GHARDA INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Machine Learning Lab BE Computer (Semester-VII)

Experiment No.6: Data Visualization

Aim- To study, understand and implement data visualization using the PCA algorithm.

Theory-

Principal Component Analysis is basically a statistical procedure to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.

Each of the principal components is chosen in such a way so that it would describe most of them still available variance and all these principal components are orthogonal to each other. In all principal components, the first principal component has a maximum variance.

Uses of PCA:

- It is used to find inter-relation between variables in the data.
- It is used to interpret and visualize data.
- The number of variables is decreasing and it makes further analysis simpler.
- It's often used to visualize genetic distance and relatedness between populations.

These are basically performed on a square symmetric matrix. It can be a pure sum of squares and cross-products matrix or Covariance matrix or Correlation matrix. A correlation matrix is used if the individual variance differs much.

Objectives of PCA:

- It is basically a non-dependent procedure in which it reduces attribute space from a large number of variables to a smaller number of factors.
- PCA is basically a dimension reduction process but there is no guarantee that the dimension is interpretable.
- The main task in this PCA is to select a subset of variables from a larger set, based on which original variables have the highest correlation with the principal amount.

Principal Axis Method: PCA basically searches a linear combination of variables so that we can extract maximum variance from the variables. Once this process completes it removes it and searches for another linear combination that gives an explanation about the maximum proportion of remaining variance which basically leads to orthogonal factors. In this method, we analyze total variance.

Eigenvector: It is a non-zero vector that stays parallel after matrix multiplication. Let's suppose x is an eigenvector of dimension r of matrix M with dimension r*r if Mx and x are parallel. Then we need to solve Mx=Ax where both x and A are unknown to get eigenvectors and eigenvalues.

Under Eigen-Vectors we can say that Principal components show both common and unique variance of the variable. Basically, it is variance focused approach seeking to reproduce total variance and correlation with all components. The principal components are basically the linear combinations of the original variables weighted by their contribution to explain the variance in a particular orthogonal dimension.

Eigenvalues: It is basically known as characteristic roots. It basically measures the variance in all variables which is accounted for by that factor. The ratio of eigenvalues is the ratio of explanatory importance of the factors with respect to the variables. If the factor is low then it is contributing less to the explanation of variables. In simple words, it measures the amount of variance in the total given database accounted for by the factor. We can calculate the factor's eigenvalue as the sum of its squared factor loading for all the variables.

PCA for Data Visualization

For a lot of machine learning applications, it helps to be able to visualize our data. Visualizing 2 or 3-dimensional data is not that challenging. However, even many datasets used are multi-dimensional because of multiple features. We can use PCA to reduce that multi-dimensional data into 2 or 3 dimensions so that we can plot and hopefully understand the data better.

Steps involved

1. Load the dataset
2. Standardize the data
3. Calculate principal components for a given dataset
4. Visualize 2D projection
5. Calculate the variance of each principal component

Code -

```
import pandas as pd
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
# load dataset into Pandas DataFrame
df = pd.read_csv(url, names=['sepal length','sepal width','petal length','petal width','target'])
from sklearn.preprocessing import StandardScaler
features = ['sepal length', 'sepal width', 'petal length', 'petal width']
# Separating out the features
x = df.loc[:, features].values
# Separating out the target
y = df.loc[:,['target']].values
# Standardizing the features
x = StandardScaler().fit_transform(x)
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents
             , columns = ['principal component 1', 'principal component 2'])

finalDf = pd.concat([principalDf, df[['target']]], axis = 1)

Import matplotlib.pyplot as plt
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('2 component PCA', fontsize = 20)
targets = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
colors = ['r', 'g', 'b']
for target, color in zip(targets,colors):
   indicesToKeep = finalDf['target'] == target
                    ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1'],
finalDf.loc[indicesToKeep, 'principal component 2'], c = color, s = 50)
ax.legend(targets)
ax.grid()
Pca.explained_variance_ratio_
```

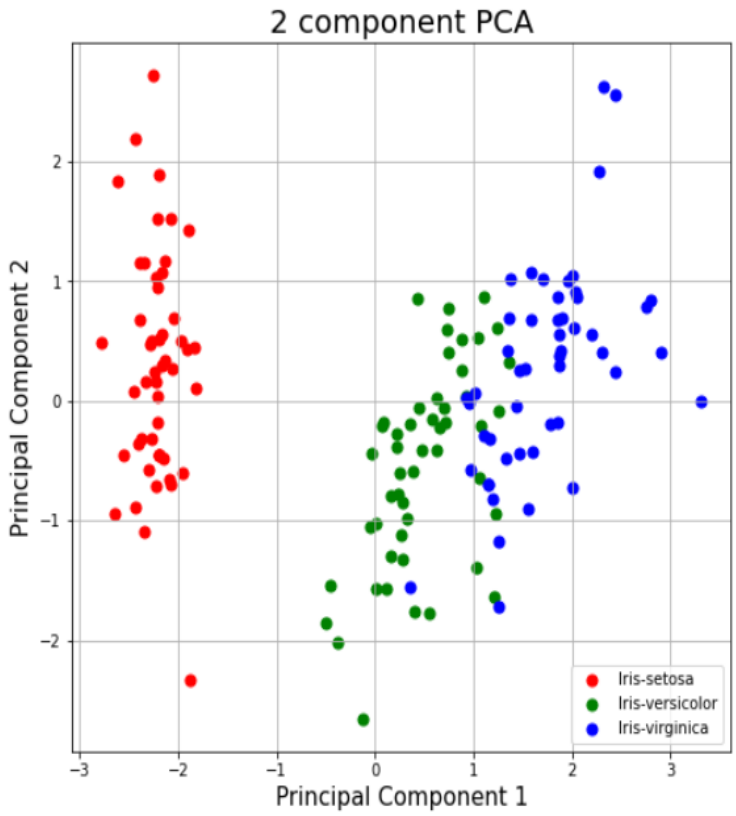
Results

```
import pandas as pd
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
# load dataset into Pandas DataFrame
df = pd.read_csv(url, names=['sepal length','sepal width','petal length','petal width','target'])
from sklearn.preprocessing import StandardScaler
features = ['sepal length', 'sepal width', 'petal length', 'petal width']
# Separating out the features
x = df.loc[:, features].values
# Separating out the target
y = df.loc[:,['target']].values
# Standardizing the features
x = StandardScaler().fit_transform(x)
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents
            , columns = ['principal component 1', 'principal component 2'])

finalDf = pd.concat([principalDf, df[['target']]], axis = 1)
```

```
impor
fig = plt.figure(figsize
ax = fig.add_subplot(1,1,
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('2 component PCA', fontsize = 20)
targets = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
colors = ['r', 'g', 'b']
for target, color in zip(targets,colors):
   indicesToKeep = finalDf['target'] == target
   ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1'], finalDf.loc[indicesToKeep, 'principal component 2'], c = color, s = 50)
ax.legend(targets)
ax.grid()
pca.explained_variance_ratio_

# last output means -> these two values are varience of each component
# addition of these two variables means -> there is
```

```
array([0.72770452, 0.23030523])
```

```
array([0.72770452, 0.23030523])
```



Discussion