Statistical Modeling of Travel Time on the Motorway

Zhiwei Zhang

Kgs. Lyngby July 2006

Master Thesis Project

Abstract

Effective prediction of highway travel time is essential to many advanced traveler information and transportation management system. This thesis proposes 3 different prediction schemes to predict highway travel time in certain stretch of Denmark, using a linear model where the coefficients vary as smooth functions of the departure time, also the principle components and partial least squares regression. The methods are straightforward to implement and applicable to different circumstances.

Key words: Travel Time Prediction, Linear Regression, Time-Varying Coefficients, Time Series Analysis, Principle Components, Partial Least Squares.

Preface

This thesis was written as a part of my studies for a Master of Science degree at the Department of Informatics and Mathematical Modelling (IMM) of the Technical University of Denmark (DTU), under the supervision of Professor Klaus Kaae Andersen.

The project was carried out in the period from the 15th of January 2006 to the 15th of July 2006, credited as a 30 ECTS point project.

Acknowledgement

I would like to thank my supervisor Klaus Kaae Andersen for his great suggestions and advices. Without his help, it is unimaginable for me to complete this thesis smoothly and successfully. I also would like to appreciate my friends and colleagues Minze Su, Shenze Wu, Thomas Sørensen and Yuan He for all your support and assistance, and finally the seniors in Danske Vejdirektoratet for giving me the opportunity to do the project.

Contents

List of Abbreviations

EM-*Expectation Maximization*
CST-*Current Status Travel Time*
WLS-*Weighted Least Squares*
ACF-*Autocorrelation Function*
AR-*Autoregressive*
CCF-*Cross Correlation Function*
IRF-*Impulse Response Function*
WRLS-*Weighted Recursive Least Square*
PCA-*Principle Component Analysis*
PLS-*Partial Least Squares*
NIPALS-*Nonlinear Iterative Partial Least Squares*
MSE-*Mean Square Error*
OSCORESPLS-*Classical Orthogonal Scores Partial Least Squares*
TOD-*Time of Day*
Mcount5-*Mean of Number of Passing Vehicles per 5 Minutes*
GAMS-*Generalized Additive Models*
ROLS-*Recursive Ordinary Least Squares*
PCL-*Principle Component Loadings*
ANOVA-*Analysis of Variance*
CV-*Cross Validation*
OLS-*Ordinary Least Square*
UCL-*Upper Control Limit*
LCL-*Lower Control Limit*

Chapter 1

Introduction

Congestion has become a serious problem on many of the urban freeways around the world. The dynamic nature of the congestions makes trip planning difficult and subject to unpredictable consequences due to unknown or unforeseeable traffic events. In recent years, many strategies based on advanced transportation technologies have been proposed to promote more efficient use of the existing roadway networks in order to ease congestion. Many of these systems require, directly or otherwise, reliable prediction of travel times. Dynamic route-guidance, in-vehicle information, congestion management and automatic incident detection systems can all benefit from accurate and implementable travel time prediction techniques [1].

Not surprisingly, there has been a considerable amount of work on the subject of traffic forecasting, which show different prediction performances and explanation powers. The focus of this thesis is to compare several widely used modeling methods of the issue and propose insightful suggestions according to different conditions. Those methods include: Linear Regression with Time-Varying Coefficients, Principal Component and Partial Least Squares.

In this thesis, numeric data are obtained by double loop detectors on vehicle speed, and traffic flow aggregated over 30 seconds intervals. These equipments are properly installed over all highways in Denmark and such data are also called Automated Vehicle Identification (AVI) data. Although we have double loop detector data in mind when developing the methodology, the technique can be used for other forms of sensor data as long as reliable speed estimates can be derived from the direct sensor measurements. Data from probe vehicle or AVI technologies can also be seamlessly incorporated into the framework. Note that the travel time prediction here is based on any two subsequent points of a freeway network for any departure time.

The rest of the thesis is organized as follows. Chapter 2 presents the descriptive statistics of obtained data, which is helpful to identify statistical characteristics in order to select a suitable model. Chapter 3 describes the theory of prediction methods used here. Chapter 4 then states how we implement those methods to travel time prediction and build suitable models. Chapter 5 focuses on comparison among those methods with a collection of training dataset and test dataset as well as consequent conclusions. Finally, suggestions of future works will be addressed subsequently.

The names of loop detectors and their relative distances (from a certain starting point which is not on this map) are marked on the map; from which one can directly calculate the distance between any two loop detectors by taking the absolute value of subtraction of those relative distances.

2.1.1 Structure

The first several lines of original data are shown below:

Time	PN2_Count	PN2_Velocity	…	M0_Count	M0_Velocity
01-09-2005 06:00:00	7	101	…	6	97.67
01-09-2005 06:00:30	3	84	…	14	97.29
01-09-2005 06:01:00	4	99.5	…	11	98.55
01-09-2005 06:01:30	5	102.6	…	7	88.71
…	…	…	…	…	…

Table 2.1: Original data structure

The first column is the recording time and date for the data obtained from those 34 loop detectors, and the subsequent ones are traffic flows (count) and vehicle velocities corresponding to certain loop detector. Generally, there are 29883 rows and 69 columns data gathered through 63 weekdays of 3 consecutive months.

Not surprisingly, detector malfunction was always existed during the data collection process, which caused a large amount of missing data. The following table shows its description:

Month	Completely Missing Rows	Number of Missing Values Denoted as '-1'
September	22nd from 08:56 to 09:00	9664
October	The whole day of 14th	35965
November	18th from 09:36 to 10:00:30	35936

Table 2.2: Missing data description

Approximately 6% missing data are found from the original data, which suggests more sophisticated handling method.

2.2 Original Data Preprocessing

2.2.1 Determination of Training Set & Test Set

The so called 'training set' data used in this chapter and the following sections of model formulation are from September and October 2005 of first 17 induction loops, which are located at nearly half of the stretch. Meanwhile, the remaining data will be separated into two parts of test set compared to training set: one is from the other 17 induction loops of September and October, denoted as data of 'same period but different part of stretch'; another is from first 17 induction loops of November, denoted as data of 'same part of stretch but different period'. Those test sets are used for formulated model validation.

2.2.2 Analysis of Missing Data

Missing or corrupted loop data are unavoidable in practice and causes problems [1]. Since it causes lack of information and violation of the statistical assumption, it could be a potential threat to the validity of our research study. Therefore, it is necessary to use certain method to reach a minimum effect.

There are many of missing data handling methods, from simplest listwise deletion to complicate Hot – deck imputation; of which deletion methods are primarily not recommended, due to large amount of data lost and statistical power reduction [2]. As for the imputation approach, considering the missing data here are substantial, combined with the principles of avoiding subjective (Hot - deck) and biased estimation (Regression imputation), the so-called '*Expectation Maximization* (EM)' is used here to overcome the problem. This is done by following procedures:

- E step – Find the initial predicted values from a linear regression method;
- M step – Substitutes the missing data with the predicted values from E step to produce a covariance matrix and using maximum likelihood function, repeatedly estimate missing values;
- Repeat the above two steps until convergence between successive covariance matrices obtained [3].

2.2.3 Calculation of Response Variable

This thesis project is concentrating on prediction of highway travel time. Therefore, it is naturally that the response variable of the prediction model lies in travel time in appropriate scale.

The loop detector data recorded are number of passing vehicles (flow) and harmonic mean of velocities, both of which are aggregated over 30 seconds intervals. For simplicity, the time interval is increased up to 5 minutes, by taking harmonic mean of recorded velocities and summating flows within each 5 minutes, respectively:

$$V(d,l,t) = \frac{10}{\sum_{k=1}^{9} \frac{1}{V_{j+k,0.5}}} \quad (d \in D, l \in L, t \in T) \tag{2.1}$$

$$N(d,l,t) = \sum_{k=0}^{48} N_{j+k,0.5} \quad (d \in D, l \in L, t \in T) \tag{2.2}$$

where $V(d,l,t)$ and $N(d,l,t)$ denote the harmonic mean of velocities and flow aggregated over 5 minutes interval that was measured on day d at loop l at time t; $V_{j+k,0.5}$ and $N_{j+k,0.5}$ denotes the recorded velocity and flow aggregated over 30 seconds, respectively.

We then could also consider $V(d,l,t)$ as a matrix with entries $V(d,l,t)$ that was measured on day d at loop l at time t. Therefore, the variable $TT_d(a,b,t)$ denoting travel time from loop a to b starting at time t on day d can be calculated from V [4]:

Next, we define the *current status travel time* $CST_d(a,b,t)$ as follows:

$$CST_d(a,b,t) = \sum_{i=a}^{b-1} \frac{d_{i,i+1}}{V(d,i,t) + V(d,i+1,t)} \tag{2.3}$$

where $d_{i,i+1}$ denotes the distance from loop i to loop $i+1$ and it should be calculated by known flow and velocities. This is the travel time that would have resulted from departure from loop a at time t on day d when no significant changes in traffic occurred until loop b was reached. The important difference between those travel time variables is that $TT_d(a,b,t)$ requires information across all the stretch between loop a and loop b; whereas $CST_d(a,b,t)$ emphasizes available information on hand when starting at point a that could not reflect the varying conditions of the road[4].

According to the past observations, the purpose is to predict $TT_e(a,b,\tau+\delta)$ for a new day e and nonnegative 'lag' δ. This is the travel time between loop a and loop b departing at time $\tau+\delta$ where τ is the current time. However, the AVI data can only provide information to calculate $CST_d(a,b,t)$, and then we will use it as response variable for model formulating. This is practically useful, since the general on-hand information is enough for the driver regardless of some following changing road conditions.

2.3 Explanatory Variables Relationships

It is necessary to discuss relationships between explanatory variables from training set, from which one might obtain some important hints and results for future modeling and analysis.

2.3.1 Velocity VS Flow

The following figures are made in order to investigate the relationship between the velocity and flow of vehicles (amount of passing vehicles per unit time). By analyzing the figures for each of the consecutive loop data, one could locate the corresponding information of road conditions for different parts of the stretch, e.g. congested time of day or high-occupancy part of the road. Note that each of the flow data is aggregated over 5 minutes, whereas the speed statistics are harmonic means of observed velocities during every 5 minutes.

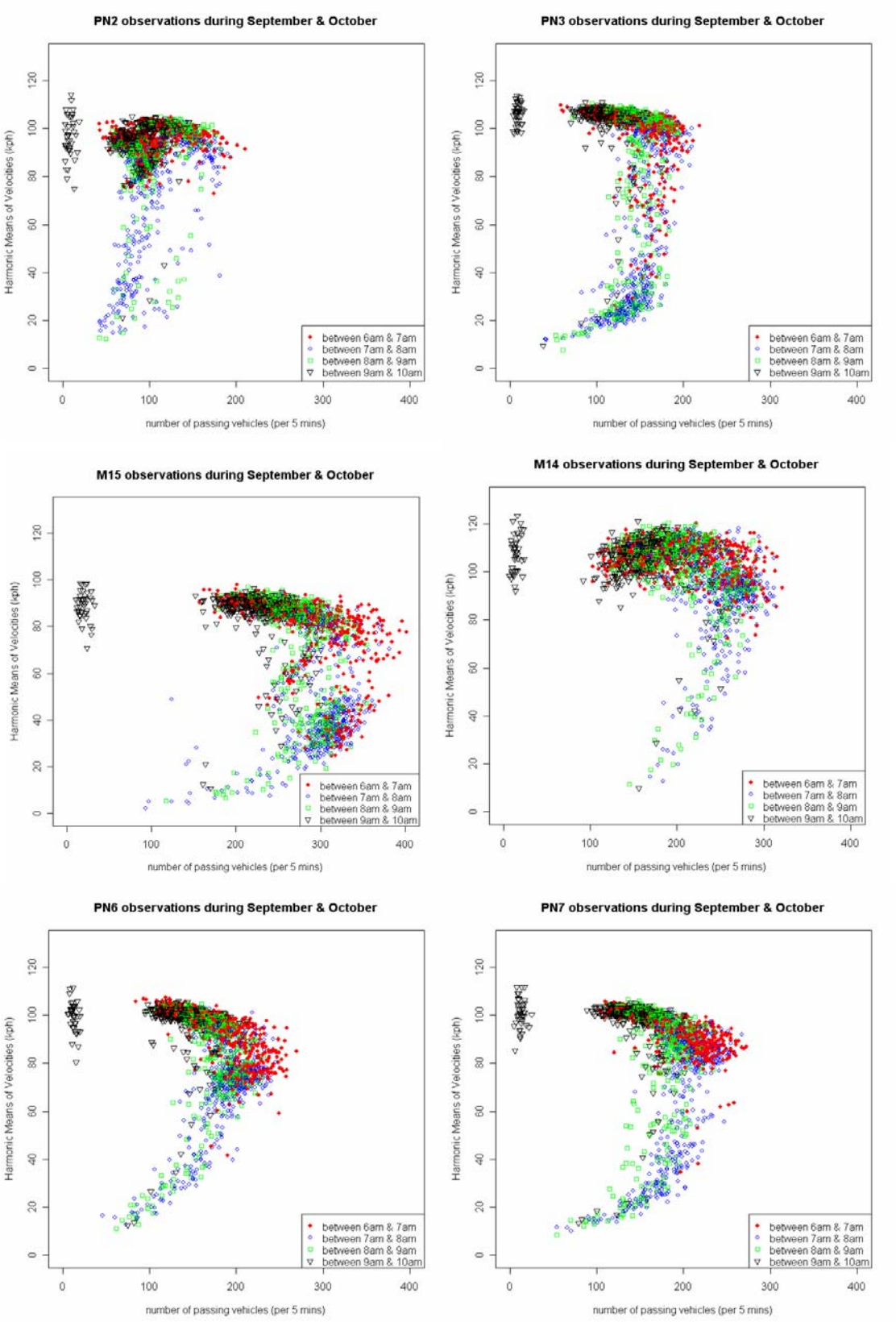


Figure 2.2: The relationship between the velocity and flow of vehicles observed in 42 weekdays during September and October 2005, using loop PN2 to PN7, in direction from South to North

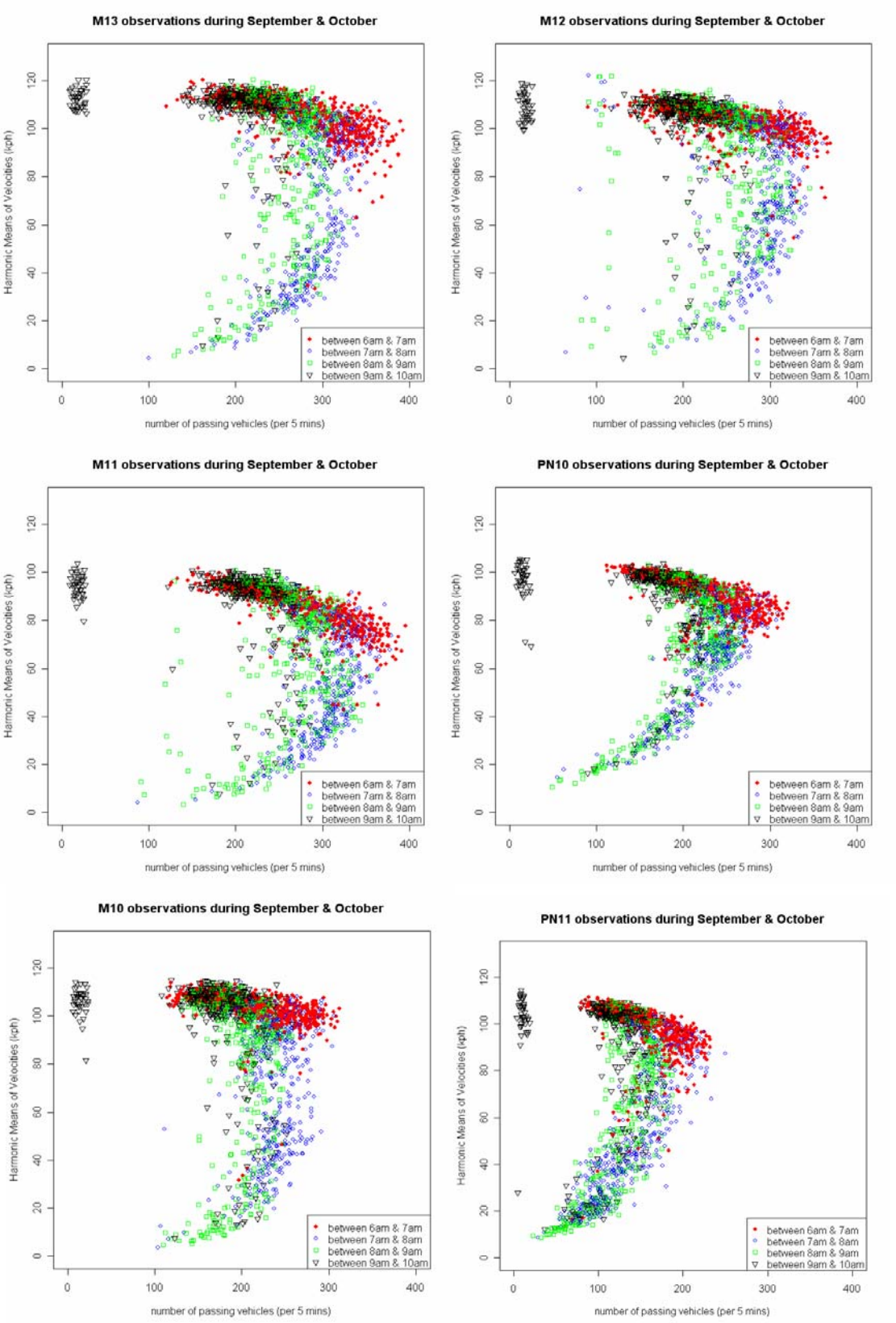


Figure 2.3: The relationship between the velocity and flow of vehicles observed in 42 weekdays during September and October 2005, using loop M13 to PN11, in direction from South to North

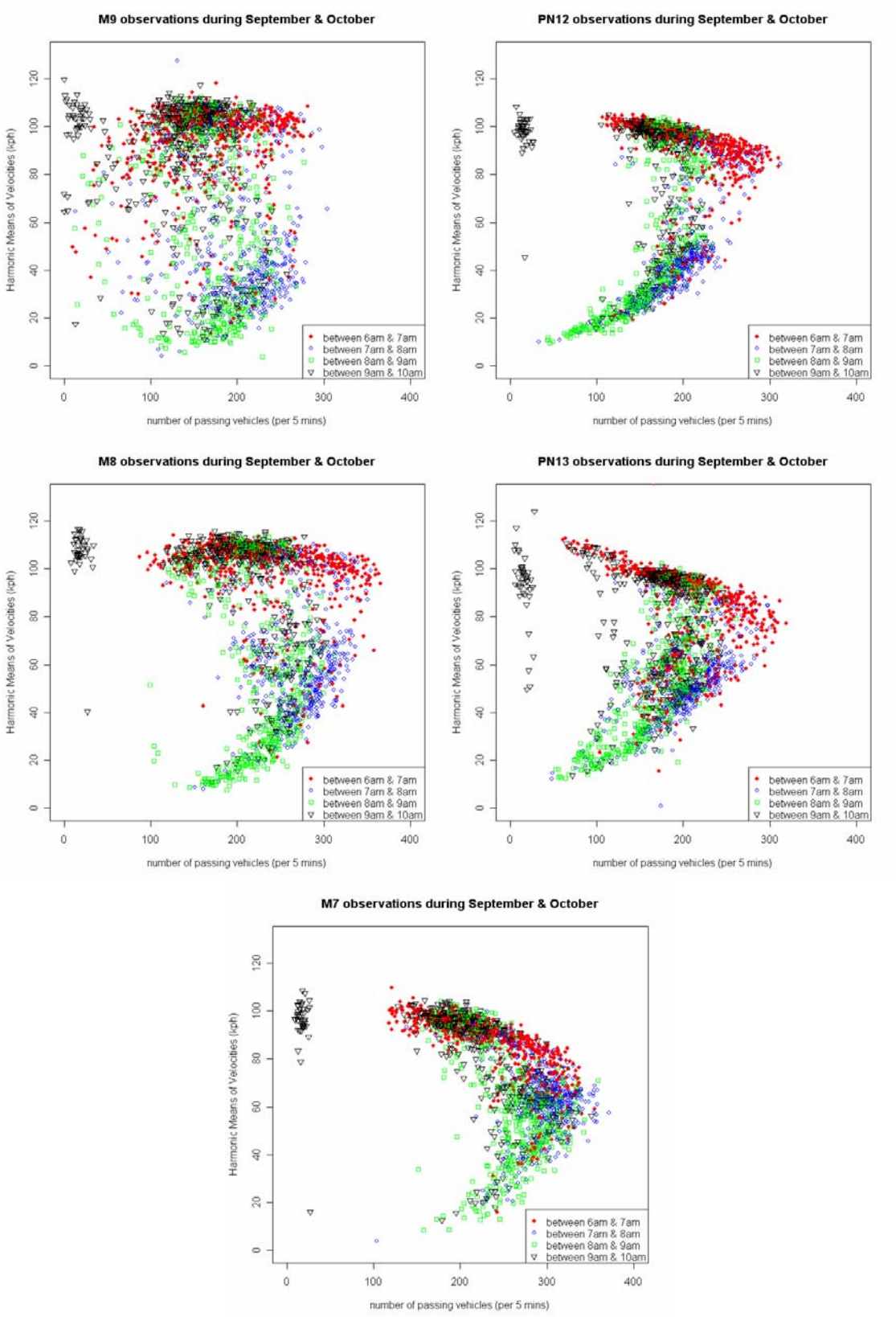


Figure 2.4: The relationship between the velocity and flow of vehicles observed in 42 weekdays during September and October 2005, using loop M9 to M7, in direction from South to North

According to the figures obtained above, first one can realize a free-flow regime, where the flow rapidly increases with only a modest decline in velocities. This is the upper part of the sickle-shaped figure, whereas the protruded part of the figure could be considered as the best condition for highway travel, with most flow and almost fastest velocity. Next, a marked drop in velocity and flow is observed and their values are highly variable and attain very low levels, named congested regime which is more between 7 am and 9 am. Finally, the situation recovers with a return to higher flows and an improvement in velocities [5].

