APPROVAL SHEET

Title of thesis: UNDERSTANDING TIME SERIES DATA CLUSTERING AND CORRELATION THROUGH VISUALIZATION

Name of Candidate: Shantanu Sengupta
Masters of Science, Computer Science, 2018

Thesis and Abstract Approved: _____
Dr. Penny Rheingans
Professor
Computer Science and Electrical Engineering Department

Date Approved: _____

ABSTRACT

Title of thesis: UNDERSTANDING TIME SERIES DATA CLUSTERING AND CORRELATION THROUGH VISUALIZATION

Shantanu Sengupta, Master of Science, 2018

Dissertation directed by: Dr. Penny Rheingans
Department of Computer Science and Electrical Engineering

Time series is an essential and ubiquitous source of data with its applications in stock markets, digital signal processing, weather forecasting, census analysis, and health monitoring data. This form of data is generated continuously and in massive amounts. To make sense of such deluges of overlapping temporal data, we employ clustering algorithms which reduce the clutter by aggregating similar shaped/behaving data into their clustered versions. In order to view if the clustering is effective and the clusters produced are tight, we need a visualization technique for displaying time series clusters and the underlying data it represents.

There has been previous work done in creating visualizations to represent time series data such as line charts, Gantt charts, stream charts, and heat maps. These visualization techniques are useful in representing a single or multiple clustered data point on a temporal scale, but none of them can represent the distribution of values within each of the clusters. As a result, this shortcoming calls for new visualization techniques that combine temporal representation

techniques with statistical representation techniques.

This visualization aims to help users visualize overall clusters in the time series data and identify interesting trends and patterns in them. In addition to viewing the temporal characteristics of such clusters, the visualization should also represent information about the distribution of data within a cluster. The proposed visualization achieves this by representing various statistical aspects in the form of box plots superimposed upon line charts. The proposed visualization method helps users understand if there exists a correlation between two different time series data occurring in the same time domain. This feature can help users explore the causality and periodicity relationships between the two different time series data.

In this research, we demonstrate the results of using this visualization method for finding the clustering and correlation within temperature and pressure time series data for 18 cities. We also discuss an application of this visualization in understanding the effectiveness of help-seeking behavior on student grades. This application would allow users to correlate office hours attendance with a student's performance in a course.

This research contributes towards a better understanding of the properties and quality of different time series clustering algorithms through a visual representation of a cluster distribution. It also introduces a novel approach in visualizing the correlation between two simultaneously occurring time series.

UNDERSTANDING OF TIME SERIES DATA CLUSTERING AND CORRELATION THROUGH VISUALIZATION

by

Shantanu Sengupta

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
2018

Advisory Committee:
Dr. Penny Rheingans, Advisor
Dr. Marc Olano
Dr. Katherine Gibson
Dr. Charles Nicholas

Table of Contents

List of Figures

Chapter 1

Introduction

Time series data is a form of data where the values are indexed in natural temporal order. The analysis of time series data involves extracting meaningful patterns, trends, statistics and characteristics from the data. Its applications include, but are not limited to, helping researchers understand gene data [Ernst et al., 2005], forecast network performance metrics [Gutierrez and Wiesinger-Widi, 2016], predict global solar irradiance [Martín et al., 2010], economic forecasting [Hamilton, 1989], temperature forecasting [Doganis et al., 2006], effect of air pollution on health [Dominici et al., 2002], stock market analysis [LeBaron et al., 1999], workload projections [Chalder et al., 2003], understanding diseases and effect of vaccinations [Anderson et al., 1984], etc.

Clustering is one of the most useful data mining tools for extracting meaningful information from time series data. This form of unsupervised learning tool helps users work with unlabeled data and understand the underlying structure of complex and massive datasets. It is difficult to use supervised learning algorithms for time series data because of their large size and complexity [Aghabozorgi et al., 2015]. This shortcoming is the reason unsupervised clustering mechanisms work better for time series data. Clustering algorithms are heavily employed in applications such

as fMRI data analysis [Goutte et al., 1999], gene ontology [Ernst et al., 2005], and rule prediction analysis [Das et al., 1998].

1.1 Problem Domain

Data visualization plays a significant role in helping users view the effectiveness and reliability of clustering algorithms on time series data. It helps validate the clusters generated after applying data processing algorithms such as Independent Component Analysis (ICA) visually to identify if similar data points have been grouped successfully into the same cluster [Himberg et al., 2004]. It also plays the role of an interactive exploratory tool for discovering interesting patterns and trends in temporal data. In some cases, it can help users understand trends of seasonality and periodicity sometimes seen in time series data [van Wijk and van Selow, 1999]. Some visualizations, such as density based and distance-density based visualizations, are particularly helpful for viewing fuzzy clustering mechanism on time series data [Ultsch and Mörchen, 2005].

Visualization plays a major role in understanding the clusters generated by a clustering algorithm. We can start by understanding the role visualization plays in viewing clusters for static data. In Figure 1.1, we can see unclustered two dimensional static data before it undergoes clustering algorithm. In Figure 1.2, we can see static data points clustered into three groups using the k-means algorithm. Here visualization helps users visualize the original data points and the clusters generated. Visualization helps users understand the result of the clustering

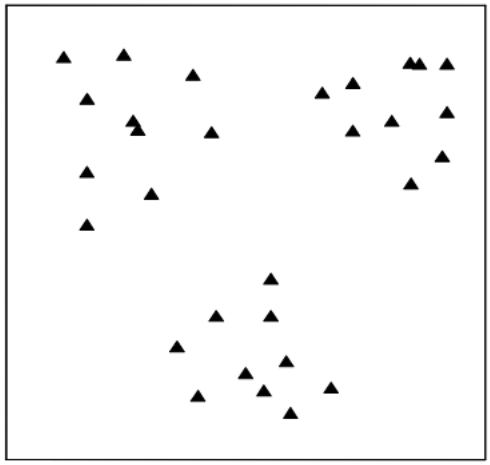


Figure 1.1: Two-dimensional input data with three clusters [Jain, 2010]

algorithm. We would like to introduce a visualization that helps perform the same for time series data. The current work done in time series cluster visualizations misses out on two areas of time series analysis. First, it focuses heavily on representing the clusters themselves with the assumption that the clustering algorithm was successful in aggregating similar data points within the same cluster. However, in some cases, clustering algorithms can be inefficient and ineffective too. One of the significant shortcomings of the k-means clustering algorithm is its inability to determine a good value for k. Also, the results of k-means clustering algorithm vary with different initial seeds and are highly sensitive to scale. Similarly, hierarchical clustering suffers from high time complexity for large datasets and is highly sensitive to outliers [Chen et al., 2005]. Second, most of the time series cluster visualizations are focused on only understanding the single time series that undergoes clustering. They are unable to explain relationships of causality and correlations with other time series cluster

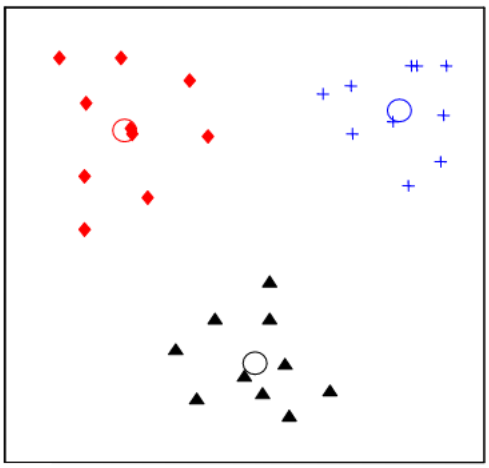


Figure 1.2: Final clustering obtained by K-means algorithm at convergence [Jain, 2010]

data, i.e., they are unable to perform multiple time series analysis. Multiple time series analysis works with more than one-time series and aims at finding dynamic interrelationships among the variables of each of the time series. These play a significant role in creating predictive and forecast models.

Creating a visualization that allows users to see these models in action can help them understand the underlying data and models used for determining the relationships between multiple time series data. Also, users can use this visualization as an exploratory tool to find more patterns, trends, and outliers in the time series data.

1.2 Research Goals

My research goals are to create a visualization that solves and satisfies the following objectives:

- Users can use the visualization to understand the distribution within time

series clusters.

- Users can use the visualization to understand the effectiveness of a clustering algorithm on time series data.
- Users can use the visualization to understand how two different time series data correlate with each other.

1.3 Visualization Goals

To address the two shortcomings of existing visualization techniques, we propose a new visualization technique which can help users assess the validity of time series clustering algorithms. The new visualization method introduces a novel approach for viewing correlation among multiple time series. Furthermore, this visualization can allow users to view additional statistical details about the time series clusters, which were abstracted while clustering. Using this technique, we can understand if the clusters produced were tight and can adequately represent the underlying data points.

1.4 Thesis Statement

Visualizing multiple time series in their clustered format using multiple coordinated views can help users understand important relationships among these time series and assist with forecast and prediction, which could not have been done using traditional time series visualization and analysis techniques. The time

series clusters represented visually using this visualization technique can be supplemented with the representation of cluster distribution and additional statistical characteristics.

1.5 Approach

Clustering algorithms, when applied to time series data, can transform the data into meaningful yet abstracted versions of the original data. Effectively representing these forms of data by an amalgamation of traditional visualization approaches can provide deeper insights into the data without requiring the user to be familiar with newer visualization techniques. It builds on the knowledge and familiarity of line charts, box plots, and stacked bar charts into time series analysis. This form of clustering and visualization is best while working with massive overlapping time series datasets since not a lot of previous work has been done to view time series clusters visually.

The visualization offers novel capabilities to analyze time series data. The users can have five views for viewing the time series data. They are as follows :

1. Unclustered original data view
2. Unclustered original data view but color-coded according to clusters
3. Clustered data view
4. Clustered data view with the distribution
5. Clustered data view with correlation to other time series

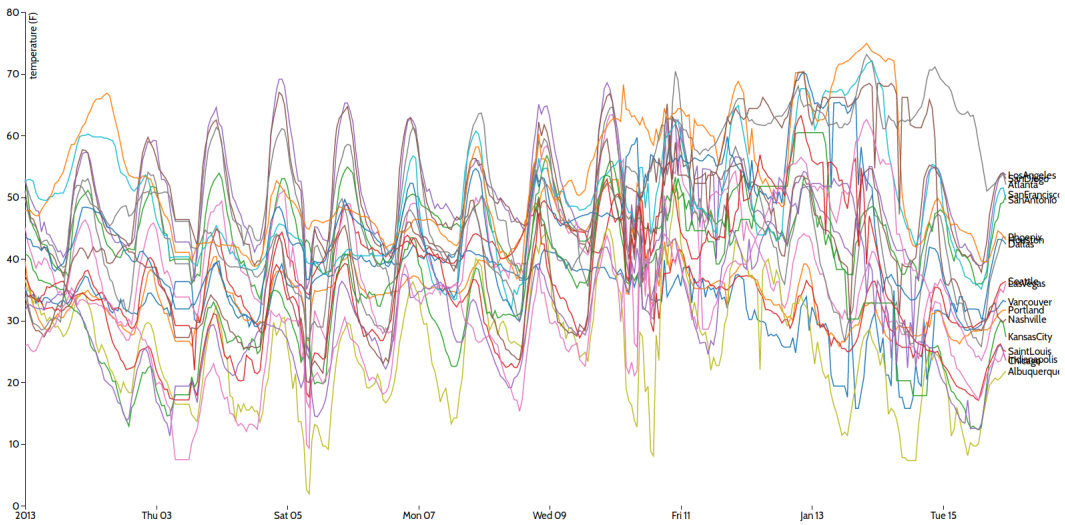


Figure 1.3: Visualization of Original Time Series Data in Unclustered Format

We will be looking at the use of this visualization technique to understand the temperature data for 18 cities for the first 15 days of January 2013 [Mario et al., 2017]. Suppose, we need to summarize the entire temperature data into four major trends and understand their trajectory through the time domain. This data, when represented in an unclustered format as seen in Figure 1.3, can appear cluttered and confusing to comprehend. It is difficult to interpret each trajectory because of their closeness in paths and overlapping nature. However, it is apparent that many of the data paths have similar or parallel trajectories. Here, clustering algorithms can effectively reduce the number of visual elements from the unclustered view and declutter the visualization.

Clustering these data points based on their similarity in trajectory gives the users an idea of how the overall data can be summarized. We use a k-means clustering method with k=4 to cluster this data. The users can use the second view which transforms the unclustered data and color codes it by its cluster as

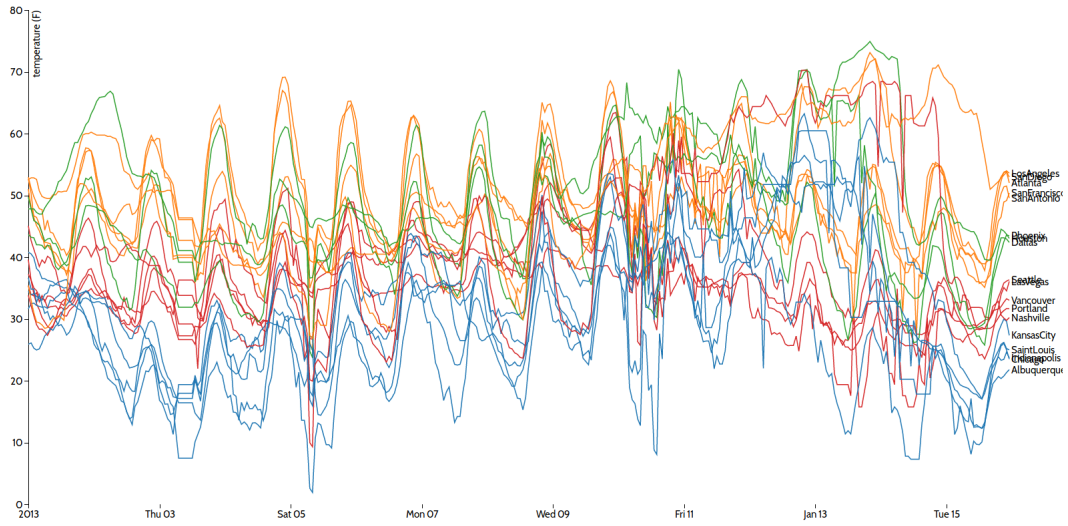


Figure 1.4: Visualization of Original Time Series Data as it undergoes clustering when number of clusters = 4

seen in Figure 1.4 to understand if the clustering algorithm clustered together similar-looking data points. This visualization method introduces the human element in verifying the effectiveness of the clustering technique. As can be seen in Figure 1.4, the k-means clustering algorithm clusters the 18 cities into four groups of colors - red, orange, green and blue. All the time series data points in the same cluster have the same color. We can observe slight similarity in paths for time series data points that have the same color. In Figure 1.3, we can focus our attention on two groups of cities. The first group of cities - Los Angeles, San Diego, San Antonio, San Fransisco and Atlanta, and the second group of cities - Seattle, Las Vegas, Vancouver, Portland and Nashville have similar trajectories. The users can visualize whether similar looking data points were clustered in the same group since the same color denotes data points within the same cluster.

However, it is still difficult for a user to comprehend the path of individual

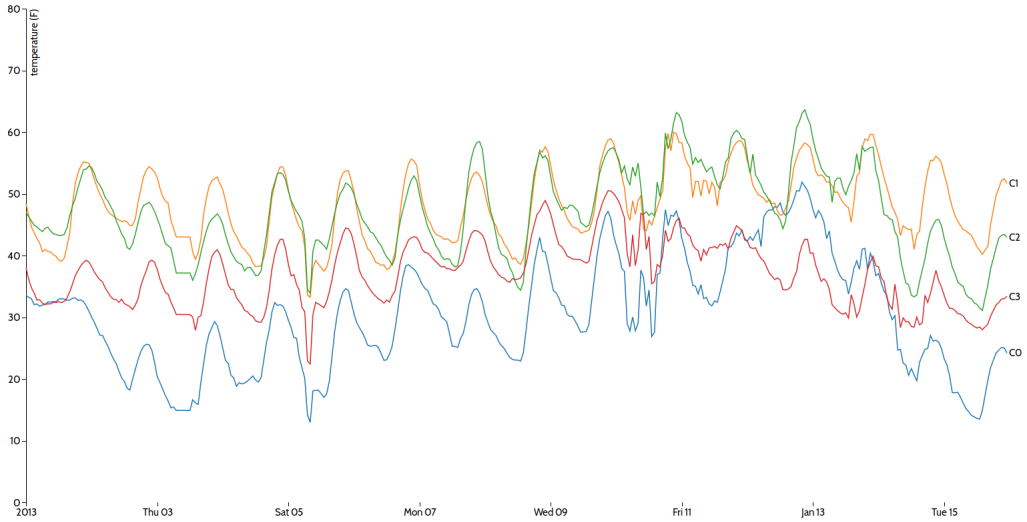


Figure 1.5: Visualization of Clustered Time Series when Number of Clusters = 4

data points because of the clutter. This figure still represents the entire time series dataset consisting of 18 cities' temperature. As a result, visualizing only the clusters for these similar looking data together reduces the overall clutter in the visualization by a significant amount as can be seen in Figure 1.5. The user can cluster the entire temperature data into a fewer number of data points than the original 18 data points. To check the effectiveness of the clustering algorithm visually, we will perform clustering with k=4 and color code each data point in the same cluster with the same color as can be seen in Figure 1.5. Now we can see that the orange and red clusters represent the two groups of cities that we had discussed earlier. The users need to decide the number of clusters depending on the level of detail required. Their choice for the number of clusters also depends on whether they can view the time series clusters distinctly and with sufficient detail to be able to answer relevant questions.

Though clustering reduces the overall visual elements in the visualization, a

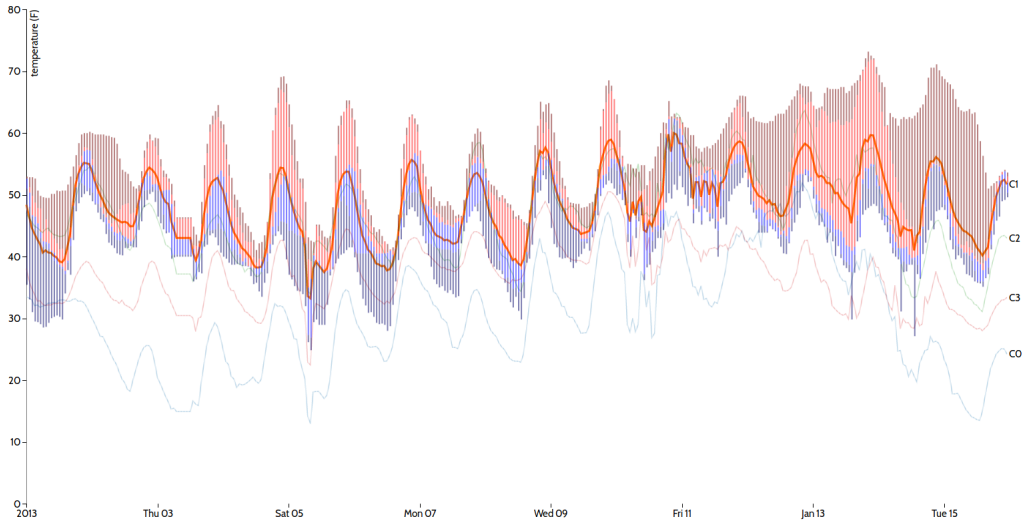


Figure 1.6: Visualization of Distribution within Clustered Time Series when Number of Clusters = 4

user still cannot distinguish whether the clustering was efficient and tight. To be able to tell whether the clustering is tight, the users need to be able to view the distribution and spread of the data within a cluster. Here, the users can use the fourth view to view the distribution of data points within a cluster. We achieve this by representing the distribution of each cluster at that point of time with a stacked bar chart centered around the median value of the cluster. This representation of the clustered data points using a traditional approach may seem like an oversimplification of the entire data because it summarizes all the underlying unique data points. This representation abstracts some minute and distinct characteristics of the data that may be important for the user to understand the effectiveness of the clustering algorithm. Having a form of stacked bar charts amalgamated with traditional line charts allows users to view such details without removing the focus from the actual time series trajectory. This

view also allows users to see the spread of the cluster and determine if the clustering algorithm was tight. We can see this in Figure 1.6 where we use this visualization to understand temperature data for 18 cities. The users can see the median values of the time series data by following the trajectory of the boundary between the light blue and light red rectangles. Users can compare this median line with the cluster trajectory line which represents the geometric mean values of the times series data. For a distribution where the value of median is equal to the geometric mean indicates that the data is distributed uniformly and symmetrically within the distribution. The users can see the selected cluster by its highlight with a thick opaque line while the other clusters are transparent in the background. We will look at each aspect of the stacked bar chart representing the distribution visual element in the later sections.

The final view is useful in understanding how this data correlates with other time series data. This view allows users to view multiple time series occurring at the same time in such a way they can see their interdependency and causality on the same panel. This capability allows the user to perform multiple time series analysis using this visualization technique. We will discuss this capability in detail in the later sections.

1.6 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 describes the previous approach towards visualizing time series data with an emphasis on data

preprocessing, clustering techniques and the overall visualization of the data. Chapter 3 is devoted to explaining the approach to creating the overall visualization. It addresses individually all the steps involved in detail such as clustering time series based on shape, visualizing the cluster distribution, and visualizing the correlation between multiple time series. Chapter 4 examines an application of the visualization tool to understand clustering and correlation between temperature and pressure data for 18 cities. Chapter 5 discusses an application of the visualization tool to understand the effect of help-seeking behavior on student grades. Chapter 6 explains the future work in this research area and possible improvements. Lastly, Chapter 7 concludes the entire paper and the research.

Chapter 2

Related Work

Research done in understanding time series clustering addresses the effectiveness of each algorithm, distance metrics, and data processing methods for different types of time series data. This research also explains the shortcomings and limitations of each clustering algorithm for grouping similar time series data. Information visualization has made it tremendously easy for users for understanding time series clusters generated by clustering algorithms. The existing visualization techniques that exist for viewing time series data, however, cannot be extended for viewing clusters and helping users understand the correlation of one time series over another. Moreover, the problem with the existing techniques in viewing time series clusters and correlation is that they are highly application specific, and highly unintuitive. In this section, we will see the existing time series clustering algorithms and review the visualization methods to understand time series clusters.

2.1 Clustering Algorithm

The approach to visualizing time series clusters starts by first clustering the time series data using clustering algorithms. Clustering is the most common form of unsupervised form of learning. It helps find structure in unlabeled data. It helps

characterize massive datasets by a small number of representative data points that summarize the entire dataset in a more compact and meaningful way. Some of the most popular clustering algorithms are the k-means and hierarchical clustering algorithms [Liao, 2005]. Each of these clustering algorithms has different distance metrics to determine the criteria for similarity and dissimilarity among data points. In the following section, we will discuss these two clustering algorithms: k-means and hierarchical clustering in detail and the distance metrics generally used for time series clustering.

2.1.1 K-Means Clustering Algorithm

K-means is one of the most popular cluster analysis algorithms. This algorithm reduces n observational data points into k clusters such that these k clusters effectively represent the underlying n data points [Hartigan and Wong, 1979].

Suppose we need to cluster the data points $x_1, x_2, ..., x_n$ and K is the number of clusters centroids with L_i indicating the label for each of the n data points [Hartigan and Wong, 1979]. The k-means algorithm, in a nutshell, can be explained using the below two expressions [Hartigan and Wong, 1979]:

1. Randomly choose K initial centroids C_1, C_2, ... , C_K

2. Repeat until convergence, i.e., the updated centroids are the same as the old centroids.

   For every data point i, set

L_i = Minimum for all j (Distance between data point x_i and centroid C_j)

$L_i = \text{Minimum}_{j \epsilon K}(\text{distance}(x_i, C_j))$

For every centroid j, set

C_j = Geometric mean of data points within the cluster

The advantage of the k-means algorithm is that the clusters produced by this algorithm are tighter compared to other clustering mechanisms such as hierarchical clustering. It is also computationally faster for smaller values of k. The disadvantage of k-means is that we cannot decide a good value for k in advance. Also, random initialization of initial centroids results in different outcomes every time the k-means clustering algorithm is run [Abbas,]. An effective method of choosing a suitable value of k is using the elbow method and silhouette method, which we will discuss in the later sections [Kodinariya and Makwana, 2013].

2.1.2 Hierarchical Clustering

Hierarchical clustering is a form of cluster analysis algorithm that works by building a hierarchy, or tree-like structure, of clusters [Jain et al., 1999]. There are generally two types of hierarchical clustering - agglomerative and divisive. They differ in their clustering approach. Agglomerative starts by keeping the original data points as individual clusters and combining them into larger clusters iteratively until the result is just one cluster. By contrast, divisive hierarchical clustering starts with just one cluster and splits it into more clusters until the result is all the original data points [Jain et al., 1999].

Consider n data points that are present initially to be clustered using hierarchical clustering. We create a distance matrix between all these n points to get an $n \times n$ matrix. The agglomerative hierarchical clustering algorithm can be explained in a nutshell by the following pseudo-code:

1. Assign each item to its cluster such that if there are n items, then you have n clusters. Let the distances between the clusters equal the distances between the items they contain.

2. Assign each item to its cluster such that if there are n items initially, then consider n clusters each containing their corresponding n data points. Let the distances between the clusters equal the distances between the items they contain.

