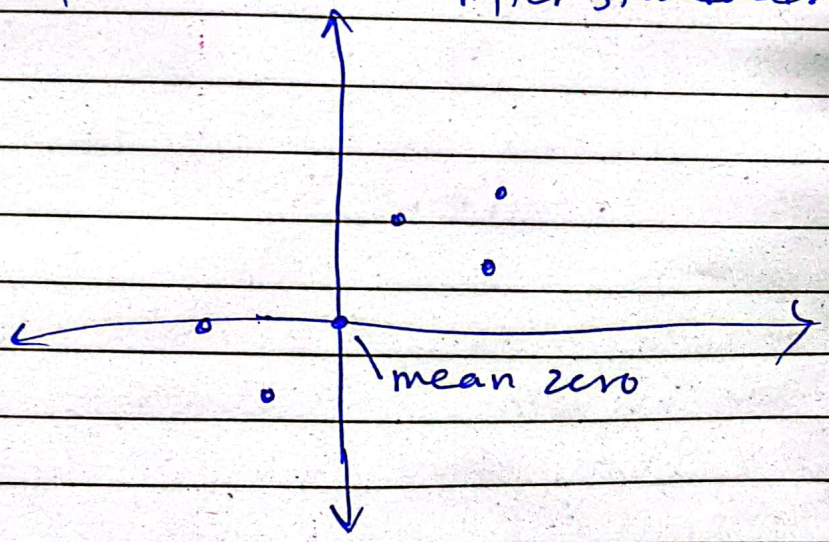
M T W T F S DATE:

Begin with two dimension dataset

X1	X2
1	2
2	1
3	2
4	4
5	3
6	5

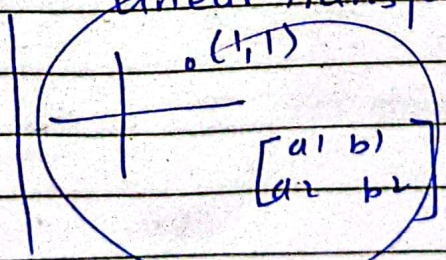


After Standardscer



- 1 step standarize the data
- calculate the covariance matrix data:

Linear Transformation

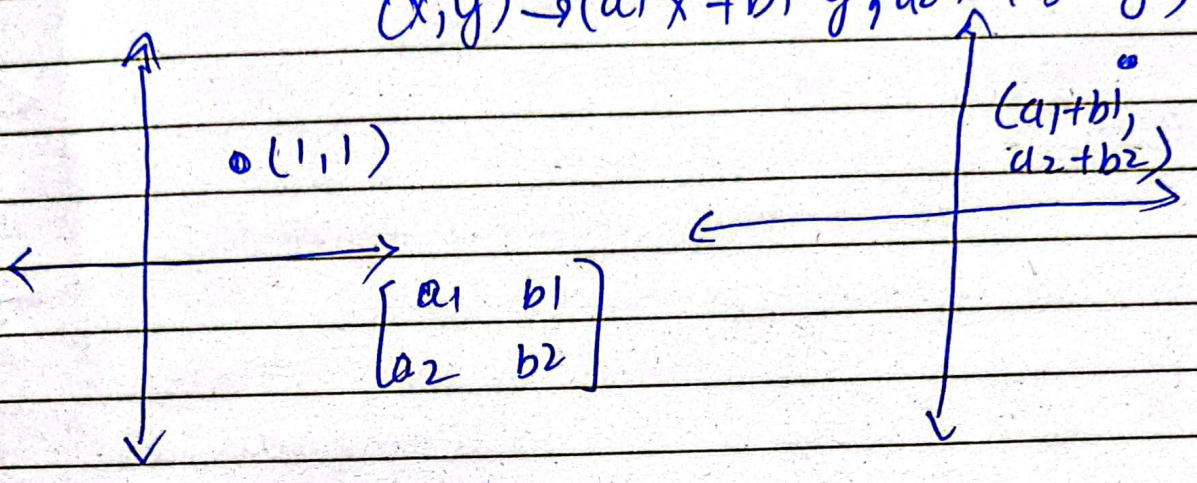


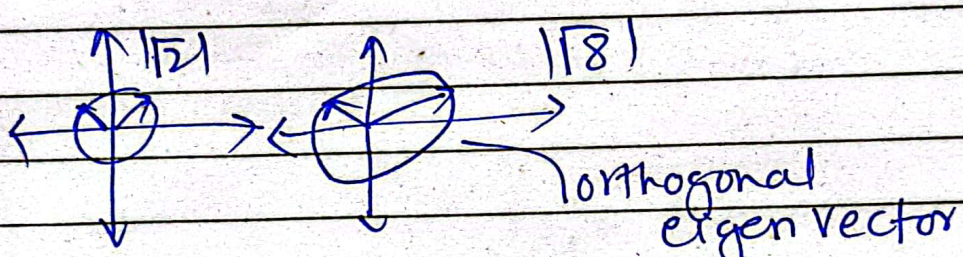
$(x,y) \rightarrow (a_1 * x + b_1 * y,\ a_2 * x, b_2 * y)$

MINDMAPS DATE:

- Linear Transformation of Data:

$(x, y) \rightarrow (a1*x + b1*y, a2*x + b2*y)$





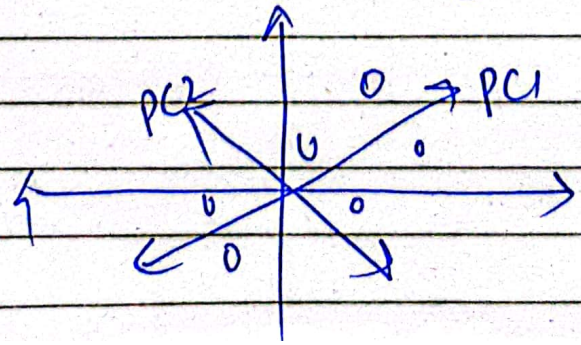
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$

- Eigen

$\begin{bmatrix} a1 & b1 \\ a2 & b2 \end{bmatrix}$

- Apply linear Transformation:

$$\begin{bmatrix} var() & cov(X,Y) \\ cov(X,Y) & var(Y) \end{bmatrix}$$



M T W T F S DATE:_____

- EigenValue measure variance explained:

- **PCA Steps**
  - Get original data
  - Calculate Covariance Matrix
  - Calculate EigenVector
  - sort EigenVectors by EigenValues
  - Choose N largest EigenValues
  - project original data onto Eigen veetor

PCA - Manual Implementation in python

df = pd.read_csv( ) cancer tumers

scaled_X = scaled.fit_transform(df)

scaled_X - scaled_X.mean(axis=0)

covariance_matrix = np.cov(scaled_X, rowvar =False)

eigen_values, eigen_vectns = np.linalg.eig (covariance matrix)

NOTES DATE:

PCA N Features Spaces ---> N PC space

num_components = 2

sorted_key = np.argsort(eigen_values)[::-1]

eigen_values, eigenvector = eigen_values[sorted_key]
[: num of components]

eigen_vectors
[:, sorted_key]

principle_comp = np.dot(scaled X, eigen_vectors)

priciple_components

plt.scatterplot(prinevpal comonents[:,0],
pneipal_components[:,1]

Scikit-learn Implemention

df = pd.read_csv(" ")

scaler = scaler.fit_transfer()

from sklearn.decomposition import PCA

help(PCA)

pca_model = PCA(n_components = 2)

PCA_model.fit(scaled_x)

pca_model.transform(scaled_x)

M T W T F S DATE:_____

```
pc = pca_model.fit_transform(scaled_x)

plt.scatter(pc_result[:,0], pc_result[:,1],
            c = cancer_dictionary['target'])

pca_model.components_

df_comp = pd.DataFrame(pca_model.components_,
                       index=['PC1','PC2'],
                       columns = df.columns)

pca_model.explained_variance_ratio

pca_30 = PCA(n_components = 30)
pca_30.fit(scaled_X)

pca_30.explained_variance_ratio

explained_variance = []
for n in range(1,30):
    pca = PCA(n_components=n)
    pca.fit(scaled_X)
```

Model deployment

M T W T F S DATE:

- Section Overview
  - General theory and concepts
  - Model persistence Basic
  - Model deployment as an API

- General theory and Concept
  - When is a model ready for deployment?
  - How often to retain your model?
  - When to revist model algorithm choice and assumption.

Model Deployment Consideration

- choosing a model
- purpose of deployment
- performance expectation
- Retrainng Intervals