UNIVERSITY OF OKLAHOMA

GRADIATE COLLEGE

THE APPLICATION OF MULTIVARIATE DISTANCE MATRIX REGRESSION IN TRANSPORTATION FOR TRAFFIC ANALYSIS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

in

ELECTRICAL AND COMPUTER ENGINEERING

By

MUHANAD SHAB KALEIA
NORMAN, OKLAHOMA
2017

THE APPLICATION OF MULTIVARIATE DISTANCE MATRIX REGRESSION IN TRANSPORTATION FOR TRAFFIC ANALYSIS

A THESIS APPROVED FOR THE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Hazem Refai, Chair

Dr. Thordur Runolfsson

Dr. Gregory MacDonald

To my beloved father **Reyad**, your wisdom, and support guided me to succe

To my beloved mother **Noha**, who believe in me and support me in every ste

To my beloved brother and sister **Husam** and **Shahla**, who gave my life joyfulne

To my future **wife**,

To my friends, and my peo

I dedicate this work to you, thank you from my heart.

Muhanad Shab Kaleia

Acknowledgment

I would like to extend my heartfelt appreciation and gratitude to my advisor, Dr. Hazem H. Refai, for the support and guidance he has provided throughout my Master journey at The University of Oklahoma.

I would also like to express my sincere gratitude to the distinguished members of my thesis committee. Dr. Thordur Runolfsson, and Dr. Gregory MacDonald.

I want to thank my friends that inspired me during a difficult time when I needed words of encouragement, for their help and support in my journey at Oklahoma University. I couldn't have done with without them. I wish them the best in their life.

Special appreciation goes to the extended OU-TULSA family of professors, students, and staff, as well as to Michelle Farabough for her assistance in editing this thesis.

Contents

List of Figures

Abstract

A critical function of intelligent transportation systems is studying and analyzing the effects of road condition variables (e.g. construction, severe weather, and the like) on traffic to aid in improving road designs, estimating travel time, and increasing safety. In this thesis, Multivariate Distance Matrix Regression (MDMR), a well-studied algorithm applied in brain research, is explored and applied in the transportation domain to assess the relationship and the effects of traffic conditions on transportation system performance.

The Multivariate Distance Matrix Regression (MDMR) is utilized to study the relationship between input experimental factors and the association of response variables. When studying transportation, input factors can be represented as any factor that may have an effect on traffic, and response variables can be represented by traffic speed values over time for each segment of a road. The output is represented as a probability Value (P-Value) for each segment of the road as an indication of an effect of the studied factor on that specific segment. The National Performance Management Research Dataset (NPMRDS), (i.e., a probe-based traffic dataset) was used to study traffic performance based on specific factors by applying MDMR under different traffic scenarios.

Moreover, a novel clustering algorithm for time series data is proposed by optimizing the F-statistic (i.e., a measurement metric to study the significance difference of two or more groups) to find the best segregation of time series between two or more groups. The clustering algorithm gave promising preliminary results when compared with K-means.

Chapter 1

Introduction

In the transportation domain, traffic can be characterized by two parameters-travel time and speed values over time. These parameters are greatly affected by several factors (e.g., severe weather, road conditions, accidents, time of day). Studying the impact of such factors on traffic is extremely important in traffic management and planning to aid in making decisions for improving road designs, estimating travel time, and, more importantly, increasing safety. Studying the relationship between traffic and factors affecting it requires collecting, storing, and processing various types of data by leveraging several technologies. For example, Road Weather Information Systems (RWIS) [3] can be used as a weather dataset for providing temperature measurements, humidity, wind speed, and precipitation information. The conflation of different types of data (e.g., weather, accidents, and travel time) under consideration is both complex and challenging [4]. Data must be aggregated, and several locations must be merged to retrieve useful information about traffic in a specific location. In this thesis, the impact of factors on traffic performance is analyzed using the National Performance Management Research Dataset [5]

(NPMRDS) and a multivariate statistical analysis framework, namely Multivariate Distance Matrix Regression (MDMR) [6].

1.1 National Performance Management Research Dataset

The emergence of the Internet of Things (IoT) and digitization of urban infrastructure variables (e.g. transportation, weather) enables the collection of continuous data associated with many aspects of our modern life. The dataset we used in this thesis to characterize traffic is the National Performance Management Research Dataset (NPMRDS). NPMRDS is a probe-based traffic data collected using automobile-probes that report location and speed at regular intervals of time to the cloud. Location is determined by standard GPS equipment housed inside the vehicle (e.g., smart phone). Reported speed and location values are matched with a map detailing speed values and travel time for every segment in each roadway, where each roadway is divided into a set of segments. When multiple speed values are reported for the same segment, speed is averaged over all received values [5]. Data is provided by INRIX, a commercial third party, with no smoothing, filtering, or removing outliers [5]. Detecting anomalous points and patterns in the NPMRDS dataset permits departments of transportation to answer important questions (e.g., what factors affects traffic performance, which segments contain congestion, and when did the congestion occur). The dataset is archived and published monthly. Observations (like travel time and speed values) are reported for each Traffic Message Channel (TMC) segment in 5-minutes intervals on any given day. Figure 1.1 presents information on each road segment on Oklahoma Interstate Highway 35 (I-35), which is then divided into multiple segments. Following is an explanation of each field in the

NPMRDS table.

- **Datasource**: observation value sourceâĂŤpassenger car, truck, or both
- **TMC**: roadway segment number
- **Road**: roadway/highway name (e.g., I-35)
- **Direction**: traffic direction: northbound, southbound, eastbound, or westbound
- **Latitude/Longitude location**: location of each segment, specified by starting lat/long and ending lat/long
- **Miles**: segment length in miles
- **Road order**: sequence of segments

datasource	tmc	road	direction	intersection	state	county	zip	start_latitude	start_longitude	end_latitude	end_longitude	miles	road_order
NPMRDS (Passenger vehicles)	111+05483	I-35	NORTHBOUND	US-77/EXIT 1	OK	LOVE	73459	33.726982	-97.159583	33.7358544	-97.1491926	0.859282	550
NPMRDS (Passenger vehicles)	111P05483	I-35	NORTHBOUND	US-77/EXIT 1	OK	LOVE	73459	33.7358544	-97.1491926	33.7408862	-97.1419804	0.541804	551
NPMRDS (Passenger vehicles)	111+05484	I-35	NORTHBOUND	OK-153/EXIT 5	OK	LOVE	73459	33.7408862	-97.1419804	33.791804	-97.134814	3.621269	552
NPMRDS (Passenger vehicles)	111P05484	I-35	NORTHBOUND	OK-153/EXIT 5	OK	LOVE	73459	33.791804	-97.134814	33.799153	-97.134814	0.507691	553
NPMRDS (Passenger vehicles)	111+05485	I-35	NORTHBOUND	OK-32/EXIT 15	OK	LOVE	73448	33.799153	-97.134814	33.9368734	-97.1347614	9.577229	554
NPMRDS (Passenger vehicles)	111P05485	I-35	NORTHBOUND	OK-32/EXIT 15	OK	LOVE	73448	33.9368734	-97.1347614	33.944567	-97.134791	0.53149	555
NPMRDS (Passenger vehicles)	111+05486	I-35	NORTHBOUND	OSWALT RD/EXIT 21	OK	LOVE	73448	33.944567	-97.134791	34.0225966	-97.1460262	5.47257	556
NPMRDS (Passenger vehicles)	111P05486	I-35	NORTHBOUND	OSWALT RD/EXIT 21	OK	LOVE	73448	34.0225966	-97.1460262	34.031081	-97.148333	0.600866	557
NPMRDS (Passenger vehicles)	111+05487	I-35	NORTHBOUND	OK-77 SCENIC/EXIT 24	OK	LOVE	73453	34.031081	-97.148333	34.0662588	-97.1495834	2.444318	558
NPMRDS (Passenger vehicles)	111P05487	I-35	NORTHBOUND	OK-77 SCENIC/EXIT 24	OK	LOVE	73453	34.0662588	-97.1495834	34.074312	-97.1494146	0.556456	559
NPMRDS (Passenger vehicles)	111+05488	I-35	NORTHBOUND	US-70/EXIT 29	OK	CARTER	73401	34.074312	-97.1494146	34.1346564	-97.155861	4.205706	560
NPMRDS (Passenger vehicles)	111P05488	I-35	NORTHBOUND	US-70/EXIT 29	OK	CARTER	73401	34.1346564	-97.155861	34.143044	-97.1553052	0.580423	561
NPMRDS (Passenger vehicles)	111+05489	I-35	NORTHBOUND	US-70/OK-199/EXIT 31	OK	CARTER	73401	34.143044	-97.1553052	34.167702	-97.1673652	1.864337	562
NPMRDS (Passenger vehicles)	111P05489	I-35	NORTHBOUND	US-70/OK-199/EXIT 31	OK	CARTER	73401	34.167702	-97.1673652	34.177475	-97.167832	0.676667	563
NPMRDS (Passenger vehicles)	111+05490	I-35	NORTHBOUND	12TH AVE/EXIT 32	OK	CARTER	73401	34.177475	-97.167832	34.187932	-97.165905	0.734548	564
NPMRDS (Passenger vehicles)	111P05490	I-35	NORTHBOUND	12TH AVE/EXIT 32	OK	CARTER	73401	34.187932	-97.165905	34.1921746	-97.1644892	0.304385	565
NPMRDS (Passenger vehicles)	111+05491	I-35	NORTHBOUND	OK-142/EXIT 33	OK	CARTER	73401	34.1921746	-97.1644892	34.1981512	-97.1638136	0.415306	566
NPMRDS (Passenger vehicles)	111P05491	I-35	NORTHBOUND	OK-142/EXIT 33	OK	CARTER	73401	34.1981512	-97.1638136	34.205908	-97.163837	0.535877	567
NPMRDS (Passenger vehicles)	111+05492	I-35	NORTHBOUND	BROOKS RD/EXIT 40	OK	CARTER	73401	34.205908	-97.163837	34.299112	-97.159304	6.49224	568
NPMRDS (Passenger vehicles)	111P05492	I-35	NORTHBOUND	BROOKS RD/EXIT 40	OK	CARTER	73458	34.299112	-97.159304	34.307294	-97.159261	0.566554	569
NPMRDS (Passenger vehicles)	111+05493	I-35	NORTHBOUND	OK-53/EXIT 42	OK	CARTER	73458	34.307294	-97.159261	34.3290876	-97.1537066	1.540633	570
NPMRDS (Passenger vehicles)	111P05493	I-35	NORTHBOUND	OK-53/EXIT 42	OK	CARTER	73458	34.3290876	-97.1537066	34.3363798	-97.1506572	0.533012	570.1
NPMRDS (Passenger vehicles)	111+05494	I-35	NORTHBOUND	US-77/EXIT 47	OK	CARTER	73458	34.3363798	-97.1506572	34.394248	-97.141693	4.068178	572
NPMRDS (Passenger vehicles)	111P05494	I-35	NORTHBOUND	US-77/EXIT 47	OK	MURRAY	73030	34.394248	-97.141693	34.402091	-97.140677	0.545415	573
NPMRDS (Passenger vehicles)	111+05495	I-35	NORTHBOUND	US-77/EXIT 51	OK	MURRAY	73030	34.402091	-97.140677	34.448298	-97.132769	3.37118	574
NPMRDS (Passenger vehicles)	111P05495	I-35	NORTHBOUND	US-77/EXIT 51	OK	MURRAY	73030	34.448298	-97.132769	34.453856	-97.138121	0.490287	575
NPMRDS (Passenger vehicles)	111+05496	I-35	NORTHBOUND	OK-7/EXIT 55	OK	MURRAY	73030	34.453856	-97.138121	34.5022824	-97.1712342	3.880164	576

Figure 1.1: NPMRDS TMC identification table

Figure 1.2 illustrates observations values represented in the following fields:

- **Measurement timestamp**: reporting time represented by the day-of-the-month and five-minute epoch during the day
- **Speed**: miles per hour

- **Travel time in minutes**: self-explained

- **Data density**: Data density: three possible levels that represent number of vehicles reporting speed value (i.e., Level A represents one to four reporting vehicles; Level B indicates five to nine reporting vehicles, and Level C represents 10 or more reporting vehicles) [7].

datasource	tmc_code	measurement_tstamp	speed	travel_time_minutes	data_density
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:20:00	68	0.76	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:25:00	71	0.73	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:30:00	66	0.78	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:35:00	74	0.7	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:40:00	72	0.72	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:45:00	67	0.77	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:50:00	68	0.76	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 16:55:00			
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:00:00	75	0.69	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:05:00	70	0.74	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:10:00	78	0.66	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:15:00	74	0.7	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:20:00	72	0.72	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:25:00	64	0.81	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:30:00	64	0.81	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:35:00	73	0.71	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:40:00	73	0.71	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:45:00	73	0.71	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:50:00			
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 17:55:00	65	0.79	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:00:00	68	0.76	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:05:00	72	0.72	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:10:00	72	0.72	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:15:00	68	0.76	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:20:00	63	0.82	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:25:00	70	0.74	A
NPMRDS (Passenger vehicles)	111+05483	2017-02-26 18:30:00	70	0.74	A

Figure 1.2: NPMRDS observations table

By linking the TMC identification table with the observation table, travel time data can be analyzed for extracting important traffic performance knowledge. Figure 1.4 illustrates I-35 plotted on Google Maps, where location information for each segment is taken from the TMC identification table. Figure 1.3 illustrates speed values for one I-35 segment on a specific day between 6 AM and 1 PM.