3. Find the closest pair of clusters and merge them into a single cluster.

4. Compute new distances between the new cluster created and each of the old clusters.

5. Repeat steps 3 and 4 until we cluster all the items into a single cluster containing all the n items.

The advantages of hierarchical clustering are that it does not require the number of clusters in advance to determine effective clusters and is easy to implement. The disadvantages of hierarchical clustering are that it is sensitive to noise and outliers, and it is difficult to identify the correct number of clusters from the dendrogram.

2.2 Distance Metrics

This metric helps determine the similarity or dissimilarity between two data points. In other words, this behaves like the objective function of the clustering algorithm. In this section, we discuss five such distance metrics: Euclidean, Dynamic Time Warping, Manhattan, Minkowski, and Chebyshev distance.

2.2.1 Euclidean Distance

The Euclidean distance between two points in Euclidean space is the straight-line distance between them [Singh et al., 2013]. Suppose there are two distinct time series data x and y with n discrete time values for the same time space. We can define the Euclidean distance between these time series as

$$d_{euclidean}(x, y) = \sqrt[2]{\sum_{i=1}^{n}(x_i - y_i)^2}$$

where x_i and y_i is the value for time series at the i^{th} time for x and y respectively.

An advantage of Euclidean distance is that it is not affected by the addition of new data points to the cluster analysis which may also be outliers.

2.2.2 Dynamic Time Warping Distance

The dynamic time warping (DTW) distance is a distance metric generally used for comparing temporal forms of data. The DTW distance metric is generally used in applications that involve comparing audio and visual signals. The basic intuition

behind this metric is to compare two signals by distorting them non-linearly in the temporal dimension to see how similar they are to each other. The advantage of dynamic time warping is that it can be used to compare two unequal time series. Consider two distinct time series x and y consisting of m and n discrete time values respectively. The DTW distance algorithm can be summarized as follows:

```
function DTWDistance(x,y)
    DTW = array [0...n, 0...m]
    DTW[0, 0] = 0
    for i = 1 to n
        DTW[i, 0] = ∞
    for i = 1 to m
        DTW[0, i] = ∞
    for i = 1 to n
        for j = 1 to m
            DTW[i, j] = d(x[i], y[j]) + minimum(DTW[i-1, j], DTW[i, j-1], DTW[i-1, j-1])
    return DTW[n, m]
end function
```

DTW distance metric is considered more robust than Euclidean distance because of its ability to compare two time series of different lengths.

2.2.3 Manhattan Distance

The Manhattan distance between two points is the absolute distance between them in the cartesian coordinates [Singh et al., 2013]. Suppose there are two distinct time series x and y with n discrete time values for the same time space. We can define the Manhattan distance between these time series as

$$d_{manhattan}(x, y) = d_{manhattan}(y, x) = ||x - y||$$

$$d_{manhattan}(x, y) = |x_1 - y_1| + |x_2 - y_2| + ... + |x_n - y_n|$$

$$d_{manhattan}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

where x_i and y_i are the values for the time series at the i^{th} time for x and y respectively.

Manhattan distance is based on absolute difference value and is preferable to Euclidean distance for high dimensional datasets because of its ability to provide better relative contrast than Euclidean distance for different values in a dataset [Aggarwal et al., 2001]. The reason being the multivariate values for data points are aggregated into a single variable. So if two points are close on most variables but more discrepant on one of them, the Euclidean distance will exaggerate that difference, but Manhattan distance will be unaffected by it and affected more by the closeness of the other variables.

2.2.4 Minkowski Distance

Suppose there are two distinct time series x and y with n discrete time values for the same time space. We can define the Minkowski distance between these time series as

$$d_{minkowski}(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}$$

where x_i and y_i is the value for time series at the i^{th} time for x and y respectively [Van de Geer, 1995]. Here, p is called the Minkowski exponent.

The Minkowski distance for $p = 1$ corresponds to the same as Manhattan distance, and $p = 2$ corresponds to the same as Euclidean distance. The choice

for the value of p remains an open issue though some research uses the Minkowski clustering index or silhouette width indexes for determining a suitable value of the Minkowski exponent [de Amorim and Mirkin, 2014].

The only disadvantage to Minkowski approach is that it is difficult to perceive the Minkowski distance between two data points visually in our mind because unlike Euclidean and Manhattan, it cannot be perceived spatially [Lu et al., 2016].

2.2.5 Chebyshev Distance

The Chebyshev or Tchebychev distance between two points is the maximum absolute distance between them in any Cartesian coordinate [Singh et al., 2013]. Suppose there are two distinct time series data x and y with n discrete time values for the same time space. We can define the Chebyshev distance between these time series as

$$d_{chebyshev}(x, y) = max_{i\epsilon n}(|x_i - y_i|)$$

where x_i and y_i is the value for the time series at the i^{th} time for x and y respectively.

The primary advantage of the Chebyshev distance is that it is efficient and takes less time to compute the distances between the data points. However, Chebyshev distance considers only one single feature for representing a dataset, which might not offer enough description of the dataset to lead to useful cluster analysis.

2.3 Linkage Criteria

Before a clustering algorithm is performed, we need to determine the method used to measure the distance between different clusters. This method is called the linkage criteria between clusters. The linkage criteria is a function that determines the pairwise distances between observations. The most commonly used linkage criteria are

- Single Linkage or Minimum Linkage : In this form of linkage criteria, we determine the similarity of two clusters on the similarity of their most similar members i.e. the elements in those two clusters which have the minimum distance among each other [Wilks, 2011]. Mathematically, the linkage criteria function can be described by the expression

$$D(X, Y) = min_{x \epsilon X, y \epsilon Y}(d(x, y))$$

  where X and Y are any two sets of elements considered as clusters, and d(x,y) denotes the distance between the two elements x and y.

- Complete Linkage or Maximum Linkage : In this form of linkage criteria, we determine the similarity of two clusters on the similarity of their most dissimilar members i.e. the elements in those two clusters which have the maximum distance among each other [Wilks, 2011]. Mathematically, the linkage criteria function can be described by the expression

$$D(X,Y) = max_{x\epsilon X, y\epsilon Y}(d(x,y))$$

where X and Y are any two sets of elements considered as clusters, and d(x,y) denotes the distance between the two elements x and y.

- Average Linkage or Mean Linkage : In this form of linkage criteria, we determine the similarity of two clusters on the average distance between all their members [Wilks, 2011]. Mathematically, the linkage criteria function can be described by the expression

$$D(X,Y) = \frac{1}{|X|.|Y|}\sum_{x\epsilon X}\sum_{y\epsilon Y} d(x,y)$$

where X and Y are any two sets of elements considered as clusters, and d(x,y) denotes the distance between the two elements x and y.

2.4 Choosing an effective cluster

One of the major limitations of the k-means algorithm is that the choice for k, i.e., the number of clusters, should be known in advance. In this research, we consider two major methods that help find the number of clusters for time series data.

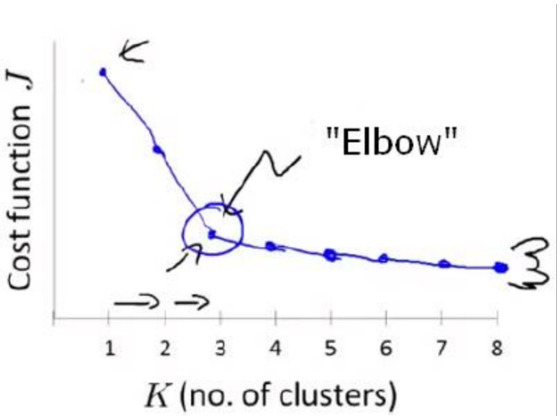


Figure 2.1: Depiction of a good elbow graph with a sharp elbow joint [Kodinariya and Makwana, 2013]

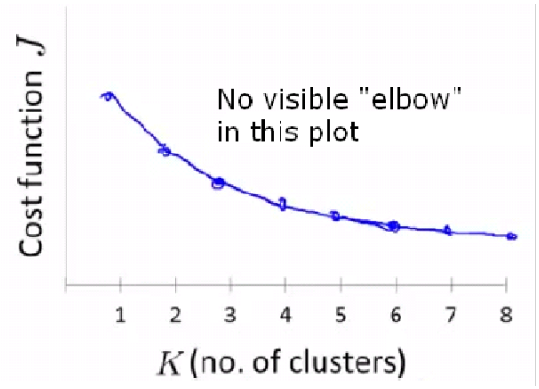


Figure 2.2: Depiction of a bad elbow graph with no distinguishable elbow joint [Kodinariya and Makwana, 2013]

2.4.1 Elbow Method

The first method is called the Elbow Method which is a visual technique for determining an optimum number of clusters for the k-means algorithm [Kodinariya and Makwana, 2013]. It works by assessing the sum of squared errors between each of the points in a cluster with the cluster centroid.

Mathematically, we can define the sum of squared errors as

$$SSE = \sum_{i=1}^{K} \sum_{x \epsilon c_i} distance(x, c_i)^2$$

where SSE = Sum of Squared Errors

K = Number of Clusters

$x \epsilon C_i$ = Data points within Cluster i

C_i = Centroid of Cluster i

distance(a,b) is any suitable distance metric between data point a and b

As expected, the sum of squared errors decreases as the number of clusters increases because as soon as the number of centroids is equal to the number of original data points and they converge to the original data points resulting in a sum of square errors of zero. Therefore, we need to find an intermediate cluster value which strikes a balance between the number of clusters and least sum of the square of errors. The idea behind the elbow method is to choose a value of k at which SSE decreases abruptly, which produces an elbow effect in the graph as seen in Figure 2.1. However, the disadvantage of elbow graphs is that the results may not always contain an elbow like path. Sometimes there is no visible elbow or multiple elbow points as can be seen in Figure 2.2.

2.4.2 Silhouette Method

The second method that helps find a suitable number of clusters is called the silhouette method [Rousseeuw, 1987]. Unlike the elbow method which focusses on the sum of the squared distance between points and their centroids, the silhouette method focusses on the distance of points from points in the same cluster and

distance from the points in the other clusters.

The advantage of using silhouettes is that they depend only on the actual partition of the objects and not on the clustering algorithm. As a result, this method can be used to improve upon the results obtained using the clustering algorithm. This method can also be used to compare the results of different clustering algorithms when applied on the same data [Rousseeuw, 1987].

To begin creating the silhouette score $s(i)$ for a data point I, we calculate the average dissimilarity scores $a(i)$ with other points within the same cluster and $b(i)$ with other points in different clusters [Rousseeuw, 1987].

We can mathematically express average dissimilarity as

(i) = Average distance of data point i with all data points in the same cluster

b(i) = Minimum of all the average distances of data point i with data points in other clusters

The value $a(i)$ for a data point i indicates its closeness with other data points in the same cluster. The smaller the value of $a(i)$, the better is the placement of that point in that cluster. The value $b(i)$ for a data point i indicates its closeness with the data points in the next closest cluster. This neighboring cluster indicates that this is the next best cluster for this data point [Rousseeuw, 1987]. Using these two dissimilarity measures, we can define the silhouette value for a data point.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

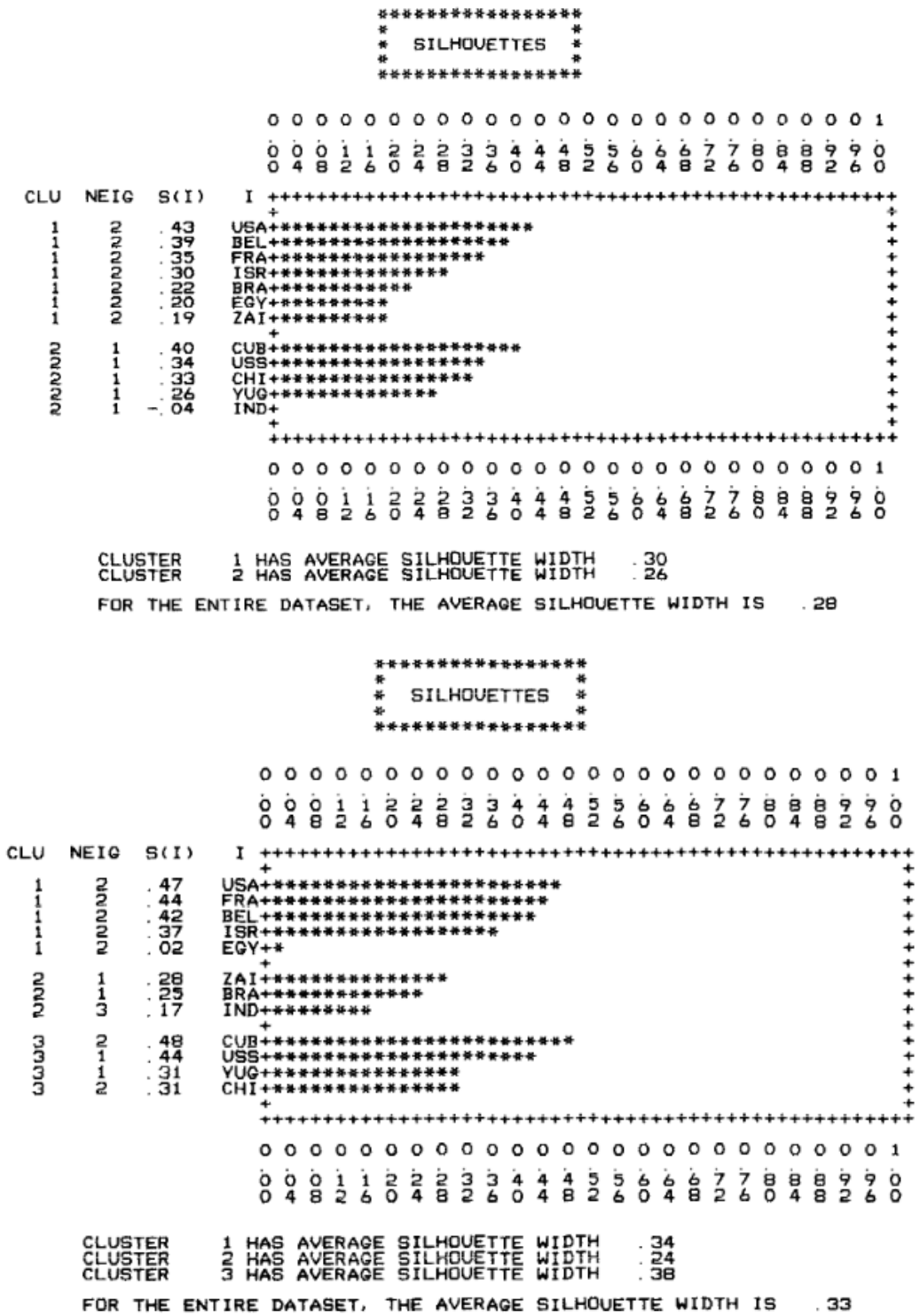
which is stated as

$$\alpha(x) = \begin{cases} 1 - \frac{a(i)}{b(i)} & a(i) < b(i) \\ 0 & a(i) = b(i) \\ \frac{a(i)}{b(i)} - 1 & a(i) > b(i) \end{cases}$$

As a result, the range for $s(i)$ lies between [-1,1]. If $s(i)$ is close to 1, then $a(i) << b(i)$, which indicates that the data point is very well matched with its cluster. If $s(i)$ is close to -1, then it indicates that the data point is better matched with its neighboring cluster. A value of $s(i)$ equal to 0, indicates that the data point is equally matched to both the cluster and its neighboring cluster.

The next step after obtaining the silhouette values is to plot the graph to understand the effectiveness of the clustering approach. We plot all the silhouette values for data points in a cluster in increasing order as can be seen in Figure 2.3. The user can use this representation to understand how the data points are placed within a cluster. A wider silhouette indicates large $s(i)$ values indicating a good cluster [Rousseeuw, 1987]. It can be seen in Figure 2.3 that the silhouette for k=2 is narrower than k=3, thereby indicating that k=3 is a better value for k. Another metric to determine a good value of k is to calculate the average silhouette width for different values of k and choose the value of k which is the maximum [Rousseeuw, 1987]. The average silhouette width is simply the average of the s(i) for all objects i in the whole data set. As can be seen in Figure 2.3, the average silhouette width for k=2 is 0.28 and for k=3 is 0.33, thereby indicating that k=3 is a better choice for clusters than k=2.

Figure 2.3: Silhouettes for k=2 and k=3 clusters respectively after clustering 12 countries data using k-median algorithm [Rousseeuw, 1987]



2.5 Understanding Time Series Data and Clustering Mechanisms

Time series data is a set of data points ordered chronologically. This form of data is generally known for its large size, high dimensionality, and dynamic nature. For our research, we will be working with time series data which is static. This property allows us to use data processing algorithms generally used for static data such as clustering algorithms.

Clustering algorithms play an essential role in data mining and in understanding the structure of unlabeled data. They partition the dataset into groups such that data points within a cluster are more similar to each other than data points in any other cluster. The appeal of using clustering algorithms for time series data lies in their ability to help users find patterns and anomalies in massive unlabeled datasets.

Clustering time-series algorithms can be classified into three major categories on the basis of the size of the time series undergoing clustering - whole time series clustering, subsequence clustering, and time point clustering [Aghabozorgi et al., 2015].

Whole time series clustering is the same as conventional clustering on time series data in its entirety whereas subsequence clustering is clustering applied to the time series extracted from a sliding window. Subsequence clustering, also known as STS clustering, is commonly used as a subroutine in various other algorithms for classification and prediction. Despite its usefulness in other algorithms, it has been found that clustering on the basis of subsequences is

meaningless because the clusters generated using this algorithm are inherently random [Keogh and Lin, 2005]. In time point clustering, the clustering of data points is done on the basis of a combination of their temporal proximity and the similarity of their corresponding values [Zolhavarieh et al., 2014]. In this category of time series clustering, not all points are clustered, some of them are considered as noise and ignored [Aghabozorgi et al., 2015].

There has been significant research done in reviewing the effectiveness of different time series clustering algorithms and distance metrics. Liao has reviewed the following clustering algorithms used for time series data: relocation clustering, agglomerative hierarchical clustering, k-means and fuzzy c-means, and self-organizing maps [Liao, 2005]. He has also examined the similarity/dissimilarity metrics of Euclidean distance, root mean square distance, Minkowski distance, Pearson's correlation coefficient, short time series distance, dynamic time warping distance, and KullbackLiebler distance. Liao also evaluates the results of the clustering mechanisms depending on whether the ground truth is known in advance. Liao concludes that the effectiveness of the clustering algorithm depends on whether it works directly on the raw data, on features extracted from the raw data, or on models built using the raw data [Liao, 2005].

Saas et al. propose that there does not exist a consensus on which clustering algorithm works best for time series data since it depends on the goals of clustering and the application domain [Saas et al., 2016].

The original k-means algorithm suffers from zero accuracy guarantees due to the randomness in the initial seeds. As a solution, Arthur and Vassilvitskii

suggest an improvement to the existing k-means algorithm called k-means++, which proposes a method to choose the initial seeds before applying the k-means algorithm on the dataset [Arthur and Vassilvitskii, 2007]. When k-means and k-means++ were applied on real-world datasets, it was observed that k-means++ terminates twice as fast as k-means with better results [Arthur and Vassilvitskii, 2007].

2.6 Visualizing Time Series and Time Series Clusters

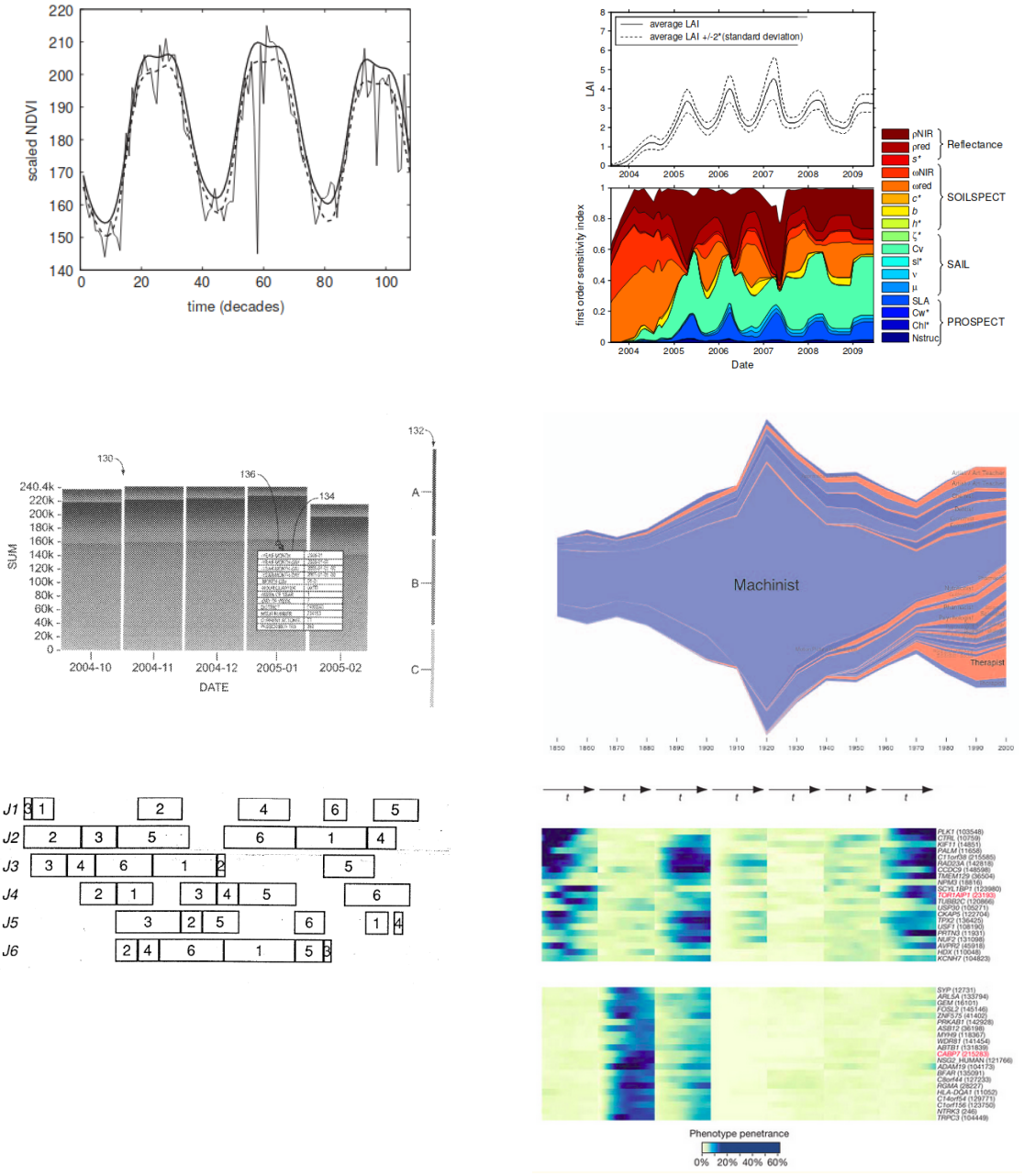
Time series data is generally represented in a two-dimensional form where one axis represents time, generally on a linear scale and the other axis represents non-temporal values of the data points. Examples of time series data represented on a linear scale are line charts, stacked area charts, bar charts, and stream graphs.

In line charts, the time series can be represented with the x-axis representing time and y-axis representing values. In the line chart seen in the top-left image in Figure 2.4, the x-axis represents time, and the solid line represents the Normalized Difference Vegetation Index [Jönsson and Eklundh, 2004].

The stacked area chart is the representation of values of each group on top of each other continuous in the time domain. This graph allows users to see the evolution of values over a period as seen in top-right Figure 2.4 [Le Maire et al., 2011].

The stacked bar chart is the discrete representation of values of each group on top of each other. This graph allows users to see the evolution of values over a period as seen in middle-left Figure 2.4 [Hao et al., 2010].

Figure 2.4: Time series data in the form of multi-line series [Jönsson and Eklundh, 2004](top-left), stacked area chart [Le Maire et al., 2011] (top-right), stacked bar chart (middle-left) [Hao et al., 2010], stream graphs (middle-right) [Bostock and Heer, 2009], gantt charts (bottom-left) [Yamada and Nakano, 1992], heatmap (bottom-right) [Neumann et al., 2010]



Another variation of stacked area chart is stream graphs where the stacked area is displaced around a central horizontal axis. It derives its name from its similarity in appearance to a flowing liquid as seen in middle-right Figure 2.4 [Bostock and Heer, 2009].

Gantt Charts are a highly application specific time series visualization which is used for project management. As can be seen in the bottom left Figure 2.4, this visualization represents the time allotted to different activities and the percentages of the activity completed [Yamada and Nakano, 1992].

Time series data can also be visualized on two scales of time as seen in heat maps where the original data is represented in the form of color intensity on a two-dimensional diagram where one axis can correspond to time as seen in Figure 2.4 [Neumann et al., 2010].

