Part 2: Data Analysis and Pattern Recognition ­– Summary

In this section of the project, we sought to find patterns in graphical representations of our data. For this, we utilized unsupervised learning, Dimensionality Reduction and Clustering. Empty values in the data were filled with the means of the neighboring points and the data itself was scaled. Since this is unsupervised learning, the dataset contained no labels and was split into training and testing data (tests size is 20% of dataset).

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generatedThe first step of our analysis involved Dimensionality Reduction (DR). The five DR methods we selected were Principal Component Analysis (PCA), T-distributed Stochastic Neighbor Embedding (TSNE), Truncated Singular Value Decomposition (SVD), Isomap Embedding (ISMP), and Locally Linear Embedding (LLE). The results are shown below as scatter plots. The PCA and SVD methods are likely so similar since PCA uses SVD but centers the data before doing so. From these results, we chose to utilize TSNE and SVD data for the next step in the analysis (Clustering). We elected not to use ISMP and LLE since, visually, they do not seem likely to yield good results when evaluated with various clustering methods. SVD shows some patterns like rings on the left side of the figure and a long line that extends to the right. At the current stage however, TSNE shows somewhat well-defined clusters of data. For our dataset TSNE appears to be the best suited form of Dimensionality Reduction.

Figure 5. LLE DR

Figure 4. ISMP DR

Figure 2. TSNE DR

Figure 1. PCA DR

Figure 3. SVD DR

The next step in analysis involves Clustering. The selected methods for clustering were KMeans, DBSCAN, and Agglomerative Clustering (Hierarchical). Each algorithm has its own method for calculating the number of clusters (*n*), but since we know how many independent labels we have, we set *n* = 5 for most algorithms. Each algorithm was tested with each dataset (3 algorithms x 2 datasets = 6 results). First we’ll compare the clustering on the SVD dataset. From these results, DBSCAN appears the worst since there are some random clusters (red and blue) and not the clearest separation between them. Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedKmeans and AC are very similar, yet the border for the purple cluster in AC does not seem well defined. In the KMeans plot, the border appear to be the best defined. It is interesting to note that both KMeans and AC distinguised the long line into separate clusters on the same point (see red star on Fig. 6). Although only a small dip in the line, if these clusters are meant to define different operating states of Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generatedthe plant, even a small deviation is important to identify. We believe KMeans performed the best for the SVD dataset. Figures 9-11 show the clustering results on the TSNE data. Again, the elbow method was used to select epsilon for DBSCAN. The initial results were not good and so the value was slightly varies until the results shown in Fig. 10 were produced. DBSCAN seems to perform the worst with the TSNE data. Although there are good distinctions for some clusters, there are also small pockets of other clusters (cyan, pink, purple) that do not look visually distinct enough to be there own cluster. On the other hand, KMeans and AC seem to perform better. One thing to note is that (in the bottom-left corner of the plot) both algorithms seemed to separate those clumped data points at different areas. Visually, one could imagine separating the whole bottom-left clump into its own cluster and perhaps a different clustering method would. In the AC plot, the blue-green clump (I) seems to overextend to the left into the yellow, but, for this same area, Kmeans has a better distinction between those two clusters. In the end KMeans seems to perform the best out of the three algorithms.

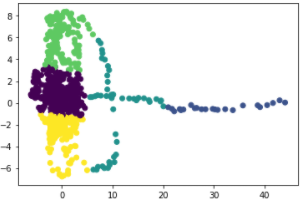


Figure 8. AC (SVD)

Figure 7. DBSCAN (SVD)

Figure 6. KMeans (SVD)

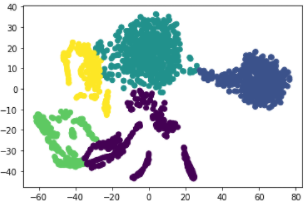


Figure 10. DBSCAN (TSNE)

Figure 11. AC (TSNE)

Figure 9. KMeans (TSNE)