CONSENSUS CLUSTERING OF TIME SERIES DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYÇA YETERE KURŞUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

JANUARY 2014

Approval of the thesis:

CONSENSUS CLUSTERING OF TIME SERIES DATA

submitted by **AYÇA YETERE KURŞUN** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen _____
Head of Department, **Scientific Computing**

Prof. Dr. İnci Batmaz _____
Supervisor, **Department of Statistics, METU**

Assist. Prof. Dr. Cem İyigün _____
Co-Supervisor, **Department of Industrial Engineering, METU**

Examining Committee Members:

Prof. Dr. Gerhard-Wilhelm Weber _____
Department of Scientific Computing, METU

Prof. Dr. İnci Batmaz _____
Department of Statistics, METU

Assist. Prof. Dr. Cem İyigün _____
Department of Industrial Engineering, METU

Assoc. Prof. Dr. Ceylan Yozgatlıgil _____
Department of Statistics, METU

Assist. Prof. Dr. Serhan Duran _____
Department of Industrial Engineering, METU

Date:_____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: AYÇA YETERE KURŞUN

Signature :

ABSTRACT

CONSENSUS CLUSTERING OF TIME SERIES DATA

Yetere Kurşun, Ayça

M.Sc., Department of Scientific Computing

Supervisor : Prof. Dr. İnci Batmaz

Co-Supervisor : Assist. Prof. Dr. Cem İyigün

January 2014, 78 pages

In this study, we aim to develop a methodology that merges Dynamic Time Warping (DTW) and consensus clustering in a single algorithm. Mostly used time series distance measures require data to be of the same length and measure the distance between time series data mostly depends on the similarity of each coinciding data pair in time. DTW is a relatively new measure used to compare two time dependent sequences which may be out of phase or may not have the same lengths or frequencies. DTW aligns two time series data so that the distance between them is minimized. However, DTW is a similarity measure that is employed for single variable with standard clustering methods rather than consensus clustering. Thus our motivation is to create an algorithm that can combine the benefits of the DTW with benefits of consensus clustering, which will also provide a solution for multivariate applications. We present the results of our study both with simulated data, well known datasets from the literature and Turkey's long-term meteorological time series data between years 1950 and 2010. In all the cases we experimented with, when used with consensus clustering DTW performs better than Euclidian Distance measure. However in some cases the performance difference was insignificant, making it unnecessary to use both DTW and Consensus Clustering, due to time consuming computations. This thesis ends with a conclusion and the outlook to future studies.

Keywords: Consensus Clustering, Ensemble Clustering, Dynamic Time Warping, Time Series Clustering, Turkey Climate Regions

ÖZ

ZAMAN SERİSİ VERİLERİNİN ORTAK KÜMELENMESİ

Yetere Kurşun, Ayça
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. İnci Batmaz
Ortak Tez Yöneticisi : Yard. Doc. Dr. Cem İyigün

Ocak 2014, 78 sayfa

Bu çalışmanın amacı Devingen Zaman Eşleştirme (DZE) ve Ortak Kümeleme yaklaşımlarını bir araya getiren bir metodun oluşturulmasıdır. Zaman serisi verilerinin birbiri ile karşılaştırılmasında en sık kullanılan uzaklık metrikleri verilerin aynı boyutta olmasını gerektirir. Bu uzaklık metrikleri genelde verilerin zaman bazında karşılıklı gelen noktalarının yakınlıklarını kullanmaktadır. DZE zaman serisi verilerinin yakınlıklarının belirlenmesinde kısmen yeni bir metrik olup, arasında faz farkı bulunan, aynı boyutta olmayan ya da frekansları farklı olan verilerin karşılaştırılmasında kullanılabilmektedir. DZE iki zaman serisini birbirileri ile farkları en az olacak şekilde hizalamaktadır. Literatürde DZE metodu yaygın olarak tek değişken ve standart kümeleme algoritmaları ile birlikte kullanılmaktadır. Bu doğrultuda çalışmanın amacı DZE'nin ve ortak kümeleme metodolojilerinin avantajlarını bir araya getiren ve birden fazla değişkene sahip problemler için de kullanılabilecek bir algoritmanın oluşturulmasıdır. Çalışmanın sonuçları, bu çalışmaya özel yaratılan veri setleri, literatürde sık kullanılmış olan örnek veri setleri ve Türkiye'nin 1950-210 yılları arasını kapsayan uzun dönem meteorolojik zaman serisi verileri ile test edilmiştir. Tüm test verilerinde, DZE ile birlikte kullanılan ortak kümeleme algoritması standart Euclid uzaklığı ile gerçekleştirilen kümelemelerden daha iyi sonuçlar vermiştir. Bununlar birlikte bazı test durumlarında görülen fark çok küçüktür. Bu kapsamda, DZE ve ortak kümeleme algoritmasının çözümle sürelerinin uzunluğu da dikkate alındığında bu test durumları için DZE ve ortak kümeleme algoritmasının birlikte kullanımını gereksiz kılmaktadır. Tez, çalışma sonuçlarının ve gelecek dönem çalışmalarının bir özeti ile sonlanmaktadır.

Anahtar Kelimeler: Ortak Kümeleme, Devingen Zaman Eşleştirme, Zaman Serisi Kümeleme, Türkiye'nin İklim Bölgeleri

To My Baby Girl and My Husband

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Prof. Dr. İnci Batmaz and Assist. Prof. Dr. Cem İyigün for their guidance, encouragement, patience and understanding. I appreciate the time they have dedicated to me. Only with their kind support and valuable advices, I was able to complete this study. give

I also would like to express my sincere thanks to Prof. Dr. Gerhard-Wilhelm Weber, Prof. Dr. Ceylan Yozgatlıgil and Assist. Prof. Dr. Serhan Duran for their suggestions and comments for improving this study.

Also I am very grateful to my parents Halide Yetere and Ali Yetere, as they provided me with all the necessary building blocks of life.

Finally, I would like to thank to my dearest husband Birkan Kurşun. Throughout our marriage as well as in this study he inspired, encouraged and supported me. Thank you for your understanding and patience in my distressed times.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
DTW	Dynamic Time Warping
STS	Short Time Series

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction and Motivation

Clustering is the activity of unsupervised grouping of data points into classes so that the similar objects will be in the same cluster. There are varieties of clustering methods extensively used in literature such as *k*-means, hierarchical clustering, graph partitioning and so on. Since each method depends on different rationale, the results obtained from their use usually may not be the same. This situation leads to some confusion regarding which one gives the best clustering result. The common practice is to find the overlapping classes generated by different clustering methods and determine the non-overlapping observations. However, we may not come up with a solid solution with this approach. Alternatively, domain knowledge if available can be utilized to resolve this problem.

Consensus clustering is an attempt to solve this problem objectively; it tries to combine multiple clusterings of a dataset into one consolidated clustering. Consensus clustering methodologies offers benefits such as improved quality of solution, improved robustness against wide ranges of datasets, elimination of the model selection process, knowledge reuse, distributed clustering and effective consolidation of clusters depending on different views of data having multiple features [1].

Employing the clustering algorithms requires comparing two objects, thus one needs a distance (similarity) measure to define how much similar those two objects are. Commonly used similarity measures are Euclidean distance, Minkowski distance, Pearson's correlation coefficient and related distances, short time series distance and so on [2]. Mostly used time series distance measures require data to be of the same length and measure the distance between time series data mostly dependent on the similarity of each coinciding data pair in time. Here, the Dynamic Time Warping (DTW) is a relatively new measure used to compare two time dependent sequences with data which may be out of phase or may not have the same lengths or frequencies. DTW aligns two time series data so that the distance between them is minimized [2]. In literature DTW provided successful results when used for classification applications [3], while in this study it is used for clustering applications.

In this study, we aim to develop a methodology that merges DTW and consensus clustering in a single algorithm. DTW is a similarity measure that is employed for single variable with standard clustering methods rather than consensus clustering. Thus our motivation is to create an algorithm that can combine the benefits of the DTW with benefits of consensus clustering, which will also provide a solution for multivariate applications. In literature, time series clustering algorithms are used for several application areas like medicine, signal processing, economics, bio statistics and so on [4]. So we believe our approach with the DTW consensus clustering will also be applicable to those application areas.

In this study, the experimentation of our proposed DTW consensus clustering methodology will be performed by using both simulated data and well known datasets from the literature. Also by using Turkey's long-term meteorological time series data between years 1950 and 2010, we will try to identifying the climate zones of Turkey.

The DTW and consensus clustering methodologies employed in this study are introduced in Chapter 2 overviews clustering analysis, distance/similarity measures, basic clustering algorithms, time series clustering, specific distance and similarity measures for the time series data and finally consensus clustering.

Chapter 3 and Chapter 4 represent the results of our experimentation with several datasets, concluding with Chapter 5, including the future research possibilities.

1.2 Data Clustering Analysis

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) [5]. Each cluster comprises objects that are in a way more similar to one another and dissimilar to the objects in another cluster (see Figure 1.1). Thus, by seeking similarity, clustering analysis answers two basic questions: "How many groups (clusters) are there in a dataset?" and "Which data point belongs to which group (cluster)?" However there are many different ways of measuring the similarity of data points and many different ways to cluster the data into groups. So there is never a single answer to those questions but many depending on the similarity measures and clustering algorithms used.

Clustering has been used by many scientific disciplines as a data analysis technique. Clustering is useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations, including data mining, document retrieval, image segmentation, and pattern classification [5].

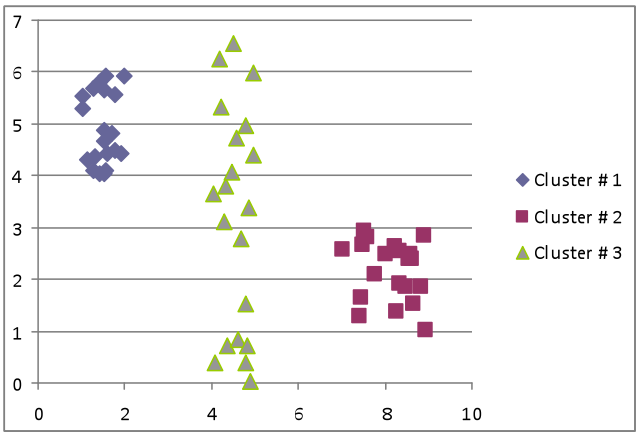


Figure 1.1 Data Clustering Example

1.3 Distance and Similarity Measures Used in Clustering Analysis

For clustering analysis, a measure to define the distance/similarity between two data points is essential. Many different distance and similarity measures are utilized in clustering analysis. The most commonly used similarity and distance measures for quantitative measures of continuous features are defined in Table 1.1. Because of the variety of feature types and scales, the distance measure (or measures) must be chosen carefully [5].

Table 1.1 Similarity/Distance Measures for Quantitative Measures [6]

Measures	Forms	Comments
Euclidian Distance	$D_{ij} = \left(\sum_{l=1}^{d} \left\| x_{il} - x_{jl} \right\|^2 \right)^{1/2}$	Features with large values and variances tend to dominate over other features.
Minkowski Distance	$D_{ij} = \left(\sum_{l=1}^{d} \left\| x_{il} - x_{jl} \right\|^n \right)^{1/n}$	Features with large values and variances tend to dominate over other features.
City-Block Distance	$D_{ij} = \sum_{l=1}^{d} \left\| x_{il} - x_{jl} \right\|$	Tend to form hyperrectangular clusters.
Sup Distance	$D_{ij} = \max_{1 \le l \le d} \left\| x_{il} - x_{jl} \right\|$	
Mahalonobis Distance	$D_{ij} = \left(\bar{x}_i - \bar{x}_j \right)^T S^{-1} \left(\bar{x}_i - \bar{x}_j \right)$	S is the within-group covariance matrix. Tend to form hyperellipsoidal clusters.

Table 1.1 (Continued) Similarity/Distance Measures for Quantitative Measures [6]

Measures	Forms	Comments
Pearson Correlation	$D_{ij} = (1 - r_{ij})/2$ $r_{ij} = \frac{\sum_{l=1}^{P}(x_{il} - \mu_{x_i})(y_{jl} - \mu_{y_j})}{S_{x_i} S_{y_j}}$ $\mu_{x_i} = \frac{1}{P}\sum_{l=1}^{P} x_{il}$ $S_{x_i} = \sqrt{\sum_{l=1}^{P}(x_{il} - \mu_{x_i})}$	Unable to detect the magnitude of differences of two variables.
Point Symmetry Distance	$D_{ij} = \min_{r=1,\ldots,N, j\neq i} \frac{\left\|(\bar{x}_i - \bar{x}_r) + (\bar{x}_j - \bar{x}_r)\right\|}{\left\|(\bar{x}_i - \bar{x}_r)\right\| + \left\|(\bar{x}_j - \bar{x}_r)\right\|}$	D_{ij} is minimize when a symmetric pattern exists.
Cosine Similarity	$S_{ij} = \cos\alpha = \frac{\bar{x}_i^T \bar{x}_j}{\left\|\bar{x}_i\right\|\left\|\bar{x}_j\right\|}$	Especially used for document clustering.

1.4 Basic Data Clustering Algorithms

There are many clustering algorithms available in the literature, each having different application areas. Traditionally clustering techniques can be classified as hierarchical clustering and partitional clustering, based on the property of clusters generated. Hierarchical clustering groups data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa, while partitional clustering directly divides data objects into some pre-specified number of clusters without an hierarchical structure. [6].

Berkhin [7] in his study has classified the clustering algorithms as follows:

- Hierarchical methods
  - Agglomerative algorithms
  - Divisive algorithms
- Partitioning relocation methods
  - Probabilistic clustering
  - k-medoids methods
  - k-means methods
- Density-based partitioning methods
  - Density-based connectivity clustering
  - Density functions clustering
- Grid-based methods

- Methods based on co-occurrence of categorical data
- Other clustering techniques
    - Constraint-based clustering
    - Graph partitioning
    - Clustering algorithms and supervised learning
    - Clustering algorithms in machine learning
- Scalable clustering algorithms
- Algorithms for high-dimensional data
    - Subspace clustering
    - Coclustering techniques

In this study we will only focus on the most commonly used algorithms, which are also utilized in the experimentation part of this study. Those algorithms are Hierarchical clustering, both agglomerative and divisive, k-medoids and k-means methods which are partitioning clustering methods.

1.4.1 Hierarchical Methods

This clustering method arranges the data into a hierarchical structure by the use of similarity/distance matrix. Hierarchical clustering creates a tree, called a "dendogram" (see Figure 1.2 (b)), representing the whole dataset, in which the data points are leaves. The internal nodes describes the similarity structure of the points, in another saying, the extend of the proximity between data points. The height of the dendogram usually gives the distance between each pair of objects or clusters, or an object and a cluster [6]. The final clustering results can be acquired by cutting the dendogram at the demanded level (frequently, the requested number *k* of clusters). The advantages and disadvantages of the hierarchical clustering are discussed in Table 1.2.

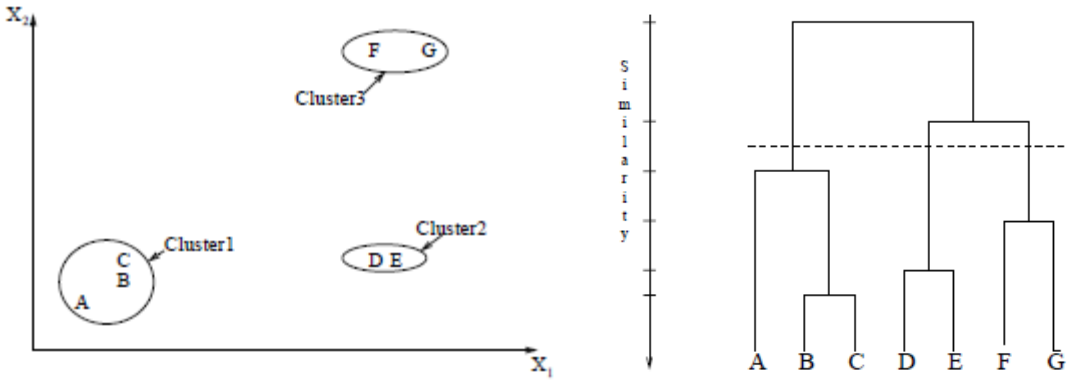


Figure 1.2 (a) An Example of a Three Cluster Problem, and (b) Regarding Dendogram Obtained by Hierarchical Clustering [5]

Table 1.2 Advantages and Disadvantages of the Hierarchical Clustering [7]

Advantages	Disadvantages
• Flexibility regarding the level of granularity, • Ease of handling any form of similarity or distance, • Applicability to any attribute type.	• The difficulty of choosing the right stopping criteria, • Most hierarchical algorithms do not revisit (intermediate) clusters once they are constructed.

1.4.1.1 Agglomerative Hierarchical Clustering

Agglomerative clustering starts with N clusters (the number of data points) each containing only one object. Then successions of merge operations are performed until finally all the data points are in the same cluster. So the general algorithm is as follows [6]:

1. Create N clusters, one for each object.
2. Calculate the distance matrix between clusters.
3. Find the minimal distance between clusters, using the following equation:

$$D(C_i, C_j) = \min_{1 \le m, l \le N, m \ne l} D(C_m, C_l), \tag{1.1}$$

4. Combine cluster C_i and C_j to form a new cluster.
5. Update the distance matrix between clusters by computing the distances between the new cluster and the other clusters.
6. Repeat steps 3– 5 until all objects are in the same cluster.

There are different definitions for a distance between two clusters; based on those definitions agglomerative clustering methods can result in different solutions. The most commonly used definitions for the closest pair of clusters are “Single Linkage,” “Average Linkage” and “Complete Linkage.” In Single Linkage the distance between two clusters is the distance between the two closest objects, whereas Average Linkage measures the distance between clusters as the distance between the cluster centroids and the Complete Linkage is the distance between most distant pair of objects (see **Figure 1.3**).

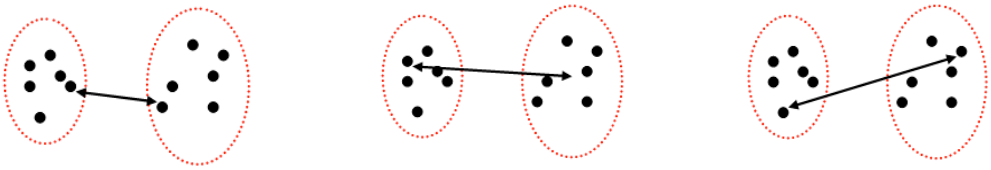


Figure 1.3 Distance Between Two Clusters

1.4.1.2 Divisive Hierarchical Clustering

As opposed to agglomerative clustering, for divisive clustering, in the beginning, the entire dataset belongs to a cluster, and then, a procedure successively divides it until all clusters are singleton clusters. So the general algorithm is as follows [7]:

1. Put all the objects in one cluster.
2. Select the cluster C_i for splitting.
3. Split cluster C_i into two new sub-clusters C_k and C_l.
4. Replace cluster C_i with the new sub-clusters C_k and C_l.
5. Repeat steps 2–4 until all clusters have exactly one object (N clusters, one for each object).

There are different divisive algorithms depending on the way they select the cluster for splitting and the way they split the selected cluster.

1.4.2 Partitioning relocation methods (*k*-means and *k*-medoids)

Partitioning relocation methods divide data into k subsets. But as it is not computationally feasible to create and evaluate every possibility, heuristics are used for finding the optimal clusters iteratively. This heuristics use different relocation methods to iteratively reassign points between the k clusters. For this purpose each cluster is associated with a cluster representative. There are two basic partitioning relocation methods: namely, *k*-means algorithm and *k*-medoids algorithm. Both of those algorithms attempt to minimize the distance between the points within the same cluster. In *k*-means algorithm, a fictitious data point is created, which is the centroid of the data points in the same cluster and the distance between this centroid and other data points are tried to be minimized. However in *k*-medoids, the algorithm chooses a particular data point as the cluster center and the distance between this data point and the other data points are tried to be minimized.