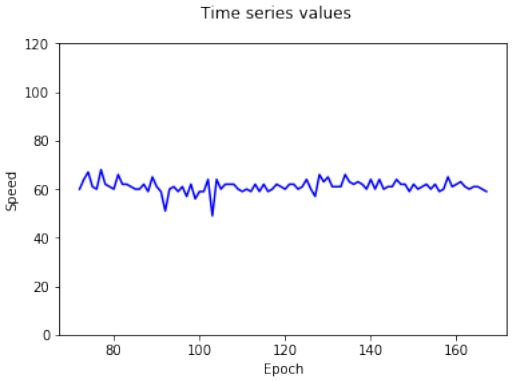

Figure 1.3: Speed values for one segment between 6 AM and 1 PM

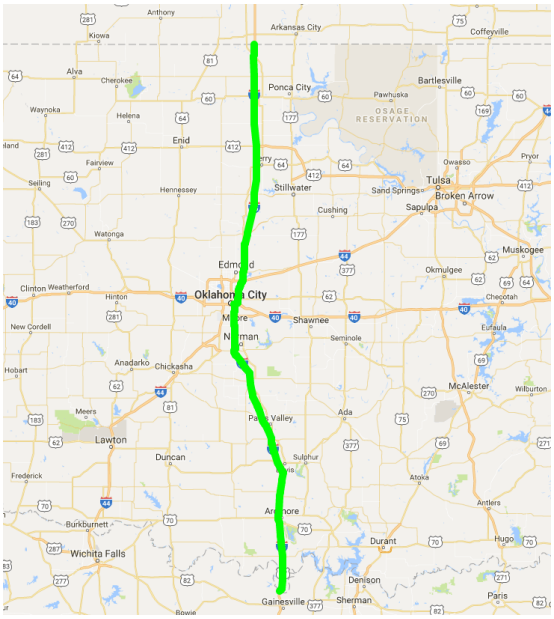

Figure 1.4: Oklahoma highway I-35

1.2 Purpose and significance of the work

The work detailed in this thesis focused on exploring and applying a Multivariate Distance Matrix Regression (MDMR) [6] algorithm in the transportation domain for studying the effects and the relationship between various factors (e.g., weather status) and traffic performance. MDMR is considered a hypothesis testing method used to reject or accept the null hypothesis based on analyzing the variance between two or more multivariate groups. The algorithm output serves as an indication of regions or road segments where traffic performs differently in the presence of the studied factors. Knowing the effects of a given factor on a specific segment in terms of traffic congestion allows a state department of transportation agency to make informed decisions for improving road designs, preventing traffic congestion, and increasing safety. This work also proposes the use of a novel clustering tool that is based on maximizing F-statistic by applying random permutation to cluster time series data. The proposed algorithm can be used on transportation data to cluster normal time series for differentiating between normal and congested traffic time series.

This thesis is organized, as follows. Discussion about related works studying the impact of factors on traffic are detailed in Chapter 2. A multivariate distance-based analytic framework is proposed in Chapter 3 for studying the effect of experimental factors on observations. Chapter 4 introduces a novel clustering algorithm used for anomaly detection. Chapter 5 concludes the thesis and details future works.

Chapter 2

Related Works

Several investigations have studied the relationships and effects between traffic conditions (or factors) and traffic performance. Most follow simple statistical approaches (e.g., univariate analysis of variance or comparison of the means performed manually between different groups) to make a statement about the studied factor.

Akin et al [8] studied traffic speed as a function of weather conditions (e.g., clear, rain, fog, or snow) and surface conditions (e.g., dry, wet, or icy). Historical weather and speed data from two highways in an Istanbul metropolitan area were analyzed. The study applied ANOVA (analysis of variance) to determine the significance of the differences between a road under adverse weather conditions and the same road under normal weather conditions. A statistical analysis of the data was applied to calculate average traffic speed difference under various conditions. Findings demonstrated that rain reduced average traffic speed by 8 to 12%, while wet surface conditions reduced average traffic speed by 6 to 7%.

The effect of using Variable Speed Limit (VSL) strategies on traffic stream was studied

by Soriguera, et al [9]. A dataset from a VSL experiment carried out on a freeway in Spain was used. Data included vehicle count, speed, and occupancy for three days each with a different fixed speed limit (80 km/h, 60 km/h, and 40 km/h). Results revealed that lower speed limits increase speed differences between lanes in a road; therefore, lane changing rate increases.

A study conducted by Baldasano, et al. [10] assessed the effects of changing speed limit on air pollution. Traffic data collected in 2007 and 2008 in the city of Barcelona was used to compare the effect of introducing a speed limit. Hourly traffic intensity and hourly variable speeds were used to assess air quality using an emission model. The study showed that the speed limit enhanced air quality be 5 to 7%.

The impact of various factors (e.g., weather, choice of road, time of day, and day of the week) on traffic performance was also studied in [11]. Different machine learning decision tree-based algorithms (e.g., Decision Stump, M5 model tree, M5 regression tree, RepTree, M5 rules, and linear regression) were utilized to study dependence of influencing factors and traffic performance.

Most of the works in the literature do not consider a multivariate situation, where a response variable may represented as a multivariate data. Considering only univariate response variables provides a statement to describe the interaction between the studied factors and the whole univariate response variable at once, without taking into account that a factor may have an effect on part of the road without having an effect on other parts of the road. A multivariate statistical approach (MDMR) was be explored in this thesis in order to overcome this limitation.

Chapter 3

Multivariate Distance Matrix Regression

Multivariate Distance Matrix Regression (MDMR) is a hypothesis testing method [12] for multivariate data [13] aimed at determining the relationship between inputs (i.e., predictors) and observations in experiments by way of test-statistic. The algorithm originally proposed in [14] was introduced as a new non-parametric method for multivariate analysis of variance, wherein the test-statistic is similar to FisherâĂŹs F-ratio and is calculated from the distance matrix. This chapter discusses the details of the multivariate distance matrix regression method and explains its application in the transportation domain. The case of univariate analysis of variance is detailed in section 1. Non-parametric multivariate analysis of variance is explained in section 2. Section 3 provides an explanation of the MDMR framework used in brain research. Its proposed application in the transportation domain is detailed in section 4 wherein several case studies are applied to test the algorithm.

3.1 Analysis of Variance

Analysis of variance (ANOVA) [15] is a statistical method to test the differences between the means of two or more groups by analyzing the variance so that the null hypothesis can be either accepted or rejected. The null hypothesis supposes all groups means are equal (3.1).

$$H_0 : \mu_0 = \mu_1 = ... = \mu_n \tag{3.1}$$

When rejecting the null hypothesis, at least one group mean must be different from at least one other group mean. Before proceeding to ANOVA measurements, it is important to mention inherent assumptions of this method of analysis: :

1. Observations are normally distributed, (i.e., experimental errors of samples are normally distributed)

2. Independence of observations, (i.e., each observation is independent from others)

3. Variance homogeneity, (e.i., equal variances between groups)

ANOVA is based on two estimates. Notably, mean square error (MSE) estimates population variance regardless if the null hypothesis is true. Mean square between (MSB) is based on sample mean differences. MSB estimates the variance if population means are equal. If they are not, MSB will be significantly larger than the MSE, which means the null hypothesis will be rejected. Using these two measurements F-ratio can be calculated as in equation (3.3).

$$F = \frac{Variability - between - groups}{Variability - within - groups} \tag{3.2}$$

$$F = \frac{MSB}{MSE} \tag{3.3}$$

MSE (i.e, variation within groups) can be calculated as the mean of sample variances or, in other words, can be computed by taking the difference between each point and its group mean, and then dividing the sum over the degree of freedom. While MSB represents the variation between groups, it can be calculated in (3.4) by first computing the means of the groups and, then computing the variance of the means. Finally, the variance of the means multiplied by n, where n is number of observations in each group, must be multiplied.

$$MSB = n * \sigma_M^2 \tag{3.4}$$

F-ration can be calculated using sum of squares within groups **SSW** and sum of squares between groups **SSA** as in 3.8 where n is number of observations in the ith group, and a is number of groups.

$$SS_W = \sum_{i=1}^{a} \sum_{j=1}^{n} (y_{ij} - \overline{y_i})^2 \tag{3.5}$$

$$SS_T = \sum_{i=1}^{a} \sum_{j=1}^{n} (y_{ij} - \overline{y})^2 \tag{3.6}$$

$$SS_A = SS_T - SS_W \tag{3.7}$$

$$F = \frac{SS_A/(a-1)}{SS_W/(N-a)} \tag{3.8}$$

The nominator and denominator are divided by the degree of freedom $(a-1)$ and $(N-a)$ Where N is total number of observations $N = a * n$ and a is number of groups.

Comparing MSB to MSE is considered a critical step in ANOVA for determining if the null hypothesis should be accepted or rejected. MSB estimates a larger value than MSE when population means are not equal. However, to reject the null hypothesis it is important to know how larger MSB should be. Based on F-statistic (3.3), a decision can be made about the null hypothesis using Fisher distribution [16]. F-distribution, illustrated in figure 3.1, is a continuous probability distribution that represents the null distribution and is used to find the p-value as an indicator to reject or accept the null hypothesis. If the F-ratio is located on the right tail of the distribution, the null hypothesis can be rejected.

3.2 Non-parametric multivariate analysis of variance

Univariate analysis of variance, explained in the previous section, provides a powerful hypothesis testing tool. However, a multivariate analysis of variance method that is not restricted to stringent assumptions made by ANOVA is needed. Anderson in [14] proposed a non-parametric multivariate analysis of variance as a hypothesis testing method that does not rely on stringent assumptions and provides more intuitive formulation for ANOVA. The test-statistic in [14] is a multivariate analogue to FisherâĂŹs F-ratio, where the F-ratio can be calculated directly from the distance matrix. In contrast to ANOVA,

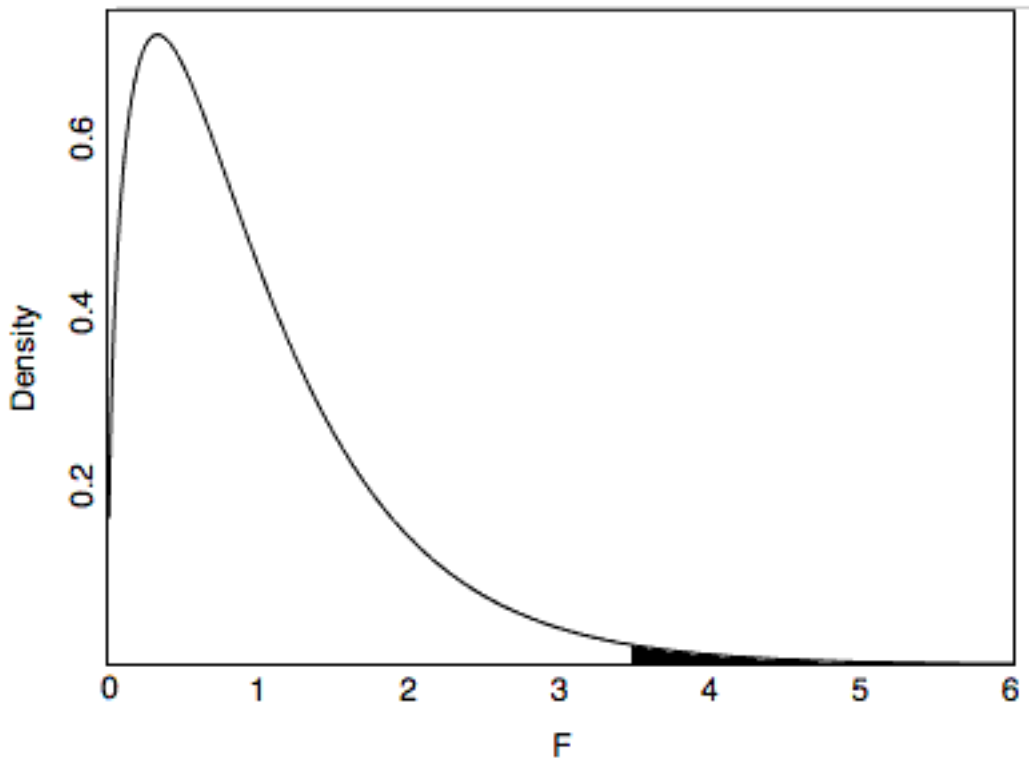

Figure 3.1: F-distribution

this method does not relay on specific distance metric to build the distance matrix. Also, it does not follow Fisher distribution. Rather, P-value is calculated by applying permutation of the observation between groups to obtain a rigorous probabilistic statement of experimental factors. The proposed method in [14] is referred to as non-parametric Multivariate Analysis of Variance and is accomplished, as follows:

1. Construct test-statistic

2. Calculate P-value using permutation

3.2.1 The test-statistic F-ratio

To calculate F-statistic in the case of multivariate âĂŤ contrasted with the F-ratio in the univariate ANOVA (3.8), the sum of squares across all variables must be determined.