However, the observation from loop M9 is an exception one needs to carefully investigate, probably because it is an easily congestion part of the whole stretch, e.g. due to important intersection or under construction during those two months of the year. Therefore, when one is building the prediction model, one probably should not consider data from this loop and implement a new one.

In addition, all the figures have some data points which are some distance away from the remarkable ‘sickle’. Obviously, those points are almost from time interval after 9 am with high velocities and very low flow, which indicates a low-occupancy period and probably waste of resources. This reminds authorities to balance highway occupancy during rush hours, e.g. use ramp metering to control flow of each time unit.

2.3.2 Current Status Travel Time VS Its Lagged Values

The purpose of this comparison is to present the proposed linear relationship between the *current status travel time* (CST) and the *exact travel time* in this thesis, where the latter statistic can be estimated by certain lagged values of the former one. Since the computation of such statistic only requires information available at time t, which means that it would be an accurate predictor only when there is a short lag. This is taken into account in the following plots and the lag does not exceed 30 minutes limit.

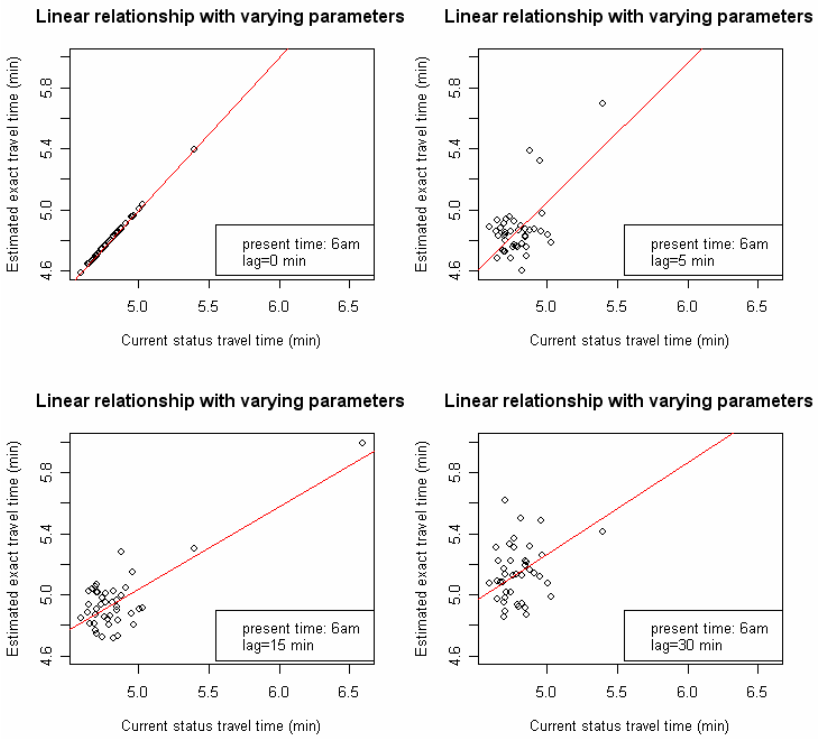


Figure 2.5: The form of linear relationship between the current status travel time and the estimated exact travel time of journeys from PN2 to M7, 6am of 42 weekdays on September & October 2005

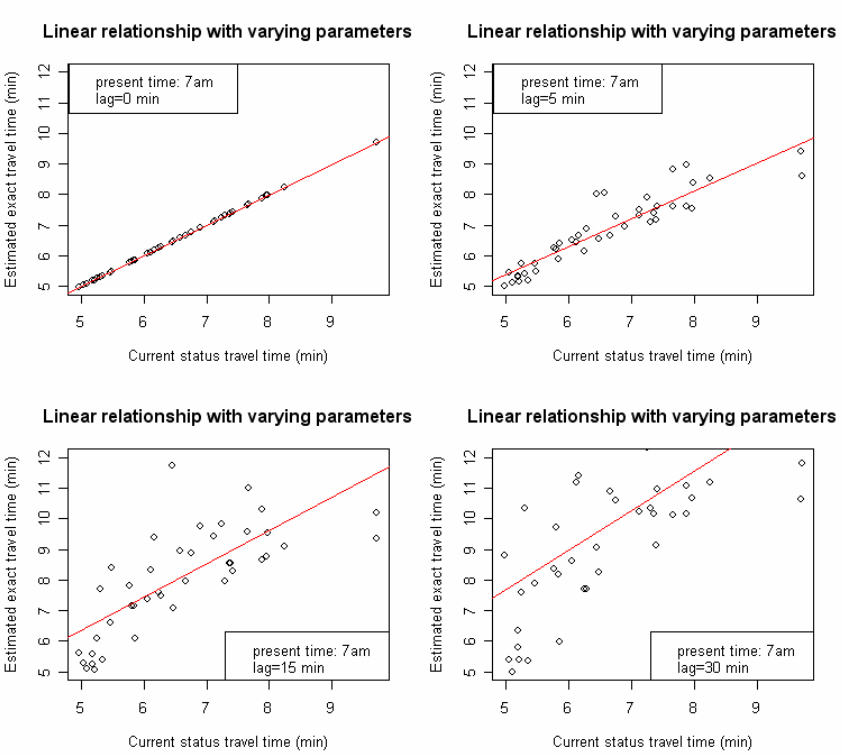


Figure 2.6: The form of linear relationship between the current status travel time and the estimated exact travel time of journeys from PN2 to M7, 7am of 42 weekdays on September & October 2005

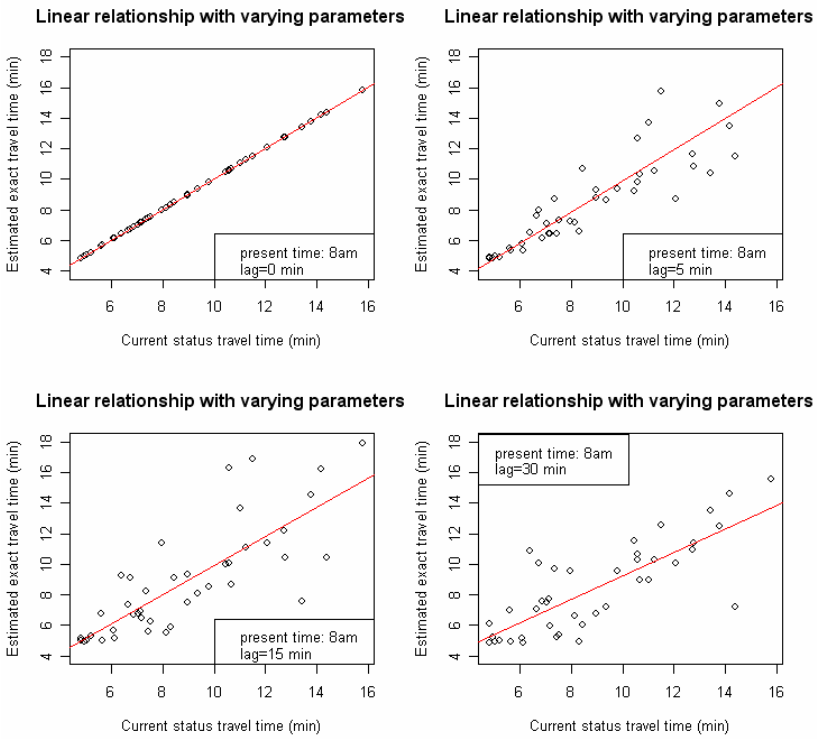


Figure 2.7: The form of linear relationship between the current status travel time and the estimated exact travel time of journeys from PN2 to M7, 8am of 42 weekdays on September & October 2005

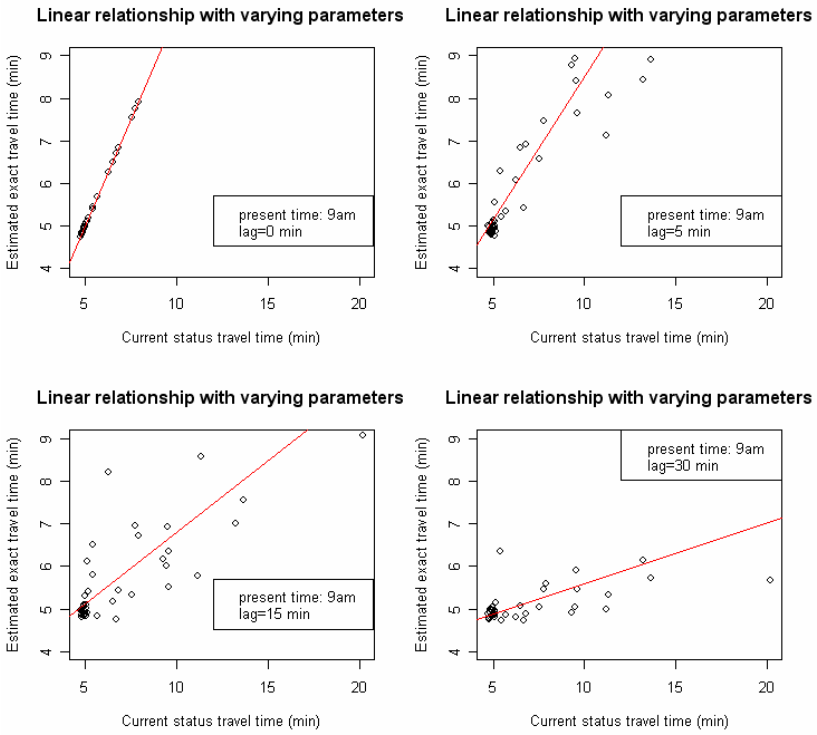


Figure 2.8: The form of linear relationship between the current status travel time and the estimated exact travel time of journeys from PN2 to M7, 9am of 42 weekdays on September & October 2005

Clearly seen, the estimated exact travel time and the current status travel time (the predictor) has a linear relationship, but there is a twist: they have parameters which themselves vary in parameterized ways-they are functions of the time of day and of the lag between the time at which you want start your journey. This leads to a *weighted least squares* (WLS) regression problem, the effect of which is to produce smooth estimates of the regression parameters, and hence a journey time predictor [5].

2.3.3 Time of Day VS Current Status Travel Time

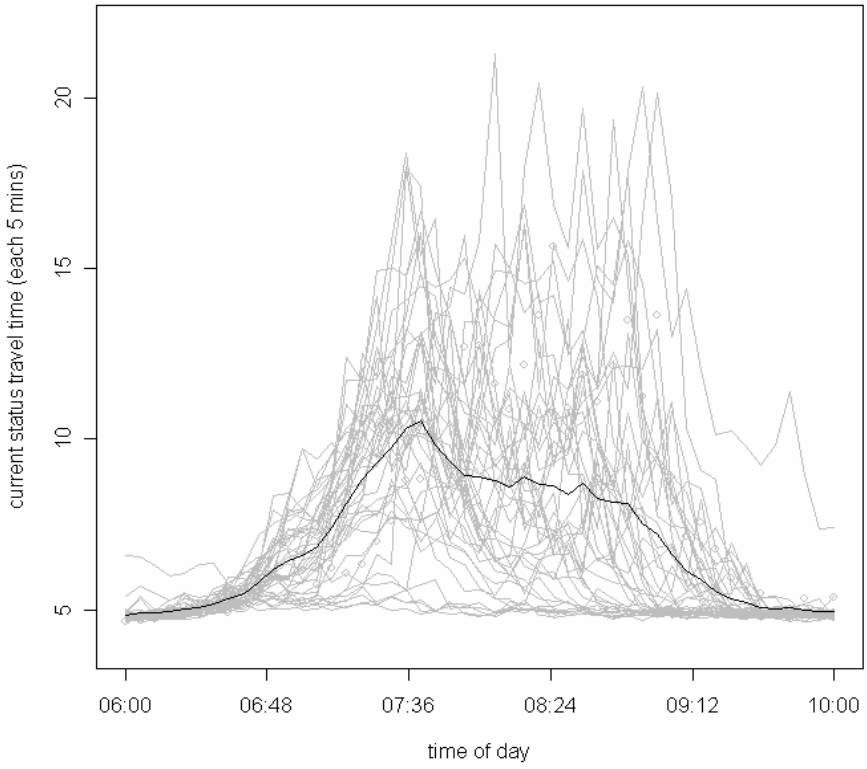


Figure 2.9: Current Status Travel Time per 5 minutes for 42 weekdays on half of the stretch

The purpose of this plot is to show the rush hours for each weekday during the two months. As one can see, the shadow grey lines stand for the raw values of CST aggregated over 5 minutes according to corresponding time interval; whereas the black line stands for the means of CST for 42 days according to corresponding time interval. Typically, the distinctive congestion and the huge variability of travel times are presented between 7am and 9 am, which could also be referred to the results of section 2.3.1. These could remind that 9 am is the common working starting time in Denmark and people are trying to reach their office before this time. Of course, their take-off time from home is starting from 7am.

2.4 Autocorrelation Function

2.4.1 Theory

Autocorrelation function (ACF) is the expected value of the product of a random variable or signal realization with a time-shifted version of itself. With a simple calculation and analysis of the autocorrelation function, we can discover a few important characteristics about our random process. These include [6]:

- How quickly the random signals or processes change with respect to the time function;
- Whether the process has a periodic component and what the expected frequency might be;

According to section 2.3.2, since a linear relationship is found between CST and its lagged values (estimates of exact travel time), we could conclude that it is an *autoregressive* (AR) process. Also we want to see which lags are been putting on significant weights by the process itself, in order to determine the lagged explanatory variables in the linear model later on. Therefore, ACF will give the hint.

2.4.2 Results

The ACF for the time series CST is plotted below:

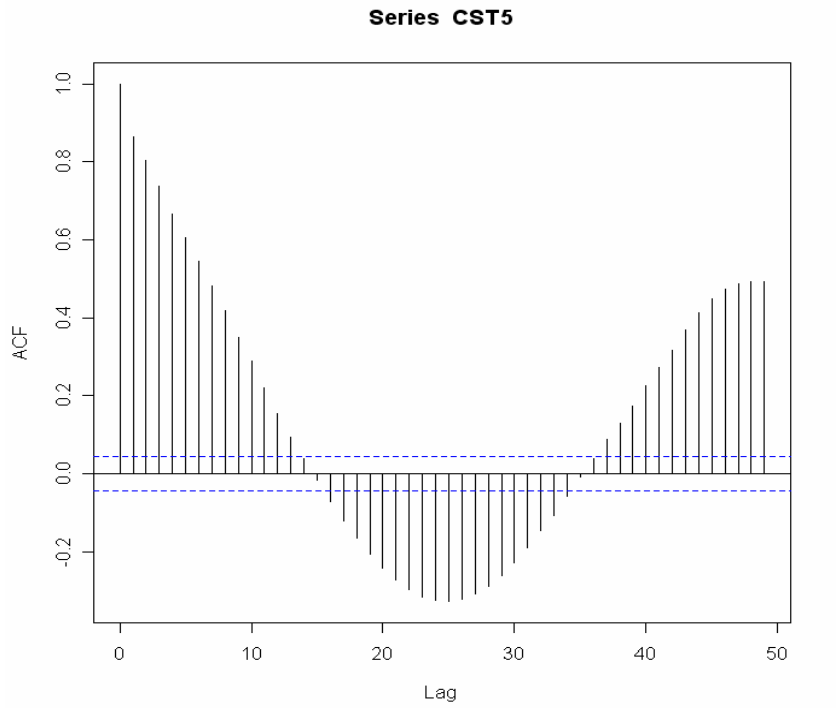


Figure 2.10: The plot of *autocorrelation function* of series CST5 (*current status travel time* per 5 minutes) up to lag 49

We first investigate the lags up to 49, which is exactly the CST5 from the day before at the same time of the day. From what Figure 2.10 presents above, it is probably suitable to use an AR(2) or AR(3) model to fit the process. We keep looking into the lags up to 98 for day-to-day variation.

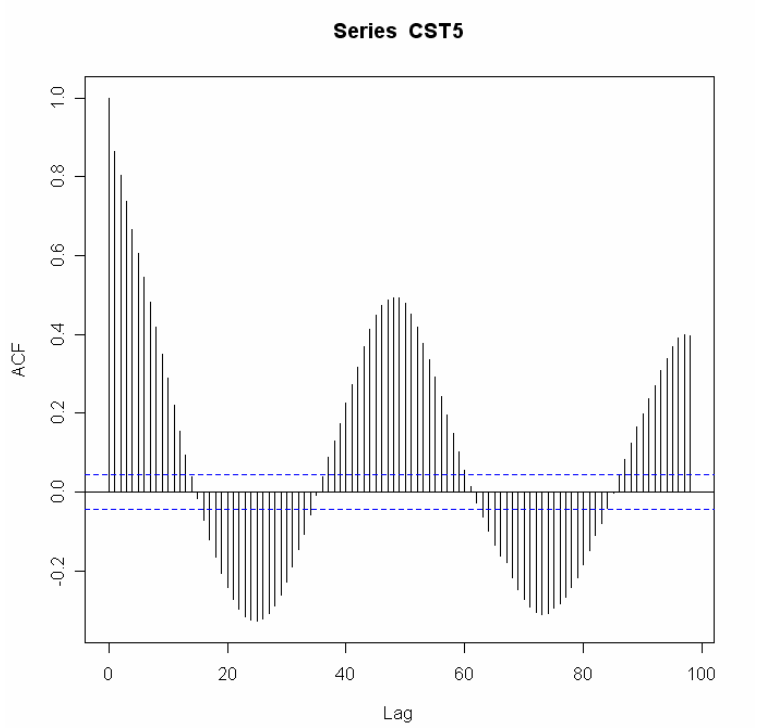


Figure 2.11: The plot of *autocorrelation function* of series CST5 (*current status travel time* per 5 minutes) up to lag 98

From lag 50 (more time steps ahead), there is no obvious strong relationship between them and CST5 compared to previous ones. Thus, those lagged values should not be considered into model formulation.

2.5 Impulse Response Function

2.5.1 Theory

The autocorrelation typically present in observed time series makes direct use of the *cross correlation function* (CCF) to study lagged relationships problematical. Essentially, the questions asked of the CCF are 1) how strongly is one series related to another, 2) is the relationship simultaneous or distributed over several time steps, and 3) if distributed, how many lags are involved and what is their relative importance [7]?

These questions can be addressed by a systems approach in which the series are regarded as input to and output from a *linear dynamic system*. *Dynamic* refers to the possible dependence of the output at time *t* on the input signal at many previous times. For such a system, the hypothetical response to a unit pulse of input at time $t = 0$ is given by the *impulse response function* (IRF) [7].

The system ideally has the following properties:
• *time invariant:* response to an input signal does not depend on absolute time
• *linear:* output response to a linear combination of inputs is the same as the linear combination of the output responses to the individual inputs
• *causal:* output at a certain time depends on the input up to that time only [7]

The process is described by the following equation

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t) \tag{2.4}$$

Where

$y(t)$ = output in time t

$u(t-k)$ = input in time $t-k$

$g(k)$ = impulse response function at lag k

The summation as written does not allow a response at time *t* to an input at time *t.* This can be remedied with no loss of generality by shifting the input one time step relative to the output (realigning the series) so that the summation effectively starts with $k = 0$. Realignment can also insure that for a nominal input and output the output response does not precede the input stimulously [7].

The model gives the output as a linear combination of past (and possibly current) input. The numbers $\{ g(k) \}$ are called the *impulse response function* (IRF). Of course, usually the IRF is unknown, and must be estimated from the data -- the input signal

$$u(t), t = 1,..., N \tag{2.5}$$

and the output signal

$$y(t), t = 1,..., N \tag{2.6}$$

2.5.2 Estimation of the IRF by Correlation Analysis

The method used here for estimating the IRF is based on reducing one of the series to white noise before computing its correlation with the other series. The need for this “prewhitening” results from the complicating effects of autocorrelation on the estimated CCF and its standard deviations [7].

The method amounts to passing the input and output series through a filter before computing the CCF. The filter is chosen such that it reduces the input series to white noise (removes the autocorrelation). The filtered input series is therefore called the “prewhitened” input. The filtered output will generally not be white noise because the filter has been designed specifically to prewhiten the input, not the output. The CCF between the prewhitened input and filtered output is an estimate of the IRF of the system [7].

The IRF or the CCF between the prewhitened input and filtered output describes the lagged correlation structure disentangled from the influence of autocorrelation. A constant 99% *confidence interval* (CI) is more appropriate for this IRF than for the CCF of the original series, however, because one of the series is now approximately white [7].

2.5.3 Implementation for the Problem

As mentioned before, there are 34 induction loops located all the way along the stretch which are analyzed in this thesis. Technically, the loop data provide information concerning vehicle amount and velocity due to certain time interval (every 30 seconds) and point (loop where it locates on the stretch). Such information could be compiled to road condition (free-flow or congestion) at that time or that point of the road.