Visualization plays a crucial role in finding periodicity or seasonality in time series data. The simplest way to represent periodical data is to overlay them in cycle plots. Another visualization technique, polar area diagram which is one of the earliest known visualizations, is used to help users understand seasonality by representing stacked area graphs. Statistician Florence Nightingale invented it as a tool for representing causes of mortality in the army from April 1854 to March 1855. As seen in Figure 2.5, this visualization is a modification of stacked area chart from a common vertex. The blue wedges represented the number of deaths from preventable diseases whereas the white wedges represented the deaths from wounds, and the black wedges measured from the center the deaths from all other causes. It played an essential role in conveying that the most significant cause of death during

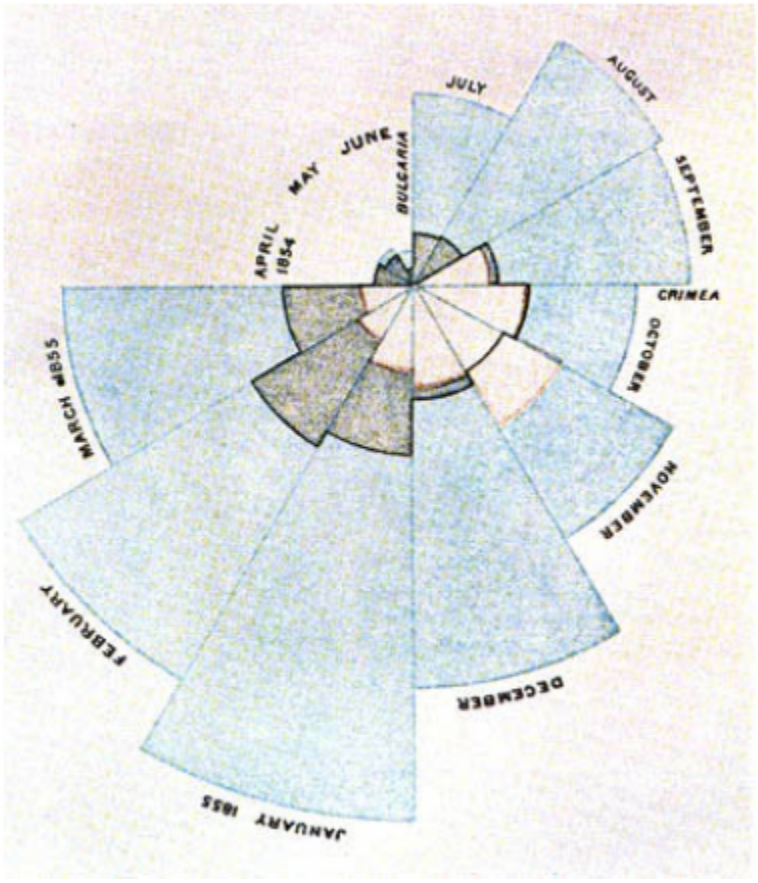


Figure 2.5: Coxcomb charts provided by Florence Nightingale in Notes on matters affecting the health, efficiency, and hospital administration of the British Army [Nightingale, 1858]

the war was not due to war-related casualties but diseases such as cholera, typhus, and dysentery [Nightingale, 1858].

Weber et al. introduced another visualization technique called spirals that helps users find periodic structure in the data [Weber et al., 2001]. They demonstrate its efficacy by presenting a comparison of spirals with traditional bar charts in Figure 2.6 for representing sunshine intensity in the same screen real estate and same coloring scheme. Though this representation technique is useful in

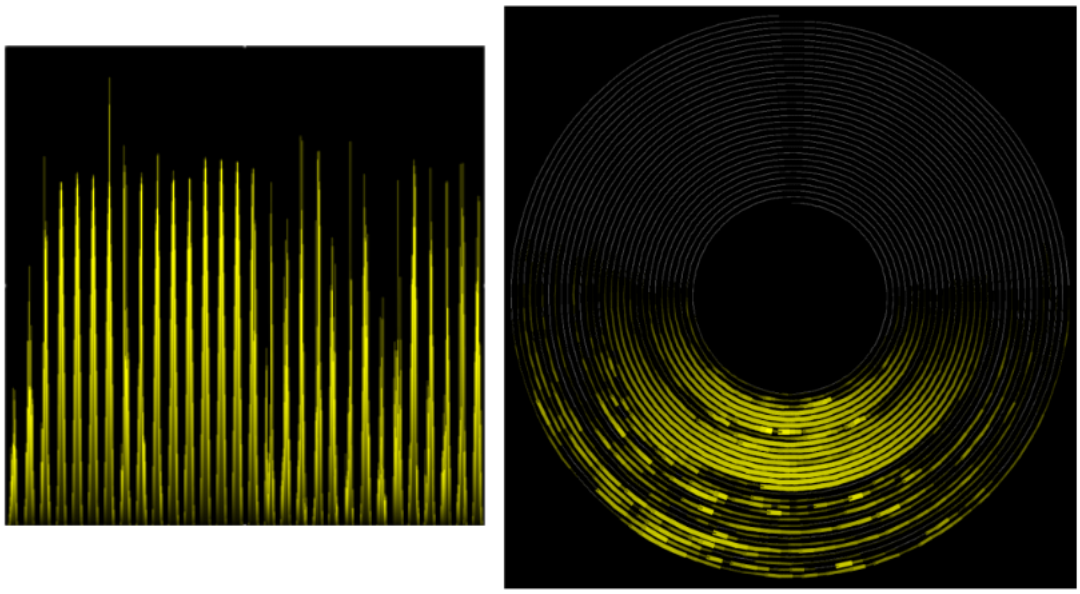


Figure 2.6: Visualization of sunshine intensity using bar chart and spiral chart. It is easier to observe trends in periodicity and sunset-sunrise patterns in spiral chart compared to the bar chart [Weber et al., 2001]

finding periodic elements in time series, it suffers from its inability to represent correlation in addition to periodicity.

There has been significant research in understanding univariate time series data by clustering it first and then visualizing the clusters. Van Wijk and van Selow developed a visualization technique to identify trends and patterns in energy consumption and represent them [van Wijk and van Selow, 1999]. Van Wijk and van Selow cluster data by their similarity in the day patterns, and then represent them using hierarchical dendrograms. However, due to shortcomings of representing time series clusters in the form of the dendrogram, they propose a new visualization that helps users answer interesting questions on seasonality and

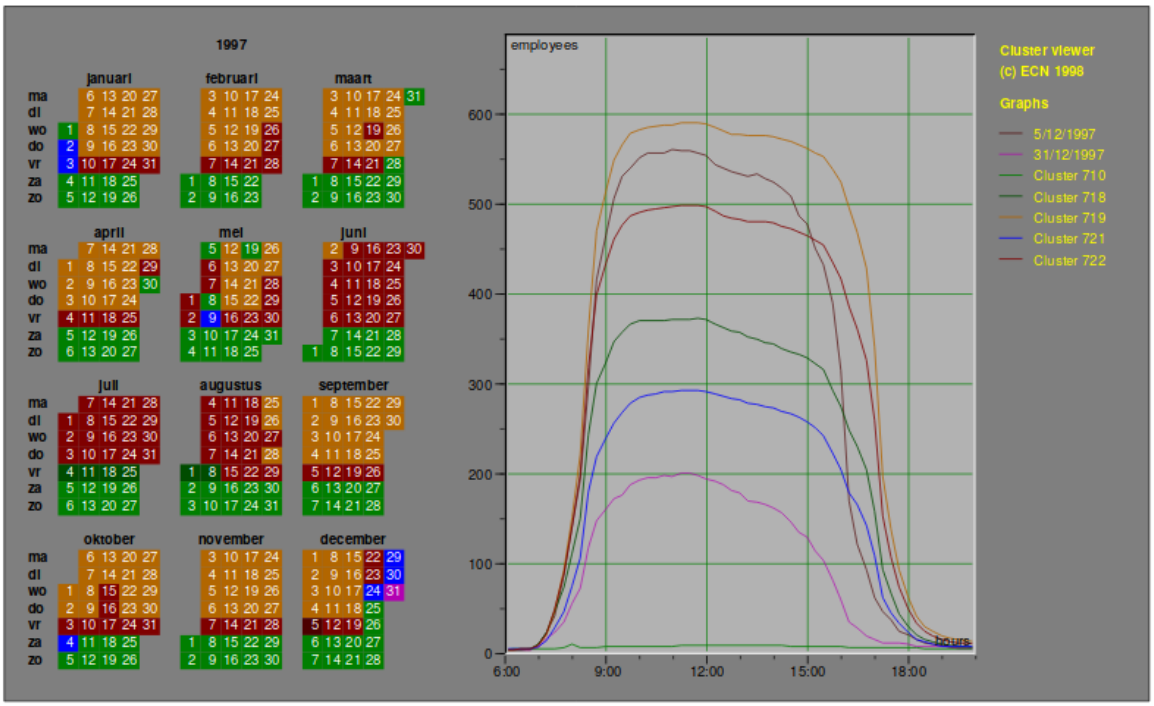


Figure 2.7: Calendar and hourly view of Employee clusters to understand Energy Usage 1998 [van Wijk and van Selow, 1999]

patterns as can be seen in Figure 2.7. In this figure, we view the results of clustering employee attendance data for a research facility ECN. A particular color characterizes each employee cluster. On the left-hand side, we can see that van Wijk and van Selow have colored each day on the calendar according to its original cluster whereas, on the right-hand side, we can see the average value per cluster [van Wijk and van Selow, 1999]. From the right-hand side of Figure 2.7, we can see the pattern in office hours where most employees arrive at 8:30 am and leave by 6:00 pm. We can also observe from the left-hand side figure that the attendance is less on Fridays and in the summer [van Wijk and van Selow, 1999].

Time series bitmaps are an alternative to representing time series in non-

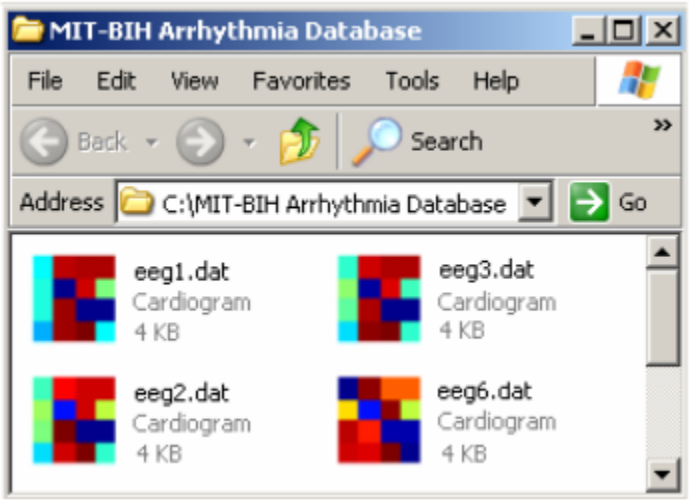


Figure 2.8: SAX representation of EEG Time Series Data [Kumar et al., 2005]

traditional methods. Kumar et al. use this method to represent four different EEG time series data into SAX symbols as seen in Figure 2.8 [Kumar et al., 2005]. Symbolic algebraic approximation (SAX) transforms the continuous time series data points by lower bounding the values to create discrete values. The advantage of using SAX symbols is that it allows users to cluster time series of different lengths. It is an efficient means of representing time series clusters. In Figure 2.8, all the SAX symbols represent congestive heart failure EEGs, but eeg6.dat is different from the remaining dat files because eeg1.dat, eeg2.dat, and eeg3.dat belong to one individual and eeg6.dat belongs to a different individual. This distinction is easily observable in Figure 2.8. Kumar et al. start by using the symbolic aggregate approximation for converting continuous time series data into discrete symbols. These discrete values are then clustered using hierarchical clustering and represented using a combination of three components: dendrogram, line charts, and SAX representations. We can see the clustered view for the SAX symbols in Figure 2.9 that groups similar looking

symbols using hierarchical clustering.

In conclusion, the visualization of time series data is limited to standard visualization methods. Moreover, the visualization methods for viewing time series clusters and correlations are highly application specific and restricted. In this research, we try to propose a visualization that represents time series clusters and views correlations between two different time series data. The users can test the clusters produced as a result of different clustering algorithms and different distance metrics. It allows users to view the tightness and distribution within a cluster statistically. The proposed visualization also allows the users to view the effect of one time series on a simultaneously occurring time series. It also gives users an integrated view of both the time series in one visualization that helps them understand the correlation between two different time series clusters.

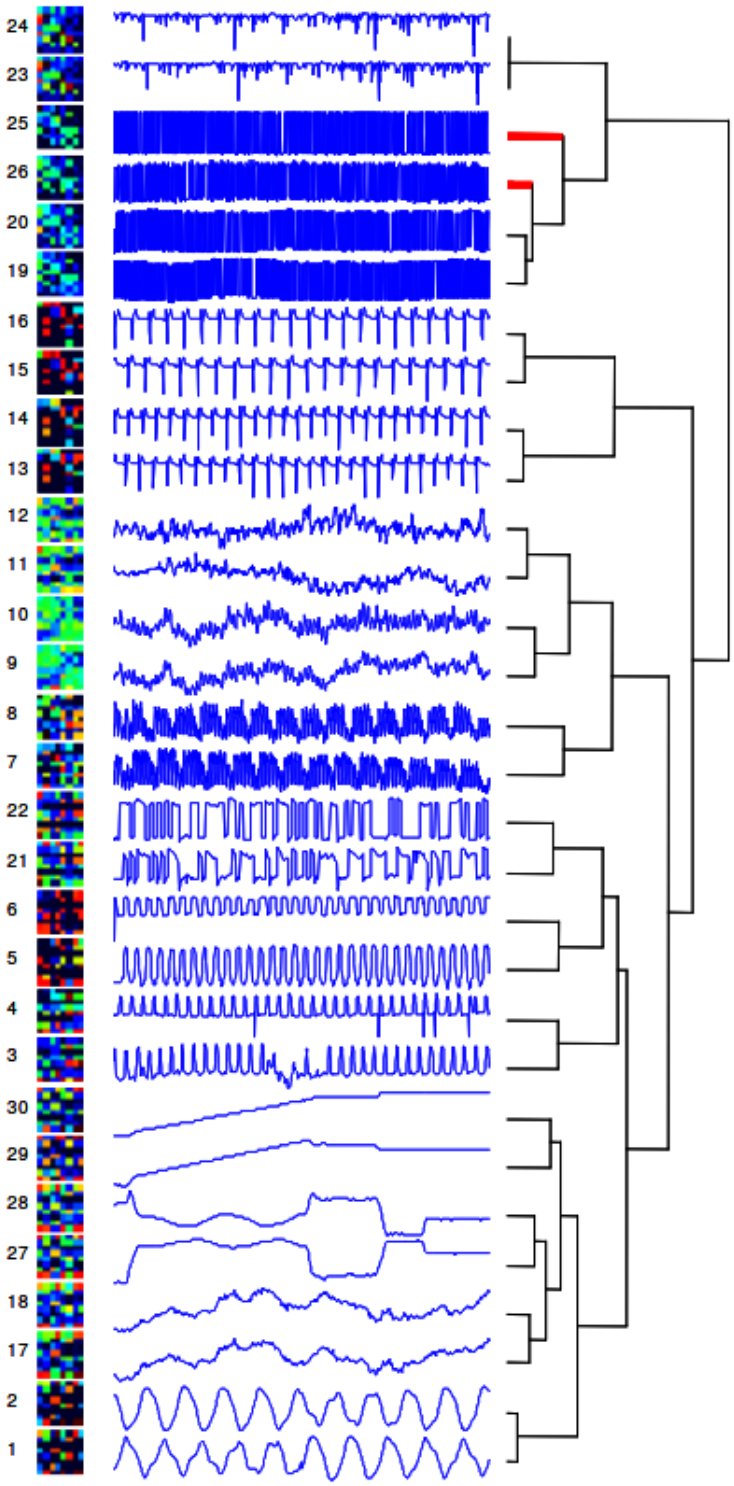


Figure 2.9: Hierarchical clustering of time series data represented in the form of SAX symbols [Kumar et al., 2005]

Chapter 3

Approach

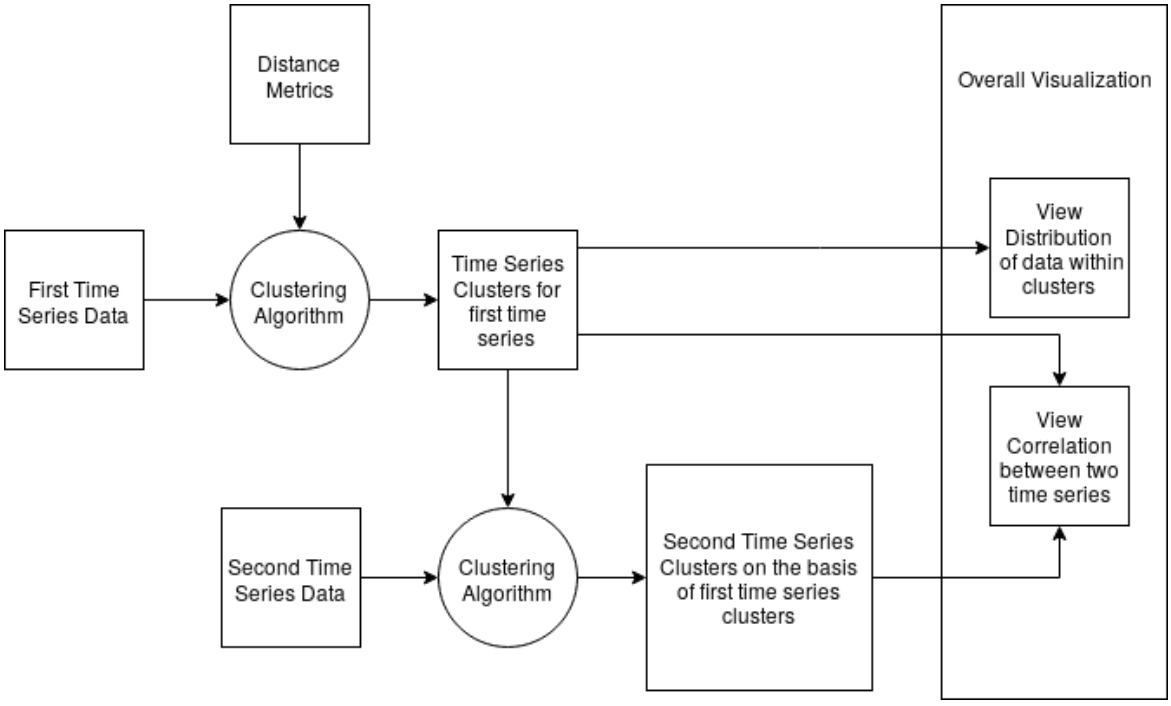


Figure 3.1: Approach overview

Figure 3.1 provides a graphical depiction of the overall flow of creating the final visualization. The approach to visualizing time series clusters and correlation is first to cluster the time series data into clusters using different clustering methods and distance metrics. Once we group the time series with similar characteristics into clusters, we can use the visualization to see the distribution within a cluster and determine the tightness and variability of data points within the cluster. To determine the correlation with a simultaneously occurring second time series, we

cluster the second time series by clusters created for the first time series. This second time series clusters are visualized in the same panel and integrated within the original time series clusters to help users understand the effects of causality and correlation. Furthermore, we introduce a panel which shows additional details about the clusters formed. The remainder of the Approach chapter is broken into three distinct sections - clustering, visualizing the distribution of data points within clusters, and visualizing correlations.

3.1 Clustering

The time series dataset can be massive, and the data points in the series could be overlapping in nature. Hence, we cannot visualize and understand the individual elements easily. As a result, we can apply clustering algorithms on time series to organize related sets or groupings based on their similarity. Clustering helps the visualization by reducing the overall clutter at the same time maintaining the integrity of the data by grouping similar characteristic objects to each other. Visualizing the time series clusters enables the users to focus on the few data points that summarize the entire dataset. If randomly chosen time series were grouped instead of using a well-defined clustering technique, the average shape would have no meaningful representation and fail to provide a context for comparison.

Liao reviews some of the clustering algorithms and distance metrics generally used for time series analysis [Liao, 2005]. The following algorithms are used for time series data clustering: relocation clustering, k-means, fuzzy c-means,

agglomerative hierarchical clustering and self-organizing maps. Liao also discusses different distance metrics such as Euclidean distance, root mean square distance, Minkowski distance, Pearson's correlation coefficient, Dynamic Time Warping, and KullbackLiebler distance [Liao, 2005].

For clustering time series data, we use two hard clustering techniques - k-means algorithm and hierarchical clustering because they are most commonly used for clustering time series data. The distance metrics available to use for the clustering algorithms are Euclidean, Manhattan, Minkowski, DTW, and Chebyshev. For hierarchical clustering, the available linkage criteria are the single linkage, complete linkage, and average linkage. For this use case, we prefer to use the k-means algorithm since it allows a choice for the number of clusters to be represented. Though, we identify an ideal number of clusters using the elbow and silhouette method. We choose a value of k ranging between four to ten such that it is a balance between representing the data at the same time it is small enough not to clutter the visualization.

3.2 Visualizing distribution of data points within clusters

Assuming that we have performed the clustering and we have used suitable metrics to determine the effectiveness of different clustering techniques, now we would like the user to be able to understand the variability of the data points within the cluster.

For this approach, we will employ a superposition of box plot like structure

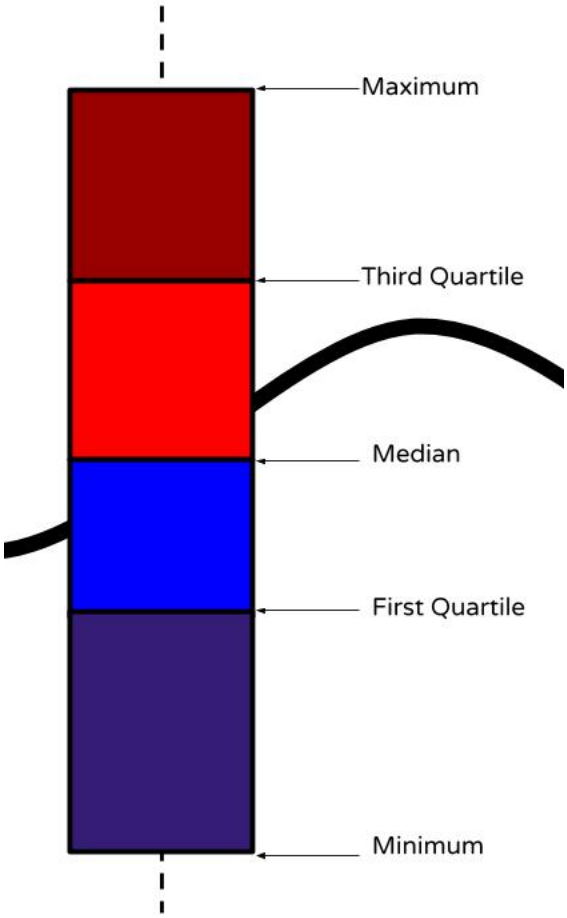


Figure 3.2: Box plot structure in the form of stacked bar charts to represent distribution of the data points within a cluster

over time series data as seen on Figure 3.2 such that it represents the distribution of values within the cluster. A box plot is an effective way of displaying the distribution of data with the use of quartiles and different categories. In Figure 3.2, the box plot like structure consists of four rectangles each representing a statistical range. The dark blue rectangle represents the range from minimum to the first quartile. The light blue rectangle represents the range from the first quartile to median. The light red rectangle represents the range from median to the third quartile.

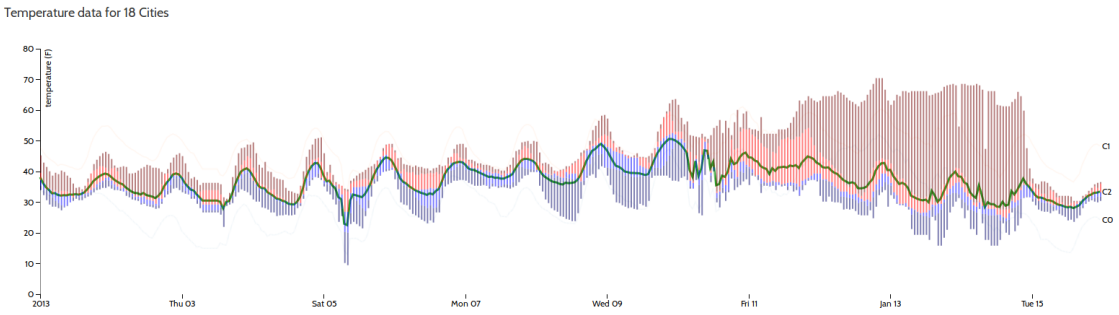


Figure 3.3: Distribution of data within a bad cluster when k=3

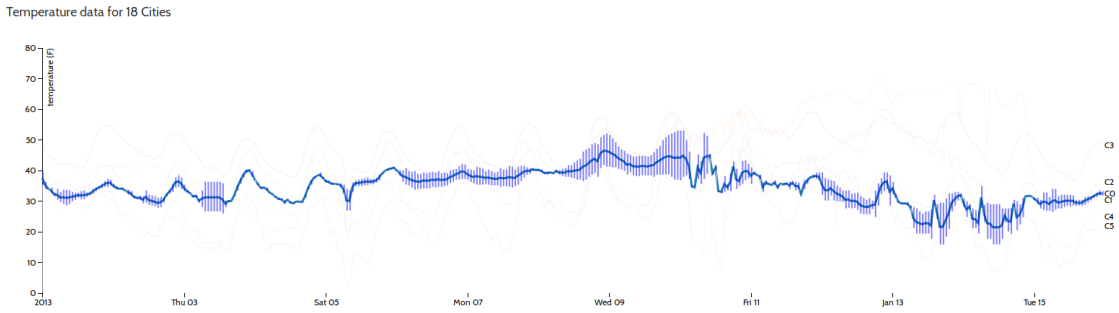


Figure 3.4: Distribution of data within a good cluster when k=6

Moreover, the dark red rectangle represents the range from the third quartile to the maximum. In Figure 3.2, each box plot like structure is superimposed over the line chart representing the time series. The box plot like structure represents the distribution of values for a particular time on the line chart.