Equation 3.9 represents the sum of squares within groups for p variables.

$$SS_w = \sum_{i=1}^{a}\sum_{j=1}^{n}\sum_{k=1}^{p}(y_{ijk} - \overline{y_{ik}})^2 \tag{3.9}$$

The formula 3.9 can be written as sum of squared Euclidean distances between each individual and its group center as in 3.10

$$SS_W = \sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \overline{y_i})^T(y_{ij} - \overline{y_i}) \tag{3.10}$$

Based on the fact that the sum of squared distances between points and their centroid is equal to the sum of squared inter-point distances divided by the number of points, total sum of square can be written as in equation 3.11

$$SS_T = \frac{1}{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 \tag{3.11}$$

where N is the total number of observations, and d_{ij} is the distance between observation i and observation j. Sum of squares within groups can similarly be written as in equation 3.12 where the variable ε_{ij} assumes a value of 1 when the observation i and j are in the same group and otherwise assumes a value of 0.

$$SS_W = \frac{1}{n}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2\varepsilon_{ij} \tag{3.12}$$

The sum of squares between groups $SS_A = SS_T - SS_W$ and the F-ratio can then be calculated as in equation (3.9). Given that the groups have different central locations then the among-group distances will be relatively large when compared to the within-group

distances; F-ratio will be relatively large.

3.2.2 Obtaining P-value using permutation

P-value (i.e., probability value) provides the probability model when the null hypothesis is true [15]. Unlike ANOVA, F-statistic does not follow Fisher's distribution, mainly because observation variables are not required to be normally distributed. Also, there is no restriction on using Euclidean distance for the analysis. The null distribution can be estimated using the permutation of the observation between groups, This is so due to the fact that if the null hypothesis is true and the groups are not actually different then observations can be shuffled randomly between groups without affecting the F-ratio, (i.e., the new F value obtained by each permutation called F^{Π}). The permutation and re-calculation of F^{Π} is then repeated by keep shuffling observations between groups, where in a one-way test the total number of possible permutation is $(an)!/(a!(n!)^a)$. This result will give an estimated distribution of the pseudo F-statistic under null hypothesis. P-value can be calculated by comparing the original F value with the original ordering of observations in their groups, given that the distribution was created by permuting observations, as in Equation 3.13

$$P = \frac{(No.\ of F^{\Pi} \geq F)}{(Total\ no.\ of\ F^{\Pi})} \tag{3.13}$$

3.3 Multivariate distance-based analytic framework for connectome wide studies

The proposed method in the previous section can be used to analyze high-dimensional data, provided by high-throughput technologies (e.g., DNA microarrays [17] and fMRI [functional magnetic resonance imaging] [18]), as an alternative to traditional dimension reduction or clustering methods. In this section, a framework based on multivariate analysis of variance for connectome-wide association [6] [2] will be presented.

The human brain connectome represents the complete set of neural connections and interactions in the brain [2]. One challenge for neuroscience is finding the relationship between variations within the connectome and environmental factors, such as disease states. MDMR [14] [6] pursued as a multivariate approach to assess associations between phenotypic and the multivariate connectome variations in the brain.

3.3.1 Brain voxel-wise analysis

fMRI measures brain activities by observing changes in blood oxygen level [18]. The human brain is structured into small cubic voxels (i.e., 3-dimensional units that embed the signals in brain scans), where the total number of voxels V is 25,000. Investigating connectivity for a large number of voxels (i.e., variables) requires mass-univariate statistical analysis. Such computations increase the potential for false positives. Multivariate methods have been explored as an alternative to univariate methods to determine associations of connectivity-phenotype. MDMR has been chosen for the following reasons.

1. The ability to examine more than one predictor at a time

2. No restriction on the distance metric since the Euclidean distance is not suitable for time series data in fMRI

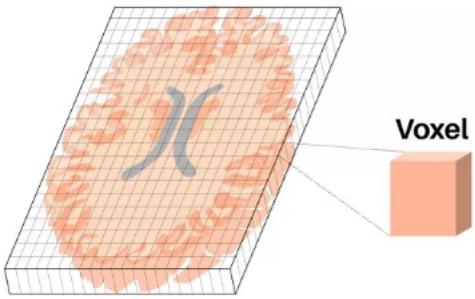

Figure 3.2: Voxel-based representation of the brain [1]

The presented work in [2] provides a framework for identifying phenotypic associations in the connectome using a two-step approach. In step one, the whole brain functional connectivity map is calculated (i.e., correlation between the voxel and all other voxels in same subject) for each voxel in the brain, then calculate the similarity between connectivity maps of all possible parings of participants using spatial correlation. Doing so results in an nxn matrix (n = number of participants). MDMR is subsequently used on each voxel to test if a variable of interest (e.g., health state) is associated with the observations. Figure 3.4 represents data format for fMRI spatio-temporal data collected for each voxel.

3.3.2 MDMR-based Connectome-wide association studies

The algorithm utilized in connectome-wide association studies commences with the assessment of subject-level connectivity using Pearson correlations. Participants' individual data sets are used to calculate the correlation between each voxel and all other voxles in the participantâĂŹs brain. The output of this step is a VxV correlation matrix, where

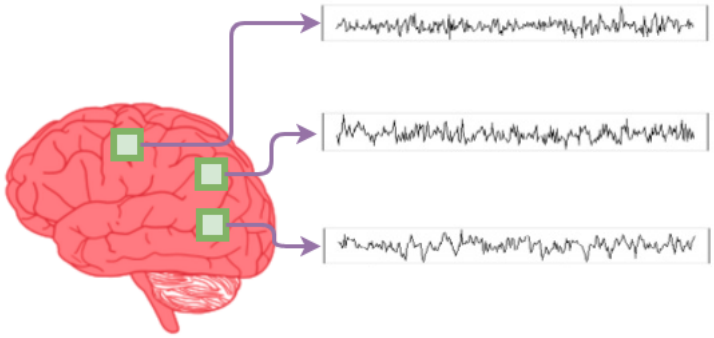

Figure 3.3: Illustration of fMRI spatio-temporal data in voxels

V is the number of voxels. Pearson correlation calculated using Equation 3.14.

$$\rho X, Y = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{3.14}$$

Next, individual differences in functional connectivity for each voxel must be calculated. The distance calculated for each voxelâĂŹs correlation between all possible pairings of participants is calculated using distance metric $\sqrt{2(1-r)}$, where r is the Pearson correlation. This metric ranges between 0 and 2. The result is an nxn matrix distance for each voxel, where, for example, element represents the dissimilar whole-brain connectivity map between participants (e.g., i and j).

Finally, MDMR is applied to find the relationship between the predictor variables and the distance between participant observations obtained in the previous calculation for testing whether or not each voxel connectivity pattern is similar under identical conditions (i.e., within group) than under different condition (i.e., between groups). The proposed algorithm in [14] and [6] [19], namely MDMR by Zappala, can be used as a hypothesis testing method. The pseudo-F-statistic can be calculated based on GowerâĂŹs matrix $G = CAC$ where $C = (I - \frac{1}{n}11^T)$ -; n is total number of participants, **I** is the identity

matrix of size n; and **1** is a vector of n 1s. $A = (-\frac{1}{2}d_{ij}^2)$ so that matrix A is multiplied by C to centralize the data.

A standard multivariate regression model can be written as in 3.15, where X is the design matrix of size nxm (i.e, first column is 1s representing the intercept) and Y is the response matrix of size nxn representing the similarity matrix explained earlier and centered using Gower's form.

$$Y = X\beta + \epsilon \tag{3.15}$$

The hat matrix H can be calculated as $H = X(X^TX)^{-1}X^T$ with size nxn. In this way, the relationship between the predictor variables and the dissimilarities of observation can be found using the pseudo-F statistic 3.16, where the numerator corresponds to variations **between** groups and the denominator corresponds to the variation **within** groups.

$$F = \frac{tr(HG)/(m-1)}{tr[(I-H)G]/(n-m)} \tag{3.16}$$

To estimate null distribution, observations between groups are shuffled and F is re-calculated for each permutation. Then, the P-value is computed using Equation 3.13. Figure 3.4 illustrates an overview of this framework.

3.4 Multivariate distance-based regression for traffic analysis

The MDMR algorithm has been applied successfully in different domains that are characterized by high-dimensional and multivariate data for assessing the effect of specific factors

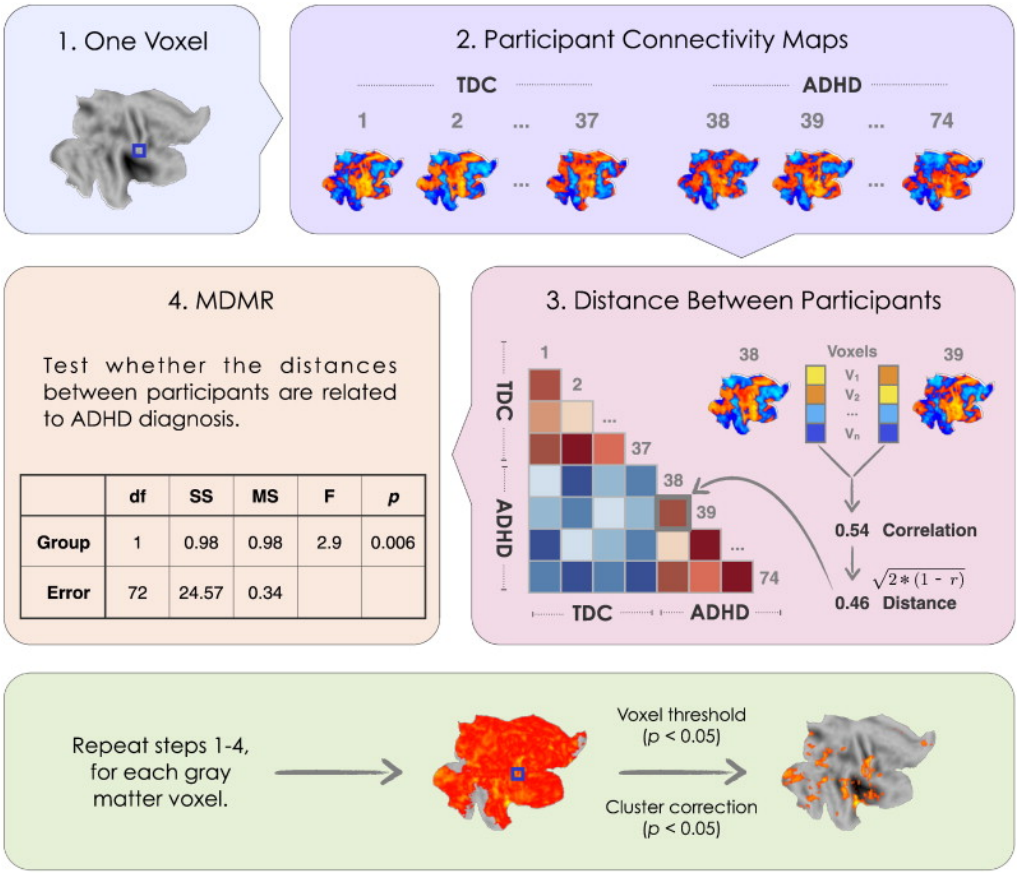

Figure 3.4: Illustration of MDMR-based framework for CWAS analytic [2]

or variables on observations. In transportation, studying the effect of specific conditions (e.g., weather, traffic hours, construction) on traffic is considered extremely important for discovering traffic patterns to aid in improving road design or, more importantly, mitigating traffic problems in specific places under specific circumstances. Applying the MDMR algorithm in this domain is a powerful tool for statistical tests of factors and their effects on traffic performance.

In this thesis, the MDMR algorithm is implemented on and applied to NPMRDS for studying traffic speed under different conditions. Several scenarios were used to test and validate the algorithm. For the first scenario, I-35 data from February 2017 was selected to study the time of day during peak (6 am - 10 am) and non-peak (10 am -

2 pm) hours. The second scenario studied the factor day of month and by comparing traffic performance during the same hours on different days. The obtained results for each scenario are detailed in this section.

