CSE 564 - Visualization: Lab Report 2

April 5, 2017

1 Introduction

In this assignment, I have attempted to research an interesting problem - Which factors most influence the world ranking of a university. I have used the CWUR - 2015 dataset, attached in the submission folder. The data has 1000 rows for universities and has over 11 attributes such as National rank, publications, citations, quality of education etc.

2 How to run:

The main file to open in browser after creating a local server is assignment2.html. All other HTML files will be called from within the assignment2.html file Enter the following command into the command line:
python -m SimpleHTTPServer 8000
This command creates a local host at port 8000. Simple open this port in the webbrowser by navigating to
localhost:8000
and click on assignment2.html to view the project.

3 Task 1: Data Reduction

In this task, I implemented random and stratified sampling to reduce the number of data rows and retain a subset for easier data handling.

3.1 Random Sampling

Figure 1 shows the code snippet used to implement random sampling. The data reduced from 1000 rows to 440 rows.

3.2 Adaptive Sampling

This technique is more complicated and involved running kmeans on the dataset to form clusters and then pick datapoints proportional to the cluster size. Figure 2 shows the code

```
def random_sampling(data_frame, fraction):
    rows = random.sample(data_frame.index, (int)(len(data_frame)*fraction))
    print len(rows)
    return data_frame.ix[rows]
```

Figure 1: Code snippet for Random Sampling

for running K-means on the dataset and Figure 3 shows the elbow curve generated. According to this, the optimum cluster size is chosen to be 3.

```
K = range(1,10)
KM = [KMeans(n_clusters=k).fit(inputData) for k in K]
centroids = [k.cluster_centers_ for k in KM]
D_k = [cdist(inputData, cent, 'euclidean') for cent in centroids]
cIdx = [np.argmin(D,axis=1) for D in D_k]
dist = [np.min(D,axis=1) for D in D_k]
avgWithinSS = [sum(d)/inputData.shape[0] for d in dist]
wcss = [sum(d**2) for d in dist]
tss = sum(pdist(inputData)**2)/inputData.shape[0]
bss = tss-wcss
kIdx = 3-1
```

Figure 2: Code snippet for K-means

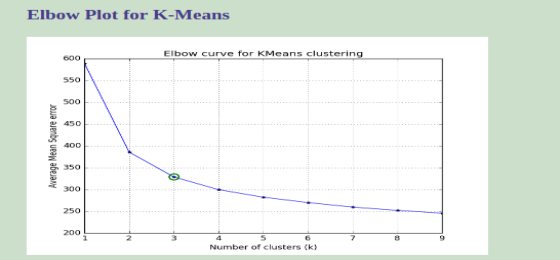

Figure 3: K-means Elbow Curve

Figure 4 shows the code snippet used to perform adaptive sampling on the dataset using the optimum number of clusters obtained from K-means. The dataset reduced from 1000 rows to 439 rows.

```
k_means = Kcluster.KMeans(n_clusters=cluster_count)
k_means.fit(data_frame)
data_frame['label'] = k_means.labels_
adaptiveSampleRows = []

for i in range(cluster_count):
    adaptiveSampleRows.append(data_frame.ix[random.sample(data_frame[data_fra


adaptiveSample = pd.concat(adaptiveSampleRows)
del adaptiveSample['label']
print len(adaptiveSample)

return adaptiveSample
```

Figure 4: Code snippet for Adaptive/Stratified Sampling

4 Task 2: Dimensionality Reduction

In this Task, I have reduced the dimensionality of my dataset using Principal Component Analysis. This was done to retain the attributes that account for the most variance in the data and represent its intrinsic dimensionality.

4.1 Principal Component Analysis

In this part, I attempted to discover the intrinsic dimensionality of the dataset. This was done by forming a covariance matrix using the dataset, plotting its eigen values corresponding to the Principal Components. This is called a Scree Plot and dimensions where eigen value is greater than one are considered for intrinsic dimensionality.
According to the Scree Plot, the intrinsic dimensionality of the dataset is two.