Considering information gathered from different loops as different time series (also each series stand for information from one certain loop), IRF is implemented here to describe time (lag) and space-related (different parts of the stretch) information among those series, especially looking into what the road condition varies and how it affects subsequent points of the stretch at certain time step.

Below here are plots of IRF for harmonic mean of velocities between starting point PN2 and other 16 points (exactly the data from training set), which is aggregated over 1, 2 and 5 minutes (per lag), respectively.

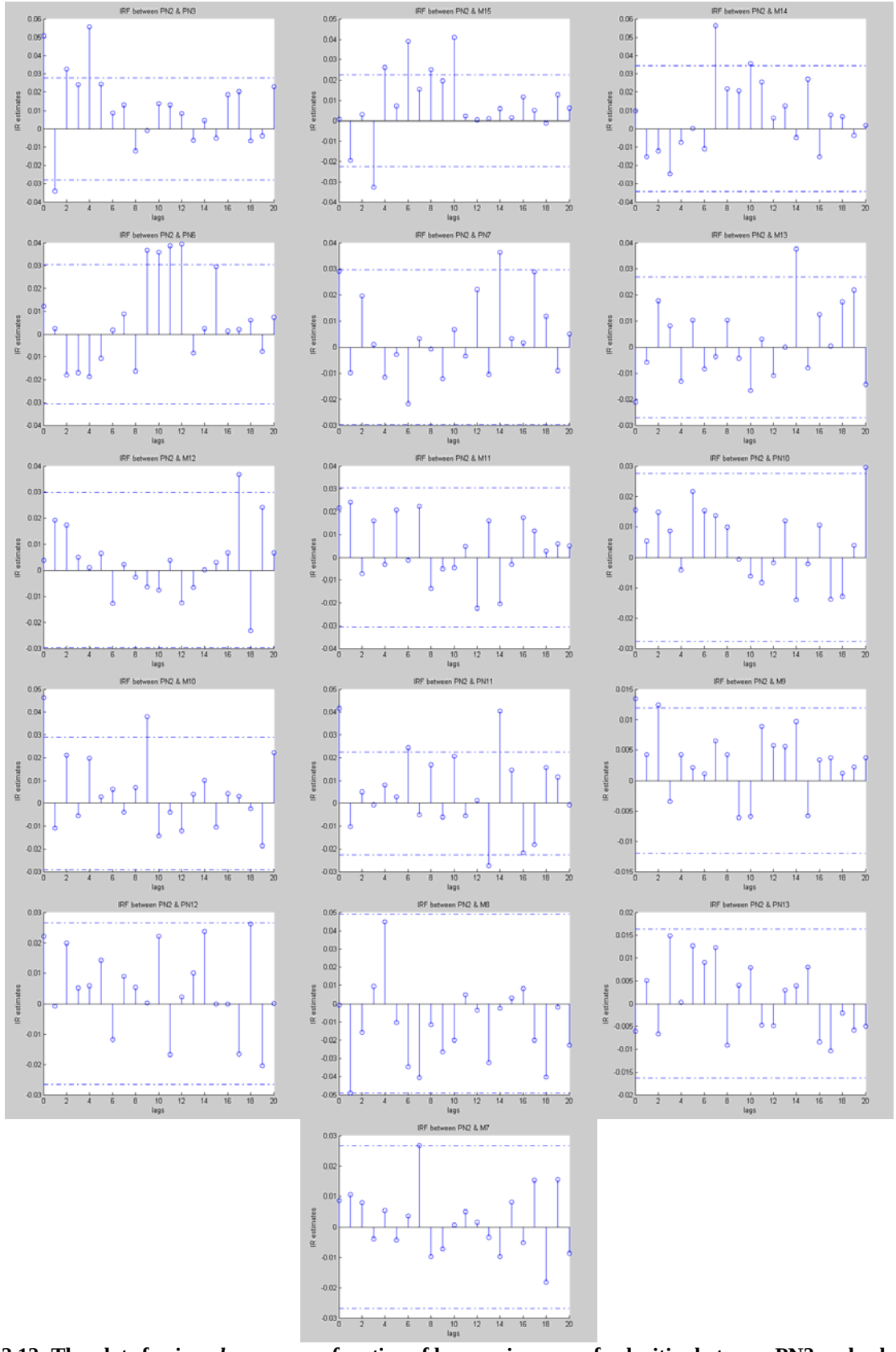


Figure 2.12: The plots for *impulse response function* of harmonic mean of velocities between PN2 and subsequent 16 loops on Sept. & Oct. 2005, aggregated 1 minute

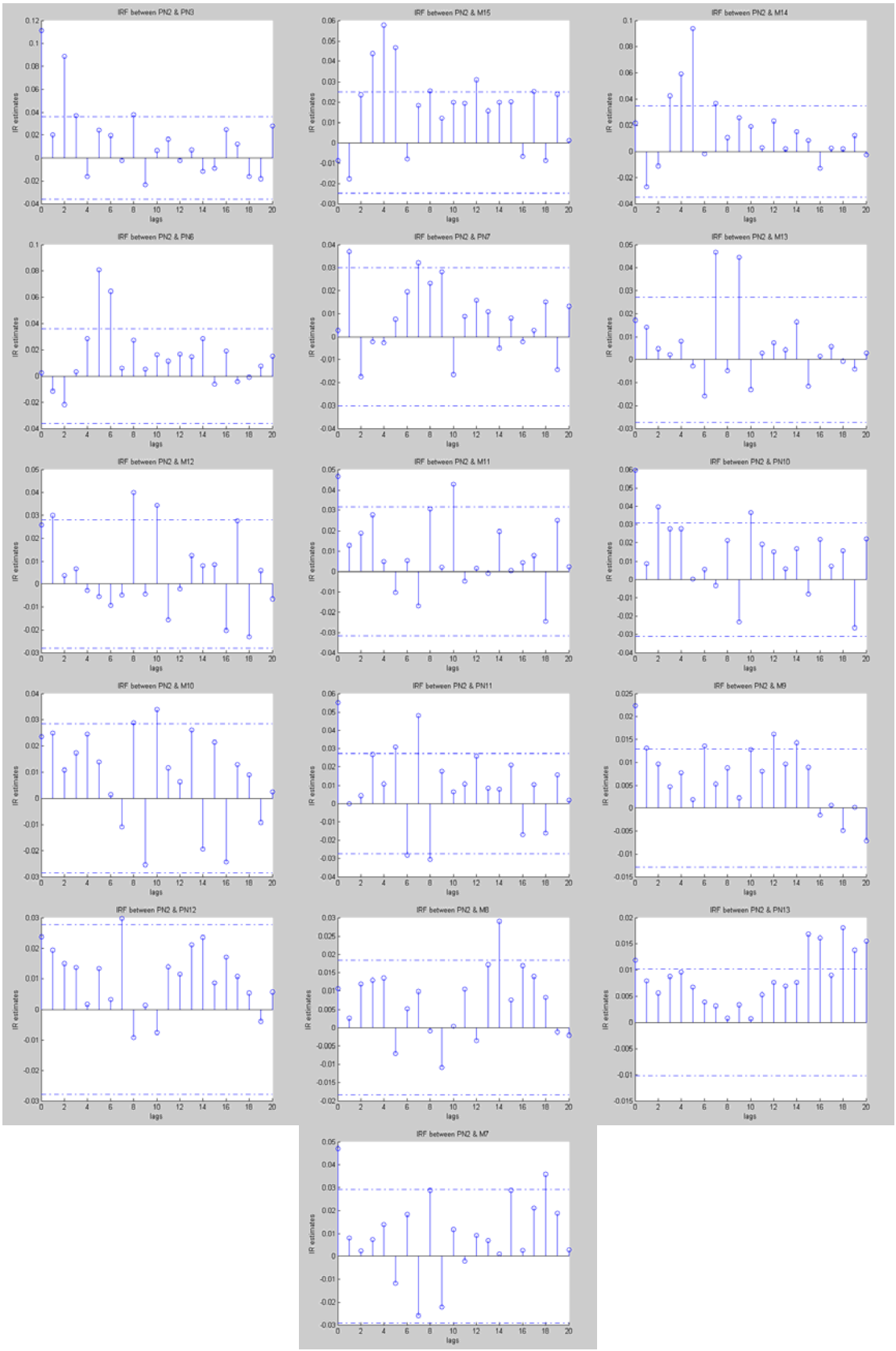


Figure 2.13: The plots for *impulse response function* of harmonic mean of velocities between PN2 and subsequent 16 loops on Sept. & Oct. 2005, aggregated 2 minutes

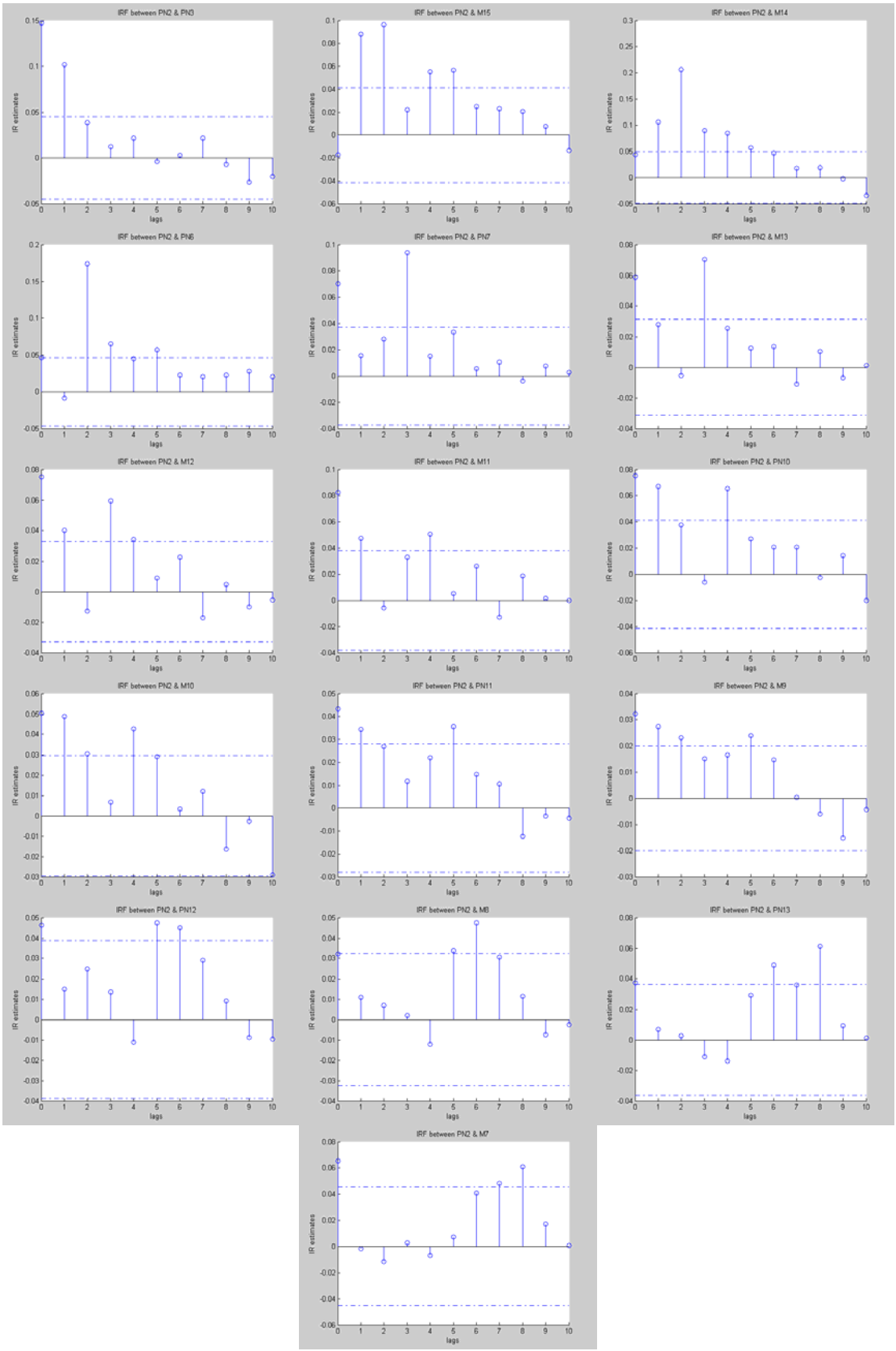


Figure 2.14: The plots for *impulse response function* of harmonic mean of velocities between PN2 and subsequent 16 loops on Sept. & Oct. 2005, aggregated 5 minutes

Chapter 2

Descriptive Statistics

2.1 Original Data Description

This thesis project concentrates on data collection of 34 double loop detectors, with corresponding code numbers (e.g. M7), distributed into an approx. 16 km long's stretch. The road map is
As follows:

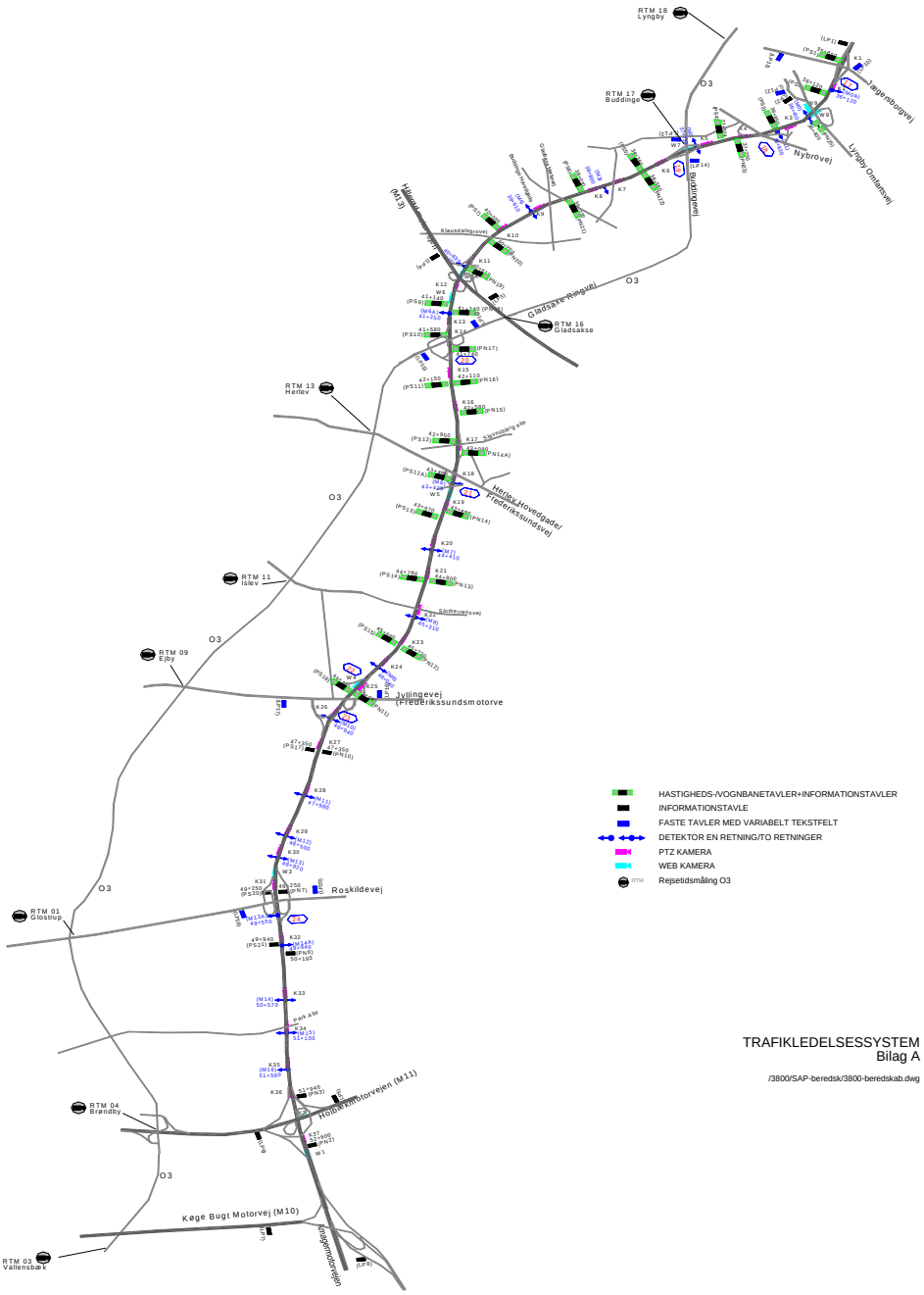


Figure 2.1: The road map of traffic system in the researched motorway

If one assumes that the road condition starting from one point (where certain induction loop locates) could be moving forward along with the road through point by point, the characteristic of which is pretty much close to that of a wave. The main difference probably is just one single direction vs. all around. Actually, what happens is surely the case.

During a certain interval of time, there will be certain amount of vehicles entering some point of the road. After a while (some specific lags), same amount of vehicles will arrive at other points of the road. Imaginably, those vehicles are carrying on 'road condition' more or less along with their movement. The above plots are IRF between the first loop and subsequent loops for the training set, which can be referred to the 'movement' of road condition from first point to others according to certain time steps (lags). For example, the vertical lines in the plots which surpass the confidence interval horizontal lines (both directions) mean the condition happened in the starting point (in this case it is where loop PN2 locates) moment ago would happen again after some time steps (lags) where such vertical line is in accordance with (the *X* axis). Obviously, the more distant away from the starting point, the more lags necessary for the rebound of the road conditions. Also of course, due to local phenomena, some points will not happen anything at all like the previous ones even though the subsequent ones do happen.

Chapter 3

Theoretical Methods

3.1 Linear Regression with Time-varying Coefficients

3.1.1 Introduction

Since a time series is an outcome of a stochastic process and thus an observation of a dynamic phenomenon, methods which normally are related to the analysis and modeling of static phenomena, are often applied. A class of such methods is closely related to the ordinary regression analysis [8].

Regression analysis is any statistical method where the mean of one or more random variables is predicted conditioned on other (measured) random variables. In particular, there is linear regression, logistic regression, Poisson regression, supervised learning, and unit-weighted regression. Regression analysis is more than curve fitting (choosing a curve that best fits given data points); it involves fitting a model with both deterministic and stochastic components, where the deterministic one is called the predictor and the stochastic one is called the error term [9].

Regression analysis is probably the most widely used form of analysis of dependency, which is used to explore the static relationship between a set of independent variables (*X*'s) and a single dependent variable (*Y*). A regression model is a linear combination of independent variables that corresponds as closely as possible to the dependent variable, whose purposes are description, inference and prediction [10]. Within time series analysis the observations occur successively in time and most frequently with equidistant time delay. Therefore an index *t* is introduced to denote the variable at time origin t. We can take a look at the most general form the regression model is written:

$$Y_t = f(X_t, t; \theta) + \varepsilon_t \tag{3.1}$$

where $f(X_t, t; \theta)$ is a known mathematical function of the $p+1$ independent variables $X_t = (X_{1t}, ..., X_{pt})^T$ and t; but with unknown parameters $\theta_t = (\theta_1, ..., \theta_m)^T$. The independent variable t is introduced to indicate that the model class described by (3.1) contains models where f is a function of *t*. ε_t is a random variable with $E[\varepsilon_t] = 0$ and $V[\varepsilon_t] = \sigma_t^2$. Furthermore it is assumed that $Cov[\varepsilon_{ti}, \varepsilon_{tk}] = \sigma^2 \Sigma_{ij}$ i.e. $\sigma_t^2 = \sigma^2 \Sigma_{ii}$. Finally ε_t and X_t are assumed to be independent [8].

For the general solving procedure, regression is usually posed as an optimization problem as we are attempting to find a solution where the error is at a minimum. The most common error measure that is used is the least squares estimates: this corresponds to a Gaussian likelihood of generating observed data given the (hidden) random variable. In a certain sense, such method is an optimal estimator (according to Gauss-Markov theorem).

3.1.2 General Algorithm

According to section 2.3.2, there exists linear relationship between the *current status travel time* and the *exact travel time*. Here the lagged values of *current status travel time* are used to estimate the *exact travel time*. Thus, linear regression analysis is the primary consideration in this thesis. Note that such relationship varies with the choice of current time t and lag δ.

If the unknown parameter (or coefficient) θ is varying along with the time, which means that observations in the past should be given less weight than present observations in the least squares criterion. Therefore, a forgetting factor or a discount factor λ should be considered to measure the weight under the *Weighted Recursive Least Square* (WRLS) estimates as described below.

Consider the following model of a linear system with discrete time [11]:

$$Y_t + \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} = \omega_1 U_{t-1} + ... + \omega_m U_{t-m} + \varepsilon_t \tag{3.2}$$

where $\{Y_t\}$ is the output signal, $\{\varepsilon_t\}$ is white noise, uncorrelated with the external signal $\{U_t\}$. The model (3.2) may now be written as a linear regression form [11]:

$$Y_t = X_t^T \theta + \varepsilon_t \tag{3.3}$$

where $X_t^T = (-Y_{t-1},...,-Y_{t-p},U_{t-1},...,-U_{t-m})$ and time varying unknown parameters $\theta^T = (\phi_1,...,\phi_p,\omega_1,...,\omega_s)$.