3.4.1 Scenario 1 - peak versus non-peak traffic hours

In this scenario, traffic data was segregated into two sets: 1) speed values during peak hours (6 am - 10 am) and 2) traffic speed values during non-peak hours (10 am - 2 pm). Traffic data was collected during February 2017. Each set contains data for 28 days (or 56 observations). This scenario compared the effect of time of day on traffic speed to determine which road segments are affected during the peak hours and which remain constant throughout the entire day.

The MDMR algorithm was applied. Number of permutations used to estimate the null distribution was 10,000. Figure 3.5 illustrates the obtained P-value for each I-35 road segment, where segments with a p-value less than 0.0001 (i.e., threshold for rejecting null hypothesis) indicate that traffic behavior is different between peak and non-peak hours (colored in red). In other words, the null hypothesis was rejected for segments with P-value less than 0.0001. Road segments with P-value greater than 0.001 (colored in green) indicate that traffic behavior is similar during peak and non-peak hours.

The estimated F-distribution for I-35 Segment 622 is illustrated in Figure 3.6. In this case, the null hypothesis is rejected. P-value is 0.0001 and the original F-value is 3.573, where the obtained F-measure is positioned at the very right tail of the estimated F-distribution. In other words, traffic behavior on this segment is considered dissimilar between peak and non-peak hours (See Figure 3.6). The null hypothesis for Segment 613

is accepted, because P-value = 0.903 and F-value = 0.7749 (See Figure 3.7).

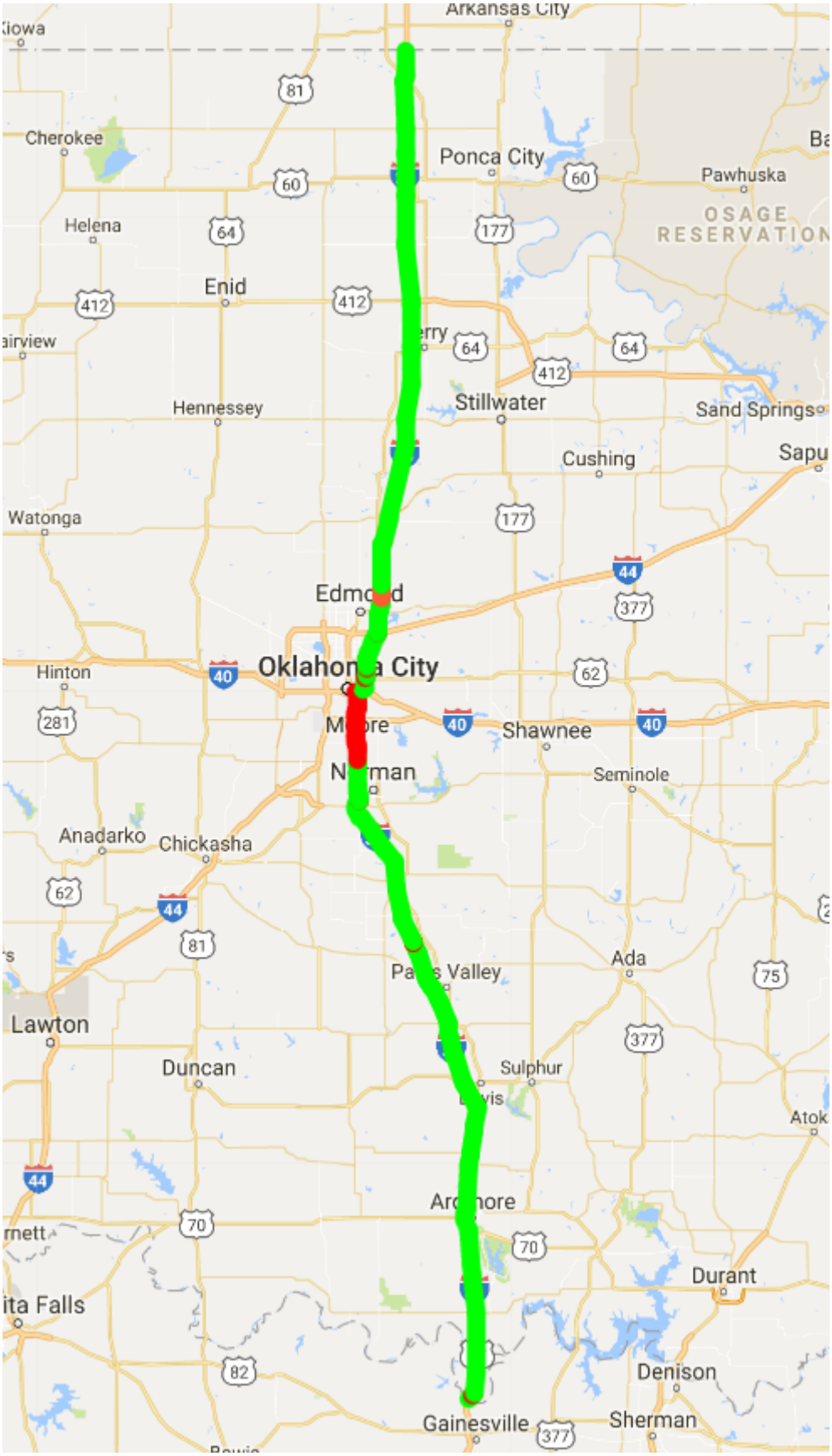

Figure 3.5: Obtained P-value for experiment 1, rejected hypothesis colored in red

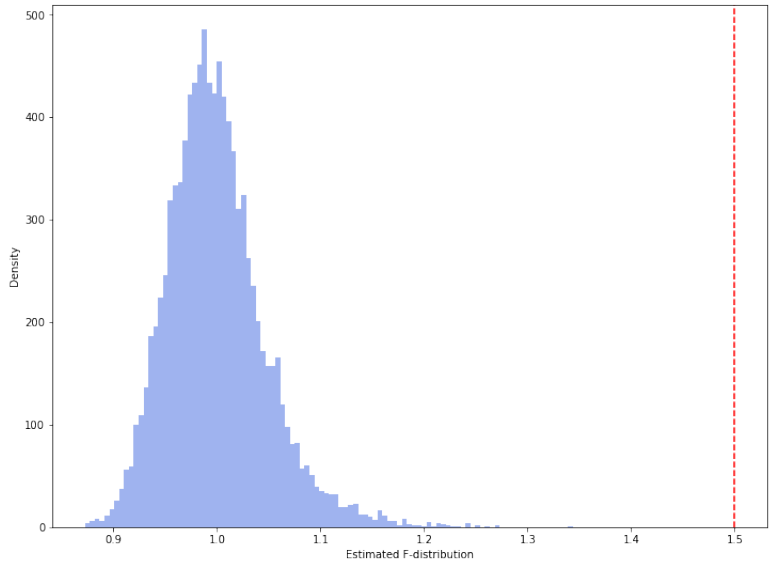

Figure 3.6: The estimated F-distribution for segment 622. Obtained F-value in red

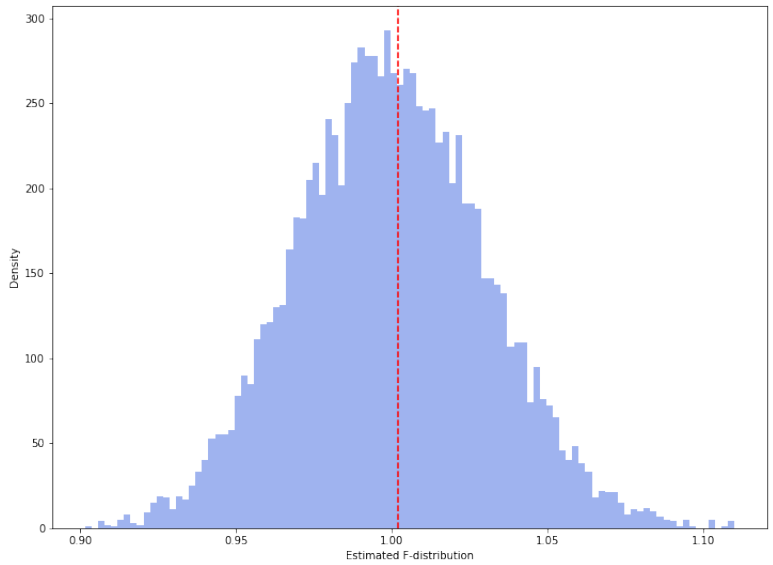

Figure 3.7: The estimated F-Value for segment 613, Obtained F-value in red

3.4.2 Scenario 2 - traffic in different days of the month

The objective the Scenario 2 was measuring traffic speed during the same traffic period (i.e, 10 a.m. - 2 p.m.). Data from the first group was collected February 1 through 14; data from the second group was collected February 15 through 28. Highway I-35 is composed of 185 segments (i.e., variables). The goal for this scenario was testing the framework by observing two groups that are similar except for the range of days in a given month. Like scenario 1, MDMR was applied on the data with 10,000 permutations. Figure 3.8 depicts obtained p-value for each segment.

3.4.3 Discussion

Results of Scenario 1 demonstrated that traffic performance tends to be different between peak and non-peak hours in segments located in Oklahoma City as illustrated in Figure 3.5, while traffic on other segments located outside of Oklahoma City region tended to have similar behavior under the studied factor (i.e., peak vs non-peak hours). To further demonstrate these results, speed values for segments with p-value less than the chosen 0.0001 threshold were compared with segments with P-value > 0.0001. Results are plotted in Figure 3.9 and Figure 3.10 , respectively. Figure 3.9 clearly shows a difference between time series during peak hours, where most values have an anomalous (congested) pattern; most time series during non-peak hours have a normal traffic behavior. In contrast, Figure 3.10 depicts a segment with a high P-value $= 0.9$ (i.e., the null hypothesis is accepted), showing speed values time series are superimposed over one another, which indicates no difference in traffic behavior on that particular segment during peak or non-peak hours.

3.4.4 Choosing the threshold P-Value

The chosen threshold of P-value is used to reject the null hypothesis of a studied segment. Choosing the correct value as a threshold is subjective to the studied domain. In the literature, Anderson, et al. [14] used the value of 0.01 to reject the null hypothesis in the ecological multivariate data domain. Zapala et al [6] used the value of 0.001 as a threshold to compare the obtained P-value for testing associations between gene expressions. Shehzed et al [2] studied the impact of health factors on associations between brain neurons and chose 0.0001 as a threshold for P-value to reject the null hypothesis. In the domain studied in this thesis (i.e., transpiration), we empirically determined the value 0.0001 as a threshold for P-value by comparing different threshold values (e.g., 0.1, 0.01, 0.001, 0.0001) and observing the speed time series under different threshold values as in Figure 3.9 and Figure 3.10 where we found using the threshold value 0.0001 is the best value in the traffic domain that characterized by NPMRDS dataset.

3.4.5 Calculation of complexity

Algorithmic and time complexity are considered the most important aspects of any algorithm when the scale of the application is enlarged. Accordingly, the complexity for each step of the algorithm was calculated. Correlation computation complexity was $O(V^2N)$ where V is number of segments of a road and N is total number of observations. Distance matrix for all segments can be calculated using $O(N^2V)$. Permutation is determined by $O(PNV)$. Although the algorithm is implemented sequentially, computation time can be significantly reduced by paralleling the code to be executed in a parallel way.