The reason behind the choice for box plots was that users could use box plots to answer questions about average, median, quartiles, minimum and maximum. Users could also use it to determine outliers and their values. Users could also use this visualization element to understand how tightly the data were grouped and if there was any symmetry in the data.

In Figure 3.3 and in Figure 3.4, we can see two time series clusters for temperature of 18 cities for k=3 and k=6 respectively. As can be seen from these

figures, the cluster distribution is tighter for k=6 in Figure 3.4 compared to k=3 in Figure 3.3, indicating a better cluster quality. In both the figures, the values within the highlighted cluster is spread across some ranges but in case of Figure 3.4 overall distribution of the values within the cluster is narrower, thereby indicating a tighter cluster.

3.3 Visualizing Correlations

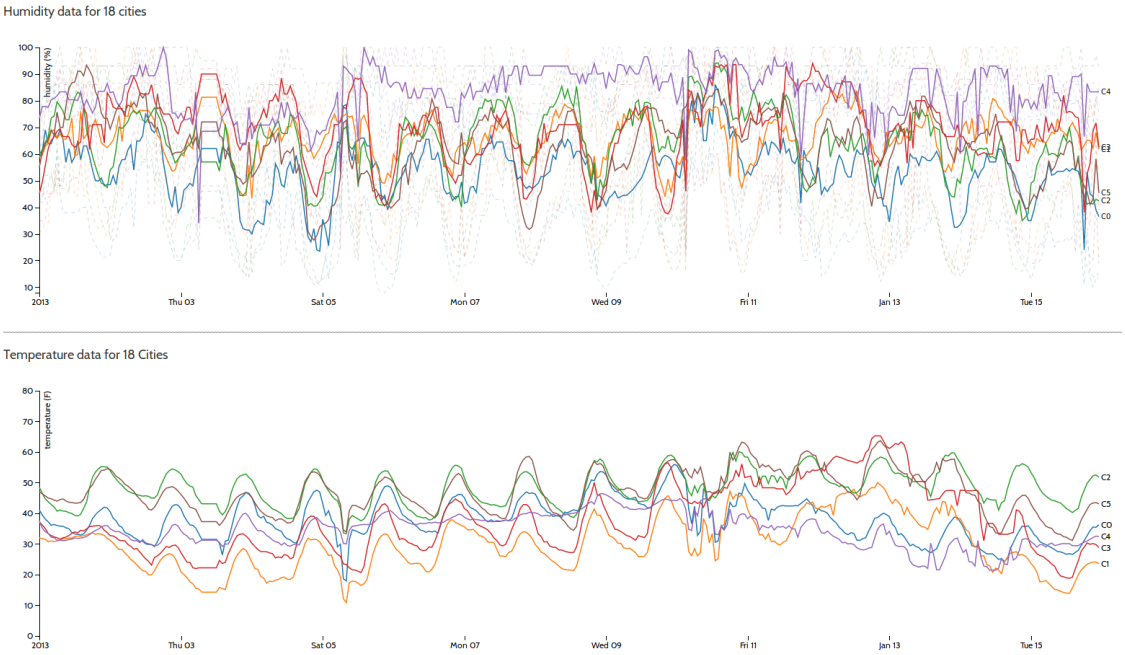


Figure 3.5: Panel View with two time series represented individually

In addition to being able to cluster data and view the distribution of the time series data, we would like to enable users in understanding how two different time series data correlate to each other. One way of implementing correlation between both the time series is to view them on the same panel, with refactored and rescaled y-scales for each of the individual time series graphs as can be seen in Figure 3.5.

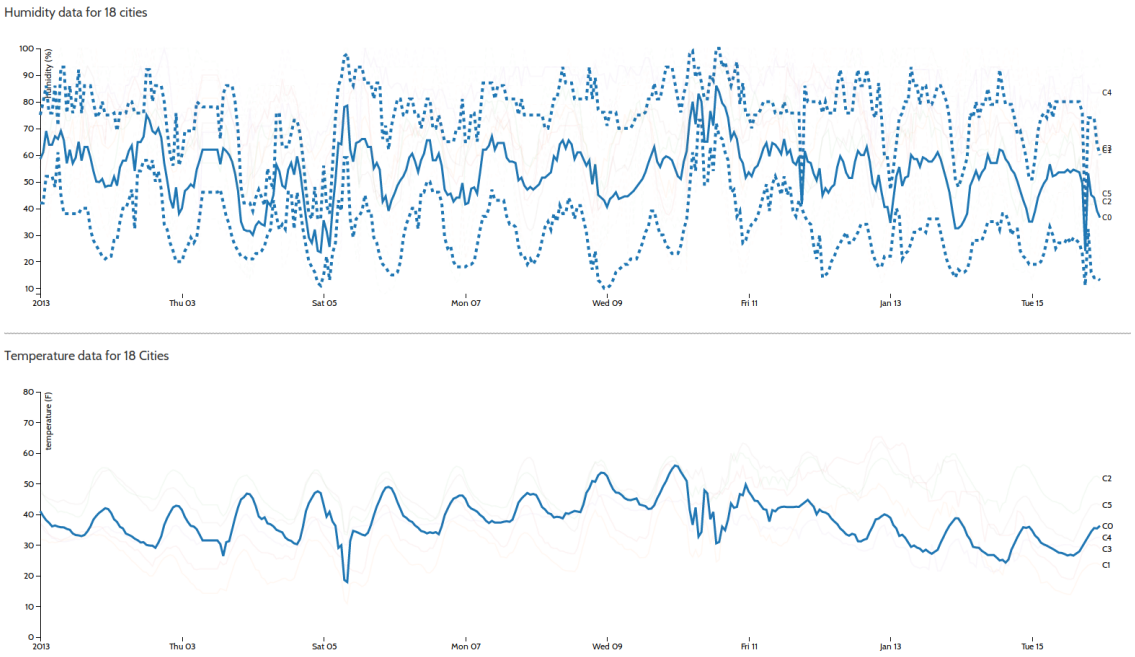


Figure 3.6: Panel View to focus on one cluster in both the time series simultaneously

In this figure, we are looking at the humidity and temperature time series data for 18 cities when grouped into six clusters. This design choice leads to a visualization where it is difficult to understand if there exists any correlation between the two time series clusters. Even if we were able to focus on one cluster in both the time series data as seen in Figure 3.6, we would still be unable to understand if there was a correlation between the two time series.

Another novel way of integrating both the time series is to implement the events of one time series in the form of line charts and the second time series in the form of circles scattered according to their occurrence and their radius corresponding to their intensity as seen in Figure 3.7. The radius of the circles are normalized with respect to the largest value in the second time series cluster since the correlation for different time series could be different. The presence of circles localized in a time

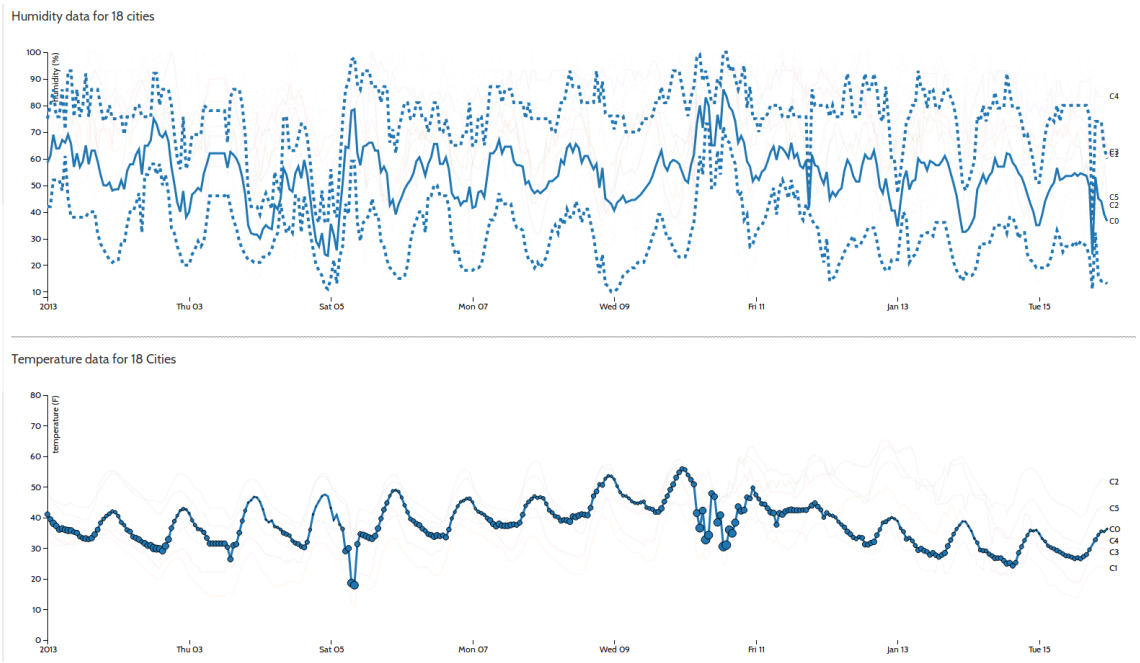


Figure 3.7: Multi-coordinated view of both the time series in addition to an integrated view to show correlation between two time series data

period may also indicate the presence of an event indicating correlation. The circles are centered at the time where there is a correlation between the two time series. This approach resolves the shortcomings with the multiple coordinated view. The user can now focus on both the time series data in the same graph.

To solve the problems created by superimposing both the visualizations, we stick to using two independent representation of the time series, but this time in the form of a multi-coordinated view as seen in Figure 3.7. Though this requires users to coordinate between both the time series to understand any correlation effects, this creates less clutter and brings more focus to the time series data. This multi-coordinated visualization technique helps transform the visualization into an exploratory form of tool.

Chapter 4

Case Study - Weather data

In this section, we discuss the results of the visualization techniques to help users understand the clustering and correlation of hourly weather time series data for 18 cities over a span of the first half of January 2013 [Mario et al., 2017]. For this application, we analyze the hourly temperature and humidity time series data during the same time domain.

4.1 Visualization Goals

The aim of using this visualization method is to understand how the temperature data for 18 cities can be summarized and if there lies a correlation between the temperature and humidity data of these cities. We would like the users to be able to answer the following questions about this case study through this visualization:

1. How can the overall temperature data for these 18 cities can be summarized efficiently? Which cities have similar trends in weather for the first half of January 2013?

2. What are the chances that it is going to be humid when temperature is low and when it is high?

3. Which group of cities have the highest and lowest temperature doing a particular time period?

4. Is there any correlation evident between temperature and pressure data to make a general statement about temperature and humidity?

5. What are the maximum and minimum temperatures within a group of cities that have similar temperature trajectories?

4.2 Data Ingestion

The temperature data and humidity data for 18 cities is available in two separate CSV files obtained from Kaggle [Mario et al., 2017]. The dataset was originally acquired using Weather API on the OpenWeatherMap website.

The temperature data contains columns for every city and their temperature at different times. The pressure data contains the records of the city and their humidity at the same times as the temperature data. The rows in this file correspond to the time series temperature and humidity data for 18 cities in the United States in first half of January 2013. This time span and the cities were randomly selected. The temporal data in both the CSV files are regularly spaced with hourly intervals between subsequent time data points. The temperature is present in kelvin (K) temperature scale, but it is transformed to Fahrenheit since it is easier to perceive city temperatures in Fahrenheit. The humidity data is represented in percentage.

We ingest the data from this file into the visualization framework with the use of a D3 API called d3.csv() function which contains a path parameter to the

CSV file. It transforms the entire CSV file into an array of objects which undergoes preprocessing to be available for creating visualization elements in the tool.

4.3 Data Preprocessing

The temperature data for the 18 cities for the first first half of January 2013 is now ready for ingestion. It does not require any preprocessing before it can be used by our visualization method.

However, some of the entries in the humidity CSV file are missing for the 18 cities during the interval chosen for analysis. As a result, we explore two options that help perform data analysis in case of missing values [Sarstedt and Mooi, 2014]. The first approach is to omit the entries for missing values for all the cities, but this is not a good choice for our visualization technique since it converts the time series into an irregularly spaced time series. The second approach is to fill the missing values with the average of all the other values present for the entry. We employ the second approach as it helps us retain the regularity in the time intervals between subsequent entries.

4.4 Deciding the number of clusters

We can see the elbow plots for each distance metric in Figure 4.1. As we can see from the figures, DTW and Minkowski are the best choices among all the distance metrics because they have less variation in values indicating that the clusters produced generally converge to give the same results. These two distance

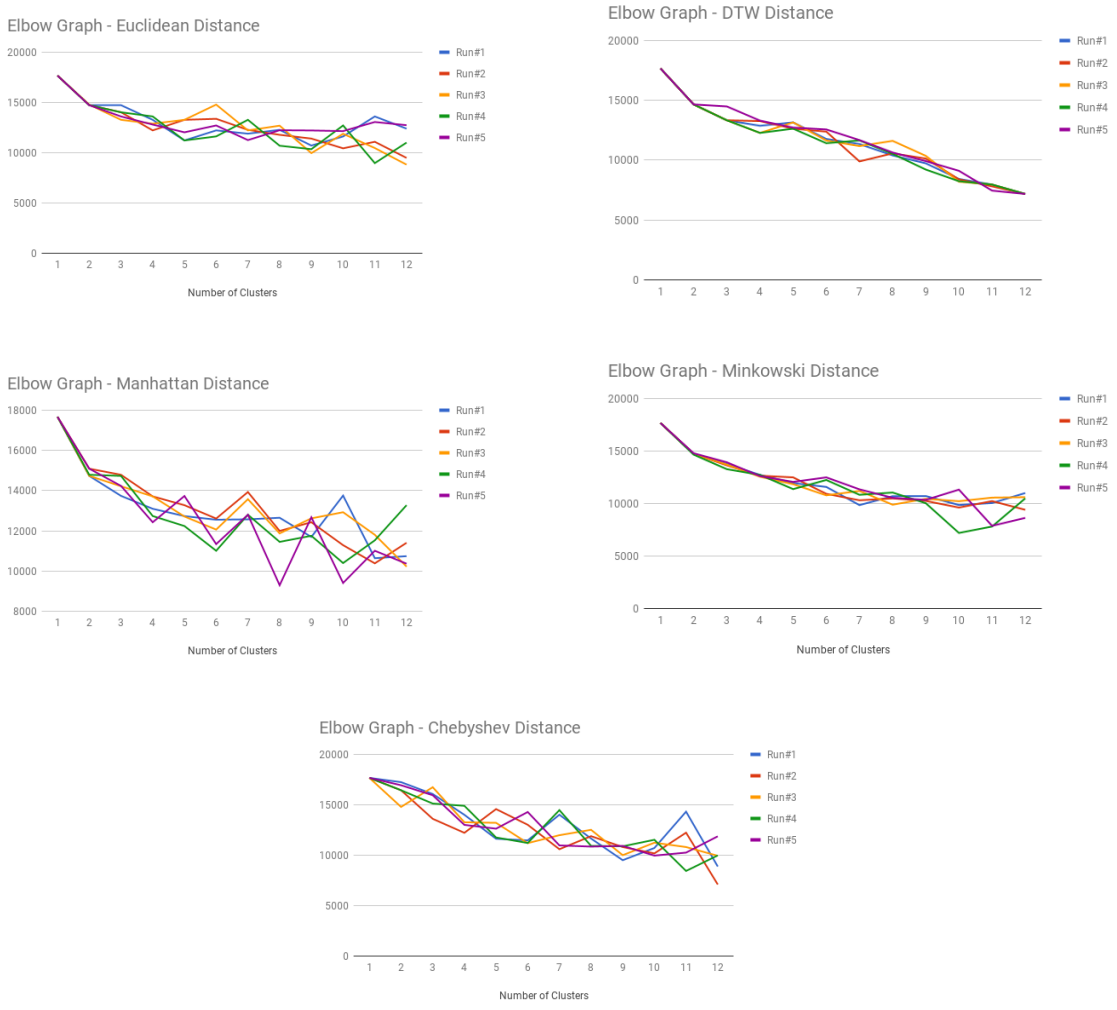


Figure 4.1: Sum of squared errors vs Cluster size for different distance metrics - Euclidean, Dynamic Time Warping, Manhattan, Minkowski, and Chebyshev

metrics also generally tend to decrease in the sum of squared errors as the number of clusters increase. For this particular application, we will use DTW distance metric since the sum of squared errors for DTW for the same cluster size were lower than that of Minkowski.

Now in the elbow graphs, there is no sharp elbow joint present, which is one of the shortcomings of using an elbow graph. We need to find a value which allows us to identify many distinct patterns in the data. Choosing a higher value for the number of clusters is still not a disadvantage since the sum of square error decreases with the increase in clusters. However, this value has to be small enough not to make the visualization cluttered. As a result, we will be choosing a value of k=4 since it is small enough not to clutter the visualization and large enough to give us distinct patterns in the temperature data.

4.5 Shape Based Clustering of Time Series Data

Figure 4.2 represents the temperature in its unclustered format. Here the x-axis represents the time, and the y-axis represents the temperature in Fahrenheit. A single colored line in this visualization represents a temperature through the first half of January. This visualization is highly cluttered and is completely unusable as an exploratory tool. It not only fails to give any intuition on the trends and patterns in the scores of the cities, but it also fails to answer the visualization goals we began with while developing this visualization method. Despite its cluttered format, we can observe that there are many lines which have similar trajectories through time.

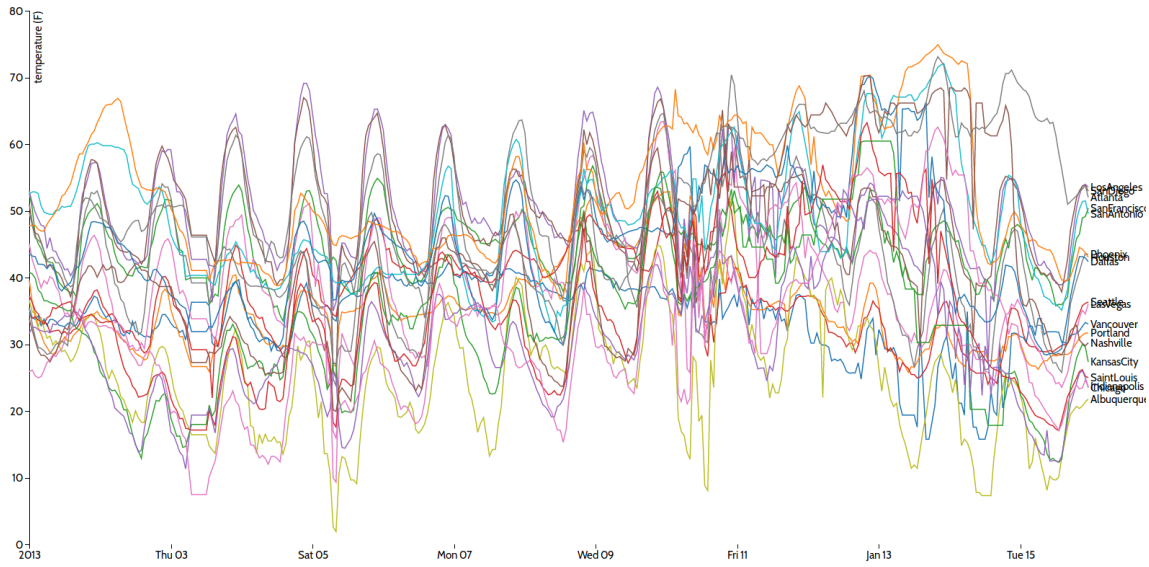


Figure 4.2: City temperature data for first half of January 2013 in unclustered format

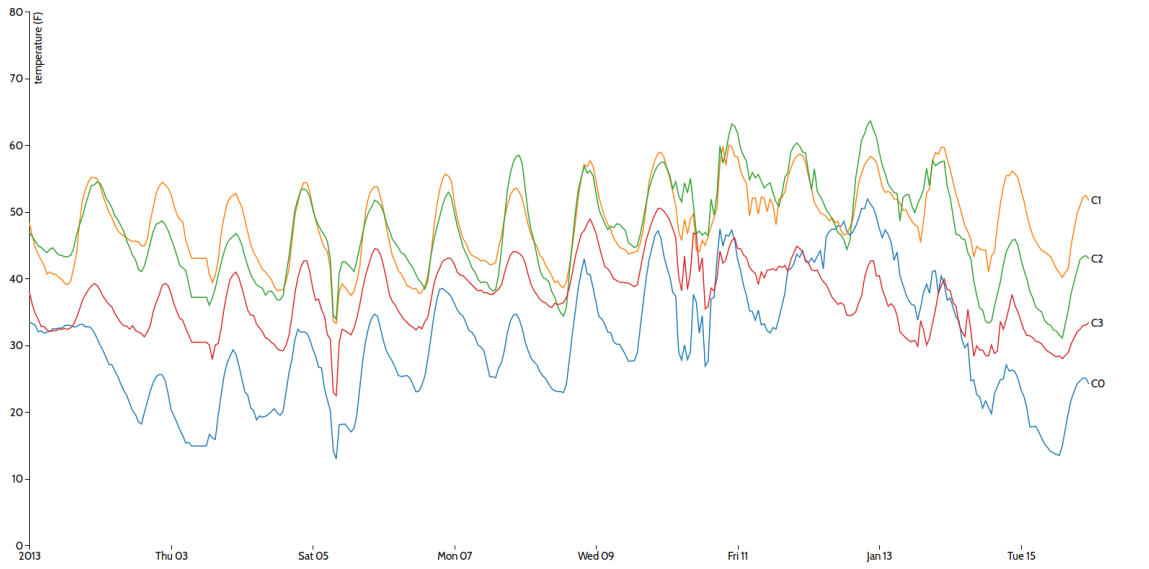


Figure 4.3: City temperature data for first half of January 2013 clustered based on shape

We can cluster these similar trajectory lines by clustering this time series data by shape.

The clustering algorithm used for this case study was a standard K-means algorithm with four clusters and a thousand iterations. The reason we chose four

clusters was that it is small enough not to clutter the visualization, and large enough to give us distinct patterns in the temperature data. Moreover, a thousand iterations were considered a good choice because the data points converged successfully into these four clusters by the end of the thousandth iterations. The distance metric chosen for the k-means algorithm was dynamic time warping. The rationale behind this choice for the distance metric was this metric could effectively group two time series having the same trend in temperature data even if they were shifted by some time interval. As can be seen from Figure 4.3, the visualization now contains only four clusters that adequately represent the temperature of the 18 cities for first half of January 2013. This can be observed from the fact that the time series clusters closely represents the original time series data. Also the distribution of the data within a cluster represents the original data points, thereby maintaining the integrity of the original data points. From this figure 4.3, we can see that the trends of peaks and valleys are the same for all the clusters indicating the same periods for rise and fall in temperature during the day. Clustering helps the user to group together similar shaped time series data into one cluster, thereby reducing the overall clutter in the visualization.

4.6 Behavior-based clustering of Time Series Data

Clustering the cities temperature data by shape allows us to view groups of cities that have a similar trend in temperature variation in the first first half of January. The overall clustered view of the temperature allows us to view the

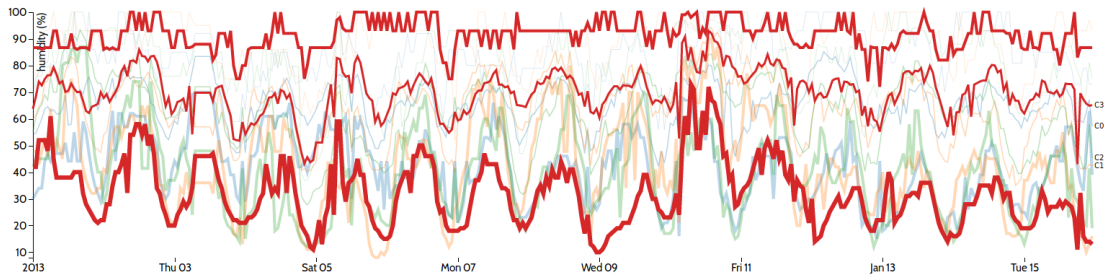


Figure 4.4: Clustered view of humidity data for 18 cities when clustered on the basis of similar temperature trajectories

summarized dataset with ease.

In addition, we would also like to know if the temperature data for these cities is in any way correlated to the humidity data for these cities during the same time span. Ideally, there is an inverse correlation between temperature and relative humidity. The higher the temperature, higher the air molecules can hold onto the water molecules. As a result, the relative humidity decreases with rise in temperatures.