For *weighted least squares*, the parameter estimates at time origin t are:

$$\widehat{\theta}_t = \arg\min_{\theta} S_t(\theta) \tag{3.4}$$

where

$$S_t(\theta) = \sum_{s=1}^{t} \beta(t,s)\left[Y_s - X_s^T \theta\right]^2 \tag{3.5}$$

Assume that the sequence of weights satisfies

$$\beta(t,s) = \lambda(t)\beta(t-1,s)\,; \quad 1 \le s \le t-1 \tag{3.6}$$

$$\beta(t,t) = 1 \tag{3.7}$$

which means that

$$\beta(t,s) = \prod_{j=s+1}^{t} \lambda(j) \tag{3.8}$$

The weight in the computation of the parameter estimate at time t of the squared residual at time s is computed as the product all the intermediate weighting factors.

Since this WLS problem is solved by the criterion of 3.4, we can obtain the following solution:

$$\hat{\theta}_t = R_t^{-1} h_t \tag{3.9}$$

where

$$R_t = \sum_{s=1}^{t} \beta(t,s) X_s X_s^T ;\ h_t = \sum_{s=1}^{t} \beta(t,s) X_s Y_s . \tag{3.10}$$

Then the recursive formulas can be calculated as follows by using the previously computed values [11]:

$$R_t = \lambda(t) R_{t-1} + X_t X_t^T \tag{3.11}$$

$$h_t = \lambda(t) h_{t-1} + X_t Y_t \tag{3.12}$$

The updating equation for the estimate of the parameter vector θ may be found by using the updates of R_t and h_t

$$\begin{aligned} \hat{\theta}_t &= R_t^{-1} h_t = R_t^{-1}\left[\lambda(t) h_{t-1} + X_t Y_t\right] \\ &= R_t^{-1}\left[\lambda(t) R_{t-1} \hat{\theta}_{t-1} + X_t Y_t\right] \\ &= \hat{\theta}_{t-1} + R_t^{-1} X_t \left[Y_t - X_t^T \hat{\theta}_{t-1}\right] \end{aligned} \tag{3.13}$$

Thus, combining with equation (3.11) and (3.13), we have the WRLS method with a forgetting structure [11].

As called forgetting factor, it is here assumed that $\lambda(t) = const = \lambda$. It determines the exponential discount of past observations, the choosing value of which satisfies $0.70 < \lambda^p < 0.95$, where p is the number of parameters in the model. However, it is natural to choose a forgetting factor according to some criterion, e.g. the value which gives the minimum variance of the prediction error [8]. Once the optimum forgetting factor is determined, it should be used again in equation 3.13 recursively to find out the time-varying parameters of the regression model 3.3.

3.2 Principle Component Analysis

3.2.1 Introduction

Principle component analysis (PCA) is a method for re-expressing multivariate data by rotating the original coordinates system to a new coordinates system so that the first few dimensions account for as much of the available information as possible [12]. Normally the information of a data set is expressed by the total variance of the variables in the data set, and PCA is concerned with explaining the variance – covariance structure of a set of variables through a few linear

combinations of these variables [13].

Although *p* components (equal to the number of variables in the original data set) are required to reproduce the total system variability, often much of this variability can be accounted for by a small number *k* of the principle components. If so, there is almost as much information in the *k* components as there is in the original *p* variables. The *k* principal components can then replace the initial *p* variables, and the original data set, consisting of *n* measurements on *p* variables, is reduced to a data set consisting of *n* measurements on *k* principal components [13].

The principle components solution often reveals relationships that were not previously suspected and thereby allows interpretations that would not ordinarily result [13]. And it has the property that each component is uncorrelated with all others, which has the advantage of eliminating multicollinearity when using the results in an analysis of dependence, such as multiple regression [12].

3.2.2 General Algorithm

Ordinarily, a data set obtained comprises *p* variables and *k* observations, and different variables have different scales, so that variables with large scales capture most variability of the data set and largely impact the results of analysis. In order to avoid this problem, all the variables are standardized first, that is, the values of each variable are centered and divided by the standard deviation of each variable.

$$X_{scale} = \frac{X - X_{mean}}{X_{sdev}} \tag{3.14}$$

In this way each variable is normalized to zero mean and unit variance, and in the space the data set in the first two dimensions may look like below.

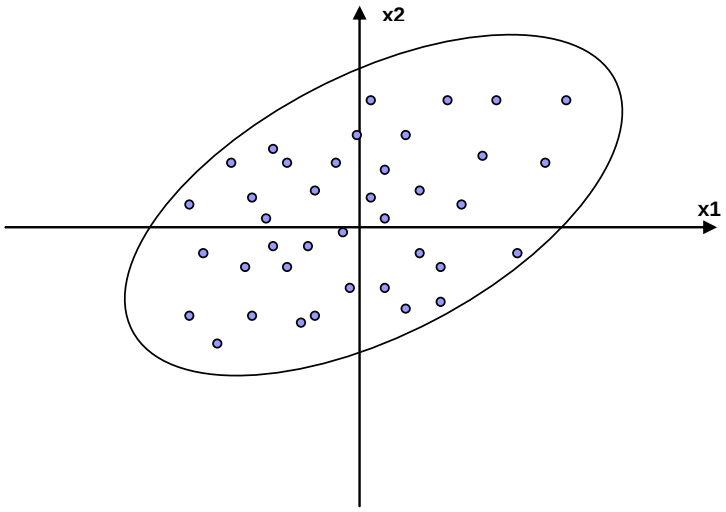


Figure 3.1: A data set in the space (first two dimensions)

The shape of data set in the space commonly likes a hyper-ellipsoid. Geometrically, the first principal component (PC1) represents the direction of the data set with the largest variation, the second principal component (PC2) with the second largest variation, etc., and actually they are the directions of longest axis, second longest axis, etc. of the hyper-ellipsoid respectively, which also means that principal components are orthogonal to each other.

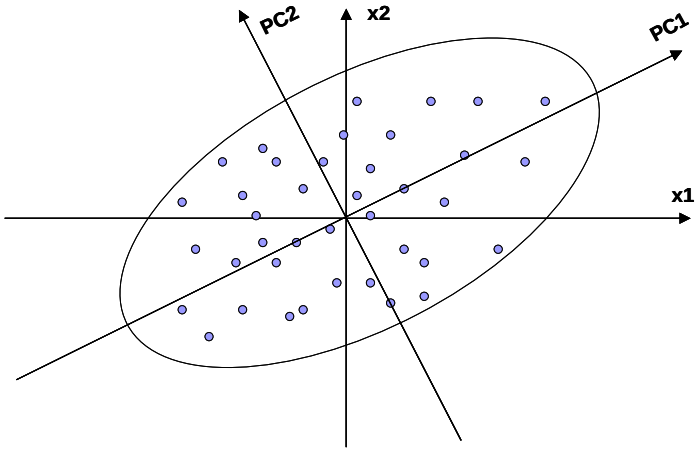


Figure 3.2: The possible direction of PC1 and PC2

Algebraically, principal components are particular linear combinations of the original variables $X = [x_1, x_2, \ldots, x_p]$. For PC1 the purpose is to find the linear combination $u_1 = (u_{11}, u_{12}, \ldots, u_{1p})'$ to maximize the variance of the elements of $z_1 = X u_1$, which may be written as follows [12]:

$$\mathrm{var}(z_1) = \frac{1}{(n-1)} u_1' X' X u_1 \tag{3.15}$$

Because X is standardized, the term $1/(n\text{-}1)X'X$ is just the sample correlation matrix R. Substituting yields [12]

$$\mathrm{var}(z_1) = u_1' R u_1 \tag{3.16}$$

For the sake of making var(z_1) meaningful, a constraint is imposed on the length of u_1 vector, which is stated as $u_1' u_1 = 1$. The problem thus modified can be described as follows [12]:

$$\begin{gathered} \text{Choose } u_1 \text{ to maximize } u_1' R u_1 \\ \text{Subject to the constraint } u_1' u_1 = 1 \end{gathered} \tag{3.17}$$

From the knowledge of linear algebra, this problem can be referred to as an eigenvalue - eigenvector problem.

$$R u_1 = \lambda_1 u_1 \quad \text{or} \quad (R - \lambda_1 I) u_1 = 0 \tag{3.18}$$

The vector u_1 is called an eigenvector and λ_1 is called an eigenvalue. Provided the matrix R is of full rank, then the solution will consist of p positive eigenvalues and associated eigenvectors [12].

It is easy to verify that postmultiplying the original data $X = [x_1, x_2, \dots, x_p]$ by the eigenvectors $U = [u_1, u_2, \dots, u_p]$ yields the matrix of principal component scores $Z = [z_1, z_2, \dots, z_p]$ [12].

$$\begin{aligned} z_1 &= Xu_1 = x_1u_{11} + x_2u_{12} + \dots + x_pu_{1p} \\ z_2 &= Xu_2 = x_1u_{21} + x_2u_{22} + \dots + x_pu_{2p} \\ &\vdots \\ z_p &= Xu_p = x_1u_{p1} + x_2u_{p2} + \dots + x_pu_{pp} \end{aligned} \tag{3.19}$$

Eigenvalues $\lambda_1 > \lambda_2 > \dots > \lambda_p$ are exactly the variances of the associated principal components by

$$\mathrm{var}(z_i) = u_i' R u_i = u_i' \lambda_i u_i = \lambda_i u_i' u_i = \lambda_i \qquad i = 1, 2, \dots, p \tag{3.20}$$

The associated eigenvectors $u_1, u_2, \dots, u_p$ denote the direction of the corresponding principal component in X space. The scores $z_1, z_2, \dots, z_p$ is the projections of the original data on the principal components, and an alternative explanation is the new coordinates of the data in a coordinates system with axes of principal components.

The principal components are mutually uncorrelated so that the covariance between any two principal components is equal to zero, which leads to eliminating multicollinearity.

$$Cov(z_i, z_j) = 0 \qquad i, j = 1, 2, \dots, p \tag{3.21}$$

The total variance of the data set is simply the sum of the variances of the individual components, which equal to the sum of variances of the original variables.

$$\lambda_1 + \lambda_2 + \dots + \lambda_p = \sum_{i=1}^{p} \mathrm{var}(z_i) = \sigma_1^{\,2} + \sigma_2^{\,2} + \dots + \sigma_p^{\,2} = \sum_{i=1}^{p} \mathrm{var}(x_i) \tag{3.22}$$

$\sigma_1^{\,2}, \sigma_2^{\,2}, \dots, \sigma_p^{\,2}$ are the variances of original variables respectively. For the standardized date set, $\sigma_1^{\,2}, \sigma_2^{\,2}, \dots, \sigma_p^{\,2}$ are all equal to 1, thereby this sum of variances is also the same as the number of variables.

$$\lambda_1 + \lambda_2 + \dots + \lambda_p = \sigma_1^{\,2} + \sigma_2^{\,2} + \dots + \sigma_p^{\,2} = p \tag{3.23}$$

Consequently, the proportion of total variance explained by the *i*th principal component is [13]

$$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \dots + \lambda_p} \qquad i = 1, 2, \dots, p \tag{3.24}$$

If most (≥70%) of the total variance of the data set can be attributed to the first one, two, or several components, then these components can replace the original p variables without much loss of information for subsequent analysis [13]. This is the Simplification & Dimension Reduction of the

data set.

Another useful application of PCA solution is the principal component loadings. Simply said, the loadings are the projections of original coordinates' axes on the principal components. They can be calculated by the correlation matrix of the principal component scores (Z) with the original data set (X), and help to interpret the relationships between the principal components and the original variables [12]. The correlation matrix is given by the expression below.

$$corr(X, Z) = \frac{1}{(n-1)} X' Z_s \tag{3.25}$$

where X is the matrix of the original standardized data set, and Z_s is the matrix of standardized principal components given by [12]

$$Z_s = ZD^{-1/2} \tag{3.26}$$

where D is the diagonal covariance matrix of the principal components and denoted as $D = diag(\lambda_1, \lambda_2, ..., \lambda_p)$ [12]. From equation 3.19 we know $Z = XU$, then after these transformations equation 3.25 yields

$$corr(X, Z) = \frac{1}{(n-1)} X' XUD^{-1/2} \tag{3.27}$$

Because $1/(n\text{-}1)X'X$ is just the sample correlation matrix R and we know from equation 3.18 that R can be re-expressed as UDU' [12], then substituting into the above to obtain

$$corr(X, Z) = (UDU')UD^{-1/2} = UD^{1/2} \tag{3.28}$$

because $U'U=I$ [12]. Thus, we have the result of the principal component loadings calculated by the correlation matrix between the principal components Z and the original variables X, and the loadings denote as

$$F = UD^{1/2} \tag{3.29}$$

which are determined by the eigenvectors and the square root of eigenvalues. It maybe more clear to be denoted in an equivalent form.

$$F = \begin{bmatrix} u_{11}\sqrt{\lambda_1} & u_{21}\sqrt{\lambda_2} & \cdots & u_{p1}\sqrt{\lambda_p} \\ u_{12}\sqrt{\lambda_1} & u_{22}\sqrt{\lambda_2} & \cdots & u_{p2}\sqrt{\lambda_p} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1p}\sqrt{\lambda_1} & u_{2p}\sqrt{\lambda_2} & \cdots & u_{pp}\sqrt{\lambda_p} \end{bmatrix} \tag{3.30}$$

where $u_{ij}\sqrt{\lambda_i}$ is the loading of the ith original variable on the jth principal component. By the matrix of loadings, the relationships between the original standardized data set X and the principal

component scores Z can be expressed as follows.

$$X = ZF' \tag{3.31}$$

or in an equivalent form

$$\begin{aligned} x_1 &= z_1 u_{11}\sqrt{\lambda_1} + z_2 u_{21}\sqrt{\lambda_2} + \ldots + z_p u_{p1}\sqrt{\lambda_p} \\ x_2 &= z_1 u_{12}\sqrt{\lambda_1} + z_2 u_{22}\sqrt{\lambda_2} + \ldots + z_p u_{p2}\sqrt{\lambda_p} \\ &\vdots \\ x_p &= z_1 u_{1p}\sqrt{\lambda_1} + z_2 u_{2p}\sqrt{\lambda_2} + \ldots + z_p u_{pp}\sqrt{\lambda_p} \end{aligned} \tag{3.32}$$

If the first m principal components are retained to describe the original variables, the equation above is transformed into

$$X = Z_m F_m' + E \tag{3.33}$$

where F_m is a matrix of the first m columns of loadings matrix F, Z_m is a matrix of the first m columns of scores matrix Z, and E is the errors. Or it can be equivalently expressed as

$$\begin{aligned} x_1 &= z_1 u_{11}\sqrt{\lambda_1} + z_2 u_{21}\sqrt{\lambda_2} + \ldots + z_m u_{m1}\sqrt{\lambda_m} + \varepsilon_1 \\ x_2 &= z_1 u_{12}\sqrt{\lambda_1} + z_2 u_{22}\sqrt{\lambda_2} + \ldots + z_m u_{m2}\sqrt{\lambda_m} + \varepsilon_2 \\ &\vdots \\ x_p &= z_1 u_{1p}\sqrt{\lambda_1} + z_2 u_{2p}\sqrt{\lambda_2} + \ldots + z_m u_{mp}\sqrt{\lambda_m} + \varepsilon_p \end{aligned} \tag{3.34}$$

where $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p$ stand for the error of variable $x_1, x_2, \ldots, x_p$ respectively.

From the principal component loadings, we can determine the amount of variance of each original variable accounted for by any number of principal components [12]. The general expression for the variance accounted for in variable x_i by the first m retained principal components is [12]

$$\sum_{j=1}^{m} (u_{ji}\sqrt{\lambda_j})^2 \tag{3.35}$$

When all the principal components are retained, the result of above equation is 1. This can be a potential method for variable selection which will be discussed in the following chapter.

3.3 Partial Least Squares Regression

3.3.1 Introduction

Partial Least Squares (PLS) regression is a multivariate data analysis technique which can be used to relate several response (*Y*) variables to several explanatory (*X*) variables. The method aims to identify the underlying or latent factors or linear combination of the *X* variables, which account for most of the variation in the response (the *Y* dependent variables) [14]. Note that the emphasis of PLS is on predicting the responses and not necessarily on trying to understand the underlying relationship between the variables.

In PLS, one set of latent variables is extracted for the set of manifest independents and another set of latent variables is extracted simultaneously for the set of manifest response (dependent) variables. The extraction process is based on decomposition of a cross-product matrix involving both the independent and response variables. The *X*-scores of the independent latent(s) are used to predict the *Y*-scores or the response latent(s), and the predicted *Y* scores are used to predict the manifest response variables. The *X*- and *Y*- scores are selected by PLS so that the relationship of successive pairs of *X* and *Y* scores is as strong as possible [15].

PLS regression is probably the least restrictive of the various multivariate extensions of the multiple linear regression models. This flexibility allows it to be used in situations where the use of traditional multivariate methods is severely limited, such as when there are fewer observations than predictor variables. Furthermore, such method can be used as an exploratory analysis tool to select suitable predictor variables and to identify outliers before classical linear regression [16].

3.3.2 PLS & Other Multiple Regression Techniques

As in multiple linear regressions, the main purpose of *partial least squares* regression is to build a linear model:

$$Y=XB+E \tag{3.36}$$

where *Y* is an *n* cases by *m* variables response matrix, *X* is an *n* cases by *p* variables predictor (design) matrix, *B* is a *p* by *m* regression coefficient matrix, and *E* is a noise term for the model which has the same dimensions as *Y*. Usually, the variables in *X* and *Y* are centred by subtracting their means and scaled by dividing by their standard deviations.

Both *principal components regression* (PCR) and *partial least squares* regression produce factor scores as linear combinations of the original predictor variables, so that there is no correlation between the factor score variables used in the predictive regression model. For example, suppose we have a data set with response variables *Y* (in matrix form) and a large number of predictor variables *X* (in matrix form), some of which are highly correlated. A regression using factor extraction for this type of data computes the factor score matrix $T=XW$ for an appropriate weight matrix *W*, and then considers the linear regression model:

$$Y=TQ+E \tag{3.37}$$

where Q is a matrix of regression coefficients (loadings) for T, and E is an error (noise) term. Once the loadings Q are computed, the above regression model is equivalent to model 3.36, where $B=WQ$, which can be used as a predictive regression model.

PCR and PLSR differ in the methods used in extracting factor scores. In short, the former produces the weight matrix W reflecting the covariance structure between the predictor variables, while the latter produces the weight matrix W reflecting the covariance structure between the predictor and response variables. The structure of PLS can be shown below:

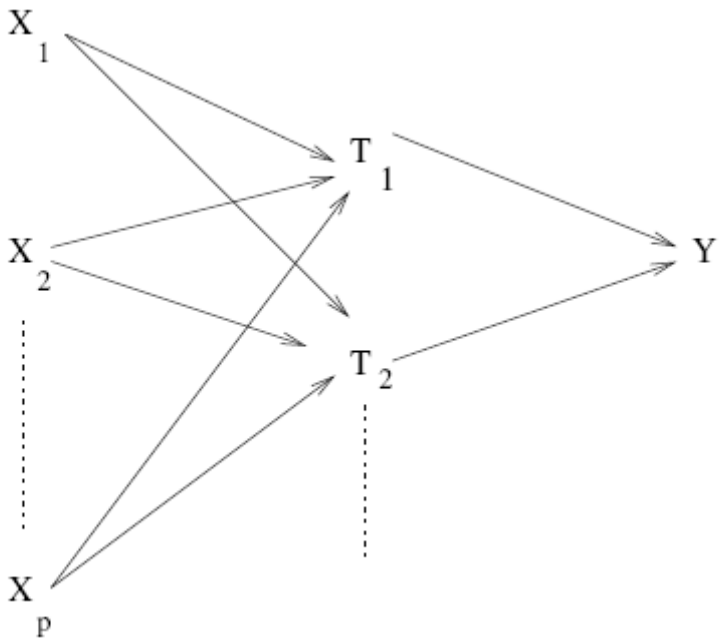


Figure 3.3: Schematic representation of PLS

T_k is a linear combination of the predictor X's, which plays a role as a 'bridge' between predictors and response Y's.

For establishing the model, PLS regression produces a p by c weight matrix W for X such that $T=XW$, i.e., the columns of W are weight vectors for the X columns producing the corresponding n by c factor score matrix T. These weights are computed so that each of them maximizes the covariance between responses and the corresponding factor scores. Ordinary least squares procedures for the regression of Y on T are then performed to produce Q, the loadings for Y (or weights for Y) such that $Y=TQ+E$. Once Q is computed, we have $Y=XB+E$, where $B=WQ$, and the prediction model is complete.

One additional matrix which is necessary for a complete description of partial least squares regression procedures is the p by c factor loading matrix N which gives a factor model:

$$X=TN'+F \tag{3.38}$$

where F is the unexplained part of the X scores [17].

Generally, PLS regression combines features from PCR and other multiple regression techniques.

3.3.3 One Classical Algorithm Description

NIPALS Algorithm

The standard algorithm for computing PLS regression components (i.e., factors) is *nonlinear iterative partial least squares* (NIPALS). There are many variants of the NIPALS algorithm which normalize or do not normalize certain vectors. For given two data blocks, X, N times (observations) P matrix (variables), and Y, N times M matrix, the algorithm reduces to [17]:

1. Select a P-weight vector w, e.g., a non-zero row of X. Normalize it to length 1.
2. Compute a score vector $t = Xw$.
3. Compute a Y-loading vector $q = Y^{\mathrm{T}}t$.
4. Compute a Y-score vector $u = Yq$.
5. Compute a new weight vector $w_1=X^{\mathrm{T}}u$. Scale w_1 to length 1.
6. If $|w-w_1|<$eps, the convergence is obtained, otherwise $w=w_1$ and start at 2.