Berkhin [7] in his study explained that representation by *k*-medoids has two advantages: it presents no limitations on attribute types and the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster, and therefore, it is insensitive to the presence of outliers. But in *k*-means a cluster is represented by its centroid, which is the mean of points within a cluster. Thus *k*-means only with numerical attributes and can be negatively affected by a single outlier. On the other hand, centroids have the advantage of clear geometric and statistical meaning.

The general *k*-means algorithm is as follows [6]:

1. Select the number of clusters, *k*.
2. Select *k* data points as the *k* cluster centers (i.e; randomly).
3. For each object find the cluster to be in by assigning them to the nearest cluster center.
4. Re-calculate the *k* cluster centers.
5. Repeat steps 3–4 until cluster membership does not changes from the previous iteration.

The general *k*-medoids algorithm is as follows [7]:

1. Select the number of clusters, *k*.
2. Form the *k* medoids by selecting *k* data points as the *k* cluster centers (i.e; randomly).
3. For each object find the cluster to be in by assigning them to the nearest medoid.
4. For every cluster, select a non-medoid object to replace with the cluster medoid.
5. For each object find the cluster to be in by assigning them to the nearest medoid.
7. Calculate the total distances from the medoids for all clusters obtained through steps 4 and 5.
8. Select the lowest distance alternative.
9. Repeat steps 4–8 until cluster medoids does not changes from the previous iteration.

1.5 Time Series Data Clustering

When the values of a data do not change over time then this data is called to be static. Many of the clustering analysis in the literature are performed with those kinds of datasets. For a dynamic dataset, as in time series data, comprise values that changes with time. Time series data is of interest because of its pervasiveness in various areas ranging from science, engineering, business, finance, economic, health care, to

government [2]. So there is also a need for clustering algorithms and similarity/distance measures dealing with time series data.

1.5.1 Clustering Algorithms for Time Series Data

Mainly there are no specific clustering algorithms dealing especially with the time series data. However here in this section we will try to name some of the general purpose clustering algorithms that have been used in the literature for the time series clustering analysis.

Liao in his survey research [2] regarding the clustering of time series data, extensively identifies the time series clustering approaches. The most commonly used algorithms surveyed in this study are summarized in Table 1.3.

Table 1.3 Commonly Used Clustering Algorithms

Clustering algorithm	Distance measure	Variable	Ref.
Agglomerative hierarchical	Euclidean	Single	[11] [12] [13]
	Root mean square	Single	[14]
	Kullback–Leibler distance	Single/ Multiple	[15] [16]
	J divergence and symmetric Chernoff information divergence	Multiple	[17]
	Based on the assumed independent Gaussian models of data errors	Single	[18]
k-Means (including the modified *k*-means)	Euclidean	Single / Multiple	[19] [20] [21]
k-Medoids	DTW	Single	[22]
	Euclidean	Single	[23]
Fuzzy c-means (including the modified Fuzzy c-means)	Euclidean	Single	[24]
	Two cross-correlation based	Single / Multiple	[24]
	Short time series (STS) distance	Single	[25]

1.5.2 Similarity and Distance Measures for Time Series Data Clustering

Even though clustering algorithms may not require a special version or an adjustment for analysis of time series data; this is not the case for similarity and distance measures. In this part of our study we will try to introduce, from Table 1.3, the most commonly used similarity and distance measures for the time series data.

1.5.2.1 Euclidian and Root Mean Square Distance

The Euclidian distance is already introduced in Section 1.3Distance and Similarity Measures Used in Clustering Analysis, Table 1.1. For time series data the same formula is used, where x_i and x_j are d dimensional vectors, d being the length of the time series data. Root Mean Square distance (D_{RMS}) is simply the average geometric distance,

$$D_{RMS} = \frac{\left(\sum_{l=1}^{d} \left| x_{il} - x_{jl} \right|^2 \right)^{1/2}}{d}. \tag{1.2}$$

1.5.2.2 Kullback-Leiber Distance

The Kullback-Leibler distance measures how different two probability distributions are. Let p_i and p_j be the transition probability matrices for two Marko chains with d probability distributions each, and p_{ilk} and p_{jlk} be the transition probabilities from l to k in p_i and p_j. Then the Kullback-Leibler distance of two probability distributions is [2],

$$D(p_{il}, p_{jl}) = \sum_{k=1}^{d} p_{ilk} \log\left(\frac{p_{ilk}}{p_{jlk}} \right). \tag{1.3}$$

1.5.2.3 Dynamic Time Warping

DTW algorithm is known for being efficient as the time series similarity measure, which minimizes the effects of shifting and distortion in time by allowing “elastic” transformation of time series in order to detect similar shapes with different phases [8]. Speech recognition has been a well known applications area of DTW for a long time, but in literature DTW was also used in very diverse areas such as bioinformatics, chemical engineering, signal processing, robotics and aligning

biometric data, signatures and fingerprints [4]. Details of the DTW algorithm will be discussed in "CHAPTER 2."

1.5.2.4 Cross-Correlation

Dissimilarity based on cross correlation ($\rho_{i,j}^2(\tau)$) of two time series data (x_i and x_j) can be represented with Equation (1.4), where max is the maximum lag [2].

$$D(i,j) = \sqrt{\frac{1-\rho_{i,j}^2(0)}{\sum_{\tau=1}^{\max}\rho_{i,j}^2(\tau)}}, \tag{1.4}$$

1.5.2.5 Short time series (STS) distance

Möller-Levet et al. [25] proposed the STS distance measure by considering each time series as a piecewise linear function [2]. The STS distance between the series x_i and x_j (x_i and x_j are *d* dimensional vectors, *d* being the length of the time series data) is defined as in Equation (1.5). Here t_k represents the time for data point x_i and x_j.

$$D(i,j) = \sqrt{\sum_{k=1}^{d}\left(\frac{x_{j(k+1)}-x_{jk}}{t_{(k+1)}-t_k}-\frac{x_{i(k+1)}-x_{ik}}{t_{(k+1)}-t_k}\right)^2}, \tag{1.5}$$

1.5.2.6 J divergence and symmetric Chernoff information divergence

J divergence and symmetric Chernoff information divergence is used for spectral matrix estimators for different stationary vector series [2]. Details for those distance measures can be found in the study of Kakizawa et al. [17].

CHAPTER 2

METHODS

2.1 Dynamic Time Warping

Dynamic Time Warping algorithm is a time-series similarity measure, which can be used for data that may be out of phase or may not have the same lengths or frequency for that matter. DTW algorithm is as follows;

Step 1. Generating the Cumulative Distance Matrix

The first step is to compare each point in one time series data with every other point in the second time series data, generating a matrix. So the cumulative distance between time series data points is calculated using dynamic programming technique.

Given two time series $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$, the cumulative distance matrix can be found using equations (2.1), (2.2) and (2.3). In those equations, the Euclidian distance between two data points is normally used for defining $d(x_i, y_j)$.

$$dtw(1, j) = d(x_1, y_j) + dtw(1, j-1), \quad (2.1)$$

$$dtw(i,1) = d(x_1, y_j) + dtw(i-1,1), \quad (2.2)$$

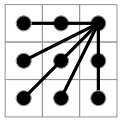
$$dtw(i, j) = d(x_i, y_j) + \min(dtw(i-1, j-1), dtw(i-1, j), dtw(i, j-1)). \quad (2.3)$$

In the above formulation one step dynamic programming is used, however the depth (time window) of algorithm can be defined specific to the problem nature. Figure 2.1 shows different step size (or constraints) examples that might be used with DTW algorithm.

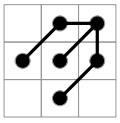
The Euclidean distance measure can be seen as a special case of DTW with step size being equal to zero. However, this special case can only be defined when the two time series have the same length.

The detailed algorithm for creating the cumulative distance matrix is given in Table 2.1.

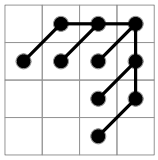
(a) Standard constraint for step size one



(b) Version for the standard constraint for step size two



(c) Constraint suggested by Sakoe et al. [33]



(d) Constraint suggested by Sakoe et al.[33]

Figure 2.1 Different Step Size (or Constraint) Examples for DTW Algorithm

Table 2.1 Detailed Cumulative Distance Matrix Algorithm

Detailed Cumulative Distance Matrix Algorithm with Euclidian Distance
$n = size(X)$
$m = size(Y)$
$dtw(1:n, 1:m) = 0$
$dtw(1,1) = (x_1 - y_1)^2$
for $i = 2:n$
$dtw(i,1) = (x_i - y_1)^2 + dtw(i-1,1)$
end
for $j = 2:m$
$dtw(1,j) = (x_1 - y_j)^2 + dtw(1,j-1)$
end
for $i = 2:n$
for $j = 2:m$
$dtw(i,j) = (x_i - y_j)^2 + \min(dtw(i-1,j-1), dtw(i-1,j), dtw(i,j-1))$
end
end
$Cummulative\_Distance = (dtw(n,m))^{1/2}$

Step 2. Finding The Optimal Path

The optimal warping path is the minimum distance path on the cumulative distance matrix. The minimum distance path is a sequence of points $p = (p_1, p_2, \ldots, p_K)$ with $p_l = (p_i, p_j) \in [1:n] \times [1:m]$ for $l \in [1:K]$, $i \in \mathbb{Z}$, $j \in \mathbb{Z}$ and $l \in \mathbb{Z}$, satisfying the following conditions:

Boundary condition: The starting and ending points of the warping path must be the first and the last points of aligned sequences, $p_1 = (1,1)$ and $p_K = (n,m)$.

Monotonicity condition: $n_1 \le n_2 \le \ldots \le n_K$ and $m_1 \le m_2 \le \ldots \le m_K$. This condition preserves the time-ordering of points.

Step size condition: Limits the warping path making big shifts in time while aligning sequences. Step size condition can be formulated as $p_{l+1} - p_l \in \{(1,1), (1,0), (0,1)\}$ for a single step size.
So starting in reverse order with $p_K = (n,m)$ and finishing with $p_1 = (1,1)$, the simple procedure for the optimal path is described in (2.4) [9]:

$$p_l = (p_i, p_j) = \begin{cases} (1, j-1), & \text{if } i = 1, \\ (i-1, 1), & \text{if } j = 1, \\ \arg\min\{dtw(i-1, j-1), dtw(i-1, j), dtw(i, i-1)\}, & \text{otherwise.} \end{cases} \quad (2.4)$$

Table 2.2 Detailed Optimal Warping Path Algorithm

Detailed Optimal Warping Path Algorithm
$[i, j] = size(dtw)$
$K = 0$
while $(i > 1) \& \& (j > 1)$
if $i == 1$
$j = j - 1$
else if $j == 1$
$i = i - 1$
else
if $dtw(i-1, j) == \min(dtw(i-1, j-1), dtw(i-1, j), dtw(i, j-1))$
$i = i - 1$
else if $dtw(i, j-1) == \min(dtw(i-1, j-1), dtw(i-1, j), dtw(i, j-1))$
$j = j - 1$
else
$i = i - 1$

Table 2.2 (Continued)Detailed Optimal Warping Path Algorithm

15: $j = j - 1$
16: end
17: end
18: $K = K + 1$
19: $path(K,:) = (i, j)$
20: end

2.2 An Example for the Dynamic Time Warping Algorithm:

Suppose we compare two time series data,

Time Series Data-1: $X = [0 \quad 0 \quad 2 \quad 4 \quad 7 \quad 12 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]$
Time Series Data-2: $Y = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 3 \quad 8 \quad 13 \quad 1 \quad 0]$

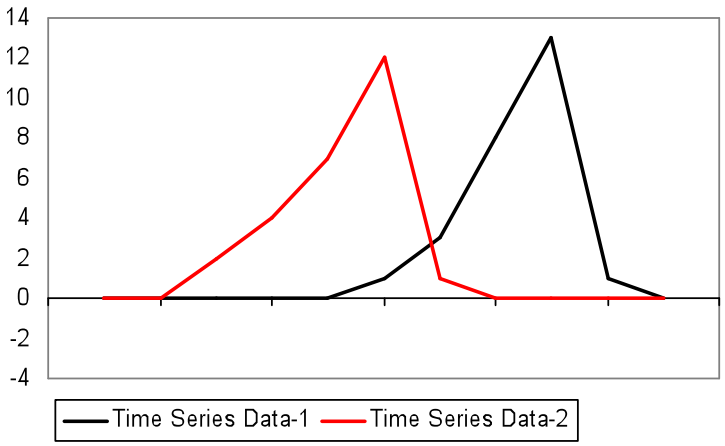


Figure 2.2 Graphic Representation of the Time Series Data

Actually the two series is quite similar with peaks and slopes with a small phase shift. However, pairwise comparison of the data points would indicate the data is not similar. When these two time series are compared by Euclidian Norm, the distance between the series is 20.69. Yet, if we use DTW, the distance between the two series is only two. The cumulative distance matrix is presented in Figure 2.3 with optimal warping path highlighted with red color.

11	0	214	214	214	214	214	215	224	288	429	8	**4**
10	0	214	214	214	214	214	215	224	260	365	7	**4**
9	0	214	214	214	214	214	215	222	196	301	6	**4**
8	0	214	214	214	214	214	214	213	132	237	5	**4**
7	1	214	214	214	214	214	213	194	68	148	**4**	5
6	12	213	213	213	213	213	190	99	19	**4**	125	269
5	7	69	69	69	69	69	74	18	**3**	39	75	124
4	4	20	20	20	20	20	10	**2**	18	99	108	124
3	2	4	4	4	4	4	**1**	2	38	159	160	164
2	0	0	0	0	0	**0**	1	10	74	243	244	244
1	0	**0**	**0**	**0**	**0**	**0**	1	10	74	243	244	244
		0	0	0	0	0	1	3	8	13	1	0
		1	2	3	4	5	6	7	8	9	10	11

Figure 2.3 Cumulative Distance Matrix and the Optimal Warping Path

2.3 Consensus Clustering

In literature, the idea of using several runs of one or more clustering algorithms, different parameters of an object or dataset resamples, to create better clusters is know as consensus clustering, clustering aggregation or in other words ensemble clustering. In this study we will use the term "consensus clustering" to indicate this idea. However the only reason for using consensus clustering is not only to obtain better clustering. Ghosh et al. [1] and Ghaemi et al. [26] lists other reasons to use consensus clustering as follows:

- Improved quality of solution: Better clustering results as compared to a single clustering solution
- Novelty: Achieving a better clustering solution that can not be obtained by using any single algorithm.
- Robust clustering: Obtaining good results across wide ranges of domain and datasets by constructing 'meta' clustering models.
- Stability and confidence estimation: Clustering solutions with lower sensitivity to noise, outliers, or sampling variations.
- Model selection: Cluster ensembles provide an approach to the model selection problem by considering the match across the base solutions to determine the final number of clusters to be obtained [27].
- Knowledge reuse: A consensus solution can combine different clusterings of the objects due to past projects to get a more consolidated clustering.
- Multiview clustering: Effective consolidation of clusters depending on different views of data having multiple features.
- Distributed computing (Parallelization and Scalability): Consolidation of parallel clustering results from distributed sources of data or features when it

is not possible to first collect the entire data (subset of the features of each object etc.) at a central site.

Consensus clustering is generally a two-stage approach. First stage is to create diversity of clustering and the second stage is to obtain a consensus across those diverse solutions by utilizing an algorithm (see Figure 2.4). Diversity can be achieved by several mechanisms [1] [26]:

- Using different clustering algorithms.
- Using different initialization points or parameters.
- Using different subsets of data or creating resamples from the original data.
- Using different features of data.

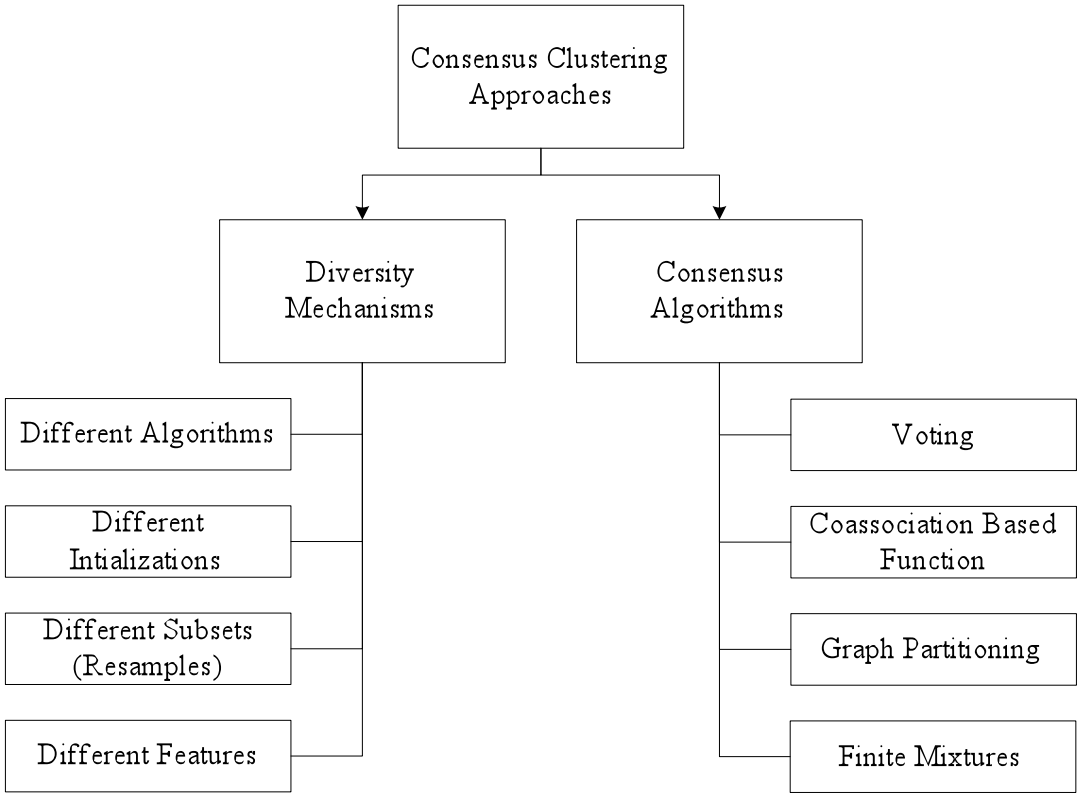


Figure 2.4 Consensus Clustering Approaches [1] [26]

2.3.1 Consensus Clustering Algorithms

2.3.1.1 Voting

Voting is the most simple consensus approach. While using this approach initially a diversity mechanism is used to create a group of different solutions, then a voting algorithm is used to assign data points to clusters in order to determine the final consensus clustering. Voting algorithm simply assigns the cluster labels to the data points by determining the majority vote. Dudoit et al. [31] and Fischer et al. [32] propose a consensus functions that is based on voting.

2.3.1.2 Coassociation Based Function

This approach is actually a pairwise similarity based approach. In this approach a combined coassociation matrix is generated by using the ratio of a number of clusterings in which the two data points are shared the same clusters to the total number of clusterings in the ensemble [26]. This matrix is actually a similarity matrix defining the similarity between the data points. This matrix can be solved by using one of the many clustering algorithms in order to obtain a final consensus solution. This approach due to its simplicity and intuitiveness was used in many studies ([10], [29], [30]).

