CSE 601: Data Mining and Bioinformatics

Project 1: PCA & Apriori Algorithm

Principal Component Analysis (PCA) Report

By

Shri Sai Sadhana Natarajan (50247664)

Sneha Parshwanath (50248890)

Soumya Venkatesan (50246599)

Introduction:

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of correlated dimensions into a set of linearly uncorrelated dimensions called principal components. It is the most common form of factor analysis.

The transformation is defined in such a way that first principal component is the direction of greatest variability (covariance) in the data, second is the next orthogonal (uncorrelated) direction of greatest variability and so on. The basic concept of PCA is to combine related dimensions, and focus on uncorrelated or independent ones, especially those along which the data have high variance

Implementation:

1. The data is read from the given input file and split into two lists one containing the diseases data (which is the last column in the dataset) and other containing the features data for the diseases (all columns in the dataset leaving the last).
2. PCA is implemented using the following steps:
    a. Mean vector is calculated by taking the mean of all rows.
    b. The original data is centred by subtracting the mean from them.
    c. Covariance matrix is computed by multiplying the mean centred data with its transpose and dividing the product by total length
    d. From the covariance matrix, eigen values and eigen vectors are obtained by using eig() function from linalg package under numpy library
    e. The eigen values are the sorted in decreasing order and the top two eigen values are taken (since we are reducing to two dimensions).
    f. The two eigen vectors corresponding to the top two eigen values are taken.
    g. These represent the direction with highest variance and the dimensionally reduced data is computed by taking the product of the top two eigen vectors and the original data.
3. SVD is implemented using TruncatedSVD() from sklearn.decomposition library on the original data. The number of components is set to two in order to fit the results in two dimensional space.
4. TSNE is implemented using TSNE() from sklearn.manifold library on the original data. The number of components is set to two in order to fit the results in two dimensional space. To obtain well defined clusters, learning rate is set to 100 and initial embedding is set to ‘pca’ since it is more globally stable.
5. Finally, the new dimensions obtained using the three algorithms is visualised using scatter plot.

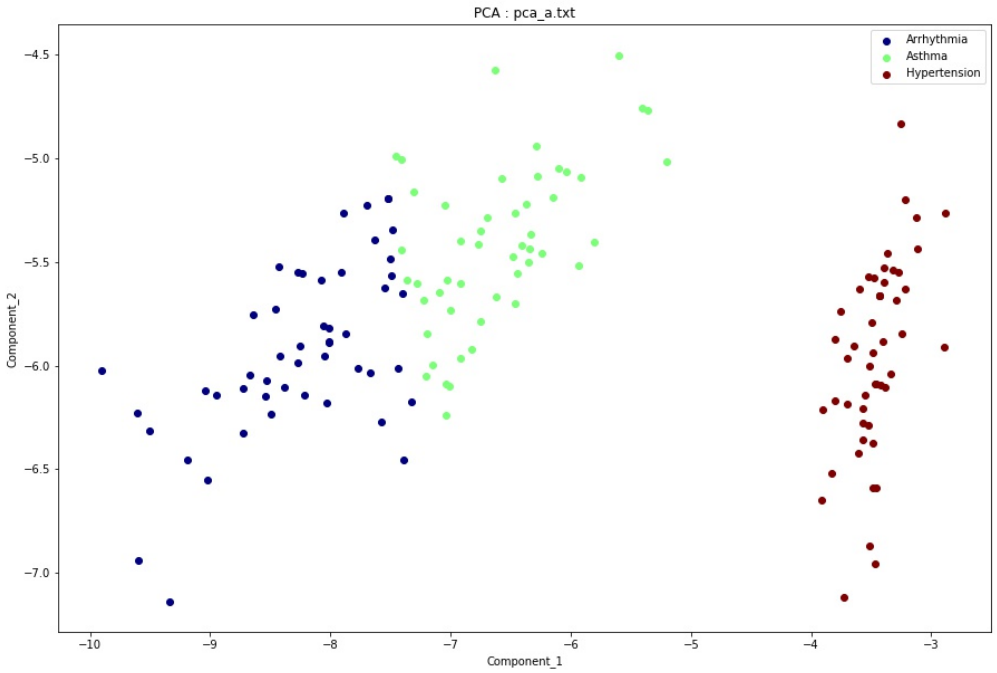
Results:

The scatter plots obtained after passing the datasets pca_a, pca_b and pca_c through the dimensionality reduction algorithms PCA, SVD and TSNE are as shown

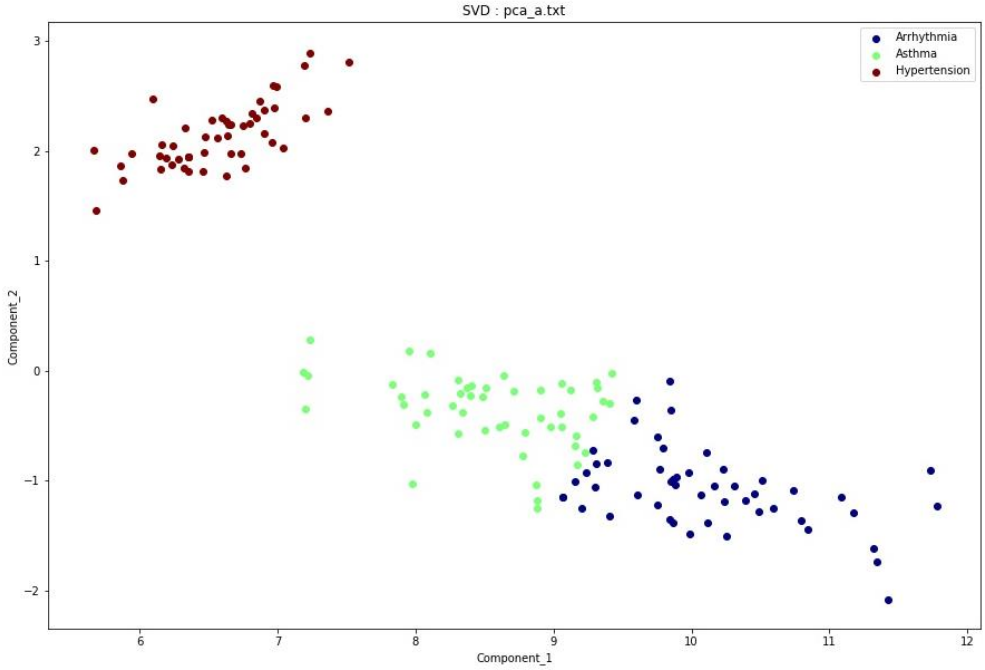
- pca_a contains 150 health record with three different diseases Arrhythmia, Asthma and Hypertension
- pca_b contains 386 health record with five different diseases CA, CD, HVD, Septic and TB
- pca_c contains 428 health record with seven different diseases ALL, AML, Brest Cancer, COPD, Colen Cancer, Diabetes and Obesity