To be able to correlate the city temperature clusters with their corresponding humidity data, we need to cluster the humidity data by the temperature clusters. Therefore, clustering the humidity by temperature clusters reduces the overall visual elements in the visualization as seen in Figure 4.4. In this figure, the x-axis represents time in hourly intervals, and the y-axis represents the humidity for the clusters measured in percentage. In Figure 4.4, we can highlight a particular cluster in the temperature panel, and three corresponding humidity lines get highlighted in the humidity panel. The thickest line indicates the lowest humidity in that cluster; the second thickest line indicates the highest humidity in that

cluster and the thinnest line indicates the average of the humidity values in that cluster. The intuition behind indicating all these three values is that the average value by itself is not an accurate measure of the humidity values in the cluster since the humidity time series are not clustered by their similarity in shape. As a result, for a particular time period, the humidity cluster can have widely varying values. Hence, we would like the users to be able to view the distribution of values in the humidity clusters as well.

4.7 Statistical Attributes of Time Series Data

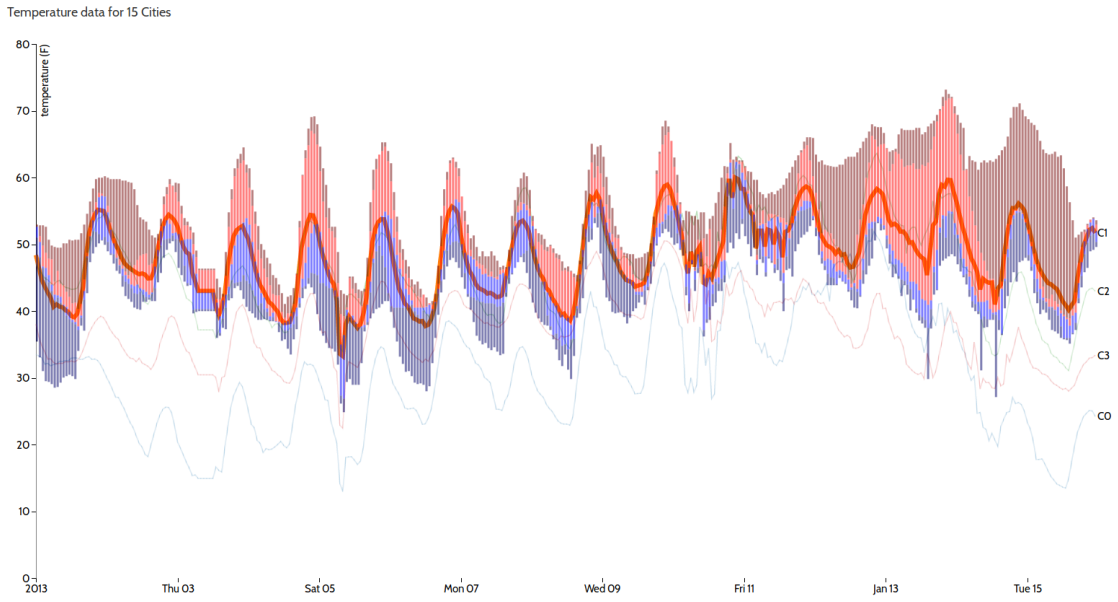


Figure 4.5: Visualizing the distribution within temperature data for 18 cities in the first first half of January 2013

With this visualization, one of the key questions we would like to answer are which cities have similar temperature patterns and if they have similar patterns how

close are they to each other. With that aim in mind, we have integrated floating box plots superimposed over the temperature line chart to show the variability of the data within a cluster. It also gives us the trends on the maximum and minimum temperature that a group of cities may have.

With clustering, we have removed the unnecessary clutter from the visualization, but we have also abstracted out the unique and interesting occurrences in the temperature dataset. As a result, we would like the user to be able to see the spread of the cluster so that they can understand how tight was the cluster. In the case of time series data, it may so happen that the tightness in a cluster is localized for a particular time that is responsible mostly for selecting a data point within a cluster. This tightness at a particular time period means that at that time period, all the data points have very close temperature and the variability is very less. We will discuss this further in detail with examples in the Results section.

The original line series representing the clusters characterizes the geometric mean of all the data points within the cluster. We can see additional statistical characteristics for the cities cluster such as median, maximum and minimum. As we can see in Figure 4.8, we have represented the distribution of the cluster at a particular time with a stacked bar chart which functions like a box plot is providing us with insights on the maximum, minimum, median, upper quartiles, lower quartiles, and outliers. These overlaid stacked bar charts provide additional answers to questions such as what is the highest temperature among the cluster.

4.8 Results

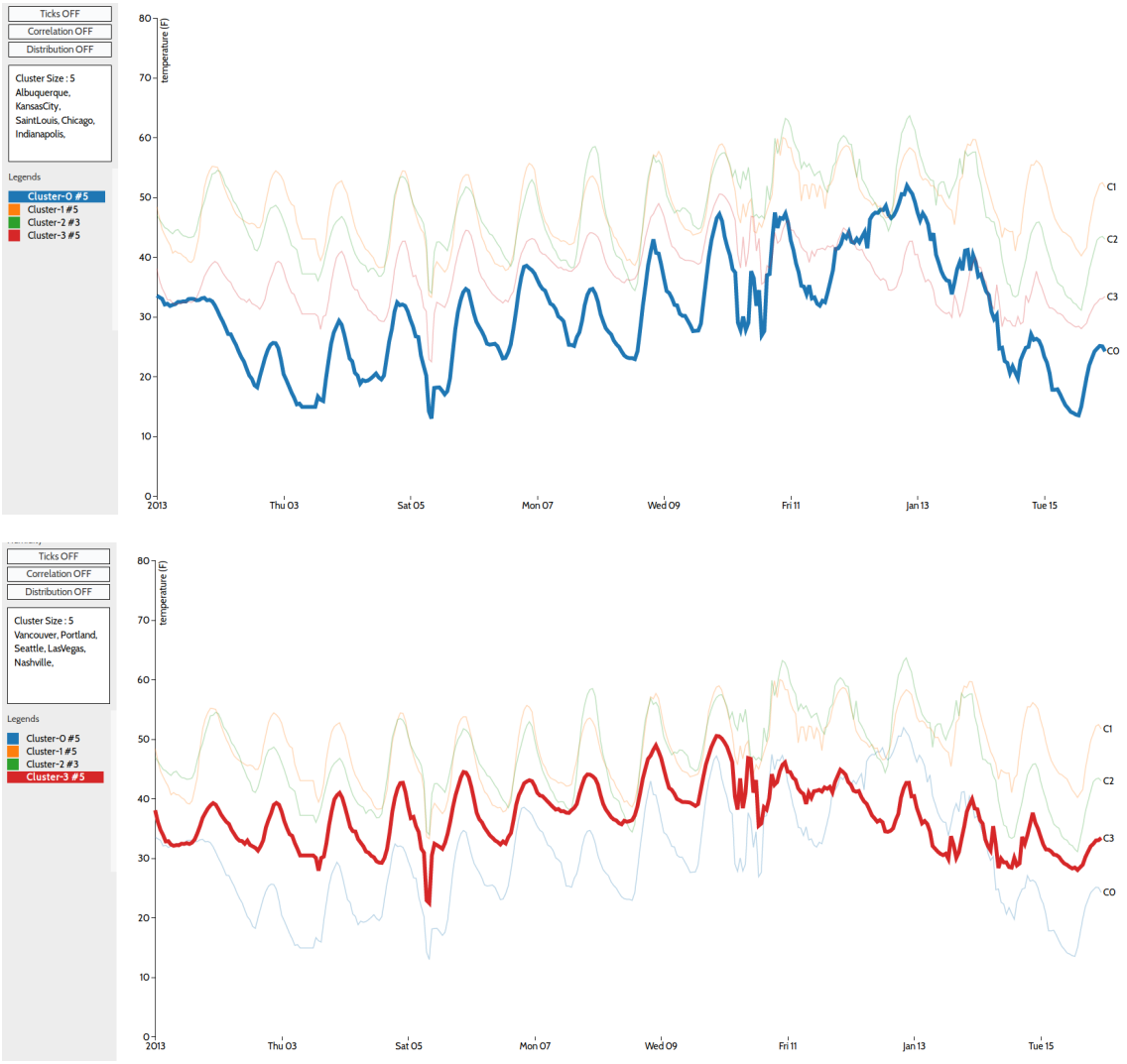


Figure 4.6: Temperature Cluster - Blue and Red

We have summarized the weather dataset for 18 cities into four clusters, which groups the cities with time series temperature data together. From Figure 4.6 and Figure 4.7, we can see the four clusters each distinguishable with a unique color and additional details displayed on the left hand side panel. We can see the size of the cluster and the cities grouped together within this cluster, indicating that these

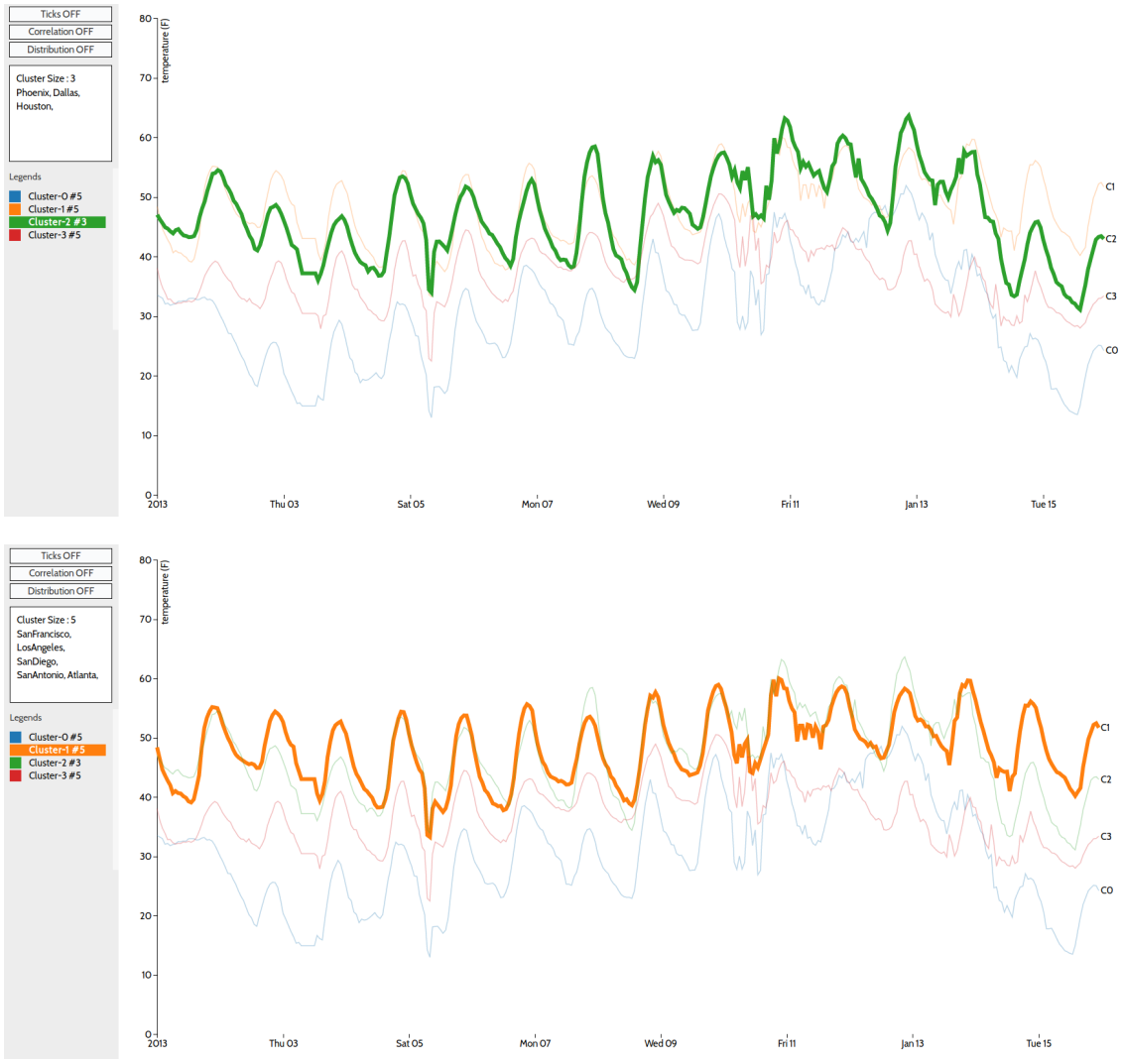


Figure 4.7: Temperature Cluster - Green and Orange

cities have a similar temperature trajectory.

As can be seen from Figure 4.6, we can say that during that the blue cluster (containing Albuquerque, Kansas City, Saint Louis, Chicago, and Indianapolis) has the lowest temperature until 11 January where the red cluster (containing Vancouver, Portland, Seattle, Las Vegas, and Nashville) intersperses to be the lowest temperature up to 14 January.

Similarly, from Figure 4.7, we can say that during the entire first half of January, the green cluster (containing Phoenix, Dallas, and Houston) and orange cluster (containing San Francisco, Los Angeles, San Diego, San Antonio, and Atlanta) superimpose over each other for specific time periods to have the highest temperature.

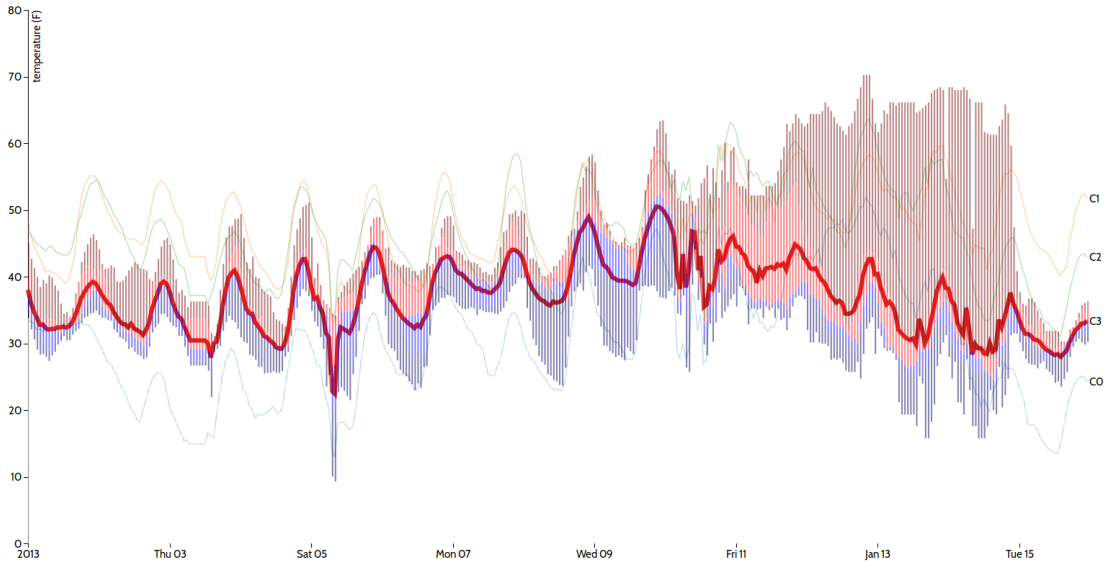


Figure 4.8: Visualizing the distribution within red cluster containing Vancouver, Portland, Seattle, Las Vegas, and Nashville

We can also see how faithfully the data points within a cluster follow the

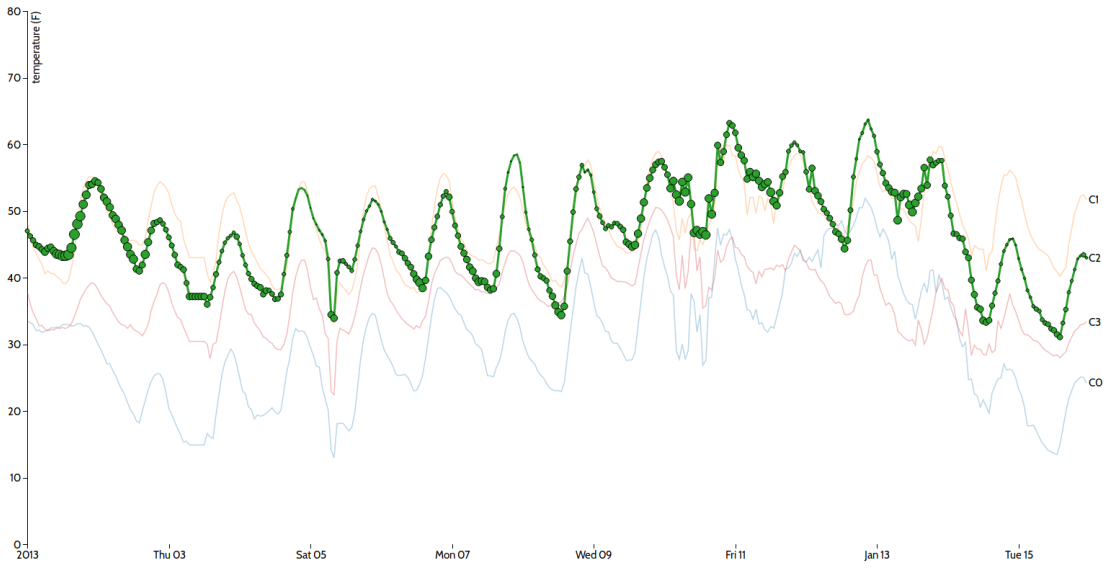


Figure 4.9: Visualizing the effects of humidity on the temperature of the cities in the green cluster

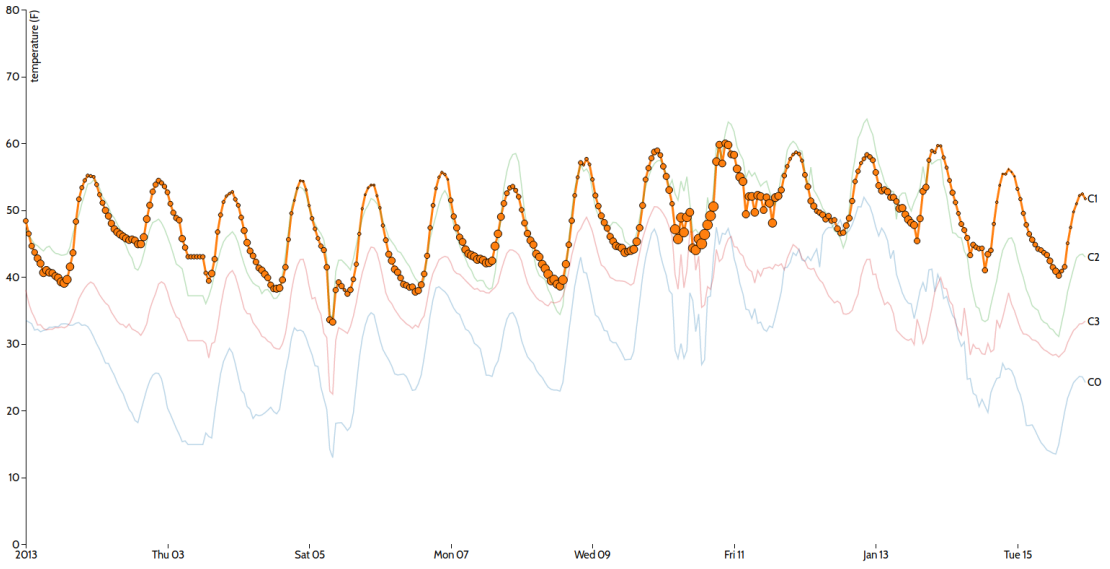


Figure 4.10: Visualizing the effects of humidity on the temperature of the cities in the orange cluster

cluster. In Figure 4.8, we can see that the red cluster is tight up to the first seven days in January compared to the remaining seven days where we can see that the

clustering is not tight and the variability is too significant. This considerable variation in temperature values indicates the presence of outliers in the second half of the 14 day period.

Looking at the correlation of humidity data superimposed on temperature data for each cluster, we get a visualization as can be seen in Figure 4.9 and Figure 4.10. In these figures, we look at the correlation of humidity data for the orange and green cluster respectively. On a preliminary observation, we can notice that the larger radius circles are concentrated on the valleys, and the smaller radius circles are concentrated on the peaks. The radius of the circle is proportional to the humidity value at the corresponding temperature at that time. The larger the circle, the larger is the humidity value and the smaller the circle, the smaller is the humidity value. So from both the integrated views of the green and orange clusters, we can say that for higher temperature, the radius of the circles is smaller indicating a low humidity value, whereas the opposite can be said for lower temperatures where the large radius of the integrated circles indicate high humidity value. In conclusion, we can observe an inverse correlation between the temperature and humidity at the same time. This kind of inverse correlation pattern can be observed in all the clusters. It is more prominent in the green and orange clusters compared to the remaining two clusters.

Theoretically, it is generally stated that the higher the temperature, the more moisture it can hold, as a result the relative humidity decreases, which is evident in this visualization.

Chapter 5

Case Study - Information Visualization in Education

Information visualization is an indispensable part of data mining and analysis. It relies heavily on human cognition to identify unstructured insights by visualizing the data and identifying exciting aspects in it. Current research in education involves helping academicians understand the role of student performance and the effects of several factors on it. Information visualization makes it easier to identify critical effects of such factors on student data. In this case study, we focus on the use of the proposed visualization method to understand the relationship between time series help-seeking behavior data and simultaneously occurring performance time series data for the introductory course of CMSC 201 for two semesters. This method can be used for understanding the correlation between any two generalized simultaneously occurring time series data.

5.1 Visualization Goals

The aim of using this visualization method is to understand if there is a correlation between the performance of students in the course and the office hours attendance of the students. We would like the users to be able to answer the following questions about this case study through this visualization:

1. How can the overall performance of a class be represented efficiently?

2. How does the overall performance of the entire class for a course look with assistance and without assistance?

3. Which office hour events in the semester have the most significant/least impact on the student's grade?

4. Which were the best performing groups, lowest performing groups, groups with maximum improvement and most significant decline? How does the office hour attendance look for each of these groups?

5. For which event and duration of help-seeking behavior was there the most impact on a student's grade?

6. How many times did attending an office hour lead to an improvement in grade for a student? How many times has missing office hours result in a decline in grade?

7. For repeating events such as homework and projects, are there any similarities in office hour trends?

8. Which group of students (best/worst) attends office hours more during different times of the semester?

5.2 Student Performance and Previous Visualization Techniques

Most of the research aims at exploring patterns and correlation in student performance due to various time series factors. Gkatzia et al. discuss several such

factors that are time series data in the education domain [Gkatzia et al., 2013]. The factors that they collect from students and teachers in the form of time series data can be broadly classified into effort, frustration, and difficulty. They have classified effort into three subfactors - the numbers of hours the students have studied, the level of revision they have done and the number of lectures they attended. They have classified frustration into subfactors such as the understandability of the content, the presence of other deadlines and any pertinent health, personal issues, and their severity. They describe difficulty as the student's perception of difficulty with the content as the course progresses. Visualizations using visual elements to show the correlation among time series data can help researchers view new relationships between these time series factors.

Data processing and visualization are two critical tools used in educational data mining. Data processing manipulates and transforms raw data into useful information which can then be passed to visualization tools for effective representation. This form of processing includes format conversion, validation, sorting, summarization, aggregation, analysis, and reporting. Visualization, on the other hand, includes encoding this processed data in pictures and graphics thereby making it easier for users to understand it effectively.

Before diving into the existing visualization tools and techniques, we will discuss the following topic: the purpose of building this tool for our research, the data processing involved, and the representation techniques we intend to use for the processed data. The intuition behind building this tool is to view if there exists any correlation between help-seeking behavior in students and their grades.

The help-seeking behavior data and the student grade data are in time series format. Due to its massive nature, we will employ clustering techniques to reduce the raw data into essential and unique time series data points. This visualization is composed of an amalgamation of line-charts and shifted stacked bar charts to represent clustered time series clusters and the distribution of data within the clusters.

Researchers have spent a considerable amount of time understanding factors that impact student performance. This research is generally aimed at finding methods for improving student grades as well as for identifying reasons that lead to a decline in grades. In our thesis, we will be looking at the effects of help-seeking behavior, specifically office hours attendance, on student performance. Using this visualization, we would like to answer questions on how does attending office hours at a certain point in the semester for a specific duration affect a student's performance and overall grade.

Most of the research done in understanding student performance can be classified into two categories. The first category deals with understanding how collecting and analyzing student performance data help in improving the quality of education and the second category deals with understanding the impact of various factors on student performance.

Educational data mining (EDM) is an interdisciplinary research area that deals with helping researchers understand educational context data. Visualization plays a crucial role in helping researchers understand student performance using visual elements and graphics. Presentation of data in graphical format helps users

Figure 5.1: The main screen of eduViz. The top panel allows side-by-side exploration of grades based on Date, Assignment, and Subject. Information can be filtered, and distinct views can be chosen so that the user can compare grades as desired. The bottom panel allows grade assignment using the partition slider shown on the bottom left. The resulting grades are shown in the scatterplot (bottom left) and histogram (bottom right) [Friedler et al., 2008]

understand information quickly, pinpoint emerging trends, and identify relationships and patterns.