2.3.1.3 Graph Partitioning

Clusters can be represented as edges on a hypergraph in which data points are the vertices. Thus edges connecting the vertices define the data points within the same cluster. To solve the problem of consensus clustering Strehl et al. [30] proposed an approach called Hypergraph-Partitioning Algorithm (HGPA). This algorithm re-partitions the data using the given clusters. To obtain the consensus clusters the hypergraph is partitioned into k unconnected components of approximately the same size by cutting a minimal number of hyperedges [30]. Strehl et al. [30] also proposed the Meta-CLustering Algorithm (MCLA) in their study which also uses hypergraphs. They define the idea in MCLA as "to group and collapse related hyperedges and assign each object to the collapsed hyperedge in which it participates most strongly."

2.3.1.4 Finite Mixtures

While using the finite mixture models the main assumption is to model cluster labels as random variables drawn from a probability distribution described as a mixture of

multinomial distributions. The most known study using the mixture models was done by Topchy et al. [28]. In their study authors propose a probabilistic model of consensus using a finite mixture of multinomial distributions in a space of cluster labels. Combined clustering can be found by solving a simple maximum log-likelihood problem by using the expectation maximization algorithm.

2.4 Consensus Methodology Used In This Study

The consensus methodology developed by Monti et al. [10] proposes a resampling-based approach for class discovery and clustering validation. The proposed methodology provides a method to represent the consensus across multiple runs of clustering algorithms [10]. This approach is an coassociation matrix based approach for consensus clustering. Monti et al. [10] expressed their motivation for the proposed methodology as to increase the robustness and stability of clusters to sampling variability. They have also explained that their method can also be used to represent the consensus over multiple runs of a clustering algorithm with random restart so as to account for the sensitivity to the initial conditions. Even though it was not mentioned in their paper, their approach is also suitable for obtaining a consensus result for different clustering algorithms, as it is suggested by Simpson [34]. Also it is suitable for multiview (multivariate) clustering, giving way for the utilization of different features of the data.

So Monti et al.s' proposed methodology, by using different clustering algorithms, different initialization points or parameters, different features of data and resamples from the original data, has the benefits of "Improved Quality of Solution," "Novelty," "Robust Clustering" and "Stability" which are discussed in Section 2.3.

In this study, we use this proposed approach for achieving a consensus clustering result, by also including the usage of different clustering algorithms. The multivariate case will be discussed in Section 2.5. This consensus clustering approach can simply be summarized as follows [10]:

For a selected bootstrapping technique with different clustering algorithms and number of clusters;

1. Resample the dataset for h iterations (in our case the square distance matrix developed by DTW will be resampled).
2. Select a number of clustering algorithms for the consensus solution, $K = \{k-means, Agglomerative\ Nesting, ...\}$

Starting from the first clustering algorithm, $k = 1$, repeat Step 3 and 4 for all the clustering algorithms:

3. Apply the clustering algorithm to each and every resampled dataset.

4. Compute the consensus clustering matrix using all the runs (each and every resampled dataset) from the same algorithm. Here, the consensus clustering matrix for the k^{th} clustering algorithm C_k^* can be generated using the following equation:

$$C_k^*(i,j) = \frac{\sum_h C_k^h(i,j)}{\sum_h I_k^h(i,j)}. \tag{2.5}$$

Here, C_k^h is the connectivity matrix corresponding to the h^{th} iteration of the k^{th} clustering algorithm, where

$$C_k^h(i,j) = \begin{cases} 1, & \text{if items } i \text{ and } j \text{ belong to the same cluster,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

Furthermore, I_k^h is the indicator matrix corresponding to the h^{th} iteration of the k^{th} clustering algorithm such that

$$I_k^h(i,j) = \begin{cases} 1, & \text{if items } i \text{ and } j \text{ are present in the same resampled} \\ & \text{dataset,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

5. Combine the consensus clustering matrices obtained for each algorithm using weights (w_k) in order to form the following merged matrix C^*

$$C^*(i,j) = \sum_k w_k C_k^*(i,j). \tag{2.8}$$

6. Use the merged matrix C^* as a similarity matrix and obtain the final clustering solution.

2.5 Multivariate Problems

As mentioned earlier in Section 2.3, one of the benefits offered by consensus clustering is to obtain a single consolidated partition by effectively combining all the clusterings of different aspects of the data. For the multivariate case there can be several different approaches to tackle the problem. This study deals with the following two approaches:

- **Combining the similarity matrices into a single merged similarity matrix and obtaining the clusters with the consensus clustering algorithm:** As for each variable before creating a consensus solution with the algorithm defined in Section 2.4, one can create an ensemble using the similarity matrices of each variable using the DTW methodology. This can be simply done by using (2.9), where DTW_n represents the similarity matrix for the n^{th} variable:

$$DTW\_Merged = \frac{\sum_{n=1}^{N} DTW_n}{N}. \tag{2.9}$$

- **Combining the merged matrix of each variable and obtaining a final consensus clustering:** It is also possible to use the same approach defined in Section 2.4 to obtain the ensemble of variables using the merged matrices defined in Step 5 of the consensus algorithm (see Figure 2.5).

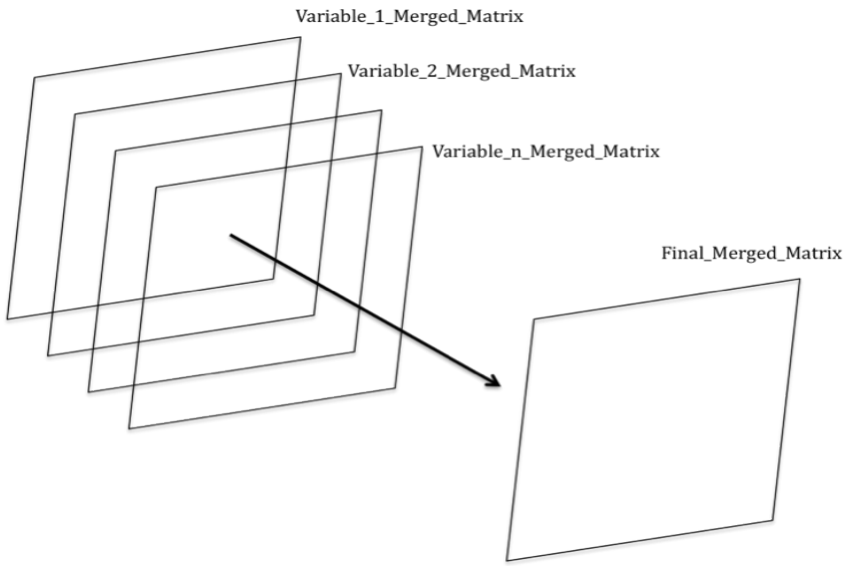


Figure 2.5 Final Merged Matrix

In that case one can obtain the final merged matrix using Equation (2.10), where $C_n^*(i,j)$ represents the merged matrix for the n^{th} variable obtained by using the consensus clustering algorithm:

$$C^*(i,j) = \frac{\sum_{n=1}^{N} C_n^*(i,j)}{N}. \tag{2.10}$$

CHAPTER 3

EXPERIMENTATION

In order to test our proposed approach we have experimented with four distinct datasets. Two of those datasets were created specific for this study. The other two datasets are used in the literature for testing of the classification algorithms for time series data. Those two datasets are "Synthetic Control Dataset" and "Daily and Sports Activities Dataset" [35].

In literature "Agglomerative Hierarchical" clustering and "*k*-means" Clustering algorithms with Euclidian distance measure are the mostly used algorithms for time series data clustering analysis. Hence we have used those algorithms in order to compare the performance of theirs to that of our proposed algorithm. Initially we will discuss the performance of just DTW as a time series distance measure when compared to that of the Euclidian distance measure. For that purpose we will be using different window sizes (namely, window sizes 1, 2, 3, 4 and 5).

For obtaining consensus clustering merged matrices, we have utilized the R Package that was created by Simpson [34]. For all datasets we have used Agglomerative Nesting (Hierarchical Clustering), Partitioning Around Medoids, Divisive Analysis Clustering and *k*-means as different clustering algorithms within the consensus clustering algorithm. All four clustering algorithms were equally weighted for calculation of the final consensus clustering merged matrix. Different parameter sets used for different clustering algorithms are presented in Table 3.1. In order to obtain the clustering labels, for final clustering we solved the merged matrices obtained from both Agglomerative Nesting (Hierarchical) and *k*-means clustering.

Table 3.1 Parameter Sets Used for Different Clustering Algorithms

Algorithm Definition	Distance Measure	Method	Other Parameters
Agglomerative Nesting (Hierarchical Clustering)	Euclidean	Average Linkage	R defaults
Partitioning Around Medoids	Euclidean	-	R defaults
Divisive Analysis Clustering	Euclidean	-	R defaults
k-means	-	Hartigan-Wong	R defaults

3.1 Experimentation with Simulated Dataset -1

For initial experimentation a dataset with eight clusters, each having 50 randomly generated time series data with 100 time points, are created. All the time series data have a sinusoidal behavior with decreasing frequency. Cluster1 has a doubled period when compared to Cluster 2. Cluster 1-2, Cluster 5-6, and Cluster 3-4, Cluster 7-8 has the same behavior, respectively, except Cluster 3-4 and 7-8 has smaller amplitude than Cluster 1-2 and 5-6. Yet all those clusters have overlapping data points. The representation of the dataset is presented in Figure 3.1.

Run times for the generation of similarity matrices, run times for the clustering algorithms, errors and the identified cluster labels from the real cluster labels was presented in Table 3.2, Table 3.3 and Table 3.4 respectively.

In order to obtain the error values, once the clustering is performed, for each cluster the real cluster label is found by declaring the cluster label with majority as the dominant real cluster label. After the real cluster label is found for each cluster, error values are calculated using the following equation:

$$E_i = \frac{\sum_{j=1}^{N_i} L_j}{N_i}, \text{ where} \tag{3.1}$$

$$L_j = \begin{cases} 1, & \text{if the real clutser label of object } j \text{ is not equal to the} \\ & \text{dominat real cluster label,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

In Equation (3.1), E_i represents the total error of Cluster i and N_i represents the number of objects in Cluster i.

From the run time results for the generation of similarity matrices (see Table 3.2), it can easily be seen that DTW is computationally very expensive since even with a time window of size one the ratio between the run times of DTW and that of Euclidian is nearly 2500. This ratio increases to nearly 5000 for time window equal to five.

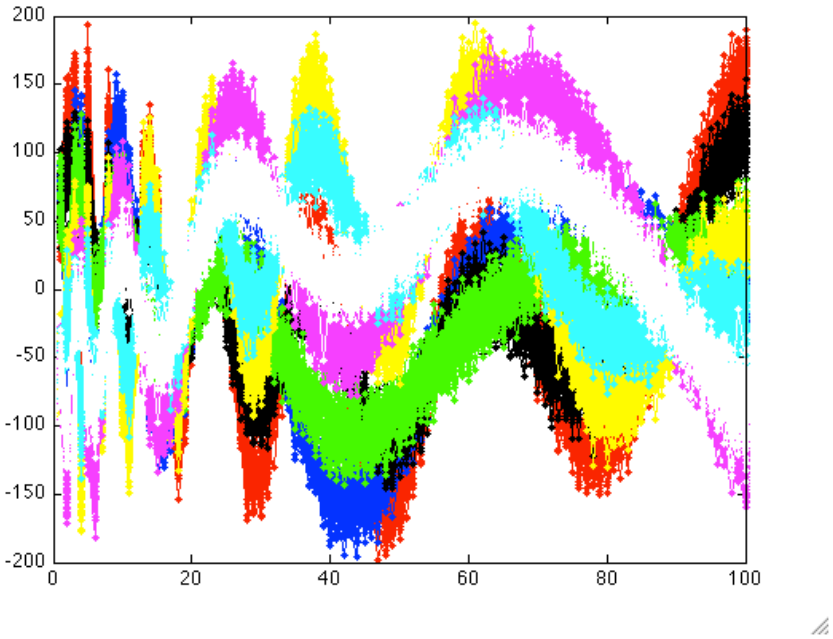


Figure 3.1 Simulated Dataset -1

Table 3.2 Generation Times (Sec) for Similarity Matrices - Simulated Dataset -1

	Generation Times (Sec)
dtw window 1	1536.2000
dtw window 2	1714.4000
dtw window 3	2003.6000
dtw window 4	2439.5000
dtw window 5	3071.5000
Euclidian Distance	0.6372

The results show that all the distance measures and clustering algorithms were able to identify all the original cluster labels (see Table 3.4). Yet *k*-means was not able to cluster all the data points truly with any of the window sizes. For this dataset it can be seen that the error rate of the clustering algorithms increases for all algorithms as the time window size for DTW increases. Even though this is the case consensus clustering algorithm makes less errors for larger window sizes when compared with the conventional clustering algorithms.

Table 3.3 Run Times (Sec) for Clustering Algorithms - Simulated Dataset -1

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.5478	47.5886	8135.3794	8175.9848
dtw window 2	0.3865	47.5702	7710.4353	7751.2336
dtw window 3	0.4065	63.8660	7705.3883	7746.7287
dtw window 4	0.3533	61.3791	7525.4211	7585.5954
dtw window 5	0.4061	54.2084	7755.4284	7813.6511
Euclidian Distance	0.3398	48.8171	7720.4343	7764.1053

Table 3.4 Errors and Clusters for Clustering Algorithms - Simulated Dataset -1

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
	0.0000	0.0025	0.0000	0.0000
dtw window 1	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8
	0.0000	0.0025	0.0000	0.0000
dtw window 2	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8
	0.0000	0.0050	0.0000	0.0000
dtw window 3	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8
	0.0075	0.0125	0.0000	0.0000
dtw window 4	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8
	0.1475	0.0300	0.0100	0.0075
dtw window 5	1,2,3,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8
Euclidian	0.0000	0.0000	0.0000	0.0000
Distance	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8	1,2,3,4,5,6,7,8

3.2 Experimentation with Simulated Dataset -2

In order to observe the performance difference more realistically between the algorithms we created the second dataset which would be harder to cluster. We created a dataset with four clusters, each having 50 randomly generated time series data with 100 time points. All the time series data have a sinusoidal behavior with a superimposed upward polynomial behavior. Each time series data has also a randomly generated phase shift. Cluster1 has a doubled period when compared to Cluster 2. All clusters has the same behavior except that Cluster 1-2 has a smaller standard deviation than Cluster 3-4. Yet all those clusters have overlapping data points. The representation of the dataset is presented in Figure 3.2.

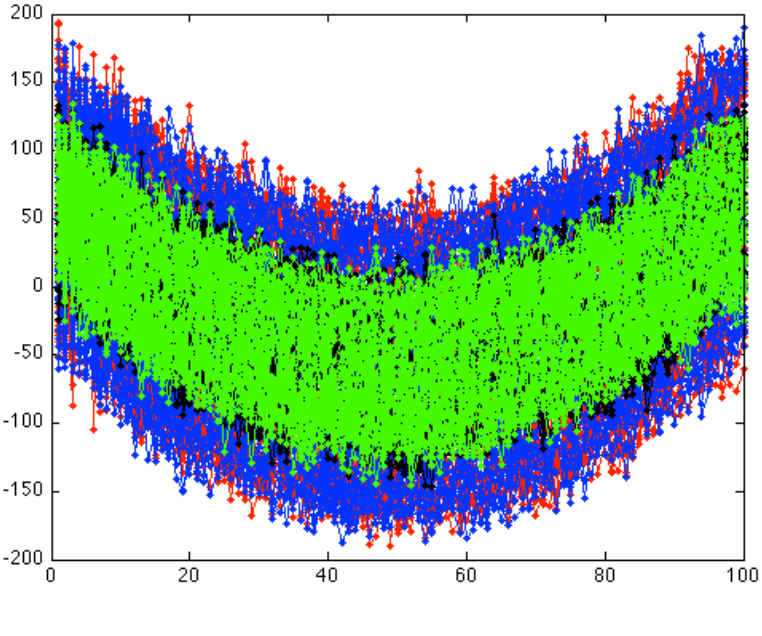


Figure 3.2 Simulated Dataset -2

In order to define the similarities between the four simulated clusters we can use the cross-correlation coefficient with zero lag. This matrix is given in Table 3.5. Coefficients presented in this matrix are the average absolute coefficients over the 50 randomly generated samples. The diagonal of the matrix defines the auto-correlation coefficient of each cluster. As can be seen from the matrix clusters 1-3, 2-4 and 3-4 have rather high correlation coefficients when compared to clusters 1-2, 1-4 and 2-3. Thus one can expect to have mislabeling for clusters 1-3, 2-4 and 3-4.

Table 3.5 Correlation Coefficients for Four Clusters

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	0.5120	0.2179	0.4302	0.3230
Cluster 2	0.2179	0.5170	0.3123	0.4318
Cluster 3	0.4302	0.3123	0.5204	0.4702
Cluster 4	0.3230	0.4318	0.4702	0.4802

Run times for the generation of similarity matrices, run times for the clustering algorithms, error rates and the identified cluster labels from the real cluster labels for $k \in \{4,5,6,7,8\}$ was presented in Table 3.6, Table 3.7, Table 3.8 Table 3.9, Table 3.10, Table 3.11, Table 3.12, Table 3.13, Table 3.14, Table 3.15, Table 3.16 .

From the run times for the generation of similarity matrices it can easily be seen that DTW is computationally very expensive since even with a time window size of one,

the ratio between run times of DTW and Euclidian is nearly 1800. It increases to nearly 3400 for time window size equal to five.

Regarding the clustering algorithm run times, clustering errors and cluster discovery results, following observations were made:

- Results show that the consensus clustering algorithm with DTW (window size equal to one) as the distance measure perform better for all $k \in \{4,5,6,7,8\}$, when compared both with respect to the conventional clustering algorithms (AGNES, *k*-means) and with respect to the similarity distance measures (DTW window size 2-5 and Euclidian Distance Measure).
- As the final clustering algorithm *k*-means (with DTW window size equal to one) perform better (between 1.4% and 25%) at three cases out of five, with two cases AGNES and *k*-means have the same errors.
- Consensus clustering is computationally very time consuming, and hence expensive.
- Leaving out the consensus clustering results, when DTW is compared with the Euclidian distance measure, DTW always have better results than Euclidian distance measure whit *k*-means clustering algorithm.
- Euclidian distance measure perform better when used with conventional clustering algorithms (AGNES, *k*-means), rather than consensus clustering algorithm.
- *k*-means (both as conventional clustering algorithm and final clustering algorithm for consensus clustering) performs better for finding true clusters when compared to the hierarchical clustering.
- Euclidian distance measure, for all cases, failed to find the 4^{th} cluster when used with AGNES algorithm.
- As expected, when *k* increases, the number of clusters detected truly also increases and the clustering errors decreases (see Figure 3.3). When DTW window size increases, the number of clusters detected truly increases and the clustering errors either increases or stays the same (see Figure 3.3).