For Dataset: pca_a

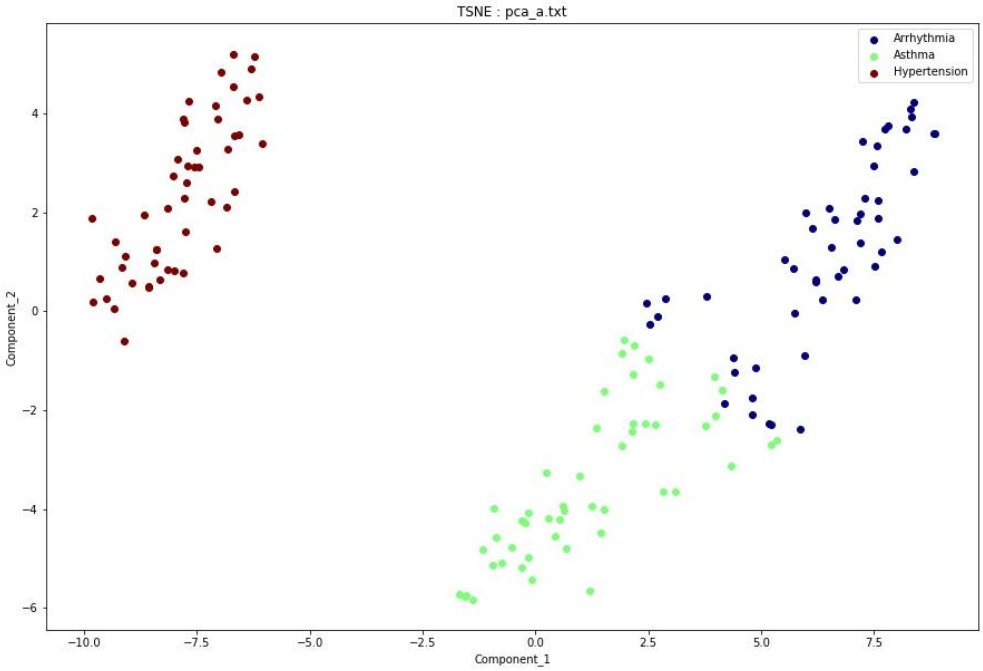
1. PCA



2. SVD

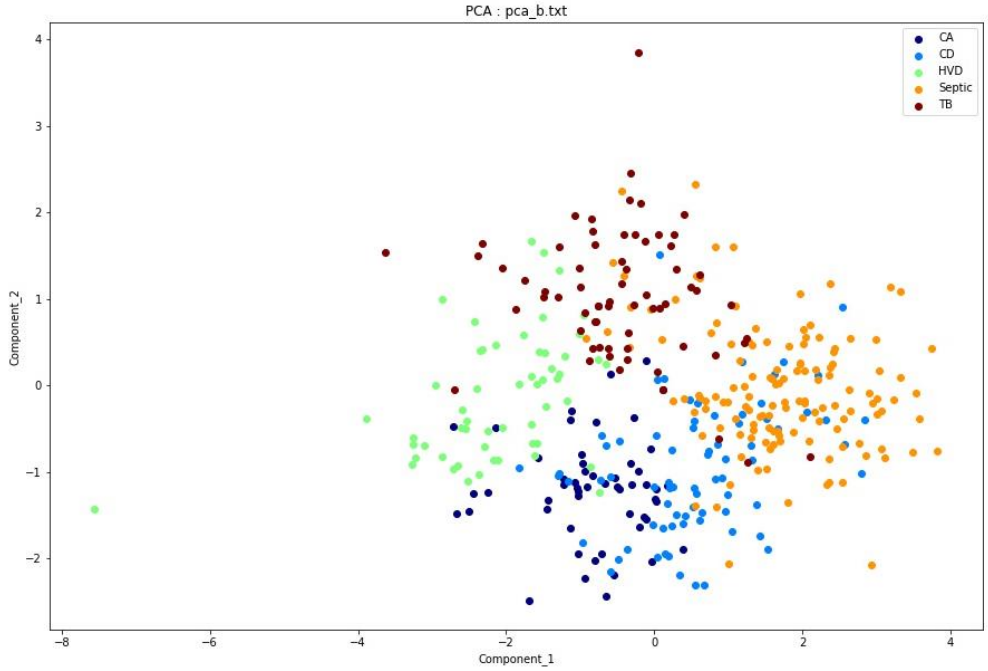

SVD : pca_a.txt
Arrhythmia
Asthma
Hypertension
Component_2
Component_1
3
2
1
0
−1
−2
6
7
8
9
10
11
12

