PCA
Principal Component Analysis

Dr. Prashant Sing Rana

www.psrana.com

Reference Material

Videos

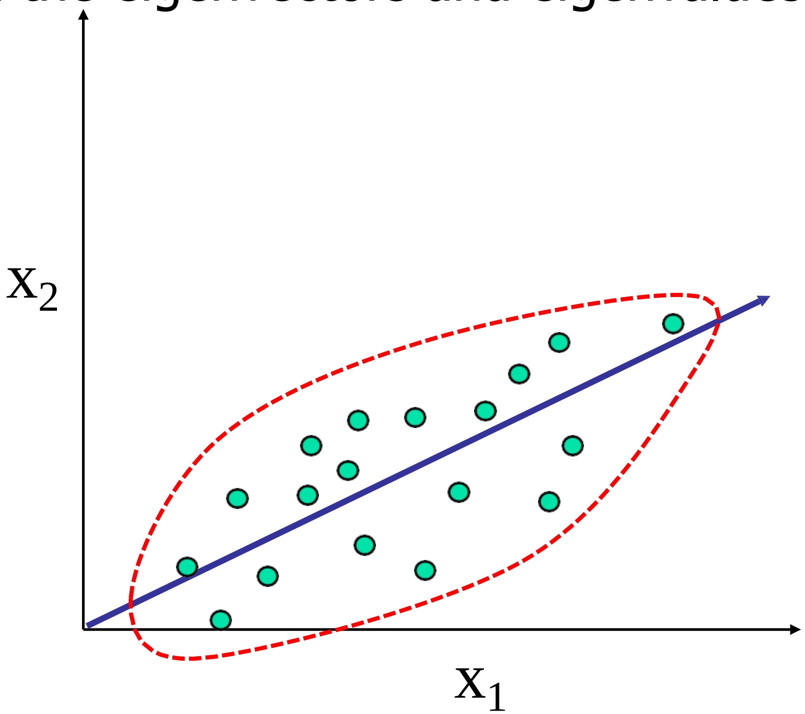
Part 1 (10 min): https://www.youtube.com/watch?v=83x5X66uWK0

Part 2 (12 min): https://www.youtube.com/watch?v=o0NNUeWNnL4

Part 3 (7 min): https://www.youtube.com/watch?v=peolsYcAxuU

Principal Component Analysis

- PCA is used in dimension reduction e.g 1000 features reduce to 10.
- Visualization is easy.
- The resultant data are projected onto a much smaller space, resulting in dimensionality reduction.
- We need to find the eigenvectors and eigenvalues.



Correlation and Covariance

Given Dataset (Two Features)

A	B
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	1.6
1	1.1
1.5	1.6
1.1	0.9

The Correlation and Covariance b/w A and B:

- **Correlation:** correl (A1:A10,B1:B10) = 0.925
- **Covariance:** covar (A1:A10,B1:B10) = 0.554
- Both the terms measure the **relationship** and the **dependency** between two variables.
- "**Covariance**" indicates the **direction** of the linear relationship between variables.
- "**Correlation**" measures both the **strength and direction** of the linear relationship between two variables.

Variance and Standard Deviation

Given Dataset (One Feature)

A
2.5
0.5
2.2
1.9
3.1
2.3
2
1
1.5
1.1

To find the relationship **within a variable** standard deviation and variance is used.

- Standard Deviation of A is 0.785 using STDEV(A1:A10)
- Variance is the square of SD.
- **Standard deviation is** a measure of the dispersion of observations within a data set relative to their **mean**.
- **Variance** describes the **variability** of observations from its mean.

PCA Example

Input:
Given Dataset (Two Features)

F1	F2
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	1.6
1	1.1
1.5	1.6
1.1	0.9

Output:
Find Principal Components

PC1, PC2

PCA Example

Step 1: Find the covariance for (x,x), (x,y) and (y,y)

$$\operatorname{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Key Points:
cov(x,y) and cov(y,x) is same.