Table 3.6 Generation Times (Sec) for Similarity Matrices - Simulated Dataset -2

	Generation Times (Sec)
dtw window 1	391.1945
dtw window 2	441.8830
dtw window 3	499.8164
dtw window 4	599.4900
dtw window 5	740.1800
Euclidian Distance	0.2181

Table 3.7 Run Times (Sec) for Clustering Algor. - Simulated Dataset -2 (k=4)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.3461	9.0636	1588.4601	1595.6962
dtw window 2	0.0229	9.5095	1612.2358	1619.2281
dtw window 3	0.0167	9.5828	1591.2608	1599.8411
dtw window 4	0.0989	12.1372	1606.2564	1613.2364
dtw window 5	0.0911	10.7992	1620.2566	1627.9906
Euclidian Distance	0.0186	12.1208	1594.2668	1603.2860

Table 3.8 Errors and Clusters for Clustering Algor. - Simulated Dataset -2 (k=4)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.4350	0.3700	0.3650	0.3600
	1,2,3	1,2,3	1,2,3	1,2,3
dtw window 2	0.3800	0.4100	0.4300	0.3900
	1,2,3	1,2,3,4	1,3,4	1,2,3,4
dtw window 3	0.4250	0.4500	0.3850	0.4100
	1,2,3	1,3,4	1,2,3,4	1,2,3,4
dtw window 4	0.4600	0.4800	0.5000	0.5100
	1,2,3	1,2,3,4	1,3,4	1,2,3,4
dtw window 5	0.5600	0.4800	0.5750	0.5600
	1,2,3	1,2,3	1,3,4	1,3,4
Euclidian Distance	0.4400	0.4350	0.5550	0.4900
	1,2,3	1,2,3	1,3,4	1,2,3,4

Table 3.9 Run Times (Sec) for Clustering Algor. - Simulated Dataset -2 (k=5)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0271	11.8210	1588.1776	1597.3620
dtw window 2	0.0160	13.7690	1612.2158	1623.0052
dtw window 3	0.0160	13.1753	1591.2185	1599.6934
dtw window 4	0.0884	18.4880	1606.2456	1615.4424
dtw window 5	0.0887	15.9433	1620.2240	1631.5009
Euclidian Distance	0.0180	14.4006	1594.2175	1605.3910

Table 3.10 Errors and Clusters for Clustering Algor. - Simulated Dataset -2 (k=5)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.3900	0.3200	0.2500	0.2300
	1,2,3	1,2,3,4	1,2,3	1,2,3,4
dtw window 2	0.3350	0.3400	0.2800	0.2900
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 3	0.3500	0.3650	0.3950	0.3650
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 4	0.4450	0.4400	0.4800	0.4400
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 5	0.5150	0.4650	0.4550	0.4850
	1,2,3,4	1,2,3,4	1,2,3	1,2,3,4
Euclidian Distance	0.3650	0.4250	0.4550	0.4750
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4

Table 3.11 Run Times (Sec) for Clustering Algor. - Simulated Dataset -2 (k=6)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0201	14.5091	1588.2101	1597.8628
dtw window 2	0.0167	15.5018	1612.2005	1622.9132
dtw window 3	0.0186	16.7971	1591.2324	1602.5481
dtw window 4	0.0881	17.0572	1606.1987	1618.5764
dtw window 5	0.0886	18.2975	1620.2120	1633.1699
Euclidian Distance	0.0184	15.9662	1594.2463	1605.4089

Table 3.12 Errors and Clusters for Clustering Algor. - Simulated Dataset -2 (k=6)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2950	0.2400	0.1750	0.1400
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 2	0.3350	0.3250	0.1950	0.2000
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 3	0.3500	0.3550	0.2150	0.2200
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 4	0.4150	0.4650	0.4200	0.4050
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 5	0.4750	0.4800	0.4650	0.4850
	1,2,3,4	1,2,3,4	1,2,3	1,2,3,4
Euclidian Distance	0.3100	0.3700	0.4600	0.4550
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4

Table 3.13 Run Times (Sec) for Clustering Algor. - Simulated Dataset -2 (k=7)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0881	17.5801	1588.1547	1600.0209
dtw window 2	0.0886	18.0287	1612.2263	1624.4970
dtw window 3	0.0891	18.6063	1591.1950	1603.5106
dtw window 4	0.0886	19.2363	1606.2396	1620.5226
dtw window 5	0.0912	18.7671	1620.1921	1637.3047
Euclidian Distance	0.0890	16.6778	1594.2226	1608.1250

Table 3.14 Errors and Clusters for Clustering Algor. - Simulated Dataset -2 (k=7)

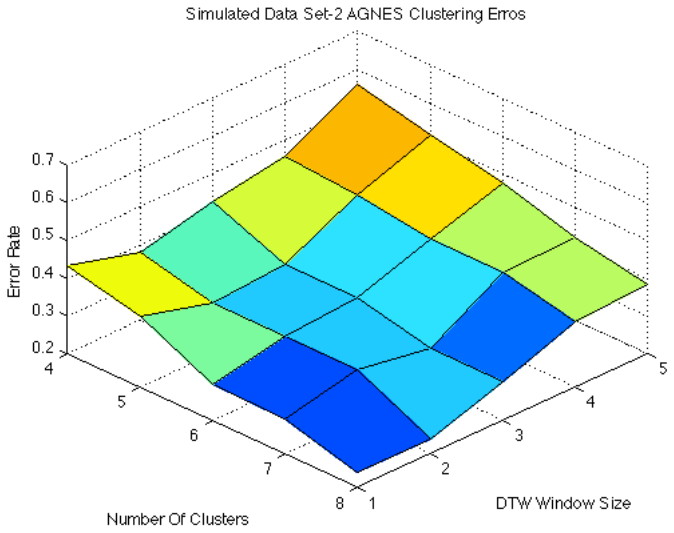
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2950	0.1050	0.0300	0.0300
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 2	0.3350	0.3450	0.1800	0.1950
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 3	0.3050	0.3400	0.2300	0.2350
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 4	0.4150	0.3900	0.3450	0.3000
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 5	0.4200	0.4150	0.4500	0.4600
	1,2,3,4	1,2,3,4	1,2,3	1,2,3,4
Euclidian Distance	0.2500	0.3100	0.4350	0.4300
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4

Table 3.15 Run Times (Sec) for Clustering Algor. - Simulated Dataset -2 (k=8)

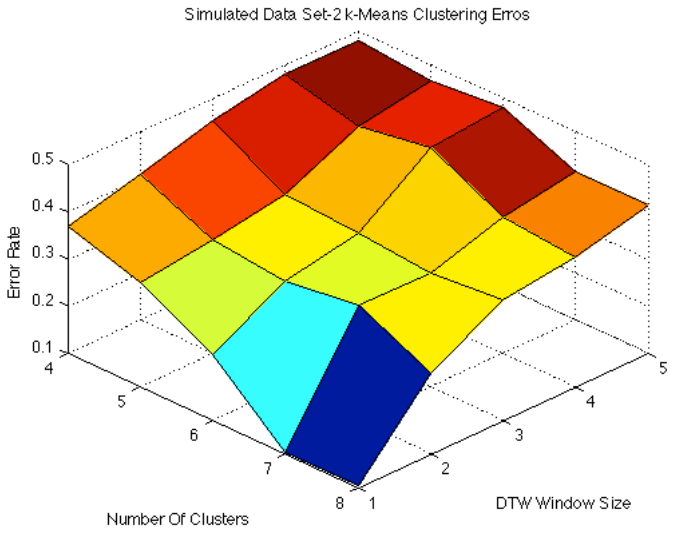
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0887	20.3169	1588.1639	1598.4081
dtw window 2	0.0886	19.1207	1612.2275	1624.5023
dtw window 3	0.0889	20.8761	1591.2329	1604.0647
dtw window 4	0.0887	22.4597	1606.2067	1621.3637
dtw window 5	0.0950	19.3942	1620.2126	1639.2593
Euclidian Distance	0.0886	19.0565	1594.2481	1606.8935

Table 3.16 Errors and Clusters for Clustering Algor. - Simulated Dataset -2 (k=8)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2400	0.1050	0.0200	0.0200
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 2	0.2400	0.2700	0.1650	0.1050
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 3	0.3050	0.3550	0.2300	0.2250
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 4	0.3750	0.3750	0.2500	0.3150
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
dtw window 5	0.3850	0.4150	0.3800	0.3900
	1,2,3,4	1,2,3,4	1,2,3,4	1,2,3,4
Euclidian Distance	0.2500	0.2400	0.4150	0.3950
	1,2,3	1,2,3,4	1,2,3,4	1,2,3,4

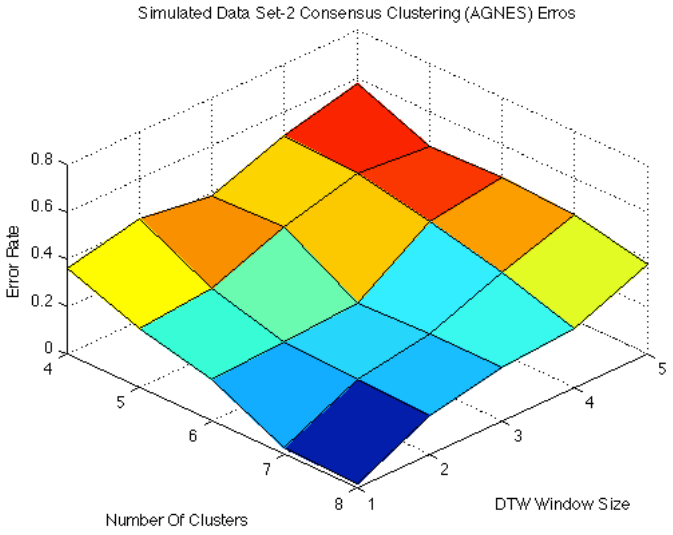


(a) Agglomerative Nesting (AGNES)

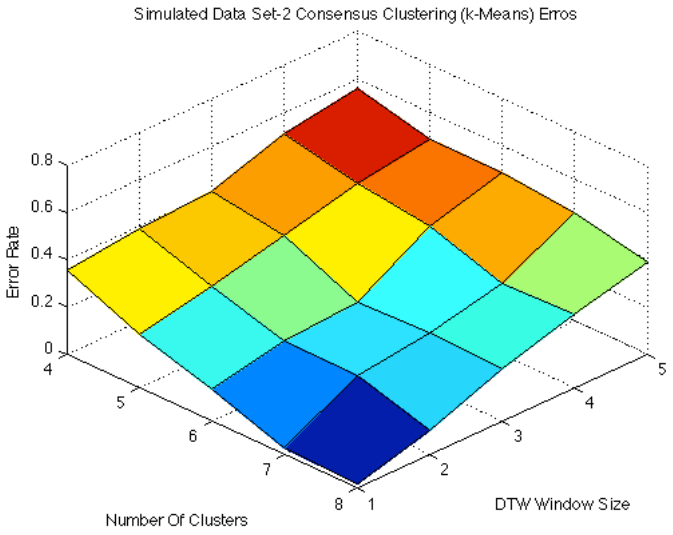


(b) *k*-means

Figure 3.3 Error Rates with Respect to Window Size and Number of Clusters



(c) Consensus Clustering (AGNES)



(d) Consensus Clustering (*k*-means)

Figure 3.3 (Continued) Error Rates with Respect to Window Size and Number of Clusters

3.3 Experimentation with Synthetic Control Dataset

Synthetic Control Dataset was obtained from UCI Machine Learning Repository [35]. This dataset contains 600 examples (six clusters each having 100 time series observations) of control charts synthetically generated by the process of Alcock et al. [36]. There are six different classes of control charts, namely "Normal," "Cyclic," "Increasing trend," "Decreasing trend," "Upward shift" and "Downward shift." Figure 3.4 shows ten examples from each class.

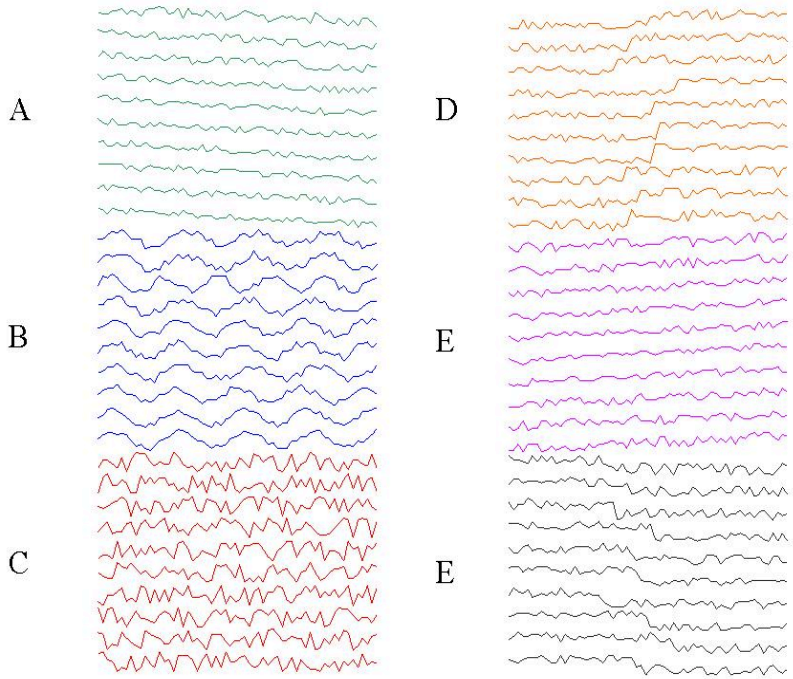


Figure 3.4 Synthetic Control Dataset

Run times for the generation of similarity matrices, run times for the clustering algorithms, error rates and the identified cluster labels from the real cluster labels for $k \in \{6,7,8,9,10\}$ was presented in Table 3.19, Table 3.20, Table 3.21, Table 3.22, Table 3.23, Table 3.24, Table 3.25, Table 3.26 and Table 3.27.

From the run time results for the generation of similarity matrices it can easily be seen that DTW is computationally very expensive since even with a time window of size one, the ratio between run times of DTW and Euclidian is nearly 500. It increases to nearly 900 for time window size equal to five.

Considering the clustering algorithm run times, clustering errors and cluster discovery results, following observations were made:

- Results show that consensus clustering algorithm with DTW (window size equal to one) as the distance measure perform better for all $k \in \{6,7,8,9,10\}$, when compared both with respect to the conventional clustering algorithms (AGNES, *k*-means) and with respect to the similarity distance measures (DTW window size 2-5 and Euclidian Distance Measure).
- As the final clustering algorithm *k*-means (with DTW window size equal to one) performs better for all the cases (between 1.2% and 15%).
- Consensus clustering is computationally very time consuming, and hence expensive.
- Leaving out the consensus clustering results, when DTW is compared with Euclidian distance measure, DTW always have better results than Euclidian distance measure whit *k*-means clustering algorithm.
- Euclidian distance measure perform better with conventional clustering algorithms (AGNES, *k*-means) in three cases out of five when compared to consensus clustering algorithm.
- As expected, when *k* increases, the number of clusters detected truly also increases and the clustering errors decreases (see Figure 3.5). When DTW window size increases, the number of clusters detected truly increases and the clustering errors either increases or stays the same (see Figure 3.5).

Table 3.17 Generation Times (Sec) for Similarity Matrices – Synth. Cont. Dataset

	Generation Times (Sec)
dtw window 1	1308.8000
dtw window 2	1402.8000
dtw window 3	1609.6000
dtw window 4	1941.0000
dtw window 5	2468.5000
Euclidian Distance	2.6839

Table 3.18 Run Times (Sec) for Clustering Algor. - Synthetic Con. Dataset (*k*=6)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.2571	119.0178	13396.2312	13477.6702
dtw window 2	0.2017	123.7814	13514.9075	13588.6206
dtw window 3	0.2391	153.5004	13619.9366	13692.0911
dtw window 4	0.8808	205.8110	13499.0220	13620.8210
dtw window 5	0.8576	276.4131	13378.2213	13488.5771
Euclidian Distance	0.2628	153.5539	13442.9459	13526.6285

Table 3.19 Errors and Clusters for Clustering Algor. - Synthetic Con. Dataset (k=6)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.3900	0.2167	0.2033	0.1767
	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.3717	0.3783	0.1950	0.1933
	1, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.4050	0.3833	0.1867	0.1867
	1, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3917	0.3767	0.3450	0.2750
	1, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.4067	0.3833	0.3267	0.3683
	1, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,3,4,5,6
Euclidian Distance	0.2283	0.2433	0.3317	0.2650
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5	1,2,3,4,6

Table 3.20 Run Times (Sec) for Clustering Algor. - Synthetic Con. Dataset (k=7)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2412	144.7325	13395.8776	13487.2521
dtw window 2	0.2042	143.9999	13514.9527	13647.5288
dtw window 3	0.2462	171.1087	13619.9096	13739.9510
dtw window 4	0.8063	300.2434	13498.9735	13633.2264
dtw window 5	0.8031	443.8419	13378.0446	13576.3871
Euclidian Distance	0.2398	251.2110	13442.9460	13547.6816

Table 3.21 Errors and Clusters for Clustering Algor. - Synthetic Con. Dataset (*k*=7)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.2233	0.2133	0.1383	0.1317
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.3717	0.2150	0.1867	0.1883
	1, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.4000	0.3750	0.1883	0.1850
	1, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3217	0.3667	0.2667	0.2750
	1, 2, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.3583	0.3750	0.3200	0.3283
	1, 2, 3, 4, 5, 6	1, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
Euclidian Distance	0.2283	0.2333	0.2583	0.2350
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6

Table 3.22 Run Times (Sec) for Clustering Algor. - Synthetic Con. Dataset (*k*=8)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.2084	172.9747	13395.8845	13493.3732
dtw window 2	0.2054	157.1848	13514.9753	13684.1874
dtw window 3	0.2393	199.5846	13619.8873	13778.6623
dtw window 4	0.8030	322.5613	13498.9606	13714.1831
dtw window 5	0.8109	413.2476	13378.0515	13552.9046
Euclidian Distance	0.2063	269.1474	13442.9395	13605.4401

Table 3.23 Errors and Clusters for Clustering Algor. - Synthetic Con. Dataset (k=8)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2233	0.2033	0.1917	0.1767
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.3717	0.2100	0.1817	0.1900
	1, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.3317	0.2117	0.1800	0.1800
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3217	0.2967	0.2083	0.2050
	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.3550	0.3150	0.3017	0.3167
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
Euclidian Distance	0.2283	0.2267	0.2267	0.2317
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6

Table 3.24 Run Times (Sec) for Clustering Algor. - Synthetic Con. Dataset (k=9)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.8086	214.3973	13396.0509	13536.2734
dtw window 2	0.8049	197.0057	13514.9142	13698.5777
dtw window 3	0.8019	246.2865	13619.8966	13784.3393
dtw window 4	0.8015	347.5116	13498.9528	13740.9422
dtw window 5	0.8059	356.7311	13378.0390	13622.9971
Euclidian Distance	0.8013	363.1410	13442.9469	13611.2687

Table 3.25 Errors and Clusters for Clustering Algor. - Synthetic Con. Dataset (k=9)

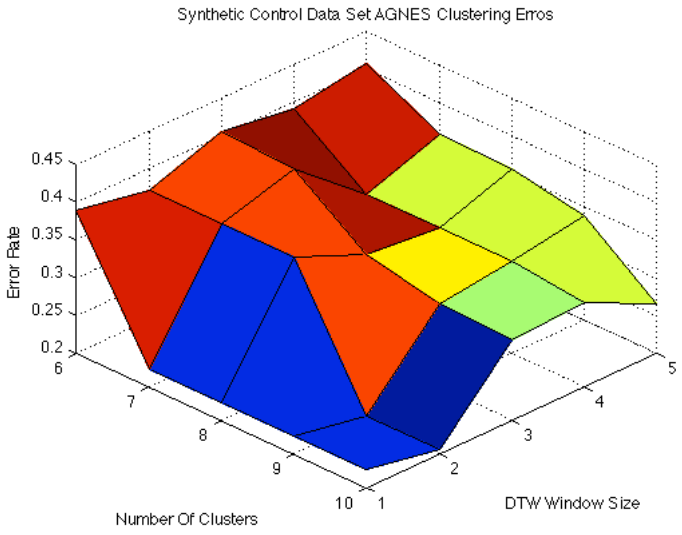
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2233	0.2100	0.1400	0.1383
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.2067	0.2117	0.1850	0.1433
	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.3100	0.2117	0.1817	0.1783
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3217	0.2267	0.2117	0.1883
	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.3383	0.2583	0.2933	0.2983
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
Euclidian Distance	0.2283	0.2317	0.2133	0.2217
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6