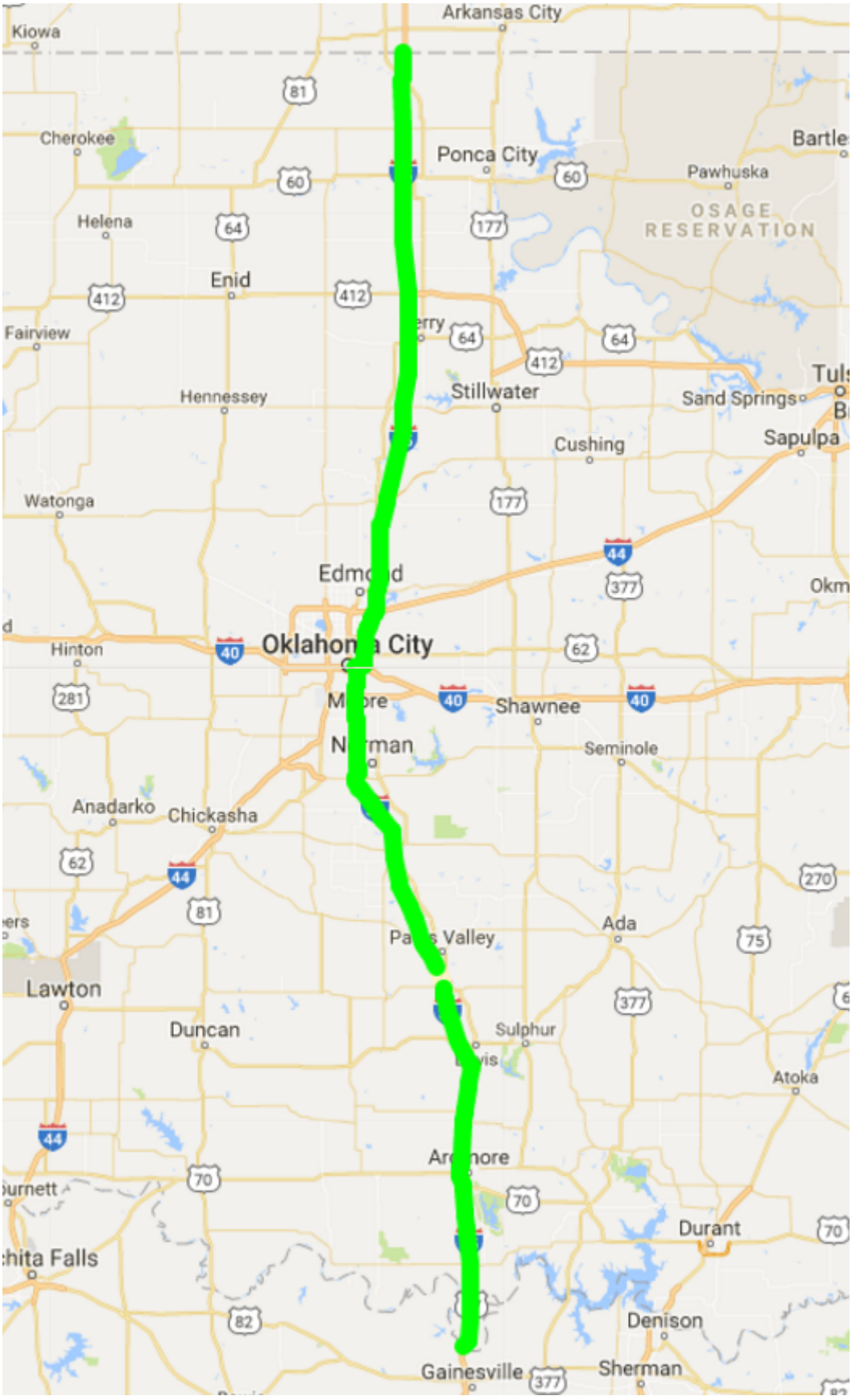

Figure 3.8: Obtained P-value for scenario 2, rejected hypothesis colored in red

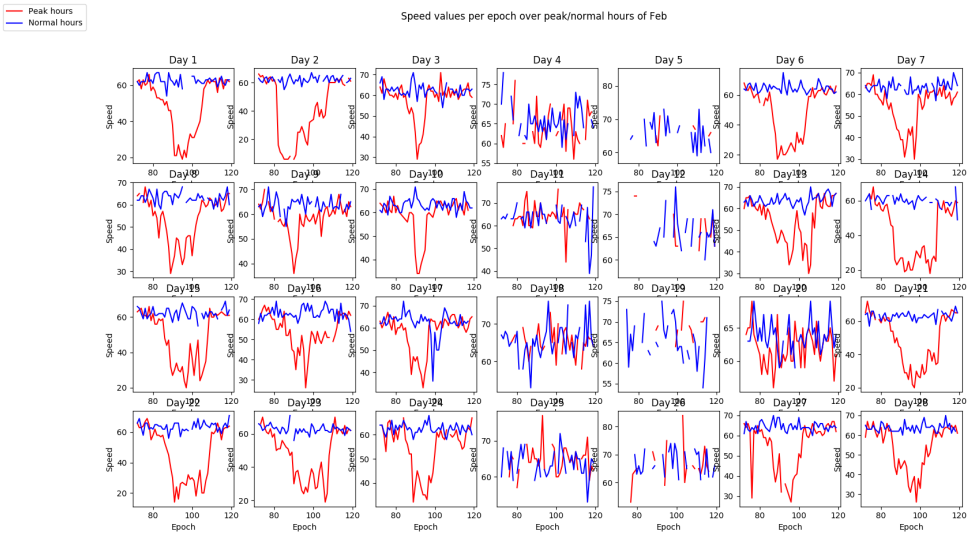

Figure 3.9: Speed values peak versus non-peak hours when P-value < 0.0001

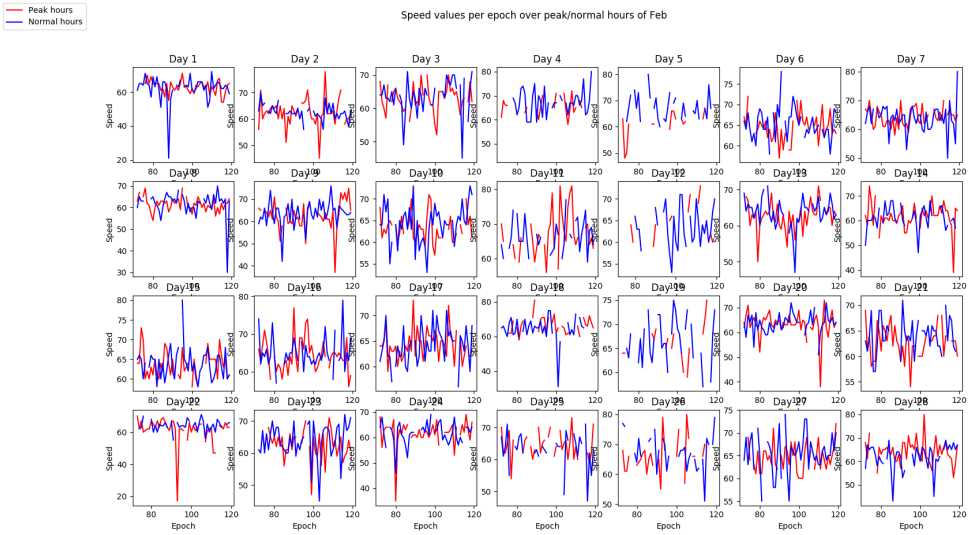

Figure 3.10: Speed values peak versus non-peak hours when P-value $\simeq 0.9$

3.4.6 Conclusion

In this chapter, a powerful tool was explored and successfully applied in the transportation domain for analyzing the effect of conditional factors (e.g., weather) on traffic performance, based on a MDMR. NPMRDS was used to apply and test the algorithm on traffic data. Two scenarios were used to observe the effect of different factors (i.e., time of day, day of month) on traffic performance. A threshold of P-value was chosen to reject the null hypothesis, which implies a relationship between the studied factor and the observed response variables.

Chapter 4

Clustering Time Series Using Analysis of Variance

In Chapter 3, a multivariate distance matrix regression (MDMR) algorithm was introduced; its use in the transportation domain demonstrated the ability to find similarities/dissimilarities of traffic performance in road segments under different factors. In this chapter, a novel clustering algorithm based on the permutation of F-statistic calculation is proposed. The algorithm can be used to cluster time series data for finding anomalous patterns.

4.1 Methodology

The foundation of the MDMR algorithm, explained in Chapter 3, is primarily based on calculating the F-statistic. Since the null distribution is unknown, random permutations are applied between groups to estimate the F distribution so that the null hypothesis can be rejected or accepted. The proposed clustering algorithm uses F-statistic to objectively

maximize the similarities within groups and maximize differences between different groups by shuffling elements between clusters to reach a maximum (i.e., desired) F-statistic value that represents the best segregation of elements that gives maximum variance. The following steps illustrate the algorithm.

1. Create two (or more) groups that represent desired clusters and randomly assign data elements to each group.

2. Calculate the distance matrix between elements, where dynamic time warping (DTW) [20] is used as a distance metric (i.e., measure similarities between two temporal sequences).

3. Obtain F-statistic value, as illustrated in Equation 3.16

4. Apply permutation between elements in the groups to find the solution that maximize the F-statistic. Brute-force search 1 and Integer Non-Linear Programming (INLP) are applied to find the optimal solution for maximum F-value.

5. The solution for highest F-statistic value is considered the optimal clustering of given elements

4.1.1 Integer Nonlinear Programming

The proposed brute-force algorithm 1 comes at an extremely high computational cost, and is, frankly, impractical. To overcome this issue and increase algorithm performance, optimization techniques are applied to facilitate a search for the optimal separation and to group elements so that maximum F-statistic is an objective function. This problem is classified as 4.1 integer nonlinear programming (INLP) and formulated, as follows:

Algorithm 1 MDMR clustering - Brute force search

Input: A set of of N data vectors where each vector $V = \{v_1, ..., v_N\}$
Number of clusters K
Output: Data vectors partitioned into clusters

```
F_max = 0, X_max = 0
d = calculate_distance_matrix(V)
for <i = 1 to 2^n − 1> do
    x = create_design_matrix(i)
    F = calculate_F_statistic(x, d)
    if F > F_max then
        F_max = F
        X_max = x
    end if
end for
return F_max, X_max
```

$$\max_{x} \quad f(x) = \frac{tr(HG)/(m-1)}{tr[(I-H)G]/(n-m)} \qquad (4.1)$$
$$x_i \in [1, m], \ \forall i \in n.$$

The problem described in 4.1 is considered a nonlinear integer problem due to the non-linearity nature of the objective F-statistic function and the decision variables or the design matrix X takes integer values. Genetic algorithm [21] is applied as an optimization solver to search for the optimal solution that maximizing the objective function.

Notably, different libraries and algorithms (e.g., Pyomo, PuLP, APmonitor and Matlab optimization tool) have previously been explored to solve the optimization problem. To date, only Matlab was able to solve the stated problems. Because other options have limitations, INLP problems were solved using only the Matlab optitool library [22].

4.1.2 Validation

To evaluate the quality of a clustering algorithm several scalar measurements can be used. These are typically categorized in two categories:

- External Index: a ground truth data set is used to compare the obtained clustering results by the algorithm with the labeled data to evaluate results [23] (e.g., Cluster purity [24], Rand index (RI) [25], F-measure [26], Entropy [26])

- Internal Index: quality of clustering structure is determined without the need for a ground truth dataset [23] (e.g., Sum of Squared Error [SSE] [27] [28] [23], Root-Mean-Squared Standard Deviation [RMSSTD] index [27]).

In this thesis, the Sum of Squared Error was used as an internal index to evaluate clustering results. The Sum of Squared Error is given in Equation 4.2. Sum of Squared Error represents the intra-cluster variance, the smaller SSE, the more consistent of clustering results.

$$SSE = \sum_{i=1}^{m} \sum_{j=1}^{n} \parallel x_{ij} - C_i \parallel^2 \tag{4.2}$$

m: is number of clusters

n: number of elements in a cluster

x_{ij} is a time series j in cluster i

C_i: the centroid of cluster i, the centroid of a cluster can be found using Algorithm 0

Algorithm 2 Find the centroid of a cluster

Input: Set of of n data vectors that represents a cluster where each vector $V = \{v_1, ..., v_n\}$
Output: Vector of points represents the centroid time series

1: $distance_matrix = Calculatedistancematrixbetweenalltimeseries$
2: $total_distance = []$
3: **for** <$i = 1 \ \ to \ \ n$> **do**
4: $\quad total_distance.append(sum(distance_matrix[i])$
5: **end for**
6: Return time series with minimum distance between all other time series in the same cluster as the centroid time series

4.2 Results

The proposed algorithm was implemented in Python and tested on NPMRDS. Several scenarios were performed on I-35 road segments data. This section discusses several scenarios and results after applying the proposed MDMR clustering algorithm on the data.

4.2.1 Scenario 1 âĂŞ Peak and non-peak Hours

In Scenario 1, the proposed algorithm was applied to cluster time series values taken from road segment 642 of I-35. Data was collected during peak traffic hours and non-peak hours during the first 10 days of February 2017. Total number of time series is 20. To compare results, a K-Means algorithm [29] was used to cluster the data. Figures 4.1 and 4.2 depict the results of MDMR and K-Means clustering results, respectively.

