CSE 601: Data Mining and Bioinformatics
Project 1 - Part B:

Dimensionality Reduction

PCA

SVD

t-SNE

University at Buffalo

Prashanth Seralathan #50204883
Manishkumarreddy Jarugu #50206843
Haneesh Reddy Poddutoori #50208280

Dimensionality Reduction Algorithm:

PCA Algorithm:

PCA deals with dimensionality reduction, In general the high-dimensional data provides the best accurate result, but if the data is closely observed and if the correlation is taken into account the same data can be represented in much low dimensional space that still preserves most of the variation in the data.

Such a dimension is called as *"Principal Components"* - The components which are best suitable for representing the entire data in a much smaller dataspace.

Below are the steps involved in implementation of PCA algorithm,

1. Consider a D-dimensional data set.
2. Compute the mean vector of length D for the given data set.
3. Adjust the original data set by subtracting each row with the mean vector.
4. Compute the covariance matrix for the adjusted data set.
5. Compute the eigenvalues and eigenvectors of the covariance matrix.
6. Sort the eigenvectors based on the eigenvalues in decreasing order.
7. Consider the first K eigenvectors to reduce the original data to K dimensions.

SVD Algorithm:

SVD algorithm works on the concept of representing the original data matrix in the form of product of three matrices - Two orthogonal matrices and one diagonal matrix D.

$$X = UDV^T$$

The speciality of this representation is that the matrix U represents the eigenvectors and the matrix D represents the eigenvalues. So when we want to pick out the components that represent the maximum variance(Say first 2 principal components) we shrink the matrices in such values are represented along those eigenvectors/eigenvalues pair.

Considering the relation between PCA and SVD, PCA can be interpreted as Singular value decomposition of a data matrix where the columns have been centered by their means.

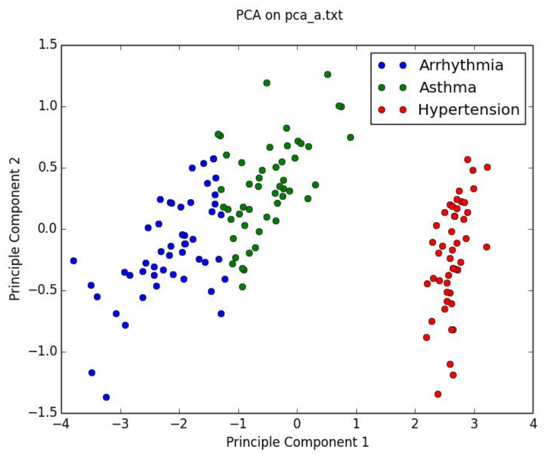
<u>t-SNE Algorithm:</u>

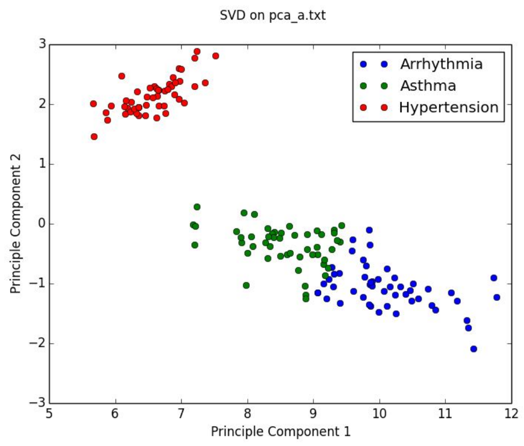
t-Distributed Stochastic Neighbor Embedding (t-SNE) is another technique for dimensionality reduction which uses probabilistic technique. It works on the distribution that measures the pairwise similarities between the input objects in such a way to create neighbor embedding.

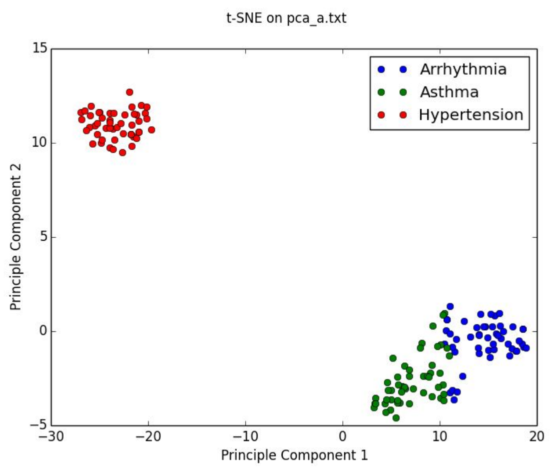
t-SNE works it's way on how to represent the data using less dimensions that matches multiple distributions of data that are present in the input - This t-SNE would provide good clusters of data grouped together. The t-SNE works based on a gradient descent and thus if the number of iterations stop early or when the convergence fails to happen we might not get proper plots that contains the clusters of data.

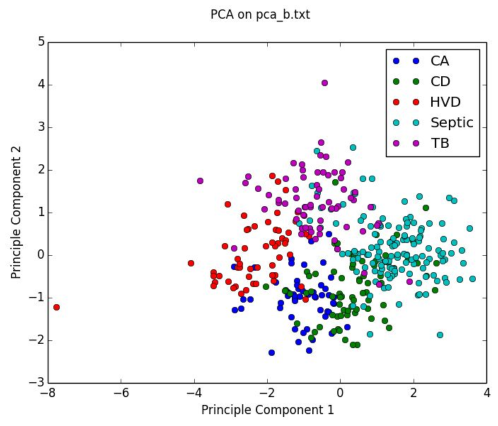
t-SNE is computationally expensive and has limitations when it comes to using large datasets. Thus at times PCA is used to reduce the dimensions before giving it to t-SNE. In our case particularly when trying to plot the pca_a.txt that contains just three diseases we get a clear segregation of data.