The results of the iterations are two score vector, one for X, t, and one for Y, u. Assuming that these results are good choices, the question was now how to get the next pair of (t,u) score vectors. Svante suggested that X should be adjusted for the score vector and regression of Y onto t should be computed and Y adjusted for the results found. This gives

7. Compute the loading vector $n = X^{\mathrm{T}}t/(t^{\mathrm{T}}t)$
8. Adjust X for what has been found: $X_{\mathrm{new}} = X\text{-}tn^{\mathrm{T}}$
9. Compute regression of Y onto t: $b=(Y^{\mathrm{T}}t)/(t^{\mathrm{T}}t)$
10. Adjust Y for what has been selected: $Y_{\mathrm{new}}=Y\text{-}t\ b^{\mathrm{T}}$
11. If more pairs (t,u) are needed go to 1, with $X=X_{\mathrm{new}}$ and $Y=Y_{\mathrm{new}}$

The X loading matrix N is made up of n-vectors, and because of the assumption that the original X matrix comprises p variables, therefore

$$N = \begin{bmatrix} n_{11} & n_{21} & \cdots & n_{p1} \\ n_{12} & n_{22} & \cdots & n_{p2} \\ \vdots & \vdots & \ddots & \vdots \\ n_{1p} & n_{2p} & \cdots & n_{pp} \end{bmatrix} \tag{3.39}$$

where the first column represents the first X loading vector on the first PLS component, the second column represents the second X loading vector on the second PLS component, etc., and n_{ij} is the loading of the ith original variable on the jth PLS component.

By the loading matrix N, the relationship between the original standardized data set X and the PLS component scores matrix $T = (t_1, t_2, \ldots, t_p)$ can be expressed as equation 3.16.

or in an equivalent form

$$\begin{aligned} x_1 &= t_1 n_{11} + t_2 n_{21} + \ldots + t_p n_{p1} \\ x_2 &= t_1 n_{12} + t_2 n_{22} + \ldots + t_p n_{p2} \\ &\vdots \\ x_p &= t_1 n_{1p} + t_2 n_{2p} + \ldots + t_p n_{pp} \end{aligned} \tag{3.40}$$

Normally we choose first k PLS components to do the regression as the final model, and also choose the first k ones when doing the data analysis by a combination method of PLS and for example LDA. Therefore, the equation 3.38 is transformed into

$$X = T_k N'_k + E \tag{3.41}$$

where N_k is a matrix of the first k columns of loadings matrix N, T_k is a matrix of the first k columns of scores matrix T, and E is the error term which also could be referred as unexplained part of the X scores, or it can be equivalently expressed as

$$\begin{aligned} x_1 &= t_1 n_{11} + t_2 n_{21} + \ldots + t_k n_{k1} + \varepsilon_1 \\ x_2 &= t_1 n_{12} + t_2 n_{22} + \ldots + t_k n_{k2} + \varepsilon_2 \\ &\vdots \\ x_p &= t_1 n_{1p} + t_2 n_{2p} + \ldots + t_k n_{kp} + \varepsilon_p \end{aligned} \tag{3.42}$$

where $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_p$ stand for the error of variable $x_1, x_2, \ldots, x_p$ respectively.

From the X loadings N, we can determine the amount of variance of each original variable accounted for by any number of PLS components. The general expression for the variance accounted for in variable x_i by the first k retained PLS components is

$$\sum_{j=1}^{k} (n_{ji})^2 \tag{3.43}$$

When all the PLS components are retained, the result of above equation is 1 and it is also a potential method for variable selection [16].

3.3.4 Determination of Number of Components

It is quite essential to find out how many components are necessary for the subsequent regression model as PCA. The following statistics are important for such evaluation:

Mean Square Error

Technically, the *mean square error* (MSE) of a predicted value $\hat{y}$ of an observation y in a statistical model is defined as:

$$MSE(\hat{y}) = E\left[(\hat{y} - y)^2\right] \tag{3.44}$$

Equation 3.44 is calculated for each observation. Since PLS regression model concerns substantial observations, the MSE is expanded as a performance evaluation statistic by summing the squared differences and taking their means over all observation:

$$MSE = \frac{\sum_{i=1}^{n} (\hat{y} - y)^2}{n} \tag{3.45}$$

Note that, the smaller value of MSE it calculates, the better performance of the model one obtains.

The NIPALS algorithm mentioned in the previous section, it is also referred as the *classical orthogonal scores* algorithm (OSCORESPLS). The first iteration of such algorithm does the regression using the first PLS component, and each subsequent iteration adds one more PLS component to build the model till the last iteration which uses entire PLS components to build the model. If one assume that the *X* (explanatory) matrix consists of *p* variables, so that there are *p* models with 1, 2, …, *p* components respectively, and correspondingly *p* MSE values are calculated. In addition, one should calculate the MSE for a model with zero components. Each MSE is an uncertainty value, and its standard error determines the uncertainty and is calculated by [18]:

$$\text{Standard Error} = \frac{sd\left(\text{Modelling Error}^2\right)}{\sqrt{n}} \tag{3.46}$$

where the numerator of equation 3.46 is the standard deviation of squared modeling error of *n* observations. Below is an example plot for MSE values and their corresponding standard errors of all possible models [18]:

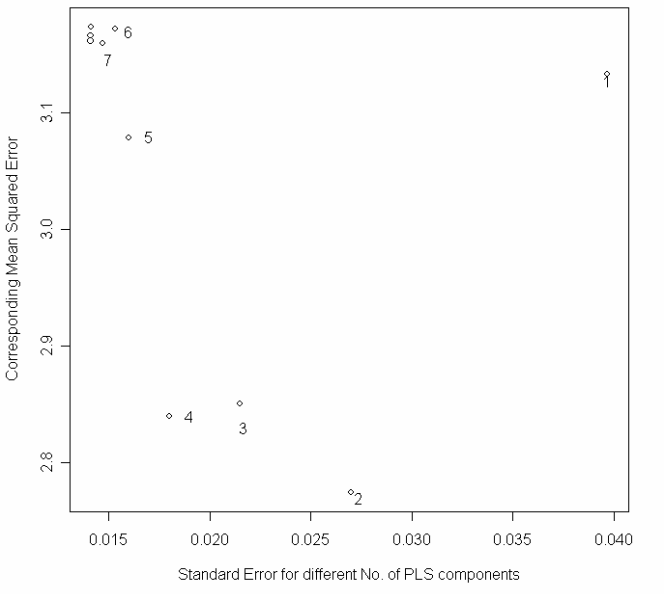


Figure 3.4: MSE curve and its standard errors

In a common sense, the model with the least MSE value is to be chosen, however due to the uncertainty of MSE values and the criterion for choosing the model as less complex as possible, we choose the model with the least components within one standard error of the best, therefore the model with two components for the example above is chosen [18].

Chapter 4

Statistical Implementation of the Theoretic Methods

4.1 Linear Regression with Time-varying Coefficients

4.1.1 Determination of Minimum Adequate Model

According to section 2.4, i.e. the estimation of *autocorrelation function* (ACF), the response series *current status travel time* per 5 minutes (CST5) is more close to an AR(2) or AR(3) process. Thus, it suggests that we may consider first 2 lagged values as well as the exact 1 day after lagged value to predict the response. Initially, the model could be the following form:

$$\begin{aligned} CST5(t) = {} & CST5(t-1) + CST5(t-2) + CST5(t-49) + mcount5(t-1) + mcount5(t-2) \\ & + mcount5(t-49) + TOD + weekday + \varepsilon \end{aligned} \tag{4.1}$$

with corresponding model parameters and prediction error ε. TOD and mcount5 stand for *time of day* and *mean of number of passing vehicles per 5 minutes*. As for the lagged explanatory variables of CST5 and mcount5, other than past observations of one and two time difference, observations of same time of past one day is included as well. In addition, 'weekday' is considered as the only factor in the model.

First, one needs to decide the function forms of covariates in the initial model 4.1. Sometimes the relationship between response and certain covariate is not specified by some explicit function form coming from any theory or mechanistic model. Under this circumstance, the *Generalized Additive Models* (GAMS) is used by fitting non-parametric smoothers to the data without specifying any particular mathematical model to describe the non-linearity [19]. Note that smooth terms are represented using penalized regression splines with smoothing parameters selected by GCV/UBRE or by regression splines with fixed degrees of freedom (mixtures of the two are permitted). Multi-dimensional smoothes are available using penalized thin plate regression splines or tensor product splines [20].

After formulating the GAMS, the following plots are generated to evaluate certain relationships between response and covariates:

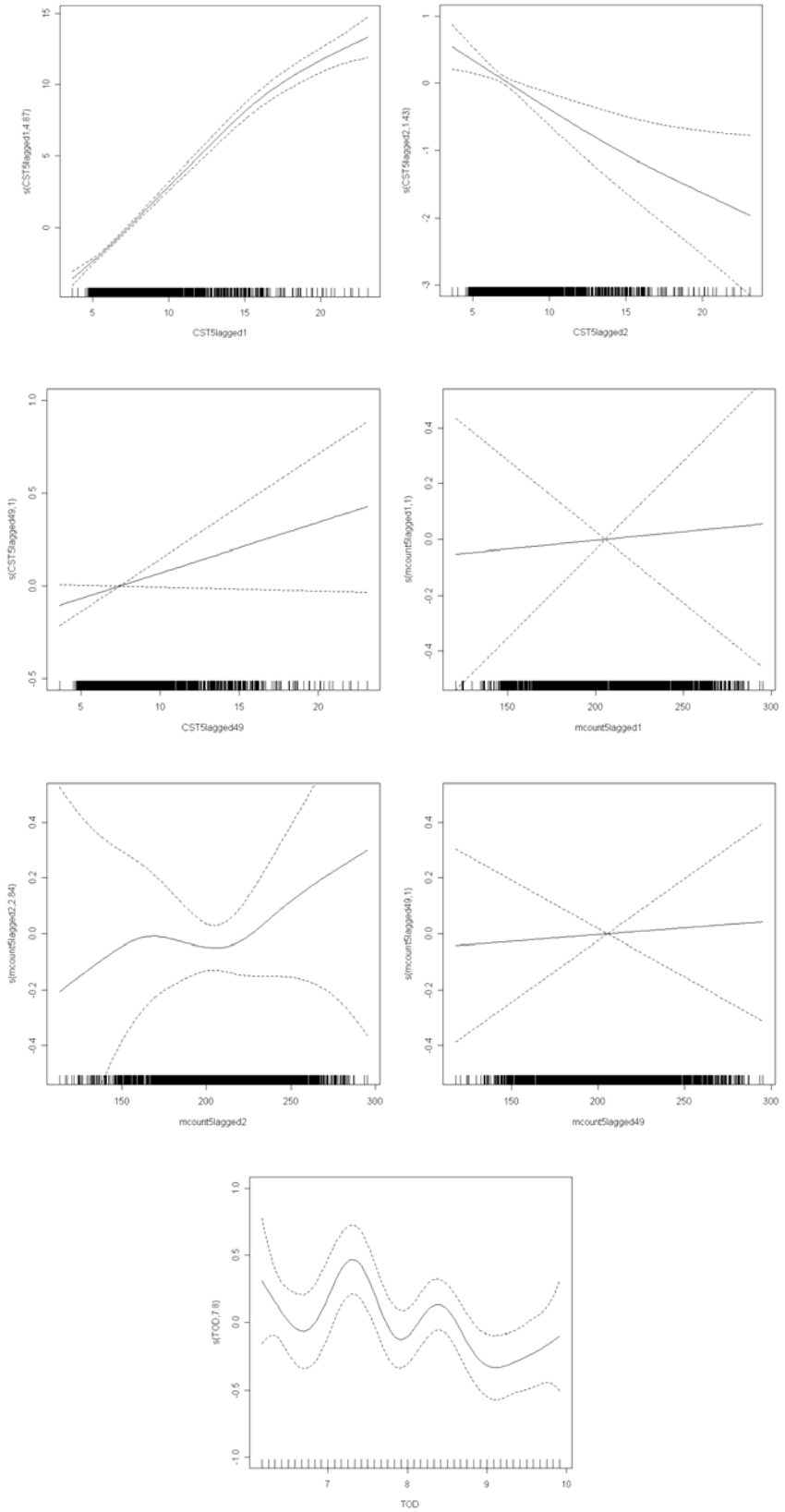


Figure 4.1: The plots of non parametric spline smoothers for each covariate (continuous) of initial model 4.1

Clearly seen linearly explained, there is no need to make non-parametric splines for other covariates

except for 'TOD' and 'mcount5lagged2'. Note that the following outliers are taken away according to the above figure:

$$CST5 \,\&\, CST5lagged1 \,\&\, CST5lagged2 \,\&\, CST5lagged49 \geq 30 \tag{4.2}$$

$$mcount5 \,\&\, mcount5lagged1 \,\&\, mcount5lagged2 \,\&\, mcount5lagged49 \leq 50 \tag{4.3}$$

After updating the initial GAMS, the coefficients and other statistics for the new model is shown below:

Family: gaussian
Link function: identity
Formula:
CST5 ~ CST5lagged1 + CST5lagged2 + CST5lagged49 + mcount5lagged1 + s(mcount5lagged2) + mcount5lagged49 + weekday + s(TOD)
Parametric coefficients:

	Estimate	Std. Error	t value	Pr (>|t|)
(Intercept)	-3.257e-01	6.822e-01	0.477	0.633140
CST5lagged1	9.938e-01	3.519e-02	28.237	< 2e-16 ***
CST5lagged2	-1.063e-01	3.519e-02	-3.021	0.002552**
CST5lagged49	3.519e-02	1.456e-02	2.417	0.015728 *
mcount5lagged1	6.730e-04	2.854e-03	0.236	0.813626
mcountlagged49	-3.512e-05	1.951e-03	-0.018	0.985642
weekdayMonday	8.795e-02	1.078e-01	0.816	0.414840
weekdayThursday	1.334e-01	1.080e-01	1.235	0.216897
weekdayTuesday	4.342e-01	1.167e-01	3.722	0.000204 ***
weekdayWednesday	4.525e-02	1.089e-01	0.415	0.677948

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Approximate significance of smooth terms:

	edf	Est. rank	F	p -value
s(mcount5lagged2)	1.000	1.000	0.516	0.473
s(TOD)	7.889	9.000	5.227	4.76e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
R-sq.(adj) = 0.822 Deviance explained = 82.4%
GCV score = 2. Scale est. = 2.0014 n = 1883

Table 4.1: Some statistics for the initial model 4.1

Here it shows the 3 lagged value of mcount5 and weekday should be removed from the model. By several steps of analysis of covariance, the minimum adequate model is formulated as the following form:

$$CST5(t) = CST5(t-1) + CST5(t-2) + s(TOD) \tag{4.4}$$

The corresponding statistics are as follow:

Family: gaussian
Link function: identity
Formula:
CST5 ~ CST5lagged1 + CST5lagged2 + s(TOD)
Parametric coefficients:

	Estimate	Std. Error	t value	Pr (>|t|)
(Intercept)	0.65267	0.10112	6.455	1.38e-10 ***
CST5lagged1	1.00614	0.03281	30.664	< 2e-16 ***
CST5lagged2	-0.09111	0.03283	-2.775	0.00558 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Approximate significance of smooth terms:

	edf	Est. rank	F	p -value
s(TOD)	8.071	9.000	9.569	2.20e-14 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
R-sq.(adj) = 0.821 Deviance explained = 82.2%
GCV score = 2.0306. Scale est. = 2.0186 n = 1883

Table 4.2: Some statistics for minimum adequate model 4.4

In model 4.4, there is one spline term, which can be referred to the following figure:

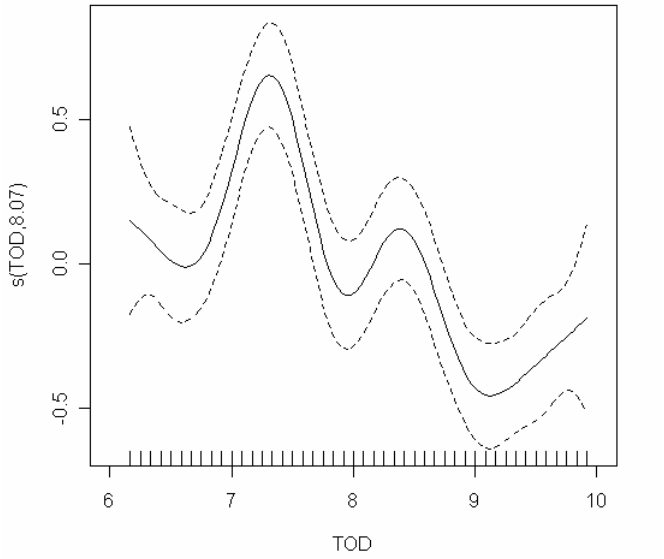


Figure 4.2: Plot for the non-parametric fit of TOD (time of day)

Thus, it is necessary to find a function form to estimate such relationship presenting in figure 4.2. We use 3 important points in the figure, which are the peak, the starting and ending point, to determine such estimate. There are two ways:

- Straight line estimation;
- Combination of curve & straight line estimation.

The following table can be referred as a short description of the methods:

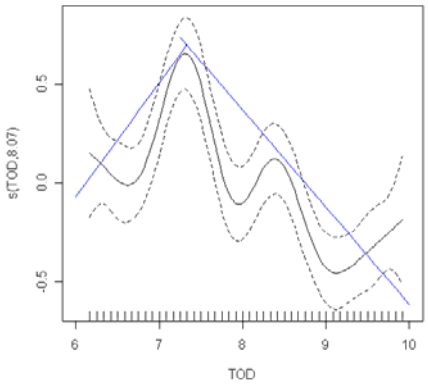
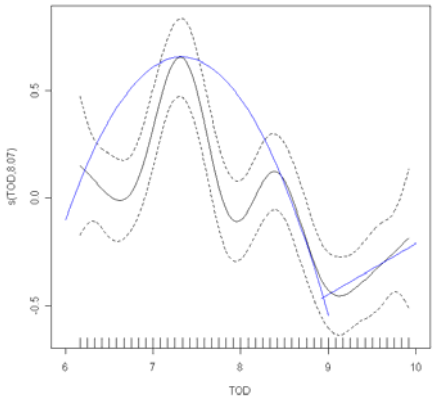
Method	Straight line estimation	Combination of curve & straight line estimation
Formula	$\begin{cases} y = -3.54 + 0.58TOD \\ TOD < 7.33 \\ y = 4.31 - 0.49TOD \\ 7.33 \le TOD \le 10 \end{cases}$	$y = 2.99 \times (TOD - 3.85) - 0.43 \times$ $\begin{cases} (TOD - 3.85)^2 - 4.54 \\ TOD \le 9 \\ y = -2.61 + 0.24TOD \\ 9 < TOD \le 10 \end{cases}$
Fitting plot		
Variance of estimation error	9.6432	5.8691

Table 4.3: The estimation summary of two different methods for non-parametric fit of covariate TOD

Results show that the second method is preferred for the formulation of non-linear term 'TOD' in minimum adequate model 4.4.

4.1.2 Implementation of WRLS and Variables Selection

According to section 2.2.2, the parameters for the regression model 4.1 are time varying. After the model simplification described in previous section, the so-called algorithm *weighted recursive least square* (WRLS) is used here on model 4.4 to obtain varying values of parameters and optimum forgetting factor. The theoretic algorithm is presented in section 3.1.2.