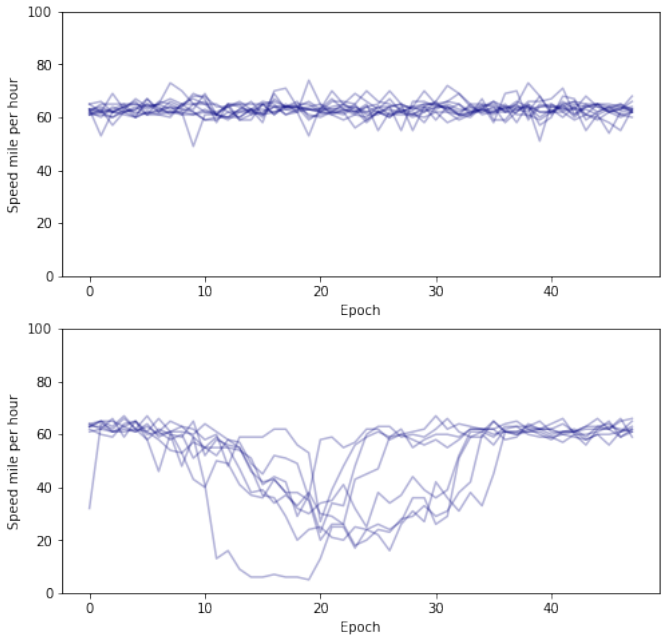

Figure 4.1: MDMR-based clustering for segment 642

Comparing Figure 4.1 and 4.2, we see that the two clustering algorithms gave identical

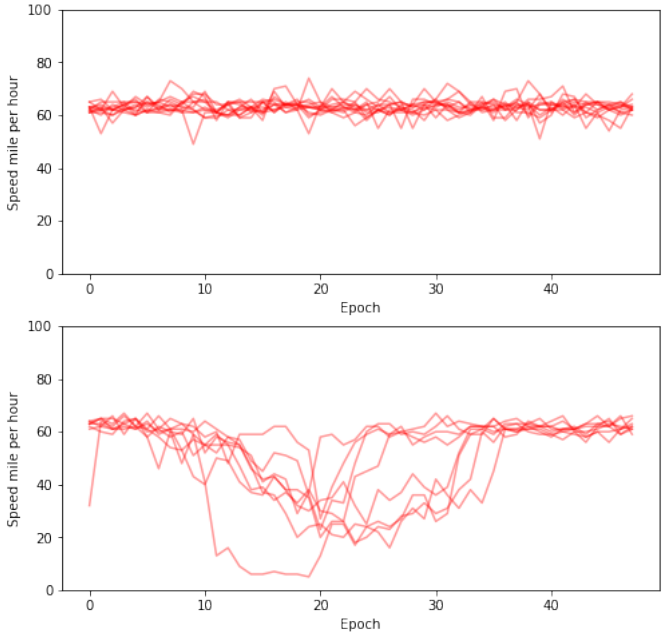

Figure 4.2: K-Means clustering for segment 642

results. To evaluate the quality of clustering algorithm the Sum of Squared Error was applied, as explained in Equation 4.2, as an internal index. The obtained Sum of Squared Errors for both the proposed clustering algorithm and K-means is $SSE = 9102$, where Centroids for both algorithms are obtained using Algorithm 2.

4.2.2 Scenario 2 âĂŞ Whole Day Clustering Over a Month

In the previous scenario, time series data were collected over specific hours (peak and non-peak) each time series comprised of 48 points. In Scenario 2, grouping data over a time frame of the day was substituted for time series as a whole day of a specific month. Figure 4.3 depicts the results of clustering 28 time series taken over February, each time series represent one day of February (i.e., 288 points).

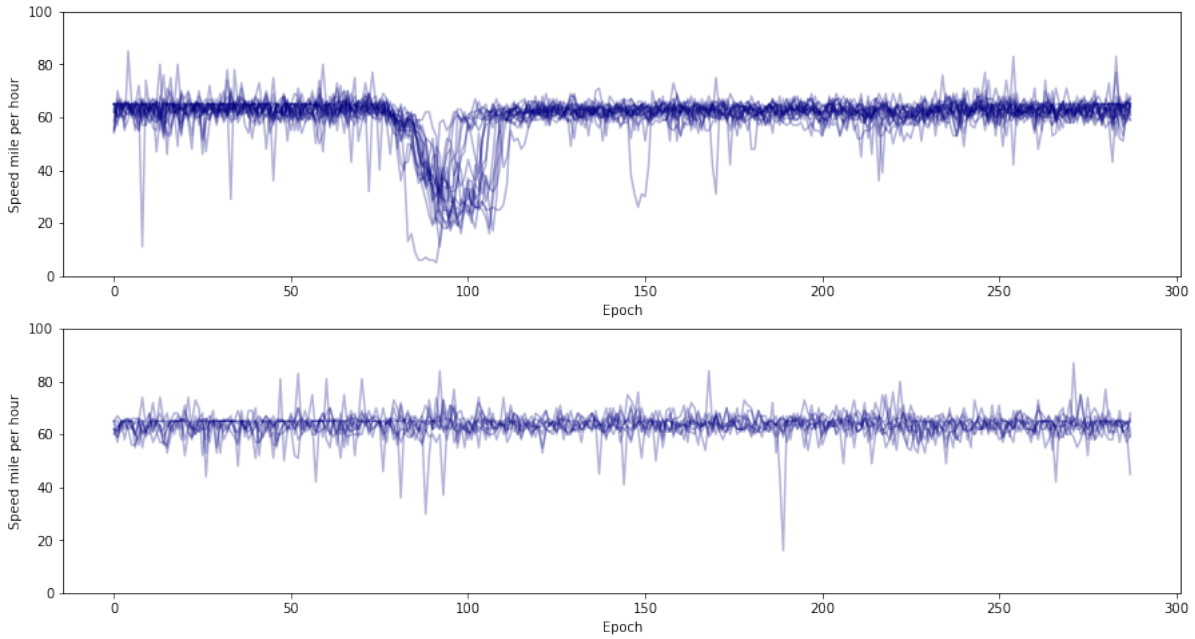

Figure 4.3: MDMR clustering for one segment over 24 hours over February

For comparison, identical data were clustered using K-means algorithm, as illustrated in Figure 4.4. Results provided by the proposed clustering algorithm gave comparable results to the results obtained by K-means revealing the applicability of this algorithm for clustering time series data. We evaluated the clustering algorithm using Sum of Squared Error (SSE). The obtained SSE for the proposed algorithm 64551.0, same SSE obtained for K-means 64551.0.

The same data was clustered into three groups instead of two. We obtained clustering

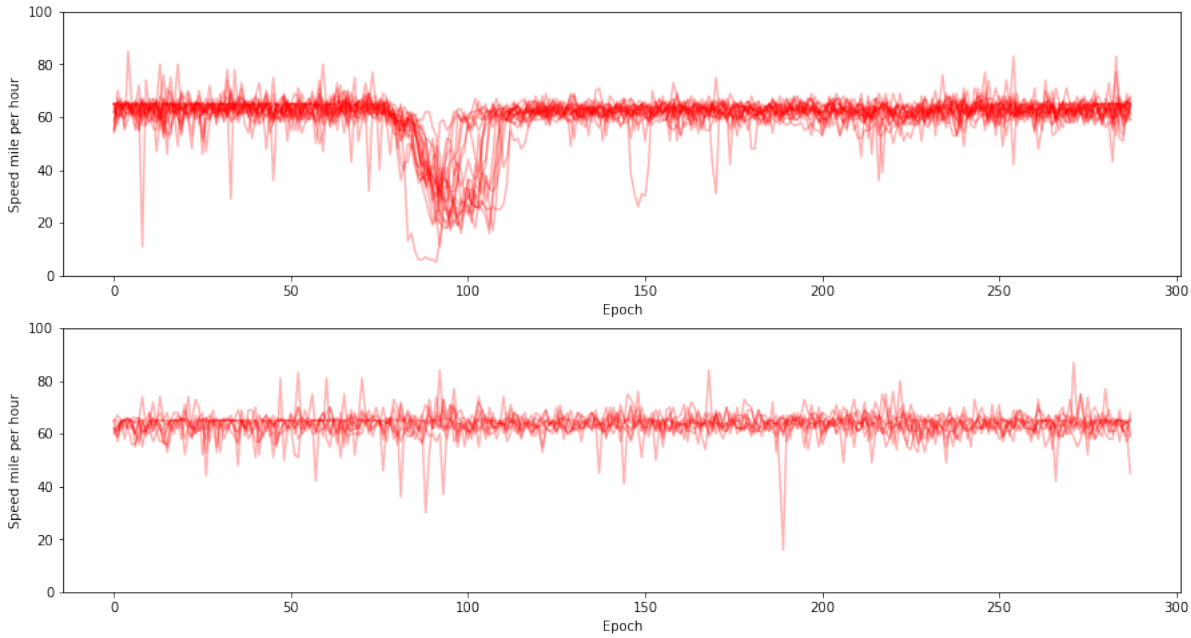

Figure 4.4: K-means clustering for one segment over 24 hours over February

results as shown in Figure 4.5 and by using K-Means as in Figure 4.6. The obtained SSE for our algorithm is 66416, while K-Means with SSE = 61023, meaning the K-means clustering results were better than our algorithm in this case.

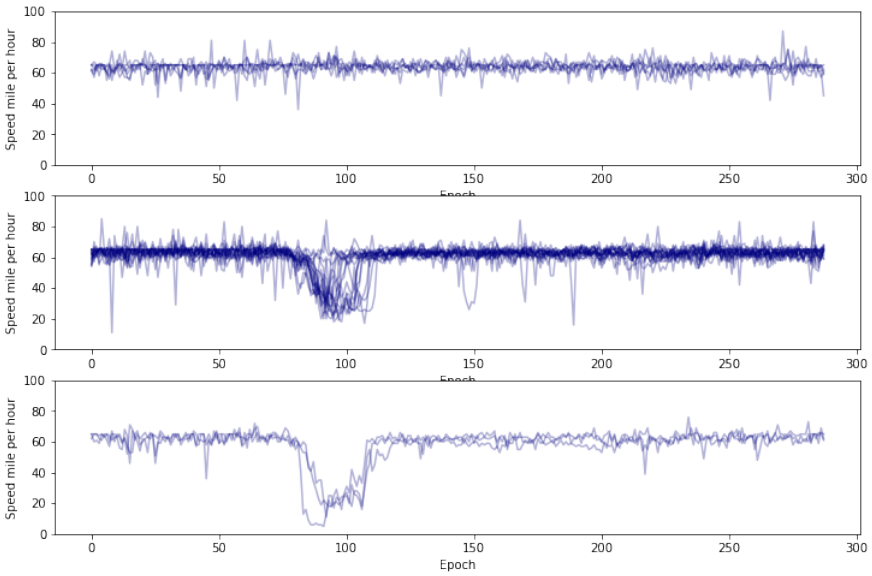

Figure 4.5: MDMR clustering to three clusters. One segment over 24 hours over February

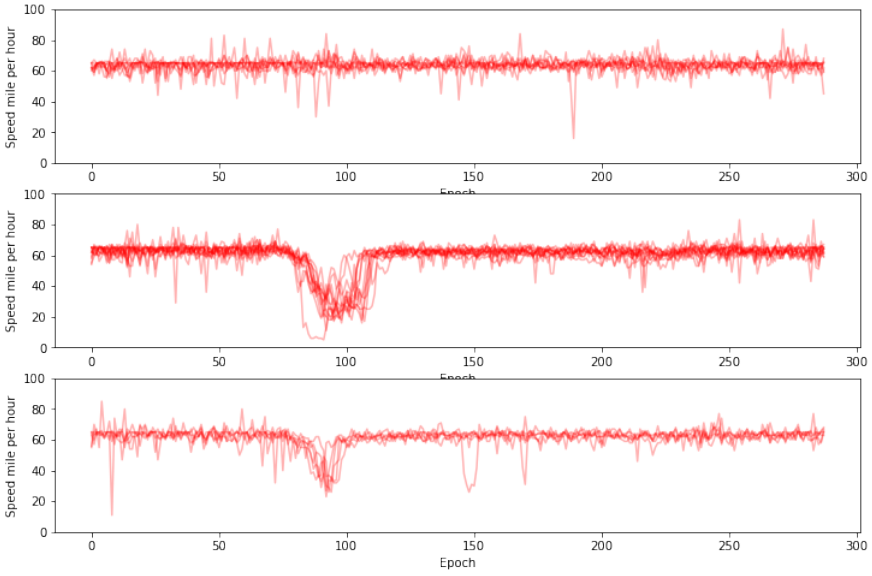

Figure 4.6: K-Means clustering to three clusters. One segment over 24 hours over February

4.2.3 Scenario 3 - Clustering of concatenated multiple days

Scenario 3 was based on clustering time series traffic data taken from NPMRDS. Each time series represents the concatenation of three subsequent days during February and March 2017. Total number of observations were 20 time series. Figure 4.7 depicts clustering for the aforementioned data. Data were collected only during weekdays. Congestion is periodic for all time series, where a clear separation between clusters is harder to obtain. Obtained Sum of Squared Error is 101478. Figure 4.8 illustrates the clustering result for the same data using K-means, wherein all time series are clustered into one cluster with SSE = 106806

Another scenario was conducted using the same data in addition to weekends days. Figure 4.10 depicts clustering results, where a more accurate separation between congested time series and normal time series was made. The obtained SSE for the porposed algorithm is equal to 147013.0 outperforming K-means Figure 4.9 with SSE = 152132.