Table 3.26 Run Times (Sec) for Clustering Algor. - Synthetic Con. Dataset (k=10)

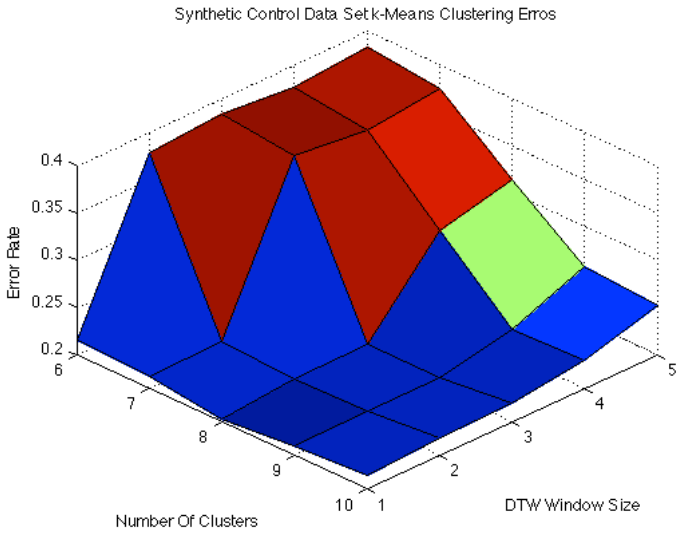
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.8101	245.9032	13396.0678	13560.4635
dtw window 2	0.8059	224.3224	13515.2466	13733.3051
dtw window 3	0.8041	257.1076	13619.9774	13812.7344
dtw window 4	0.8097	394.0911	13499.0686	13730.5603
dtw window 5	0.8024	320.1215	13378.1050	13656.1541
Euclidian Distance	0.8050	456.9493	13443.0447	13588.9502

Table 3.27 Errors and Clusters for Clustering Algor. – Synt. Con. Dataset (*k*=10)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.2233	0.2150	0.1400	0.1383
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.2067	0.2183	0.1383	0.1467
	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.3067	0.2200	0.1600	0.1533
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3117	0.2300	0.1983	0.1950
	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.2650	0.2533	0.2633	0.2650
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
Euclidian Distance	0.2283	0.2317	0.2317	0.2050
	1, 2, 3, 4, 5, 6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6

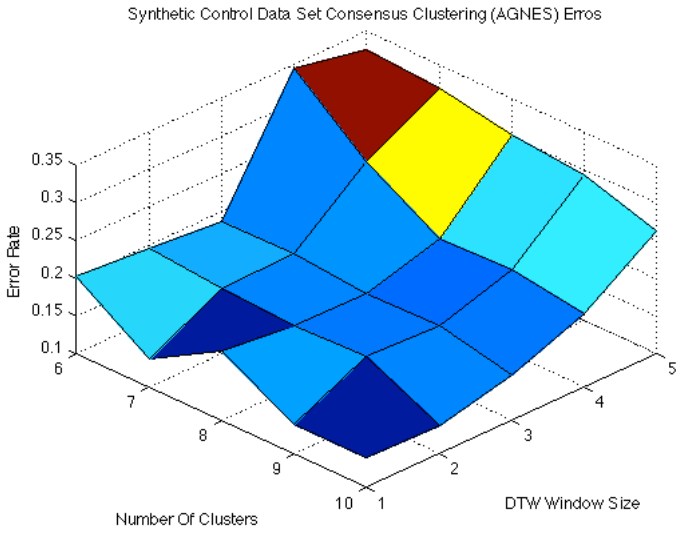


(a) Agglomerative Nesting (AGNES)

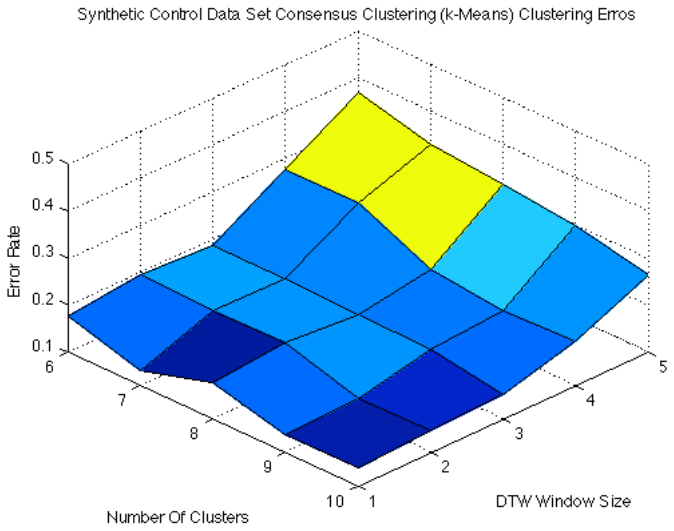


(b) *k*-means

Figure 3.5 Error Rates with Respect to Window Size and Number of Clusters



(c) Consensus Clustering (AGNES)



(d) Consensus Clustering (*k*-means)

Figure 3.5 (Continued) Error Rates with Respect to Window Size and Number of Clusters

3.4 Experimentation with Daily and Sports Activities Dataset

Daily and Sports Activities Dataset was obtained from UCI Machine Learning Repository [35]. The dataset was created by Altun et al. [37]. In this dataset there are 19 activities performed by eight subjects (four female, four male, between the ages 20 and 30). The subjects were asked to perform the activities in their own style and were not restricted on how the activities should be performed. For this reason, there are inter-subject variations in the speeds and amplitudes of some activities. There were five measurement points (torso, right arm, left arm, right leg, left leg) and nine different sensors at each point (*x, y, z* accelerometers, *x, y, z* gyroscopes, *x, y, z* magnetometers).

Altun et al. [37] mentioned that the best identifiers for the human activities are leg accelerometers and magnetometers. So in our study we had only used the data obtained from Right Leg *z*-accelerometer and Right Leg *z*-magnetometer. We have also only included the following 7 activities in our study: standing, ascending and descending stairs, walking in a parking lot, running on a treadmill with a speed of 8 km/h, cycling on an exercise bike in horizontal positions, and jumping.

3.4.1 Right Leg Accelerometer Data

Data for Right Leg Accelerometer is presented in Figure 3.6. All the seven activities are presented for a single performer.

When the clustering algorithm run times, clustering errors and cluster discovery results are considered, following observations were made:

- *k*-means clustering algorithm with DTW (window size equal to one) as the distance measure performs better for all $k \in \{7,8,9,10,11\}$, when compared with other algorithms (AGNES, Consensus Clustering and with respect to the similarity distance measures (DTW window size 2-5 and Euclidian Distance Measure).
- *k*-means clustering algorithm with DTW (window size equal to one) as the distance measure, identify all the true clusters, while Euclidian distance fail to identify all the true clusters.
- When DTW is compared with Euclidian distance measure, DTW have better results than Euclidian distance measure regardless of the clustering algorithm used.
- *k*-means (both as conventional clustering algorithm and final clustering algorithm for consensus clustering) performs better for finding true clusters when compared to hierarchical clustering.
- Euclidian distance measure, for all cases, failed to find the 2^{nd} , 3^{rd} and 6^{th} clusters when used with AGNES algorithm and 2^{nd} and 6^{th} clusters when used with *k*-means algorithm.
- As expected, when k increases, the number of clusters detected truly also increases and the clustering errors decreases (see Figure 3.7).

- When DTW window size increases, the number of clusters detected truly increases and the clustering errors either increases or stays the same (see Figure 3.7).

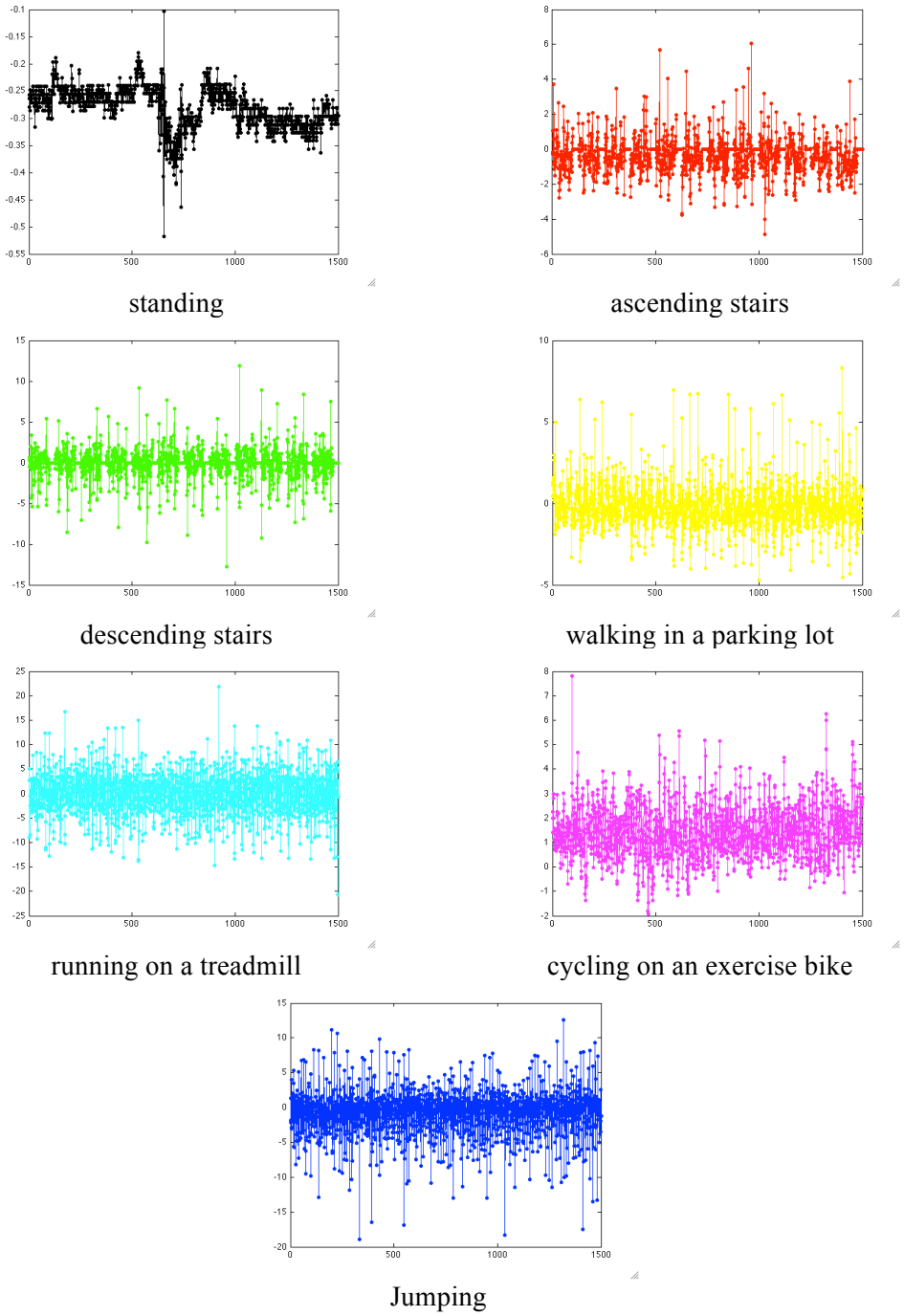


Figure 3.6 Daily and Sports Activities Dataset – Right Leg Accelerometer

Table 3.28 Generation Times (Sec) for Similarity Matrices - Right Leg Accelerometer

	Generation Times (Sec)
dtw window 1	6714.5609
dtw window 2	7659.0000
dtw window 3	8717.1000
dtw window 4	10355.0000
dtw window 5	12625.0000
Euclidian Distance	0.1838

Table 3.29 Run Times (Sec) for Clustering Algor. - Right Leg Acc. (*k*=7)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.0109	4.1308	126.2235	130.4206
dtw window 2	0.0207	3.4265	125.6267	128.9476
dtw window 3	0.0198	3.7485	124.8268	127.6829
dtw window 4	0.0198	3.5619	124.2296	127.5414
dtw window 5	0.0202	4.1236	126.0298	129.3655
Euclidian Distance	0.0097	4.2221	127.6304	130.5787

Table 3.30 Errors and Clusters for Clustering Algor. - Right Leg Acc. (*k*=7)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.6071	0.3571	0.4286	0.4286
	1,5,6,7	1,2,3,5,6,7	1,3,5,6,7	1,2,3,5,6,7
dtw window 2	0.4643	0.4107	0.4464	0.4464
	1,2,5,6,7	1,2,3,5,6,7	1,2,5,6,7	1,2,3,5,6,7
dtw window 3	0.4464	0.4821	0.5179	0.5000
	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7
dtw window 4	0.5357	0.4643	0.5000	0.5000
	1,2,5,6	1,2,5,6,7	1,2,3,5,6,7	1,2,5,6,7
dtw window 5	0.5536	0.5536	0.5714	0.5893
	1,4,5,6	1,2,3,5,6	1,2,4,5	1,2,5
Euclidian Distance	0.6964	0.5357	0.6786	0.5536
	1,5,7	1,3,4,5,7	2,4,5,6,7	1,2,3,4,5,7

Table 3.31 Run Times (Sec) for Clustering Algor. - Right Leg Acc. (*k*=8)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.0081	4.6239	126.2424	129.4190
dtw window 2	0.0213	3.8210	125.6657	129.0018
dtw window 3	0.0217	4.0602	124.8719	127.6840
dtw window 4	0.0215	3.9075	124.2659	127.3343
dtw window 5	0.0211	5.0715	126.0647	129.3947
Euclidian Distance	0.0089	4.4435	127.6868	130.3328

Table 3.32 Errors and Clusters for Clustering Algor. - Right Leg Acc. (*k*=8)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.4643	0.3571	0.4286	0.4464
	1,2,5,6,7	1,2,3,5,6,7	1,3,5,6,7	1,2,3,5,6,7
dtw window 2	0.4643	0.4107	0.4464	0.4464
	1,2,5,6,7	1,2,3,5,6,7	1,2,5,6,7	1,2,3,5,6,7
dtw window 3	0.4464	0.4643	0.4464	0.5000
	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7	1,2,3,5,6,7
dtw window 4	0.5357	0.4464	0.4464	0.4464
	1,2,5,6	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7
dtw window 5	0.5536	0.4643	0.5714	0.5714
	1,4,5,6	1,2,5,6,7	1,2,5,6	1,2,5,6
Euclidian Distance	0.6786	0.4821	0.6607	0.4821
	1,4,5,7	1,3,4,5,7	2,4,5,6,7	1,3,4,5,7

Table 3.33 Run Times (Sec) for Clustering Algor. - Right Leg Acc. (*k*=9)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.0088	4.9190	126.2432	129.5848
dtw window 2	0.0214	4.0726	125.6660	128.8466
dtw window 3	0.0213	4.2046	124.8636	128.0212
dtw window 4	0.0211	4.3267	124.2655	127.8415
dtw window 5	0.0212	5.8849	126.0613	129.5155
Euclidian Distance	0.0089	5.1156	127.7083	130.6620

Table 3.34 Errors and Clusters for Clustering Algor. - Right Leg Acc. (*k*=9)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.4643	0.3571	0.4107	0.3750
	1,2,5,6,7	1,2,3,5,6,7	1,3,4,5,6,7	1,2,3,5,6,7
dtw window 2	0.4643	0.4107	0.4107	0.3929
	1,2,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 3	0.4464	0.4464	0.4643	0.4643
	1,2,5,6,7	1,2,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 4	0.5357	0.4107	0.4643	0.4821
	1,2,5,6	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7
dtw window 5	0.5536	0.4464	0.5357	0.5357
	1,4,5,6	1,2,5,6,7	1,2,5,6,7	1,2,4,5,6
Euclidian Distance	0.6786	0.4821	0.6429	0.5000
	1,4,5,7	1,3,4,5,7	2,4,5,6,7	1,3,5,6,7

Table 3.35 Run Times (Sec) for Clustering Algor. - Right Leg Acc. (k=10)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0090	5.0064	126.2475	129.8662
dtw window 2	0.0220	4.2638	125.6657	129.1115
dtw window 3	0.0214	4.4556	124.8648	128.7936
dtw window 4	0.0217	4.7448	124.2680	127.7549
dtw window 5	0.0224	6.4064	126.0581	129.6015
Euclidian Distance	0.0101	5.7489	127.6635	131.6771

Table 3.36 Errors and Clusters for Clustering Algor. - Right Leg Acc. (k=10)

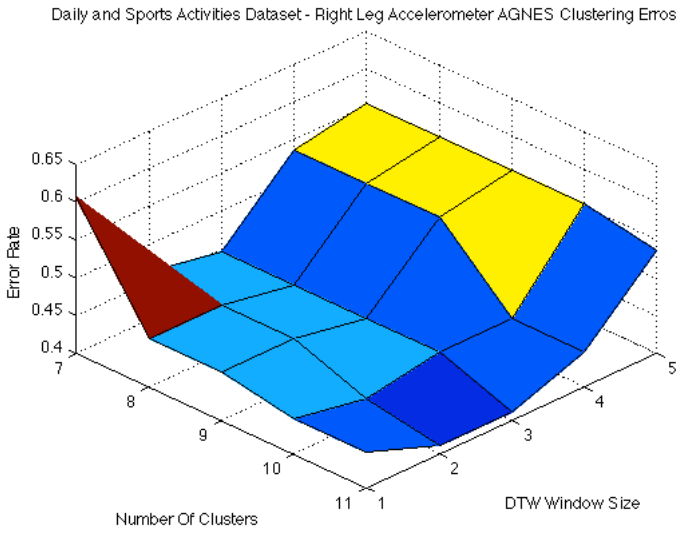
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.4464	0.3571	0.4286	0.3571
	1,2,4,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 2	0.4286	0.4107	0.4107	0.3929
	1,2,5,6,7	1,2,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 3	0.4464	0.4464	0.4643	0.4107
	1,2,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 4	0.4464	0.4107	0.4643	0.4821
	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7
dtw window 5	0.5536	0.4464	0.5000	0.5179
	1,4,5,6	1,2,5,6,7	1,2,4,5,6,7	1,2,4,5,7
Euclidian Distance	0.6607	0.5179	0.6429	0.5179
	1,4,5,7	1,3,4,5,7	2,4,5,6,7	1,3,5,6,7

Table 3.37 Run Times (Sec) for Clustering Algor. - Right Leg Acc. (k=11)

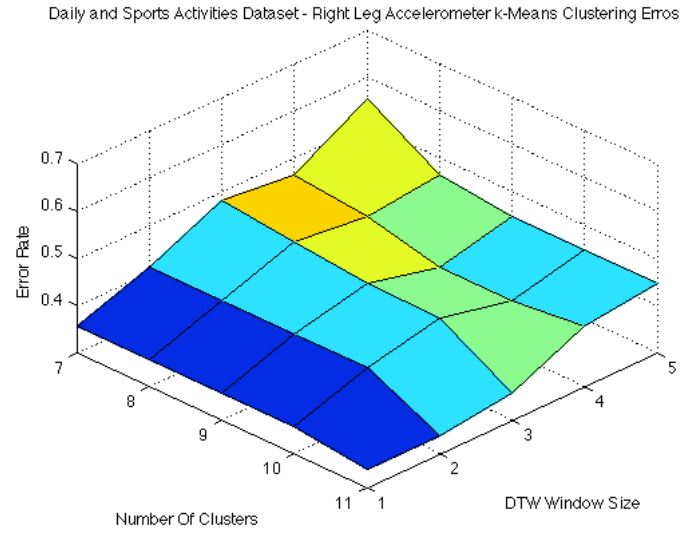
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0204	5.1082	126.4409	129.9592
dtw window 2	0.0219	4.4792	125.6597	129.2141
dtw window 3	0.0216	4.5380	124.8597	128.4587
dtw window 4	0.0214	5.0576	124.2564	127.8032
dtw window 5	0.0222	6.3390	126.0483	129.8735
Euclidian Distance	0.0226	6.1510	127.6548	131.3690