Note that in model 4.4, due to the non-linear relationship between response variable CST5 and explanatory variable TOD, we use its (TOD) combination of curve & straight line estimation of non-parametric fit to replace its original data. According to table 4.3, the replacing formula is as follows:

$$\begin{cases} y = 2.99 \times (TOD - 3.85) - 0.43 \times (TOD - 3.85)^2 - 4.54 & TOD \le 9 \\ y = -2.61 + 0.24TOD & TOD > 9 \end{cases} \quad (4.5)$$

First we have a look at one-step prediction, which is used one time step (5 minutes) ahead to predict the travel time. Normally, the criterion for optimum forgetting factor lies in the value which gives the minimum variance of the residuals (prediction error) [8]. The following plot can be the reference for such variable selection:

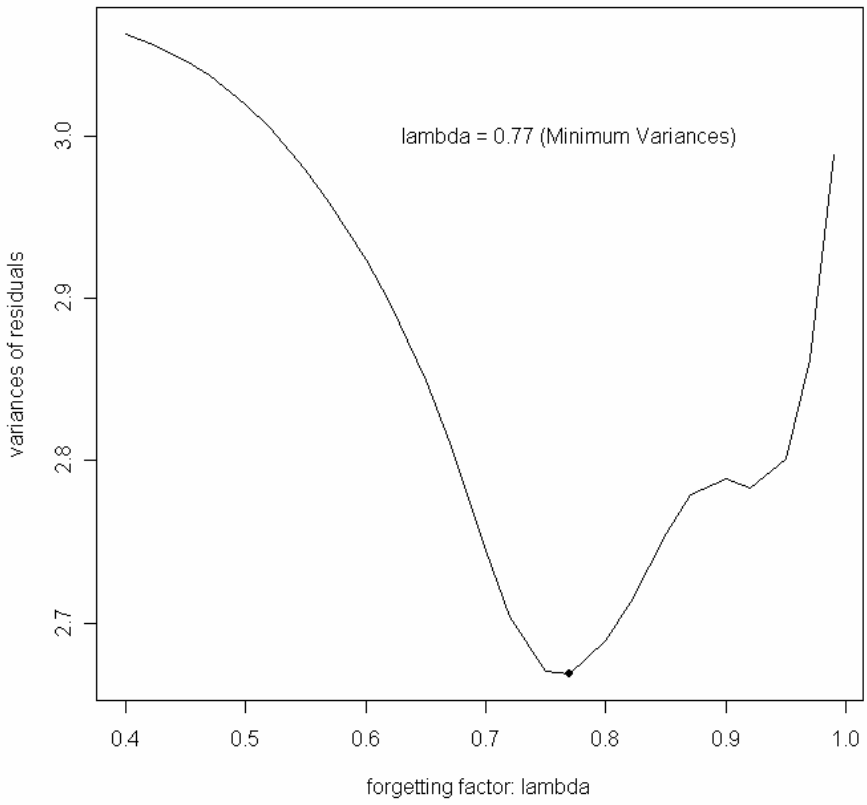


Figure 4.3: The plot for selection of optimal forgetting factor λ according to minimum variance of prediction error (residuals)

As the plot shows, when λ=0.77 the minimum variance of residuals could be obtained approx.:

$$\sigma^2_{\min} = 2.668229$$

for one-step prediction, and the varying parameters for model 4.4 are shown in the following figure:

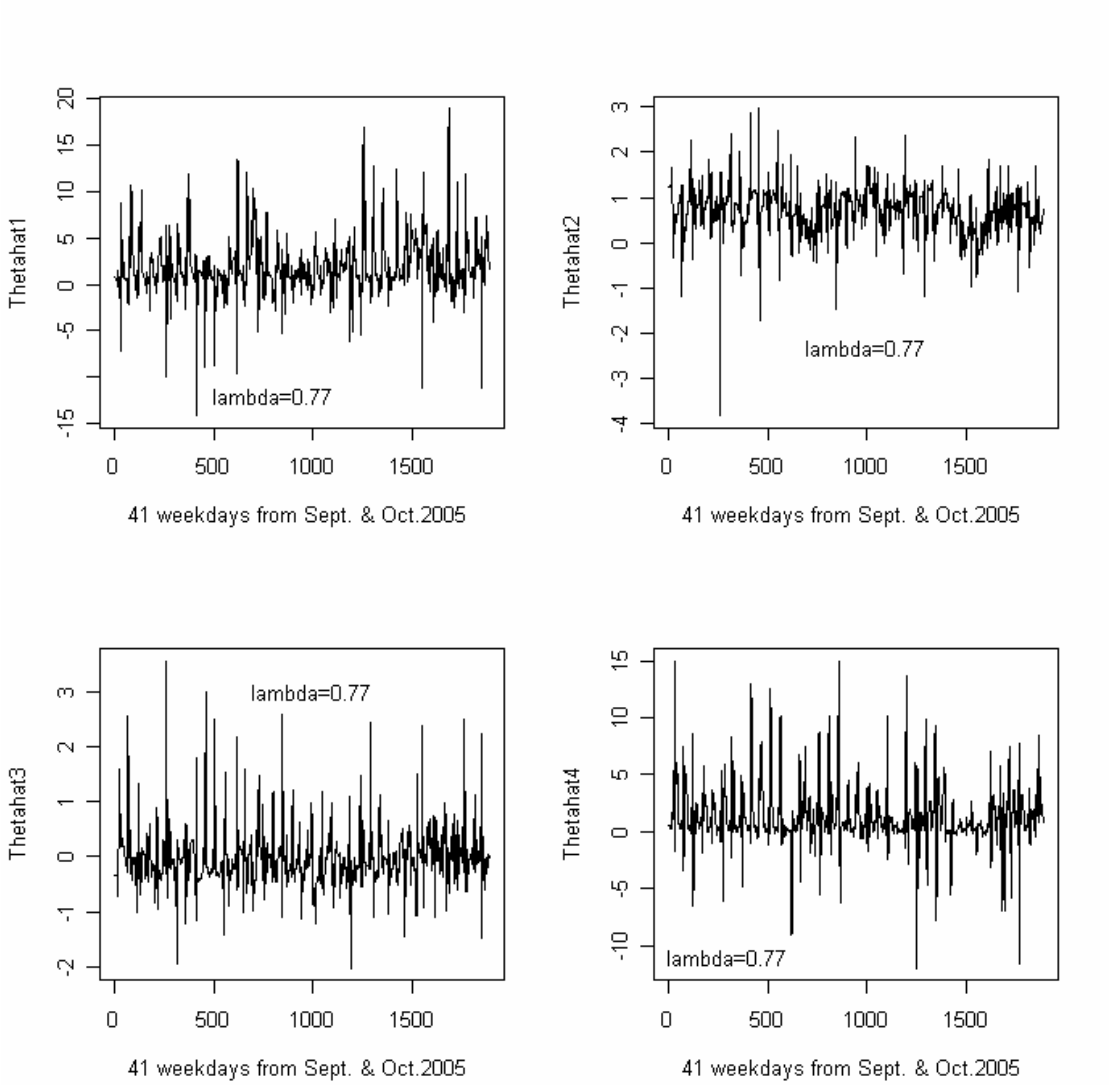


Figure 4.4: The time varying parameters per 5 minutes of 41 weekdays from 6:00~10:00 on Sept. & Oct. 2005 for the initial RLS regression model, when forgetting factor lambda is optimized

The numbers of *X*-axis mean the model parameters' corresponding time in chronicle order. The model seems not so stable, since the ranges of parameters are not strictly within some small enough intervals. This is probably due to the following reasons: first, almost 10% of missing values in the training set made the imputation process unstable and come out effective outliers; second, imprecise estimation of non-parametric fit of explanatory variable 'TOD' results bigger variances of prediction errors.

Next, we look further into *k* steps prediction and their corresponding optimal forgetting factors λ, which is important to know when we should implement such factor. As before, λ is determined by minimum variances of prediction errors.

Prediction Step *k*	Optimum Forgetting Factor λ
1	0.77
2	0.75
3	0.75
5	1
8	1
10	1

Table 4.4: Optimum forgetting factor λ for different prediction steps of RWLS algorithm, each step 5 minutes

Table 4.4 shows that forgetting factor will not be necessary when the prediction step is more than 3, which means that it only works on 15 minutes (3 steps) afterwards prediction. When calculating furthermore steps, RWLS turns to *recursive ordinary least squares* (ROLS).

4.2 Principle Component Analysis

4.2.1 Calculation of Principle Components & Their Related Statistics

If we consider that the whole stretch is divided into small parts one after another by points where certain loop detectors locate, the total traveling time is just the summation of such between each point. Ideally, it would be a perfect regression model when also brings certain lagged values of each partly traveling time to predict the total one. However, one would say this is extremely violating the rule of 'parsimony'. To overcome the trick, PCA is probably a good option.

According to the theory described in section 3.2.1, the first few principle components account for as much of the available information from all the explanatory variables as possible. Thus, it greatly functions as model simplification and dimension reduction. In addition, it is good for explaining correlation structures in the explanatory matrix.

After calculation of PCs by the algorithm presented in section 3.2.2, one needs to determine the least number of them to be used for model formulation. This is achieved by looking into their explained cumulative variance of all explanatory variables.

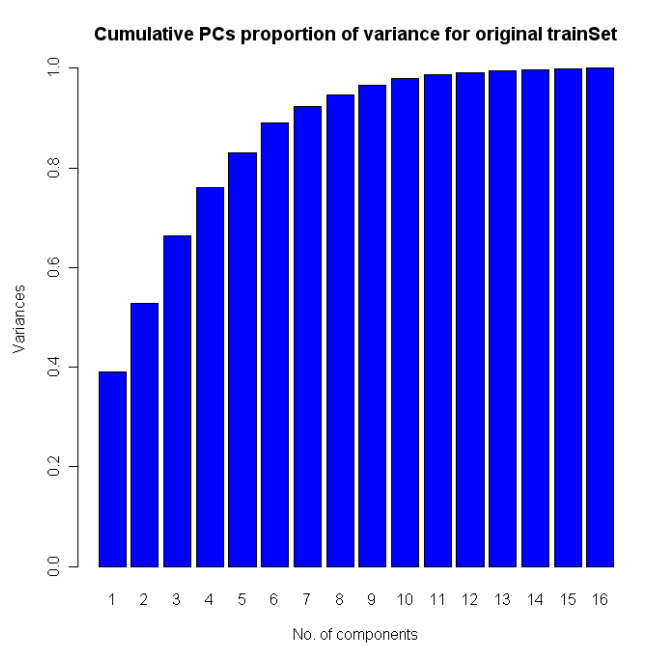


Figure 4.5: Cumulative portion of PCs explained variance for original trainset

There are totally 16 consecutive parts of the stretch from original training set, which outcome 16 PCs corresponding to 16 variables. The above figure shows that the information from first 3 PCs is enough, because they explain cumulatively 70% of the total variance. Next, we take a look into the scores of those 3 components to extract outliers. The following bi-plots are referred:

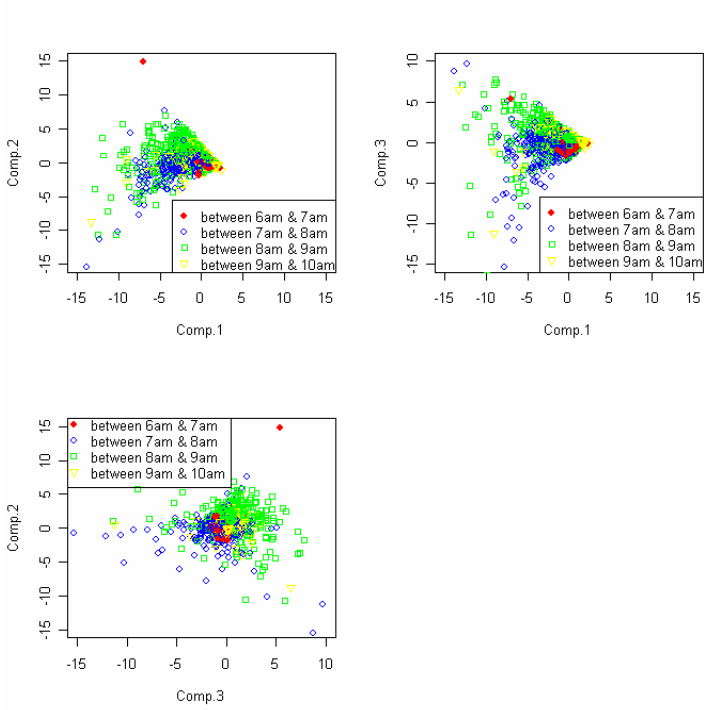


Figure 4.6: Scatter bi-plots of first 3 PCA scores for original trainset

Obviously, there are some points which are far away from the remaining large part of the observations. They are statistically defective and then could essentially result in bias estimation. Thus, such points are considered as outliers and should be taken away from the training set before

subsequent analysis. The description of those outliers and the scores' bi-plots of components after removing them are as follow:

Outlier Row Names	Weekday	Time of Day	Date
172	Tuesday	08:00am	6th Sept.
173	Tuesday	08:05am	6th Sept.
174	Tuesday	08:10am	6th Sept.
175	Tuesday	08:15am	6th Sept.
180	Tuesday	08:40am	6th Sept.
559	Friday	07:35am	16th Sept.
560	Friday	07:40am	16th Sept.
562	Friday	07:50am	16th Sept.
605	Monday	07:20am	19th Sept.
1150	Tuesday	07:50am	4th Oct.
1343	Monday	07:35am	10th Oct.
1345	Monday	07:40am	10th Oct.
1409	Tuesday	09:00am	11th Oct.
1505	Thursday	08:50am	13th Oct.
1572	Tuesday	06:15am	18th Oct.
1844	Tuesday	08:30am	25th Oct.
1849	Tuesday	08:55am	25th Oct.
1851	Tuesday	09:05am	25th Oct.

Table 4.5: Outliers extracted from training set by PCA

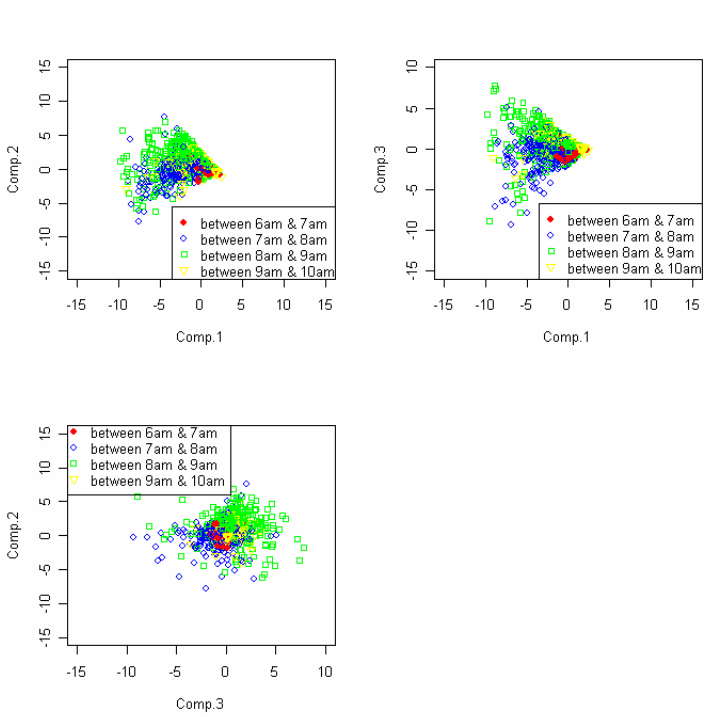


Figure 4.7: Scatter bi-plots of first 3 PCA scores for trainset.out

As discussed before in section 3.2.2, PCL is very useful to explain the correlation of *X* variables as well as identification their features on principle components. At first, one should look into the loading plots below:

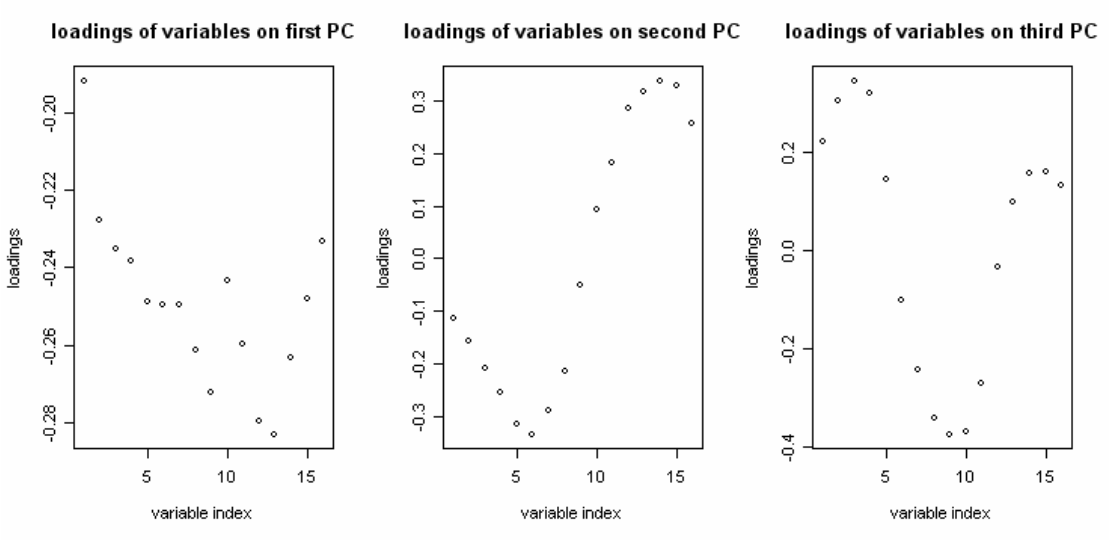


Figure 4.8: Scatter plots of variable loadings on first 3 PCs for trainset.out

Then we can draw following conclusions from those plots:

- The first PC does not show much difference for features inherited from most of the variables, since their loadings on it just range from -0.23 to -0.28;
- The second and third PC could identify some of the important variables which have comparatively significant loadings on them. Those variables exactly stand for certain points of the road where corresponding loops locate. Therefore, the above plot tells that the 3rd, 6th, 9th, 10th and 14th variables, which are actually data from loop M15, PN7, M11, PN10 and PN12, should be paid solid attention.

Note that the second conclusion is the basis for an alternative simplified PCR which will be described in the subsequent section.

4.2.2 Analysis of Variance-Using Principle Components Regression for Prediction

According to last section, the first 3 PCs are enough to interpret the information contained in all the X variables. Here their scores are used to predict the response CST5 by a regression model, also certain lagged values of them are taken into account because the response is more close to an AR(2) or AR(3) process:

$$\begin{aligned} CST5 = {} & PC1(t-1) + PC1(t-2) + PC1(t-49) + PC2(t-1) + PC2(t-2) + PC2(t-49) \\ & + PC3(t-1) + PC3(t-2) + PC3(t-49) + \varepsilon \end{aligned} \tag{4.6}$$

Then analysis of variance is implemented as before in section 4.1.1 to obtain the following minimum adequate model:

$$\begin{aligned} CST5 = {} & PC1(t-1) + PC1(t-2) + PC1(t-49) + PC2(t-1) + PC2(t-2) + PC2(t-49) \\ & + PC3(t-2) + PC3(t-49) + \varepsilon \end{aligned} \tag{4.7}$$

4.2.3 An Alternative Simplified PCR Model

The second conclusion from the PCL plots draws our attention to few important *X* variables, which show significant loadings on 2nd and 3rd PC. Since such components contain most of the *X* variables information, those few variables could also bear enough information to predict the response. Thus, the new dataset is extracted from those important variables of the original training set (outliers away). Only data from loop M15, PN7, M11, PN10 and PN12 are considered, the number of total components is then five.

Likewise, at first the necessary number of components is determined by the following plot:

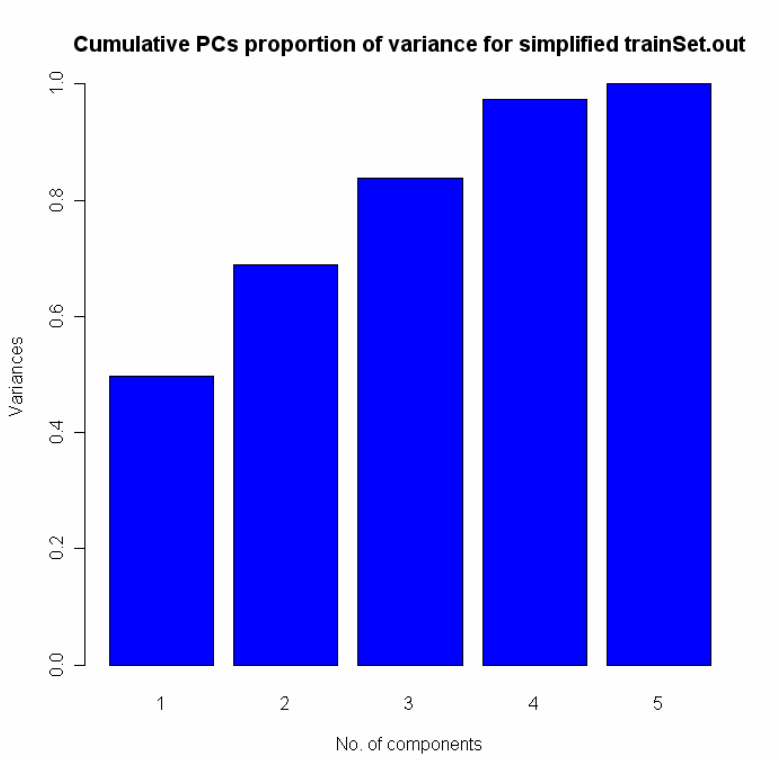


Figure 4.9: Cumulative portion of PCs explained variance for simplified trainSet.out

The first 2 PCs cumulative proportion of explained variance reaches over 70%, which proves they contain enough information for the subsequent analysis.

In order to identify the possible new outliers, it is also necessary to look into the components scores bi-plot:

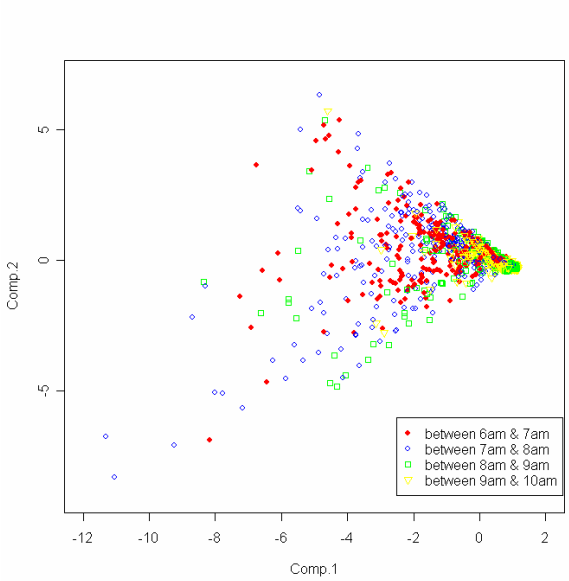


Figure 4.10: Scatter bi-plots of first 2 PCA scores for simplified trainset.out

The outliers are still existed. For remaining the majority of points, the criterion is defined by the following 2 inequalities:

$$Comp1 \geq -6 \tag{4.8}$$

$$-5 \leq Comp2 \leq 5 \tag{4.9}$$

Thus, any point that does not conform the above inequalities are considered outliers and then removed away.