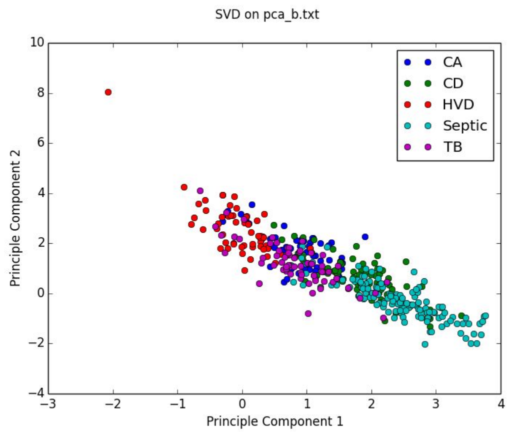
<u>Plots:</u>

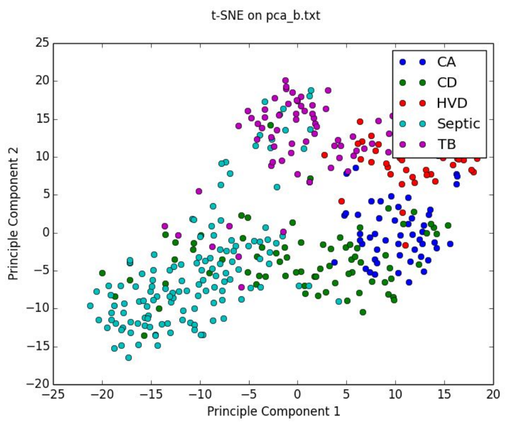


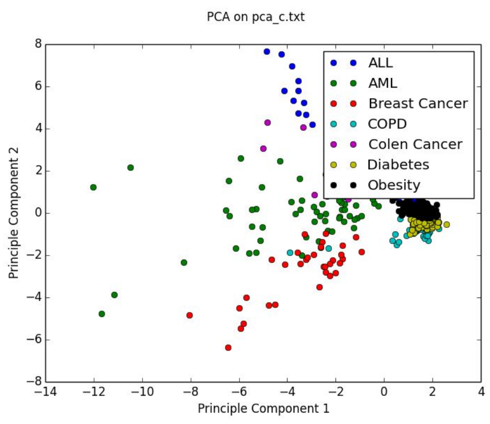


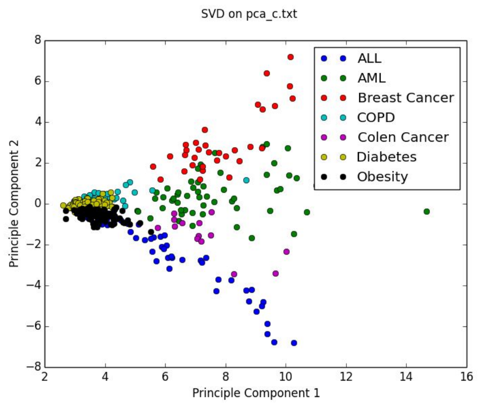

t-SNE on pca_a.txt
Arrhythmia
Asthma
Hypertension
Principle Component 1
Principle Component 2
15
10
5
0
−5
−30
−20
−10
0
10
20

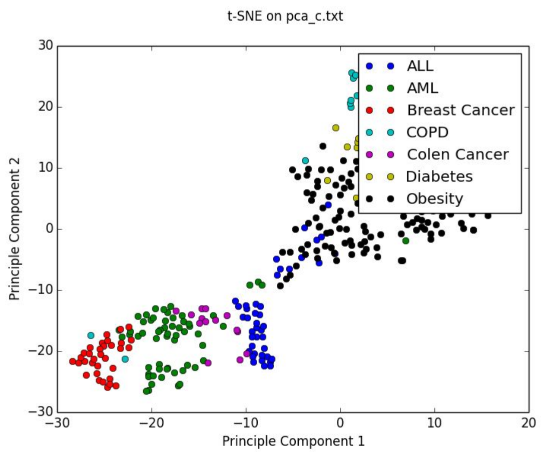

PCA on pca_b.txt
CA
CD
HVD
Septic
TB
Principle Component 1
Principle Component 2
5
4
3
2
1
0
−1
−2
−3
−8
−6
−4
−2
0
2
4


SVD on pca_b.txt
CA
CD
HVD
Septic
TB
Principle Component 1
Principle Component 2
10
8
6
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
4


t-SNE on pca_b.txt
CA
CD
HVD
Septic
TB
Principle Component 1
Principle Component 2
25
20
15
10
5
0
−5
−10
−15
−20
−25
−20
−15
−10
−5
0
5
10
15
20







In our data inputs, t-SNE gave best results when the data is represented in a lower dimension as the data seemed nicely clustered in groups, but considering the fact that it's computationally expensive to calculate t-SNE it might be a better idea to initially reduce the dimension using PCA before and pass on the reduced dimensions to the t-SNE.

References:

1) http://bigdata-madesimple.com/decoding-dimensionality-reduction-pca-and-svd/
2) https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b