Table 3.38 Errors and Clusters for Clustering Algor. - Right Leg Acc. (k=11)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.4464	0.3393	0.4286	0.3571
	1,2,4,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 2	0.4107	0.3393	0.3929	0.3929
	1,2,4,5,6,7	1,2,4,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 3	0.4107	0.3571	0.4107	0.4464
	1,2,5,6,7	1,2,3,4,5,6,7	1,2,3,5,6,7	1,2,3,5,6,7
dtw window 4	0.4464	0.4286	0.4643	0.4643
	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7	1,2,5,6,7
dtw window 5	0.5357	0.4464	0.5000	0.4821
	1,2,4,5,6	1,2,5,6,7	1,2,4,5,6,7	1,2,3,5,7
Euclidian Distance	0.6429	0.4821	0.6071	0.5000
	1,4,5,7	1,3,4,5,7	2,4,5,6,7	1,3,5,6,7

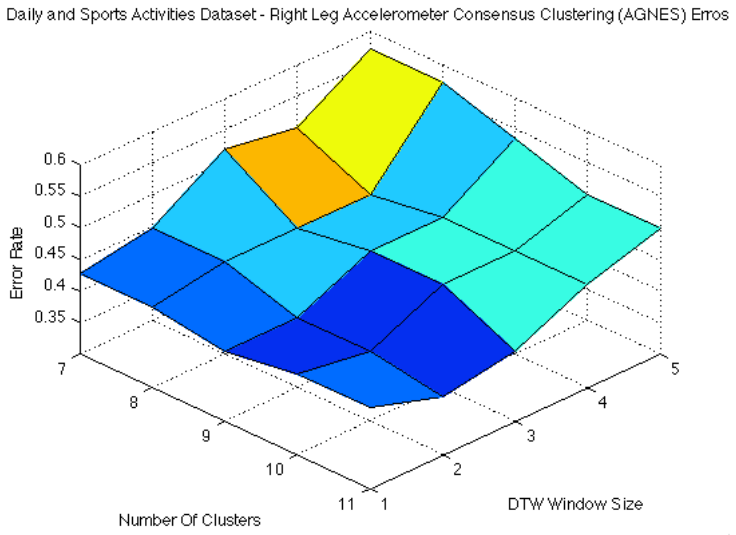


(a) Agglomerative Nesting (AGNES)

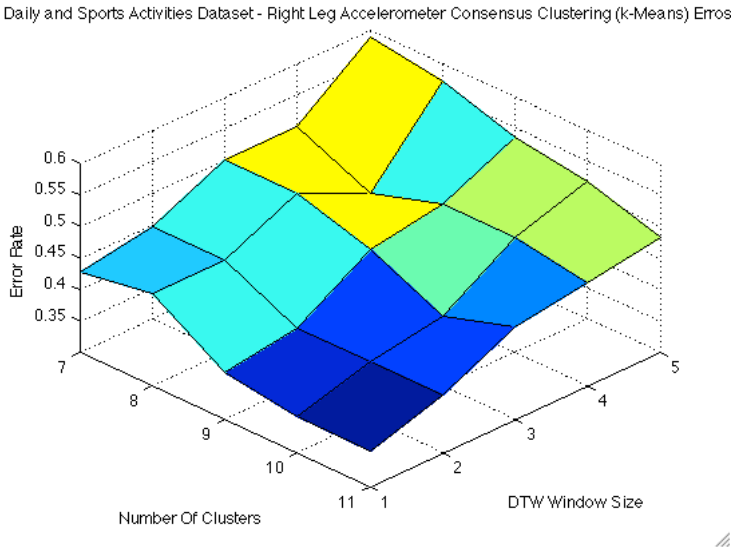


(b) *k*-means

Figure 3.7 Error Rates with Respect to Window Size and Number of Clusters



(c) Consensus Clustering (AGNES)



(d) Consensus Clustering (*k*-means)

Figure 3.7 (Continued) Error Rates with Respect to Window Size and Number of Clusters

3.4.2 Right Leg Magnetometer Data

Data for Right Leg Magnetometer is presented in Figure 3.8. All the seven activities are presented for a single performer.

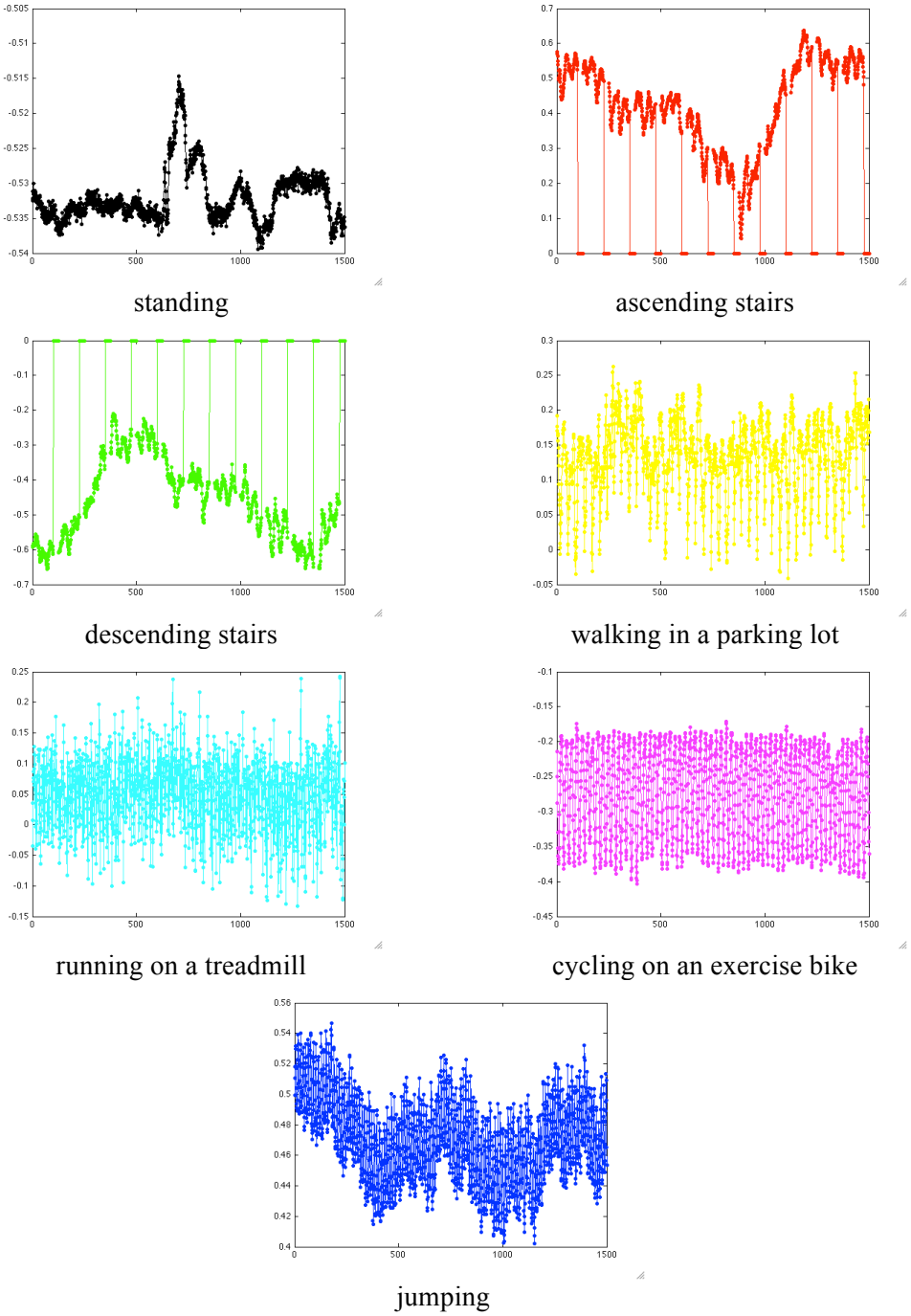


Figure 3.8 Daily and Sports Activities Dataset – Right Leg Magnetometer

Following observations were made, when the clustering algorithm run times, clustering errors and the clusters that were truly identified were considered:

- Results show that consensus clustering algorithm with DTW (window size equal to one) as the distance measure perform better for $k \in \{9,10,11\}$, while Euclidian Distance with AGNES perform better for $k \in \{7,8\}$, when compared with other algorithms and similarity distance measures.
- Results show that AGNES clustering algorithm discover all the clusters while other clustering algorithms failed to discover all the clusters.
- Within consensus clustering results, DTW have better results than Euclidian distance measure.
- *k*-means clustering algorithm and Consensus Clustering algorithm (AGNES) for all cases, failed to find the 7^{th} cluster, regardless of the distance measure used.
- As expected, when *k* increases, the number of clusters detected truly also increases and the clustering errors decreases (see Figure 3.9).
- When DTW window size increases, the number of clusters detected truly increases and the clustering errors either increases or stays the same (see Figure 3.9).

Table 3.39 Generation Times (Sec) for Similarity Matrices - Right Leg Magnetometer

	Generation Times (Sec)
dtw window 1	6776.0248
dtw window 2	7465.1000
dtw window 3	8630.2000
dtw window 4	10449.0000
dtw window 5	12921.0000
Euclidian Distance	0.2048

Table 3.40 Run Times (Sec) for Clustering Algor. - Right Leg Mag. (k=7)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.2133	3.2085	124.8416	127.4041
dtw window 2	0.0222	3.0854	123.8244	126.2854
dtw window 3	0.0221	3.1448	157.8456	160.3321
dtw window 4	0.0221	3.0164	123.8547	126.3899
dtw window 5	0.0220	3.1424	124.4652	127.0785
Euclidian Distance	0.0222	2.9026	124.4638	126.9531

Table 3.41 Errors and Clusters for Clustering Algor. - Right Leg Mag. (k=7)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.4464	0.4643	0.4286	0.4464
	2,3,4,5,6,7	1,2,3,4,6	2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.4643	0.4643	0.4286	0.4464
	1,2,3,4,6	1,2,3,4,6	2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.4464	0.4464	0.4286	0.4286
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	2,3,4,5,6
dtw window 4	0.4464	0.4464	0.4286	0.4286
	2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	2,3,4,5,6
dtw window 5	0.4286	0.4464	0.4286	0.4286
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	2,3,4,5,6
Euclidian Distance	0.4107	0.4464	0.4464	0.4464
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	2,3,4,5,6

Table 3.42 Run Times (Sec) for Clustering Algor. - Right Leg Mag. (*k*=8)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.0151	3.1987	124.8452	127.4962
dtw window 2	0.0222	3.1743	123.8267	126.4271
dtw window 3	0.0220	3.2909	157.8512	160.4235
dtw window 4	0.0221	3.1460	123.8610	126.4211
dtw window 5	0.0219	3.1532	124.4586	127.0770
Euclidian Distance	0.0220	3.1081	124.4641	126.9716

Table 3.43 Errors and Clusters for Clustering Algor. - Right Leg Mag. (*k*=8)

	Agglomerative Nesting (AGNES)	*k*-means	Consensus Clustering (AGNES)	Consensus Clustering (*k*-means)
dtw window 1	0.4107	0.4107	0.4286	0.4286
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.3929	0.4107	0.4286	0.4286
	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.4107	0.4107	0.4286	0.3929
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.4107	0.4107	0.4286	0.3929
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.3929	0.4107	0.4286	0.4286
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	2,3,4,5,6
Euclidian Distance	0.4107	0.4286	0.4286	0.4286
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	1,2,3,4,5,6

Table 3.44 Run Times (Sec) for Clustering Algor. - Right Leg Mag. (k=9)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0205	3.2591	124.8459	127.7219
dtw window 2	0.0251	3.3551	123.8240	126.6241
dtw window 3	0.0277	3.3646	157.8493	160.4894
dtw window 4	0.0240	3.2791	123.8613	126.5741
dtw window 5	0.0236	3.3805	124.4586	127.2090
Euclidian Distance	0.0219	3.2346	124.4670	127.1211

Table 3.45 Errors and Clusters for Clustering Algor. - Right Leg Mag. (k=9)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.3929	0.3929	0.3929	0.4107
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 2	0.3750	0.3929	0.3929	0.4107
	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.3929	0.3929	0.3929	0.4107
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3929	0.3929	0.3929	0.4107
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.3750	0.3929	0.3929	0.3929
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
Euclidian Distance	0.3929	0.4107	0.4286	0.3929
	1,2,3,4,5,6,7	1,2,3,4,5,6	2,3,4,5,6	1,2,3,4,5,6

Table 3.46 Run Times (Sec) for Clustering Algor. - Right Leg Mag. (k=10)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0208	3.4684	124.8425	127.9803
dtw window 2	0.0222	3.4706	123.8243	126.8878
dtw window 3	0.0221	3.5385	157.8537	160.8704
dtw window 4	0.0219	3.4613	123.8632	126.7882
dtw window 5	0.0220	3.5158	124.4619	127.3015
Euclidian Distance	0.0217	3.3759	124.4672	127.2440

Table 3.47 Errors and Clusters for Clustering Algor. - Right Leg Mag. (k=10)

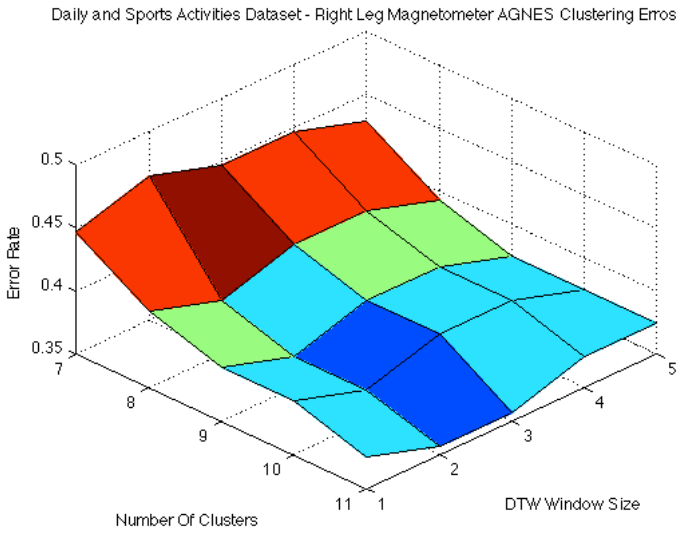
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.3929	0.3929	0.3929	0.3571
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
dtw window 2	0.3750	0.3929	0.3929	0.3929
	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 3	0.3929	0.4107	0.3929	0.3571
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 4	0.3929	0.3929	0.3929	0.3571
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
dtw window 5	0.3750	0.4107	0.3929	0.3571
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
Euclidian Distance	0.3750	0.3929	0.4286	0.3929
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6

Table 3.48 Run Times (Sec) for Clustering Algor. - Right Leg Mag. (k=11)

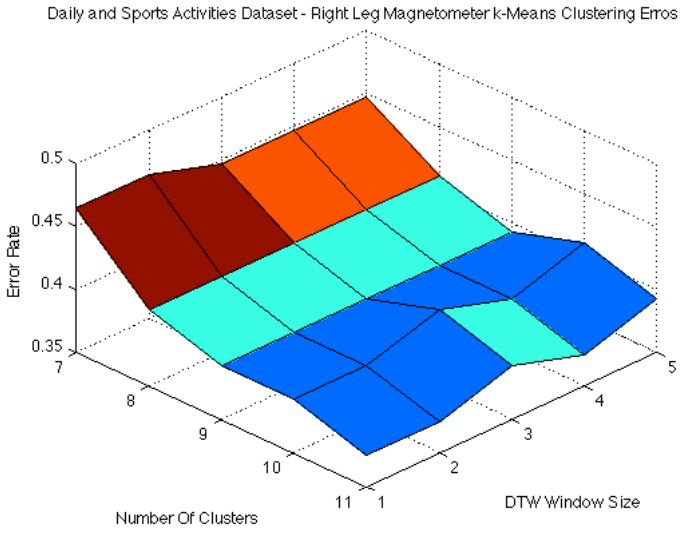
	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.0204	3.6876	124.8402	128.0768
dtw window 2	0.0219	3.6766	123.8236	127.3289
dtw window 3	0.0219	3.8103	157.8535	160.9896
dtw window 4	0.0228	3.6252	123.8652	127.0269
dtw window 5	0.0223	3.7256	124.4688	127.6225
Euclidian Distance	0.0217	3.6469	124.4680	127.6618

Table 3.49 Errors and Clusters for Clustering Algor. - Right Leg Mag. (k=11)

	Agglomerative Nesting (AGNES)	k-means	Consensus Clustering (AGNES)	Consensus Clustering (k-means)
dtw window 1	0.3750	0.3750	0.3750	0.3393
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
dtw window 2	0.3571	0.3750	0.3750	0.3393
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
dtw window 3	0.3571	0.3929	0.3750	0.3393
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
dtw window 4	0.3750	0.3750	0.3750	0.3750
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6
dtw window 5	0.3750	0.3929	0.3750	0.3393
	1,2,3,4,5,6,7	1,2,3,4,5,6	1,2,3,4,5,6	1,2,3,4,5,6,7
Euclidian Distance	0.3571	0.3750	0.3929	0.3571
	1,2,3,4,5,6,7	1,2,3,4,5,7	1,2,3,4,5,6,7	1,2,3,4,5,6

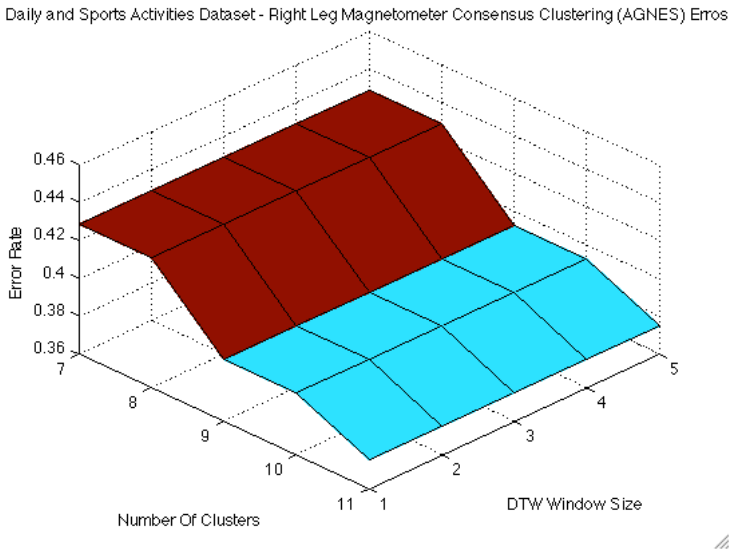


(a) Agglomerative Nesting (AGNES)

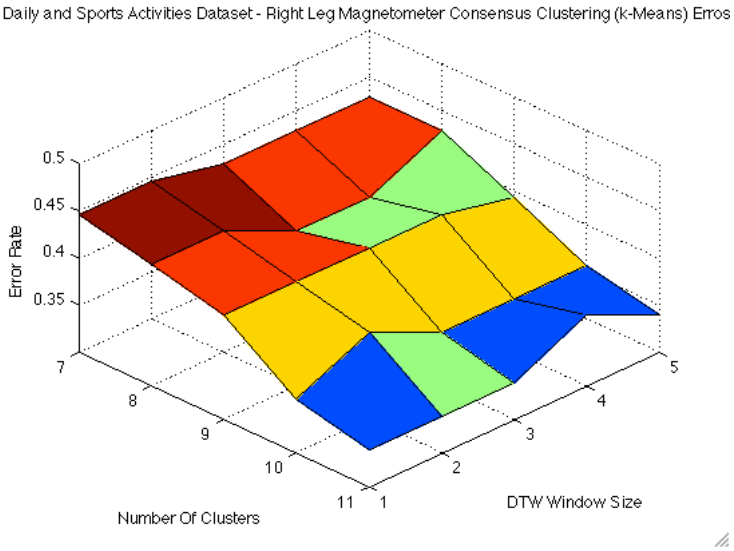


(b) *k*-means

Figure 3.9 Error Rates with Respect to Window Size and Number of Clusters



(c) Consensus Clustering (AGNES)



(d) Consensus Clustering (*k*-means)

Figure 3.9 (Continued) Error Rates with Respect to Window Size and Number of Clusters

3.5 General Discussion of The Results

For most of the cases we experimented with, DTW provides better results than the Euclidian Distance measure. Mostly usage of window size equal to one provides better results than the usage of other window sizes. However, consensus clustering with DTW is computationally very expensive when compared to the usage Euclidian Distance and the conventional clustering algorithms. Thus this feature makes it harder to work with large datasets having too many time series samples and data points. Also in some cases the performance difference is around 1%, which makes it unnecessary to use both DTW and Consensus Clustering simultaneously.