Some of the previous work in EDM also uses visualization to understand student grades and factors which affect it. The purpose of such visualization tools is also to identify scope for improvements in existing teaching methodologies. Friedler et al. propose a tool called eduViz that enables teachers to explore grade patterns to identify individual needs and promote fairer student assessment.

Teachers can use eduViz to see the trajectory of student scores and compare the scores relative to the entire class for specific course events [Friedler et al., 2008]. This tool also helps them gauge areas needing focus and improvement. Additionally, Friedler et al. use eduViz as an exploratory tool to provide a grade assignment interface for teachers. Teachers can use this to see the number of students assigned to each grade range using a scatter plot and histogram in a multi-coordinated view [Friedler et al., 2008]. Friedler et al. also discuss the effectiveness of the tool eduViz compared to existing visualization techniques such as Blackboard, Microsoft Excel, and WebCT. They further explain how eduViz solves some of the problems users generally face using Excel, Blackboard and WebCT [Friedler et al., 2008]. There are several shortcomings of eduViz that we intend to address through our visualization tool. One of the major shortcomings of eduViz is that it focusses on just one student at a time and thereby, cannot be simultaneously used by viewers to view the entire class grades at once. It also does not give an overview of the entire classroom so that the user can look at the overall trends in the performance. Since eduViz is a multiple coordinated view panel, it requires the users to coordinate two different panels to understand relations of causality and correlation.

Most of the visualization techniques employed in research are limited to existing and standard visualization techniques such as line charts, bar charts, scatter plots and pie charts. However, there is research that uses novel visualization techniques for the field of EDM. Kay et al. in their seminal work introduce several novel visualization techniques that help users understand the

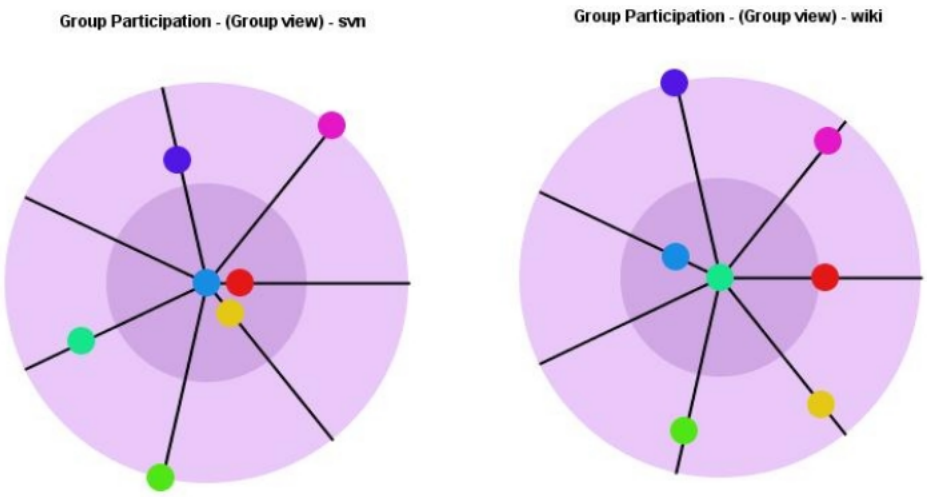


Figure 5.2: Activity Radar for representing individual contribution within a team for SVN and Wiki [Kay et al., 2006]

effect of group activity collected during a semester-long software development project course [Kay et al., 2006]. They use this visualization to answer five significant questions about team leadership, mutual performance monitoring, backup behavior, adaptability, and team orientation. Kay et al. have used an activity radar, as seen in Figure 5.2, to indicate the individual contribution of members within a team according to their contribution to SVN and Wiki Pages. A colored dot indicates each of the individuals within a group, and his or her contribution to the project is proportional to how close he or she lies to the center of the circle. The circle itself has two tints: light and dark tint. The separation of these tints indicates the average contribution of the members of the group. The dark tint indicates an above average level contribution whereas the light tint indicates a below average contribution. This kind of representation of individual components within a group is in line with one of the goals we have for representing

time series clusters, where we would like our visualization to represent the distribution of points within a cluster. However, we would also like to view additional characteristics such as median, interquartile ranges, minimum and maximum to be able to view the tightness of the cluster.

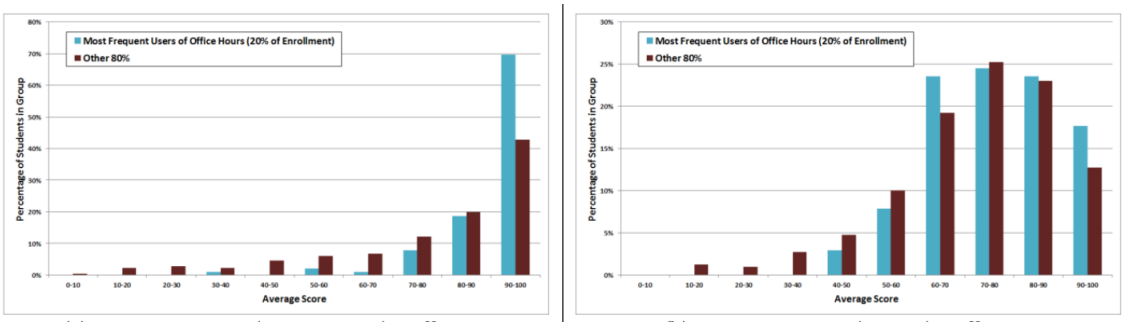
5.2.1 Effect of Help-Seeking Behavior Data on Student Grades

In this case study, we propose a visualization method to look at how help-seeking behavior, in particular attending office hours, impacts student grades. We analyze the office hours and performance time series data for the introductory course CMSC 201 during the term Spring 2017 and Fall 2017. We examine the data to understand if there exist any patterns or trends between these two distinct time series. We use this visualization method as an exploratory analysis tool to understand how events in the office hour data correlate with the performance of the students in the Spring semester.

There has been significant research on understanding student performance in the recent years. Education data mining (EDM) is an area of data mining that focusses on understanding education data to understand how the quality of education can be improved. EDM allows teachers, parents, and policymakers to improve the quality of education by understanding teaching techniques, course planning, resource allocations and student grades. It uses methods and tools from the broader field of data mining for detecting patterns in extensive collections of educational data. Pattern detection in such datasets would have been otherwise

impossible because of the enormous volume of data that exist in the educational domain [Heeren and Fagen, 2015].

5.2.2 Help-seeking behavior Background and Previous Work



(a): Data Structures (Assignments), Fall 2014

(b): Data Structures (Exams), Fall 2014

Figure 5.3: Comparison of frequent and infrequent attendees for office hours [Heeren and Fagen, 2015]

Nadler defines help-seeking behavior as an activity consisting of three components: a person in need of help, a source of help, and a specific need for help [Nadler, 2015]. In an educational setting, the person in need of help is the student, the source of the help is the teaching assistants and instructors, and the help sought could be assistance needed for solving assignments or concept related help. Some of the previous work in understanding the effectiveness of office hours deals with understanding how the most frequent users (twenty percent of enrollment) of office hours perform in comparison to the remainder of the class for programming assignments and exams. It uses data visualization as a method to explain its findings compellingly such as a bar chart matrix as seen in Figure 5.3 to provide a comparison between the frequent and infrequent users of office hours. They find

that students visiting office hours frequently perform better than their infrequent counterparts for programming assignments, but they perform the same in exams. This finding is explained by the fact that the office hours generally focus on the programming assignments more than on exams.[Heeren and Fagen, 2015] as seen in Figure 5.3.

5.3 Data Ingestion

The student grade data and office hour data for the introductory course CMSC 201 for term Spring 2017 and Fall 2017 is available in two separate CSV files. In another CSV file, there are various course events listed according to their date of occurrence and their weight in the overall grade.

The Spring 2017 student grade file contains the grades data for 274 students whereas the Fall 2017 student grade file contains the grades for 491 students collected over a span of three and a half months. This file contains the scores that students have scored for various course events in the semester such as Homeworks, Labs, Projects, Surveys and Exams. The time series data in the grades file is in the form of a irregularly spaced time series data.

The help-seeking files contains the records of students and their attendance at the office hour for various topics throughout the semester. This file contains fields such as student id, date attended, topic needed help with, name of the TA who assisted, notes, waiting time and duration of attendance.

We ingest the data from this file into the visualization framework with the

use of a D3 API called d3.csv() function which contains a path parameter to the CSV file. It transforms the entire CSV file into an array of objects which undergoes preprocessing to be available for creating visualization elements in the tool.

5.4 Data Preprocessing

The first step in preprocessing the student grade data is to transform the individual grades into percentage scores. We compute the new scores such that they consider the performance of all the past course events. The intuition behind this conversion is to understand the overall percentage score of a student as the semester progresses. As a result, the final score achieved due to this conversion is the same as the final score achieved by the student in the semester.

To obtain a student's performance in the course at a point in time, the entire data has to be changed from an irregularly spaced time series to a regularly spaced one. Doing this allows us to have better accuracy while using the K-means clustering algorithm to create the clusters. It also helps to represent the student performance on a day to day basis. This conversion of irregularly spaced time series to regularly spaced time series is the second step in preprocessing the data.

The office hour data is also preprocessed to group all the events based on dates and duration. This preprocessing is essential since most of the students have attended the office hours for different durations. Grouping them by a duration interval can help the users understand the effectiveness of attending different office hour intervals. It can help users understand if attending office hours for longer

duration has any effects on a student's performance compared to attending office hours for shorter durations. Also, the office hour attendance data is not continuous and regularly spaced for the entire semester. Hence, we need to create additional data points for converting it into regularly spaced time series data. We need to convert it into regular spaced data so that the student performance which is continuous and regularly spaced can be correlated with this data.

5.5 Deciding the number of clusters

In this section, we discuss two methods used to determine the number of clusters for the k-means algorithm.

The first method employed is the elbow method [Kodinariya and Makwana, 2013]. We plot the sum of squared errors vs. the number of clusters for five different distance metrics :

1. Euclidean distance
2. DTW distance
3. Manhattan distance
4. Minkowski distance
5. Chebyshev distance

We can see the plots for each distance metric in Figure 5.4. As we can see in the elbow graphs in these figures, the elbow joints for all the distance metrics

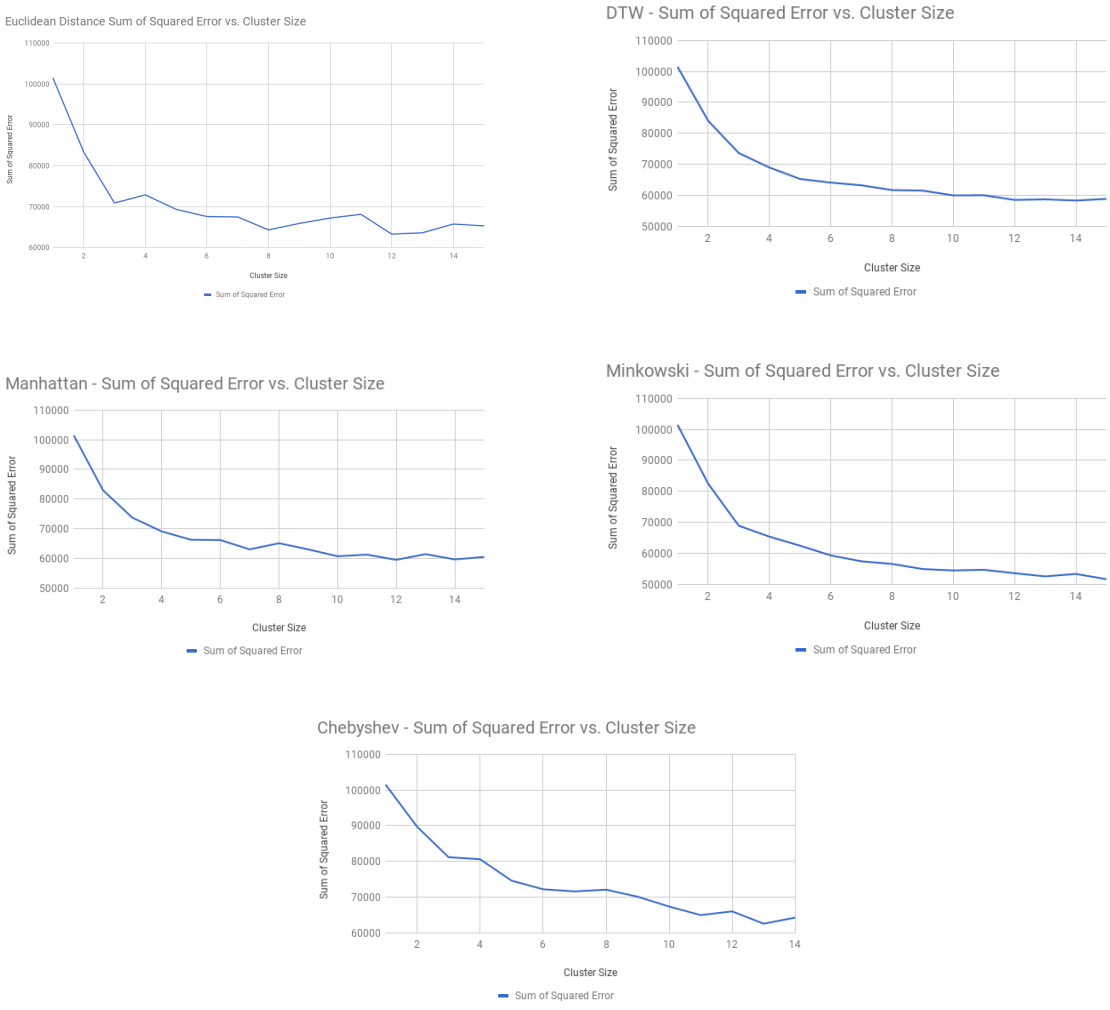


Figure 5.4: Sum of squared errors vs cluster size for different distance metrics - Euclidean, DTW, Manhattan, Minkowski, and Chebyshev

except Euclidean distance are visible. For Euclidean distance, the elbow joint is present at k=3. However, this value of k may not be a good choice if we would like to understand the effects of office hours on student grades since there will be only so few student grade clusters for us to base our results. If the number of clusters is too small, then the findings made using this tool will be very general. Another disadvantage of choosing so few clusters is that the smaller the choice of clusters more is the abstraction of the unique data points. If this distinct unique points are abstracted, then the user can no longer understand interesting aspects such as anomalies or patterns in the data.

We need to find a value which allows us to identify many distinct patterns in the data. Choosing a higher value for the number of clusters is still not a disadvantage since the sum of square error decreases with the increase in clusters. However, this value has to be small enough not to make the visualization cluttered. This can be decided visually by choosing different values of clusters and viewing the cluster distribution. Depending on the visibility of centroids and their distribution, a suitable value for k can be chosen.

Let us look at the effect of increasing the clusters in viewing the student grades using the DTW as the distance metric in Figure 5.5. We see that with the increase in the number of clusters, more distinct patterns can be viewed. However, in the case of k=20, we can see that the overall number of visual elements start to become cluttered. In the visualization for k=20, we can see that many of the clusters start overlapping on each other thereby making it difficult to observe their trajectory through time.

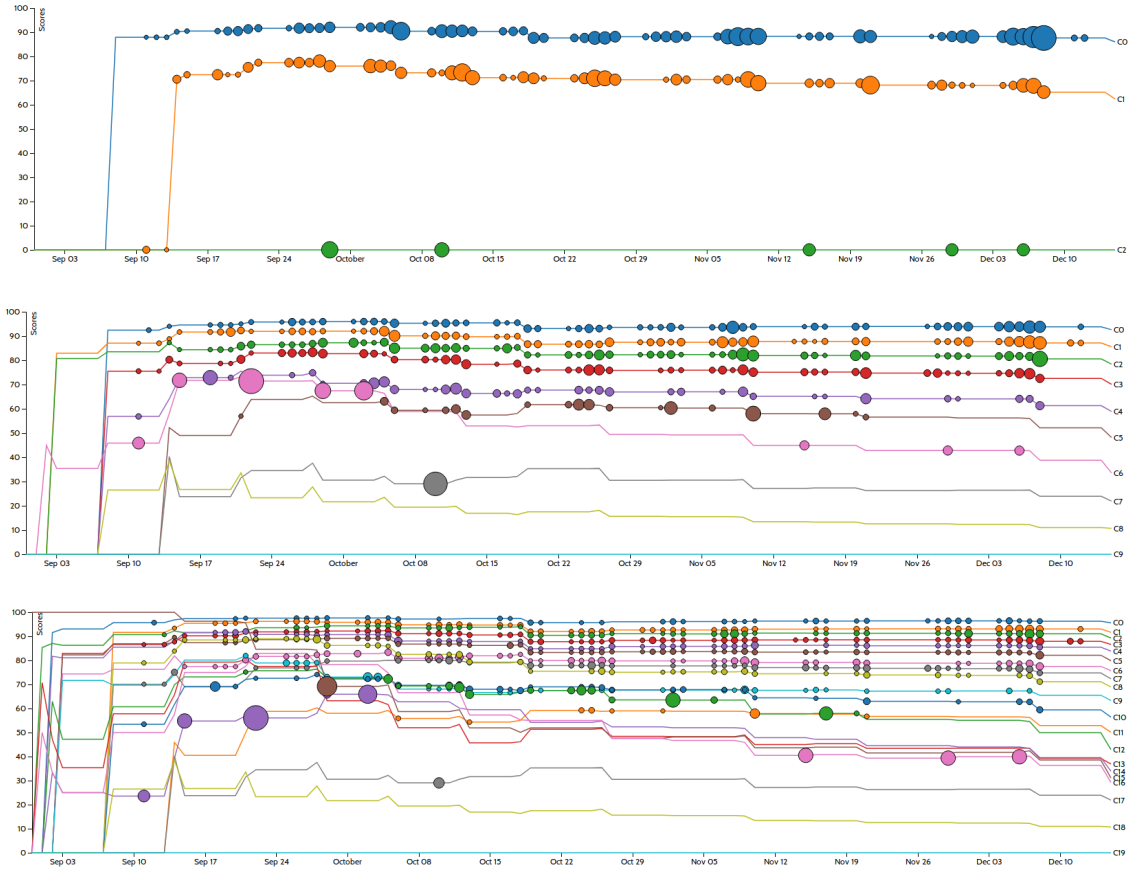


Figure 5.5: Student grade clusters for different values of k = 3, 10 and 20 respectively with DTW distance

The number of clusters chosen for this case study was ten clusters. The reason we chose ten clusters was that the minimum square error was sufficiently less for ten clusters. Also, the ten clusters can adequately represent the performance of the class for course CMSC 201 during the term Spring 2017 and Fall 2017 respectively without abstracting many details.

5.6 Shape Based Clustering of Time Series Data

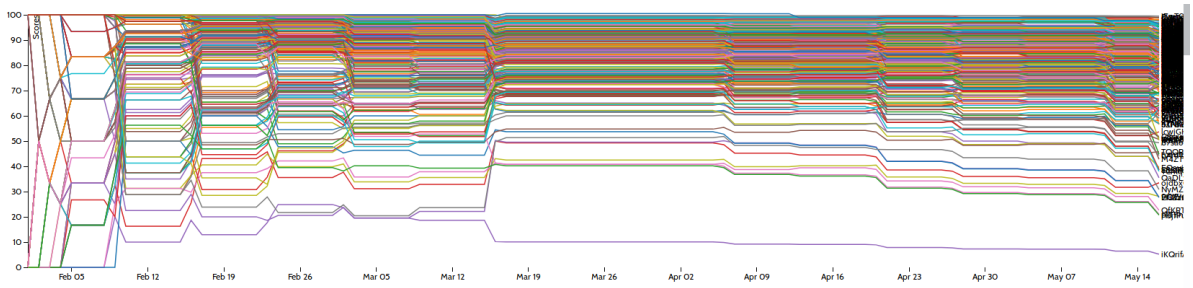


Figure 5.6: Student grade data for CMSC 201 Spring 2017 to unclustered format

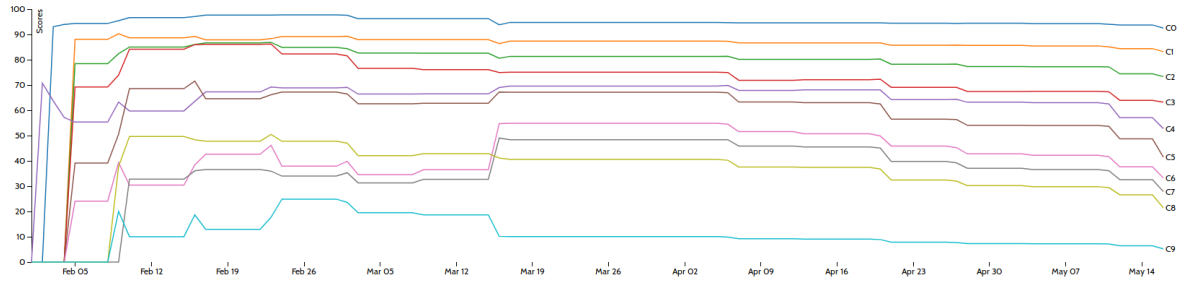


Figure 5.7: Student grade data for CMSC 201 Spring 2017 clustered based on shape

Figure 5.6 and Figure 5.8 represents the student grade data for the Spring 2017 and Fall 2017 respectively in its unclustered format. Here the x-axis represents the time, and the y-axis represents the score of the student as a percentage. A single colored line in this visualization represents a student performance through the

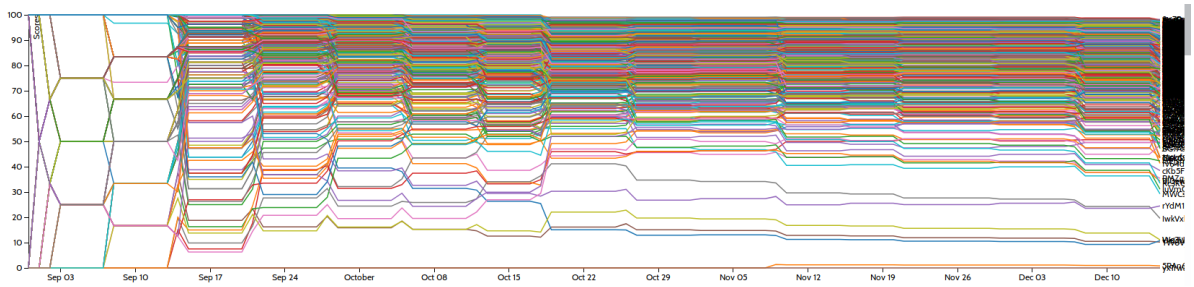


Figure 5.8: Student grade data for CMSC 201 Fall 2017 to unclustered format

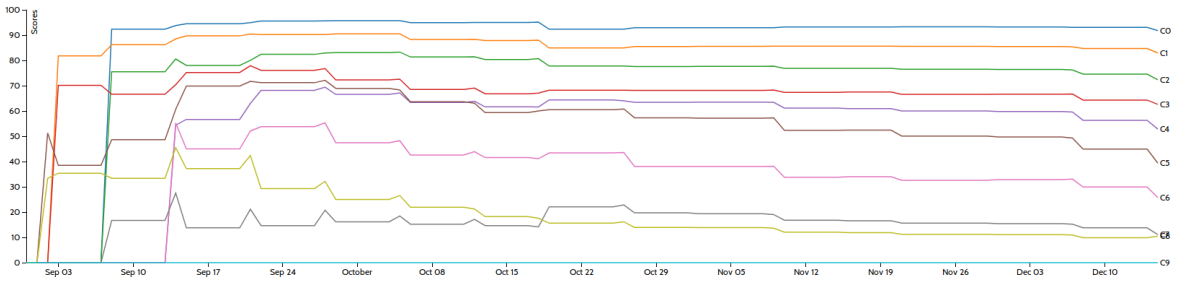


Figure 5.9: Student grade data for CMSC 201 Fall 2017 clustered based on shape

semester. This visualization fails to give any intuition on the trends and patterns in the scores of the students. It is highly cluttered and is completely unusable as an exploratory tool. It fails to answer the visualization goals we began with while developing this visualization method. Despite its cluttered format, we can observe that there are many lines which have similar trajectories through time. We can cluster these similar trajectory lines by clustering this time series data by shape.

The clustering algorithm used for this case study was a standard K-means algorithm with ten clusters for Spring 2017 in Figure 5.7 and hierarchical clustering for Fall 2017 in Figure 5.9. We chose two different clustering techniques for the Spring and Fall 2017 time series data so that we could see the difference in clusters created by these two different clustering algorithms. We will further see the distribution of values within these clusters in Section 5.10 Statistical

Attributes of Time series data.