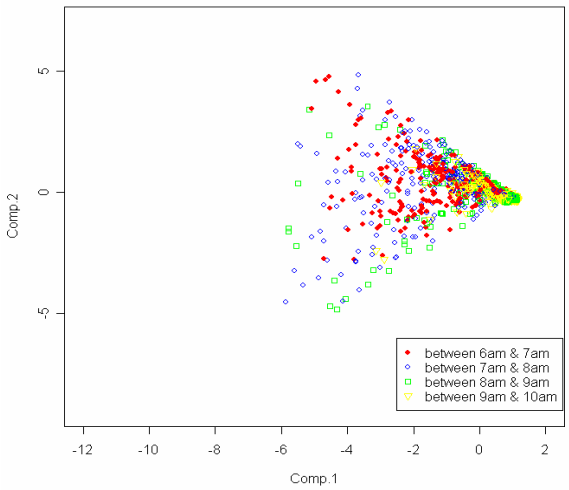


Figure 4.11: Scatter bi-plots of first 2 PCA scores for outliers removed simplified trainset.out

Similarly, the initial simplified PCR model has the following form as model 4.6:

$$CST5 = sPC1(t-1) + sPC1(t-2) + sPC1(t-49) + sPC2(t-1) + sPC2(t-2) + sPC2(t-49) + \varepsilon \tag{4.10}$$

The same process of ANOVA is performed to reach the following minimum adequate model:

$$CST5 = sPC1(t-1) + sPC1(t-2) + sPC1(t-49) + sPC2(t-1) + sPC2(t-2) + sPC2(t-49) + \varepsilon \quad (4.11)$$

which is exactly the same as model 4.10. This is due to the equal significance of each X variable.

4.3 Partial Least Square Regression

For the initial regression model 4.1, the PLS finds the best orthogonal linear combinations of X variables for predicting response variables. The question still is: how many of PLS components are necessary to sufficiently explain the response without breaking the rule of 'parsimony'?

As it referred to section 3.3.4, a research for the combination of cross validation MSE and their corresponding standard errors is implemented to answer the above question. We interpret it by the following plot:

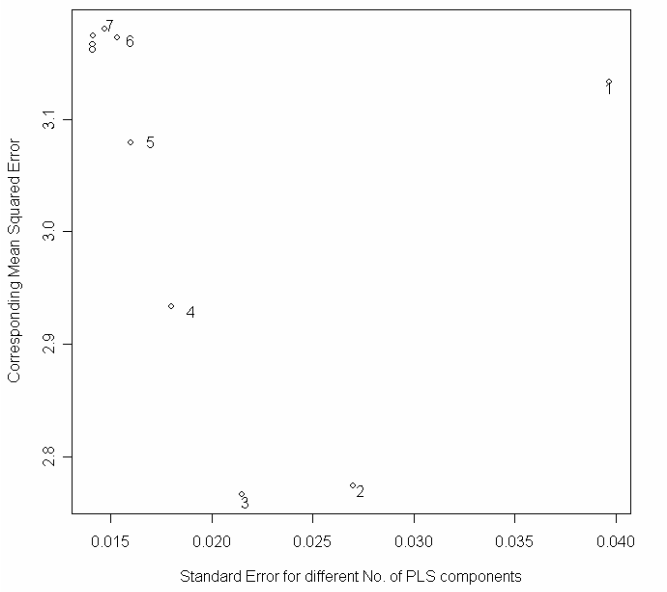


Figure 4.12: MSE of CV and their corresponding standard errors according to certain number of PLS components

The numbers in the above plot stand for number of components. When considering only first 3 PLS components, it will have the best combinative results:

- It has the least MSE value of all and such value increase largely if looks into any other bigger number of components;
- It has a very much acceptable low value of standard error, which is an important merit comparing to its right neighboring point. Though less number of components and close value of MSE, such point has a much bigger value of standard error.

A clear comparison is made for these 2 points in the table below:

No. of Components	MSE	Standard Error
3	2.766245	0.021749
2	2.773952	0.026568

Table 4.6: Comparison of some statistics for 2 possible values of the number of PLS components

Next, the performance of model is initially evaluated by the scores and loading plots as it below:

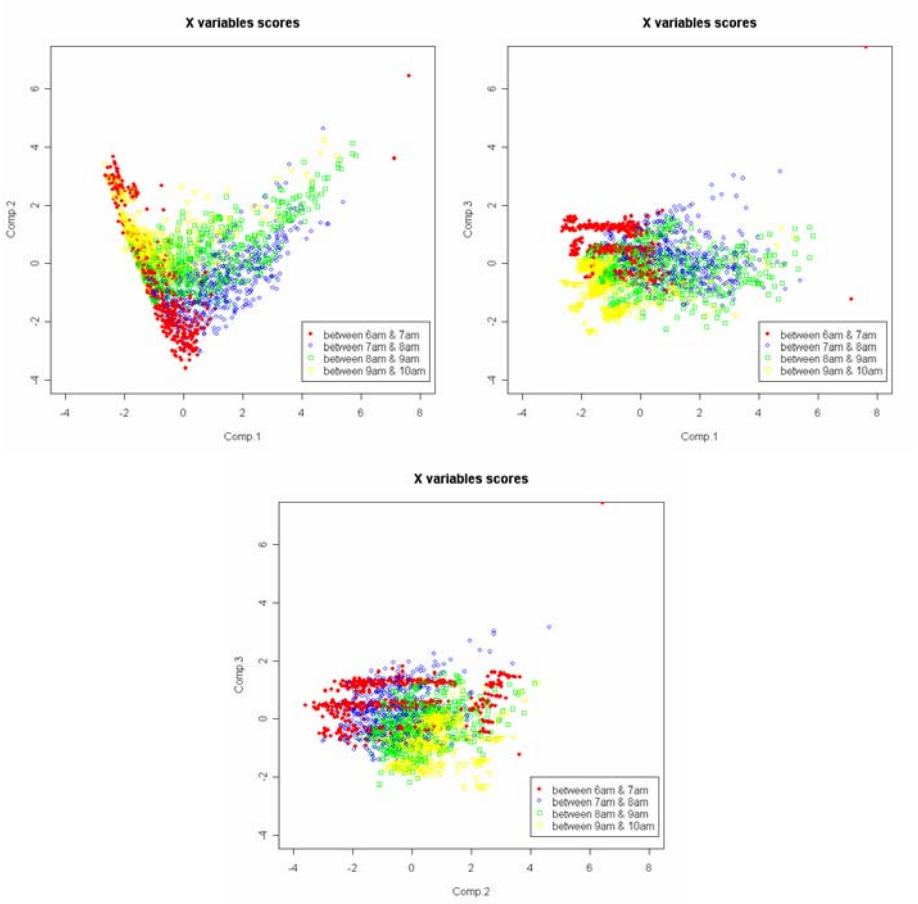


Figure 4.13: Scatter bi-plots of first 3 PLS components scores for PLSR model

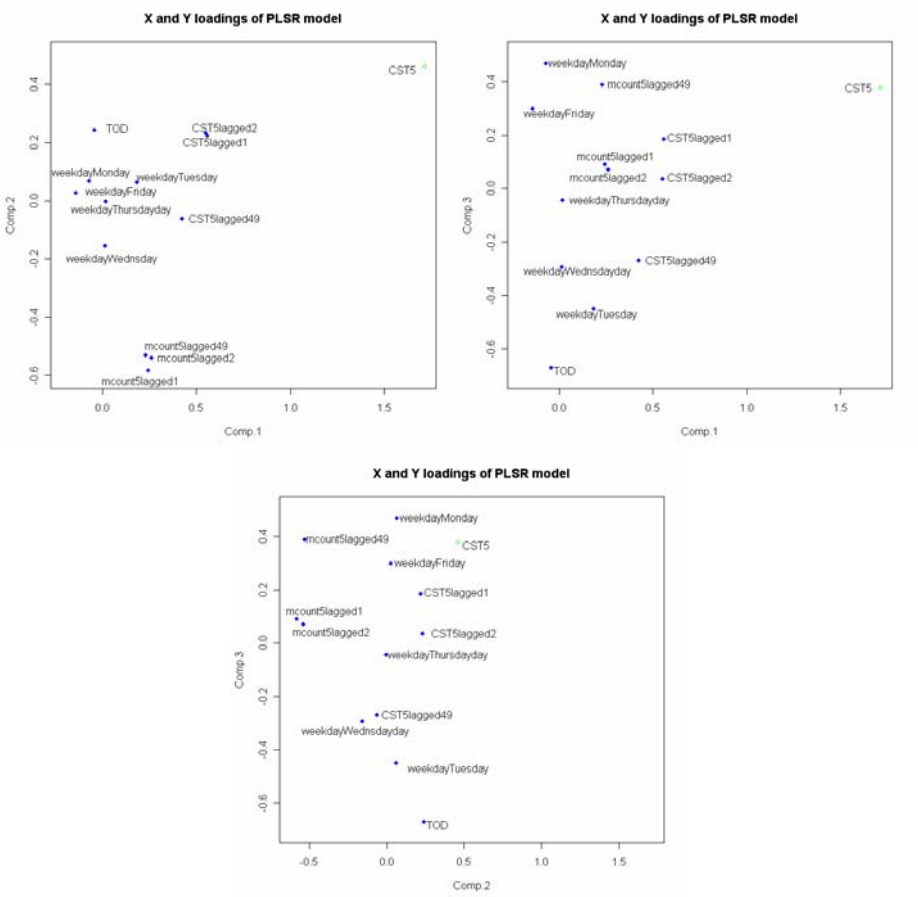


Figure 4.14: Scatter bi-plots of *X* & *Y* variables loadings on first 3 PLS components for PLSR model

From the loading plots above, one can conclude that such PLSR model is not a bad one for predicting the response, since the loadings of response variable '*CST5*' on those PLS components are comparatively bigger enough compared to most of the explanatory variables in most cases.

4.4 Brief Summary

The above methods are proved to be statistically feasible and have their corresponding pros and cons during the modeling process.

Further researches will be carried out in the later section to compare each method's prediction performance and then conclude certain use of methods under different circumstances

Chapter 5

Performance Evaluation & Conclusions

5.1 Prediction Performance Comparison

This thesis presents 3 different methods to predict the travel time for the certain stretch. In order to evaluate their corresponding prediction performance, some statistics of prediction errors (residuals) are calculated according to certain prediction steps:

$$\varepsilon_{i,k} = Y_{i+k} - \theta_i \cdot X_i \tag{5.1}$$

where *i* is the index of certain variable, *k* is the prediction steps which means *k* lags, Y_{i+k} stands for the (*i*+*k*)th observation, X_i stands for the *i*th matrix of explanatory variable and θ_i stands for its corresponding coefficient.

Those statistics of errors (equation 5.1) will be presented later in this section according to certain method, both implemented through training set and test set. Thus, those datasets are again described as follows:

Dataset	Loops incl.	Time	No. of observations
Training set	From PN2 to M7	Sept. & Oct. 2005	20244
testset.sametime	From PN14 to M0	Sept. & Oct. 2005	20244
testset.difftime	From PN2 to M7	Nov. 2005	10122

Table.5.1 Description of training set and test set used in this thesis

The loops are all allocated from south to north and can be referred one by one through the road map of Figure 2.1. As for the test set, it is split into two parts referring to training set:

- Same time period but different part of stretch, as testset.sametime;
- Same part of stretch but different time period, as testset.difftime.

Before comparing model performances, we should take a look at the fitting residuals plots (by training set) for each method:

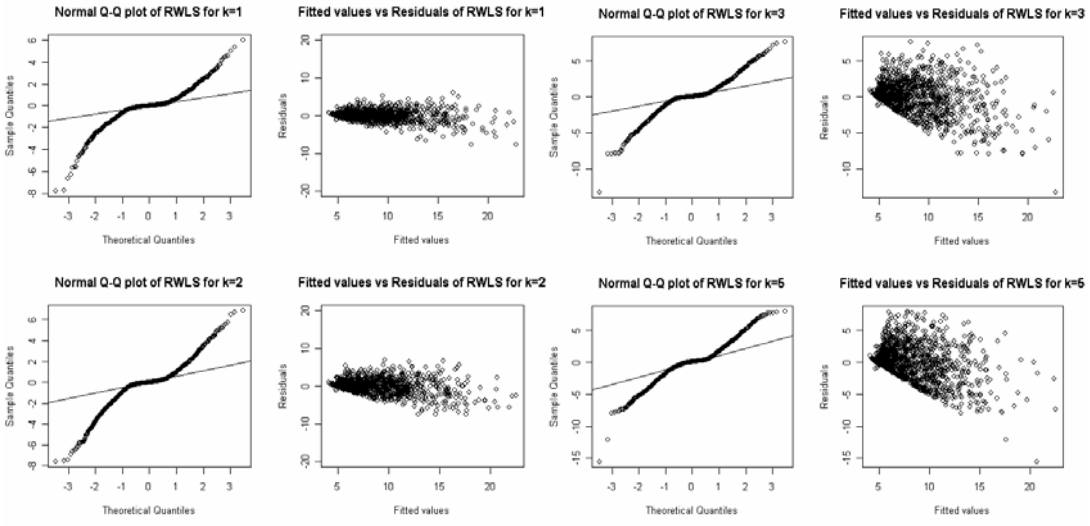


Figure 5.1: Fitting residuals plots of RWLS for different prediction step *k*

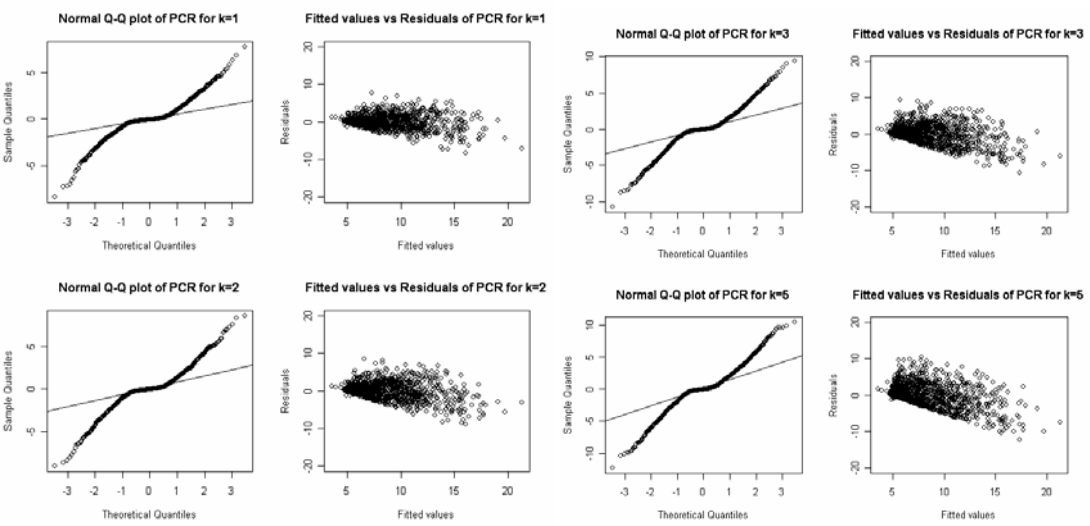


Figure 5.2: Fitting residuals plots of PCR for different prediction step *k*

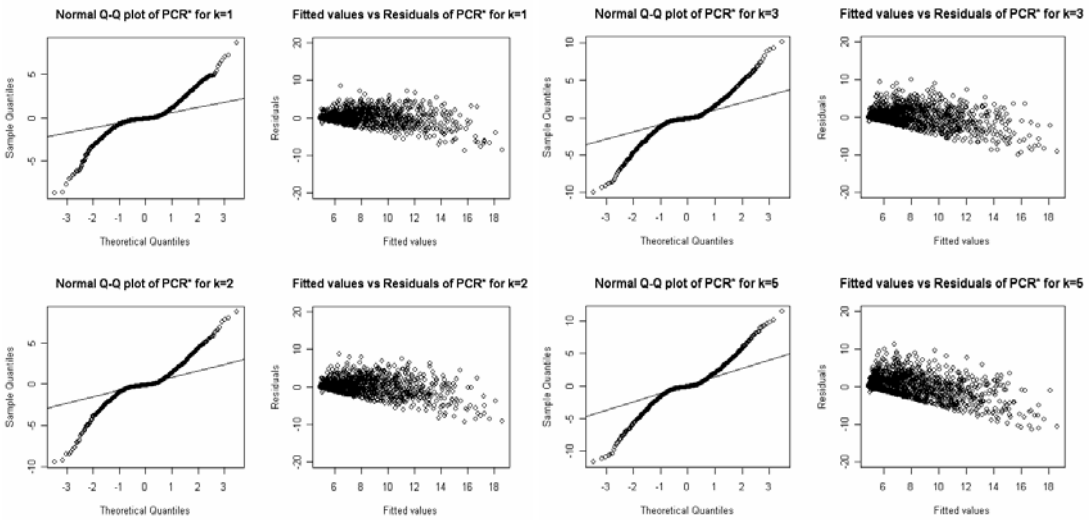


Figure 5.3: Fitting residuals plots of simplified PCR for different prediction step *k*

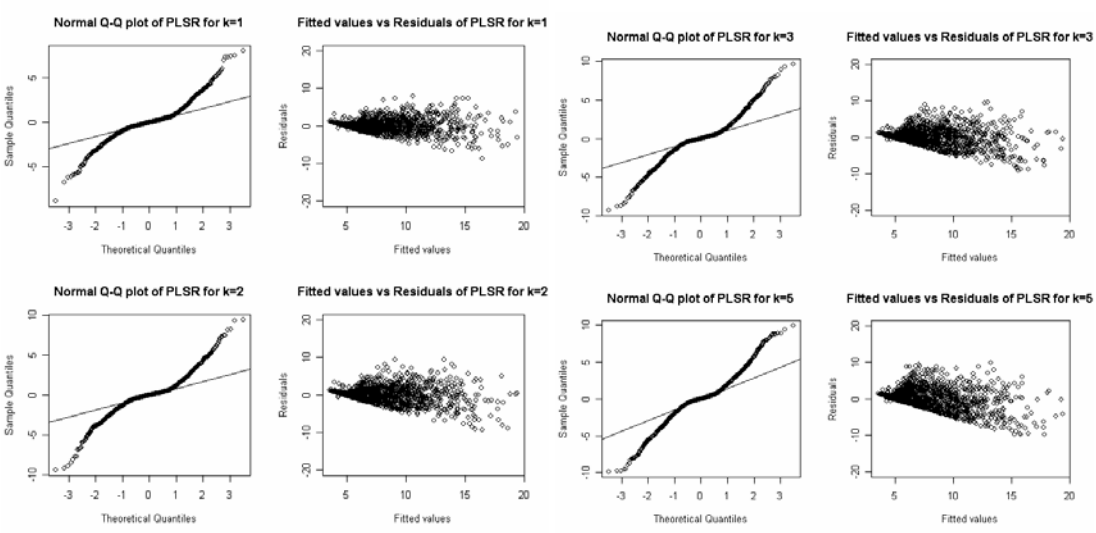


Figure 5.4: Fitting residuals plots of PLSR for different prediction step *k*

One can obtain the following information from the above plots:

- The fitting residuals from each prediction method in the Q-Q plots are all shown marked S-shape, indicative of non-normality. This proves that the models are not fully adequate.
- Theoretically, a good model must account for the variance-mean relationship adequately and produce additive effects on the appropriate scale. A plot of residuals against fitted values should look like the sky at night (points scattered at random over the whole plotting region), with no trend in the size or degree of scatter of the residuals [18]. Typically, within prediction step k=2, the variance of errors is constant; when getting larger steps, an slight expanding and fan-shaped pattern is shown which indicates the variance increases with the mean.

Next, the tables below list the means and UCL/LCL of prediction errors calculated from different statistical methods, both implemented in training set and test set.

Note that *ordinary least square* is also included due to clearer comparison and better interpretation. In addition, the *simplified principle component regression* is denoted by 'PCR*'.