But in all the cases we experimented with, when used with consensus clustering DTW performs better than Euclidian Distance measure, both regarding the errors and cluster discoveries. In addition, generally *k*-means (both as conventional clustering algorithm and final clustering algorithm for consensus clustering) is better in performance (errors and the number of clusters detected truly) compared to hierarchical clustering when DTW is used as a distance measure.

Also, dataset we have created (Simulated Data-2) backed up our initial expectation that DTW would perform better with data having phase shifts. This point is open for further experimentation with simulated datasets as the phase shift properties of real datasets are hard to observer if it wasn't considered in the data collection and mentioned in the dataset description.

Finally it should be mentioned that all this conclusions are dependent on the dataset's properties and need to be experimented with more data in detail in order to be expressed firmly.

CHAPTER 4

DETERMINING TURKEY'S CLIMATE REGIONS USING CONSENSUS CLUSTERING

4.1 Dataset Description

The dataset that was used in order to define the Turkey's Climate Regions, is Turkey's long term meteorological data, which was recorded at 244 stations of the Turkish State Meteorological Service (TSMS) over the period 1950–2010. There are 13 variables in the dataset. We have only used nine variables. Those variables are,

- monthly mean air temperatures,
- monthly minimum air temperatures,
- monthly maximum air temperatures,
- monthly minimums of mean temperatures,
- monthly maximums of mean temperatures,
- monthly averages of minimum temperatures,
- monthly averages of maximum temperatures,
- monthly precipitation totals (in millimeters),
- monthly relative humidity (in per cent).

In order to create a complete dataset, Iyigun et al. [38], Aslan et al. [39] and Yozgatligil et al. [40] applied data preprocessing and minimized the number of missing values. The details of the dataset and the preprocessing performed was discussed in detail in References [38], [39], [40], [41]. In this study the data is also standardized prior to its use, to prevent domination of any variable with large measurement values.

4.2 Clustering Analysis Results

We have performed multivariate clustering analysis for $k \in \{7,8,9,10,11,12\}$ using DTW with window size equal to one as a similarity measure and consensus clustering with multivariate strategies described in Section 2.5.

For obtaining consensus clustering merged matrices we have utilized the R Package that was developed by Simpson [34]. For all datasets we have used Agglomerative Nesting (Hierarchical Clustering), Partitioning Around Medoids, Divisive Analysis

Clustering and *k*-means as different clustering algorithms within the consensus clustering algorithm. All four clustering algorithms were equally weighted for calculation of the final consensus clustering merged matrix. Different parameter sets used for different clustering algorithms are presented in Table 4.1. In order to obtain the clustering labels, for final clustering we solved the merged matrices obtained with both Agglomerative Nesting (Hierarchical Clustering) and *k*-means.

Table 4.1 Parameter Sets Used for Different Clustering Algorithms

Algorithm Definition	Distance Measure	Method	Other Parameters
Agglomerative Nesting (Hierarchical Clustering)	Euclidean	Average Linkage	R defaults
Partitioning Around Medoids	Euclidean	-	R defaults
Divisive Analysis Clustering	Euclidean	-	R defaults
k-means	-	Hartigan-Wong	R defaults

The results are presented in Figure 4.1, Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5 and Figure 4.6 for $k \in \{7,8,9,10,11,12\}$ respectively.

Results show us that up to $k = 10$ Agglomerative Nesting as a final clustering algorithm does not provide reliable results as it forms singleton clusters (clusters with only one element), with the exception of Combined Merged Matrices $k = 7$ and $k = 10$. Also for $k = 11$ and $k = 12$, Agglomerative Nesting create clusters with elements less than or equal to five, for Combined Merged Matrices. Even though final clustering algorithm have an effect on shaping the final clustering results, when results are considered it is hard to say whether consensus clustering methodology for combining multivariate data or final clustering algorithm has a greater effect on the clustering results.

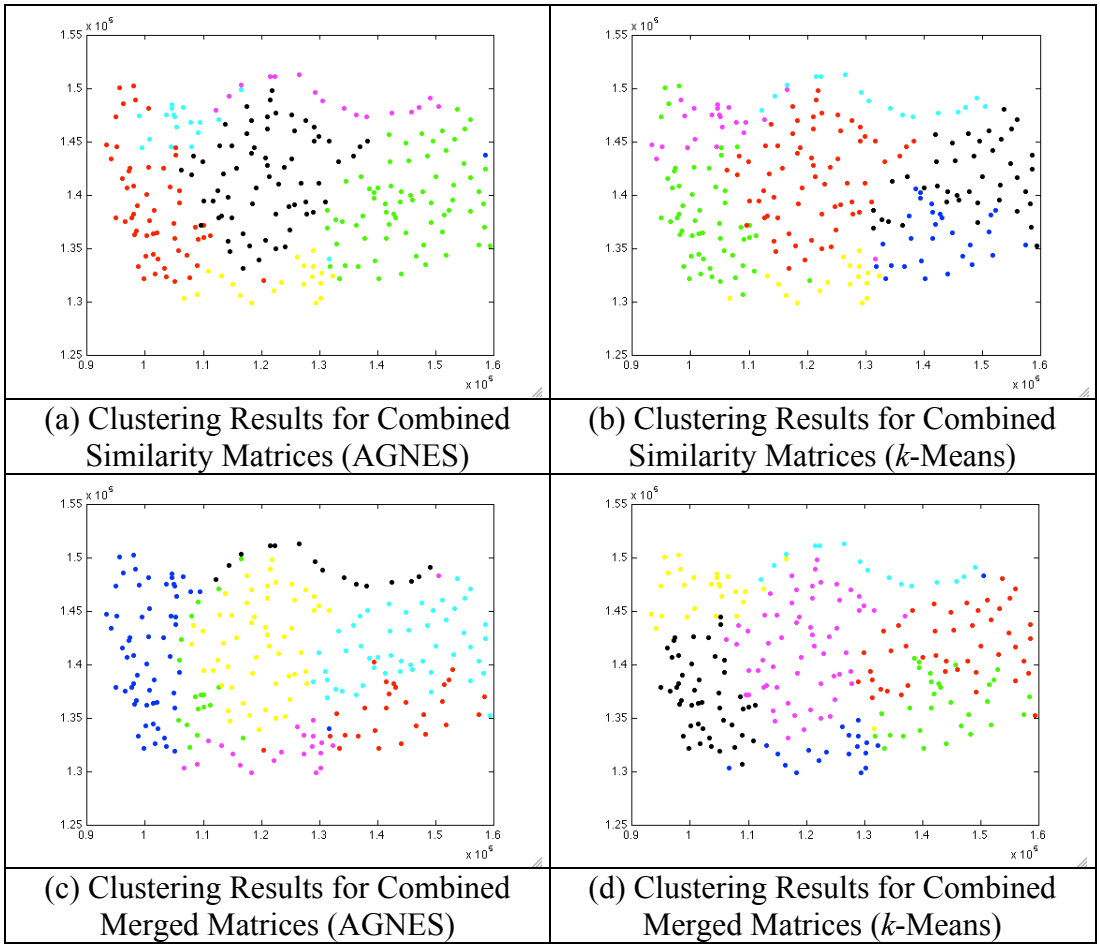
When Combined Similarity Matrices is uses as a methodology for combining multivariate data it is observed that, with the use of similarity matrices both AGNES and *k*-means algorithms create a cluster for the Marmara Region. Mediterranean Region, Central Anatolian Region and Black Sea Region have fairly stable cluster definitions having a consensus among the methods used with the increasing cluster numbers. Yet the cluster definitions for East Anatolian Region, Marmara Region, Aegean Region and Western Anatolia Region change a lot between clustering methods use when there is an increase in the cluster numbers.

The percentage differences between the results obtained by Iyigun et al. [38] and this study is as follows:

- Clustering Results for Combined Similarity Matrices (AGNES): 29.51 %
- Clustering Results for Combined Similarity Matrices (*k*-Means): 34.02 %
- Clustering Results for Combined Merged Matrices (AGNES): 36.48 %
- Clustering Results for Combined Merged Matrices (*k*-Means); 30.74 %

When results are compared with the results obtained by Iyigun et al. [38], result for "Clustering Results for Combined Similarity Matrices (AGNES)" and "Combined Merged Matrices (*k*-Means)" more coincide with the findings of Iyigun et al. for $k = 12$. Our proposed algorithms created two distinct Mediterranean regions which can be named as Western and Eastern Coastal Mediterranean Regions, whereas Iyigun et al.s' study suggested that there is only a single Coastal Mediterranean Region. Also with our algorithm Eastern Anatolia is divided in to 3 clusters. Yet Iyigun et al.s' study show that there are 4 clusters for Eastern Anatolia.

However it is not possible to tell which algorithm performed better as there is no original cluster labels. So there is a need to expert judgment in order to conclude which clustering is better regarding the real regional definition.



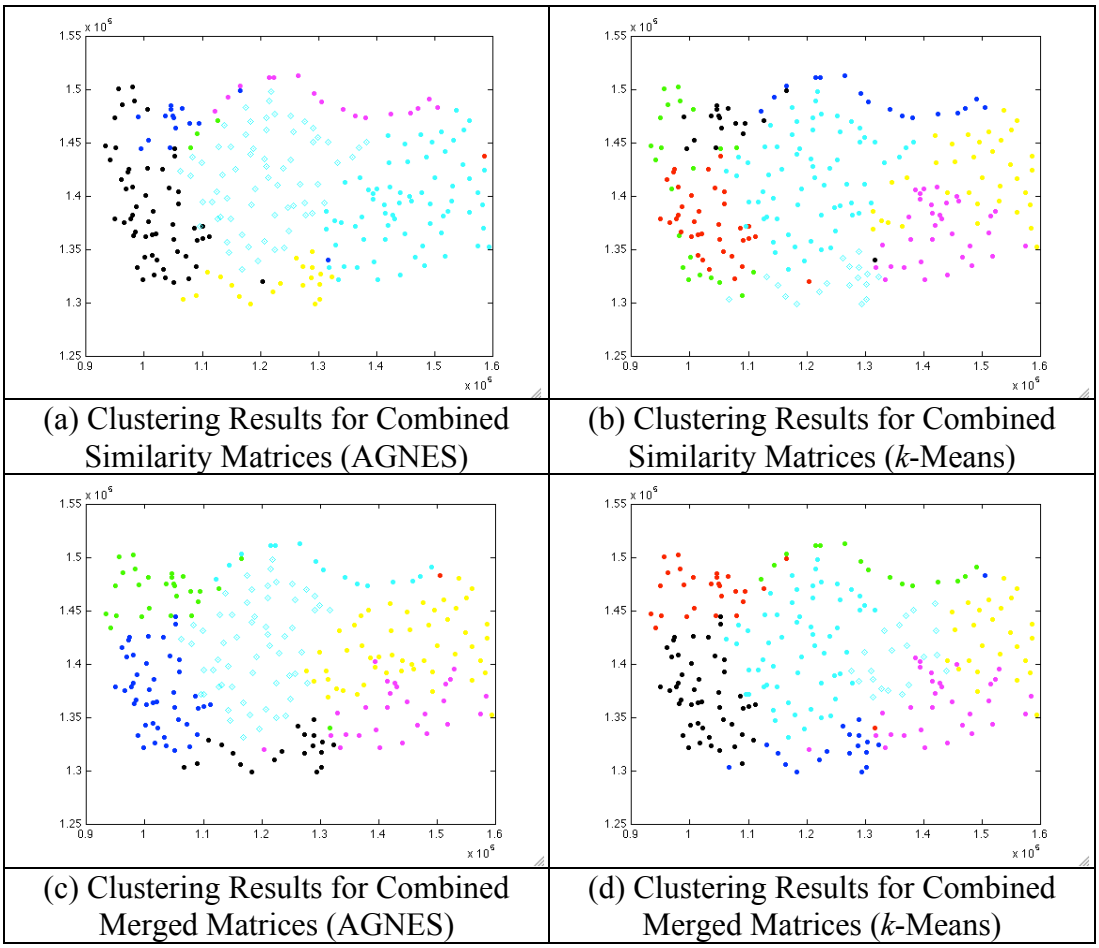
(a) Clustering Results for Combined Similarity Matrices (AGNES)

(b) Clustering Results for Combined Similarity Matrices (*k*-Means)

(c) Clustering Results for Combined Merged Matrices (AGNES)

(d) Clustering Results for Combined Merged Matrices (*k*-Means)

Figure 4.1 Clustering Results for *k*=7



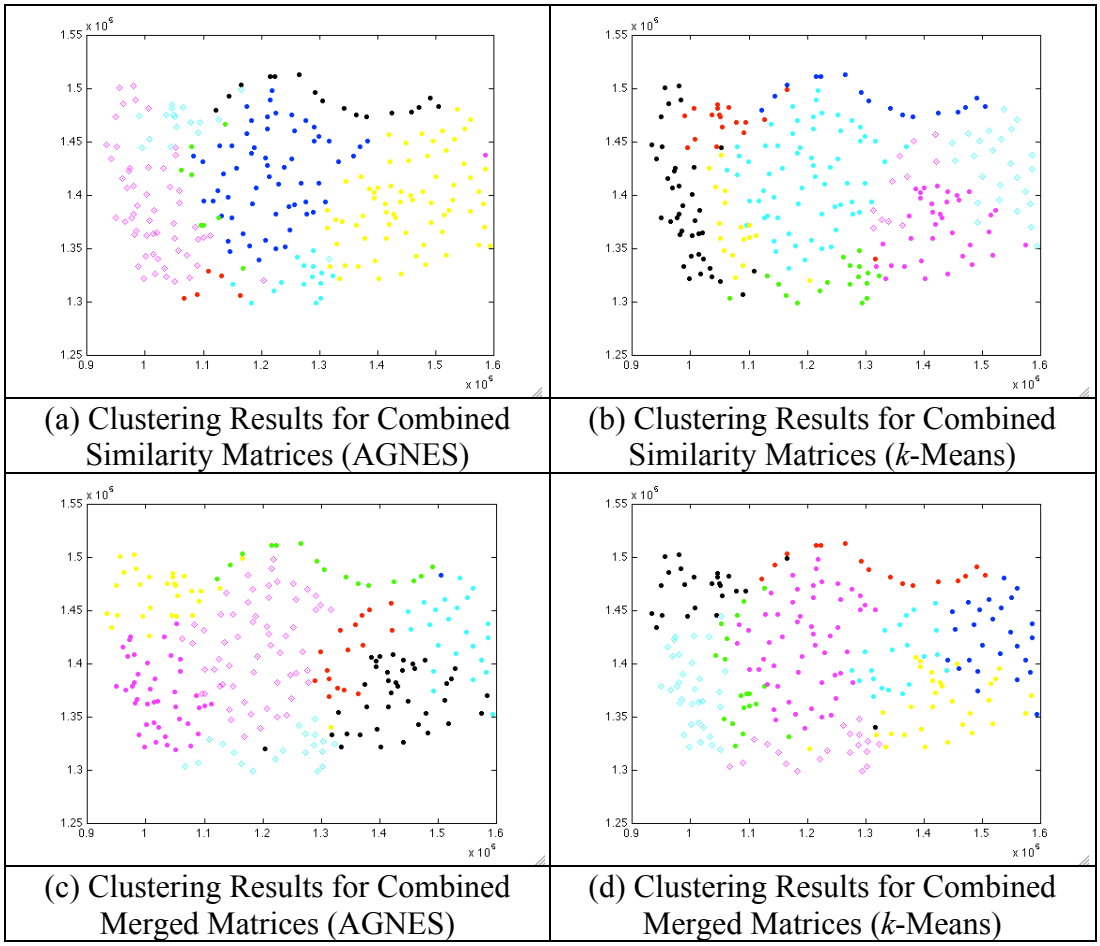
(a) Clustering Results for Combined Similarity Matrices (AGNES)

(b) Clustering Results for Combined Similarity Matrices (*k*-Means)

(c) Clustering Results for Combined Merged Matrices (AGNES)

(d) Clustering Results for Combined Merged Matrices (*k*-Means)

Figure 4.2 Clustering Results for *k*=8



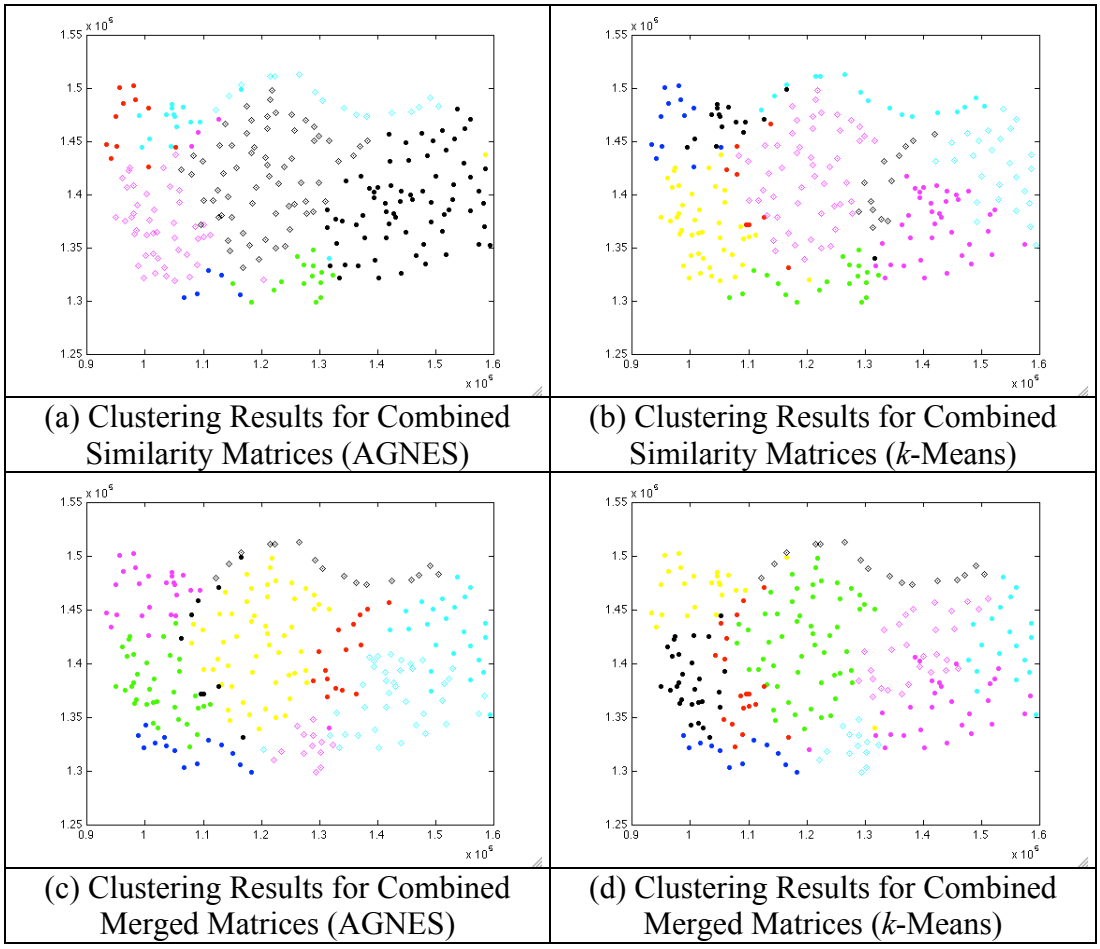
(a) Clustering Results for Combined Similarity Matrices (AGNES)