Figures 5 and 6 show the process.

```
stdinput = StandardScaler().fit_transform(inputSamples)
cov_mat = np.cov(stdinput.T)
eig_vals1, eig_vecs1 = np.linalg.eig(cov_mat)
cor_mat1 = np.corrcoef(stdinput.T)
eig_vals, eig_vecs = np.linalg.eig(cor_mat1)

y = eig_vals
x = np.arange(len(y)) + 1

df_eig = pd.DataFrame(eig_vals)
df_eig.columns = ["eigan_values"]
df_pca = pd.DataFrame(x)
df_pca.columns = ["PCA_components"]
sample = df_eig.join([df_pca])
```

Figure 5: Code snippet for Scree Plot

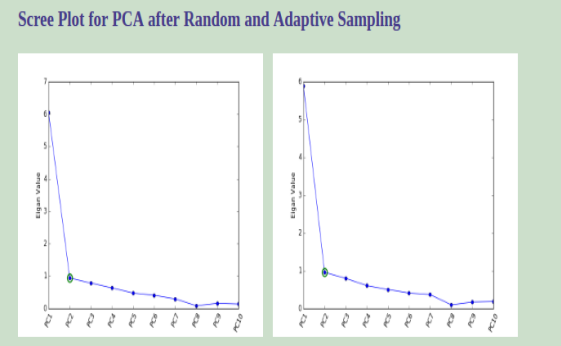

Figure 6: Scree plot for adaptive and random sampling

Dimensionality reduction is a technique which reduces the dimensions of the dataset so only the most useful dimensions which explain the bulk of the data are retained and the redundant data is removed. In this task, I have used Principal Component Analysis. However, dimensionality reduction can also be accomplished by Multi-Dimensional Scaling(MDS), Locally Linear Embedding(LLE) and Support Vector Decomposition(SVD)

4.2 Squared Loadings of Attributes

I have visualized the three highest PCA loadings - the three attributes which explain the most variance and in turn affect the world ranking of a university the most.

For random sampling, these three attributes are: National Rank, Influence and Quality of faculty
For adaptive sampling, these three attributes are: National Rank, Publications and Influence

5 Task 3: Visualization

In the third and final task, I used the D3.js library to create visualizations representing my findings, shown in Figures 9 and 10.

5.1 2D ScatterPlot of data in top two PCA vectors

I visualized the data projected into the top two PCA vectors using the 2D scatterplot. The green dots which represent adaptive sampling are a little more distributed than random, taking care of the outliers.

```
squared_loadings = []
a = np.array(loadings)
a = a.transpose()
for i in range(len(a)):
    squared_loadings.append(np.sum(np.square(a[i])))
df_attributes = pd.DataFrame(pd.DataFrame(dataframe).columns)
df_attributes.columns = ["attributes"]
df_sqL = pd.DataFrame(squared_loadings)
df_sqL.columns = ["squared_loadings"]
sample = df_attributes.join([df_sqL])
sample = sample.sort_values(["squared_loadings"], ascending=[False])
```

Figure 7: Code for Squared Loadings

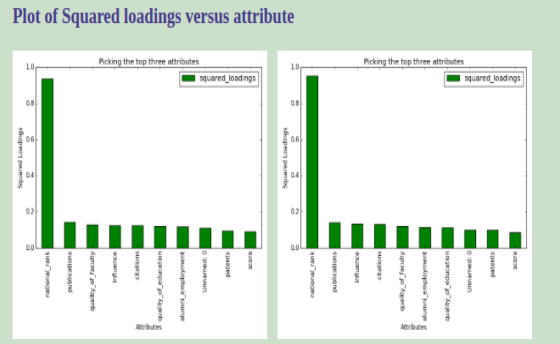

Figure 8: Visualization of Squared Loadings

```
d3.csv(filename, function(error, data) {
    data.forEach(function(d) {
        d.x = +d.x;
        d.y = +d.y;
        d.type = +d.type;
    });

    var xValueR = function(d) { return d.x;};
    var yValueR = function(d) { return d.y;};

    var labels = ['Random', 'Adaptive']
    xScale.domain([d3.min(data, xValueR), d3.max(data, xValueR)]);
    yScale.domain([d3.min(data, yValueR), d3.max(data, yValueR)]);
```

Figure 9: D3 code to parse csv files

```
svg.selectAll("circle")
    .data(data)
    .enter()
    .append("circle")
    .attr("r", 3)
    .attr("cx", function(d){
        return xScale(d.x);
    })
    .attr("cy", function(d){
        return yScale(d.y);
    })
    .style("fill", function(d) {
        return color[d.type-1];
    })
    .attr("stroke", "black")
    ;
```