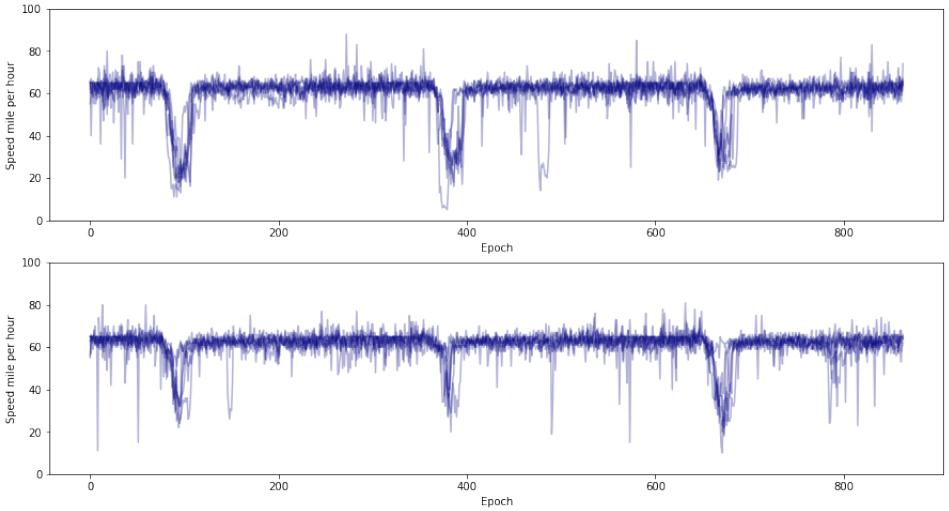

Figure 4.7: Clustering for one segment over February and March for three concatenated days per time series, weekends are not considered

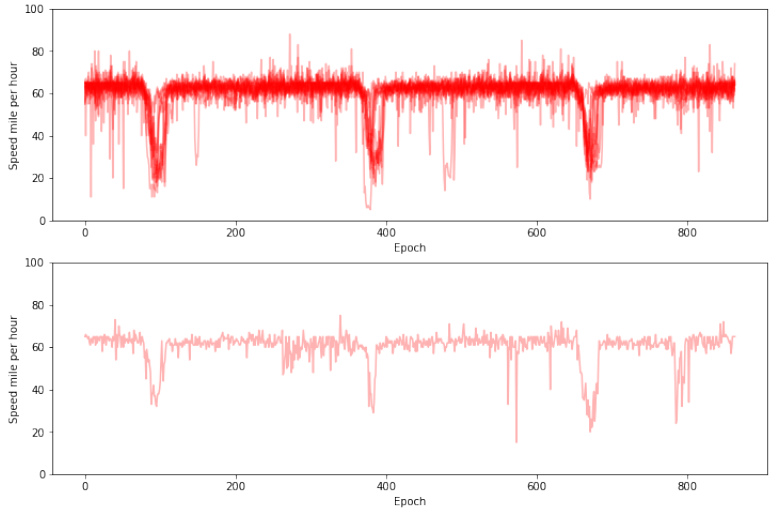

Figure 4.8: K-means clustering for one segment over February and March for three concatenated days per time series, weekends are not considered

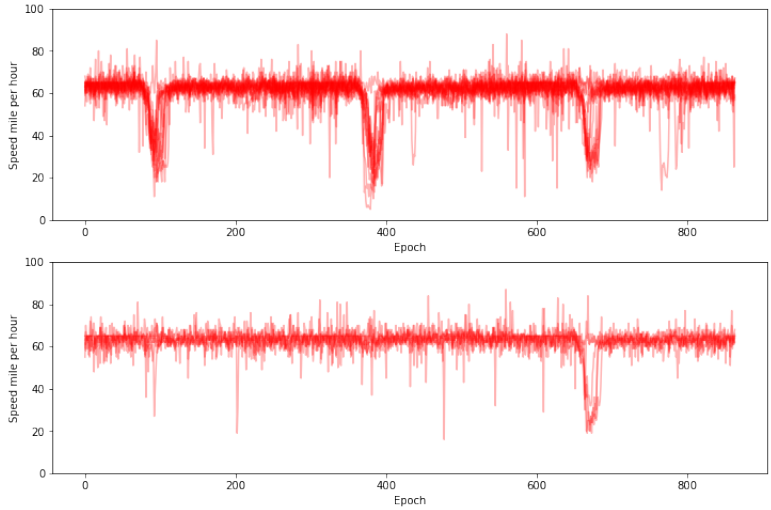

Figure 4.9: K-means Clustering for one segment over February and March for three concatenated days per time series, weekends are considered

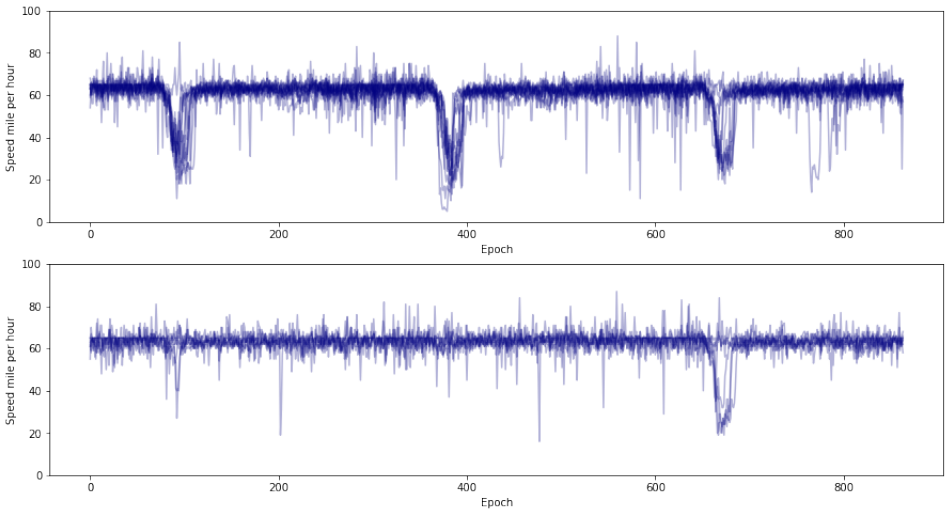

Figure 4.10: Clustering for one segment over February and March for three concatenated days per time series, weekends are considered

4.3 Multivariate Distance Matrix Regression Clustering Algorithm

A univariate time series clustering algorithm was proposed in the previous section. Several applications and domains produce multivariate time series (i.e., each observation has more than one variable). Multivariate time series clustering is difficult because conventional time series distance metrics (e.g. DTW) work exclusively with univariate time series. The following two methodologies can be applied to cluster multivariate time series:

- Multivariate Distance Matrix Regression method. The interaction and effect between variables in an observation-level is calculated using correlation metric (e.g., Pearson correlation). subsequently, a distance matrix for each variable is created between all observations for same variable. Figure 4.11 depicts clustering results by applying this methodology, clustering is applied on a segment 642.

- Apply a distance metric that can find the distance between mutlivariate time series data. A modified version Dynamic Time Warping (DTW) may apply [30].

4.4 Discussion

Results in the previous section demonstrate that the proposed clustering algorithm is suitable for clustering time series data with a comparable results with K-means clustering time series. Anomaly detection is a favorable outcome of the proposed clustering algorithm for detecting time series with anomalous patterns by following the assumptions

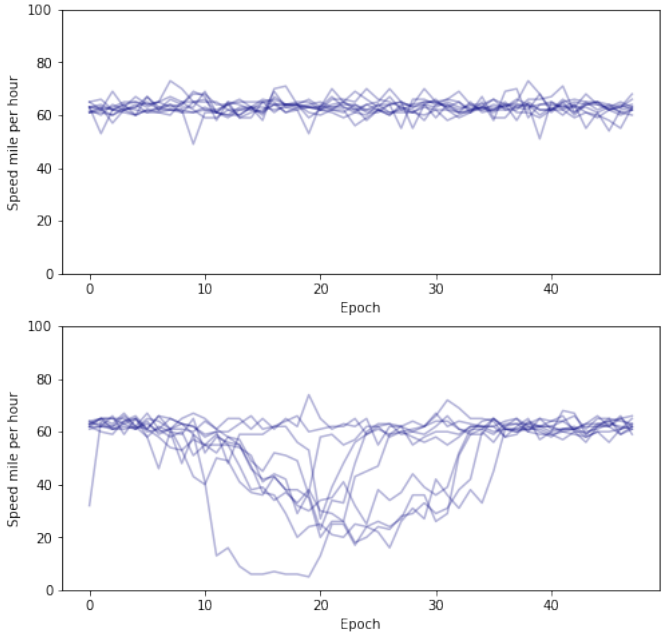

Figure 4.11: Multivariate Time Series Clustering. One of I-35 segment is clustered

that normal data is clustered together and that time series with anomalous patterns can form other clusters.

The proposed clustering algorithm has the following limitations for anomaly detection:

1. Number of clusters should be determined prior to data clustering.

2. The proposed algorithm is suitable for univariate time series. For multivariate time series, a distance metric that can measure the distance between multivariate time series must be used [30]

4.5 Conclusion and Future Works

A novel clustering algorithm based on analysis of variance was proposed. The objective function is defined by the F-statistic, wherein the goal is to determine the value of the design matrix X that maximizes the F-statistic to obtain the optimal separation between

groups. Optimal solution is obtained by applying permutation between groupsâĂŹ elements. Two search methods applied brute force and integer nonlinear programming methods to find the optimal solution for maximizing F value. The algorithm was then implemented in Python and tested on NPMRDS data to cluster traffic speed values over time. Several scenarios were conducted. Obtained results were compared with K-means clustering and demonstrated that results are comparable to K-means.

Chapter 5

Conclusion

In this work, a powerful tool based on MDMR was presented to analyze the effect of independent variables (or factors) on observations. The proposed framework was applied and successfully tested using NPMRDS in the transportation domain to study and compare traffic performance under various conditions (e.g. traffic hours, construction, weather). Departments of transportation can use this tool to make valid statements about traffic conditions and to aid designers in making decisions about road design improvements and ways to increase traffic safety. A novel clustering algorithm was proposed in this thesis to cluster time series data and for use as anomaly detection. The clustering algorithm is based on the F-statistic with a goal of optimizing this measure by finding the best solution for an optimal F-value. A brute force and genetic algorithm optimization were applied to find an optimal clustering solution. The algorithm was tested using NPMRDS, and results were compared with K-means. Preliminary results show that the proposed clustering algorithm provided accurate results, and that it can be used for time series clustering as well as for anomaly pattern detection.

5.1 Future Works

- Conduct more scenarios on the proposed MDMR by considering additional variables and studying their effects on traffic performance (e.g., effects of fog on traffic performance for specific road segments). Results should be validated by department of transportation information.
- Compare the proposed clustering method with other clustering algorithms
- Utilize and compare different distance metrics in the algorithm.
- Explore distance metric for multivariate time series data

References

[1] G. Torre, Knowingneurons, J. Frohlich, and J. Chen, "The brain's building blocks: Of protons and voxels," Sep 2017. [Online]. Available: http://knowingneurons.com/2017/09/27/mri-voxels/

[2] Z. Shehzad, C. Kelly, P. T. Reiss, R. C. Craddock, J. W. Emerson, K. McMahon, D. A. Copland, F. X. Castellanos, and M. P. Milham, "A multivariate distance-based analytic framework for connectome-wide association studies," *Neuroimage*, vol. 93, pp. 74–94, 2014.

[3] 2017. [Online]. Available: http://rwis.tulsa.ou.edu

[4] M. Williams, D. Cornford, L. Bastin, R. Jones, and S. Parker, "Automatic processing, quality assurance and serving of real-time weather data," *Computers I& Geosciences*, vol. 37, no. 3, pp. 353 – 362, 2011, geoinformatics for Environmental Surveillance. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098300410002554

[5] K. Kaushik, E. Sharifi, and S. E. Young, "Computing performance measures with national performance management research data set," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2529, pp. 10–26, 2015. [Online]. Available: https://doi.org/10.3141/2529-02

[6] M. A. Zapala and N. J. Schork, "Multivariate regression analysis of distance matrices for testing associations between gene expression patterns and related variables." *Proc Natl Acad Sci U S A*, vol. 103, no. 51, pp. 19 430–19 435, 2006. [Online]. Available: http://dx.doi.org/10.1073/pnas.0609333103

[7] FHWA, 2017. [Online]. Available: https://connectdot.connectsolutions.com/p13dy8kw6mr

[8] D. Akin, V. P. Sisiopiku, and A. Skabardonis, "Impacts of weather on traffic flow characteristics of urban freeways in istanbul," *Procedia - Social and Behavioral Sciences*, vol. 16, no. Supplement C, pp. 89 – 99, 2011, 6th International Symposium on Highway Capacity and Quality of Service. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042811009785

[9] F. Soriguera, I. MartÃŋnez, M. Sala, and M. MenÃľndez, "Effects of low

speed limits on freeway traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 77, no. Supplement C, pp. 257 – 274, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X17300396

[10] J. M. Baldasano, M. Gonçalves, A. Soret, and P. Jiménez-guerrero, "Air pollution impacts of speed limitation measures in large cities : The need for improving traf fi c data in a metropolitan area," *Atmospheric Environment*, vol. 44, no. 25, pp. 2997–3006, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.atmosenv.2010.05.013

[11] K. Sigakova, G. Mbiydzenyuy, and J. Holmgren, "Impacts of traffic conditions on the performance of road freight transport," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2947–2952.