3. TSNE

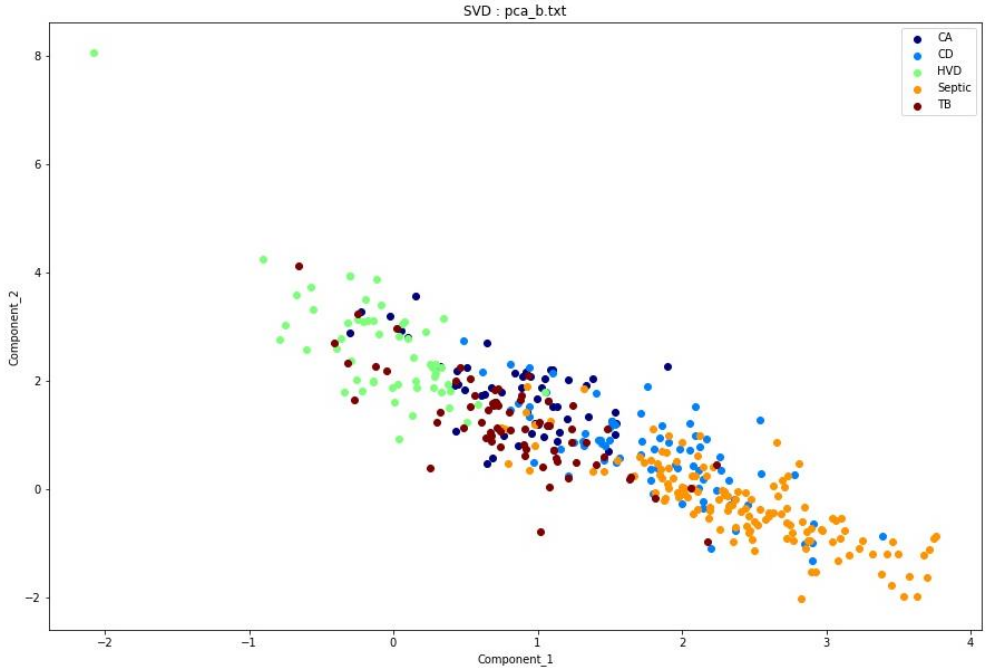

TSNE : pca_a.txt
Arrhythmia
Asthma
Hypertension
Component_2
Component_1
4
2
0
−2
−4
−6
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

