Weekly Report 3 - PCA

Ganji Varshitha
AI20BTECH11009

Introduction

PCA is a data-visualization and data-preprocessing technique for training data before learning a machine learning model. It comes under dimensionality reduction.

Algorithm

It can be seen as projecting the original data point vector into a lower dimensional space where information loss is minimum. Let us assume projection of $\mathbf{x}$ in the direction of $\mathbf{w}$ be $\mathbf{z} = \mathbf{w}^\top \mathbf{x}$.
Maximising variance of $\mathbf{z}$ gives $\mathbf{w}$ such that all principal components are orthogonal to each other and $\|w\| = 1$
Let X be matrix where rows corresponding to data points and columns corresponding to features

Algorithm 1 PCA algorithm

```
Standardize the features with 0 mean and 1 variance. Call the new matrix as Z.
Calculate covariance of Z.
Calculate the eigenvalues and eigenvectors of covariance matrix of Z.
Sort the eigenvectors in decreasing order of their corresponding eigenvalues.
Select the optimal number of eigenvectors, say K.
Let P be a matrix with each column corresponds to sorted K eigenvectors. The reduced dataset matrix becomes transform of $(ZP)^\top$
```

Code Snippets

```python

def standardize(data):
  dataset = data.values
  A = dataset.T
  V = np.empty((A.shape))
  for i in range(A.shape[0]):
    for j in range(A.shape[1]):
      V[i,j] = (A[i,j] - np.mean(A[i]))/np.std(A[i])
  return V.T
```

```
# Standardizing the data
X = standardize(data)
# Calculating the covariance matrix
cov = np.cov(X.T)
# Finding eigenvalues and eigenvectors of cov matrix
eigen_values, eigen_vectors = np.linalg.eigh(cov)
print("Eigenvalues:\n",eigen_values)
print("Eigenvectors:\n",eigen_vectors)
```

```
# Printing principal components
print("First principal component: \n",eigen_vectors[:,2])
print("Second principal component: \n",eigen_vectors[:,1])
print("Third principal component: \n",eigen_vectors[:,0])
```

```
# Using two first principal components
projmatrix = np.zeros((3,2))
projmatrix[:,0] = eigen_vectors[:,0]
projmatrix[:,1] = eigen_vectors[:,2]

# Reducing dataset to 2 dimensions:
X_red = (X@projmatrix).T
print(X_red)
```

Key points

- It assumes correlation among the features. If the features are not correlated the algorithm may not be useful
- Standardizing the features is necessary as the algorithm is sensitive to the scale.
- It assumes linear relationship between features and does not work well for non linear relationships.
- It is a statistical process which acts as a tool in EDA(Exploratory data analysis).
- PCA is not robust to outliers.
- Since the reduced data matrix has features as linear combination of original features, it lacks interpret-ability.
- It helps in increasing model performance for high dimensional data regression problems and prevents over fitting.