[12] R. C. Mittelhammer, "Hypothesis-testing methods," in *Mathematical Statistics for Economics and Business*. Springer, 1996, pp. 595–675.

[13] L. F. Egeren, "Multivariate statistical analysis," *Psychophysiology*, vol. 10, no. 5, pp. 517–532, 1973.

[14] M. J. Anderson, "A new method for non-parametric multivariate analysis of variance." *Austral Ecology*, vol. 26, pp. 32–46, 2001.

[15] D. M. Lane, "Analysis of variance." [Online]. Available: http://onlinestatbook.com/2/analysis_of_variance/ANOVA.html

[16] [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda3665.htm

[17] E. M. Southern, "Dna microarrays: history and overview," *DNA arrays: Methods and Protocols*, pp. 1–15, 2001.

[18] S. A. Huettel, A. W. Song, and G. McCarthy, *Functional magnetic resonance imaging*. Sinauer Associates Sunderland, 2004, vol. 1.

[19] B. H. Mcardle, M. J. Anderson, S. Ecology, and N. Jan, "Fitting Multivariate Models to Community Data : A Comment on Distance-Based Redundancy Analysis FITTING MULTIVARIATE MODELS TO COMMUNITY DATA : A COMMENT ON DISTANCE-BASED REDUNDANCY ANALYSIS," vol. 82, no. 1, pp. 290–297, 2014.

[20] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, pp. 1–23, 2008.

[21] y. Heinz Mühlenbein and Thilo Mahnig, "Mathematical analysis of evolutionary algorithms for optimization."

[22] "Matlab optimization toolbox," <The year of your version, you can find it out using ver>.

[23] S. Aghabozorgi, A. Seyed, and T. Y. Wah, "Time-series clustering âĂŞ A decade review," *Information Systems*, vol. 53, pp. 16–38, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.is.2015.04.007

[24] S. C. Sripada, "Comparison of Purity and Entropy of K-Means Clustering and Fuzzy C Means Clustering," *Indian Journal of Computer Science and Engineering*, vol. 2, no. 3, pp. 343–346, 2011. [Online]. Available: http://www.ijcse.com/docs/IJCSE11-02-03-105.pdf

[25] M. Chiş, S. Banerjee, and A. E. Hassanien, *Clustering Time Series Data: An Evolutionary Approach*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 193–207. [Online]. Available: https://doi.org/10.1007/978-3-642-01091-0_9

[26] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *International Journal of computers and communications*, vol. 5, no. 1, pp. 27–34, 2011.

[27] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 911–916.

[28] Q. Zhao and P. FrÃď nti, "Wb-index: A sum-of-squares based index for cluster validity," *Data & Knowledge Engineering*, vol. 92, no. Supplement C, pp. 77 – 89, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169023X14000676

[29] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[30] D. Cao and J. Liu, "Research on dynamic time warping multivariate time series similarity matching based on shape feature and inclination angle," *Journal of Cloud Computing*, vol. 5, no. 1, p. 11, 2016. [Online]. Available: http://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-016-0062-z

Appendix A

Python implementation of Multivariate Distance Matrix Regression.

```
def calCorrelationSubjectLevel(df):
    """
    This function calculate the temporal pearson correlation in a
       subject-level between all tmcs
    Each subject consists of number of TMCs where every TMC can be
       represented by a set of values
    representing the speed over time, so the data can be represented
       by tuples (tmc, time, speed).
    The correlation metric that will be used is Pearson correlation
    The output will be an array VxV where V is the number of TMCs


    Parameters
    ----------
    df : dataframe that contains the data


    Returns
    -------
    y : a VxV array of pearson correlation, where V is the number of
       TMCs
    Notes
    -----


    """
    tmc_list = df.tmc_code.unique()
    v = tmc_list.shape[0]
```

```
correlation_array = np.zeros((v, v))
bar_i = 0
# Define the counters for the output correlation array
vi, vj = 0, 0
bar = progressbar.ProgressBar(maxval=v*v, widgets=[progressbar.
    Bar('=', '[', ']'), ' ', progressbar.Percentage()])
bar.start()
# Sort the data based on time
df = df.sort_values(['road_order', 'day', 'time'], ascending=True
    )
tmc_order_list = []
tmc_code_list = []
for tmc_current in tmc_list:
    vj = 0
    current_instance = df[df['tmc_code'] == tmc_current].speed.
        values
    current_instance_order = df[df['tmc_code'] == tmc_current].
        road_order.values[0]

    tmc_order_list.append(current_instance_order)
    tmc_code_list.append(tmc_current)
    for tmc_next in tmc_list:
        next_instance = df[df['tmc_code'] == tmc_next].speed.
            values

        # Calculate the correlation
        p_correlation = pearsonr(current_instance, next_instance)
            [0]
```

```
                correlation_array[vi, vj] = p_correlation
                vj = vj + 1
                bar.update(bar_i + 1)
                bar_i = bar_i + 1
            vi = vi + 1
        bar.finish()
        return correlation_array, tmc_order_list, tmc_code_list

def calDissimilarity(cor_matrix, distance_metric='p'):
    """
    Calculate the dissimilarities between all possible tmc pairs
    The input will be a correlation matrix of the size [n,v]
    where n: is number of instances in all groups,
          v: is the number of tmcs (voxels)
    The distance metric to be used is Eucledian

    The output is a distance matrix with the size of [n,n]

    Parameters
    ----------
    cor_matrix : array of the correlation matrix [n,v]
    distance_metric: in default it takes eucedian, or the
       dissimalrities can be calculated from the correlation, just
       set
                    the value of this parameter to 'pearson'

    Returns
```

```
    -------
    y : [nxn] array of distance matrix
    Notes
    -----

    """
    n, v = cor_matrix.shape
    distance_array = np.zeros((n, n))

    for i, instance_i in enumerate(cor_matrix):
        instance_i = map(lambda l: 0 if np.isnan(l) == True else l,
           instance_i)
        for j, instance_j in enumerate(cor_matrix):
            instance_j = map(lambda l: 0 if np.isnan(l) == True else
               l, instance_j)
            if distance_metric == 'eucledian':
                dist = distance.euclidean(instance_i, instance_j)
            elif distance_metric == 'p':
                p_correlation = pearsonr(instance_i, instance_j)[0]
                dist = (2*(1 - p_correlation))**(1.0/2)

            distance_array[i, j] = dist
    return distance_array

def calGowerCenteredMatrix(A):
    """
    Gower centered matrix is used to centralize the distance matrix,
    A = (-1/2 dij**2),
```

```
    G = CAC,
    C = (I - 1/n*1*1') ; I is the identity [nxn], 1 is a vector (
        column) of n ones
    Parameters
    ----------
    A = (-1/2 dij**2),
    Returns
    -------
    G: Gower matrix
    Notes
    -----
    Equations are copied from the work find in this paper:
    A multivariate distance-based analytic framework for connectome-
        wide association studies
    """
    n = A.shape[0]
    I = np.identity(n)
    ones = np.ones((n, 1))
    C = I - 1/n * ones.dot(ones.transpose())
    G = C.dot(A).dot(C)

    return G

def calculateA(distance_matrix):
    """
    In order to calculate the Gower centralized dissimilarity matrix
        using Gower equation
    we need to calculate A,
```

```
    which is nothing but A = (-1/2 dij**2),

    Parameters
    ----------
    distance_matrix: [nxn] array
    Returns
    -------
    A: (-1/2 dij**2),
    Notes
    -----
    Equations are copied from the work find in this paper:
    A multivariate distance-based analytic framework for connectome-
       wide association studies
    """
    d_square = np.multiply(distance_matrix, distance_matrix)
    A = -0.5*d_square
    return A

def calulateH(X):
    """
    calculate the hat matrix, where it can be calculated as follow:
    H = X(X'X)^-1X'

    Parameters
    ----------
    X: [NxM] array, where N is total number of subjects, M is the
       number of predictor or regressor variables,
    and first column is always 1s, whose relationship to the
```

```
        dissimalarties matrix is of interest
    Returns
    -------
    H: hat matrix
    Notes
    -----
    Equations are copied from the work find in this paper:
    A multivariate distance-based analytic framework for connectome-
        wide association studies
    """
    Xt_X = np.dot(X.transpose(), X)
    Xt_X_inv = np.linalg.inv(Xt_X)
    X_Xt_X_inv = np.dot(X, Xt_X_inv)
    H = np.dot(X_Xt_X_inv, X.transpose())
    return H

def calPseudoFratio_v2(H, G, m, n):
    """
    Pseudo F statistic can be calculated:
    F = [tr(HG) / (m - 1)] / [tr[(I - H)G] / (n - m)]

    Parameters
    ----------
    G: Gower centralized matrix
    H: Hat matrix
    n: number of participants
    m: number of variables
    Returns
```

```
    -------
    F pseudo statistic
    Notes
    -----
    Equations are copied from the work find in this paper:
    Zappala paper
    """
    I = np.identity(n)
    num = np.trace(H.dot(G)) / (m - 1)
    denum = np.trace((I - H).dot(G)) / (n - m)
    F = num / denum
    return F

def calPvalueByPermutation_v2(H, G, m, n, F):
    """
    Apply permutation and calculate pseudo F-Ratio,
    The permutation should be applied on G matrix by randomly
       permuting rows and columns simultaneously
    P = number of F' >= F / total number of F' ---- (4)

    Parameters
    ----------
    G: Gower centralized matrix
    H: Hat matrix
    n: number of participants
    m: number of variables
    F: original F ratio
    Returns
```

```
    -------
    P-Value
    Notes
    -----
    Equations are copied from the work find in this paper:
    A multivariate distance-based analytic framework for connectome-
       wide association studies
    """
    # Calculate total number of permutation
    max_iteration = min(math.factorial(n), 10000)

    F_pi = np.zeros(max_iteration)
    for i in range(max_iteration):
        G_permuted = simultaneousPermutaion(G)
        F_pi[i] = calPseudoFratio_v2(H, G_permuted, m, n)

    p_value = F_pi[F_pi >= F].shape[0] / max_iteration

    return p_value, F_pi

def simultaneousPermutaion(G):
    # Get the size of the matrix
    n = G.shape[0]
    p = np.random.permutation(range(n))
    l = zip(range(n), p)
    permutation_list = list(set(tuple(sorted(t)) for t in l))

    G_permuted = G.copy()
```

```
    for i, j in permutation_list:
        G_permuted = swap(G_permuted, i, j)

    return G_permuted
```

Python code for time series clustering:

```
from tslearn.clustering import TimeSeriesKMeans
from tslearn.datasets import CachedDatasets
from tslearn.preprocessing import TimeSeriesScalerMeanVariance,
    TimeSeriesResampler
import tslearn.utils as util

print("k-means")
seed = 1
number_of_groups = 2
dba_km = TimeSeriesKMeans(n_clusters=number_of_groups, n_init=1,
    metric="dtw", verbose=True, max_iter_barycenter=50, random_state=
    seed)
timeseries_dataset = util.to_time_series_dataset(time_series)
y_pred = dba_km.fit_predict(timeseries_dataset)

# Plotting the clustering results
number_of_groups = 2
fig = plt.figure(0, figsize=(12,8))
for yi in range(number_of_groups):
    plt.subplot(number_of_groups, 1, 1 + yi)
    plt.ylim(ymax = 100, ymin = 0)
    for xx in timeseries_dataset[y_pred == yi]:
```

```
        plt.plot(xx.ravel(), "red", alpha=.3)
        plt.ylabel('Speed mile per hour')
        plt.xlabel('Epoch')
    plt.plot(dba_km.cluster_centers_[yi].ravel(), "b")



H = calulateH(X)
F = calPseudoFratio_v2(H, G, number_of_variables,
   total_number_of_participants)
# Obtain the best clustering that gives the maximum F-statistic
F_pi, max_X = calFvalueByPermutation_v2(X, G, number_of_variables,
   total_number_of_participants, F)
```

```
function f = objfun(x, G, m, n)
    %OBJFUN Objective function.
    H = calcH(x);
    I = eye(n);
    f = -(trace(H*G) / (m -1)) / (trace((I - H)*G) / (n - m));
end


function H = calcH(x)
    n = size(x,2);
    design_matrix = ones([n 2]);
    design_matrix(:,1) = x;
    H = design_matrix * inv(transpose(design_matrix)*design_matrix) *
        transpose(design_matrix);
end
```

```
n = size(G,1);
m = 2;


% Make a starting guess at the solution
x0 = zeros([1 n]);
for i = 1:((n)/2)
    x0(i) = 1;
end
f = @(x)objfun(x,G,m,n);
lb = zeros([1 n]);
ub = ones([1 n]) * (m - 1);
%options = optimoptions('@ga','MaxGenerations',300,'Display','none');
options = gaoptimset('Display', 'iter');
[x,fval] = ga(f,n,[],[],[],[],lb,ub,[], (1:n), options);
```