(b) Clustering Results for Combined Similarity Matrices (*k*-Means)

(c) Clustering Results for Combined Merged Matrices (AGNES)

(d) Clustering Results for Combined Merged Matrices (*k*-Means)

Figure 4.3 Clustering Results for *k*=9



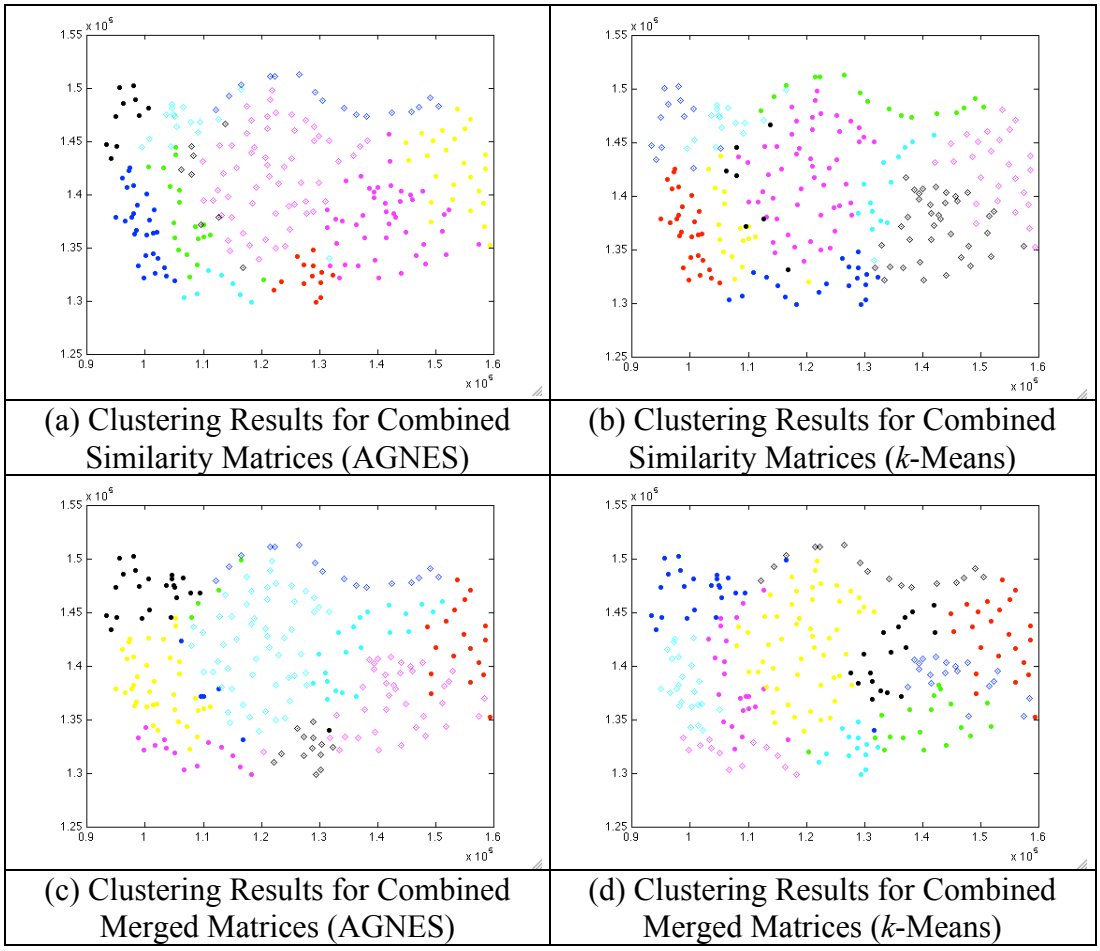
(a) Clustering Results for Combined Similarity Matrices (AGNES)

(b) Clustering Results for Combined Similarity Matrices (*k*-Means)

(c) Clustering Results for Combined Merged Matrices (AGNES)

(d) Clustering Results for Combined Merged Matrices (*k*-Means)

Figure 4.4 Clustering Results for *k*=10



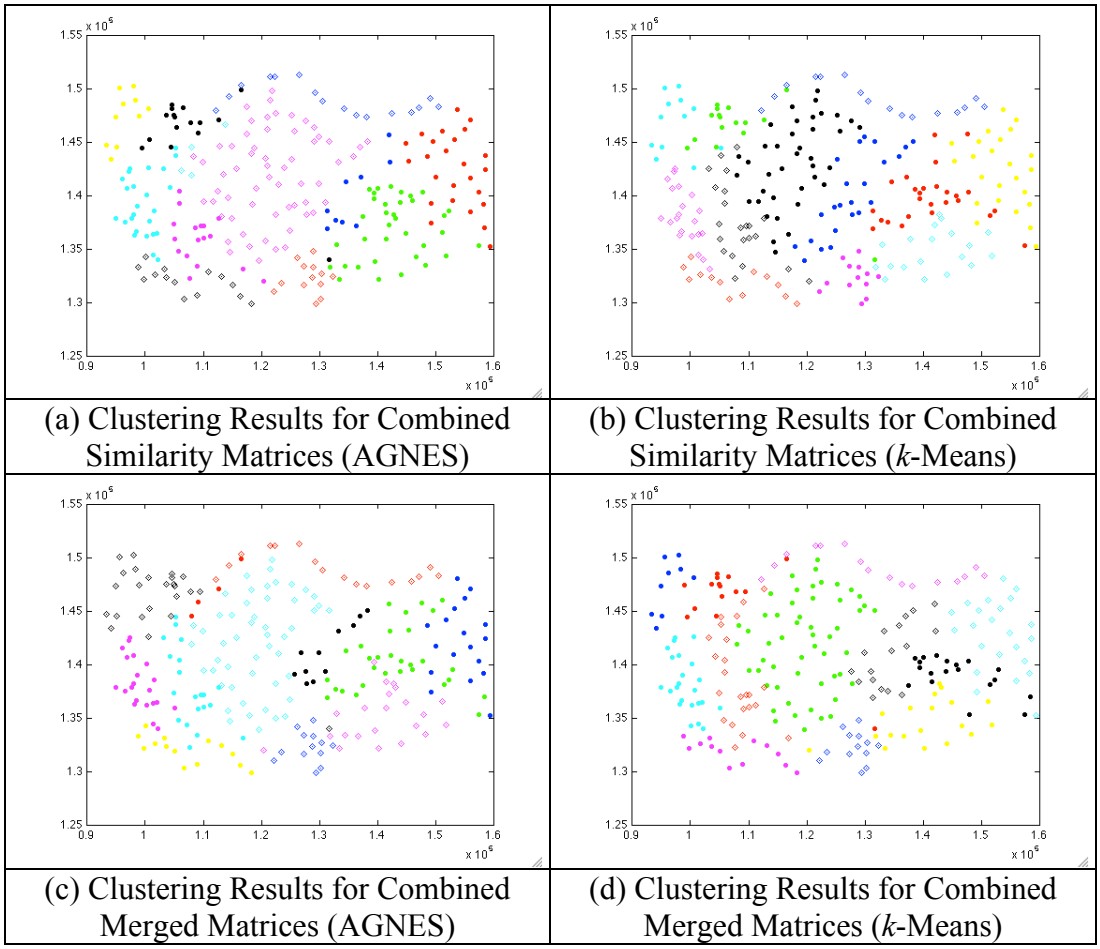
(a) Clustering Results for Combined Similarity Matrices (AGNES)

(b) Clustering Results for Combined Similarity Matrices (*k*-Means)

(c) Clustering Results for Combined Merged Matrices (AGNES)

(d) Clustering Results for Combined Merged Matrices (*k*-Means)

Figure 4.5 Clustering Results for k=11



(a) Clustering Results for Combined Similarity Matrices (AGNES)

(b) Clustering Results for Combined Similarity Matrices (*k*-Means)

(c) Clustering Results for Combined Merged Matrices (AGNES)

(d) Clustering Results for Combined Merged Matrices (*k*-Means)

Figure 4.6 Clustering Results for *k*=12

CHAPTER 5

CONCLUSION AND FUTURE WORK

Our initial suggestion was that, better clustering results can be obtained by using a similarity measure (DTW for our study) that is suitable for time series data, rather than simple distance measures (Euclidian Distance Measure etc.) used for common applications. In literature DTW provided successful results when used for classification applications [3], while in this study it is used for clustering applications. So in this study we have discussed the use of DTW as a similarity measure for time series data in clustering applications and whether it performs better or not. In Chapter 3 we have discussed the results of our experimentation and conclude that for most of the cases we experimented with DTW provides better results than the Euclidian Distance measure. However, consensus clustering with DTW is computationally very expensive when compared to the usage of Euclidian Distance measure. Thus this feature makes it harder to work with large datasets having too many time series samples and data points. As a feature work it will be also beneficial to use additional distance measures (cross-correlation etc.) to compare with DTW.

Also, dataset we have created (Simulated Data-2) backed up our initial expectation that DTW would perform better with data having phase shifts. This point is open for further experimentation with simulated datasets as the phase shift properties of real datasets are hard to observer if it was not considered in the data collection and mentioned in the dataset description.

In addition to our discussions with DTW we also discussed consensus clustering in this study. We were also expecting that with the benefits provided with consensus clustering approach we could obtain even better results for the time series data. In all the cases we experimented with, when used with consensus clustering DTW performs better than Euclidian Distance measure, both regarding the errors and cluster discoveries. However in some cases the performance difference was around 1%, which makes it unnecessary to use both DTW and Consensus Clustering, due to time consuming computations.

With the use of consensus clustering we also introduce to methodologies for multivariate clustering using DTW. We used this multivariate approach in the real world problem of defining Turkeys' Climate Regions described in Chapter 4. The results were compared to the results obtained by Iyigun et al. [38]. The results of

both studies have coinciding futures, yet it is not possible to tell which algorithm performed better as there is no original cluster labels. So there is a need to expert judgment in order to conclude which clustering is better regarding the real regional definition.

In this study we only analyzed Turkeys' Climate Regions by using the available data as long time series data. As a future work extension, it is also possible to analyze the available data as short-time series (i.e; 10 year periods) and demonstrate how the climate region definitions and the number of climate regions change over the years.

Finally, it should be mentioned that regarding the usage of DTW with Consensus Clustering and the multivariate problem approach, all the conclusions of this study are dependent on the dataset properties and need to be further experimented with different types of datasets in detail in order to come up with more solid conclusions.

REFERENCES

[1] J. Ghosh and A. Acharya, *Cluster Ensembles*, WIREs Data Mining and Knowledge Discovery Vol.1, 305-315 (2011).

[2] T.W. Liao, *Clustering of Time Series Data – A Survey*, Pattern Recognition Vol. 38, 1857-1874 (2005).

[3] Y. Jeong, M. K. Jeong, O.A. Omitaomu, *Weighted Dynamic Time Warping for Time Series Classification*, Pattern Recognition Vol. 44, 2231–2240 (2011).

[4] E. Keogh, C. A. Ratanamahatana, *Exact Indexing of Dynamic Time Warping*, Knowledge and Information Systems, Springer-Verlag London (2004).

[5] A.K. Jain, M.N. Murty and P.J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys Vol. 31, No. 3 (1999).

[6] R. Xu and D. Wunsch II, *Survey of Clustering Algorithms*, IEEE Transactions On Neural Networks Vol. 16, No. 3 (2005).

[7] P. Berkhin, *A Survey of Clustering Data Mining Techniques*, Grouping Multidimensional Data: Recent Advances in Clustering, Springer Berlin Heidelberg (2006).

[8] P. Senin, *Dynamic Time Warping Algorithm Review*, Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA (2008).

[9] M. Muller, *Chapter 4: Dynamic Time Warping*, Informational Retrieval for Music and Motion, Springer Meinard (2007).

[10] S. Monti, P. Tamayo, J. Mesirov and T. Golub, *Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data*, Machine Learning Vol. 52, 91-118 (2003).

[11] C. Goutte, P. Toft, E. Rostrup, *On Clustering FMRI Time Series*, Neuroimage Vol. 9 (3) , 298–310 (1999).

[12] C.T. Shaw, G.P. King, *Using Cluster Analysis to Classify Time Series*, Physica D Vol. 58, 288–298 (1992).

[13] D. Piccolo, *A Distance Measure for Classifying ARMA Models*, J. Time Ser. Anal. Vol. 11 (2), 153–163 (1990).

[14] J.J. van Wijk, E.R. van Selow, *Cluster and Calendar Based Visualization of Time Series Data*, Proceedings of IEEE Symposium on Information Visualization, San Francisco, USA (1999).

[15] R.H. Shumway, *Time–Frequency Clustering and Discriminant Analysis*, Stat. Probab. Lett. Vol. 63, 307–314 (2003).

[16] M. Ramoni, P. Sebastiani, P. Cohen, *Multivariate Clustering by Dynamics*, Proceedings of the 2000 National Conference on Artificial Intelligence (AAAI-2000), San Francisco, USA, 633–638 (2000).

[17] Y. Kakizawa, R.H. Shumway, N. Taniguchi, *Discrimination and Clustering for Multivariate Time Series*, J. Amer. Stat. Assoc. Vol. 93 (441), 328–340 (1998).

[18] M. Kumar, N.R. Patel, J. Woo, *Clustering Seasonality Patterns in the Presence of Errors*, Proceedings of KDD '02, Edmonton, Alberta, Canada (2002).

[19] L.M.D. Owsley, L.E. Atlas, G.D. Bernard, *Self-Organizing Feature Maps and Hidden Markov Models for Machine Tool Monitoring*, IEEE Trans. Signal Process. Vol. 45 (11), 2787–2798 (1997).

[20] M. Vlachos, J. Lin, E. Keogh, D. Gunopulos, *A Wavelet Based Anytime Algorithm for K-Means Clustering of Time Series*, Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, USA (2003).

[21] J.G. Wilpon, L.R. Rabiner, *Modified k-Means Clustering Algorithm for Use in Isolated Word Recognitio*, IEEE Trans. Acoust. Speech Signal Process. Vol. 33 (3), 587–594 (1985).

[22] T.W. Liao, B. Bolt, J. Forester, E. Hailman, C. Hansen, R.C. Kaste, J. O'May, *Understanding and Projecting the Battle State*, 23rd Army Science Conference, Orlando, USA (2002).

[23] K. Kalpakis, D. Gada, V. Puttagunta, *Distance Measures for Effective Clustering of ARIMA Time-Series*, Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, USA, 273–280 (2001).

[24] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, P. Boesiger, *A New Correlation-Based Fuzzy Logic Clustering Algorithm For fMRI*, Mag. Resonance Med. Vol. 40, 249–260 (1998).

[25] C.S. Möller-Levet, F. Klawonn, K.-H. Cho, O. Wolkenhauer, *Fuzzy Clustering of Short Time Series and Unevenly Distributed Sampling Points*, Proceedings of the 5th International Symposium on Intelligent Data Analysis, Berlin, Germany (2003).

[26] R. Ghaemi, M. N. Sulaiman , H. Ibrahim , N. Mustapha, *A Survey: Clustering Ensembles Techniques*, World Academy of Science, Engineering and Technology, Vol. 50 (2009).

[27] J. Ghosh, A. Strehl and S. Merugu, *A Consensus Framework for Integrating Distributed Clusterings Under Limited Knowledge Sharing*, Proceedings of NSF Workshop on Next Generation Data Mining, Baltimore, USA, 99–108 (2002).

[28] A. Topchy, A. Jain and W. Punch, *A Mixture Model for Clustering Ensembles*, Proceedings of SIAM International Conference on Data Mining, 379-390 (2004).

[29] A. L. N. Fred and A. K. Jain, *Data Clustering Using Evidence Accumulation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 835-850 (2002).

[30] A. Strehl and J. Ghosh, *Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions,* Journal of Machine Learning Research, February, 583-617 (2002).

[31] S. Dudoit and J. Fridlyand, *Bagging to Improve the Accuracy of a Clustering Procedure*, Bioinformatics Vol. 19, 1090–1099 (2004).

[32] B. Fischer and J.M. Buhmann, *Bagging for Path-Based Clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25 (11), 1411–1415 (2003).

[33] H. Sakoe and S. Chiba, *Dynamic Programming Algorithm Optimization for Spoken Word Recognitio*, IEEE Trans Acoustics Speech Signal Process ASSP Vol. 26, 43–49 (1978).

[34] T. I. Simpson, "R Package-clusterCons" (2011).

[35] K. Bache and M. Lichman, *UCI Machine Learning Repository*, [http://archive.ics.uci.edu/ml]. Irvine, CA, University of California, School of Information and Computer Science (2013).

[36] R.J. Alcock and Y. Manolopoulos, *Time-Series Similarity Queries Employing a Feature-Based Approach*, 7th Hellenic Conference on Informatics, Ioannina, Greece (1999).

[37] K. Altun, B. Barshan, and O. Tunçel, *Comparative Study on Classifying Human Activities with Miniature Inertial and Magnetic Sensors*, Pattern Recognition, Vol. 43 (10), 3605-3620 (2010).

[38] C. Iyigün, M. Turkes, I. Batmaz, C. Yozgatligil, V. Purutcuoglu, E. Kartal Koc and M. Z. Ozturk, *Clustering Current Climate Regions of Turkey by Using a Multivariate Statistical Method*, Theoretical and Applied Climatology, January (2013).

[39] S. Aslan, C. Yozgatlıgil, C. Iyigün, I. Batmaz, M. Turkes and H. Tatli., *Comparison of Missing Value Imputation Methods for Turkish Monthly Total Precipitation Data*, 9th International Conference on Computer Data Analysis and Modeling: Complex Stochastic Data and Systems, Minsk, Belarus, 137–140 (2010).

[40] C. Yozgatligil, S. Aslan, C. Iyigün and I. Batmaz, *Comparison of Missing Value Imputation Methods in Time Series: The Case of Turkish Meteorological Data*, Theor Appl Climatol, Vol. 112 (1-2), 143-167 (2013).

[41] E. Kartal, C. Iyigün, F. M. Fahmi, C. Yozgatlıgil, V. Purutcuoglu, I. Batmaz, G. Koksal and M. Turkes, *Identifying Climate Zones of Turkey by Hierarchical Clustering Method*, Journal of Statistical Research Vol. 8 (1), 13-25 (2011).