Method	Mean of Errors $E_{k,\varepsilon}$ (by different prediction steps k)				UCL/LCL of Errors $\pm 2\sigma_{k,\varepsilon}$ (by different prediction steps k)			
	k=1	k=2	k=3	k=5	k=1	k=2	k=3	k=5
OLS	-1.5354e-05	4.8895e-05	0.0002	0.0004	3.8835	4.4805	5.0158	5.8680
RWLS	-0.0115	-0.0079	-0.0076	0.0047	3.2827	4.0263	4.7055	5.7806
PCR	0.0001	0.0002	0.0005	0.0010	2.8064	3.5460	4.1659	5.1305
PCR*	0.0002	0.0003	0.0005	0.0009	3.0719	3.6053	4.0939	4.9798
PLSR	-0.0002	-0.0002	-5.9292e-5	5.9797e-5	3.8175	4.3970	4.9053	5.7550

Table 5.2: Comparison of prediction errors from different methods for trainset.out

Method	Mean of Errors $E_{k,\varepsilon}$ (by different prediction steps k)				UCL/LCL of Errors $\pm 2\sigma_{k,\varepsilon}$ (by different prediction steps k)			
	k=1	k=2	k=3	k=5	k=1	k=2	k=3	k=5
OLS	0.1738	0.1740	0.1742	0.1752	8.0506	8.3185	8.6417	9.3253
RWLS	-0.0590	-0.0568	-0.0566	-0.0674	7.5640	7.9978	8.3378	8.2084
PCR	1.8557	1.8558	1.8565	1.8580	6.8316	7.1089	7.4644	8.2418
PCR*	1.7097	1.7125	1.7152	1.7200	14.5296	14.3714	14.1533	13.7152
PLSR	1.7187	1.7191	1.7197	1.7210	7.4709	7.6667	7.9178	8.4476

Table 5.3: Comparison of prediction errors from different methods for testset.sametime

Method	Mean of Errors $E_{k,\varepsilon}$ (by different prediction steps k)				UCL/LCL of Errors $\pm 2\sigma_{k,\varepsilon}$ (by different prediction steps k)			
	k=1	k=2	k=3	k=5	k=1	k=2	k=3	k=5
OLS	0.0257	0.0260	0.0261	0.0265	1.9993	2.5004	2.9197	3.6029
RWLS	-0.0066	-0.0048	-0.0040	0.0102	1.4928	1.1791	2.1717	3.6504
PCR	0.3535	0.3540	0.3539	0.3539	2.9457	3.4240	3.8674	4.5243
PCR*	0.3623	0.3624	0.3623	0.3625	2.9418	3.3656	3.7466	4.4648
PLSR	0.2780	0.2776	0.2772	0.2766	2.5349	3.0998	3.5773	4.3813

Table 5.4: Comparison of prediction errors from different methods for testset.difftime

The conclusions for those results are drawn below according to certain circumstance:

- For the training set, each method is statistically feasible due to the acceptable values of $2\sigma_{k,\varepsilon}$ (variances of prediction errors) in corresponding prediction step k. Also can be referred by the values of $E_{k,\varepsilon}$ (mean of prediction errors) which are very close to 0, they are unbiased for fitting. In addition, RWLS shows some benefit compared to OLS, but is inclined to have similar performance when $k \geq 5$. Among all of them, PCR is the best fitting method.
- For the test set of same time period but different part of stretch, the prediction performance for each method is very bad. Thus, it is difficult to predict travel time on a different stretch, probably due to big changes of road conditions, geographic conditions or any factor which could affect.
- For the test set of same part of stretch but different time period, the prediction performance for each method is even better than that of fitting performance. This is possible because of the following 2 reasons: first, the data collection in November is probably more accurate than that of previous two months; second, substantial amount of missing data for both of the datasets could potentially make the imputation process biased, which resulted in better imputation for November. On the other hand, the predictions turn to be biased for PCR, PCR* and PLSR due to values of $E_{k,\varepsilon}$ which stay far away from 0. Also one can see PLSR is better than any of the PCR method. This proves that PLS finds linear combinations of the predictors that better explains the response than that of PCA. Generally, RWLS is the best for prediction almost without bias and again, it turns into OLS when prediction step $k \geq 5$.
- The fitting and prediction performance of PCR* is very close to that of PCR, in some cases it is even better.

5.2 Ideas & suggestions for future works

The previous section presents the fitting and prediction results of different methods and the corresponding conclusions. Not surprisingly, each method has its own pros and cons, which need

future improvement of the analysis. Here are some ideas:

- It is necessary to reduce bias when using PCR and PLSR for prediction.
- It is quite willingly to acquire homogeneity of residuals, i.e. improve the models when predicting travel time long steps behind.
- Use prediction model that is fitted by same stretch of the road because of variations of local phenomena.
- Recursively updating is beneficial, but mostly for small prediction steps.
- Simplified PCR (only considering critical points of the stretch) is more recommended when compared with normal PCR, due to similar prediction performances and principle of 'parsimony'.
- Probably beneficial from combination of models: using information of PCR or PCR* and make the RWLS simplified.

Bibliography

[1] X. Zhang, John A. Rice. Short-term travel time prediction using a time-varying coefficient linear model. Technical report, University of California at Berkeley.
[2] Kutner, M., C. Nachtschiem, W. Wasserman, and J. Neter (1996). Applied linear statistical models (4th ed.) McGraw-Hill.
[3] M. Kang. Statistical analysis of missing data: an overview. Technical report, University of Illinois at Urbana-Champaign
[4] John Rice, Erik van Zwet. A simple and effective method for predicting travel times on freeways. 2001 IEEE Intelligent Transportation Systems Conference Proceedings, page 227-232
[5] R. Gibbens, Wiebke. Werft. MIDAS and journey time predictors. Significance, page 102-105, Royal Statistical Society.
[6] Michael Haag. Autocorrelation of random processes. Connexions Web site, Apr 5, 2005. Available at http://cnx.org/content/m10676/2.4/.
[7] Chatfield, C., 1975, The analysis of time series: Theory and practice, Chapman and Hall, London, page 263. *[cross-correlation function, impulse response function]*.
[8] Henrik Madsen. Time series analysis. Informatics and Mathematical Modeling, Technical University of Denmark.
[9] Fox, J., *Applied Regression Analysis, Linear Models and Related Methods.* (1997), Sage
[10] James M. Lattin, J. Douglas Carroll, Paul E. Green, Doug Carroll, Jim Lattin. Analyzing multivariate data. Stanford University.
[11] Henrik Madsen, Jan Holst. Modeling Non-linear and non-stationary time series. Informatics and Mathematical Modeling, Technical University of Denmark.
[12] James M. Lattin, J. Douglas Carroll, and Paul E. Green: Analyzing Multivariate Data. Thomson Learning, 2003, USA
[13] Richard A. Johnson, Dean W. Wichern: Applied Multivariate Statistical Analysis, Fifth Edition. Prentice Hall, 2002, USA
[14] M. Talbot. Biomathematics and Statistics. ECRR, Edinburgh.
[15] Dave Garson. Quantitative methods in public administration. Technical paper, North Carolina State University.
[16] Randall D. Tobias (1997). An introduction to partial least squares regression. Cary, NC: SAS Institute
[17] Geladi, P. and Kowalski, B. (1986). Partial least squares regression: A tutorial.
[18] Wu, S., *In silico prediction of inhibition of CYP1A2,* Informatics and Mathematical Modeling, Technical University of Denmark, DTU, 2006
[19] Michael J. Crawley (2004). Statistics, an introduction using R. John Wiley & Sons, Ltd.
[20] Simon Wood. The mgcv package in R.

Appendix

R Codes of Main Functions

1. Data Preprocessing & Descriptive Statistics

 # Missing Data Imputation

```
Imp <- function(dataset){ #dataset: dataframe which has missing data
   library(norm)
   dt1 <- .code.to.na(dataset,-1)
   dt2 <- dt1[,3:70]
   dt2 <- as.matrix(dt2)
      for (k in 1:68)    {
                  s <- prelim.norm(dt2[,k])
                  thetahat <- em.norm(s)
                  rngseed(1234567)
                  ximp <- imp.norm(s,thetahat,dt2[,k])
                    dt1[,k+2] <- ximp
                             }
       return(dt1)
                        }
```

 # Response Calculation

```
CST <- function(D,V,a,b)   {
     Tstar <- 0
         for (i in a:(b-1))  {
             T <- 2*D[i]/(V[i]+V[i+1])
             Tstar <- sum(T+Tstar)
                              }
     Tstar <- Tstar*60 #transform to minute
         return(Tstar)
                           }
#D:a vector containing distances between loops;
#V:a vector containing velocities from loops at certain time;
#a: index for the starting point of journey;
#b:index for the ending point of journey
```

 # Harmonic Mean of Velocities and Summation of Count Number in 5 Minutes for Data from Each Loop

```
VlCt <- function(i,j,inv)     { #inv: time interval by second scale
```

```
     time1 <- 4*3600/inv+1      #i: index of starting point of loop
     hm <- 0                    #j: index of ending point of loop
     amount <- 0
     hm1 <- matrix(0,time1,61)
     amount1 <- matrix(0,time1,61)
for(n in c(1,2,5:9,12:16,19:23,26:30,33:37,40:43,47:51,54:58,61))   {
     for (m in 1:time1)  {
         if (n < 60) {#summer time
            g <- SO3[SO3[,1]>SO3[1,1]+(m-1)*inv+(n-1)*24*3600 &
                 SO3[,1]<=SO3[1,1]+m*inv+(n-1)*24*3600,i:j]
                     }
         else if (n >= 60)   { #winter time
           g <- SO3[SO3[,1]>SO3[1,1]+(m-1)*inv+(60*24+1)*3600
                & SO3[,1]<=SO3[1,1]+m*inv+(60*24+1)*3600,i:j]
                             }
         hm[m] <- length(g[,2])/sum(1/g[,2]) #harmonic mean
         amount[m] <- sum(g[,1])
                         }
     hm1[,n] <- hm
     amount1[,n] <- amount
                                                                    }
hm2 <- hm1[,c(1,2,5:9,12:16,19:23,26:30,33:37,40:43,47:51,54:58,61)]
amount2 <-
amount1[,c(1,2,5:9,12:16,19:23,26:30,33:37,40:43,47:51,54:58,61)]
     hmv <- as.vector(hm2)
     mat <- as.vector(amount2)
        list(hmv = hmv,mat = mat)
                              }
```

 # Plot of Harmonic Mean of Velocities in 5 minutes VS Time of Day

```
MT <- SO3[1,1] #SO3: imputed dataframe of September & October
for (i in 1:49) {
   MT[i] <- SO3[1+10*(i-1),1]
             }
plot(MT,CST5[1:49],ylim=c(4,22),xlab='time of day',ylab='current
     status travel time (each 5 mins)',col='grey',type='l')
for (k in 2:42) {
lines(MT,CST5[(49*(k-1)+1):(49*k)],col='grey')
             }
```

 # Impulse Response Function

```
IRF <- function(X,Y,t) {     #X: input series
   library(MASS)             #Y: output series
   library(tseries)          #t: length of series
```

```
   a <- ar(X,order.max = 6)
   Alfa <- a$resid
   Alfa <- na.omit(Alfa)
   AR <- a$ar
   Yhat <-0
   Beta <-0
      for (i in 7:t)  {
       Yhat[i-6] <-
Y[i-1]*AR[1]+Y[i-2]*AR[2]+Y[i-3]*AR[3]+Y[i-4]*AR[4]+Y[i-5]*AR[5]+Y[i-
6]*AR[6]
       Beta[i-6] <- Y[i]-Yhat[i-6]
                       }
 CCV <- ccf(Alfa,Beta,lag.max = 20,type = 'covariance',plot = F)
   va <- var(Alfa)
   IR <- CCV[[1]]/va
   IR <- as.vector(IR)
 plot(-20:20,IR,xlab = 'lag',ylab = 'impulse response function
(IRF)',type = 'h')
    abline(h=0)
                        }
```

 # Preparation of Dataframe Including X and Y Variables

```
DataFrame <- function(dataset,n)  {
                          #dataset: original training or test dataset
TOD <- matrix(0,49,42)            #n: number of days for the dataset
wd <- matrix(0,49,42)
t1 <- dataset[,1]
t1 <- as.character(t1)
t1 <- substr(t1,12,19)
for (m in 1:n) {
   for (k in 1:49) {
      wd[k,m] <- as.character(wd[k,m])
      wd[k,m] <- dataset[(482*(m-1)+(1+10*(k-1))),2]
     TOD[k,m] <- as.character(TOD[k,m])
     TOD[k,m] <- t1[482*(m-1)+(1+10*(k-1))]
                    }
TOD <- strptime(TOD,"%H:%M:%S")
wd <- as.vector(wd)
TOD <- as.vector(TOD)
for (i in 1:(49*n))      {
   if (wd[i]=='1') {
      wd[i] <- 'Friday'
                    }
   else if (wd[i]=='2')  {
```

```
      wd[i] <- 'Monday'
                          }
   else if (wd[i]=='3')    {
      wd[i] <- 'Wednsday'
                          }
   else if (wd[i]=='4')  {
      wd[i] <- 'Tuesday'
                          }
   else if (wd[i]=='5')    {
      wd[i] <- 'Thursday'
                          }
                       }
wd <- as.data.frame(wd)
TOD <- as.data.frame(TOD)
wt <- cbind(wd,TOD)
names(wt)[1] <- 'weekday'
wt <- wt[50:(49*n),]
d1 <- ts(CST5)
d2 <- ts(mcount5)  #CST5 and mcount5 calculated by previous function CST
                    # and VlCt
for (i in c(1,2,49))    {
      d1 <- ts.union(d1,lag(CST5,-1*i))
      d2 <- ts.union(d2,lag(mcount5,-1*i))
                          }
d1 <- na.omit(d1)
d1 <- as.data.frame(d1)
d2 <- na.omit(d2)
d2 <- as.data.frame(d2)
names(d1)[1] <- 'CST5'
names(d1)[2] <- 'CST5lagged1'
names(d1)[3] <- 'CST5lagged2'
names(d1)[4] <- 'CST5lagged49'
names(d2)[2] <- 'mcount5lagged1'
names(d2)[3] <- 'mcount5lagged2'
names(d2)[4] <- 'mcount5lagged49'
Regdata <- cbind(d1,d2[,2:4],wt)
tod1 <- seq(6,10,by=1/12)
tod2 <- rep(tod1,(n-1))
Regdata[,9] <- tod2
  Return(Regdata)
                                    }
```

2. Linear Regression with Time-Varying Coefficients

#Recursive Weighted Least Squares

```
TVP <- function(lambda,k,dataset,n)       { #lamuda: forgetting factor
  library(MASS)                            #k: number of prediction step
  dimension <- dim(dataset)  #dataset: prepared dataset including X & Y
  mdq <- dataset[,c(1:3,9)]              #n: number of rows of outliers
  nd <- dimension[1]/49  #nd:number of days
  for (i in 1:nd)  { #result of non-paramatric fit of GAMS
    mdq[(1+49*(i-1)):(37+49*(i-1)),4] <-
    2.99*(dataset[(1+49*(i-1)):(37+49*(i-1)),9]-3.85)-0.43*
    (dataset[(1+49*(i-1)):(37+49*(i-1)),9]-3.85)^2-4.54
    mdq[(38+49*(i-1)):(49+49*(i-1)),4] <-
    -2.61+0.24*dataset[(38+49*(i-1)):(49+49*(i-1)),9]
                    }
    a <- dimension[1]-n #remaining rows for d1
    b <- matrix(1,1,a)
    X <- rbind(b,t(mdq[,2:4]))
    Y <- matrix(0,a,1)
    Y[,1] <- mdq[,1]
    Thetahat1 <- 0 #estimates of parameters
    Thetahat2 <- 0
    Thetahat3 <- 0;Thetahat4 <- 0;
    CSThat <- 0 #predicted travel time
    re <- 0 #residuals between observed travel time and predicted travel
            time
  for (i in 98:(a-k))  {
       if (i==98)     {
          R <- X[,1:98]%*%t(X[,1:98]) #make initial estimates from first
                                       2 days
          h <- X[,1:98]%*%Y[1:98,]
          Thetahat <- ginv(R)%*%h
          Thetahat1[1] <- Thetahat[1,1]; Thetahat2[1] <- Thetahat[2,1];
          Thetahat3[1] <- Thetahat[3,1]; Thetahat4[1] <- Thetahat[4,1];
          CSThat <- t(X[,1:98])%*%Thetahat
          re <- Y[1:98,]-CSThat
                      }
       else if (i>98)     {
          R <- lambda*R+X[,i]%*%t(X[,i])
          h <- lambda*h+X[,i]*Y[i,]
          Thetahat <- ginv(R)%*%h
          j <- i-97
```

```
          Thetahat1[j] <- Thetahat[1,1]; Thetahat2[j] <- Thetahat[2,1];
          Thetahat3[j] <- Thetahat[3,1]; Thetahat4[j] <- Thetahat[4,1];
          CSThat[i] <- t(X[,i])%*%Thetahat
          re[i] <- Y[i+k,]-CSThat[i]
                          }
                        }
    va <- var(re[99:(a-k)]); me <- mean(re[99:(a-k)])
  list(Thetahat1 = Thetahat1,Thetahat2 = Thetahat2,
       Thetahat3 =  Thetahat3,Thetahat4 = Thetahat4,
       va = va, re = re, me = me,CSThat = CSThat)
                                    }
```

 # Ordinary Least Squares

```
OLS <- function(k,dataset)     {  #k: prediction steps
      a <- dim(dataset)            #dataset: training or test dataset
      b <- matrix(1,1,a[1])
      X <- rbind(b,t(dataset[,c(2:3,9)]))
      Y <- matrix(0,a[1],1)
      Y[,1] <- dataset[,1]
      thetahat <- ginv(X%*%t(X))%*%X%*%Y
      CSThat <- 0
      re <- 0
    for (i in 1:(a[1]-k)) {
      CSThat[i] <- t(X[,i])%*%thetahat
      re[i] <- Y[i+k,]-CSThat[i]
                        }
     va <- var(re); me <- mean(re)
    list(thetahat = thetahat,va = va, re = re, me = me)
                                    }
```

#Prediction Errors of RWLS of Different Prediction Steps

```
RlsPE <- function(k,dataset,coeff)  { #k: prediction steps
      a <- dim(dataset)                #dataset: training or test dataset
      b <- matrix(1,1,a[1])           #coeff: calculated RWLS model
                                       #coefficients from training set
      X <- rbind(b,t(dataset[,c(2:3,9)]))
      Y <- matrix(0,a[1],1)
      Y[,1] <- dataset[,1]
      CSThat <- 0
      re <- 0
      Thetahat <- rbind(coeff[[1]],coeff[[2]],coeff[[3]],coeff[[4]])
    for (i in 1:(a[1]-k)) {
      CSThat[i] <- t(X[,i])%*%Thetahat[,i]
      re[i] <- Y[i+k,]-CSThat[i]
```

```
                          }
     va <- var(re); me <- mean(re)
    list(va = va, re = re, me = me)
                                      }
```

3. Principle Components Analysis

#Components Calculation & Regression Analysis

```
library(stats)
aSegCST5 <- matrix(0,2058,16)
    #CST5 for first 16 segments of the road stretch
for (j in 1:16) {
    for (k in 1:2058)   {
         aSegCST5[k,j] <- CST(D,Nspeed5[k,],j,j+1)
                        }
                }
PCA1 <- princomp(aSegCST5,cor=T,scores=T)
names(PCA1); variance <- PCA1[[1]]^2
Cuvar <- 0 #cumulative proportion of variance
for (i in 1:16) {
    Cuvar[i] <- sum(variance[1:i])/sum(variance)
               }
barplot(Cuvar,names.arg=1:16,col='blue',main='Cumulative PCs
proportion of variance for original trainSet',xlab='No. of
components',ylab='Variances')
abline(h=0.7,lty=2)
text(3,0.95,'critical point:Var=0.7')
bSegCST5 <- aSegCST5[-c(172:175,180,559,560,562,605,1150,1343,1345,
                    1409,1505,1572,1844,1849,1851),]
y <- CST5[-c(172:175,180,559,560,562,605,1150,1343,1345,1409,1505,
          1572,1844,1849,1851)]
y <- as.matrix(y)
PCA2 <- princomp(bSegCST5,cor=T,scores=T)
PC1 <- ts(PCA2$score[,1]); PC2 <- ts(PCA2$score[,2]);
PC3 <- ts(PCA2$score[,3]);
  for (i in c(1,2,49))   {
     PC1 <- ts.union(PC1,lag(PCA2$score[,1],-1*i))
     PC2 <- ts.union(PC2,lag(PCA2$score[,2],-1*i))
     PC3 <- ts.union(PC3,lag(PCA2$score[,3],-1*i))
```

```
                          }
PC1 <- na.omit(PC1); PC1 <- as.data.frame(PC1);  PC1 <- PC1[,-1]
PC2 <- na.omit(PC2); PC2 <- as.data.frame(PC2); PC2 <- PC2[,-1]
PC3 <- na.omit(PC3); PC3 <- as.data.frame(PC3); PC3 <- PC3[,-1]
y1 <- y[-(1:49),]; y1 <- as.data.frame(y1)
names(y1) <- 'CST5'
names(PC1)[1:3] <- c('PC1lagged1','PC1lagged2','PC1lagged49')
names(PC2)[1:3] <- c('PC2lagged1','PC2lagged2','PC2lagged49')
names(PC3)[1:3] <- c('PC3lagged1','PC3lagged2','PC3lagged49')
PCA3 <- cbind(y1,PC1,PC2,PC3)
PCR1 <- lm(CST5~PC1lagged1+PC1lagged2+PC1lagged49+PC2lagged1+
  PC2lagged2+PC2lagged49+PC3lagged1+PC3lagged2+PC3lagged49,data=PCA3)
summary(PCR1)
PCR11 <- update(PCR1,~.-PC3lagged1)
summary(PCR11)
```

#Prediction Errors

```
PE <- function(dataset,model,k)   {
#dataset:new dataset,null if calcilates training set's prediction errors;
    a <- dim(dataset)[1]         #model: formulated model for prediction;
                                  #k: number of prediction steps
    re <- 0
    ftv <- predict(model,dataset)
    for (i in 1:(a-k))   {
      re[i] <- dataset[i+k,1]-ftv[i]
                         }
    va <- var(re)
    me <- mean(re)
    list(va = va, me = me, re = re, ftv = ftv[1:(a-k)])
                                      }
```

4. Partial Least Squares Regression

#PLSR & Components Selection

```
PLSR <- function(dataset,x,y,Ndata) {#dataset: dataset to do regression
library(pls)   #x: explanatory matrix y: response matrix
              #Ndata: new dataset for testset validation
n <- dim(x)[[2]] #number of whole PLC components
pm <- mvr(y~x,data=dataset,ncomp=n,method='oscorespls',scale=T,
          validation='LOO',model=T,x=T,y=T)
MSE <- MSEP(pm,estimate='all',newdata=Ndata);
```

```
cvMSE <- 0
for (i in 0:n) {
cvMSE[i+1] <- MSE[[1]][4*i+2]
               }
  return(pm)
                                        }
```