5.7 Behavior-based clustering of Time Series Data

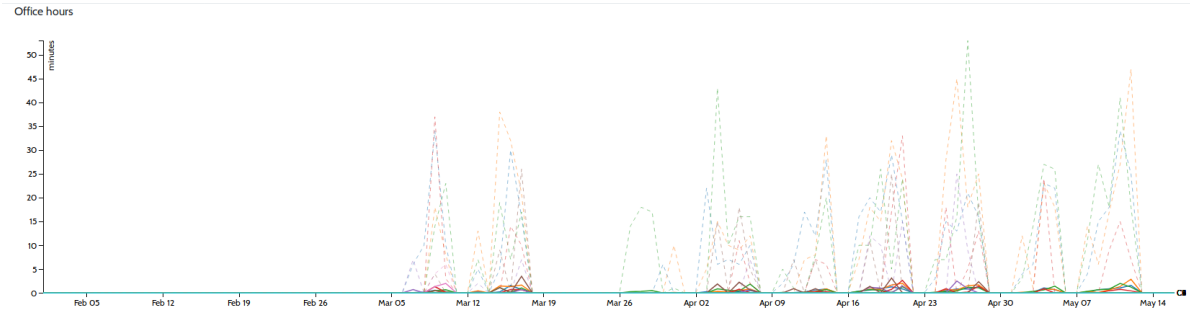


Figure 5.10: Clustered view of office hour data for Spring 2017 when grouped by behavior

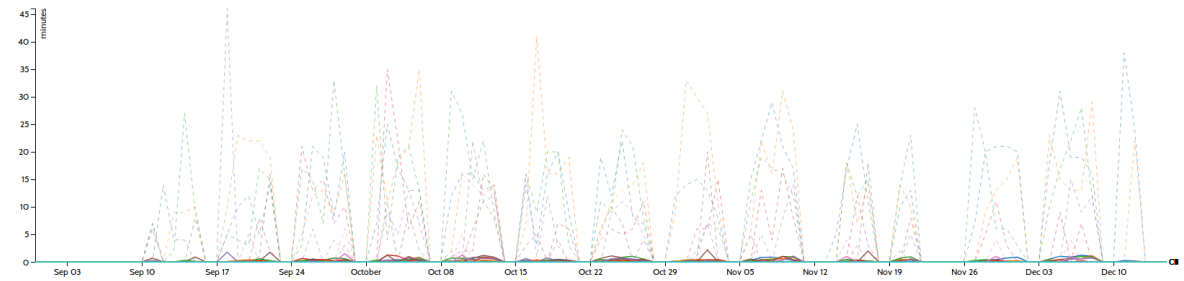


Figure 5.11: Clustered view of office hour data for Fall 2017 when grouped by behavior

Clustering the student grade data by shape has given us distinct clusters such that these clusters effectively encompass the performance of the entire class. However, the office hour data for these students are still unclustered.

To be able to correlate the student grade data with their corresponding office hour data, we need to cluster the office hour data by the clusters created by grouping similar shaped student grades together. Therefore, clustering the office hour data based on their grade clusters reduces the overall visual elements in the visualization

as seen in Figure 5.10 and 5.11 for the Spring 2017 and Fall 2017 semesters. In this figure, the x-axis represents time, and the y-axis represents the time spent by a student for office hours in minutes.

The Figure 5.11 and 5.10 contains three details about the office hour behavior of a cluster. The dashed lines represents the minimum and maximum time attended by a cluster and the solid line represents the average time attended by a cluster.

5.8 Statistical Attributes of Time Series Data

One of the key reasons for creating an entire novel visualization technique to represent time series clusters was to assess the effectiveness of the clustering algorithm. We can view the distribution within the clusters to understand if the clustering algorithm was valid or not. In this case study, we replace the original 274 Spring 2017 data points and 491 Fall 2017 with ten clusters. Since the data points within the cluster have been highly abstracted, the unique and distinctive features of these data points have been missed out on in this clustered visualization. To get a peek into how the data within a cluster is distributed, we have supplemented the original student grade data with the distribution of data within a cluster in the form of time series data. This representation technique uses elements of a box plot and overlays them on the time series line chart. This form of statistical representation allows the user to look at the performance of the students.

The original line series representing the clusters characterizes the geometric mean of all the data points within the cluster. We would like the ability to view

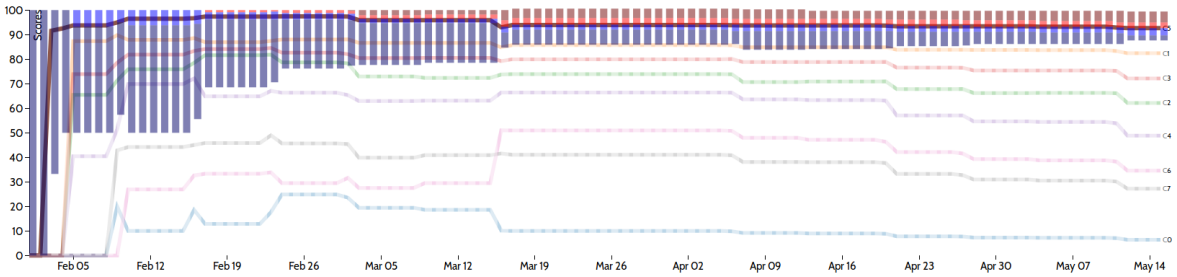


Figure 5.12: Visualizing the distribution within student grade clusters for Spring 2017

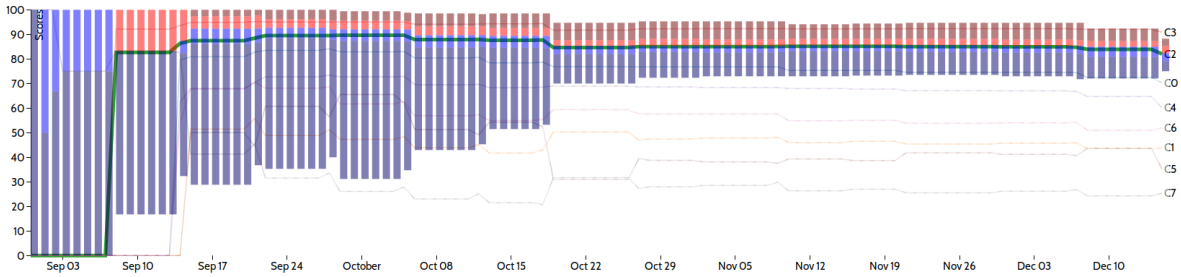


Figure 5.13: Visualizing the distribution within student grade clusters for Fall 2017

additional statistical characteristics for a cluster. As we can see in Figure 5.12 for Spring 2017 and Figure 5.13 for Fall 2017, we have represented the distribution of the cluster at a particular time with a stacked bar chart. This bar chart functions like a box plot which provides us with insights on the maximum, minimum, median, upper quartiles, lower quartiles, and outliers. These overlaid stacked bar charts provide additional answers to questions such as what the best scores within the cluster were, the worst scores, median scores and understand outliers.

In the Figure 5.12 for Spring 2017 and Figure 5.13 for Fall 2017 clusters, we can see the data within the highlighted cluster is distributed over time. We can see that the data begins with high variance initially, but later the variability decreases and the cluster closely follows the median values of the distribution.

5.9 Results

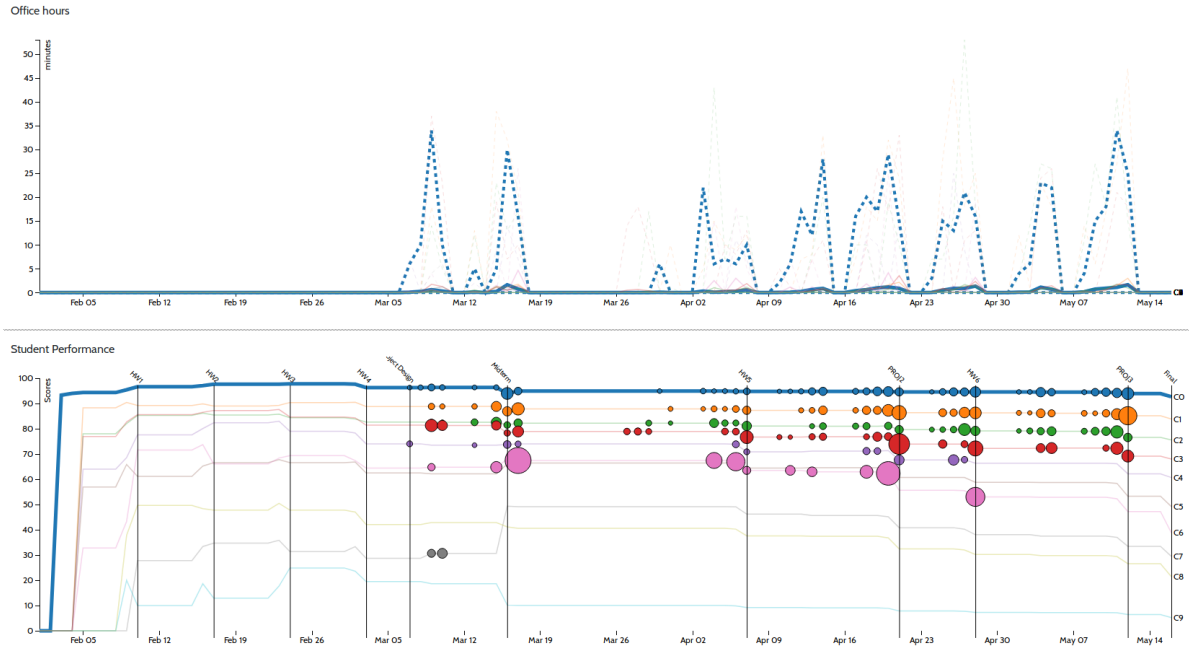


Figure 5.14: Visualizing the office hour and student grade time series data of the best performing cluster for Spring 2017

We have summarized the overall performance of the class by these ten clusters in Figure 5.7 for Spring 2017 and 5.9 for Fall 2017. The grade performance of the clusters can be seen in the lower half of the visualization with the corresponding office hour data represented on the upper half.

For the Spring 2017 semester, the best performing group is represented in blue in Figure 5.14, and the worst performing group is represented in teal in Figure 5.15. Looking at all the different groups of students, we can say that the better performing group attended office hours more than the worse performing group. In the case of the best performing group, they attended office hours more frequently and for longer durations as seen in Figure 5.14 compared to the worst performing group which did

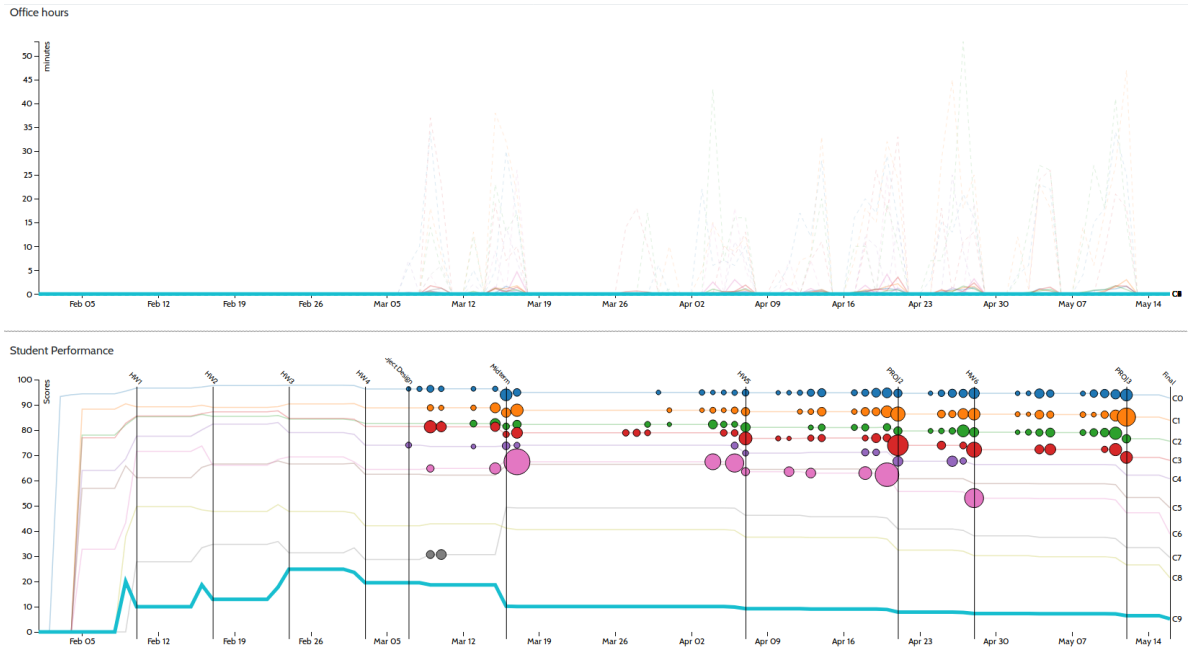


Figure 5.15: Visualizing the office hour and student grade time series data of the worst performing cluster for Spring 2017

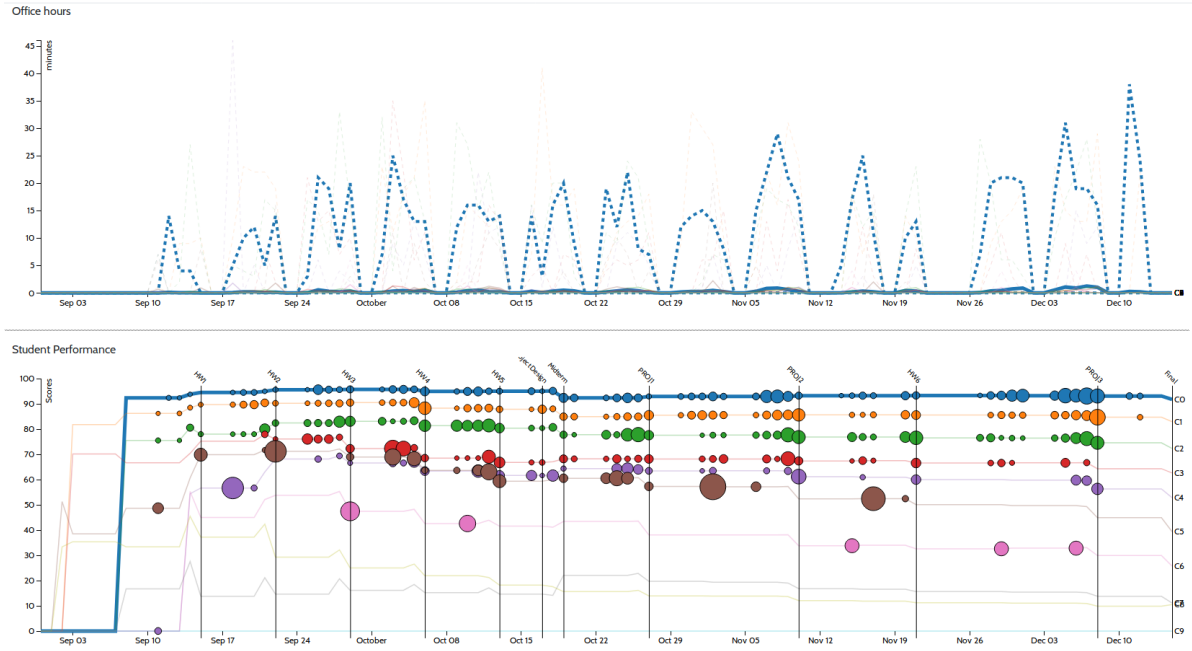


Figure 5.16: Visualizing one of the best performing clusters in Fall 2017 dataset

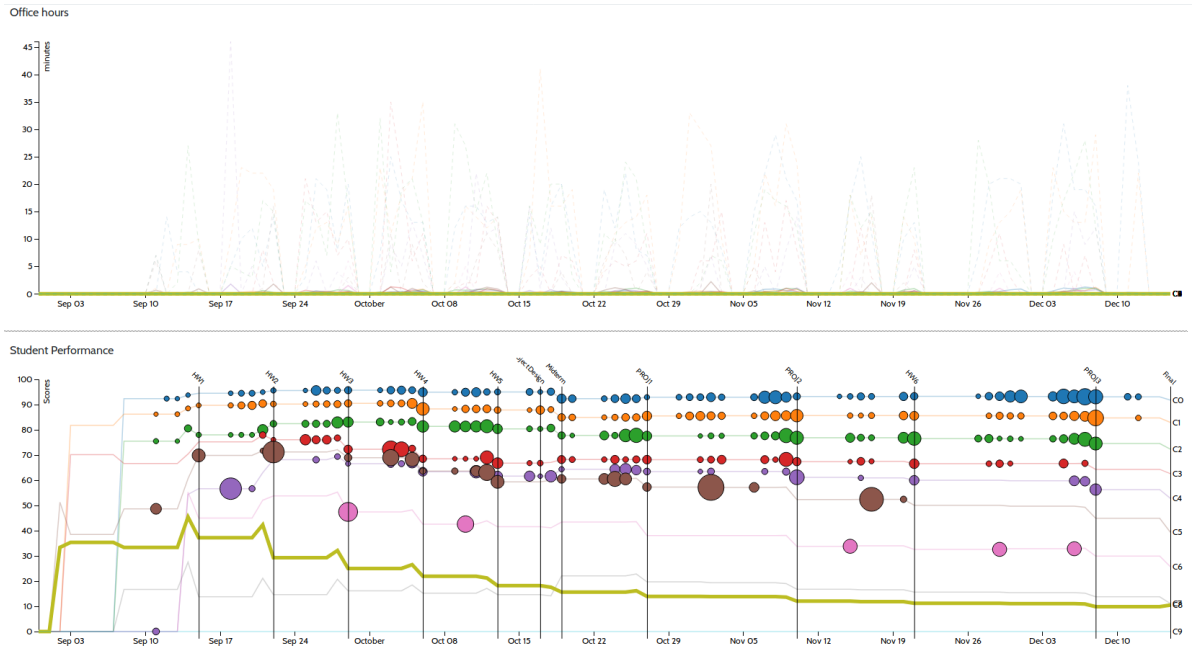


Figure 5.17: Visualizing one of the worst performing clusters in Fall 2017 dataset

not attend office hour at all in Figure 5.15.

Similarly for the Fall 2017 data, we can see that the best performing student clusters in blue attend office hours more frequently and reasonably regularly as seen in Figure 5.16. On the other hand, the lowest performing group in yellow for the Fall 2017 semester do not attend any office hours as can be seen in 5.17.

In both the semesters, we can see that as the performance of the students decreases, the office hour attendance frequency and duration also decreases. This visualization helps to answer the same questions that earlier required multiple coordinated views. It can be observed that the better-performing groups of student attend office hours more in comparison to the worse performing groups. The intensity or the duration of attendance of office hours in the better-performing groups are similar and also heavily localized around specific dates.

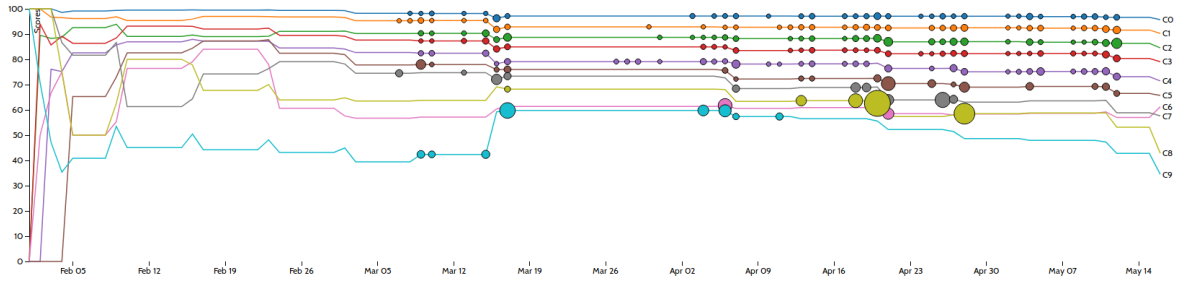


Figure 5.18: Visualizing the office hour data integrated with the grade clustered data for only those students who attended office hours for Spring 2017

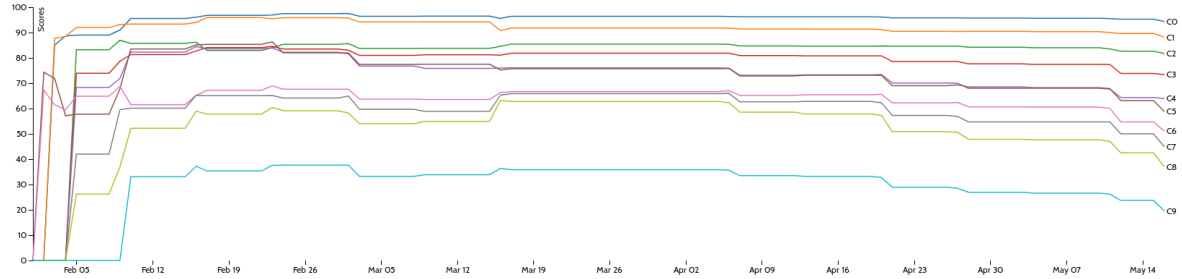


Figure 5.19: Visualizing the office hour data integrated with the grade clustered data for only those students who did not attend office hours for Spring 2017

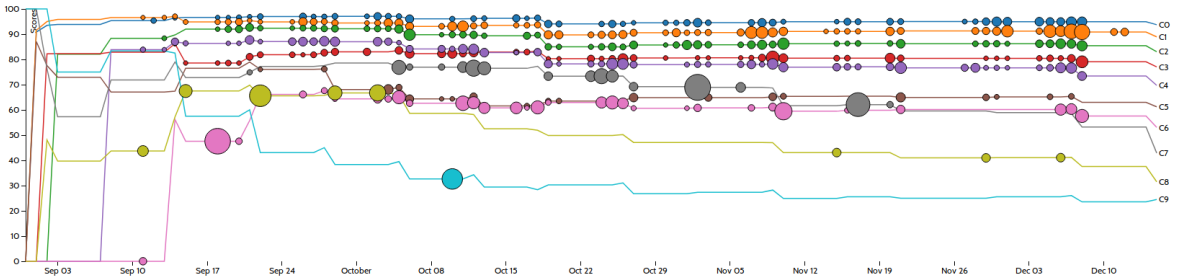


Figure 5.20: Visualizing the office hour data integrated with the grade clustered data for only those students who did attend office hours for Fall 2017

If we compare only students who sought help in the form of office hours, we can view their integrated grade-office hour visualization in Figure 5.18 for the Spring 2017 semester and in Figure 5.20 for Fall 2017 semester. In these figure, we can

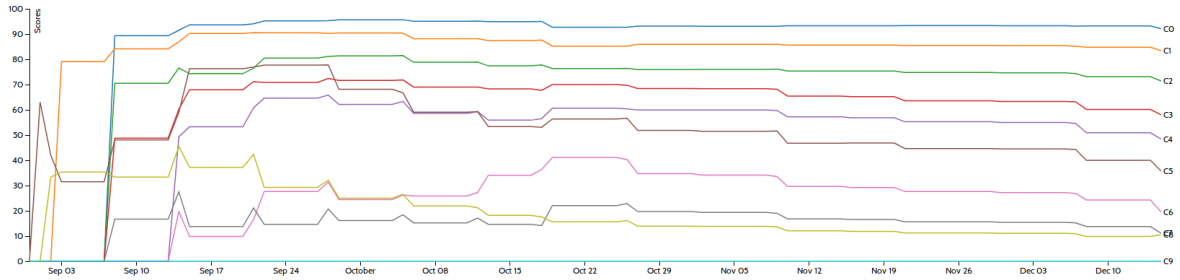


Figure 5.21: Visualizing the office hour data integrated with the grade clustered data for only those students who did not attend office hours for Fall 2017

observe that, of the students who attend office hours, the cluster of students who had performed the worst attended minimal office hours. And student clusters who performed well attended office hours much more frequently and usually well before the due dates.

However, we cannot say the same for all the students who did not attend office hours. We can see the office-grade visualization for students who did not attend any office hours for Spring 2017 in Figure 5.19 and for Fall 2017 in Figure 5.21. As expected, since these groups of students did not attend any office hours, there are no circular overlays or bubbles for office hour intensity over the grade lines. In this visualization, we can see that there do exist students who, despite not seeking any help, performed at par with students who attended office hours. It would be interesting to obtain other factors that impact the grades of students and view them in an integrated format to see the causal relationship that exists between these factors.

In this tool, we also provide an additional panel on the left-hand side that gives the user additional information about their selection of cluster. The panel can

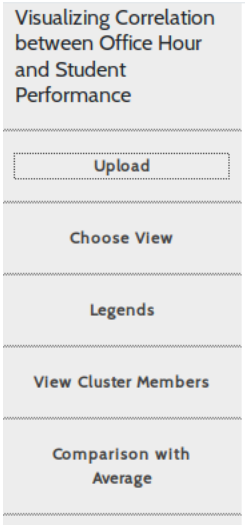


Figure 5.22: A panel that allows user to view details about clusters selected by user

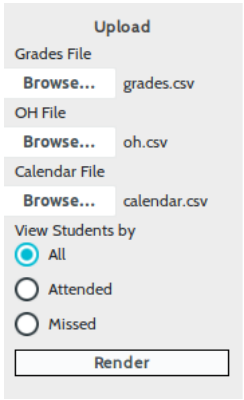


Figure 5.23: A sub panel that allows users to upload the grade, office hours and calendar files. It also allows the instructor to view all students, students who only attended office hours and students who did not attend any office hours

be seen in 5.22. This panel consists of four sub-panels.

The first sub-panel as seen in Figure 5.23 allows the users to upload the files required to render the visualization. The three required files are the grades file, the office hour attendance file and the calendar file. In addition, this section also allows the user to filter the student by their attendance type. There are three types of

attendance - all students, only students who attended office hours and only students who missed office hours. Clicking on the "Render" button allows the data analysis and processing to be proceeded followed by the rendition of the visualization.