For two attributes (x,y):

$$C = \begin{pmatrix} \operatorname{cov}(x,x) & \operatorname{cov}(x,y) \\ \operatorname{cov}(y,x) & \operatorname{cov}(y,y) \end{pmatrix}$$

For three attributes (x,y,z):

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

PCA Example

Step 2: Find the covariance for (x, x), (x, y) and (y, y)

X	Y	$(X-\bar{X})$	$(X-\bar{X})^2$	$(Y-\bar{Y})$	$(Y-\bar{Y})^2$	$(X-\bar{X})(Y-\bar{Y})$
2.5	2.4	0.69	0.476	0.49	0.24	0.338
0.5	0.7	-1.31	1.716	-1.21	1.464	1.585
2.2	2.9	0.39	0.152	0.99	0.98	0.386
1.9	2.2	0.09	0.008	0.29	0.084	0.026
3.1	3	1.29	1.664	1.09	1.188	1.406
2.3	2.7	0.49	0.24	0.79	0.624	0.387
2	1.6	0.19	0.036	-0.31	0.096	-0.06
1	1.1	-0.81	0.656	-0.81	0.656	0.656
1.5	1.6	-0.31	0.096	-0.31	0.096	0.096
1.1	0.9	-0.71	0.504	-1.01	1.02	0.717
			5.55		6.45	5.54

$\bar{X}=1.81$

$\bar{Y}=1.91$

cov (x, x) = sum($(X-\bar{X})^2$) / 9 = 5.55/9 = 0.6165

cov (y, y) = sum($(Y-\bar{Y})^2$) / 9 = 6.45/9 = 0.7165

cov (x, y) = sum($(X-\bar{X})(Y-\bar{Y})$) / 9 = 5.54/9 = 0.6154

$$C=\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

PCA Example

Step 3: Find the Eigen Value and Eigen Vector

The Eigen values (latent roots) of S are solutions (λ) to the characteristic equation given below

$$|\mathbf{S} - \lambda \mathbf{I}| = 0$$

$$C = \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{....... (1)}$$

$$\begin{bmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{bmatrix} = 0 \quad \text{........(2)}$$

$$\lambda_1 = 0.4908, \lambda_2 = 1.2840 \quad \text{........(3)}$$

(Eigen Values)

Find Eigen Vector

Put the values of eq3 in eq2 and multiply with $\begin{bmatrix} X \\ Y \end{bmatrix}$

$$\begin{bmatrix} 0.1257 & 0.6154 \\ 0.6154 & 0.2257 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = 0 \qquad \begin{bmatrix} -0.6675 & 0.6154 \\ 0.6154 & -0.5675 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = 0$$

$$0.1257X_1 + 0.6154Y_1 = 0 \qquad -0.6675X_2 + 0.6154Y_2 = 0$$

$$0.6154X_1 + 0.2257Y_1 = 0 \qquad 0.6154X_2 - 0.5675Y_2 = 0$$

$$\begin{bmatrix} -0.735 & 0.677 \\ -0.678 & -0.73 \end{bmatrix} \Leftarrow \text{Eigen Vectors for } \lambda_1 \text{ and } \lambda_2$$

PCA Example

Step 4: Find the Principal Components

Eigen Values ➡ $\lambda_1 = 0.4908, \lambda_2 = 1.2840$

Eigen Vectors for λ_1 and λ_2 ➡ $\begin{bmatrix} -0.735 & 0.677 \\ -0.678 & -0.73 \end{bmatrix}$

$\lambda_1 < \lambda_2$

so, PC1 is ➡ $-0.678 \quad -0.73$

PC2 is ➡ $-0.735 \quad 0.677$

PCA Implementation

Libraries

In [21]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

The Data

Let's work with the cancer data set again since it had so many features.

In [22]:
```
from sklearn.datasets import load_breast_cancer
```

In [23]:
```
cancer = load_breast_cancer()
```

PCA Implementation

```
In [26]: df = pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
         #(['DESCR', 'data', 'feature_names', 'target_names', 'target'])

In [27]: df.head()
```

Out[27]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...

5 rows × 30 columns

PCA Implementation

PCA Analysis

Converts any distribution to standard normal distribution where mean=0 and S.D= 1

```
In [30]: from sklearn.preprocessing import StandardScaler

In [32]: scaler = StandardScaler()
         scaler.fit(df)

Out[32]: StandardScaler(copy=True, with_mean=True, with_std=True)

In [33]: scaled_data = scaler.transform(df)
```

PCA with Scikit Learn uses a very similar process to other preprocessing functions that come with SciKit Learn. We instantiate a PCA object, find the principal components using the fit method, then apply the rotation and dimensionality reduction by calling transform().

We can also specify how many components we want to keep when creating the PCA object.

PCA Implementation

PCA Analysis

```
In [34]: from sklearn.decomposition import PCA

In [35]: pca = PCA(n_components=2)

In [36]: pca.fit(scaled_data)

Out[36]: PCA(copy=True, n_components=2, whiten=False)
```

PCA Implementation

In [36]:
```
pca.fit(scaled_data)
```
Out[36]:
```
PCA(copy=True, n_components=2, whiten=False)
```

Now we can transform this data to its first 2 principal components.