Figure 10: D3 code to plot circles

```
def analyse_pca(random_sample, stratified_sample):
    pca = PCA(n_components=2)
    pca_random = pd.DataFrame(pca.fit_transform(random_sample))
    pca_stratified = pd.DataFrame(pca.fit_transform(stratified_sample))
    createFile(pca_random, pca_stratified, "pca_output.csv")
```

Figure 11: Generate PCA 2D

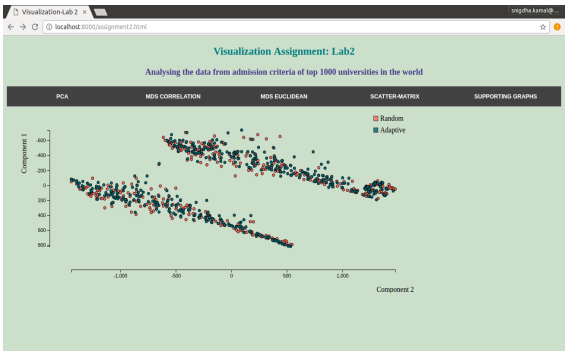

Figure 12: PCA 2D scatterplot

5.2 2D ScatterPlot of MDS - Correlation and Euclidean

Figures 13 and 14 represent the data visualized after Multi-Dimensionality Scaling of data - using both Correlation and Euclidean distances as measures of similarity.

5.3 Scatterplot-matrix of three highest PCA attributes

The three highest PCA attributes obtained for Random Sampling were - National Rank, Influence and Quality of faculty, shown in Figure 15
The three highest PCA attributes obtained for Adaptive Sampling were - National Rank of

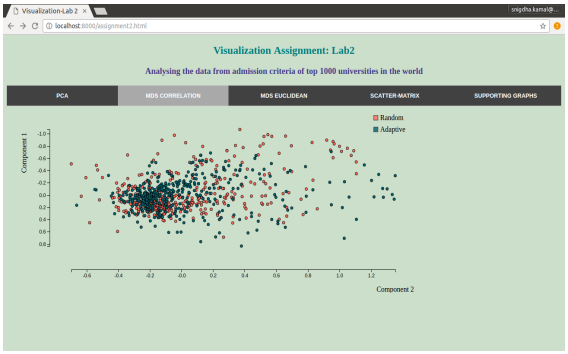

Figure 13: MDS Correlation

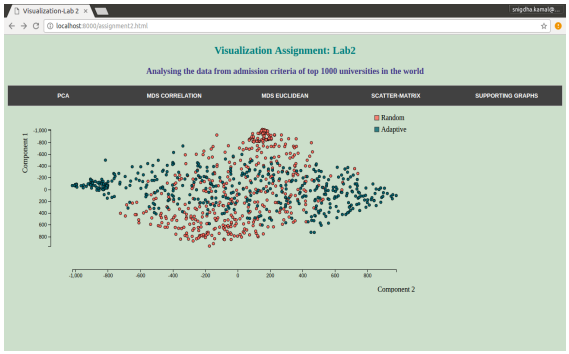

Figure 14: MDS Euclidean

the University, Publications and Influence as shown in Figure 16

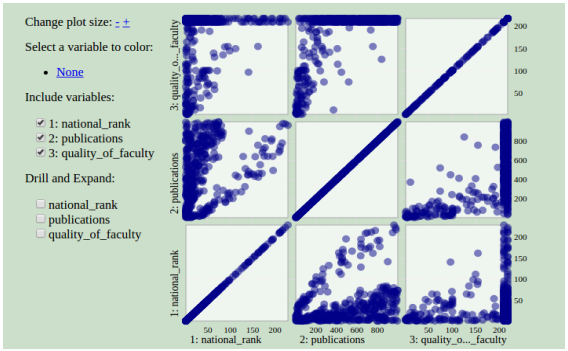

Figure 15: Random Sampling

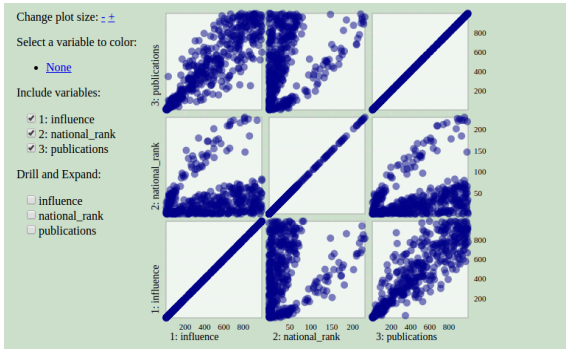

Figure 16: Adaptive Sampling