For Dataset: pca_b

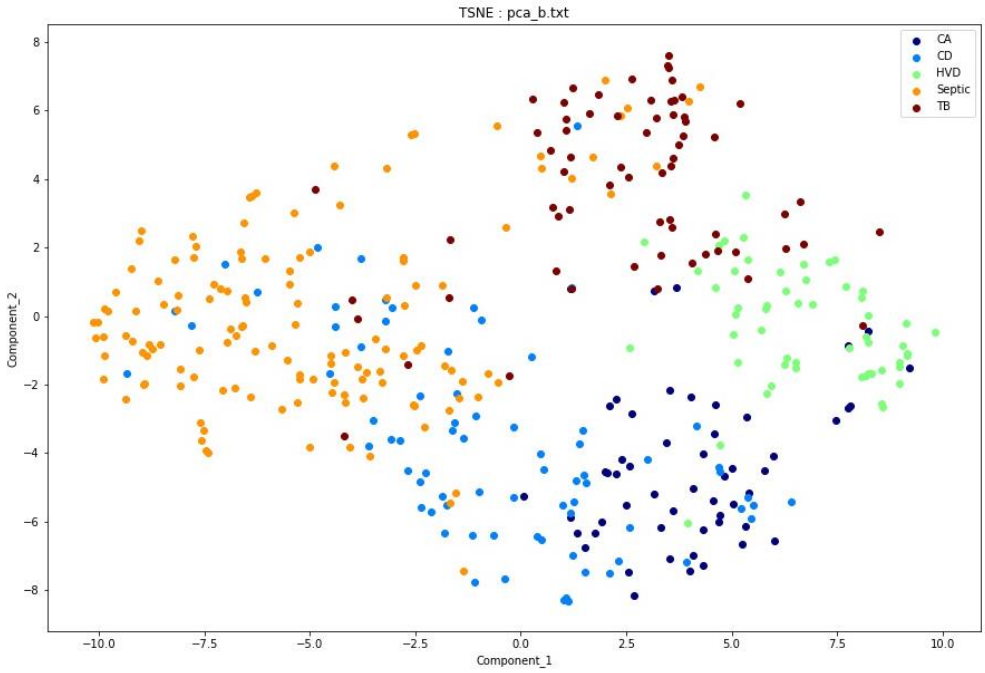
1. PCA



2. SVD

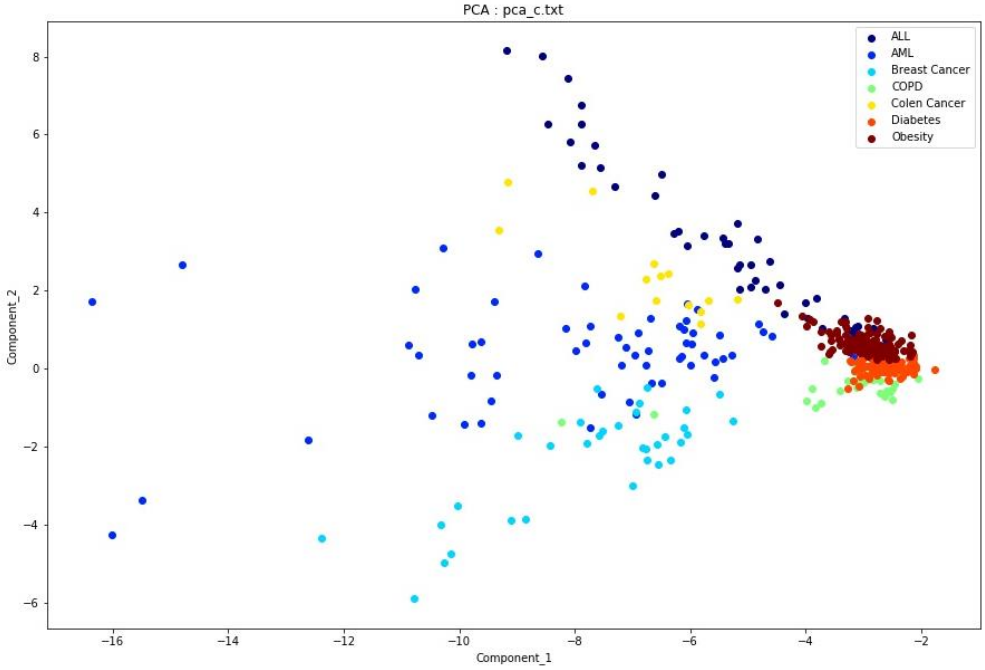


3. TSNE

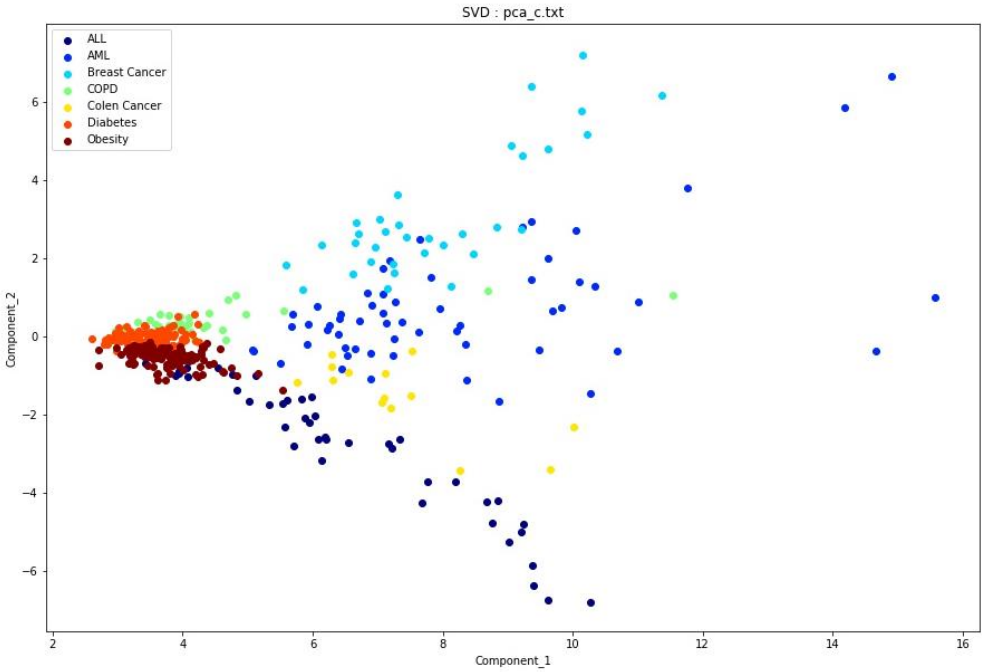


For Dataset: pca_c

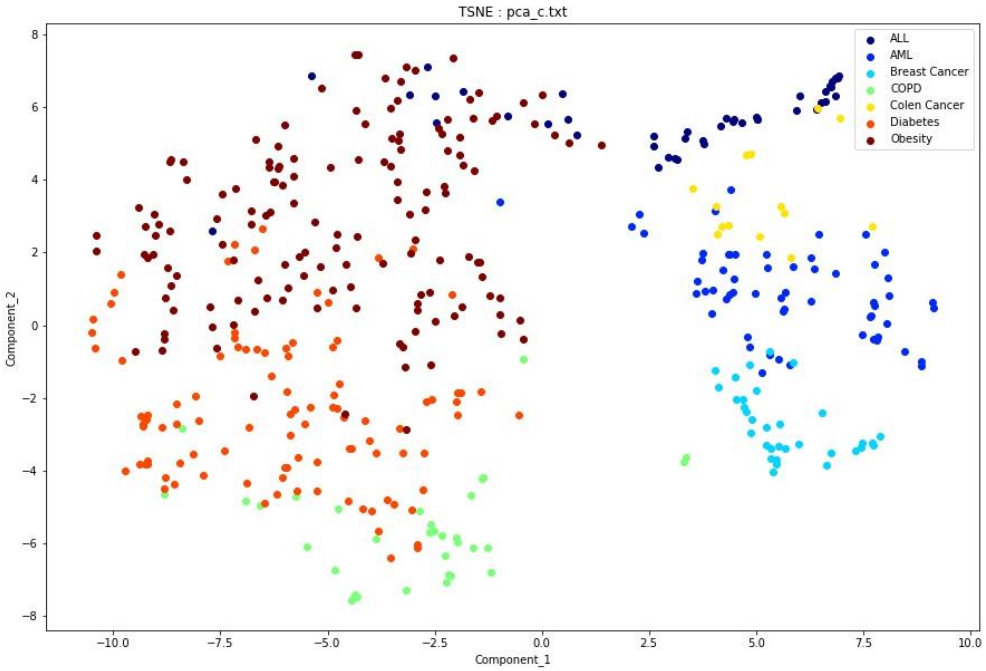
1. PCA



2. SVD


SVD : pca_c.txt
ALL
AML
Breast Cancer
COPD
Colen Cancer
Diabetes
Obesity
Component_2
Component_1

3. TSNE


TSNE : pca_c.txt
ALL
AML
Breast Cancer
COPD
Colen Cancer
Diabetes
Obesity
Component_2
Component_1

Analysis:

PCA measures the deviation of original data from the mean whereas SVD measures it from zero. TSNE performs dimensionality reduction using probability distribution.

It can be observed from the plots that PCA and SVD are almost similar whereas TSNE is not. This is because PCA and SVD use almost the similar technique for dimensionality reduction. If SVD is performed on mean centred data instead of original data, it will give the same result as that of PCA.

For low dimensional data like pca_a dataset, PCA and SVD give better distinction between the data points compared to TSNE. But when it comes to high dimensional data like pca_c dataset, TSNE gives better distinction between data points and clear visualisation compared to PCA and SVD.

Thus, based on the observation it can be concluded that PCA and SVD work good for low dimensional data whereas TSNE works good on high dimensional data.

References:

- ✓ Lecture Slide: PCA
- ✓ http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html
- ✓ http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
- ✓ https://matplotlib.org/api/_as_gen/matplotlib.pyplot.scatter.html
- ✓ https://matplotlib.org/users/colormaps.html