In [37]:
```
x_pca = pca.transform(scaled_data)
```

In [38]:
```
scaled_data.shape
```
Out[38]:
```
(569, 30)
```

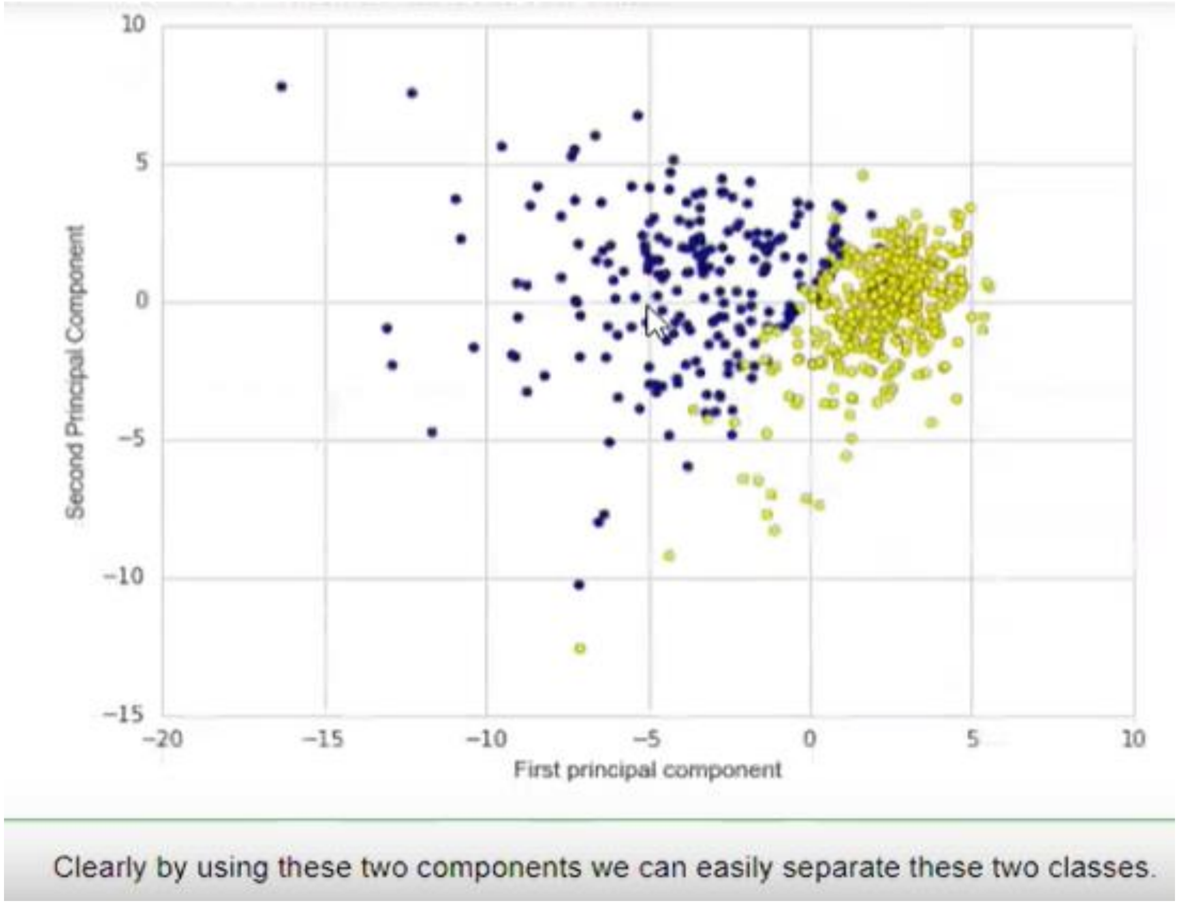
In [39]:
```
x_pca.shape
```
Out[39]:
```
(569, 2)
```

Great! We've reduced 30 dimensions to just 2! Let's plot these two dimensions out!

In [52]:
```
plt.figure(figsize=(8,6))
plt.scatter(x_pca[:,0],x_pca[:,1],c=cancer['target'],cmap='plasma')
plt.xlabel('First principal component')
plt.ylabel('Second Principal Component')
```

PCA Implementation

Visualization



Thanks

Learning by Doing