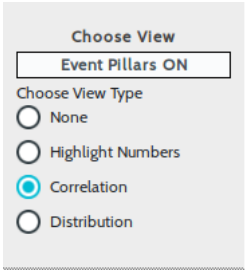


Figure 5.24: A sub panel that allows users to choose different modes to explore the data. The four supported modes are none, numbers mode, correlation mode and distribution mode

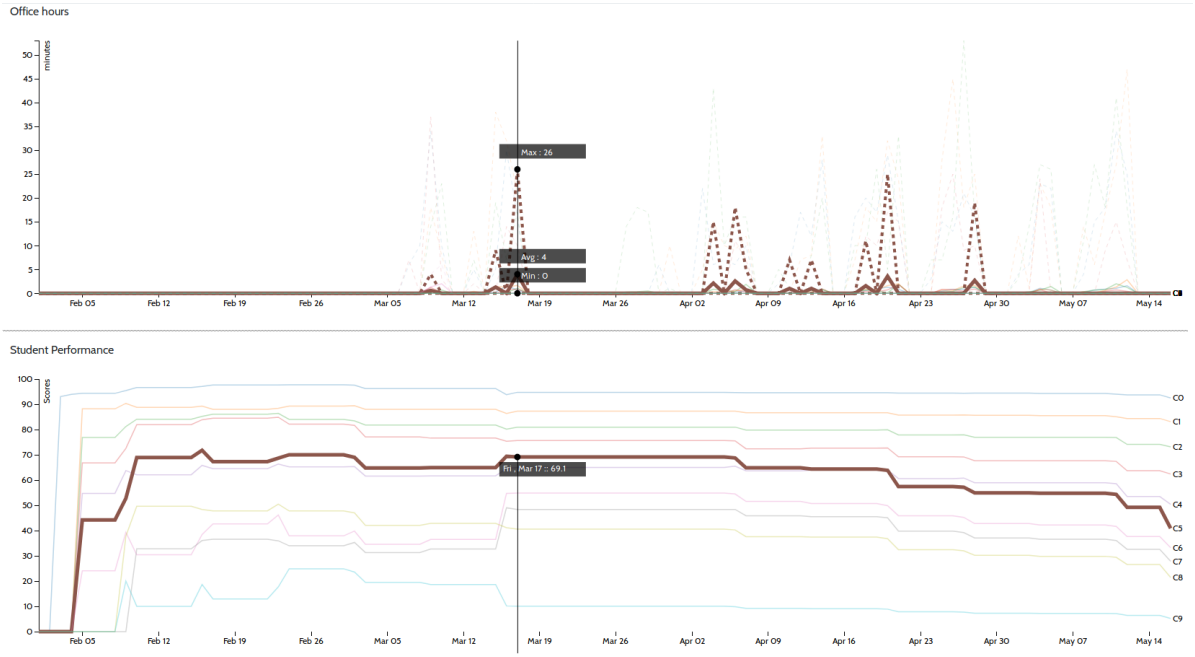


Figure 5.25: Number mode ON in the visualization panel

The second sub-panel as seen in Figure 5.24 allows the users to choose the type of view for the visualization. There are four types of view available for the

user. The first is to disable all the other views marked by "None". The second view is called the "Highlight Numbers" view which allows the user to view the data with actual values in the graph. This can be seen in detail in Figure 5.25. The third view is the "Correlation" view which creates the integrated line-bubble view as seen in Figure 5.9. The last view is "Distribution" View which helps the users understand the distribution within a cluster as seen in Figure 5.13.

Since this tool is built for exploratory purposes, we add another feature "Event Pillars" into this tool that gives the users some perspective on the events that occur during the semester. The user can enable/disable course event pillars that identify significant events that occur during the course. This feature can be seen in Figure 5.9 where the vertical lines in the visualization marked by the events occurring is represented.

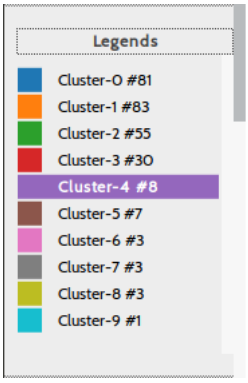


Figure 5.26: A sub panel that allows users to view details about the selected cluster

The third sub-panel as seen in Figure 5.26 allows the users to see the cluster name and cluster size along with its assigned color. The entry corresponding to the selected cluster gets highlighted to inform the user of their choice and bring focus to the cluster detail.

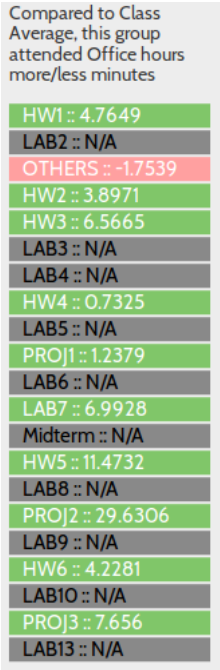


Figure 5.27: A sub panel that allows users to compare the office hours attendance of the selected cluster with the class average

The fourth sub-panel as seen in Figure 5.27 allows the user to perform a comparative analysis of the selected cluster with the class average. The rationale behind this pane is to allow users to compare the selected cluster with the other clusters. The elements in this panel indicate the event and a comparison of the number of minutes attended by this cluster with the class average. When the number of minutes attended by a cluster is more than the class average, it is highlighted in green with the difference following the event name measured in minutes. When the number of minutes attended by a cluster is less than the class average, the element is highlighted in red. If the selected cluster contains students who have not attended any office hours, then the event name is followed by N/A and is highlighted in grey.

In addition to the existing capabilities, a user can now click an office hour

Thu, 2-16-2017

Students attended an average of 1.556 minutes

To discuss the following items (Total entries - 11)

Username	Time Entered	Time Received Help	Time Finished seeking Help	Help Duration	Wait Duration	TA Name	Question	Notes	Help Category	Time Spent Personally
eRBMOkYazfCecm5	12:03:52	12:07:42	12:15:40	7	3	KaQwaXtCpSCr1V6	1 and 9	Review for the midterm, what are the control structures	Other	
43G5t1a7EVENe6X	12:31:37	13:01:29	13:04:18	2	29	KaQwaXtCpSCr1V6	how to line up the numbers	Help with spacing the print statement	HW/Project	
YSbNlt9NkMtPbXZ	12:43:08	13:05:29	13:19:15	13	22	qTffn66alc2gHRX	I have some questions about the review packet for exam 1	Needed help with reveiw guide	Other	
yd5Ad7VQunPkad1	13:33:30	14:41:47	15:12:11	30	68	jmcvqefUu5zYm7w	Constants / table greeting	formatting problems	HW/Project	
wV8npfS2R6j6QGB	14:20:44	15:04:14	15:12:18	8	43	jmcvqefUu5zYm7w	Are in-line comments needed inside functions?	formatting problems	Other	
7HInf1qFxMjtpS	14:43:48	15:22:25	15:43:05	20	38	7gJpDywcVTlu3yC	loop doesn't work as expected	Lab 7 nested loop problems	LAB	
zC9lVu4MErZgcEJ	15:33:08	16:09:28	16:28:51	19	36	xzHI6xMCrs4wF9k	I am getting an error I do not understand or know how to fix.	needed help getting numbers into list	HW/Project	
3kBtOiBlbCF7dQT	15:50:23	16:14:29	16:25:14	10	24	7gJpDywcVTlu3yC	How can I get to a specific character in a string in a list of strings?	Lab 7 help	LAB	
4SecKwg8rB8eeNO	15:59:29	16:23:48	16:29:20	5	24	xzHI6xMCrs4wF9k	Checking the final output and formatting	Needed help with casting int to string, and formatting help	HW/Project	
y3hyGrWBYFBAMtG	16:38:04	17:20:53	17:26:11	5	42	xzHI6xMCrs4wF9k	Just a few questions about how I should be formatting everything (proj1 is done otherwise.)	Needed help with formatting questions and outputs	HW/Project	
jOhY8AmI1XRnL2	17:20:22	17:37:55	17:45:21	7	17	xzHI6xMCrs4wF9k	My program is printing a string as a list	was going through and printing the [ ] and , as part of the upper case str	LAB	

Figure 5.28: Visualizing the details of the selected office hour circle

circle to view additional details about the event as can be seen in Figure 5.28. In this figure, the user selects an office circle highlighted by thick black boundaries. On clicking this circle, the user gets additional details displayed on the Office Hour Detail panel stating the date of visit, the average number of minutes spent in that session and the reasons for visiting the office hour.

The layout of this tool is such that it requires the users to view both the data points in parallel and assume that the events they look at each of the two different visualizations correspond to each other. This form of view is called a multiple coordinated view. This visualization method depends heavily on the ability of a user to coordinate and view the different visualizations to see any form of correlation.

Using this tool, we can make the following observations:

1. Students attend more office hours only before a major course event than any concept related event. In Figure 5.7, we can see that for all the groups for the Spring 2017 semester, the correlation events are present more on the days of the events compared to any time in between the events. Similarly, in Figure

5.9, we can see that for all the groups for the Fall 2017 semester, the correlation events are present more on the days of the events compared to any time in between the events.

2. Within students who attend office hours, the higher performing groups attend office hours frequently and for larger durations. In Figure 5.14 for Spring 2017 and 5.16 for Fall 2017, we can see that for the higher performing groups the attendance events are more frequent and the radius of the circle generally decreases as the performance transitions from top to bottom.

3. Students who perform poorly attend very little or no office hours and in less frequency as can be seen in Figure 5.15 for Spring 2017 and 5.17 for Fall 2017.

4. Students who performed poorly attended little or no office hours, but the opposite is not true. Some of the students who did not attend any office hours performed at par with the students who attended office hours as can be seen in Figure 5.19 for Spring 2017 and Figure 5.21 for Fall 2017.

5. There is a similarity in the number of office hours attended for some course events such as projects, homework, and labs indicating the same level of difficulty as can be seen in Figure 5.7 for Spring 2017 and Figure 5.9 for Fall 2017.

Chapter 6

Future Work

In this section, we will discuss two main areas of work - (1) overcoming limitations to specific clustering techniques and types of time series data, (2) handling cases where the number of clusters is very few or very high.

Several limitations exist in the current visualization when it comes to flexibility with the clustering technique and with the type of time series data that it can visualize. The current visualization technique works effectively for some hard clustering algorithms such as k-means and hierarchical clustering. However, for other algorithms such as self-organizing maps (SOM) or soft clustering algorithms such as fuzzy-c means, this visualization may not work. The current visualization works by assigning a data point to one cluster only. But in case of soft clustering algorithm, the data points can belong to more than one cluster. Future work for this limitation could include integrating different views for different clustering algorithms.

This visualization method also suffers from the limitations of the clustering algorithm used for clustering time series data. Clustering algorithms are ineffective in clustering irregularly spaced time series data and time series with different lengths. Even though the visualization can display the clusters produced as a result of clustering irregularly time series data, the clusters may not be a good

summarization of the underlying data. For the use cases for this visualization tool, we have converted the irregularly spaced time series to regularly spaced time series. Though, regularizing time series data has its limitations such as dilution of data due to unnecessary intermediate data points and loss of information stored in the variable durations between data points. Future work could consist of using data processing and clustering mechanisms used for analyzing irregularly spaced data.

The correlation visualization method in this research is limited to displaying correlation in events that occur in parallel in the same time domain. The visualization fails to use the superposition of the circles on the line charts for time series with different time domains. Also, this visualization technique enables viewing the effects of only one time series on another. In real-world applications, it is highly possible that the effect of causality in time series data is not just due to one single time series data but due to multiple different time series data, static values and even due to the past values of the original time series data. The current visualization needs to be extended to be able to show such relationships.

Furthermore, the visualization of the clusters is highly dependent on the choice for the number of clusters. For a poor choice of the number of clusters, the visualization may not summarize the data adequately. As a result, the clusters produced may not be tight. In case we choose an exceedingly high number of clusters, our visualization will become cluttered, which fails the purpose of clustering the original dataset. In future work, we could probably integrate visualizations of elbow, silhouette and gap statistic methods that complement the visualization by recommending good values for the number of clusters in the

k-means algorithm. It could also integrate a hierarchical dendrogram for hierarchical clustering where the users can cut the dendrogram at a particular level and view the number of clusters corresponding to this choice. We could also integrate clustering approaches that avoid choosing the number of clusters altogether such as DBSCAN [Liao, 2005] and YADING [Ding et al., 2015].

In the case of the k-means algorithm, we can also run into the issue of choosing bad initial centroids. As a result, the visualization may end up having clusters which do not contain any data from the data set and with random values along the time domain. The presence of such empty clusters may create unnecessary clutter in the visualization. Future works could include using algorithms such as k-means++ which gives a good choice of initial seeds for the k-means algorithm. The only drawback to implementing this algorithm would be that finding good initial seeds may take additional time.

Lastly, this visualization only answers questions such as if this clustering technique was effective or if the clusters produced were tight, but it cannot answer questions of what is a suitable clustering algorithm for this data. Also, it does not suggest a good choice for the number of clusters for a dataset.

Chapter 7

Conclusion

In this research, we propose a visualization method that allows users to view the clusters produced as a result of different clustering techniques. There are already visualization methods to display k-means clustering and hierarchical clustering on static data as seen in Figure 7.1 but the previous work in observing time series clusters are highly application specific. Through this research, we have suggested a technique to visualize a time series cluster for any general time series data. Also, through this research, we also propose a novel method of representing the correlation between simultaneously occurring time series data.

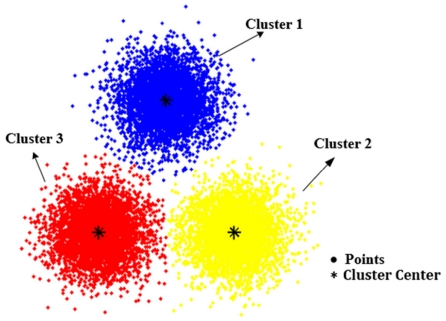


Figure 7.1: Concept diagram of k-means clustering [Zhang et al., 2017]

To show the usefulness of this technique, we discuss two applications for this visualization technique - with weather data and student performance data. Through the use of the correlation, we see the inverse correlation between temperature and pressure time series data for 18 cities. Using the clustering and correlation method,

we view the relationship between office hour attendance and student performance data for the introductory course CMSC 201 over a span of two semesters.

The introduction of this visualization method allows users to view the effectiveness of different clustering techniques and the choice for the number of clusters. This technique enables users to understand whether a clustering technique applied on a dataset produces tight clusters. It gives users an insight into the distribution of the data within a cluster. Furthermore, the ability to view correlation among time series dataset enables users to look at relationships of causality or periodicity easily.

However, the clustering and correlation technique still has several shortcomings that we have discussed in the future work. The visualization is developed only for representing the result of clustering uniformly spaced time series data. Time series data for several applications are sparse and irregularly sampled time series cannot be clustered effectively to be able to be represented visually.

While discussing the use cases for this visualization, we have assumed that the number of clusters chosen for representing the entire dataset is sufficient to summarize the data effectively. We also assume that the number of clusters that effectively summarizes the data does not clutter the visualization. In some cases, the visualization used to represent the clusters is only as good as the clustering technique used.

We developed a straightforward approach to visualize time series clusters created as a result of different clustering algorithms and different distance metrics. The visualization aims at introducing a human element in verifying the results of

the clustering algorithm, specifically, for series data. Furthermore, the choice for the number of clusters for both of the applications was tailored to provide a balance between enough detail about the dataset and small enough not to clutter the visualization.

To demonstrate the effectiveness of the approach, we applied the techniques to two different applications: weather and student performance - office hour data. Using this approach we were able to summarize the temperature data for 18 cities into five clusters and explain the correlation of these clusters with the humidity data cluster. For the student performance - office hour approach, we were able to view the general trends in student performance for CMSC 201. We also observed how office hours attendance might contribute towards the performance of these student clusters. We create a highly functional panel that in addition to the visualization technique, helps the user focus on particular clusters, compare clusters and understand the correlation patterns with the office hour data with ease. The visualization tool consisting of the novel visualization technique and panel was built to answer the visualization objectives. This visualization tool can also help viewers view any form of correlation and clustering in a uniform spaced time series data.

Overall, this research has provided a novel way to visualize time series clusters and correlation over time using a superposition of line charts with box plots or with circles in a single visualization.

Bibliography

[Abbas,] Abbas, O. A. Comparisons between data clustering algorithms. *International Arab Journal of Information Technology (IAJIT)*, pages 320–325.

[Aggarwal et al., 2001] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

[Aghabozorgi et al., 2015] Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, 53:16–38.

[Anderson et al., 1984] Anderson, R., Grenfell, B., and May, R. (1984). Oscillatory fluctuations in the incidence of infectious disease and the impact of vaccination: time series analysis. *Epidemiology & Infection*, 93(3):587–608.

[Arthur and Vassilvitskii, 2007] Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

[Bostock and Heer, 2009] Bostock, M. and Heer, J. (2009). Protovis: A graphical toolkit for visualization. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, pages 1121–8.

[Chalder et al., 2003] Chalder, M., Sharp, D., Moore, L., and Salisbury, C. (2003). Impact of NHS walk-in centres on the workload of other local healthcare providers: time series analysis. *Bmj*, 326(7388):532.

[Chen et al., 2005] Chen, B., Tai, P. C., Harrison, R., and Pan, Y. (2005). Novel hybrid hierarchical-k-means clustering method (hk-means) for microarray analysis. In *Computational Systems Bioinformatics Conference, 2005. Workshops and Poster Abstracts. IEEE*, pages 105–108. IEEE.

[Das et al., 1998] Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998). Rule discovery from time series. In *KDD*, volume 98, pages 16–22.

[de Amorim and Mirkin, 2014] de Amorim, R. C. and Mirkin, B. (2014). *Selecting the Minkowski Exponent for Intelligent K-Means with Feature Weighting*, pages 103–117. Springer New York, New York, NY.

[Ding et al., 2015] Ding, R., Wang, Q., Dang, Y., Fu, Q., Zhang, H., and Zhang, D. (2015). Yading: fast clustering of large-scale time series data. *Proceedings of the VLDB Endowment*, 8(5):473–484.

[Doganis et al., 2006] Doganis, P., Alexandridis, A., Patrinos, P., and Sarimveis, H. (2006). Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, 75(2):196–204.

[Dominici et al., 2002] Dominici, F., McDermott, A., Zeger, S. L., and Samet, J. M. (2002). On the use of generalized additive models in time-series studies of air pollution and health. *American journal of epidemiology*, 156(3):193–203.

[Ernst et al., 2005] Ernst, J., Nau, G. J., and Bar-Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, 21(suppl_1):i159–i168.

[Friedler et al., 2008] Friedler, S. A., Tan, Y. L., Peer, N. J., and Shneiderman, B. (2008). Enabling teachers to explore grade patterns to identify individual needs and promote fairer student assessment. *Computers & Education*, 51(4):1467–1485.

[Gkatzia et al., 2013] Gkatzia, D., Hastie, H., Janarthanam, S., and Lemon, O. (2013). Generating student feedback from time-series data using reinforcement learning. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 115–124.

[Goutte et al., 1999] Goutte, C., Toft, P., Rostrup, E., Nielsen, F. Å., and Hansen, L. K. (1999). On clustering fmri time series. *NeuroImage*, 9(3):298–310.

[Gutierrez and Wiesinger-Widi, 2016] Gutierrez, N. and Wiesinger-Widi, M. (2016). Augury: A time series based application for the analysis and forecasting of system and network performance metrics. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2016 18th International Symposium on*, pages 351–358. IEEE.

[Hamilton, 1989] Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384.

[Hao et al., 2010] Hao, M. C., Dayal, U., and Lyons, M. (2010). System and method for creating a value-based stacked bar chart. US Patent 7,779,344.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[Heeren and Fagen, 2015] Heeren, C. and Fagen, W. (2015). Quantitative correlation between student use of office hours and course. *122nd ASEE Annual Conference and Exposition*, pages 26.1296.1 – 26.1296.9.

[Himberg et al., 2004] Himberg, J., Hyvärinen, A., and Esposito, F. (2004). Validating the independent components of neuroimaging time series via clustering and visualization. *Neuroimage*, 22(3):1214–1222.

[Jain, 2010] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

[Jönsson and Eklundh, 2004] Jönsson, P. and Eklundh, L. (2004). Timesata program for analyzing time-series of satellite sensor data. *Computers & Geosciences*, 30(8):833–845.

[Kay et al., 2006] Kay, J., Maisonneuve, N., Yacef, K., and Reimann, P. (2006). The big five and visualisations of team work activity. In *International Conference on Intelligent Tutoring Systems*, pages 197–206. Springer.

[Keogh and Lin, 2005] Keogh, E. and Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8(2):154–177.

[Kodinariya and Makwana, 2013] Kodinariya, T. M. and Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95.

[Kumar et al., 2005] Kumar, N., Lolla, V. N., Keogh, E., Lonardi, S., Ratanamahatana, C. A., and Wei, L. (2005). Time-series bitmaps: a practical visualization tool for working with large time series databases. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 531–535. SIAM.

[Le Maire et al., 2011] Le Maire, G., Marsden, C., Verhoef, W., Ponzoni, F. J., Seen, D. L., Bégué, A., Stape, J.-L., and Nouvellon, Y. (2011). Leaf area index estimation with modis reflectance time series and model inversion during full rotations of eucalyptus plantations. *Remote Sensing of Environment*, 115(2):586–599.

[LeBaron et al., 1999] LeBaron, B., Arthur, W. B., and Palmer, R. (1999). Time series properties of an artificial stock market. *Journal of Economic Dynamics and control*, 23(9-10):1487–1516.

[Liao, 2005] Liao, T. W. (2005). Clustering of time series dataa survey. *Pattern recognition*, 38(11):1857–1874.

[Lu et al., 2016] Lu, B., Charlton, M., Brunsdon, C., and Harris, P. (2016). The minkowski approach for choosing the distance metric in geographically weighted regression. *International Journal of Geographical Information Science*, 30(2):351–368.

[Mario et al., 2017] Mario, R. H. T., David, B., and Dan, O. (2017). Historical hourly weather data 2012-2017. `"https://www.kaggle.com/selfishgene/historical-hourly-weather-data"`. [Online; accessed 20-May-2018].

[Martín et al., 2010] Martín, L., Zarzalejo, L. F., Polo, J., Navarro, A., Marchante, R., and Cony, M. (2010). Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. *Solar Energy*, 84(10):1772–1781.

[Nadler, 2015] Nadler, A. (2015). The other side of helping: Seeking and receiving help. *Oxford library of psychology. The Oxford handbook of prosocial behavior*, pages 307–328.

[Neumann et al., 2010] Neumann, B., Walter, T., Hériché, J.-K., Bulkescher, J., Erfle, H., Conrad, C., Rogers, P., Poser, I., Held, M., Liebel, U., et al. (2010). Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature*, 464(7289):721.

[Nightingale, 1858] Nightingale, F. (1858). *Notes on matters affecting the health, efficiency, and hospital administration of the British army: founded chiefly on the experience of the late war*. Harrison and Sons, St. Martin's Lane, WC.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[Saas et al., 2016] Saas, A., Guitart, A., and Periánez, A. (2016). Discovering playing patterns: Time series clustering of free-to-play game data. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, pages 1–8. IEEE.

[Sarstedt and Mooi, 2014] Sarstedt, M. and Mooi, E. (2014). Cluster analysis. In *A concise guide to market research*, pages 273–324. Springer.

[Singh et al., 2013] Singh, A., Yadav, A., and Rana, A. (2013). K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10):13–17.

[Ultsch and Mörchen, 2005] Ultsch, A. and Mörchen, F. (2005). *ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM*, volume 46. Data Bionics Research Group, University of Marburg, D-35032 Marburg, Germany.

[Van de Geer, 1995] Van de Geer, J. P. (1995). *Some aspects of Minkowski distance*. Department of Data Theory, Leiden University.

[van Wijk and van Selow, 1999] van Wijk, J. J. and van Selow, E. R. (1999). Cluster and calendar based visualization of time series data. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 4–9. IEEE.

[Weber et al., 2001] Weber, M., Alexa, M., and Müller, W. (2001). Visualizing time-series on spirals. In *Proceedings of the IEEE Symposium on Information*

Visualization 2001 (INFOVIS'01), INFOVIS '01, pages 7–14, Washington, DC, USA. IEEE Computer Society.

[Wilks, 2011] Wilks, D. S. (2011). Cluster analysis. In *International geophysics*, volume 100, pages 603–616. Elsevier.

[Yamada and Nakano, 1992] Yamada, T. and Nakano, R. (1992). A genetic algorithm applicable to large-scale job-shop problems. In *PPSN*, volume 2, pages 281–290.

[Zhang et al., 2017] Zhang, J., Chen, W., Gao, M., and Shen, G. (2017). K-means-clustering-based fiber nonlinearity equalization techniques for 64-qam coherent optical communication system. *Optics express*, 25(22):27570–27580.

[Zolhavarieh et al., 2014] Zolhavarieh, S., Aghabozorgi, S., and Teh, Y. W. (2014). A review of subsequence time series clustering. *The Scientific World Journal 2014*, pages 1–19.